HOEPLI
TECNICA
PER LA SCUOLA

LUIGI LO RUSSO
ELENA BIANCHI

SISTEMI E RETI

Per l'articolazione **INFORMATICA**
degli Istituti Tecnici
settore Tecnologico

1

HOEPLI

LUIGI LO RUSSO ELENA BIANCHI

Sistemi e reti

Per l'articolazione Informatica degli Istituti Tecnici settore Tecnologico

VOLUME 1

EDITORE ULRICO HOEPLI MILANO

UN TESTO PIÙ RICCO E SEMPRE AGGIORNATO
Nel sito **www.hoepliscuola.it** sono disponibili:
- materiali didattici integrativi;
- eventuali **aggiornamenti** dei contenuti del testo.

Via Hoepli 5, 20121 Milano (Italy)
tel. +39 02 864871 – fax +39 02 8052886
e-mail hoepli@hoepli.it

www.hoepli.it

Indice

Modulo 4 Dispositivi per la realizzazione di reti locali

1 La connessione con i cavi in rame

2 Le misure sui cavi in rame

3 La connessione ottica

4 La connessione wireless

5 Il cablaggio strutturato degli edifici

Lab. 1 Il crimpaggio di un cavo UTP RJ45 CAT 5

Modulo 5 Le reti Ethernet e lo strato di collegamento

1 La tecnologia Ethernet

2 Le collisioni in Ethernet

3 Tipologie di rete Ethernet

4 Dispositivi di rete a livello 2

Lab. 1 Utilizzo di Wireshark

Lab. 2 Il protocollo Ethernet

Lab. 3 Protocollo ARP

Modulo 6 Lo stato di rete e il protocollo TCP/IP

1 Il TCP/IP e gli indirizzi IP

2 Introduzione al subnetting

Presentazione

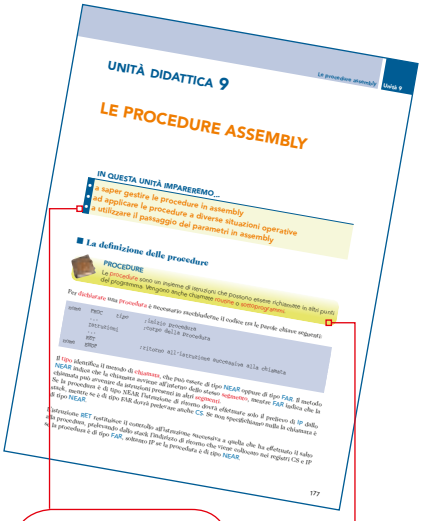
L'impostazione del presente corso in tre volumi è stata realizzata sulla base delle indicazioni ministeriali in merito a conoscenze e abilità proposte per la nuova disciplina Sistemi e Reti. L'opera è in particolare adatta all'articolazione Informatica degli Istituti Tecnici settore Tecnologico, dove la materia è prevista nel secondo biennio e nel quinto anno del nuovo ordinamento.

Il testo, arricchito di contenuti che lo rendono di facile lettura, grazie ai richiami a vocaboli nuovi, spesso in lingua inglese, e ad ampie sezioni di approfondimento, aiuta lo studente a una maggior comprensione degli argomenti, trattati fino ad oggi in modo assai nozionistico. Inoltre, le schede per il laboratorio rappresentano un valido strumento per il rafforzamento dei concetti assimilati attraverso esercitazioni operative.

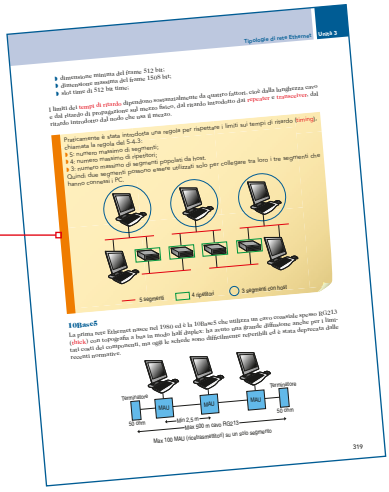
Il primo volume è strutturato in sei moduli suddivisi in unità didattiche che ricalcano le indicazioni dei programmi ministeriali per il terzo anno di studio: lo scopo di ciascun modulo è quello di presentare un intero argomento, mentre quello delle unità didattiche è quello di esporne un singolo aspetto.

Nella pagina iniziale di ogni modulo è presente un indice delle unità didattiche trattate

Indice degli obiettivi che si intendono raggiungere e delle attività che si sarà in grado di svolgere

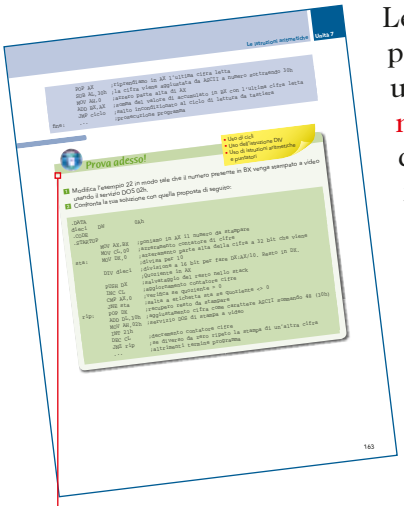


All'inizio di ogni unità didattica sono indicati in modo sintetico i contenuti

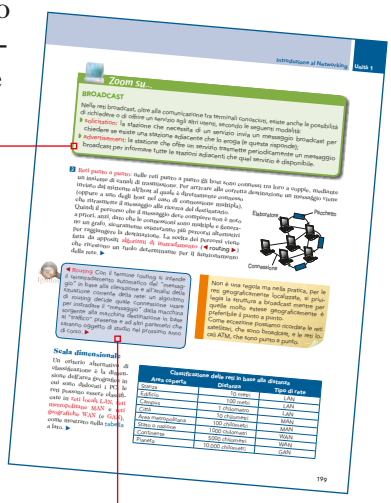


Le osservazioni aiutano lo studente a comprendere e ad approfondire

Il significato di moltissimi termini informatici viene illustrato e approfondito



Le finalità e i contenuti dei diversi argomenti affrontati sono presentati all'inizio di ogni modulo; in conclusione di ogni unità didattica sono presenti esercizi di valutazione delle conoscenze e delle competenze raggiunte, suddivisi in domande a risposta multipla, vero o falso, a completamento, ed infine esercizi di progettazione da svolgere autonomamente.

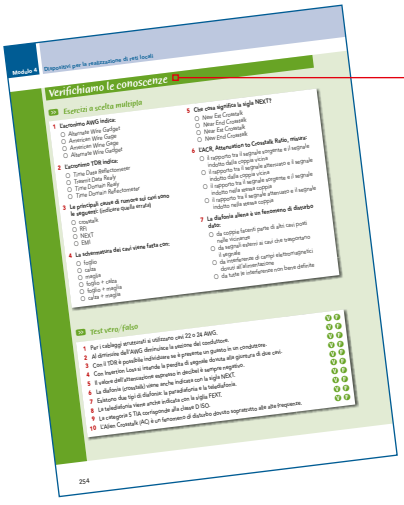


Lo studente può mettere in pratica in itinere quanto sta apprendendo nel corso della lezione

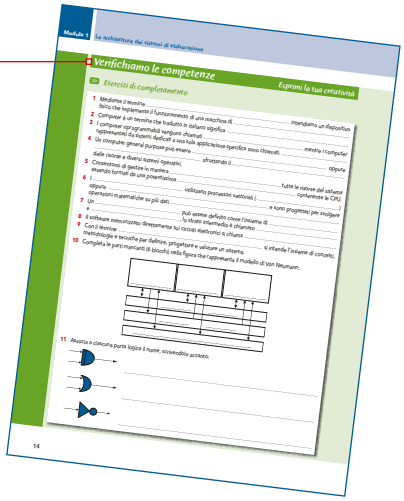
In questa sezione viene approfondito un argomento di particolare importanza

Le parole chiave vengono poste in evidenza e spiegate allo studente

Alla pagina web **http://www.hoepliscuola.it** sono disponibili le risorse online, tra cui unità didattiche integrative, numerosi esercizi aggiuntivi per il recupero e il rinforzo, nonché schede di valutazione di fine modulo.



Per la verifica delle conoscenze e delle competenze è presente un'ampia sezione di esercizi



1 MODULO LE ARCHITETTURE DEI SISTEMI DI ELABORAZIONE

OBIETTIVI

- Conoscere le architetture dei sistemi di elaborazione
- Conoscere il modello di Von Neumann e di Harvard
- Riconoscere il ruolo dei componenti di un sistema di elaborazione (CPU, RAM, I/O, bus)
- Definire i vari tipi di memorie elettroniche (RAM, SRAM, DRAM, ROM, PROM, EPROM, EEPROM)
- Conoscere come indirizzare la memoria
- Conoscere i diagrammi di temporizzazione dei principali cicli per bus sincroni e asincroni
- Conoscere le principali tecniche che migliorano le prestazioni dei computer
- Capire come si sono evolute le tecniche di elaborazione e la gestione della memoria

ATTIVITÀ

- Connettere una CPU sulla motherboard
- Definire il funzionamento e il ruolo del chipset e dei bus di espansione
- Definire e connettere gli adattatori ai tipici bus di espansione
- Definire il ruolo delle periferiche e degli adattatori
- Approfondire e sviluppare la gestione dei dispositivi di I/O

UNITÀ DIDATTICA 1

L'ARCHITETTURA DEL COMPUTER

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere le architetture dei sistemi di elaborazione
- a conoscere il modello di Von Neumann e quello di Harvard
- a riconoscere il ruolo dei componenti di un sistema di elaborazione (CPU, RAM, I/O, bus)

Tipi di computer

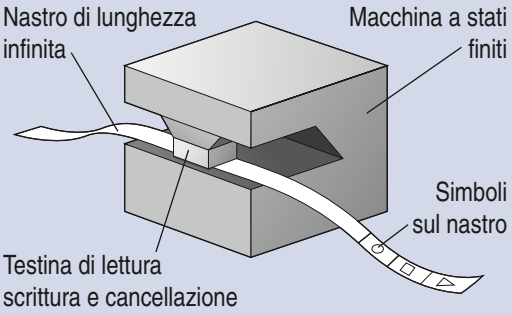
COMPUTER

Con il termine ◀ **computer** ▶ intendiamo un dispositivo fisico che implementa il funzionamento di una macchina di Turing.

◀ Computer Tradotto in italiano significa esattamente elaboratore elettronico, cioè un dispositivo in grado di elaborare un programma sulla base di dati in ingresso (input) e di restituire i risultati dell'elaborazione in uscita (output). Sinonimo di computer è anche sistema di calcolo o calcolatore. ▶

Questa definizione è abbastanza rigorosa ma non definisce le funzioni che svolge un computer e nemmeno come le svolge.
In generale possiamo affermare che esistono diversi tipi di computer a seconda del compito che devono svolgere: siamo passati da macchinari enormi che riempivano intere ali di edifici fino ai moderni computer formati da circuiti integrati delle dimensioni di pochi millimetri, in grado di controllare un robot o un telefono cellulare. Vi sono alcune caratteristiche che accomunano tutti i computer, per esempio essi possiedono almeno una memoria e una CPU, o microprocessore.
Un computer, così come una ◀ macchina di Turing ▶, svolge come compito principale l'esecuzione di programmi: infatti un computer senza un programma da eseguire è del tutto inutilizzabile in quanto verrebbero a mancare le istruzioni operative che indichino al calcolatore come eseguire l'elaborazione.

◀ Macchina di Turing È un macchinario immaginario in grado di manipolare i dati contenuti su un nastro di lunghezza infinita ideato dallo scienziato americano Alan Turing nel 1936 per dare risposta a un problema decisionale posto dal matematico tedesco David Hilbert. Possiamo sintetizzare il problema proposto con la frase seguente: "esiste sempre, almeno in linea di principio, un metodo meccanico (cioè una maniera rigorosa) attraverso cui, dato un qualsiasi enunciato matematico, si possa stabilire se esso sia vero o falso?". La risposta data da Turing fu negativa. ▶



Possiamo effettuare una prima distinzione: esistono computer riprogrammabili dall'utente per diverse applicazioni che vengono chiamati general purpose così come esistono computer rappresentati da sistemi dedicati a una sola applicazione specifica, chiamati special purpose. Per esempio un notebook può appartenere di diritto alla prima categoria, mentre un microcontrollore alla seconda.

Una seconda distinzione è basata sull'accesso condiviso o meno alle risorse hardware: un computer general purpose può essere monoutente oppure multiutente sfruttando il cosiddetto timesharing delle risorse e l'utilizzo di diversi sistemi operativi. Un computer monoutente può essere monotasking oppure multitasking ovvero può eseguire più processi in contemporanea (come avviene, in pratica, in tutti i computer moderni). Ovviamente un computer multiutente è anche multitasking.

Di seguito mostriamo una lista dei principali tipi di computer esistenti classificati in base alle dimensioni, alla potenza e al tipo di uso. Le dimensioni dei sistemi di elaborazione sono progressivamente diminuite nel corso del tempo grazie all'aumento della ◀ capacità di integrazione ▶ degli elementi elettronici di cui sono composti.

◀ Capacità di integrazione In elettronica indica un dato che determina la complessità di un circuito elettronico integrato e quantifica all'incirca quanti transistor sono in esso contenuti:
- SSI (*Small Scale Integration*): meno di 10 transistor;
- MSI (*Medium Scale Integration*): da 10 a 100 transistor;
- LSI (*Large Scale of Integration*): da 100 a 10.000 transistor;
- VLSI (*Very Large Scale Integration*): da 10.000 a 100.000 transistor;
- ULSI (*Ultra Large Scale Integration*): fino a 10 milioni di transistor. ▶

I mainframe

Sono computer utilizzati nelle grandi aziende e, in generale, ovunque sia necessario coordinare una complessa e delicata rete di computer e apparecchiature, in quanto consentono di gestire in maniera centralizzata tutte le risorse del sistema. Sono formati da una potentissima unità centrale contenente le CPU che coordinano le operazioni e le elaborano ad altissima velocità. Proprio per questo motivo necessitano di operare a temperature molto basse per dissipare il calore emesso dai dispositivi. Possiamo affermare che i mainframe sono considerati gli eredi diretti dei primi computer poiché ne rispecchiano ancora in parte la struttura di base, con una macchina centrale cui fanno capo diversi terminali secondari. L'azienda IBM è leader in questo tipo di elaboratori (IBM System Z). ▶

I supercomputer

Sono computer speciali, dotati di elevatissima capacità di elaborazione sviluppati su architetture parallele che utilizzano processori vettoriali (array CPU) oppure GPU (*Graphics Processing Unit*) e progettati per svolgere operazioni matematiche su più dati contemporaneamente. Si differenziano dai mainframe in quanto vengono solitamente destinati a svolgere applicazioni molto specifiche, quali per esempio previsioni meteorologiche a lungo periodo, simulazioni climatiche, calcolo scientifico a elevate prestazioni e altre simulazioni di eventi.
Sono potentissimi e costosissimi, tanto che possono arrivare a costare anche molti milioni di euro. Si trovano solo presso i grandi centri di ricerca che li hanno progettati; uno dei più potenti è senza dubbio il Titan prodotto dalla Oak Ridge National Laboratory (ORNL) del Tennessee (USA) che arriverà a contenere 18.000 processori GPU collegati in parallelo, e sarà l'evoluzione di Jaguar, attualmente uno dei più potenti supercomputer al mondo. ▶

I minicomputer

Sono elaboratori molto simili ai mainframe ma dal costo abbastanza ridotto rispetto a questi ultimi; presenti anche in piccole aziende o in singoli dipartimenti di ricerca, sono di dimensioni paragonabili a un armadio. In sostanza, possono essere collegati a un numero inferiore di terminali rispetto ai mainframe (IBM System i). ▶

I microcomputer

Si tratta dei primi computer dal prezzo economico, utilizzabili da una singola persona. Si diffusero a cavallo tra gli anni '70 e '80, quando la prima generazione di questi dispositivi era in realtà una console da collegare al computer destinata soprattutto agli appassionati, perché di difficile utilizzo. Solitamente, un microcomputer è dotato di un singolo processore con un ingombro generalmente ridotto. Appartengono a questa categoria di computer i personal computer, le console per i videogiochi, i tablet PC e gli home computer.

Gli home computer

Gli home computer rappresentano la seconda generazione di microcomputer che fece il suo ingresso sul mercato nella seconda metà degli anni '70 e divenne comune nel corso degli anni '80, per poi estinguersi entro i primi anni '90 con l'ascesa dei personal computer. Gli home computer, macchine a costo contenuto e di utilizzo prevalentemente domestico, contribuirono largamente a diffondere a livello popolare l'uso del computer e l'alfabetizzazione informatica di vasti strati di popolazione. Un esempio di enorme successo con oltre 10 milioni di macchine vendute fu il Commodore 64.

I personal computer

Il personal computer (◀ PC ▶) è un microcomputer destinato a un utilizzo personale da parte di un singolo individuo. Si distingue da un home computer per maggiore potenza in termini di calcolo e di immagazzinamento dati grazie a una maggiore capacità di memoria e una maggiore velocità di calcolo. Gli attuali PC sono sempre più espandibili e aggiornabili. Sono multimediali e possiedono CPU a multiprocessore.

◀ PC Il termine personal computer è stato coniato per la prima volta da Apple nel 1977 quando venne realizzato l'Apple II, il primo personal computer. Successivamente il termine, ancora attualissimo, venne adottato dall'azienda allora dominante nel settore, la IBM, che lanciò il PC IBM. ▶

Le workstation

Le workstation sono computer di tipo general purpose monoutente dotati di maggiori risorse di elaborazione e costi più alti rispetto ai normali personal computer; sono destinate per uso e compiti professionali. Vengono usati principalmente nei laboratori di ricerca e nelle università e sono adatti sia per il calcolo e la programmazione sia per la grafica avanzata (set virtuali, montaggio video, effetti speciali cinematografici ecc.). Nella figura seguente è indicata una workstation con doppio monitor per applicazioni grafiche.

I microcontrollori

I microcontrollori (in inglese, microcontroller) sono veri e propri computer contenuti in un singolo circuito integrato e dedicati a specifiche applicazioni (special purpose) in sistemi di tipo ◀ embedded ▶. Rappresentano la forma più diffusa e invisibile di computer. Comprendono la CPU, un certo quantitativo di memoria RAM e ROM (può essere PROM, EPROM, EEPROM o FlashROM) e una serie di interfacce di I/O (bus). Le periferiche integrate sono la vera forza di questi dispositivi: si possono avere convertitori ADC e convertitori DAC multicanale, timer/counter, USART, numerose porte esterne bidirezionali. Sono contenuti in quasi tutti gli apparecchi elettrodomestici e la loro capacità di calcolo è molto limitata, solitamente eseguono lo stesso programma (firmware) per tutta la durata del loro funzionamento.

◀ Embedded significa incorporato e identifica un sistema di elaborazione contenuto all'interno di un apparato o di un impianto e solitamente non separabile da esso. Si tratta di sistemi di elaborazione non accessibili direttamente e dedicati esclusivamente al controllo dell'apparato/impianto che lo ospita. Il dispositivo, indifferentemente preprogrammato o (parzialmente) riprogrammabile, è per costruzione specificamente dedicato a un ristretto numero di funzioni di misura e controllo. ▶

I sistemi barebone

Sono personal computer preassemblati, solitamente costituiti da case e scheda madre, pronti per ulteriori personalizzazioni da parte di rivenditori o utenti finali. Spesso di dimensioni ridotte sono monomarca e non consentono aggiornamenti da parte dell'utente. ▶

Il termine deriva dall'inglese e significa, letteralmente, "osso nudo". Indica anche un computer di dimensioni non standard.

I notebook o laptop

Si tratta di computer portatili che possono essere contenuti in una valigetta; attualmente molto diffusi possiedono una batteria che consente un uso mobile. In termini di capacità di memoria e potenza di calcolo possiedono le stesse caratteristiche di un qualsiasi personal computer. Il termine originale di questi calcolatori portatili è ◀ laptop ▶, anche se oggi è poco usato. ▶

◀ Laptop è un termine formato da due parole: in inglese, lap indica il piano che si crea sulle gambe quando si sta seduti, mentre top indica il busto eretto. Il termine laptop sta perciò a indicare un computer da tenere appoggiato sulle gambe. ▶

Attualmente si stanno diffondendo laptop assai particolari chiamati netbook: si tratta di computer portatili di dimensioni inferiori rispetto a un notebook. Nati per essere usati solo per navigare in Internet, seppur privi di lettore CD/DVD, si sono diffusi ampiamente in un mercato in continua espansione.

I computer palmari o PDA

I computer palmari (PDA, *Personal Digital Assistant)* sono computer di ridotte dimensioni, tali da essere portati sul palmo di una mano. Negli ultimi anni sono stati soppiantati dai tablet PC e dagli smartphone.

I tablet PC

I tablet PC sono computer delle dimensioni di una tavoletta e dotati di uno schermo touch screen (per esempio l'iPad di Apple e il Galaxy Tab di Samsung) che consente all'utente di interfacciarsi con il sistema operativo direttamente sullo schermo mediante le dita oppure nei modelli meno recenti con uno speciale pennino. Il tablet PC possiede all'incirca le stesse caratteristiche hardware di personal computer con capacità di input superiori. ▶

Gli smartphone

Gli smartphone possiedono caratteristiche molto simili ai tablet PC: oltre agli apparati elettronici di telecomunicazioni per la ricetrasmissione sono dotati anche di un nucleo elettronico di elaborazione e di una memoria dati con tanto di sistema operativo specifico (per esempio Android, iOS, RIM, Symbian, Phone7) e svariate applicazioni, oltre ovviamente ai consueti dispositivi di input/output di tipo touch screen. ▶

Le console per i videogames

Le console di videogiochi sono computer domestici di tipo special purpose con capacità di calcolo molto elevate per gestire l'elaborazione grafica dei videogiochi. Attualmente rappresentano un settore trainante per lo sviluppo di nuove e più evolute CPU e GPU. ▶

Zoom su...

MACCHINE VIRTUALI

Una macchina virtuale (virtual machine – VM) rende possibile riprodurre il funzionamento di altri sistemi operativi, di telefoni cellulari o di interi computer direttamente sul PC o su altri dispositivi mediante un processo che prende il nome di ◀ **emulazione** ▶.

◀ Emulazione Si tratta di un'operazione che consente, grazie a opportuni software o hardware (chiamati appunto di emulazione) di replicare le funzioni di un determinato sistema su un secondo sistema differente dal primo. ▶

Attraverso una macchina virtuale possiamo, per esempio, testare un particolare software simulandone appieno il funzionamento senza doverlo installare nell'ambiente reale, al riparo da eventuali rischi per il sistema ospitante. Il difetto principale delle macchine virtuali è la lentezza dovuta a un uso eccessivo della memoria. A volte consentono l'emulazione di parti di hardware, anche se non sono in grado di emulare macchine con architettura completamente diversa (a partire dalla CPU). Attraverso una macchina virtuale possiamo così ricreare una sorta di "altro computer" all'interno del nostro calcolatore. La macchina virtuale può essere usata contemporaneamente rispetto al sistema operativo effettivamente installato. Sono evidenti i vantaggi di questa soluzione: in qualsiasi momento possiamo aprire una finestra e al suo interno avviare una applicazione che appare come un computer totalmente diverso e autonomo con un diverso sistema operativo e programmi differenti. ▶

Che cos'è l'architettura di un computer?

Un sistema di elaborazione può essere definito come l'insieme di hardware e software. Il confine tra i due elementi è dato da uno strato intermedio, chiamato firmware, formato dall'insieme dei programmi memorizzati direttamente sui circuiti elettronici (software cablato). ▶

Software
Firmware
Hardware

L'architettura dei computer studia le tecniche con le quali i componenti di un sistema di elaborazione vengono progettati e uniti logicamente tra loro.

ARCHITETTURA

Con il termine "architettura" s'intende l'insieme di concetti, metodologie e tecniche per definire, progettare e valutare un sistema. Questo termine verrà usato in riferimento alla composizione e al funzionamento dei suoi componenti fondamentali.

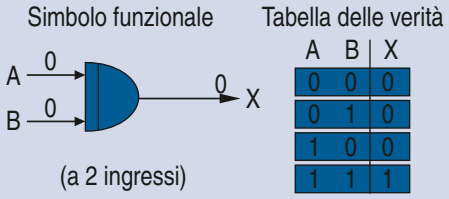
Mediante l'architettura dei calcolatori vengono progettati nuovi sistemi di elaborazione di diversa complessità formati da componenti fisici di tipo elettronico. L'architettura dei calcolatori ha come obiettivo quello di ottenere le migliori prestazioni possibili dai componenti elettronici, a differenza

dell'elettronica che si pone come obiettivo quello di produrre circuiti sempre più veloci ed efficienti. I componenti elettronici che formano un computer possono essere sintetizzati in due categorie principali:

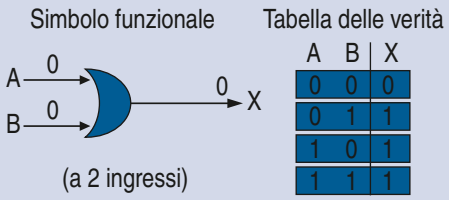
- ◀ porte logiche ▶;
- generatori di segnali.

◀ Porte logiche Sono dei circuiti elettronici elementari in grado di svolgere le operazioni logiche dell'algebra booleana basata sui valori logici VERO o FALSO (TRUE or FALSE). TRUE o FALSE corrispondono al passaggio o al non passaggio di corrente elettrica in tali circuiti e quindi livelli logici binari 0 e 1. Le tre porte principali sono AND, OR e NOT.

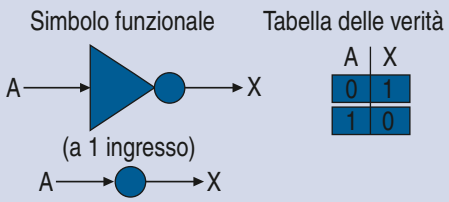
Porta AND:



Porta OR:

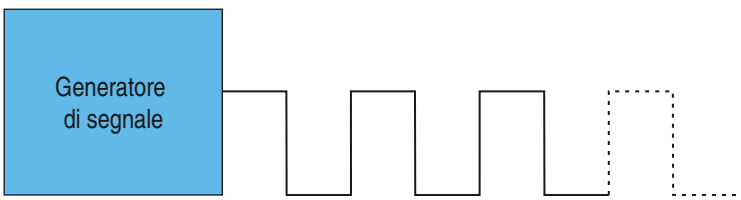


Porta NOT:



Attraverso l'assemblaggio di porte logiche vengono realizzate macchine elementari, suddivise in combinatorie e sequenziali. ▶

I generatori di segnali sono componenti in grado di produrre un segnale periodico utile a sincronizzare gli elementi hardware presenti nei computer.



Possiamo affermare inoltre che accanto alle architetture hardware si sovrappongono le architetture software che comprendono il sistema operativo, cioè il software che si occupa della gestione e del coordinamento delle risorse hardware del computer e delle relative informazioni memorizzate.

Il modello di Von Neumann

Il modello di Von Neumann descrive il comportamento di una macchina che il suo inventore chiamò stored-program computer (esecutore sequenziale dotato di programma memorizzato).

MODELLO DI VON NEUMANN

Si tratta di uno schema a blocchi che illustra il modello di funzionamento di massima di un computer, ideato dal ricercatore ungherese Janus Neumann, naturalizzato americano (John von Neumann) dopo la fuga dal nazismo tedesco, durante la metà degli anni '40 del Novecento. Il modello è stato ideato per la prima volta durante la progettazione del primo computer elettronico, chiamato IAS machine presso l'Institute for Advanced Study (IAS appunto), a Princeton negli Stati Uniti a cavallo tra il 1945 e il 1951. John von Neumann morì nel 1957 a soli 53 anni lasciando molti progetti che verranno sviluppati più tardi presso i centri di ricerca militari di Los Alamos, negli Stati Uniti.

Questo modello è in parte valido ancora oggi come punto di partenza per comprendere, in estrema sintesi, la struttura e il funzionamento di una moderna ◀ CPU ▶.

◀ CPU è l'acronimo di *Central Processing Unit*. È l'unità centrale di un computer, in grado di elaborare ed eseguire milioni di micro-istruzioni al secondo, costituita da un circuito elettronico chiamato microprocessore. L'unità di misura che indica quante istruzioni al massimo è in grado di eseguire in un secondo è il MIPS (*Million Instructions Per Second*, milioni di istruzioni per secondo), dove con "istruzioni" si intendono le istruzioni in *assembly*; siccome esistono istruzioni che impiegano tempi diversi per essere eseguite, non possiamo usare questa unità di misura per confrontare CPU diverse. ▶

È importante sottolineare che tale modello presentava non pochi limiti, e infatti oggi si parla di architetture diverse chiamate, appunto, non Von Neumann.

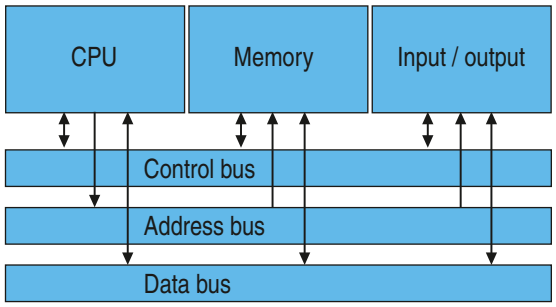
Possiamo così sintetizzare il significato dei nomi della macchina chiamata stored-program computer:

- Computer rappresenta la CPU che compie azioni di elaborazione come, per esempio, prelevare o modificare il contenuto della memoria (memory), prelevare o modificare informazioni dai dispositivi di input/output fornendo informazioni in uscita oppure leggendo informazioni in ingresso (pensiamo per esempio alla tastiera o al monitor). La CPU è in grado di eseguire le proprie azioni in modo sequenziale, cioè una alla volta, a una velocità assai elevata. La misura della velocità della CPU è strettamente legata alla frequenza del clock che indica il numero di azioni al secondo che essa deve compiere (1 GHz rappresenta una frequenza di un miliardo di operazioni elementari (cicli macchina) svolte dalla CPU in un secondo!).

La frequenza del clock è un'unità di misura che identifica quanti cicli macchina esegue la CPU (microprocessore) in un secondo. Tuttavia non dobbiamo commettere l'errore di confonderla con il numero di istruzioni al secondo eseguite dalla CPU (MIPS). Quest'ultima unità di misura ci indica il numero di istruzioni codificate ed eseguite (cioè elaborate) in un secondo. Possiamo affermare che la CPU impiega diversi cicli macchina per eseguire ciascuna istruzione in linguaggio assembly.

- Stored-program indica le istruzioni che la CPU deve eseguire. Tali istruzioni sono collocate (stored) nella memoria del computer e l'insieme delle istruzioni rappresenta il programma (program) che deve essere eseguito. Nella memoria risiedono, oltre alle istruzioni in linguaggio assembly dei programmi in corso di esecuzione, anche i dati sui quali tali programmi operano.

Questi concetti sono sintetizzati nel diagramma a blocchi (riportato a lato) determinato da Von Neumann molti anni fa. ▶

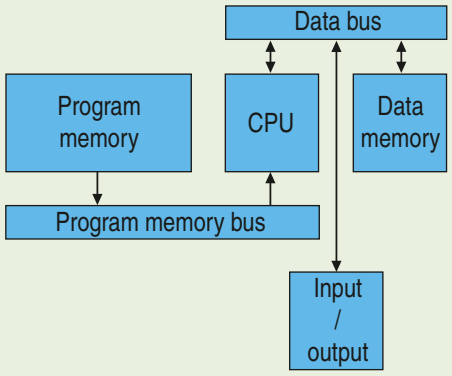


Il diagramma a blocchi evidenzia alcuni componenti principali: la CPU (*Central Processing Unit*), la memoria centrale (memory) e i dispositivi o le interfacce di ingresso e uscita (input/output). In questa rappresentazione schematica i tre bus rappresentano il quarto elemento che consente di trasferire le informazioni, dove le frecce indicano il "verso" delle informazioni nella comunicazione. Le frecce bidirezionali segnalano la possibilità di scrittura e lettura da parte del dispositivo. Non vi è una diretta comunicazione tra memoria e dispositivi di I/O; ogni trasferimento di informazioni dovrà necessariamente passare attraverso la CPU. Dobbiamo tuttavia comprendere che lo schema a blocchi mostrato è una rappresentazione di massima e non descrive la struttura fisica del sistema.

Zoom su...

IL MODELLO HARVARD

A differenza del modello Von Neumann nel quale i dati e le istruzioni condividono assieme la stessa memoria, il modello Harvard dedica due memorie distinte per i dati e per le istruzioni. Si tratta di una soluzione più efficiente e ottimizzabile, ma tuttavia molto più costosa. L'architettura Harvard è un modello applicato alla progettazione di processori specializzati come i DSP (*Digital Signal Processor*) che vengono utilizzati per il trattamento dei dati audio o video. Inoltre molti microcontrollori impiegati in applicazioni industriali utilizzano questa architettura, come i microcontrollori PIC (*Programmable Interface Controller*). ▶



La memoria

La memoria indicata nello schema viene anche denominata centrale o main memory e può essere sostanzialmente di due tipi: RAM (*Random Access Memory*) e ROM (*Read Only Memory*). La memoria RAM è ad accesso casuale (o programmato), è volatile (non permanente) ed è riscrivibile. Viene usata per dati e programmi temporanei. La memoria ROM è di sola lettura, è permanente e i dati in essa contenuti vengono memorizzati dal produttore oppure mediante la scrittura assai più lenta e costosa della lettura. Viene usata per memorizzare programmi non modificabili e per dati di avviamento (BIOS).

La memoria è il dispositivo fisico che ha il compito di immagazzinare le istruzioni e i dati. È organizzata a locazioni o celle, ciascuna delle quali è in grado di memorizzare un byte; potremmo paragonare queste celle a tanti cassetti all'interno dei quali possiamo immagazzinare un solo dato. L'accesso alle celle di memoria avviene attraverso indirizzi numerici (memory address) che identificano la casella in cui si trova il dato. Tali indirizzi si chiamano indirizzi logici in quanto non fanno riferimento allo spazio fisico di memoria centrale in cui verrà caricato e allocato il dato. Il tempo necessario per leggere o scrivere un dato su una cella di memoria è dell'ordine del nanosecondo e si definisce tempo di accesso.

L'input/output

I dispositivi di input/output rappresentano i flussi di dati in ingresso e in uscita da e verso le principali periferiche, attraverso appositi circuiti di interfaccia che consentono di tradurre e interpretare grandezze fisiche e velocità di flusso diverse tra loro.

Possono essere distinti in dispositivi di ingresso e di uscita. I primi consentono al sistema di elaborazione di acquisire ◀ segnali ▶ provenienti dal mondo esterno, mentre i secondi consentono al sistema di elaborazione di inviare segnali al mondo esterno.

La CPU gestisce la comunicazione con tali dispositivi in modo asincrono tramite un segnale particolare chiamato di interrupt (IRQ). I dispositivi di I/O spesso sono solo delle interfacce (◀ controller ▶) con una periferica vera e propria.

◀ Segnali In questo caso i segnali sono rappresentati dai bit che vengono inviati e ricevuti da tali dispositivi. Spesso rappresentano parole, numeri, suoni oppure immagini. ▶

◀ Controller Dispositivo che si affianca a un dispositivo vero e proprio e gestisce il dialogo tra quest'ultimo e il bus a esso collegato attraverso un protocollo di comunicazione rappresentato dall'insieme di regole che governano la comunicazione tra CPU e dispositivo. ▶

I bus

Possono essere paragonati ad autostrade sulle quali scorrono i bit. Le dimensioni di questi elementi variano in relazione alla struttura fisica della CPU. Il bus dati (data bus) consente la trasmissione dei dati dalla CPU agli altri elementi e viceversa (bidirezionale). Il bus indirizzi (address bus) contiene l'indirizzo della cella di memoria o del dispositivo di I/O sul quale o dal quale la CPU ha deciso di operare (monodirezionale). Infine il bus di controllo (control bus) trasporta gli ordini dalla CPU e restituisce i segnali di condizione dai dispositivi, per esempio quando i dati devono essere letti o scritti dai dispositivi e il dispositivo è pronto per riceverli oppure li ha effettivamente ricevuti.

La CPU

La CPU (*Central Processing Unit*) o più semplicemente il processore è l'elemento che svolge l'elaborazione dei dati, ed è rappresentata a livello fisico dal ◀ microprocessore ▶.

◀ Microprocessore A differenza del termine CPU con il quale intendiamo l'elemento funzionale che elabora, con il termine microprocessore o MPU (*Micro Processor Unit*) indichiamo un unico componente elettronico che racchiude al suo interno tutte le funzioni di una CPU. ▶

Come abbiamo detto precedentemente il processore ha il compito di preparare ed eseguire istruzioni. Tali istruzioni derivano spesso da un programma scritto in un linguaggio evoluto che è stato tradotto in un linguaggio comprensibile dalla macchina, chiamato appunto ◀ linguaggio macchina ▶.

◀ Linguaggio macchina Il linguaggio macchina è un linguaggio tramite il quale sono codificate tutte le istruzioni eseguibili dal processore ed è formato da un alfabeto binario, cioè è composto da due soli simboli (0 e 1). Non si tratta però di un vero e proprio linguaggio di programmazione, poiché questa espressione è riservata ai soli linguaggi di alto livello (C++, Java, Visual Basic ecc.) mediante i quali scriviamo programmi non direttamente eseguibili, che richiedono una codifica in linguaggio macchina per mezzo di un compilatore o di un interprete. ▶

Nel linguaggio macchina viene definito il set di istruzioni (instruction set) fondamentali che un processore è in grado di eseguire.

In generale il funzionamento di una CPU può essere così schematizzato:

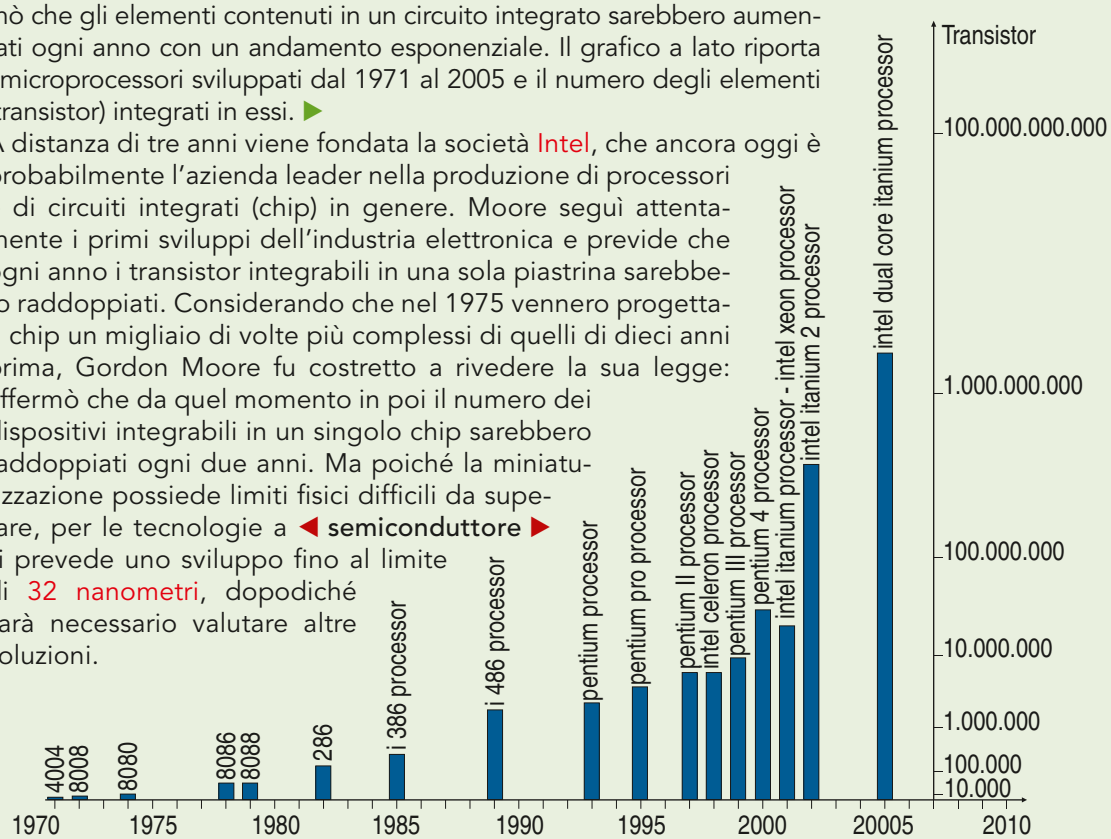
- la CPU estrae le istruzioni dalla memoria, le codifica e le esegue; le istruzioni, in generale, possono comportare la manipolazione o il trasferimento dei dati;
- il trasferimento dei dati tra i vari componenti (per esempio memoria e I/O) avviene mediante i bus di sistema;
- tutte le elaborazione descritte si susseguono in modo sincrono rispetto a un orologio di sistema chiamato Clock;
- durante ogni intervallo di tempo l'unità di controllo, contenuta nella CPU, stabilisce le operazioni da eseguire.

Zoom su...

LA LEGGE DI MOORE

Nel 1965 Gordon Moore, direttore del settore ricerca e sviluppo della Farchild Semiconductor, stimò che gli elementi contenuti in un circuito integrato sarebbero aumentati ogni anno con un andamento esponenziale. Il grafico a lato riporta i microprocessori sviluppati dal 1971 al 2005 e il numero degli elementi (transistor) integrati in essi. ▶

A distanza di tre anni viene fondata la società Intel, che ancora oggi è probabilmente l'azienda leader nella produzione di processori e di circuiti integrati (chip) in genere. Moore seguì attentamente i primi sviluppi dell'industria elettronica e previde che ogni anno i transistor integrabili in una sola piastrina sarebbero raddoppiati. Considerando che nel 1975 vennero progettati chip un migliaio di volte più complessi di quelli di dieci anni prima, Gordon Moore fu costretto a rivedere la sua legge: affermò che da quel momento in poi il numero dei dispositivi integrabili in un singolo chip sarebbero raddoppiati ogni due anni. Ma poiché la miniaturizzazione possiede limiti fisici difficili da superare, per le tecnologie a ◀ **semiconduttore** ▶ si prevede uno sviluppo fino al limite di 32 nanometri, dopodiché sarà necessario valutare altre soluzioni.



◀ Semiconduttore I semiconduttori sono materiali che hanno una resistività e una conducibilità intermedia tra i conduttori e gli isolanti. ▶

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Quale dimensione avevano i primi computer?**
- ○ pochi millimetri
- ○ riempivano ali di edifici
- ○ edifici di 20 piani
- ○ frigorifero

2 **Quale componente accomuna tutti i tipi di computer:** (*due risposte*)
- ○ monitor
- ○ tastiera
- ○ CPU
- ○ memoria
- ○ stampante

3 **La macchina di Turing:**
- ○ è un tipo di computer
- ○ è l'implementazione di un computer
- ○ è un modello di CPU
- ○ è un'architettura

4 **Un microcontrollore appartiene alla categoria:**
- ○ mainframe
- ○ special purpose
- ○ microcomputer
- ○ home computer

5 **Multitasking significa:**
- ○ capacità di integrazione
- ○ dare la possibilità a un utente di aprire finestre in contemporanea
- ○ dare la possibilità a più utenti di aprire finestre in contemporanea
- ○ eseguire più processi in contemporanea

6 **Un dato che indica la complessità di un circuito elettronico integrato è:**
- ○ capacità di integrazione
- ○ dare la possibilità a un utente di aprire finestre in contemporanea
- ○ dare la possibilità a più utenti di aprire finestre in contemporanea
- ○ eseguire più processi in contemporanea

7 **Quali tra i seguenti tipi di computer sono adatti a effettuare calcoli molto complessi in tempi molto brevi:** (*due risposte*)
- ○ personal computer
- ○ workstation
- ○ computer barebone
- ○ supercomputer

8 **Metti in ordine crescente le seguenti scale di integrazione:**
- ○ ULSI .....
- ○ VLSI .....
- ○ MSI .....
- ○ LSI .....

9 **Quali tra i seguenti tipi di computer appartiene ai microcomputer?** (*tre risposte*)
- ○ minicomputer
- ○ personal computer
- ○ home computer
- ○ console dei videogiochi
- ○ workstation

10 **Metti in ordine crescente i seguenti computer in base allo spazio di ingombro:**
- ○ mini computer .....
- ○ mainframe .....
- ○ workstation .....
- ○ netbook .....
- ○ smartphone .....

Verifichiamo le competenze

Esprimi la tua creatività

Esercizi di completamento

1 Mediante il termine ..... intendiamo un dispositivo fisico che implementa il funzionamento di una macchina di .....

2 Computer è un termine che tradotto in italiano significa .....

3 I computer riprogrammabili vengono chiamati ..... mentre i computer rappresentati da sistemi dedicati a una sola applicazione specifica sono chiamati .....

4 Un computer general purpose può essere ..... oppure ..... sfruttando il ..... delle risorse e diversi sistemi operativi.

5 Consentono di gestire in maniera ..... tutte le risorse del sistema essendo formati da una potentissima ..... contenente le CPU.

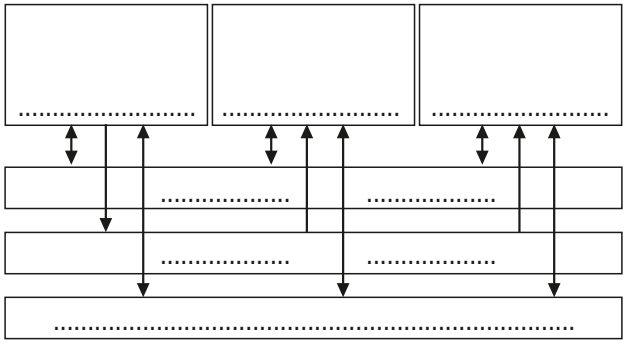
6 I ..... utilizzano processori vettoriali (.....) oppure ..... e sono progettati per svolgere operazioni matematiche su più dati .....

7 Un ..... può essere definito come l'insieme di ..... e ....., lo strato intermedio è chiamato .....

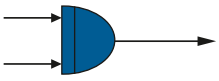
8 Il software memorizzato direttamente sui circuiti elettronici si chiama .....

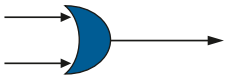
9 Con il termine ..... si intende l'insieme di concetti, metodologie e tecniche per definire, progettare e valutare un sistema.

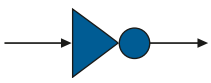
10 Completa le parti mancanti (6 blocchi) nella figura che rappresenta il modello di Von Neumann:



11 Associa a ciascuna porta logica il nome, scrivendolo accanto:


.....


.....


.....

12 Il modello di Von Neumann descrive il comportamento di una macchina che il suo inventore chiamò ..... - ....

13 L'unità di misura che indica quante istruzioni esegue la CPU in un secondo si misura in .....

14 Le operazioni elementari della CPU si chiamano .....

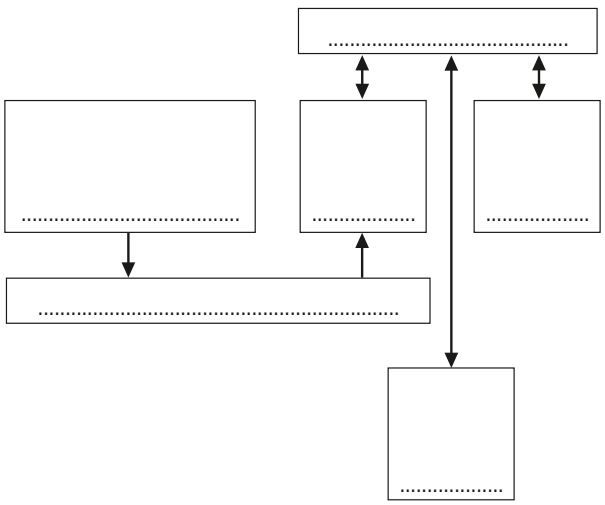
15 La memoria RAM è di tipo ..... mentre la memoria ROM è .....

16 I dispositivi di input/output rappresentano i ..... in ingresso e in uscita da e verso le principali ..... attraverso appsiti circuiti di .....

17 Il ..... è un linguaggio tramite il quale sono codificate tutte le istruzioni eseguibili dal processore.

18 La CPU ..... le istruzioni dalla memoria, le ..... e infine le .....

19 Completa le parti mancanti (6 blocchi) nella figura sottostante che rappresenta il modello di Harvard.



UNITÀ DIDATTICA 2

IL RUOLO DELLA CPU

IN QUESTA UNITÀ IMPAREREMO...

- a riconoscere il ruolo della CPU e le fasi che esegue
- a conoscere il funzionamento di ALU, UC, bus e data path
- a definire il funzionamento e il ruolo del chipset e dei bus di espansione
- a connettere una CPU sulla motherboard

Il microprocessore

Il microprocessore è un circuito integrato (chip) costituito da un ◀ monocristallo di silicio ▶ estremamente puro, sezionato finemente, e infine trattato ad altissime temperature in forni che contengono vari tipi di impurità allo stato gassoso.

◀ Monocristallo di silicio È un cristallo singolo di silicio che possiede un reticolo cristallino continuo (vedi figura a lato), senza interruzioni in tutto il solido. Può essere intrinseco, cioè contenente solo silicio ultrapuro, o drogato mediante l'aggiunta di piccolissime quantità di altri elementi che permettono di modificarne le proprietà di semiconduttore. ▶

Queste impurità devono legarsi alla struttura reticolare del cristallo, influenzandone la capacità di condurre elettricità. Il silicio diventa così un semiconduttore ed è in grado quindi di resistere al passaggio di corrente elettrica in misura maggiore rispetto ai normali conduttori come il rame, ma non tanto quanto gli isolanti.

Il microprocessore svolge fondamentalmente due funzioni: sovrintende a tutte le operazioni del sistema, generando i segnali necessari al funzionamento dei circuiti a esso collegati ed esegue i calcoli aritmetici e logici. Come abbiamo già accennato nell'unità precedente, in base a queste funzioni possiamo suddividere la CPU in due unità fondamentali: l'unità di controllo (CU, Control Unit) e l'unità aritmetico logica (ALU, ArithmeticLogic Unit). Inoltre, esso contiene anche una memoria

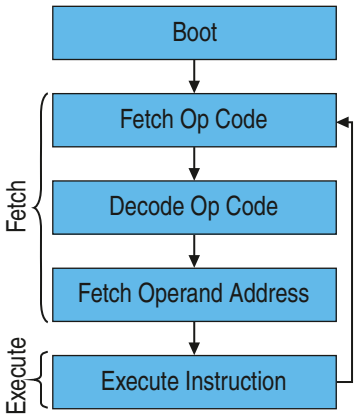
interna formata da diverse celle di memoria dedicate a scopi specifici, chiamati registri di memoria che vengono utilizzati per il controllo dell'esecuzione di un programma. Nella figura a lato un moderno chip di ◀ microprocessore ▶. ▶

◀ Il microprocessore (CPU) è un'elaborata combinazione di transistor che può essere definita circuito integrato. ▶

Il ciclo macchina

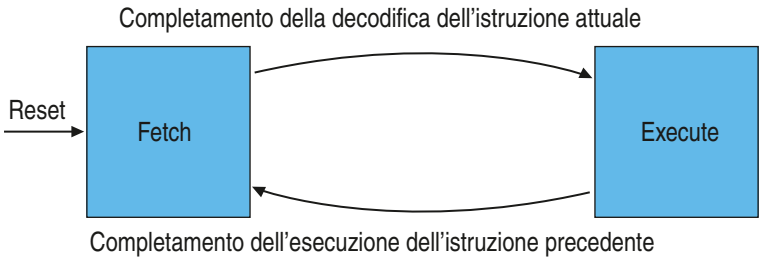
Il funzionamento di una CPU inizia con il prelevamento dalla memoria del codice macchina dell'istruzione da eseguire; tale operazione viene eseguita dalla Control Unit. L'istruzione prelevata viene trasferita in un registro specifico e quindi codificata. Dopo aver codificato, cioè tradotto l'istruzione, la CPU emette i segnali necessari all'esecuzione dell'istruzione. Il procedimento appena descritto attraverso il quale la CPU esegue un'istruzione prende il nome di ciclo macchina, che può essere idealmente suddivisa in quattro parti:

1. Fase di Fetch dell'istruzione;
2. Fase di Decode dell'istruzione;
3. Fase di Fecth degli operandi;
4. Fase di Execute.



Ogni ciclo macchina viene eseguito molto velocemente e in maniera ciclica a sua volta con altri cicli macchina, fino a un ◀ reset ▶, come indicato nel diagramma seguente:

◀ Reset Il termine indica l'azzeramento di un elemento, in questo caso per riportare il processore a una condizione iniziale di Fetch. Il temine *reset*, che significa porre a zero, si contrappone al termine set che indica invece porre a 1. ▶



Fetch dell'istruzione

Il termine Fetch significa prelevamento e identifica la fase in cui la CPU deve reperire l'istruzione da eseguire. In questa fase la CPU deve dialogare con la memoria RAM per ottenere il codice macchina dell'istruzione da eseguire.
Il prelevamento avviene in questo modo: la Control Unit legge la cella di memoria il cui indirizzo è memorizzato in un apposito registro chiamato contatore di programma (PC, Program Counter). Inoltre, dopo aver letto il codice dalla memoria, incrementa il contenuto del registro Program Counter in modo che esso "punti" all'istruzione successiva.

Decode dell'istruzione

La fase di decode rappresenta una fase interna alla CPU durante la quale avviene l'interpretazione dell'istruzione e la preparazione dei dispositivi necessari. In questa fase infatti, il codice macchina dell'istruzione viene codificato in operazioni da eseguire da parte della CPU. L'interpretazione

può avvenire attraverso circuiti logici già predisposti al momento di costruzione del processore secondo la logica cablata dal circuito, oppure attraverso microistruzioni contenute in una apposita parte della ROM secondo la logica microprogrammata. Nel primo caso, le funzioni eseguibili sono prefissate fisicamente e il sistema comporta una certa rigidità mentre nel secondo caso, l'interpretazione avviene cercando nella ROM la sequenza di passi elementari di cui è composta l'istruzione da interpretare.

Fetch degli operandi

In base alla codifica dell'istruzione il processore riconosce se è necessario o meno prelevare dalla memoria o da un registro interno un altro dato per completare l'esecuzione dell'istruzione. In tal caso viene eseguita un'operazione di lettura, dalla memoria o da un registro, chiamata appunto Fetch degli operandi.

Execute dell'istruzione

Nella fase di execute la Control Unit invia segnali che rappresentano opportuni comandi per l'esecuzione.

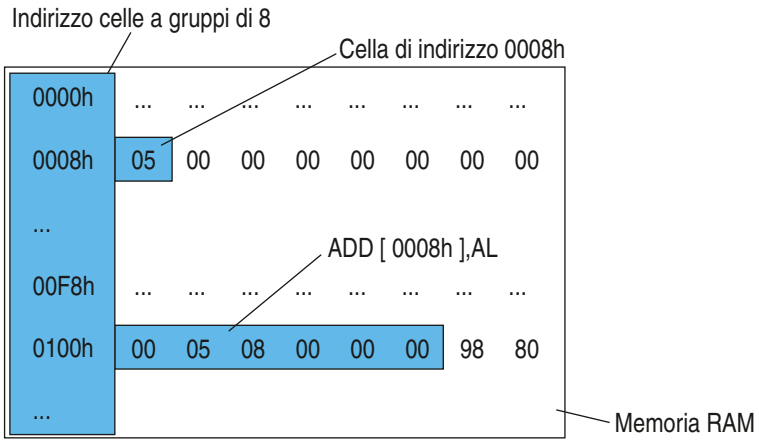
Più in dettaglio, l'esecuzione di ogni istruzione da parte della CPU richiede una serie di passi che, in linea di massima, possono essere così riassunti.

1. Preleva il codice macchina dell'istruzione di indirizzo uguale al contenuto del registro PC e inseriscilo nel registro IR.
2. Incrementa il contenuto del registro PC per "puntare" all'istruzione seguente.
3. Decodifica l'istruzione appena prelevata.
4. Se l'istruzione necessita di operandi determina dove si trovano (memoria oppure registri).
5. Se necessario preleva dalla memoria gli operandi e ponili nei registri della CPU.
6. Esegui l'istruzione.
7. Salva il risultato in un registro o in una cella di memoria.
8. Torna al punto 1.

Per esempio per effettuare la somma tra il contenuto della cella di memoria di indirizzo 0008h e il contenuto del registro AL il processore deve eseguire la seguente istruzione assembly:

```
ADD [0008h],AL
```

> Il numero presenta una lettera "h" posto dopo la cifra meno significativa. Questo indica, in linguaggio assembly, un numero espresso in base di numerazione esadecimale.



Suddividiamo le operazioni svolte dal processore:

1 Fetch

Viene prelevata l'istruzione presente all'indirizzo contenuto nel registro PC (0100h). Si tratta di una sequenza di 6 byte che rappresentano il codice macchina dell'istruzione:

00 05 08 00 00 00

Il codice dell'istruzione viene inserito nel registro IR (Instruction Register).

2 Decode

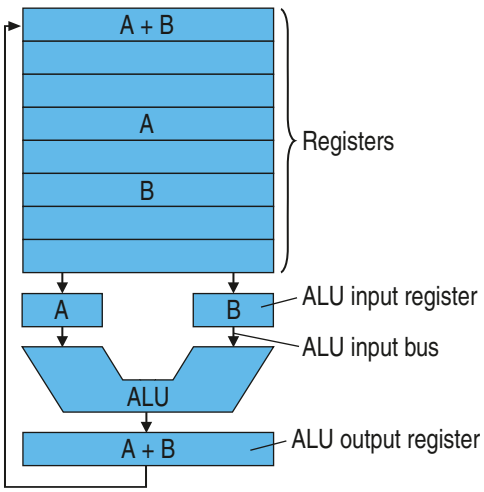
Viene decodificata l'istruzione il cui codice macchina è presente nel registro IR.

3 Fetch degli operandi

Il registro PC viene aggiornato al valore 0008h, necessario per prelevare il contenuto della cella. In tal modo viene recuperato dalla memoria il contenuto della cella di indirizzo 0008h, il cui valore è 05h.

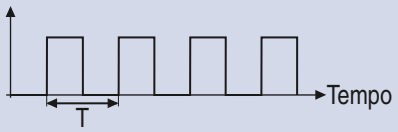
4 Execute

Il valore recuperato dal passo precedente (05h) viene sommato con il contenuto del registro AL (0Ah). Il risultato viene posto nuovamente nella cella di indirizzo 0008h che conterrà 0Fh al termine dell'esecuzione dell'istruzione. L'operazione richiede sicuramente più cicli di data path, soprattutto per la necessità di recuperare operandi dalla memoria. Nella figura a lato è rappresentato il "tipico" data path per una CPU: ▶



Il data path è una sezione della CPU che raggruppa l'ALU e i registri. Il passaggio di due operandi attraverso l'ALU e la memorizzazione del risultato in un nuovo registro viene detto ciclo di data path. Ogni istruzione viene eseguita in uno o più cicli data path, per esempio una divisione necessita di molti cicli per la sua esecuzione. La velocità con cui viene compiuto un ciclo di data path contribuisce significativamente a determinare la velocità della CPU.

◀ Clock Il clock è un circuito di temporizzazione che permette di generare un segnale a onda quadra caratterizzata da una particolare frequenza. Si tratta di un segnale che commuta continuamente i circuiti da un livello basso a uno alto, molti milioni di volte al secondo. La figura seguente mostra un esempio di un tipico segnale di clock:



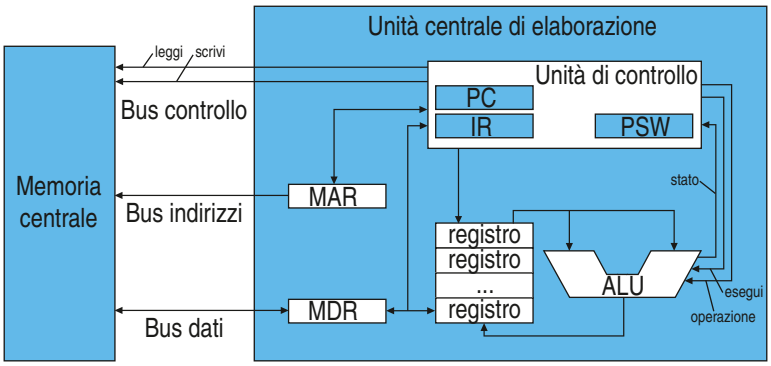
A ogni ciclo i circuiti interni del processore eseguono un'operazione: possiamo dire che il clock sincronizza l'esecuzione di tutte le operazioni della CPU e non solo. Nella formula seguente T è il periodo con cui si ripete il segnale periodico con frequenza pari a: $f = \frac{1}{T}$ Hz. ▶

Ogni ciclo macchina è scandito da un temporizzatore o ◀ clock ▶, cioè un oscillatore al quarzo che emette segnali a intervalli di tempo regolari. Il numero di cicli macchina eseguiti in un secondo di tempo prende il nome di Hertz (Hz). Valgono le seguenti uguaglianze: Hertz (Hz) = 1 ciclo al secondo, kHertz = 1000 cicli al secondo, MHertz= 1 milione di cicli al secondo, GHertz= 1 miliardo di cicli al secondo.

Ogni singola istruzione elementare richiede generalmente più di un ciclo macchina e, pertanto, più di un ciclo di clock.

L'architettura interna della CPU

L'architettura interna di un'unità centrale di elaborazione (CPU), può essere schematizzata come nella figura che segue, secondo il modello di Von Neumann:



Nello schema, oltre alla memoria centrale (RAM), possiamo individuare i seguenti elementi funzionali della CPU:

- unità di controllo (CU, *Control Unit*);
- registro PC (*Program Counter*);
  - registro IR (*Instruction Register*);
  - registro PSW (*Process Status Word*);
- ALU;
- registri generali;
- registro MAR (*Memory Address Register*);
- registro MDR (*Memory Data Register*);
- ◀ BUS ▶ di controllo;
- BUS indirizzi;
- BUS dati.

◀ Il termine BUS identifica un insieme di linee, ciascuna delle quali in grado di trasmettere un bit di informazione tra due elementi elettronici. Il segnale che viene trasferito è di tipo logico, secondo la codifica binaria di un'informazione. ▶

Il bus interno

Si tratta di un bus che collega tutti gli elementi che fanno parte della CPU; nello schema sono individuabili dal colore nero delle frecce. Si tratta di un bus generalmente di controllo, senza distinzione tra dati e indirizzi e non è da confondere con i bus esterni o di sistema che verranno illustrati in seguito.

I registri interni

REGISTRO

Il termine registro indica una particolare cella di memoria contenuta all'interno della CPU. La dimensione dei registri si esprime in bit e dipende dall'architettura specifica della CPU. In generale la funzione dei registri è quella di memorizzare temporaneamente dei dati.

Possiamo paragonare un registro a una lavagna sulla quale viene scritta un'informazione per un breve periodo di tempo. Per memorizzare un'informazione per un periodo più lungo può essere usata una cella di memoria, paragonabile a un quaderno o a un blocco note. Diversamente dalle celle di memoria i registri non possiedono un indirizzo ma un nome specifico. È importante fare

una distinzione tra i registri accessibili dal programmatore (modello di programmazione) e altri che invece risultano inaccessibili al programmatore, in quanto vengono usati direttamente dalla CPU per le proprie operazioni di controllo.
Di seguito sono illustrati i registri interni non accessibili dal programmatore.

MDR (Memory Data Register)

È un registro interno collegato direttamente al bus dati attraverso un ◀ buffer bidirezionale tri-state ▶.
Il registro non è visibile al programmatore e contiene i dati che la CPU vuole inviare o ricevere dalla memoria o dai dispositivi di I/O. Possiamo immaginare il registro MDR come una sorta di memoria di transito dove vengono immagazzinati temporaneamente tutti i dati scambiati con la memoria, prima di essere smistati presso gli altri registri interni.

◀ Buffer bidirezionale tri-state Si tratta di un circuito, che verrà illustrato in seguito, in grado di collegare due componenti elettronici in due direzioni. Il significato di *tri-state* è appunto l'essere in grado di rappresentare oltre i due stati binari tradizionali (0 e 1) un terzo stato chiamato alta impedenza. Esso funziona quindi come un interruttore in grado di staccare i due componenti a seconda del bisogno, per evitare cortocircuiti. ▶

MAR (Memory Address Register)

È un registro interno collegato direttamente al bus indirizzi. Non è visibile al programmatore e contiene gli indirizzi necessari alla selezione della cella di memoria oppure al dispositivo di I/O coinvolto nell'operazione. Durante la fase di fetch di un'istruzione il MAR contiene l'indirizzo della locazione di memoria in cui si trova l'istruzione che deve essere codificata, mentre durante la fase di execute, se si tratta di un'istruzione che fa riferimento alla memoria, contiene l'indirizzo dell'operando che deve essere letto dalla RAM.

IR (Instruction Register)

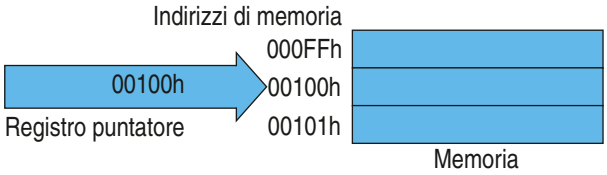
È il registro interno che riceve il codice operativo dell'istruzione prelevata durante la fase di fetch. È invisibile al programmatore e contiene temporaneamente il codice operativo dell'istruzione durante la sua codifica.

Il modello di programmazione

Il modello di programmazione è l'insieme degli elementi accessibili al programmatore, in generale tramite il linguaggio assembly e l'ambiente di debug. È formato dai registri accessibili e dall'ALU. Alcuni registri sono specializzati, cioè svolgono solo una specifica funzione mentre altri sono di uso generale, cioè possono essere usati dal programmatore per un qualsiasi scopo. Le trasformazioni dei dati contenuti nei registri vengono effettuate attraverso elaborazioni dell'ALU.

PC (Program Counter)

È un registro interno accessibile parzialmente dal programmatore che lo può usare per modificare il flusso sequenziale del programma. Il registro PC (nelle CPU Intel si chiama IP, *Instruction Pointer*) contiene, in ogni fase di avanzamento del programma, l'indirizzo di memoria in cui si trova l'istruzione successiva da eseguire. Una modifica del contenuto di questo registro provoca infatti un "salto" all'indirizzo indicato. È un registro di tipo puntatore nel senso che il dato contenuto in esso si riferisce, quindi punta, a un dato che si trova in memoria all'indirizzo corrispondente al valore contenuto nel registro stesso. La figura a lato illustra il concetto enunciato: ▶



Normalmente il contenuto di questo registro viene incrementato automaticamente dalla CU ogni volta che l'esecuzione di un'istruzione è completata, in modo da potere leggere in memoria l'istruzione successiva.

PSW (Process Status Word)

Questo registro interno, chiamato anche registro delle flag (bandiere), non ha un significato nel suo complesso; ciascun bit che lo compone si comporta come una bandierina di segnalazione che fornisce informazioni sul risultato delle operazioni aritmetico-logiche dell'ultima istruzione eseguita: per esempio il bit ZF (*Zero Flag*) segnala se il risultato dell'ultima istruzione aritmetico-logica era zero, mentre il bit CF (*Carry Flag*) segnala se l'ultima istruzione aritmetico-logica aveva un riporto.

> Il termine flag dovrebbe essere indicato al femminile, tuttavia nel gergo più comune e nella maggior parte dei testi di riferimento tende a essere usato al maschile. Pertanto possiamo parlare di flag sia al maschile che al femminile.

L'ALU, a ogni operazione aritmetico-logica, aggiorna il contenuto del registro PSW. Le informazioni contenute in questo registro sono essenziali per la costruzione degli algoritmi dei programmi. Grazie alle flag possiamo infatti realizzare le strutture fondamentali della programmazione (condizione, iterazione, sottoprogrammi): valutando il contenuto dei bit del registro PSW, il codice di un programma può alterare il flusso sequenziale dello stesso per realizzare i costrutti di scelta e iterazione e le chiamate ai sottoprogrammi.
Le flag principali sono quelli descritti di seguito:

- ZERO (ZF): contiene 1 (true) quando il risultato dell'ultima operazione è uguale a 0, mentre contiene 0 (false) quando il risultato dell'operazione è diverso da 0. Nei linguaggi evoluti di programmazione la condizione

  ```
  se (a = b)
  ```

  diventa a livello di linguaggio macchina:

  ```
  a - b
  ```

  Se il risultato della sottrazione è zero significa che i due elementi (a e b) sono uguali, pertanto ZF viene attivato (1).

  > Per evitare confusione conviene usare true (vero) e false (falso) senza usare i termini 1 e 0.

- CARRY (CF): contiene true quando si verifica un riporto dal bit più significativo del risultato durante le quattro operazioni aritmetiche basilari (somma, sottrazione, moltiplicazione, divisione). Se la somma del bit più significativo determina un riporto questo bit non viene sommato al bit successivo ma viene memorizzato nel CF per segnalare che l'operazione nel suo complesso ha determinato un riporto. Viene utilizzato per due diverse ragioni: verificare se un elemento è maggiore o minore di un altro, oppure verificare se un numero non può essere contenuto in un registro.
- OVERFLOW (OF): contiene true quando il segno del risultato dell'ultima operazione aritmetico-logica discorda dal segno degli operandi, cioè quando per esempio il prodotto di due numeri negativi dà luogo a un nuovo valore negativo. Segnala inoltre la presenza di un overflow rispetto alla precisione quando il risultato ha un'ampiezza superiore alla capacità del risultato.
- PARITY (PF): contiene true quando nel risultato dell'ultima operazione aritmetico-logica il numero di bit a 1 del risultato è pari. Viene usato principalmente negli algoritmi di comunicazione per la gestione del rilevamento degli errori di comunicazione.
- SIGN (SF): contiene true quando il risultato dell'ultima operazione aritmetico-logica è negativo, copiando nel flag SF il bit più significativo del risultato che riflette il segno secondo la codifica in complemento a 2.

I registri generali

I registri generali sono registri non specializzati, destinati a ospitare temporaneamente i dati in corso di elaborazione: i dati contenuti nei registri possono provenire dalla memoria o da altri registri, vanno all'ALU per l'elaborazione e da qui tornano ai registri dai quali vengono riportati in memoria.
I registri generali sono accessibili al programmatore e ciascun processore possiede una propria architettura e di conseguenza una propria organizzazione dei registri: noi prenderemo come esempio l'ISA x86 dei processori Intel.

CU (Control Unit)

È il blocco che invia i comandi esecutivi all'ALU in base alla decodifica dell'istruzione, e decide l'incremento dell'indirizzo di memoria contenuto nel registro PC in modo da predisporsi all'esecuzione dell'istruzione.

ALU

È il blocco che esegue le trasformazioni sui dati. Poiché i dati, indipendentemente dal loro significato sono codificati attraverso numeri binari, qualsiasi trasformazione da fare sui dati si riduce a un'operazione aritmetica o logica.
Interviene su due operandi che possono essere contenuti nei registri interni o provenire dalla memoria attraverso l'MDR e produce un risultato che può essere salvato su un registro interno o in memoria attraverso l'MDR.

Zoom su...

LA CPU E L'ESECUZIONE DELLE ISTRUZIONI

Ripercorriamo le fasi che vengono scandite ogniqualvolta la CPU deve eseguire un'ipotetica istruzione:

1. la CPU trasmette il contenuto del registro interno PC, che contiene l'indirizzo della prossima istruzione da leggere dalla memoria, al registro MAR e attiva la linea Leggi;
2. il registro MAR ha lo scopo di mantenere l'indirizzo attivo sulle linee del bus indirizzi;
3. la memoria riceve due informazioni dalla CPU: l'indirizzo della cella (dal bus indirizzi) e il segnale leggi. Questi due segnali informano la memoria che l'operazione richiesta dalla CPU è una lettura;
4. la memoria invia alla CPU il dato presente nella cella indicata tramite il bus dati. A questo punto il dato letto è nel registro MDR;
5. la CPU trasmette il contenuto del registro MDR al registro IR per la codifica dell'istruzione;
6. l'istruzione passa in esecuzione sull'ALU;
7. se l'istruzione prevede la lettura di operandi dalla memoria, questi devono essere caricati dalla memoria ai registri tramite un'operazione di fetch degli operandi, così riassumibile:
   - **7.1** la CPU trasmette l'indirizzo dell'operando da prelevare, nel registro MAR e attiva la linea Leggi;
   - **7.2** la memoria riceve due informazioni dalla CPU: l'indirizzo della cella (dal bus indirizzi) e il segnale leggi. Questi due segnali informano la memoria che l'operazione richiesta dalla CPU è una lettura;
   - **7.3** la memoria invia alla CPU il dato presente nella cella indicata tramite il bus dati. A questo punto il dato letto (operando) è presente nel registro MDR;
   - **7.4** la CPU trasmette al registro destinazione il valore dell'operando presente in MDR;

8 terminata l'esecuzione, la CPU trasmette al registro destinazione il valore prodotto dall'ALU. Se l'istruzione prevede una scrittura in memoria del valore calcolato procede nel modo seguente:
- **8.1** la CPU trasmette l'indirizzo dell'operando da scrivere in memoria nel registro MAR, trasmette il valore dell'operando da scrivere nel registro MDR e attiva la linea Scrivi;
- **8.2** la memoria riceve tre informazioni dalla CPU: l'indirizzo della cella (dal bus indirizzi), il dato da scrivere (dal bus dati) e il segnale Scrivi. Questi due segnali informano la memoria che l'operazione richiesta dalla CPU è una scrittura;
- **8.3** la memoria memorizza nella cella di indirizzo indicato nel bus indirizzi il dato ricevuto dal bus dati;

9 Si ritorna al punto 1 dopo aver aggiornato il valore di PC (prossima istruzione da eseguire).

Tale schema era valido per le CPU degli anni '80; attualmente, grazie all'incremento delle prestazioni dei chip integrati, si sono diffuse nuove e più complesse tecniche denominate generalmente con la dicitura architetture non Von Neumann. Tuttavia il principio di funzionamento resta in linea di massima ancora valido.

ALU (Arithmetic Logic Unit)

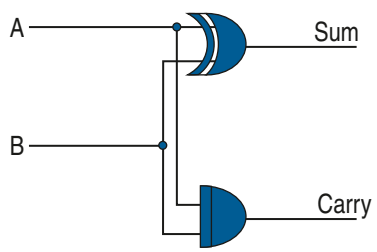
Come abbiamo già affermato, l'ALU è la parte della CPU dedicata alle operazioni aritmetiche e logiche: poiché tutte le istruzioni e i dati sui quali opera sono codificate in forma numerica (binaria), in realtà tutte le elaborazioni devono passare attraverso l'ALU, in quanto qualsiasi operazione può essere ricondotta a un'operazione aritmetico-logica. Per esempio, se vogliamo confrontare due caratteri alfabetici tra loro, siccome vengono codificati secondo la tabella ASCII in ordine progressivo (A = 41h, B = 42h, C = 43h ... a = 61h, b = 62h, c = 63h…), l'ALU effettua una semplice operazione di sottrazione tra i due valori: il settaggio del Flag di Carry indicherà se il primo carattere è maggiore o minore del secondo.

Qualunque operazione logica può essere realizzata disponendo di un insieme di operatori logici, quali per esempio AND, OR, XOR, NOT. Lo stesso concetto può essere esteso anche all'aritmetica individuando come insieme minimo la coppia addizione + negazione.

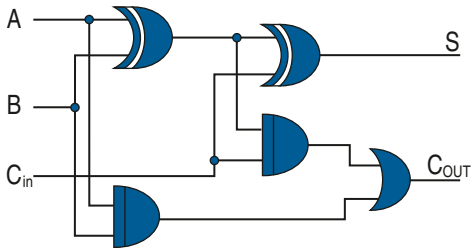
Non dobbiamo confondere la negazione in complemento a 2 con l'operatore logico NOT chiamato complemento a 1. La negazione svolge nell'aritmetica binaria le funzioni di cambio di segno in assenza di un simbolo dedicato al segno.

Ogni altra operazione aritmetica deriva da questa coppia di operatori. La sottrazione si ottiene in sostanza con la negazione del sottraendo addizionato al minuendo, quindi possiamo dire che $a - b$ equivale ad $a + (-b)$; la moltiplicazione è invece un ciclo di addizioni del moltiplicando eseguite per un numero di volte pari al moltiplicatore, la divisione è una ripetizione di sottrazioni del divisore e così via.

Possiamo immaginare l'ALU come costituita solo dalla combinazione di operatori AND, NOT, addizionatori e negatori. Lo schema a lato mostra la struttura di principio di un addizionatore di tipo half adder. ▶



La figura mostra lo schema logico di un sommatore a singolo bit, pertanto per operazioni a 8 bit, 16 bit o 32 bit il circuito dovrà ovviamente essere ripetuto per ciascun bit, tenendo conto dell'eventuale somma del bit di riporto (carry) del precedente (bit di peso superiore). Lo schema a lato mostra un sommatore a tre bit (A, B, C_{IN}) nel quale viene aggiunto il bit del riporto precedente (C_{IN}): ▶ La negazione si ottiene combinando il NOT con l'addizionatore, il complemento a 2 è infatti definito come complemento a 1 + 1.



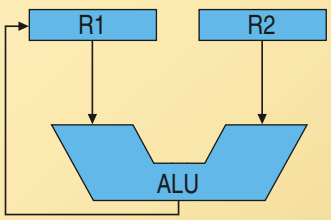
Il comportamento dell'unità aritmetico-logica può essere chiamato binario ad accumulo. Indica che i dati sui quali opera sono espressi in codice binario e il risultato delle operazioni eseguite viene sempre accumulato in uno dei due operandi, come avremo modo di imparare scrivendo alcune righe di codice assembly. Supponiamo per esempio di voler sommare il contenuto di due ipotetici registri generali R1 e R2. Se R1 contiene 5 e R2 contiene 10 effettuando la somma R1 + R2 (5 + 15), il registro posto a sinistra nell'addizione conterrà il risultato che andrà a sostituire il precedente valore in esso contenuto. Possiamo così sintetizzare il concetto enunciato:

```
ADD R1,R2
```

equivale a:

R1 = R1 + R2

In tal modo il registro R1 si comporta da accumulatore; operando in questo modo l'ALU risparmia trasferimenti di dati tra registri ottenendo presestazioni superiori. La figura seguente mostra il concetto di accumulo riportato nell'esempio:



Le architetture RISC e CISC

Le due tecnologie di riferimento per la costruzione di microprocessori sono:
- CISC (*Complex Instruction Set Computer*);
- RISC (*Reduced Instruction Set Computer*).

I processori costruiti secondo l'architettura CISC sono nati con la necessità di avere un numero elevato di istruzioni diverse, di tipo anche complesso, per semplificare il compito dei programmatori e per disporre di programmi più compatti che utilizzino minore memoria. All'interno delle CPU realizzate secondo tale architettura è presente una memoria di tipo ROM che contiene una serie di ◀ microcodici ▶ ciascuno dei quali permette di eseguire all'interno del microprocessore stesso, un'azione elementare.

◀ Microcodici Si tratta del set di operatori elementari che descrive il comportamento di una determinata istruzione assembly. Nel caso delle macchine RISC non esiste microcodice e a ogni ciclo si esegue una sola istruzione. ▶

Per eseguire le istruzioni è necessario prima di tutto trasformarle in una serie di istruzioni scritte in microcodice; in tal modo quelle più semplici richiederanno meno istruzioni in microcodice.

Il concetto costruttivo di un microprocessore RISC è invece la forte riduzione del numero di istruzioni in modo da poter conciliare la velocità del microprocessore con l'esecuzione di queste. Il fine principale della struttura RISC è quello di produrre processori ad alta velocità e dal costo ridotto, data la minore complessità del progetto. Lo svantaggio della tecnologia RISC è il fatto che per essi sono stati sviluppati sistemi operativi a minore diffusione rispetto a quelli sviluppati per i CISC, come Windows. Inoltre, una conseguenza dell'architettura RISC è la maggiore complessità dei programmi: se i processori riconoscono una quantità molto bassa di istruzioni, il programmatore deve sopperire con il software per far svolgere a essi operazioni complesse. In questo caso diventa praticamente obbligatorio studiare, di ogni porzione di codice, il metodo per renderla più veloce; l'ottimizzazione del codice diviene di primaria importanza nello sviluppo dei sistemi di tipo RISC.

I processori CISC più conosciuti sono la famiglia di CPU della Intel: 80286, 80386, 80486, Pentium, Celeron. Attualmente si stanno ormai diffondendo tecnologie ibride, denominate CISC/RISC, come quella dell'architettura Nehalem (IntelCore i).

Le generazioni dei processori

Nel 1971 la Intel commercializzò il primo microprocessore, il 4004, realizzato dal fisico Federico Faggin, che elaborava a una frequenza di 740 kHz.

FEDERICO FAGGIN

È l'inventore del primo microprocessore, l'Intel 4004, di cui ha curato la progettazione e la realizzazione, nonché il fondatore di ZiLOG, presso cui ha progettato lo Z80.

La prima generazione di microprocessori fu quella commercializzata tra il 1971 e il 1973. Fra questi va citato l'8080, il primo processore a 8 bit.

A partire dal 1973 comparve la seconda generazione di microprocessori, prodotti con le più avanzate tecnologie NMOS (MOS di tipo n). Tra essi i più noti sono lo Z80 di Zilog, i 6800 e 6809 di Motorola e l'alternativa Intel costituita dall'8085. L'inserimento dei microprocessori nei personal computer, ha fatto sì che tali circuiti fossero prodotti in grandi quantità, rendendoli più economici.

Nel 1978 sono apparsi i microprocessori a 16 bit, che costituiscono la terza generazione di questo tipo di circuiti. Sul mercato vennero commercializzati l'8086 di Intel con un rendimento dieci volte superiore all'8085, lo Z8000 Zilog e l'MC68000 Motorola.
La diffusione dei personal computer si ebbe quando IBM scelse l'8086 per i suoi PC. Il successo di vendita dei PC portò allo sviluppo di software con molteplici applicazioni in molti campi. Questo fu talmente importante da determinare il fatto che uno dei principali obiettivi dei microprocessori sviluppati successivamente da Intel fosse quello di essere compatibili a livello di software, in modo da poter eseguire qualsiasi programma realizzato per funzionare con l'8086.
Nel 1985, con la comparsa degli MC68020 e MC68030 Motorola e dei 80386 e 80486 Intel, si cominciò a parlare di quarta generazione di microprocessori. Tali circuiti, realizzati con tecnologie CMOS (MOS complementare) consentono di lavorare a frequenze superiori a 50 MHz, con un consumo di energia molto ridotto. La quinta generazione apparve nel 1993 con il processore Intel Pentium, con architetture che consentivano di raggiungere 300 MHz di frequenza. Due anni più tardi con l'avvento del Pentium Pro, Pentium II e Pentium III si raggiunsero velocità ancora superiori fino a circa 1 GHz: era la sesta generazione di microprocessori.

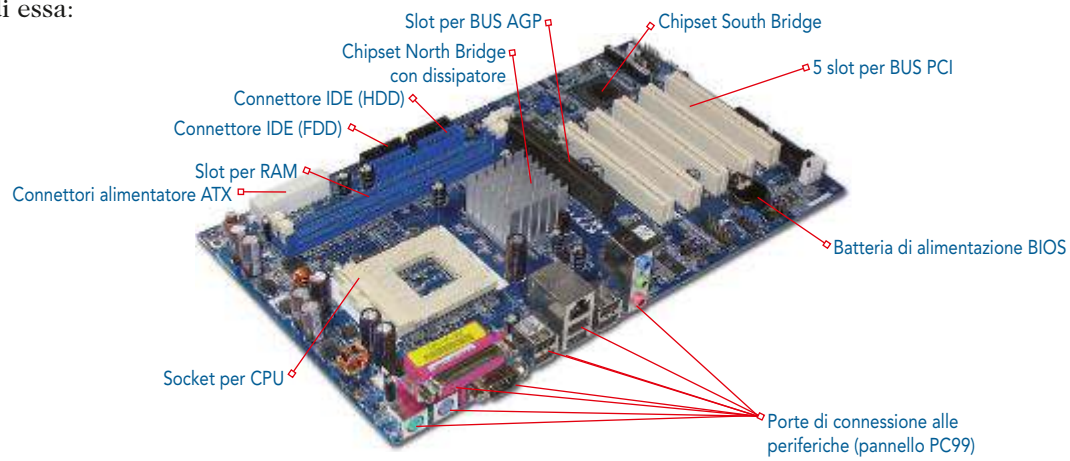
La settima generazione arrivò nel 2000: con l'avvento di un nuovo sistema operativo Windows, (Windows 2000) si affiancò una nuova microarchitettura denominata netburst implementata sul Pentium IV che raggiungeva i 3,75 GHz di frequenza di lavoro.
L'ottava generazione fu quella dei processori della serie Intel Itanium, adatti a workstation e server molto costosi, con una microarchitettura di tipo CISC.
La nona generazione si diffuse nel 2006 con i processori della serie Core: si tratta di processori multicore, con frequenze che si aggirano attorno ai 3 GHz, ma con prestazioni superiori date dall'elaborazione parallela.
La decima generazione è quella dei processori di architettura Intel Nehalem (Core i7 extreme), dotati di più core (processori) che consentono l'esecuzione simultanea di più istruzioni.

La tabella seguente illustra i principali processori rispetto all'anno di riferimento:

Generazione	Processore	Micro-architettura	Bus dati	Bus indirizzi	Frequenza di clock	Anno
Prima	8086	-	16	20	10 Mhz	1980
Seconda	80286	-	16	24	20 Mhz	1982
Terza	80386	i386	32	32	50 Mhz	1985
Quarta	80486	i486	32	32	100 Mhz	1989
Quinta	Pentium	P5	64	32	300 Mhz	1993
Sesta	Pentium Pro, Pentium II, Pentium III	P6	64	36	1 Ghz	1995
Settima	Pentium IV	Netburst	64	36	3,72 Ghz	2000
Ottava	Itanium, Itanium-2	Itanium	64	64	1,66 Ghz	2002
Nona	CPU Core 2	Core (Penryn)	64	64	3 Ghz	2006
Decima	CPU Core i7 extreme	Nehalem	64	64	3,2 Ghz	2010

La CPU nel personal computer

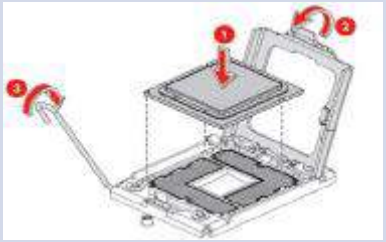
Come abbiamo visto il microprocessore è al centro delle elaborazioni effettuate dal computer; tuttavia non lo abbiamo ancora individuato tra i componenti fisici reali di un computer, per esempio di un personal computer. La CPU si trova sulla motherboard, quel componente che consente di mettere in comunicazione tra loro i diversi componenti e questi ultimi con la CPU. La figura seguente mostra una tipica scheda madre (motherboard) dove sono indicati gli elementi principali montati su di essa:



Come possiamo notare dalla figura precedente, la CPU non è presente in questo schema in quanto può venire montata sull'apposito alloggiamento, in questo caso di tipo Socket ◀ ZIF ▶. Esistono sostanzialmente due tipologie di alloggiamenti per CPU:

- alloggiamento a socket;
- alloggiamento a slot.

◀ ZIF Il termine ZIF è l'acronimo di Zero Insertion Force, e indica un alloggiamento particolare nel quale non è necessario usare la propria forza per collocare la CPU in questa sede. Per facilitare il suo inserimento viene infatti usata una piccola leva che, una volta sollevata, permette l'inserimento del processore senza alcuna pressione e, una volta riabbassata, mantiene il processore sul suo supporto. ▶



L'alloggiamento a slot prevede che il microprocessore venga connesso verticalmente alla scheda madre, mediante uno slot simile a quello usato per i bus di espansione, come illustrato nella figura a lato in cui notiamo una CPU attualmente non più utilizzata. ▶

Le CPU a slot non hanno avuto un grande successo, soprattutto perché il connettore al quale erano unite non poteva resistere a temperature elevate e tendeva a piegarsi per il peso del processore e il calore non adeguatamente dissipato.

ESEMPIO

La procedura riportata di seguito illustra come procedere all'installazione di una CPU nell'apposito alloggiamento Socket ZIF.

1. Per prima cosa dobbiamo individuare il socket nel quale alloggerà la nostra CPU.
2. Poi dobbiamo alzare la levetta accanto allo zoccolino per aprire i connettori, come vediamo nella figura a lato. ▶

3. A questo punto adagiamo il microprocessore nell'apposito alloggiamento stando attenti a far corrispondere l'angolo stondato della CPU con il relativo angolo stondato dello zoccolino, come vediamo nella figura seguente. ▼

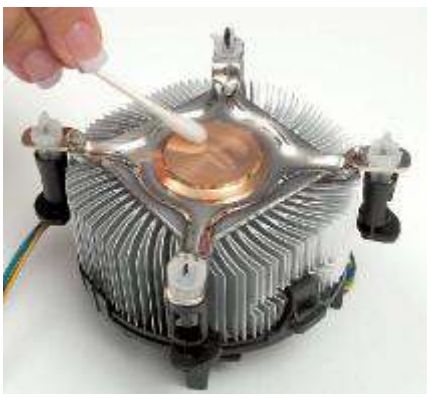
4. A questo punto dobbiamo riportare la levetta alla posizione di partenza per rendere solidale la CPU allo zoccolino che la ospita, chiudendo infine la ghiera metallica. Adesso dobbiamo aggiungere la ventolina di raffreddamento che fungerà da dissipatore di calore. Quindi, per prima cosa ripuliamo per bene la calotta della CPU da eventuali residui di polvere con un detergente alcoolico aiutandoci con un bastoncino tipo cotton fioc, come vediamo nella figura a lato. ▶

5. A questo punto adagiamo sulla calotta della CPU una minima quantità di pasta termica, circa quanto un chicco di riso. ▶

> Questa operazione a volte potrebbe non dover essere effettuata in quanto alcune ventole di dissipazione sono già provviste di pasta termica sul lato che deve aderire alla CPU.

6. Spalmate la pasta termica aiutandovi con un cacciavite piatto, come vediamo nella figura a lato. ▶

7. Ripetete la stessa operazione sul supporto della ventola che dovrà aderire al microprocessore, ripulendolo da eventuali impurità, come vediamo nella figura seguente. ▼



8. Assicurate la ventola alla scheda madre attraverso i quattro fastener nei fori appositi e premete le linguette di plastica in modo deciso verso la scheda madre, fino a quando non sentiremo che i fastener si sono agganciati bene. A questo punto ruotate le linguette nere di plastica in senso antiorario per far sì che i fastener non si possano staccare dalla scheda madre, come vediamo nella figura a lato. ▶

9 Infine dobbiamo connettere il cavetto di alimentazione della ventola di raffreddamento alla scheda madre nella presa denominata CPU FAN, come visibile nella figura a lato. ▶

A questo punto la CPU è connessa e può funzionare.

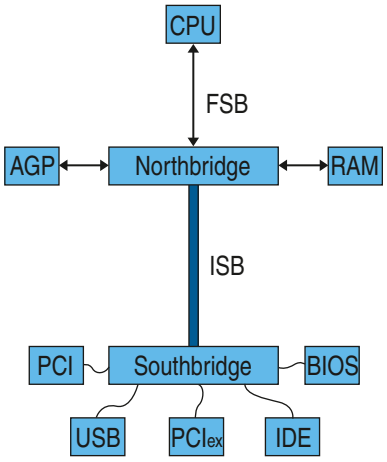
La circuiteria di corredo della CPU

Accanto alla CPU esiste una circuiteria di corredo chiamata Chipset che gestisce tutti i processi del sistema governando lo scambio di dati tra i vari componenti: processore, memorie RAM, bus di espansione verso i dispositivi di I/O (video, audio ecc.).
Il chipset è composto da due differenti circuiti integrati chiamati rispettivamente:
- northbridge;
- southbridge.

Il termine bridge, che in lingua inglese significa ponte, indica proprio il principio di funzionamento di questi dispositivi che consentono di mettere in comunicazione elementi diversi della motherboard. Il primo ha dimensioni maggiori rispetto al secondo ed è in genere quello che viene indicato come il chipset vero e proprio, mentre il secondo è più piccolo e posto generalmente,nella metà inferiore della motherboard, vicino agli slot di espansione (PCI, PCI Express e AGP).

Recentemente i costruttori di microprocessori si stanno orientando per cablare nella CPU anche questi circuiti, per esempio il processore Core i7, contiene un circuito che gli consente di gestire direttamente la memoria RAM.

Il northbridge interfacciava normalmente la CPU con le memorie centrali e il bus di espansione AGP tramite il ◀ FSB ▶, mentre il southbridge interfacciava tutti i diversi bus di espansione. I due chipset sono connessi tra loro mediante un bus denominato ISB (Internal Side Bus). ▶



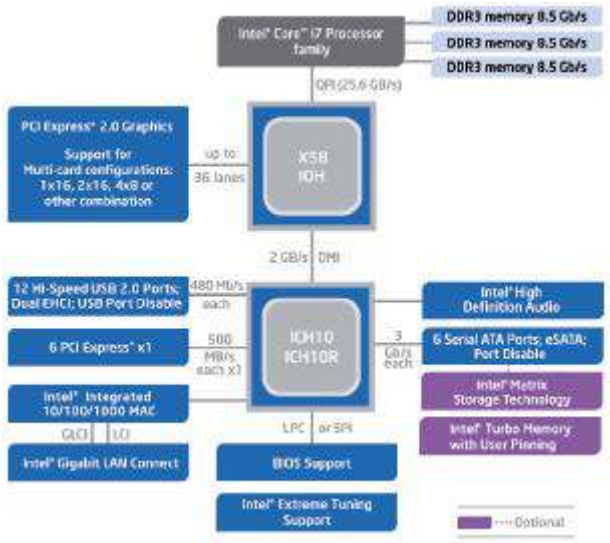
◀ FSB L'FSB (*Front Side Bus*) è il bus che trasporta i dati tra la CPU e il northbridge. In realtà è formato sia da un bus dati che da un bus indirizzi. ▶

La velocità con la quale la CPU trasmette i dati al northbridge si chiama bandwidth (banda passante). Tale valore si calcola moltiplicando i byte della dimensione del bus del processore per la frequenza di clock (cicli al secondo) per il numero di data transfer a ogni ciclo. Per esempio, un sistema con:
- processore a 16 bit (2 byte),
- FSB a 100 MHz,
- due trasferimenti a ciclo,

possiede una bandwidth di:

2 (byte) × 100 (FSB) × 2 (tc) = 400 MB/s (Megabyte al secondo).

Lo schema seguente illustra le connessioni dei due principali chipset nelle moderne CPU, nelle quali le memorie RAM sono collegate direttamente al microprocessore:



Zoom su...

LA POTENZA DEI MICROPROCESSORI

La potenza di calcolo dei microprocessori viene determinata attraverso due unità di misura:

- MIPS (*Million Instruction Per Second*);
- MFLOPS (*Million Floating point Operations Per Second*).

MIPS misura il numero di istruzioni medie necessarie per eseguire elaborazioni. Occorre tuttavia definire elaborazioni standard (benchmark) condivise tra le diverse marche e architetture di processori per poter paragonare CPU diverse mediante questa misurazione. Nonostante la definizione di tali condizioni, quali per esempio la nota benchmark Dhrystone, il valore calcolato non sempre risulta essere indicativo della potenza della macchina. Il valore non sempre è corretto: per esempio a parità di MIPS possiamo avere una CPU nettamente più performante di un'altra, in quanto questo valore non tiene conto della complessità delle istruzioni.
MFLOPS misura invece il numero di operazioni medie in virgola mobile necessarie per eseguire operazioni aritmetiche. Questo valore è più affidabile del precedente anche se è ormai in disuso.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Quali tra i seguenti elementi non è presente all'interno della CPU:

- ○ ALU
- ○ unità di controllo
- ○ memoria RAM
- ○ registri

2 Metti in ordine di esecuzione le seguenti fasi della CPU:

- ○ execute .....
- ○ decode .....
- ○ fetch dell'istruzione .....
- ○ fetch degli operandi .....

3 Reset significa:

- ○ porre a zero
- ○ porre a uno
- ○ ripartire
- ○ caricare

4 Dov'è contenuto l'indirizzo dell'istruzione durante una fase di fetch dell'istruzione?

- ○ nel registro IR
- ○ nel registro PC
- ○ nella memoria RAM
- ○ nell'ALU

5 In quale fase tra le seguenti la CPU può eseguire una scrittura sulla memoria?

- ○ Fetch dell'istruzione
- ○ Decode
- ○ Execute
- ○ Fetch degli operandi

6 Metti in ordine logico crescente le seguenti operazioni che la CPU effettua durante un ciclo completo:

- ○ legge gli operandi dalla memoria e li mette in un registro .....
- ○ incrementa il contenuto del registro PC .....
- ○ decodifica l'istruzione .....
- ○ preleva il codice macchina dell'istruzione .....
- ○ determina dove si trovano gli operandi .....
- ○ salva il risultato in un registro o in una cella di memoria .....
- ○ esegue l'istruzione .....

7 Quale tra i seguenti elementi scandisce i cicli macchina della CPU?

- ○ data path
- ○ Control Unit
- ○ ALU
- ○ clock

8 Come si chiama il bus che collega tutti gli elementi che fanno parte della CPU?

- ○ bus indirizzi
- ○ bus interno
- ○ bus di controllo
- ○ bus dati

9 L'MDR è un registro che si comporta da:

- ○ buffer bidirezionale tri-state
- ○ bus unidirezionale tri-state
- ○ bus bidirezionale tri-state
- ○ buffer unidirezionale tri-state

10 L'IR è un registro che:

- ○ riceve il codice operativo dell'istruzione
- ○ contiene l'indirizzo dell'istruzione successiva
- ○ contiene l'indirizzo dell'istruzione precedente
- ○ si comporta come un buffer tri-state

11 Da quanti cicli sono formate le istruzioni su macchine RISC?

- ○ dipende dal tipo di istruzione
- ○ due cicli
- ○ uno soltanto
- ○ da uno a sei

12 Si misura in MIPS:

- ○ il numero di operazioni per istruzione
- ○ il numero di operazioni in virgola mobile che compongono un'operazione aritmetica
- ○ il numero di istruzioni al secondo
- ○ il numero di cicli macchina al secondo

13 Unisci la generazione dei processori posta a sinistra con la relativa CPU di appartenenza posta a destra:

○ sesta	Itanium	.....
○ settima	Core i7	.....
○ ottava	Core 2	.....
○ nona	Pentium IV	.....
○ decima	Pentium III	.....

14 Quale bus tra i seguenti è connesso alla CPU?

- ○ AGP
- ○ PCI Express
- ○ FSB
- ○ ISB

Verifichiamo le competenze

Esprimi la tua creatività

Esercizi di completamento

1 Il ..... è un chip formato da un ..... di ..... estremamente puro.

2 Fetch dell'istruzione: la ..... legge la cella di memoria il cui indirizzo è memorizzato nel registro chiamato ..... . Dopo aver letto il codice dalla memoria incrementa il contenuto del registro ..... in modo che punti all'istruzione ..... .

3 La fase di decode è fase ..... alla CPU in cui la CPU interpreta il ..... dell'istruzione.

4 Il clock è un circuito di ..... .

5 Il ..... è una sezione della CPU che raggruppa l'..... e i ..... . Il passaggio di due operandi attraverso la ALU e la memorizzazione del risultato in un nuovo registro viene detto ..... .

6 Gli elementi funzionali della CPU sono:

..... registro ..... registro ..... registro .....

..... registri ..... registro ..... registro ..... bus ..... bus ..... bus .....

7 Il modello di programmazione è l'insieme degli elementi accessibili al programmatore, in generale tramite il linguaggio ..... e l'ambiente ..... .

8 Nei processori Intel il registro PC prende il nome di ..... .

9 Il flag di ..... segnala quando il segno del risultato dell'ultima operazione aritmetico-logica discorda dal segno degli operandi.

10 Il flag di ..... segnala quando c'è un riporto dal bit più significativo.

11 Il flag di ..... segnala quando il risultato dell'ultima operazione aritmetco-logica la somma dei bit a 1 è pari.

12 Il flag ..... segnala quando il risultato dell'ultima operazione aritmetico logica è negativo.

13 Il flag ..... segnala quando il risultato dell'ultima operazione è zero.

14 I ..... sono registri non specializzati destinati a contenere i dati in modo ..... .

15 La ..... invia i comandi esecutivi all'ALU e decide l'incremento del contenuto del registro ..... in modo da predisporsi all'esecuzione dell'istruzione.

16 I ..... sono un set di operatori elementari che descrive il comportamento di una determinata istruzione assembly.

17 La velocità con la quale la CPU trasmette i dati al chipset northbridge si chiama ..... e si misura in ..... .

18 Il chipset southbridge interfaccia i ..... .

UNITÀ DIDATTICA 3

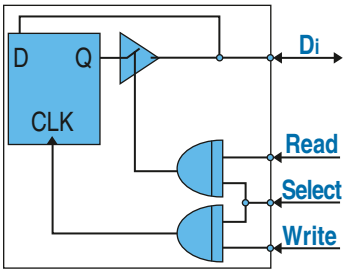
LE MEMORIE

IN QUESTA UNITÀ IMPAREREMO...

- a riconoscere i circuiti fondamentali per la memoria ad accesso casuale (flip-flop, buffer tri-state)
- a definire i vari tipi di memorie elettroniche (RAM, SRAM, DRAM, ROM, PROM, EPROM, EEPROM)
- a comprendere come viene indirizzata la memoria

La memorizzazione dei bit

La memoria contiene informazioni espresse in binario, cioè formate da bit che possono essere 0 oppure 1. Dal punto di vista fisico il bit viene immagazzinato in un dispositivo elettronico chiamato flip-flop, ossia un componente in grado di memorizzare un'informazione binaria. Esistono diverse tecnologie per la realizzazione dei circuiti di memoria, ma il modello che usa i flip-flop ci consente di comprenderne in modo immediato il significato. Lo schema a lato descrive il funzionamento di un singolo elemento di memoria, dal punto di vista elettronico, in grado di immagazzinare un solo bit. ▶



Le operazioni di accesso alle celle di memoria coinvolgono sempre blocchi di byte, quindi dobbiamo immaginare il circuito precedente come parte di altri sette elementi atti a formare un byte. Infatti il singolo bit non è accessibile singolarmente ma solo come parte di un blocco di elementi che vengono processati sempre insieme.

Come possiamo notare vi sono tre ingressi di controllo denominati: Read, Select e Write. Il segnale Di è invece il dato che può essere memorizzato oppure letto dalla memoria. Infatti l'elemento di memoria vero e proprio, cioè quello che memorizza il bit, è formato da un ◀ flip-flop di tipo D ▶.

◀ Flip-flop di tipo D Il termine D deriva dall'inglese *delay* che significa ritardo. Questo circuito mantiene inalterata la sua uscita Q indipendentemente dal valore di D. Il valore D viene copiato in Q quando si ha una commutazione sull'ingresso CLK. ▶

Il funzionamento è assai semplice: quando il segnale sul data bus D_i passa da uno stato a un altro, cioè commuta, la memoria del flip-flop continua a conservare il suo valore. La memorizzazione avviene quando vengono attivati (cioè posti a 1) gli ingressi Write e Select. I segnali Read e Write provengono dal control bus, mentre il segnale di selezione Select proviene dall'address decoder, un circuito di decodifica degli indirizzi che verrà illustrato più avanti. La lettura avviene quando vengono attivati (cioè posti a 1) gli ingressi Read e Select. La logica costruttiva del bus di controllo fa sì che i segnali Read e Write non possono mai essere contemporaneamente attivi.

Ciascun bit di una cella di memoria è collegato a un diverso filo conduttore del data bus in modo tale che tutti i bit vengano trasferiti contemporaneamente. Quando la cella viene letta ogni elemento impone il suo contenuto sul rispettivo filo conduttore del data bus. Allo stesso modo, quando ciascun elemento viene memorizzato nella cella, carica al suo interno il valore del rispettivo bit ricevuto dal filo conduttore del data bus.

I tipi di memoria

Un computer tipicamente contiene differenti tipi di memoria, essenzialmente appartenenti a tre diverse categorie: RAM, ROM e cache (queste ultime verranno trattate ampiamente in seguito).

RAM

Il termine RAM (*Random Access Memory*) indica che in tali memorie è possibile accedere in qualunque locazione di memoria e per qualunque tipo di accesso (lettura o scrittura). La caratteristica principale delle RAM è il fatto che l'informazione in esse contenuta rimane solo quando vengono alimentate (volatilità). Le RAM si suddividono in due sottocategorie chiamate RAM dinamiche (DRAM) e RAM statiche (SRAM).

La RAM dinamica mantiene i dati solo se alimentata. Le DRAM sono infatti caratterizzate da tempi di accesso che variano tra i 20 ns e i 70 ns.

> Esistono molte varianti di DRAM. Una di queste è la EDRAM o DRAM evoluta. Un'altra è la EDO RAM (*Extended Data Output RAM*). La DRAM più conosciuta è quella sincrona (SDRAM), la cui caratteristica principale è la capacità di operare in sincronismo con il clock del bus di sistema. Per quanto detto sinora, le SDRAM sono, attualmente, le uniche memorie in grado di dialogare con i bus a frequenze molto elevate.

La RAM statica è una memoria molto veloce con tempi di accesso attualmente dell'ordine delle decine di nanosecondi (da 5 a 10 ns). A parità di capacità, le SRAM sono molto più costose delle DRAM. A differenza delle DRAM, nelle SRAM non occorre il rinfresco periodico dei dati. Generalmente vengono utilizzate per realizzare memoria cache, come vedremo in seguito.

ROM

Con il termine ROM (*Read Only Memory*) indichiamo una categoria di memorie accessibili solo in lettura. In realtà, attualmente esistono particolari ROM che possono essere anche riscritte. In ogni caso tutte le ROM sono caratterizzate dal fatto che l'informazione in esse contenuta permane anche quando manca la corrente. Le memorie ROM vengono in genere utilizzate per memorizzare programmi e dati di configurazione essenziali per il funzionamento del computer che devono essere memorizzati anche quando il computer è spento. Esistono differenti tipi di ROM:

- ROM non programmabili. Esse vengono prodotte già inglobando il programma o i dati.
- PROM (*Programmable ROM*). Normalmente sono vuote al loro interno e possono essere programmate successivamente attraverso appositi programmatori di PROM, tuttavia non possono essere più modificate nel contenuto.

- EPROM (*Erasable Programmable ROM*). Normalmente sono vuote al loro interno e possono essere programmate attraverso appositi programmatori di EPROM. A differenza delle PROM, la programmazione può avvenire più volte, a patto di cancellare la vecchia programmazione tramite raggi UV (ultravioletti). Sono identificabili per la presenza di una finestrella posta nella parte superiore del circuito, come vediamo nella figura a lato, che permette di ricevere i raggi ultravioletti. ▶
- EEPROM (*Electrical Erasable Programmable ROM*). Sono identiche alle EPROM, dalle quali differiscono solo per il fatto che la cancellazione della vecchia programmazione è realizzata tramite corrente elettrica.

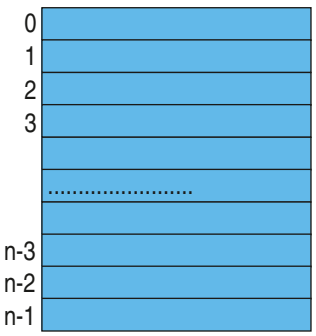
Gli indirizzi delle celle di memoria

L'INDIRIZZO DELLA MEMORIA

Ciascuna cella di memoria può essere selezionata individualmente in quanto è identificata da un indirizzo (memory address). Tale indirizzo è associato alla cella e la rende identificabile in modo univoco. Nei modelli di riferimento si assume la dimensione di una cella di memoria pari a 1 byte.

Possiamo immaginare che tutte le celle di memoria appartengano a una schiera di caselle poste sequenzialmente una dopo l'altra. L'indirizzo di ciascuna cella è definito dalla posizione relativa della cella rispetto alla prima cella. Possiamo definire spiazzamento o displacement, lo spostamento necessario per raggiungere la cella desiderata iniziando dalla prima, che normalmente ha, per definizione, indirizzo 0. Lo schema grafico è riportato a lato. ▶

La quantità di celle presenti nella memoria viene chiamato ◀ spazio di indirizzamento ▶ e il suo valore dipende dalle caratteristiche della CPU, dei bus e della scheda madre. La quantità di celle presenti nella memoria centrale si misura in byte o nei suoi multipli, in quanto ciascuna cella contiene 1 byte.



◀ Lo spazio di indirizzamento è definito dal numero di parole indirizzabili e dipende esclusivamente dal numero di bit dell'indirizzo e non dalla dimensione delle celle di memoria, chiamate "parole di memoria". ▶

Il calcolo dello spazio di indirizzamento si ottiene a partire dal numero di fili conduttori del bus indirizzi individuando tutte le possibili disposizioni con la ripetizione di due elementi (0 e 1). Sapendo che a ogni filo conduttore corrisponde un bit otteniamo il risultato applicando la formula seguente:

$$spazio\ di\ indirizzamento = 2^n$$

Pertanto, l'ultimo indirizzo di memoria valido sarà pari a:

$$indirizzo\ ultima\ cella = 2^n - 1$$

Dove n è il numero di bit del bus indirizzi. È importante sottolineare il fatto che lo spazio di indirizzamento rappresenta il numero massimo di celle indirizzabili; questo non significa che tali celle di memoria debbano essere tutte presenti. Per esempio un processore assai recente dispone di un bus indirizzi a 64 bit, pertanto consente di indirizzare 2^{64} celle di memoria (4 exabyte!); tuttavia la memoria RAM generalmente installata è pari a 8 GB.

Zoom su...

MULTIPLI DEL BYTE

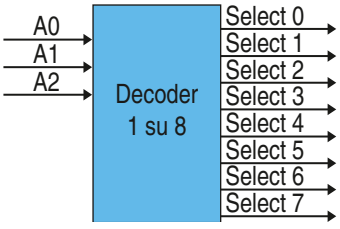
1 kB	(kilobyte)	= 1024 B	
1 MB	(megabyte)	= 1024 kB	= 1.048.576 B
1 GB	(gigabyte)	= 1024 MB	= 1024*1024 kB
1 TB	(terabyte)	= 1024 GB	= 1024*1024*1024 kB
1 PB	(petabyte)	= 1024 TB	= 1024*1024*1024*1024 kB
1 EB	(exabyte)	= 1024 TB	= 1024*1024*1024*1024*1024 kB
1 ZB	(zettabyte)	= 1024 TB	= 1024*1024*1024*1024*1024*1024 kB
1 YB	(yottabyte)	= 1024 TB	= 1024*1024*1024*1024*1024*1024*1024 kB

Per calcolare a mente lo spazio di indirizzamento possiamo ricordare questa semplice regola: 2^{10} consente di indirizzare un kilobyte, mentre a ogni decina successiva si passa al multiplo successivo, così 2^{20} è un megabyte, 2^{30} un gigabyte ecc.

- 10 bit = 1 k celle (1024 celle)
- 16 bit = 64 k celle (65536 celle)
- 20 bit = 1 M celle (RAM dei processori fino all'80286)
- 32 bit = 4 G celle (RAM dei processori fino al Pentium)
- 36 bit = 64 G celle (RAM dei processori fino al Pentium IV)
- 40 bit = 1 T celle (RAM dei processori fino all'Athlon 64)
- 64 bit = 4 E celle (RAM dei processori fino al Core i7)

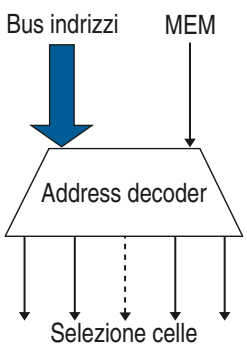
Il circuito di decodifica dell'indirizzo

L'indirizzo della cella di memoria che è stata selezionata dalla CPU è espresso in forma binaria sul bus indirizzi (address bus). È pertanto necessario un circuito di decodifica che traduca la sequenza binaria in un segnale che consenta di selezionare la cella coinvolta nell'operazione, cioè in grado di attivare il segnale Select utilizzato nei flip-flop. Il circuito di decodifica (address decoder) non è altro che un circuito elettronico chiamato decoder. Nell'esempio che segue viene mostrata una decodifica a 3 bit che produce otto segnali di selezione (figura a lato). ▶



$8 = 2^3$, ne consegue che gli indirizzi sono compresi da 0 a 7 (Select0 … Select7)

L'address decoder viene attivato dal segnale MEM proveniente dal bus di controllo. Questo segnale è necessario per evitare conflitti nell'uso del bus indirizzi da parte di altri dispositivi che lo condividono, come per esempio i dispositivi di I/O. In sostanza il segnale MEM attiva la decodifica degli indirizzi di memoria, come vedremo in dettaglio nei relativi diagrammi di temporizzazione (figura a lato). ▶

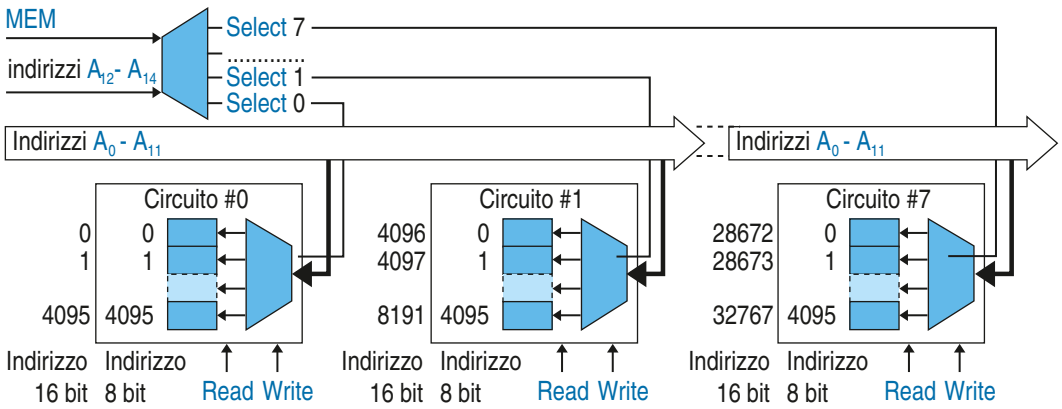


Il circuito di decodifica, integrato nei circuiti di memoria, presenta una decodifica a più livelli. Supponiamo di avere a disposizione un bus indirizzi a 16 bit e di conseguenza uno spazio di indirizzamento di 6 kB e di avere 8 circuiti di memoria ciascuno da 4 kB per un totale di 32 kB di memoria effettiva con indirizzi validi da 0 a 32.767. Ciascun componente di memoria

da 4 kB contiene 4096 celle ed è dotato di un circuito di decodifica integrato in grado di effettuare la selezione della cella mediante 12 ingressi di selezione (2^{12} = 4096) che chiameremo A_0 … A_{11}. Se gli 8 circuiti di memoria venissero collegati insieme al bus indirizzi si verificherebbe un conflitto tra gli indirizzi in quanto allo stesso indirizzo corrisponderebbero 8 celle di memoria. Inoltre avendo collegato i 12 bit del bus indirizzi e avendo lasciato liberi i rimanenti 4 bit, avremo che la cella 0 verrebbe attivata dall'indirizzo 0 ma anche dall'indirizzo 4095, 8192 ecc. Per comprendere meglio il fenomeno consideriamo la seguente tabella nella quale sono indicati gli indirizzi anche in binario (sono evidenziati i 12 bit meno significativi):

Indirizzo decimale	Indirizzo in binario a 16 bit	Cella attivata
0	*0000*000000000000	Prima cella del primo circuito
1	*0000*000000000001	Seconda cella del primo circuito
4095	*0000*111111111111	Ultima cella del primo circuito
4096	*0001*000000000000	Prima cella del secondo circuito
4097	*0001*000000000001	Seconda cella del secondo circuito
8191	*0001*111111111111	Ultima cella del secondo circuito
8192	*0010*000000000000	Prima cella del terzo circuito

Per un componente che riceve dodici bit meno significativi del bus indirizzi (parte in rosso) gli indirizzi 0, 4096 e 8192 coincidono perché i 4 bit più significativi dell'indirizzo vengono ignorati. Per risolvere il problema 3 bit più significativi (A_{12} … A_{14}) vengono collegati a un altro address decoder che possiede 3 ingressi di selezione e 2^3 uscite (8) che possono abilitare uno dei circuiti di memoria (segnale Select). In tal modo possiamo discriminare se la parte bassa dell'indirizzo appartiene al primo, secondo, … *n*-esimo banco di memoria, cioè se è compreso tra gli indirizzi *nnn*000000000000 e *nnn*111111111111:



Come abbiamo visto è necessario che ogni cella di memoria abbia, per poterla raggiungere, un indirizzo specifico. Ogni sistema a microprocessore è caratterizzato da una propria ampiezza di parola rappresentata dal numero di bit che possono essere trasferiti sul data bus tramite un'unica operazione. Normalmente l'ampiezza di parola supera il byte, consentendo al processore di leggere e scrivere più di una cella contemporaneamente. Per esempio il processore i7 possiede una parola di 8 byte ed è in grado di leggere e scrivere contemporaneamente 8 diverse celle di memoria. Possiamo pensare che tale processore possieda un data bus formato da 64 fili conduttori, ciascuno dei quali trasporta un singolo bit (D_0, D_1 … D_{63}). Avremo il seguente schema:
Sui bit D_0-D_7 si indirizzano le celle: 0, 8, 16, 24, 32 …
Sui bit D_8-D_{15} si indirizzano le celle: 1, 9, 17, 25, 33 …
Sui bit D_{16}-D_{23} si indirizzano le celle: 2, 10, 18, 26, 34 …
Sui bit D_{24}-D_{31} si indirizzano le celle: 3, 11, 19, 27, 35 …

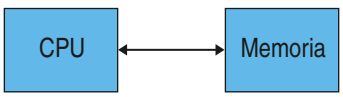
Sui bit D_{32}-D_{39} si indirizzano le celle: 4, 12, 20, 28, 36 …
Sui bit D_{40}-D_{47} si indirizzano le celle: 5, 13, 21, 29, 37 …
Sui bit D_{48}-D_{55} si indirizzano le celle: 6, 14, 22, 30, 38 …
Sui bit D_{56}-D_{63} si indirizzano le celle: 7, 15, 23, 31, 39 …

Solitamente la logica costruttiva del sistema prevede che ciascuna operazione di lettura o scrittura sulle celle di memoria trasferisca un numero di byte pari alla dimensione del bus, indipendentemente dalla cella su cui effettivamente si deve fare l'operazione, anche se essa coinvolge un solo byte. La cella interessata si deve trovare all'interno di questo blocco di memoria.

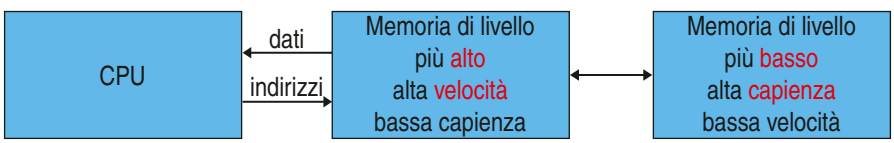
Naturalmente se abbiamo a disposizione un bus dati a 64 bit la memoria dovrà essere adeguata a esso, cioè strutturata in modo tale da presentare le sue celle a blocchi di 8.

La gestione della memoria del PC

Nell'architettura Von Neumann il canale di comunicazione tra la CPU e la memoria è il punto critico del sistema ed è chiamato collo di bottiglia (figura a lato) ▶



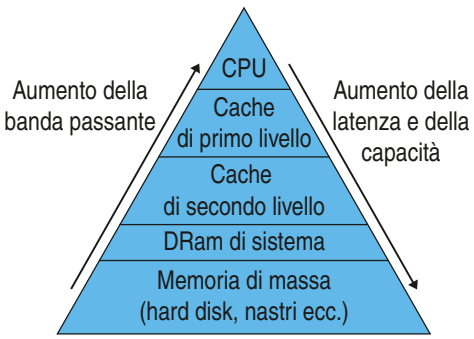
La tecnologia consente di realizzare processori sempre più veloci e memorie sempre più capienti, tuttavia la velocità di accesso delle memorie non è adeguato alla crescita repentina delle CPU. La soluzione ottimale per un sistema di gestione della memoria dovrebbe garantire costi minimi con capacità massime e bassi tempi di accesso al dato. La soluzione che è stata escogitata prevede l'uso di memorie con tempi di accesso diversi, organizzate secondo gerarchie: con l'uso di tecnologie differenti possiamo soddisfare al meglio ciascuno dei requisiti.



La memoria all'interno della scheda madre di un PC è organizzata in livelli gerarchici: ogni livello è caratterizzato da una dimensione crescente e da un tempo di accesso decrescente. La memoria RAM è molto più lenta della CPU, per cui per migliorare le prestazioni vengono combinati tipi di memoria veloce con tipi di memoria più capienti ma lente. Questa tecnica prende il nome di ◀ cache ▶.

◀ Cache Termine di origine francese che significa nascondere (*cacher*). Rappresenta una memoria temporanea che memorizza un insieme di dati che possono essere successivamente recuperati su richiesta, ad altissima velocità. L'origine del nome deriva appunto dal fatto che la memoria cache e il suo utilizzo sono trasparenti al programmatore, quindi nascosti. ▶

Pertanto la CPU legge e scrive i dati in modo diretto sulla cache di primo livello, quindi via via sulle memorie inferiori secondo lo schema a lato. ▶



Quando la cache riceve una richiesta dalla CPU, potrebbe non possedere i dati necessari: si parla in questo caso di ◀ cache miss ▶, cioè di dato mancato nella cache. Solitamente il cache miss avviene quando un dato viene richiesto per la prima volta dalla CPU alla partenza del sistema, oppure quando si tratta di una quantità di dati che non possono essere contenuti all'interno della cache.

◀ Cache miss Si ha un miss (fallimento nell'accesso) se il dato non è presente nel livello immediatamente inferiore e occorre accedere al livello più distante. Si definisce tempo di miss il tempo necessario a sostituire un dato nel livello più vicino con il blocco corrispondente nel livello inferiore, più il tempo necessario per consegnare il dato al livello richiedente, quale per esempio la CPU. ▶

Quando invece il dato viene trovato nella cache si parla di ◀ cache hit ▶, per indicare il successo nella lettura.

◀ Cache hit Si ha un hit (successo nell'accesso) quando i dati richiesti dal livello superiore, come per esempio la CPU, compaiono in qualche blocco nel livello inferiore. Si definisce tempo di hit il tempo necessario a prelevare il dato dal livello più vicino, comprendendo il tempo necessario a determinare se l'accesso è un hit o un miss. ▶

Quindi, le memorie cache fanno da tramite tra la CPU e la memoria RAM compensando il deficit legato alla lentezza della RAM. Esistono diversi livelli di cache, denominate di primo livello, che spesso risiedono all'interno del microprocessore e sono più potenti e veloci della versione integrata sulla motherboard; inoltre mantengono separati dati e istruzioni secondo l'architettura Harvard, consentendo un accesso simultaneo.

Le prestazioni della memoria

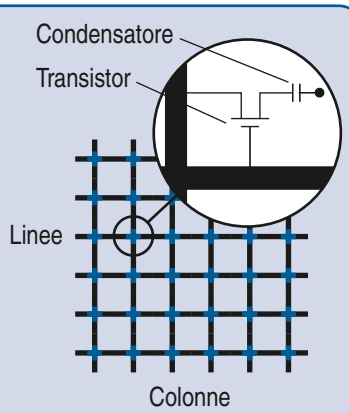
Alcuni fattori responsabili delle prestazioni della memoria sono descritti di seguito.

1. Latenza: è il tempo necessario, espresso in cicli macchina dell'FSB, affinché un dato venga letto dalla memoria. Questo intervallo è in genere impostato a 2 o 3 cicli di clock. Il valore che si ottiene è inversamente proporzionale alle prestazioni della memoria, così un valore ridotto rappresenterà prestazioni migliori.
2. Banda di trasferimento dati: misura la quantità di informazioni al secondo che vengono trasmesse con la memoria. La capacità massima teorica può essere calcolata moltiplicando il numero di bit trasferiti a ogni ciclo di clock per la frequenza di funzionamento della memoria.
3. Frequenza di funzionamento: blocchi di dati trasferiti al secondo. Possono essere impostate frequenze diverse tra l'FSB e le memorie.

L'organizzazione della memoria dinamica di un PC

Nella memoria RAM i dati sono immagazzinati, sotto forma di carica elettrica, in celle costituite da un transistor e da un condensatore. Le celle sono ordinate secondo uno ◀ schema a matrice ▶.

◀ Schema a matrice È uno schema che prevede delle righe (row line) collegate al gate del transistor, e delle colonne (bit line) a loro volta organizzate in banchi. ▶



Poiché i condensatori perdono nel tempo il loro stato di carica è necessario effettuare un'operazione di ricarica periodica chiamata refresh (rinfresco).

Possiamo affermare che la memoria RAM dinamica è la memoria principale del nostro computer. Il tempo di accesso al dato di tale memoria è in realtà formato da un insieme di tempi diversi che possono essere così riassunti:

- CAS Latency Time;
- RAS Precharge Time;
- RAS to CAS Delay;
- Active to Precharge.

Essendo sostanzialmente una tabella a due dimensioni, l'accesso ai dati richiede prima l'identificazione della riga utilizzando un segnale chiamato RAS (*Row Address Strobe*), quindi l'identificazione della colonna con il segnale CAS (*Column Address Strobe*). La pausa tra il segnale RAS e il segnale CAS, chiamato RAS to CAS Delay (ritardo RAS CAS), serve per verificare l'avvenuto indirizzamento della memoria ed è espresso in cicli di clock che devono trascorrere tra l'invio del segnale RAS e il successivo segnale CAS.
Il RAS Precharge Time indica il tempo necessario, espresso in cicli di clock, per caricare i circuiti necessari alla determinazione della riga. Il tempo chiamato Active to Precharge indica il ritardo, sempre espresso in cicli di clock, che deriva dall'indirizzamento consecutivo di due righe diverse all'interno dello stesso circuito di decodifica. Infine, il tempo chiamato CAS Latency Time rappresenta l'intervallo di tempo tra l'istante in cui il comando di lettura giunge a una certa cella di memoria e quello in cui inizia il trasferimento dei dati. La denominazione è dovuta al fatto che, per individuare la cella di memoria, l'indirizzo di colonna viene selezionato sempre per ultimo (tramite il segnale CAS), successivamente a quello di riga.

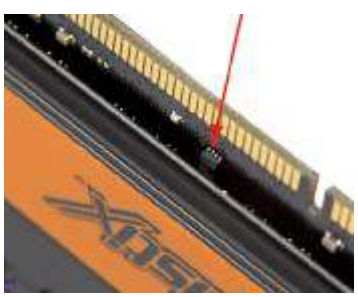
Per migliorare l'efficienza nel recupero dei dati viene usata anche la tecnica burst, mediante la quale viene anticipata la lettura delle colonne adiacenti l'ultima letta, per ridurre il tempo di latenza. In tal modo la memoria cache carica i dati secondo il principio della ◀ localizzazione spaziale ▶.

◀ Localizzazione spaziale Secondo il principio di località, in un dato istante i programmi accedono ai dati utilizzando una porzione ridottissima del loro spazio di indirizzamento. Secondo il principio di localizzazione spaziale è probabile che gli oggetti che si trovano vicini a un oggetto a cui si è fatto riferimento vengano richiesti in tempi brevi, come per esempio nella lettura sequenziale degli elementi di un vettore. La localizzazione spaziale si contrappone alla localizzazione temporale nella quale è invece probabile che un oggetto a cui si è fatto riferimento venga nuovamente richiesto in tempi brevi, come per esempio in cicli nei quali le stesse istruzioni vengono ripetute frequentemente. ▶

Vediamo quali sono i parametri che vengono utilizzati per incrementare le prestazioni della memoria centrale:

1. aumentare la frequenza del FSB;
2. ottimizzare l'architettura del controller della memoria;
3. aumentare la frequenza di lavoro delle memorie;
4. diminuire i tempi di latenza delle memorie;
5. aumentare la quantità di memoria installata.

Esiste infine una memoria EPROM chiamata SPD (*Serial Presence Detect*), solitamente montata sul chip della RAM, che contiene i dati necessari al corretto funzionamento delle memorie DRAM. Contiene informazioni sul modulo di memoria, come per esempio il tipo di memoria, il voltaggio di lavoro, la frequenza di clock, le latenze di CAS supportate, il RAS Precharge, le temperature massime di lavoro, il luogo di produzione ecc. La figura a lato mostra dov'è localizzato un modulo SPD in un chip di memoria. ▶

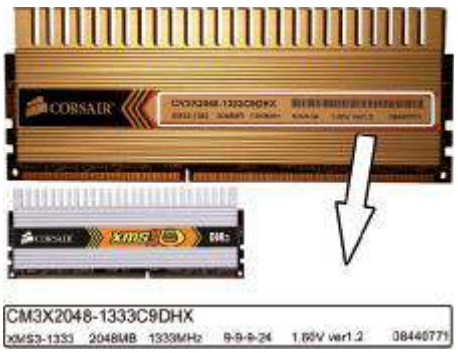


Installando più moduli di memoria, le temporizzazioni di accesso applicate saranno quelle contenute nel chip SPD del modulo più lento.

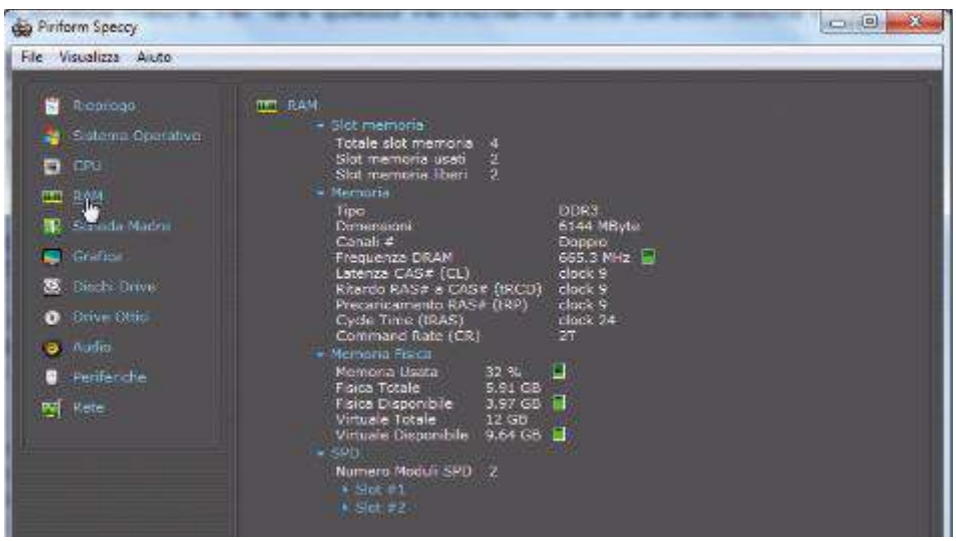
ESEMPIO

La seguente procedura illustra come inserire un chip di memoria RAM nell'apposito alloggiamento.

1. I chip di memoria RAM, oltre a mostrare la sigla della capacità espressa in gigabyte, il tipo (per esempio DDR3) e la frequenza di lavoro, contengono anche altri valori identificati da una sequenza di numeri interi. Per esempio il significato dei valori "2-4-4-8" è il seguente:
   - CAS latency è pari a 2 cicli di clock;
   - RAS a CAS Delay è pari a 4 cicli di clock;
   - RAS Precharge Delay è pari 4 cicli di clock;
   - Active Precharge Delay è pari a 8 cicli di clock.

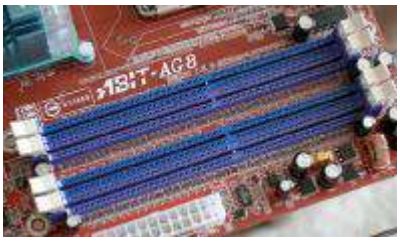


2. Quindi dobbiamo assicurarci che il tipo di chip di memoria RAM sia compatibile con i banchi presenti nella scheda madre. Per fare questo verifichiamo alcune caratteristiche della memoria nel BIOS, come vediamo nella figura a lato. ▶
   Esistono in commercio alcuni software di utilità specifici per la verifica delle informazioni dettagliate sui dispositivi installati nel computer, uno di questi è Speccy, scaricabile gratuitamente dal sito: www.pirifomr.com/speccy. Tale programma,visualizzato nella figura che segue, mostra le informazioni utili alla verifica delle informazioni che riguardano la memoria RAM, la CPU, la scheda madre ecc. ▼



3. A questo punto procuriamoci i chip di memoria adatti (vedi figura a lato): ▶

4. Gli slot nei quali andranno alloggiati i moduli di memoria sono visibili nella figura a lato: ▶



5. Bisogna inserire i moduli nell'alloggiamento facendo una leggera pressione con le dita come mostrato dalla figura a lato: ▶

6. A questo punto non resta altro che chiudere le alette laterali per bloccare i chip agli slot, come mostrato nella figura a lato: ▶

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Il flip-flop di tipo D consente di memorizzare un nuovo valore quando:**

- ○ al suo ingresso viene attivato il segnale Write
- ○ al suo ingresso vengono attivati i segnali Write e Select
- ○ a ogni commutazione dell'ingresso collegato al bus dati
- ○ al suo ingresso viene attivato il segnale Select

2 **Quale tra le seguenti caratteristiche è tipica di una memoria di tipo statico? (*due risposte*)**

- ○ basso tempo di accesso
- ○ economicità
- ○ dimensione ridotta
- ○ necessità del ciclo di refresh

3 **Quale tra le seguenti caratteristiche è tipica di una memoria di tipo dinamico? (*due risposte*)**

- ○ tempi di accesso tra i 5 ns e i 10 ns
- ○ tempi di accesso tra i 20 ns e i 70 ns
- ○ alto costo
- ○ necessità del ciclo di refresh

4 **Collega il tipo di memoria a sola lettura posto a sinistra con la caratteristica relativa posta a destra:**

a) ROM	vuote al loro interno possono essere riprogrammate solo se prima vengono cancellate elettricamente .....
b) PROM	vuote al loro interno possono essere riprogrammate solo se prima vengono cancellate con i raggi UV .....
c) EPROM	inglobano già i dati o il programma e non possono essere più modificate .....
d) EEPROM	vuote al loro interno possono essere programmate successivamente ma soltanto una volta .....

5 **Lo spazio di indirizzamento di un bus indirizzi a 30 bit è:**

- ○ 2 Gbyte
- ○ 1 Mbyte
- ○ 1 Gbyte
- ○ 4 Gbyte

6 **Avendo a disposizione 16 bit, quanti segnali di selezione si possono generare mediante un address decoder?**

- ○ 16
- ○ 256
- ○ 65536
- ○ 65535

7 **Quale segnale attiva l'address decoder della memoria?**

- ○ D0
- ○ Write
- ○ MEM
- ○ Select

8 **Il canale di comunicazione tra CPU e memoria è un punto critico chiamato:**

- ○ collo di bottiglia
- ○ spazio di indirizzamento
- ○ blocco di memoria
- ○ canale critico

9 **Come si chiama l'operazione in cui la memoria trova il dato già nella memoria senza dover scendere di livello?**

- ○ cache Miss
- ○ cache Hit
- ○ cache L1
- ○ cache L2

10 **Associa il fattore responsabile della prestazione della memoria posto a sinistra con la relativa descrizione posta a destra:**

a) latenza	quantità di informazioni trasmesse con la memoria (numero di bit trasferiti a ogni ciclo di clock moltiplicati per la frequenza) .....
b) banda di trasferimento	blocchi di dati trasferiti al secondo .....
c) frequenza di funzionamento	tempo in cicli macchina dell'FSB affinchè un dato venga letto dalla memoria .....

11 **Il tempo di CAS Latency Time rappresenta:**

- ○ il tempo necessario per verificare l'avvenuto indirizzamento della memoria
- ○ il tempo necessario per l'indirizzamento consecutivo di due righe diverse
- ○ il tempo necessario per l'identificazione della colonna
- ○ il tempo necessario per iniziare il trasferimento dei dati dopo il comando di lettura

Verifichiamo le competenze

Esprimi la tua creatività

Esercizi di completamento

1 Le memorie centrali di un computer appartengono a tre differenti categorie: ....., ..., ...

2 Ogni cella di memoria può essere identificata tramite un ..... e il contenuto ha dimensione fissa pari .....

3 Ogni sistema a microprocessore è caratterizzato da una propria ..... rappresentata dal numero di bit che possono essere trasferiti sul ..... tramite un'unica operazione.

4 La memoria di un PC è organizzata a livelli .....: a ogni livello ..... la dimensione della memoria e ..... la velocità.

5 La tecnica ..... migliora l'efficienza nel recupero dei dati; con questa tecnica viene anticipata la lettura delle colonne adiacenti all'ultima letta, per ridurre il tempo di .....

6 Nella memoria RAM i dati sono immagazzinati in celle costituite da un ..... e da un .....

7 Nell'indirizzamento della memoria secondo il modello bidimensionale il segnale ..... identifica la riga e il segnale ..... identifica la colonna. La pausa tra i due segnali è chiamato .....

UNITÀ DIDATTICA 4

IL BUS SECONDO IL MODELLO DI VON NEUMANN

IN QUESTA UNITÀ IMPAREREMO...

- a riconoscere il ruolo del BUS secondo il modello di Von Neumann
- a conoscere i segnali principali del bus di controllo
- a definire la struttura del bus dati e del bus indirizzi
- a indirizzare la memoria e i dispositivi periferici con i segnali dei BUS

La struttura a BUS

BUS

Un bus è un insieme di fili conduttori che possono trasferire segnali elettrici di tipo logico e che nel loro insieme formano la codifica binaria di un'informazione.

◀ BUS Indica che il collegamento elettrico è condiviso da più dispositivi i quali collegano le loro interfacce di ingresso/uscita agli stessi fili. In questo modo un qualunque dispositivo che sia collegato a un bus può potenzialmente mettersi in comunicazione con un qualsiasi altro dispositivo collegato allo stesso bus. ▶

Il ◀ BUS ▶ può essere rappresentato dal percorso di un autobus, i dispositivi sono paragonabili alle fermate presenti sul percorso, mentre le informazioni sono i passeggeri che possono salire sull'autobus da una qualsiasi fermata e scendere a una qualsiasi altra fermata. A differenza di un autobus tuttavia su un BUS può essere rappresentata e trasferita una sola informazione per volta. Questo significa che sul bus ci può essere solo un dispositivo che impone i propri segnali, mentre tutti gli altri tengono le loro uscite funzionalmente scollegate, e un solo dispositivo che accetta i segnali presenti sul bus mentre tutti gli altri tengono i loro ingressi funzionalmente scollegati. Vediamo il significato dei tre BUS definiti secondo il modello di Von Neumann.

Il bus dati (data bus)

Il bus dati consente il trasferimento di informazioni nelle quattro direzioni descritte di seguito.

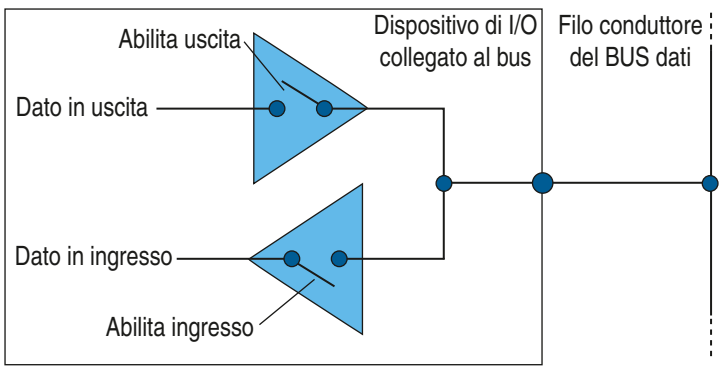
- Da una cella di memoria alla CPU (lettura o load): questa azione viene svolta dalla CPU quando deve ottenere il codice operativo di un'istruzione da eseguire durante una operazione di fetch, oppure quando un'istruzione richiede la lettura dalla memoria di un dato.

- Dalla CPU a una cella di memoria (scrittura o store): quest'azione viene svolta dalla CPU quando un'istruzione richiede la scrittura di un dato in una cella di memoria.
- Da una porta di ingresso alla CPU (acquisizione o in da una porta di I/O): quest'azione viene svolta dalla CPU quando una istruzione richiede un dato da un dispositivo di I/O (per esempio: spostamento del mouse, pressione di un tasto, lettura da disco ecc.).
- Dalla CPU a una porta di uscita (emissione o out verso una porta di I/O): quest'azione viene svolta dalla CPU quando un'istruzione richiede l'invio di un dato a un dispositivo di I/O (per esempio: invio di un dato sul disco, accensione di un pixel sullo schermo ecc.).

Non esiste un trasferimento dati diretto tra una cella di memoria e un'altra. È importante sottolineare questo aspetto perché si tratta di un errore tipico che commettono molti programmatori assembly alle prime armi.

Tutti gli altri elementi periferici, come per esempio le altre celle di memoria e le porte I/O, pur essendo elettricamente collegati al BUS devono mantenersi fisicamente connessi a esso, imponendo uno stato di ◀ alta impedenza ▶ con esso.
Dobbiamo anche stare attenti al fatto che la presenza di un dato sul bus dati è solo temporanea; i dati infatti si avvicendano sequenzialmente in funzione delle azioni svolte dalla CPU. La direzione dei dati su questo BUS cambia in relazione al comportamento dell'elemento connesso a esso, quindi in momenti diversi può comportarsi da ingresso o da uscita nei confronti del data bus (per esempio, una cella di memoria è in uscita durante una lettura e in ingresso durante la scrittura). Per questo motivo diciamo che il bus dati è bidirezionale: infatti tutti i dispositivi che si collegano al data bus devono essere in grado di rimanere funzionalmente disconnessi e di invertire la direzione del flusso dei dati. Per realizzare questa prestazione il pin di collegamento al data bus è internamente collegato a due buffer tri-state collegati in opposizione come illustrato dalla figura seguente:

◀ Alta impedenza È lo stato in cui si trova un dispositivo quando non impone alcun valore logico sul BUS con cui è collegato, risultando in tal modo virtualmente scollegato dalla linea di comunicazione verso l'esterno. Il dispositivo in stato di alta impedenza o tri-state offre un'elevata resistenza al passaggio della corrente e quindi dei segnali elettrici realizzando lo stesso risultato di un circuito aperto. ▶



Lo schema presentato mostra che il dispositivo di I/O è fisicamente scollegato al bus dati solo quando entrambe le abilitazioni dei due buffer sono disabilitate.

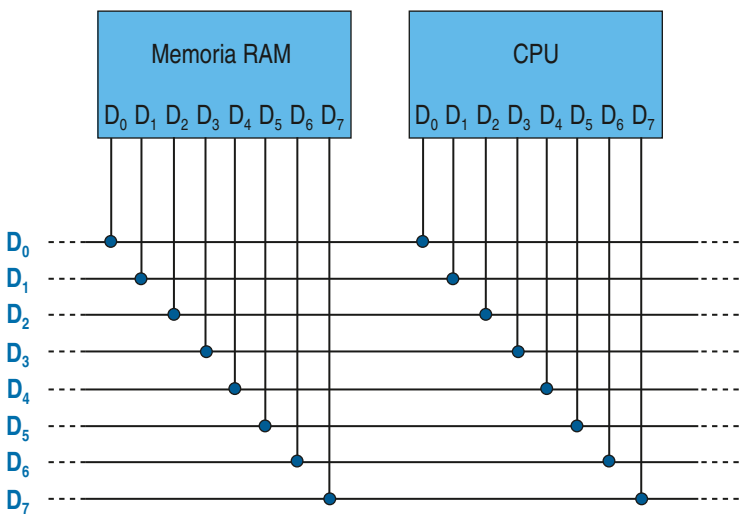
Questa condizione in cui si trovano normalmente tutti i dispositivi, prende il nome di bus flottante (*floating bus*).

Quando viene attivato il segnale chiamato abilita uscita, il buffer consente al dato di essere inviato all'esterno imponendo il segnale corrispondente a questo dato sul bus dati. In questa condizione si può trovare solo un dispositivo alla volta: se più di un dispositivo si trovasse nella suddetta condizione otterremmo un conflitto di bus nel quale due uscite sono collegate insieme, con un risultato imprevedibile.
Quando viene attivato il segnale chiamato abilita ingresso, il buffer consente al dato che si trova sul bus dati di essere copiato all'interno del dispositivo.

Questo schema descrive sia il circuito di interfaccia della CPU sia quello di memorie e I/O ma nell'ambito di ciascun tipo di operazione la direzione del flusso di dati è invertita a seconda che ci si trovi al centro o alla periferia. Per esempio, durante la fase di lettura la memoria riceve l'abilitazione per l'uscita e impone il suo dato interno sul bus dati mentre la CPU riceve l'abilitazione per l'ingresso e copia il dato che è sul bus dati al suo interno.

L'ampiezza del bus dati

L'ampiezza del bus dati è determinata dalla quantità di bit che la CPU è in grado di trasferire attraverso esso in un'unica operazione. Maggiore è l'ampiezza del data bus, maggiore è il parallelismo e quindi la rapidità con cui si svolgono le operazioni.
L'ampiezza non rappresenta un limite per la CPU: si possono infatti svolgere operazioni che coinvolgono un numero di bit superiore all'ampiezza stessa; tuttavia vengono eseguite più lentamente, in quanto rendono necessari più trasferimenti eseguiti in sequenza.
I valori di ampiezza di parola sono cresciuti nel tempo e le MPU attuali possiedono un'ampiezza di parola di 8 byte (64 bit). Lo schema seguente mostra la struttura elementare di un ipotetico bus dati con ampiezza di parola di 8 bit, al quale sono collegati due dispositivi qualsiasi, che per esempio potrebbero essere la memoria RAM e la CPU:

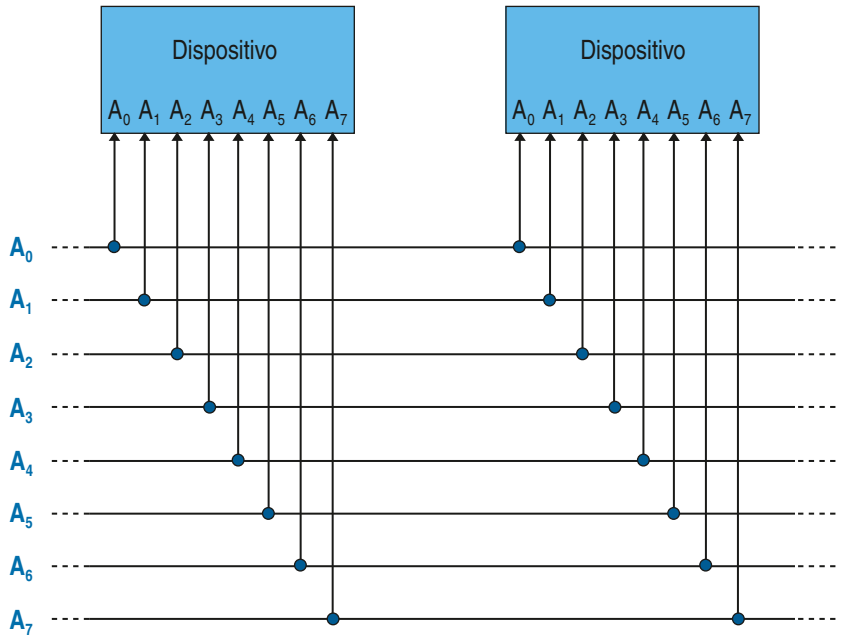


In questo esempio il bus dati è composto da 8 fili conduttori, ciascuno identificato dalla lettera D seguita da un numero che corrisponde al peso della cifra binaria che rappresenta ($D_0 = 2^0$, $D_1 = 2^1$, ... $D_7 = 2^7$). Ciascun dispositivo che si affaccia sul BUS deve avere otto piedini di ingresso o di uscita collegati con i rispettivi fili del bus. È la CPU a governare tutte le operazioni di lettura, scrittura, acquisizione ed emissione, pertanto essa rappresenta sempre uno dei due dispositivi connessi sul bus dati. L'altro dispositivo è rappresentato da una cella di memoria nei primi due casi e da un dispositivo di I/O nei secondi due casi.

Il bus indirizzi (address bus)

Il bus indirizzi (address bus) è di tipo monodirezionale in quanto i segnali che lo compongono escono dalla CPU per raggiungere tutti gli altri dispostivi e formano una sequenza di bit che rappresenta l'indirizzo dell'elemento periferico (cella di memoria o porta di I/O) il quale è coinvolto nella successiva operazione di trasferimento dati tramite il data bus. Durante questa operazione i dispositivi non coinvolti (cioè quelli che hanno un indirizzo diverso da quello specificato) rimangono in stato di alta impedenza (tri-state).

L'indirizzo trasmesso dal bus indirizzi è codificato in binario, pertanto il numero di fili conduttori che lo compongono è pari allo spazio di indirizzamento consentito.



Il bus indirizzi dello schema di cui sopra è un ipotetico BUS formato da 8 fili conduttori, ciascuno identificato dalla lettera A seguita da un numero che corrisponde al peso della cifra binaria che rappresenta ($A_0 = 2^0$, $A_1 = 2^1$, ... $A_7 = 2^7$).

Ciascun dispositivo che si affaccia sul BUS deve avere otto piedini di ingresso e di uscita collegati con i rispettivi fili conduttori del bus indirizzi.
La quantità di celle indirizzabile dal BUS viene chiamata, come abbiamo visto per la memoria, spazio di indirizzamento e la relazione tra il numero di bit del bus indirizzi e la dimensione dello spazio di memoria è data dalla seguente formula:

$$n = \log_2 M$$

Dove M indica lo spazio di memoria o spazio indirizzabile rappresentabile in byte mentre n rappresenta la dimensione del bus indirizzi espresso come numero di bit.
Per esempio dato uno spazio di memoria di 1 kB (1024 byte), vediamo quanti fili sono necessari al bus indirizzi per indirizzarla:

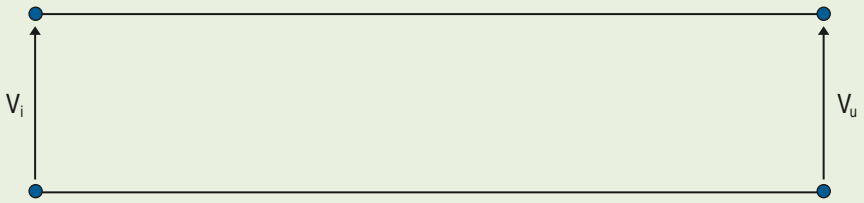
$$10 = \log_2 1024$$

Si vede quindi come un bus indirizzi formato da 10 fili è in grado di indirizzare una memoria di 1 kB.

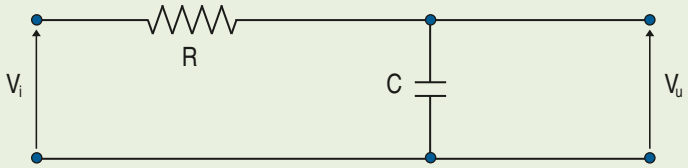
Zoom su...

SIGNIFICATO ELETTRICO DEL BUS DI CONTROLLO

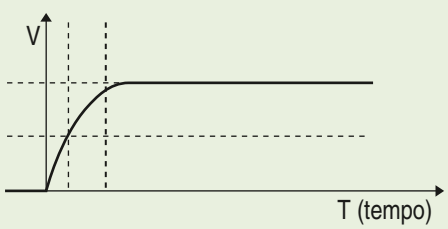
Ciascuna linea del bus dati è formata da due fili conduttori dove in uno scorre il segnale vero e proprio mentre l'altro rappresenta la massa di riferimento. Considerando i due fili come conduttori ideali, otteniamo il grafico seguente:



Il segnale viene trasferito dall'ingresso all'uscita senza distorsione; tuttavia si tratta di un modello inadeguato in quanto i segnali del BUS si evolvono rapidamente nel tempo al variare della resistività del conduttore e degli effetti capacitivi che si presentano tra due conduttori separati da un isolante. Possiamo descrivere meglio il fenomeno introducendo una resistenza in serie (R) che simula la resistività del filo, e un condensatore (C) in parallelo che simula la capacità tra i due fili:



Esaminiamo adesso come si comporta il circuito ponendo ai suoi capi un segnale rappresentato da un gradino. Il segnale in ingresso (V_i) varia da 0 a 1. Il segnale di uscita (V_u), per il fenomeno di accumulo delle cariche segue un andamento esponenziale che lo porta asintoticamente al valore massimo (1). Ipotizziamo il livello di transizione tra 0 e 1 posto circa a metà dell'ampiezza complessiva del segnale: possiamo notare che il segnale di uscita raggiunge il livello di transizione in leggero ritardo (distorsione) rispetto al momento in cui il segnale è stato generato. Il segnale del BUS può pertanto essere campionato da un dispositivo solo quando il livello di transizione è stato superato. A questo provvede il segnale di controllo che abilita il campionamento solo quando il bus è stabile. ▶
La presenza del clock serve quindi a dare tempo ai segnali sul bus per stabilizzarsi ed essere campionati.



Il bus di controllo (control bus)

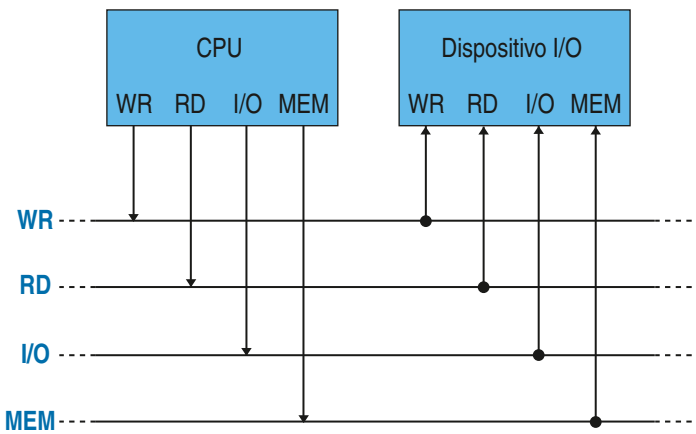
Tutte le operazioni che svolge la CPU vengono sincronizzate mediante il bus di controllo. Non si tratta di un BUS vero e proprio ma di un insieme di singoli segnali logici, ciascuno dei quali ha un significato ben preciso. Tutti i segnali di uscita del bus di controllo rappresentano comandi esecutivi che informano le periferiche su quello che devono fare.

> Tutti i segnali del bus di controllo sono sincroni rispetto a un segnale di clock comune che fa avanzare le azioni della CPU con una cadenza fissata dalla sua frequenza.

Consideriamo i quattro segnali di uscita principali di un ipotetico bus di controllo.

- RD (Write): segnale di controllo che indica l'operazione di lettura da una cella di memoria o da una porta di I/O verso la CPU.
- WR (Read): segnale di controllo che indica l'operazione di scrittura su una cella di memoria o su una porta di I/O dalla CPU.
- MEM (Memory): segnale di controllo che indica che l'operazione da eseguire sul bus dati (lettura o scrittura) è riferita a una cella di memoria.
- I/O: segnale di controllo che indica che l'operazione da eseguire sul bus dati (lettura o scrittura) è riferita a una porta di I/O.

> I segnali del bus di controllo che analizzeremo saranno considerati in logica positiva, cioè saranno attivi quando valgono 1, indipendentemente dalla effettiva struttura elettronica di particolari CPU.



I segnali RD e WR non potranno mai essere attivi contemporaneamente, se invece entrambi non sono attivi significa che i bus non sono impegnati. Anche i segnali I/O e MEM non potranno mai verificarsi attivi in contemporanea. Questi segnali infatti servono per la discriminazione dell'operazione, cioè per la selezione del dispositivo indirizzato dalla CPU in quel particolare istante, visto che il bus indirizzi è comune per entrambi gli spazi. La tabella seguente mostra il significato dei segnali di controllo di uscita:

Read	Write	MEM	I/O	Azione
0	0	0	0	Nessuna azione
1	0	0	0	Nessuna azione
0	1	0	0	Nessuna azione
1	1	0	0	Illecita
0	0	1	0	Nessuna azione
1	0	1	0	Lettura della memoria
0	1	1	0	Scrittura della memoria
1	1	1	0	Illecita
0	0	0	1	Nessuna azione
1	0	0	1	Acquisizione di un ingresso
0	1	0	1	Emissione di una uscita
1	1	0	1	Illecita

Read	Write	MEM	I/O	Azione
0	0	1	1	Illecita
1	0	1	1	Illecita
0	1	1	1	Illecita
1	1	1	1	Illecita

Le azioni illecite non si possono mai verificare in quanto l'unità di controllo (CU) interna alla CPU è programmata in modo da non generarle mai.

Consideriamo i due segnali di ingresso principali di un ipotetico bus di controllo:

- INT: è il segnale di controllo (generato dai dispositivi di I/O) che segnala alla CPU una richiesta di attenzione. Questo segnale si rende necessario perché le periferiche di I/O sono enormemente più lente della CPU e quindi i programmi eseguiti dalla CPU non possono attendere il completamento delle operazioni, pena il bloccaggio dell'intero sistema per tempi lunghissimi. Quindi, nel momento in cui viene iniziata un'operazione di I/O, la CPU passa a eseguire altri programmi fino a quando non viene interrotta da un segnale di interrupt il quale segnala che una periferica ha i dati da fornire alla CPU oppure ha completato un'operazione che aveva già iniziato. La CPU interrompe immediatamente il programma in corso, passa all'esecuzione di un particolare sottoprogramma (chiamato "procedura interrompente") che effettua le operazioni di I/O richieste e termina restituendo il controllo della CPU al programma che era stato interrotto.
- WAIT: segnale di controllo generato da una periferica che vuole indicare alla CPU di non aver ancora completato l'invio del dato richiesto sul bus dati. La CPU campiona lo stato di questo segnale ciclicamente per verificare se si trova attivo e, in questo caso, introduce un ciclo fittizio chiamato ◀ idle ▶ nel quale attende che la periferica si dichiari pronta.

◀ Idle Il termine idle, in lingua inglese, significa inattivo o non occupato e sta proprio a indicare il ciclo di inattività della CPU nella quale non effettua altre operazioni. ▶

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Quale tra i seguenti trasferimenti non è ammesso dal bus dati?**

- ○ dalla CPU a una cella
- ○ da una cella alla CPU
- ○ da una porta in uscita alla CPU
- ○ da un registro a un altro registro
- ○ da una cella a una cella

2 **Quale stato logico non può assumere un buffer tri-state?**

- ○ uno
- ○ zero
- ○ alta impedenza
- ○ due

3 **Il bus dati trasferisce dati in forma:**

- ○ parallela
- ○ seriale
- ○ multipla
- ○ flottante

4 **Il bus indirizzi consente trasferimenti di bit in modo:**

- ○ bidirezionale
- ○ multidirezionale
- ○ monodirezionale
- ○ diretto

5 **Conoscendo lo spazio di memoria *M*, il numero di fili del bus indirizzi (*n*) è dato da:**

- ○ $n = \log_2 M$
- ○ $n = M^2$
- ○ $n = \log_{10} M$
- ○ $n = \frac{M}{n^2}$

6 **Quali, tra i seguenti, sono segnali di uscita del bus di controllo? (*due risposte*)**

- ○ RD
- ○ INT
- ○ MEM
- ○ WAIT

7 **Quali, tra i seguenti, sono segnali di ingresso del bus di controllo? (*due risposte*)**

- ○ WR
- ○ WAIT
- ○ INT
- ○ I/O

8 **Quale condizione è indicata dalla presenza dell'1 logico sui segnali Read e Write (RD = 1; WR = 1)?**

- ○ lettura dalla memoria
- ○ scrittura sulla memoria
- ○ nessuna azione
- ○ illecita

9 **In un ciclo di lettura dalla memoria, quando la CPU aggiunge un ciclo idle?**

- ○ quando la memoria invia un segnale di INT
- ○ quando la memoria invia un segnale di WAIT
- ○ quando la memoria invia un segnale di RD
- ○ quando la memoria invia un segnale di MEM

10 **Quanti elementi possono essere connessi contemporaneamente al bus dati?**

- ○ nessuno
- ○ due
- ○ tre
- ○ quattro

Esercizi di completamento

1 Il ..... è un insieme di fili conduttori che trasferiscono segnali elettrici di tipo ....., che nel loro insieme formano la ..... di un'informazione.

2 La situazione in cui il bus dati è disconnesso fisicamente dai dispositivi quali l'I/O o la memoria prende il nome di .....

3 Tutti i segnali del bus di controllo sono ..... rispetto a un segnale di ..... comune che fa avanzare le azioni della CPU con una cadenza fissata dalla sua .....

4 La CPU governa tutte le operazioni di lettura, ..., ... ; rappresenta sempre uno dei due dispositivi connessi sul .....

5 I ..... che compongono il bus indirizzi formano una sequenza di bit che rappresenta l'..... della ..... o della .....

UNITÀ DIDATTICA 5

I BUS PRESENTI SUL PC

IN QUESTA UNITÀ IMPAREREMO...

- a riconoscere i tipi di bus che collegano la CPU ad altri dispositivi
- a riconoscere i bus di sistema
- a definire i tipi di bus di espansione
- a conoscere i diagrammi di temporizzazione dei principali cicli per bus sincroni e asincroni

I bus

Il BUS può essere identificato da un insieme di linee che conducono elettricità, ciascuna delle quali collegata a un ◀ pin ▶ di un dispositivo.
Per collegare due dispositivi che usano 32 piedini per comunicare, il bus sarà formato da 32 linee costituite da altrettanti fili conduttori. In tal modo la linea 1 collegherà i piedini numero 1 dei dispositivi, la linea 2 collegherà i piedini numero 2 dei dispositivi ecc. Questo anche in ragione del fatto che ciascuna linea dei bus può trasmettere solo un segnale.

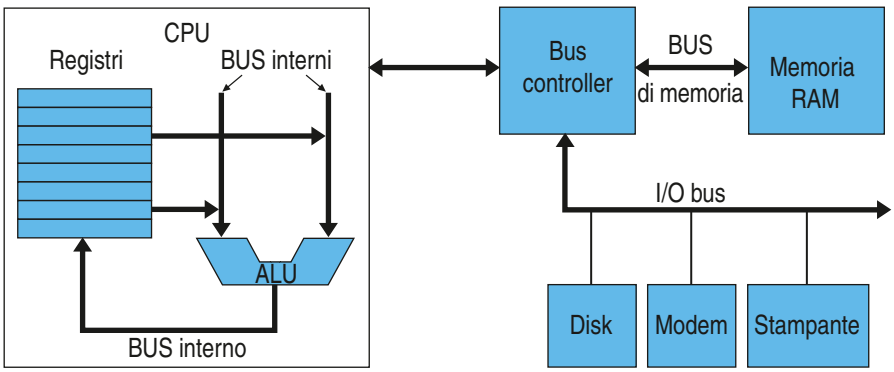
◀ Pin Comunemente chiamato piedino, si tratta di una terminazione metallica che, innestandosi nella sua controparte, stabilisce un contatto elettrico. Molto spesso a ogni piedino o pin è associato un valore binario che può commutare, consentendo ai dispositivi di dialogare tra loro. Nella figura sottostante si vedono alcuni pin di un circuito integrato. ▶

PROTOCOLLO DEL BUS

I bus si possono dividere in due categorie:
- bus interno alla CPU, trasporta dati verso e dall'ALU;
- bus esterno alla CPU, trasporta dati da e verso la memoria e i dispositivi di I/O.

Si definisce protocollo del bus l'insieme di regole precise che consentono ai vari dispositivi, rappresentati anche da schede particolari, di comunicare attraverso il bus di sistema.

Il dispositivo che intende inviare un segnale sulla linea del bus non fa altro che attivare un voltaggio particolare che identifica il dato inviato, secondo le specifiche definite dal protocollo. Il segnale viaggia sulla linea e può essere ricevuto da tutti gli altri dispositivi collegati al bus: possiamo dire che il protocollo del bus regola la trasmissione e la ricezione dei dati.



In generale possiamo affermare che il dispositivo in grado di iniziare un trasferimento dei dati è denominato master, mentre un dispositivo che inizia la comunicazione solo su comando del master assume il nome di slave. Alcuni dispositivi sono in grado di comportarsi sia come master sia come slave.

In un certo istante può esistere un solo master per cui è necessario regolare l'accesso a un bus attraverso un'operazione chiamata arbitraggio.

La tabella seguente mostra alcuni componenti dell'architettura di un computer e come si comportano nell'uso del bus.

Master	Slave	Situazione
CPU	Memoria	Fetch dell'istruzione o degli operandi
CPU	Dispositivo di I/O	Operazione di lettura o scrittura con dispositivo di I/O
CPU	Memoria	Operazione di lettura o scrittura su memoria
I/O	Memoria	DMA (*Direct Memory Access*)

Il numero di linee di un bus è detto larghezza del bus ed è direttamente proporzionale alle prestazioni: con n linee a disposizione un bus può infatti indirizzare 2^n diverse locazioni di memoria. Per aumentare le prestazioni dei bus i costruttori hanno adottato due tecniche che presentano tuttavia degli inconvenienti:

- diminuzione della durata del ciclo di bus aumentando in tal modo il numero di bit trasferiti al secondo. È di difficile implementazione perché i segnali di linee diverse si muovono a velocità diverse, presentando incompatibilità con i modelli precedenti;
- aumento della larghezza del bus, ottenendo in tal modo un numero maggiore di linee a discapito della velocità. In realtà questo finora è l'approccio più diffuso.

Possiamo sintetizzare di seguito le caratteristiche principali di un bus in termini prestazionali:

- bit rate: numero di bit al secondo (b/s) trasmessi attraverso il canale;
- larghezza: numero di linee indipendenti per la trasmissione di dati;
- velocità: frequenza del ciclo di bus per un bus sincrono;
- banda: numero (massimo) di Byte al secondo (B/s) che si possono trasmettere attraverso il canale.

Bus e sincronismo

Possiamo suddividere i bus in due categorie principali: sincroni e asincroni.

Bus sincrono

Possiede in ingresso un segnale proveniente da un oscillatore che ne sincronizza le operazioni al microprocessore. Il segnale su questa linea è un'onda quadra generata dal clock di sistema. Ha una struttura più semplice, grazie alla discretizzazione indotta dai cicli di clock, tuttavia rende difficile sfruttare al massimo le prestazioni dei dispositivi a esso connessi, rendendo le prestazioni globali vincolate al dispositivo più lento che lo utilizza.

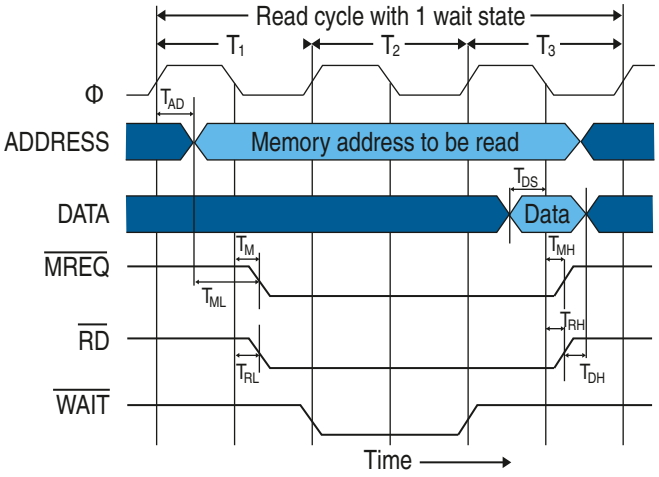
Esempio di bus sincrono con frequenza di clock a 40 MHz, ciclo del bus: 25 ns. La lettura dalla memoria richiede 40 ns dal momento in cui l'indirizzo è stabile, con 3 cicli di bus per leggere una parola.

ESEMPIO

Ciclo di lettura da memoria con bus sincrono

In questo ciclo avviene la lettura di un dato dalla memoria RAM da parte della CPU mediante un bus sincrono. L'operazione avviene in un numero definito di cicli di clock, chiamati T_1, T_2 ecc. Il grafico che segue mostra le operazioni necessarie per eseguire tale operazione.

1. Nel primo ciclo T la CPU attiva l'indirizzo della cella "leggere" (segnale ADDRESS) dopo un tempo T_{AD}.
2. Sempre nel primo ciclo T, dopo un istante chiamato T_{ML}, viene attivato il segnale MREQ (attivo basso) che attiva i chip della memoria. Contemporaneamente viene attivato dalla CPU il segnale RD che indica alla memoria che si tratta di un'operazione di lettura.
3. Sul fronte di discesa dell'istante T_2 la CPU verifica il segnale WAIT proveniente dalla memoria. Se tale segnale è basso significa che la memoria non ha ancora risposto con il dato richiesto.
4. Sul fronte di discesa di T_3 la CPU verifica il segnale WAIT: se attivo (alto) significa che il dato è disponibile sul bus dati (DATA). Prima del fronte di salita di T_3 la CPU legge il dato presente sul bus dati (DATA). Dopo alcuni istanti T_{MH} e T_{RH} la CPU disattiva i segnali di attivazione della memoria (MREQ) e di lettura (RD). ▶



Vediamo il significato dei principali segnali coinvolti:

- Φ: clock del bus;
- ADDRESS: linee di indirizzo controllate dal master (CPU);
- DATA: bus dei dati condivise da master e slave;
- MREQ: segnale di controllo generato dalla CPU che indica il tipo di indirizzo (0 = memoria, 1 = dispositivo diI/O);
- RD: tipo di operazione richiesta dalla CPU (0 = lettura, 1 = scrittura);
- WAIT: linea di controllo generata dallo slave, in questo caso dalla memoria (0 = attesa, 1= dato valido).

Il segnale WAIT generato dalla memoria informa la CPU che il dato è stabile e leggibile. In caso contrario la CPU attende un successivo ciclo T, fino a quando il segnale non è attivo (alto).

Bus asincrono

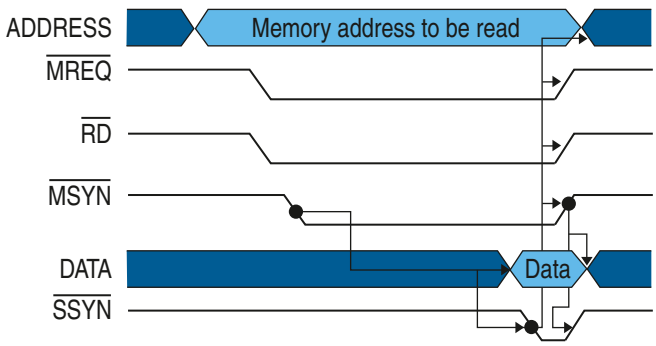
Il bus asincrono non è dotato di un clock principale. I cicli del bus possono essere della lunghezza necessaria e non devono essere uguali per tutti i dispositivi. Esso è caratterizzato da una semplicità strutturale in quanto non esiste un segnale di clock del bus, e sfrutta al meglio le prestazioni di ciascun dispositivo che lo utilizza. Possiede una circuiteria di corredo complessa che deve gestire la sincronizzazione: MSYN (*Master SYNchronization*) e SSYN (*Slave SYNchronization*).

ESEMPIO

Ciclo di lettura da memoria con bus asincrono

In questo ciclo avviene la lettura di un dato dalla memoria RAM da parte della CPU mediante un bus asincrono. La lettura, che avviene secondo la sincronizzazione fra master (CPU) e slave (memoria), è realizzata attraverso un full handshake. Il grafico che segue mostra le operazioni necessarie per eseguire tale operazione.

1. Inizialmente la CPU attiva l'indirizzo della cella "leggere" (segnale ADDRESS).
2. La CPU attiva il segnale che informa la memoria che si tratta di una lettura (RD).
3. La CPU attiva il segnale MSYN.
4. Quando il dato è reso disponibile, la memoria attiva il segnale SSYN in risposta a MSYN.
5. I segnali MSYN, RD e MREQ vengono disattivati dalla CPU in risposta a SSYN.
6. SSYN viene negato in risposta alla negazione di MSYN.



Vediamo il significato dei principali segnali coinvolti:

- MSYN: attivato dal master (in questo caso, CPU) quando è pronto per ricevere i dati, disattivato quando ha completato la lettura dei dati;
- SSYN: attivato dallo slave (in questo caso, memoria) quando i dati sono pronti sul bus, disattivato quando il master disattiva MSYN al termine dell'operazione.

L'arbitraggio del bus

Come abbiamo visto è necessario regolare l'accesso a un bus attraverso un'operazione chiamata arbitraggio che può essere fondamentalmente di tre tipi:

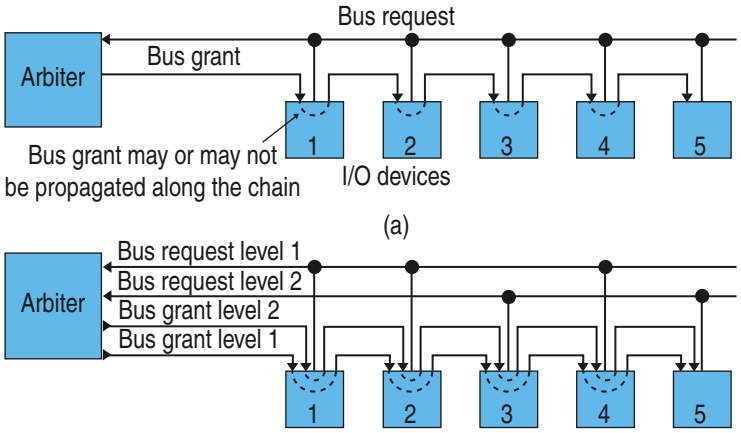
- arbitraggio centralizzato (daisy chaining a un livello);
- arbitraggio centralizzato (daisy chaining con più livelli di priorità);
- arbitraggio distribuito.

L'arbitraggio centralizzato (daisy chaining a un livello)

La concessione dell'accesso al bus avviene per un solo dispositivo alla volta ed è gestita da un dispositivo chiamato arbitro del bus. La sequenza delle operazioni dello schema (a) è così sintetizzata:

- il dispositivo master che vuole utilizzare il bus attiva la linea bus request;
- quando la linea bus request viene attivata, l'arbitro attiva la linea bus grant appena è disponibile;

- la linea bus grant viene propagata da un dispositivo a quello successivo solo se esso non ha attivato la linea bus request;
- il dispositivo che ha attivato la linea bus request deve attendere che la linea bus grant diventi attiva prima di poter utilizzare il bus.



L'arbitraggio centralizzato (daisy chaining a più livelli)

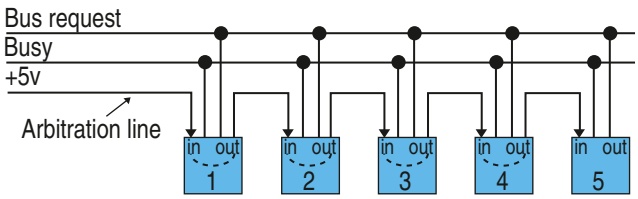
Anche in questo caso la concessione dell'accesso al bus avviene per un solo dispositivo alla volta ed è gestita da un dispositivo chiamato arbitro del bus. La sequenza delle operazioni dello schema (b) è così sintetizzata:

- ciascun livello è indipendente e funziona come descritto in precedenza tranne quando le due (o più) linee bus request sono attive contemporaneamente perché, in tal caso, l'arbitro attiva solo la linea bus grant di livello più basso (di priorità più alta).

L'arbitraggio distribuito

Non esiste l'arbitro del bus: tutti i dispositivi regolano da soli l'accesso esclusivo al bus. La sequenza delle operazioni dello schema riportato sotto è così sintetizzata:

- il dispositivo master che vuole utilizzare il bus nega la linea arbitration out;
- il dispositivo master che vuole utilizzare il bus attende che la linea arbitration in diventi attiva e che la linea busy venga negata, quindi attiva la linea busy, attiva la linea arbitration out e utilizza il bus;
- il dispositivo master che utilizza il bus nega la linea busy quando termina.

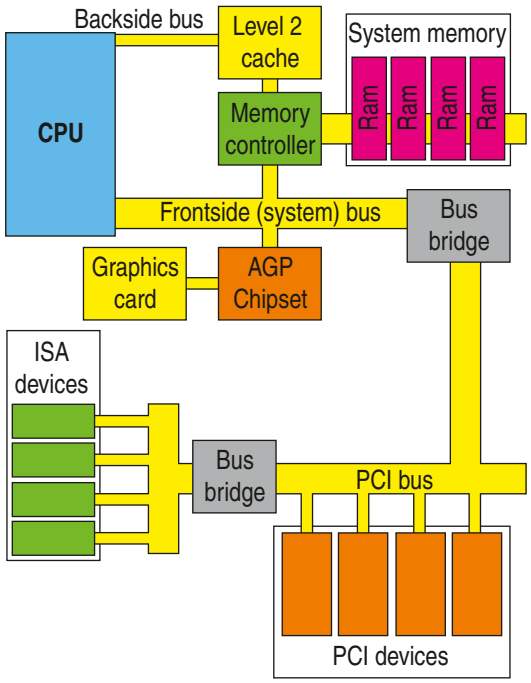


I bus principali

Venti o trent'anni fa i processori erano così lenti nell'elaborare le informazioni che i bus, durante le operazioni di lettura e scrittura, erano sincronizzati al processore stesso ed era disponibile un solo bus di collegamento, chiamato bus di sistema. Attualmente i processori sono così veloci che in molte architetture moderne esistono due o più bus, ciascuno specializzato in particolari tipi di traffico. I due bus principali sono:

- system bus o local bus;
- PCI bus o bus di espansione.

Il primo collega la CPU alla memoria di sistema ed è molto veloce, il secondo, assai più lento, mette invece in comunicazione periferiche quali per esempio il disco fisso oppure la scheda audio o la scheda video. Questi bus più lenti si connettono al system bus attraverso il chipset (north e southbridge) che svolge il compito di dirigere il traffico tra questi due bus diversi tra loro per velocità e tipologia. Esistono diversi tipi di bus di espansione, come per esempio PCI, USB, AGP, PCI Express, FireWire e altri ancora (figura a lato). ▶



Zoom su...

I BUS DI ESPANSIONE

I bus di espansione rappresentano una particolare categoria di interfacce che consentono di aggiungere nuove funzionalità al computer rendendo possibile la connessione di schede per l'uso di nuove periferiche. Questi componenti sono detti schede di espansione e si collegano direttamente negli alloggiamenti chiamati slot, posti sulla motherboard. Quando gli slot sono uguali e intercambiabili si parla di bus di espansione, cioè un insieme di segnali e circuiti elettrici che si occupano di far colloquiare le schede di espansione con la CPU, risolvendo i conflitti e le collisioni. Alcuni tra i bus di espansione più noti sono:

- VESA Local Bus
- ISA
- SCSI
- EISA
- PCI
- PCI Express
- PCI-X
- AGP
- Zorro

Front Side Bus, Back Side Bus e Bus PCI

L'FSB (*Front Side Bus*) rappresenta la connessione fisica tra il processore e la maggior parte dei componenti del computer, tra i quali la memoria RAM, i dischi e gli slot di espansione (PCI, AGP ecc.).
Il BSB (*Back Side Bus*) consente la connessione tra il processore e la memoria cache di livello 2. Questo bus opera a una velocità maggiore rispetto all'FSB, paragonabile a quella del processore. Il BSB si è evoluto notevolmente: negli anni '90 connetteva semplicemente il processore alla memoria cache che era esterna al chip della CPU. A partire dalla versione Level 2 la memoria cache era integrata nello stesso chip della CPU, rendendo a tutti gli effetti il processore integrato al BSB e alla cache Level 2. La tabella seguente mostra i bus di espansione più noti, fino agli ultimi PCI Express:

Bus Type	Bus Width	Bus Speed	MB/sec
ISA	16 bit	8 MHz	16 Mbps
EISA	32 bit	8 MHz	32 Mbps
VL-bus	32 bit	25 MHz	100 Mbps
VL-bus	32 bit	33 MHz	132 Mbps
PCI	32 bit	33 MHz	132 Mbps
PCI	64 bit	33 MHz	264 Mbps
PCI	64 bit	66 MHz	512 Mbps
PCI-X	64 bit	133 MHz	1 Gbps
PCI Express 1.0	seriale	100 MHz	2,5 Gbps
PCI Express 2.0	seriale	250 MHz	5 Gbps

Originariamente il bus PCI operava in parallelo a una frequenza di 33 MHz con una larghezza di 32 bit. Successivamente si è incrementata la velocità da 33 MHz a 66 MHz, raddoppiando i bit (da 32 a 64). In seguito con l'avvento del bus PCI-X è stata raggiunta una velocità di 133 MHz con una velocità di trasferimento di 1 Gbps. Attualmente lo standard è rappresentato dai bus di tipo PCI Express che comunicano in modo seriale e non più in parallelo, raggiungendo velocità ancora maggiori.

Il bus PCI ha rappresentato uno strumento molto innovativo, supportando la nascita e la relativa diffusione delle periferiche plug and play.

ESEMPIO

Installazione di una scheda di espansione su un bus PCI Express

In questo esempio vogliamo installare una nuova scheda video sul bus di espansione di tipo PCI Express. Per realizzarla dobbiamo innanzi tutto dotarci di una scheda video, quindi seguire le fasi descritte di seguito.

1. Molte schede madri possiedono una scheda video già montata, chiamata scheda on board. Tuttavia le prestazioni di queste schede spesso non sono adeguate ai nuovi software (soprattutto giochi). Come possiamo vedere dalla figura a lato, prima dobbiamo fare spazio alla scheda di espansione rimuovendo il pannello posteriore metallico che corrisponde allo slot nel quale intendiamo montare la scheda video. ▶

2. A questo punto, per installare fisicamente la scheda nell'appropriato slot di espansione (vedi figura a lato), dobbiamo fare pressione con entrambe le mani sui due estremi della scheda fino a quando non sarà alloggiata correttamente nello slot. Se abbiamo effettuato un'installazione corretta il supporto laterale della scheda dovrà corrispondere alla fessura liberatasi durante l'operazione precedente. ▶

Non eccedere nella pressione poiché la sottostante scheda madre è piuttosto delicata e una pressione eccessiva può danneggiare il backplane dei circuiti.

3. Il procedimento di installazione della scheda video è analogo a quello di qualsiasi altro controller di I/O di altre periferiche (audio, grafiche, video ecc.). ▶

Data la presenza di un processore grafico (GPU) all'interno della scheda video, dobbiamo sempre lasciare lo spazio necessario affinchè l'aria delle ventole di dissipazione possa defluire. Evitiamo quindi di installare sul lato delle ventole altre schede, avendo la cura di aggiungerle sempre sul lato privo di apparechiature di dissipazione del calore.

4. Come possiamo notare dall'ultima figura l'installazione di un'ulteriore scheda di espansione (audio in questo caso), avviene cercando si lasciare un po' di spazio per agevolare il passaggio di aria proveniente dalle ventole della scheda precedentemente installata.

Le periferiche plug and play

Il bus PCI è stato progettato per consentire l'utilizzo delle periferiche ◀ plug and play ▶ (PnP). Tuttavia la vera diffusione di queste periferiche si realizzò con l'avvento del sistema operativo Windows 95 che fornì il supporto a livello di sistema. Per essere utilizzate le periferiche PnP hanno bisogno di tre cose:

- BIOS PnP;
- ESCD (Extended System Configuration Data);
- Sistema Operativo PnP.

◀ Plug and play Possiamo così sintetizzare il significato di questo termine: connetti e utilizza. È stato coniato per indicare le periferiche che possono essere connesse a caldo: in qualunque istante il sistema le riconoscerà e le configurerà automaticamente. ▶

BIOS PnP

Si tratta di un software implementato a livello di memoria ROM del sistema che abilita e determina quando una periferica si è connessa.

ESCD (Extended System Configuration Data)

Si tratta di un file che contiene tutte le informazioni necessarie all'installazione delle periferiche PnP.

Sistema Operativo PnP

È il sistema operativo (da Windows 95 in poi) in grado di supportare la gestione dei ◀ driver ▶ per le periferiche PnP.
In sostanza il BIOS si accorge della periferica connessa (per esempio una pen drive) e il sistema operativo completa l'installazione dei driver necessari. Inoltre il sistema operativo svolge altri compiti tra i quali il settaggio degli elementi descritti di seguito.

◀ Driver È l'insieme delle procedure, spesso scritte in assembly, che consente al sistema operativo di pilotare un dispositivo hardware. ▶

- Interrupt Requests (IRQ)
  Un IRQ (*Interrupt Request*) è interrupt hardware che viene usato da vari elementi del computer per richiamare l'attenzione della CPU in modo del tutto asincrono. Per esempio, a ogni movimento del mouse viene inviato un IRQ informando la CPU della posizione attuale della periferica. Prima dell'avvento del bus PCI, ciascun componente hardware aveva bisogno di un IRQ specifico. Il bus PCI gestisce invece gli interrupt hardware nel chipset, consentendo in tal modo al bus di usare lo stesso segnale IRQ per più di una periferica.
- Direct Memory Access (DMA)
  Il DMA è un dispositivo che consente l'accesso diretto alla memoria da parte di una periferica senza dover sempre "scomodare" la CPU.
- Memory addresses
  Ad alcuni dispositivi periferici è assegnata una sezione di memoria per un uso esclusivo (per esempio per la scheda video).
- Configurazione degli I/O
  La configurazione dei dispositivi di I/O definisce le porte di connessione usate dai dispositivi per ricevere e inviare informazioni.

I bus di espansione

Nel sistema vi sono altri bus connessi tramite opportuni ◀ controller ▶, che sono:

- bus IDE (ATA) per connettere fino a due dischi IDE (fino a 133 MB/s);
- bus SATA (fino a 150 MB/s);
- Gigabit Ethernet (fino a 125 MB/s);
- Firewire o IEE 1394B (fino a 100 MB/s);
- USB (fino a 600 MB/s).

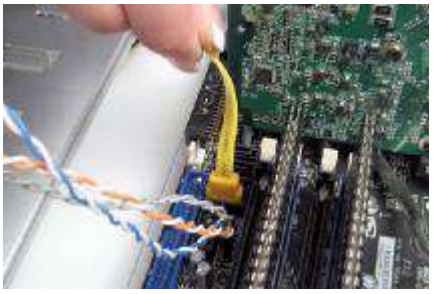
◀ Controller Il controller è un processore di I/O che gestisce le periferiche a prescindere dalla loro specifica realizzazione. La comunicazione tra il sistema operativo e la periferica è mediata da una componente software denominata driver. ▶

I bus IDE/ATA e SATA

Le sigle indicano un unico standard software per il collegamento dei dischi fissi e rimovibili. Distinguiamo le due differenti implementazioni.

- IDE/ATA: è l'acronimo di Integrated Drive Electronics ideato dalla Western Digital. La principale caratteristica di questo standard è quella di integrare nel drive anche la scheda del controller del disco. Precedentemente, gli hard disk erano dotati di un'interfaccia, di proprietà Seagate Technologies, chiamata ST506. Con IDE l'elettronica di interfaccia era minima in quanto lo standard era studiato proprio per il bus ISA avendo l'ulteriore vantaggio di essere indipendente dal tipo di disco magnetico in uso. Nel 2003,questo standard prende il nome di P-ATA (Parallel ATA), il cui connettore a 40 pin è mostrato nella figura a lato: ▶

- Nel 2004 viene introdotto un nuovo standard: il SATA (Serial ATA). Il nome indica che tale standard prevede un protocollo di tipo seriale e monodirezionale (di tipo half duplex) e mantiene le stesse specifiche del P-ATA. Il collegamento seriale risulta conveniente alle alte frequenze: lo standard SATA prevede connettori da 7 pin, il livello del segnale viene ridotto di un decimo rispetto al P-ATA, a 0,5 volt ed è consentita la connessione e disconnessione a caldo ovvero senza spegnere il sistema. La massima velocità di trasferimento è di 1,5 Gbit/s per il SATA1, 3 Gbit/s per il SATA2 e 6 Gbit/s per il SATA3. Nella figura a lato un cavo SATA3 connesso alla scheda madre di un PC e un cavo SATA3. ▶



Il bus SCSI (Small Computer System Interface)

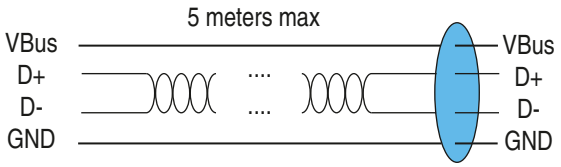
La nascita di questo standard ha permesso la costruzione e la distribuzione delle periferiche, principalmente dischi e nastri, indipendenti dalla particolare architettura di destinazione. Qualsiasi sistema infatti, una volta installata l'interfaccia SCSI, ha a disposizione un bus esterno standard su cui poter collegare le relative periferiche. Il NCITS (*National Committee on Information Technology Standards*), che svolge la sua attività sotto l'approvazione dell'ANSI, definisce gli standard a cui i produttori si devono attenere. Sono state pubblicate tre versioni di standard (SCSI-1, SCSI-2 e SCSI-3) a cui corrispondono le seguenti sigle e modalità operative:

Standard	Sigla	Mbytes/s	n bit	L. max (metri)
SCSI-1	SCSI-1	5	8	6
SCSI-2	Fast SCSI	10	8	3
SCSI-2	Fast Wide SCSI	20	16	3
SCSI-3	Ultra SCSI	20	8	1,5
SCSI-3	Wide Ultra SCSI	40	16	1,5
SCSI-3	Ultra2 SCSI	40	8	12
SCSI-3	Wide Ultra2 SCSI	80	16	12
SCSI-3	Uitra3 SCSI	160	16	12

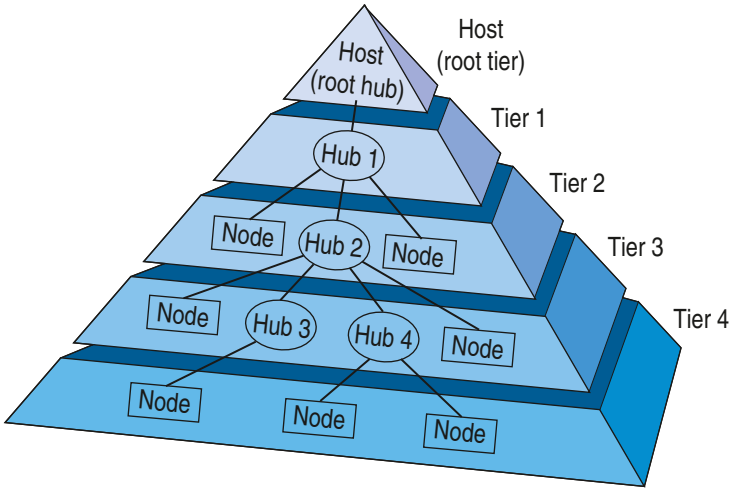
Nella figura a lato un tipico cavo a connessione SCSI. ▶

USB (Universal Serial Bus)

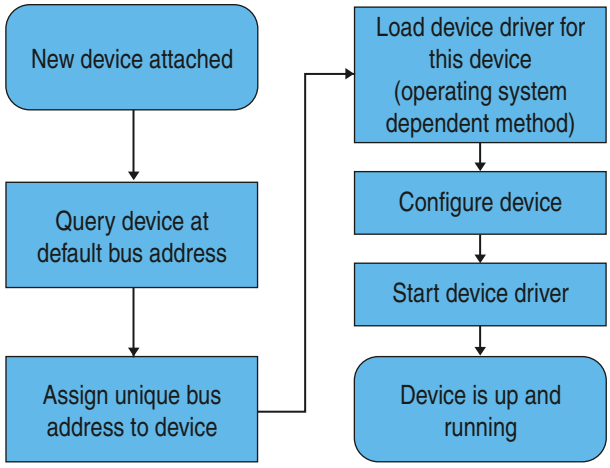
È un insieme di specifiche che consentono di connettere e configurare le periferiche plug and play al PC, arrivando a definire fino a 127 periferiche che possono operare contemporaneamente. Supporta una velocità di trasferimento fino a 12 Mbps. In pratica, l'architettura USB non pone limitazioni sul tipo di periferica connessa ma prevede solo delle specifiche e un limite superiore sulla larghezza di banda totale disponibile per una periferica. Nella figura a lato lo schema di un tipico cavo USB. ▶



Il bus dell'USB prevede una gestione a token: il controller USB del PC invia dei token (gettoni) nel bus, e la periferica alla quale è stato indirizzato risponde, accettando il token oppure inviando dati. Nella figura sottostante è riprodotto lo schema a strati (tiers) per il collegamento delle periferiche USB.



La topologia di rete dell'USB è a piramide, in cima alla quale è posizionato il controller host, e ammette connessioni con gli hub che permettono di stratificare i collegamenti a stella. Al livello degli hub viene rilevata la connessione di nuove periferiche e viene controllata l'alimentazione della periferica stessa. La connessione a caldo, senza necessità di spegnere il sistema, delle periferiche è una delle caratteristiche peculiari dell'architettura USB: una volta connessa la periferica, viene effettuata la procedura di riconoscimento chiamata bus enumeration schematizzata nel diagramma a lato. ▶



Lo standard USB garantisce la funzionalità tra i vari dispositivi collegabili, definendo delle classi di dispositivi che standardizzano anche il tipo di funzionamento. Per esempio, gli hub costituiscono una classe di dispositivi, gli HID (*Human Interface Device*), che consentono l'inserimento manuale dei dati. Il bus USB, essendo in grado di trasferire dati continui, può supportare il tipo di dati naturali associato a ogni periferica utilizzando una tecnica basata sulle priorità.
L'USB può collegare periferiche quali mouse, tastiere, scanner, macchine fotografiche digitali, stampanti, casse acustiche, microfoni e altro ancora. Per i componenti multimediali ormai lo standard USB è il metodo di collegamento più utilizzato mentre nelle stampanti, per motivi di compatibilità, sopravvivono ancora molti modelli dotati anche di porta parallela.
All'interno del computer l'USB non ha ancora rimpiazzato gli standard ATA o SCSI nonostante la loro lentezza. Essi vengono ancora utilizzati negli hard disk esterni, anche se ultimamente esistono standard come eSATA (External SATA) con velocità di trasferimento superiori.

L'USB non ha ancora rimpiazzato del tutto il connettore PS2 della tastiera: molti costruttori preferiscono mantenerlo per consentire agli utenti di poter utilizzare le economiche tastiere PS2.

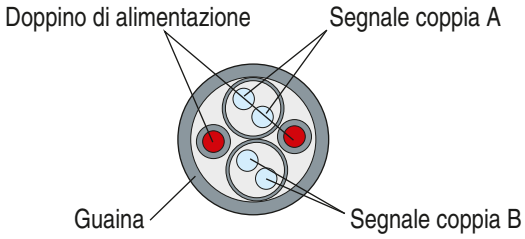
FireWire (IEEE 1394)

FireWire rappresenta un bus seriale dedicato alla comunicazione tra apparecchiature elettroniche, e presenta le seguenti caratteristiche:

- fisicamente piccolo: il cavo seriale sottile può rimpiazzare interfacce molto più costose e voluminose;
- semplice da usare in quanto non necessita di alcuna operazione di setup o terminazione;
- inseribile a caldo: consente l'aggiunta di nuovi dispositivi alla rete mentre il computer è acceso;
- di basso costo, quindi idoneo per prodotti di consumo. Grazie all'alta velocità è particolarmente adatto per collegare DVD, videocamere e scanner;
- architettura scalabile che consente di poter mescolare segnali alle diverse velocità (dal 1998 supporta anche il Gbps);
- topologia flessibile che consente di definire percorsi privati per collegamenti peer-to-peer. Consente di connettere nello stesso bus fino a 63 dispositivi;
- standard hardware e software per trasmettere dati a velocità di 100, 200 o 400 Mbps;
- interfaccia digitale ma adatta per qualsiasi segnale analogico.

In pratica possiamo considerare tale tecnologia come l'insieme delle caratteristiche offerte da USB e da SCSI.

Il cavo seriale è costituito da sei fili di rame: due sono destinati a portare l'alimentazione mentre gli altri 4, intrecciati due a due, servono per il trasporto del segnale. Ciascuna coppia intrecciata, così come l'intero cavo, è schermata, secondo il seguente schema:



I fili destinati all'alimentazione portano da 8 a 40 V_{DC} a 1,5 A. Tale alimentazione può quindi essere usata per:

- mantenere la continuità di funzionamento dell'interfaccia anche quando il dispositivo è spento;
- fornire l'alimentazione per dispositivi a basso assorbimento.

Il connettore è costruito in modo tale che i contatti restano all'interno della struttura del connettore stesso evitando quindi deformazioni per urto e contaminazioni da contatto. È possibile connettere fino a 63 dispositivi usando lo stesso segmento di bus. Ciascun dispositivo può essere distante fino a 4,5 metri da un possibile repeater. Fino a 1000 segmenti possono essere connessi insieme usando appositi bridge. L'assegnazione degli indirizzi dei nodi viene effettuata dinamicamente dal sistema quando vengono inseriti o rimossi dei dispositivi.
Sono previste due modalità operative:

- asincrona: tutte le operazioni attraverso il bus sono gestite con l'uso di segnali di interrupt controllati da un sistema host;
- isocrona: i dati vengono trasferiti a una velocità prestabilita in modalità continua senza la supervisione del sistema host. Questa modalità è particolarmente utile per la trasmissione in stream di dati tipici dei filmati video o di presentazioni multimediali.

Un'altra caratteristica del FireWire è quella di poter operare in modalità peer-to-peer, ovvero i dati possono essere trasferiti tra due dispositivi senza la supervisione del PC.

ESEMPIO

Installazione dei dischi sul bus SATA

La seguente procedura illustra come inserire i dischi negli appositi alloggiamenti. In questo esempio inseriremo un hard disk, un disco ottico e un floppy disk.

Ricordate che le sigle usate per identificare questi dischi sono le seguenti:
- HDD: Hard Disk Drive (disco fisso);
- FDD: Floppy Disk Drive (disco floppy);
- CD/DVD/R/RW: Masterizzatore CD/DVD.

1. Dopo aver inserito il disco fisso nell'alloggiamento, come indicato nella figura a lato, occorre fissarlo con le viti. Il modo in cui fissarlo dipende dal tipo di alloggiamento e di case: a volte è sufficiente avvitarlo solo da un lato, in altri casi occorre fissarlo da entrambi i lati. È importante montarlo facendolo scorrere, in modo che le porte di connessione del drive siano rivolte verso l'interno del case. ▶

2. A questo punto dobbiamo connettere i cavi: quelli del disco fisso sono normalmente due, uno per i dati (Serial ATA) e uno per l'alimentazione. Gli hard disk Serial ATA, come quello qui utilizzato, prevedono un cavo di alimentazione SATA (vedi figura a) diverso dagli ormai obsoleti di tipo Molex (vedi figura b). ▶


a)


b)

3. Il cavo SATA deve essere connesso alla scheda madre, come illustrato dalla figura a lato. ▶

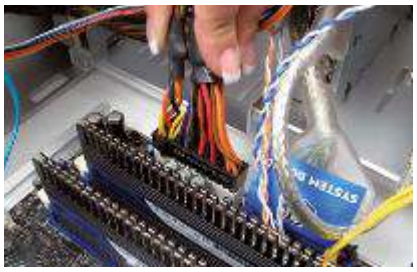
4. Occupiamoci ora dell'installazione del disco ottico. È necessario installarlo facendolo entrare dall'alloggiamento per dispositivi rimovibili da 5,25'', come si vede nella figura che segue. ▼

Se l'installazione avviene per un solo disco ottico accertati che il jumper sia settato su master oppure su single drive, opzione che dovrebbe comunque essere attiva di default. Dobbiamo attivare slave solo nel caso si intenda usare lo stesso cavo di connessione anche per un secondo masterizzatore. È importante notare che le periferiche SATA non presentano questa caratteristica.

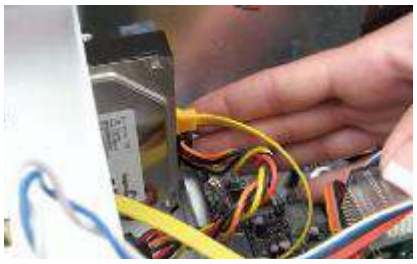
5. Adesso colleghiamo il cavo IDE alla porta posizionata sul lato interno del masterizzatore (vedi figura a lato). In realtà esistono anche masterizzatori SATA, ma non sono ancora molto diffusi. ▶

6. Volendo inserire anche un floppy disk drive, come nella figura a lato, ricordiamo di inserirlo dall'esterno, come per il CD-ROM, nell'alloggiamento da 3,5", sempre che esso sia ancora presente. ▶

7. Passiamo alla connessione delle alimentazioni. Prima di tutto inseriamo sia il cavo di alimentazione SATA per l'hard disk sia il cavo di alimentazione Molex per il masterizzatore negli appositi slot posti nella scheda madre, come si può notare nella figura a lato. ▶



8. Connettiamo ora l'altro capo dello stesso cavo di alimentazione alla porta del disco fisso (vedi figura a lato). ▶



9. Infine colleghiamo il cavo di alimentazione alla porta di alimentazione di tipo Molex del masterizzatore, come appare nella figura a lato. ▶

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Il bus interno trasporta dati:**

- ○ dalla CPU al FSB
- ○ dalla CPU alla memoria e viceversa
- ○ dalla CPU alla ALU e viceversa
- ○ dalla CPU ai dispositivi di I/O

2 **Collega i dispositivi master e slave posti a sinistra con la relativa situazione posta a destra:**

a) master: CPU slave: memoria	Operazione di lettura o scrittura su memoria	.....
b) master: CPU slave: I/O	DMA (Direct Memory Access)	.....
c) master: CPU slave: memoria	Fetch dell'istruzione o degli operandi	.....
d) master: I/O slave: memoria	Operazione di lettura o scrittura con dispositivo di I/O	.....

3 **Che cosa rappresenta il bit rate di un bus:**

- ○ frequenza del ciclo di bus
- ○ byte al secondo (B/s) che si possono trasmettere attraverso il canale
- ○ bit al secondo (b/s) trasmessi attraverso il canale
- ○ numero di bit di cui è formato

4 **Che cosa rappresenta la banda di un bus:**

- ○ frequenza del ciclo di bus
- ○ byte al secondo (B/s) che si possono trasmettere attraverso il canale
- ○ bit al secondo (b/s) trasmessi attraverso il canale
- ○ numero di bit di cui è formato

5 **In un ciclo di lettura da memoria con bus sincrono (*due risposte*)**

- ○ i segnali MREQ e RD vengono attivati nel 2° ciclo T
- ○ i segnali MREQ e RD vengono attivati nel 1° ciclo T
- ○ il segnale WAIT viene campionato nel 3° ciclo T
- ○ il segnale WAIT viene campionato nel 2° ciclo T

6 **In un ciclo di lettura da memoria con bus asincrono: (*due risposte*)**

- ○ il segnale SSYN è generato dallo slave in risposta al segnale MSYN
- ○ il segnale MSYN è generato dallo slave in risposta al segnale SSYN
- ○ il segnale RD è generato dallo slave
- ○ il segnale MREQ è generato dal master

7 **Metti in ordine sequenziale le operazioni eseguite dall'arbitro del bus con un arbitraggio centralizzato:**

a) la linea bus grant viene propagata da un dispositivo al successivo solo se il dispositivo non ha attivato la linea bus request .....
b) attiva la linea bus request .....
c) il dispositivo che ha attivato la linea bus request deve attendere che la linea bus grant diventi attiva prima di poter utilizzare il bus .....
d) attiva la linea bus grant quando il bus è disponibile .....

8 **Metti in ordine sequenziale le operazioni eseguite dall'arbitro del bus con un arbitraggio distribuito:**

a) il dispositivo master che utilizza il bus nega la linea busy quando termina .....
b) il dispositivo master che vuole utilizzare il bus nega la linea arbitration out .....
c) il dispositivo master che vuole utilizzare il bus attende che la linea arbitration in diventi attiva e che la linea busy venga negata, quindi attiva la linea busy, attiva la linea arbitration out e utilizza il bus .....

9 **Il BSB collega la CPU con:**

- ○ bus local
- ○ bus di espansione
- ○ memoria cache di livello 1
- ○ memoria cache di livello 2

10 **Il bus PCI Express comunica in modo:**

- ○ parallelo
- ○ seriale

Esercizi di completamento

1 La terminazione metallica che si innesta nella sua controparte stabilendo un contatto elettrico viene chiamata ..... oppure .....

2 Un dispositivo che inizia il trasferimento dei dati prende il nome di ....., mentre un dispositivo che inizia la comunicazione solo su comando del ..... prende il nome di .....

3 In un certo istante vi è solo un dispositivo connesso al bus che inizia il trasferimento per cui è necessario regolare l'accesso a un bus attraverso un'operazione chiamata .....

4 Il numero di linee di un bus è detto ..... ed è direttamente proporzionale alle prestazioni, infatti con ..... linee a disposizione un bus può indirizzare ..... diverse locazioni di memoria.

5 La lettura mediante bus asincrono avviene secondo la sincronizzazione fra ..... e ..... ed è realizzata attraverso un .....

6 Il daisy chaining a più livelli funziona come il daisy chaining a un livello tranne quando le linee ..... sono attive contemporaneamente perchè in tal caso l'arbitro attiva solo la linea ..... di livello più basso.

7 I due bus principali di un sistema a microprocessore sono: ..... e .....

8 Il controller USB del PC invia dei ..... nel bus e la periferica alla quale è stato indirizzato risponde, accettando il ..... oppure inviando dati secondo una struttura a .....

UNITÀ DIDATTICA 6

LA GESTIONE DEGLI I/O DAL PUNTO DI VISTA FUNZIONALE

IN QUESTA UNITÀ IMPAREREMO...

- a riconoscere il ruolo dei dispositivi di I/O nel funzionamento di un computer
- a definire il ruolo delle periferiche
- a definire il ruolo degli adattatori

I dispositivi di I/O

I DISPOSITIVI DI I/O

I dispositivi di I/O sono componenti che trasformano l'informazione rappresentata da una grandezza fisica qualsiasi in una grandezza fisica elettrica oppure trasformano l'informazione rappresentata da una grandezza fisica elettrica in una grandezza fisica di altro genere.

Il computer è un dispositivo elettronico che riconosce ed elabora solo ◀ segnali ▶ di natura elettrica (tensioni e correnti).

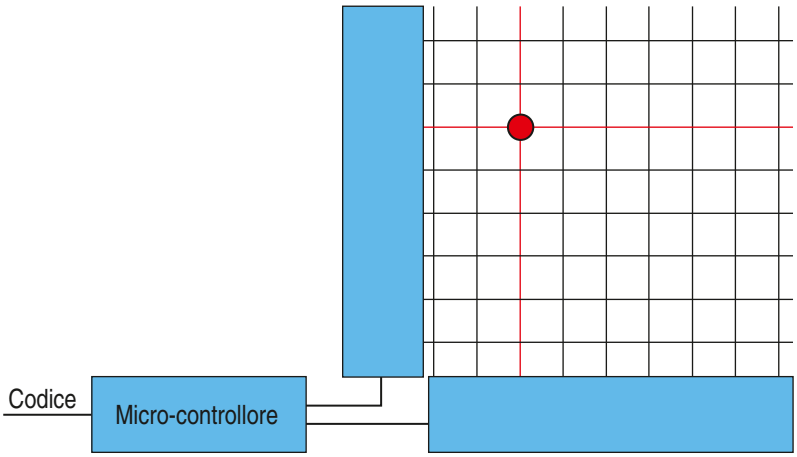
◀ Segnali Si tratta di una grandezza fisica che può assumere valori diversi nel corso del tempo: a ciascun valore viene associato un diverso significato, per esempio alla tensione di 0 volt è associato il valore logico 0 mentre alla tensione di +5 volt è associato il valore logico 1. Possiamo affermare che per ciascun valore di un segnale è associata una distinta informazione. ▶

Il computer riceve informazioni in ingresso che devono essere elaborate per fornire informazioni in uscita. Le informazioni ricevute sono di natura elettrica ma sono basate su altre ◀ grandezze fisiche ▶ che sono percepibili dagli utenti.

◀ Grandezza fisica È un fenomeno fisico misurabile al quale viene associata un'unità di misura. Sono grandezze fisiche la tensione (volt), il rumore (dB), la frequenza (Hz) ecc. Le grandezze fisiche possono essere di varia natura, suddivise in categorie secondo le tradizionali ripartizioni della fisica: ottica, magnetismo, meccanica ecc. ▶

ESEMPIO

L'utente, per esempio, utilizza la tastiera quando deve immettere dei dati nel computer, ed essa è formata da interruttori che chiudendosi alla pressione del dito forniscono un segnale di natura elettrica. In questo caso la natura iniziale del segnale associato all'informazione è di tipo meccanico. Alla pressione di un tasto della tastiera, viene trasmesso un segnale specifico al computer. La tastiera utilizza in effetti una rete di matrici che permette di identificare ogni tasto grazie a una linea e a una colonna.



Quando l'utente preme un tasto, viene stabilito un contatto elettrico tra la linea e la colonna. I segnali elettrici sono trasmessi a un microcontrollore, che invia un codice (BCD, ASCII o Unicode) al computer descrivendo il carattere corrispondente al tasto.

In generale possiamo suddividere i dispositivi di I/O in due differenti sezioni funzionali:

- periferica;
- adattatore o controller.

La periferica

È un componente in grado di comunicare con elementi di natura elettrica e di altra natura (ottica, magnetica, meccanica ecc.). Il termine periferica sta proprio a indicare che si trova al di fuori della sezione elaborativa del computer, come abbiamo visto nel modello di Von Neumann.

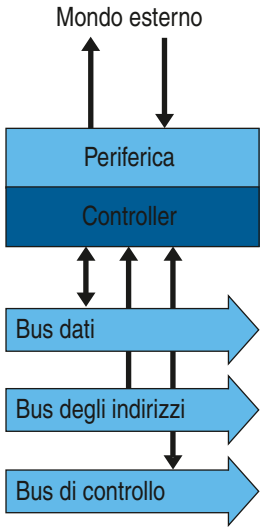
Il controller

È un componente elettronico che comunica sia con la CPU che con la parte elettronica delle periferiche adattando i segnali elettrici della CPU alle esigenze della periferica. In genere è contenuto all'interno del computer.

ESEMPIO

Il video è in realtà un dispositivo di I/O formato dalla periferica chiamata monitor o display e dal controller chiamato video adapter. Il display è il dispositivo elettro-ottico che trasforma i segnali elettrici in immagini luminose, mentre il video adapter è il dispositivo elettronico integrato alla scheda madre del computer o in una scheda aggiuntiva che trasforma i segnali della CPU in segnali per il monitor.

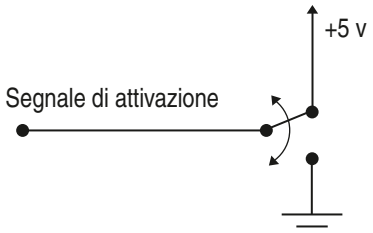
Dopo quanto abbiamo affermato, osserviamo lo schema logico riportato a lato che descrive la collocazione dei dispositivi di I/O secondo il modello di Von Neumann. ▶



Molto spesso l'adattatore è un vero e proprio computer embedded dotato di una sua CPU specializzata come accade per le GPU (*Graphics Processing Unit*) delle schede video. Come si vede dalla figura seguente sono necessarie delle ventole di raffreddamento aggiuntive per dissipare il calore generato dalla GPU. ▼

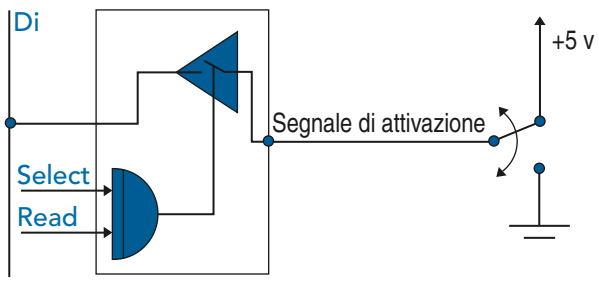
Molte schede video infatti sono dotate di un software interno e di una memoria RAM interna per l'elaborazione dei dati temporanei. Dal punto di vista della programmazione, possiamo tuttavia immaginare l'adattatore grafico come a un insieme di porte di I/O.

L'elemento di ingresso dell'I/O



Per comprendere il funzionamento dei dispositivi di I/O è necessario iniziare con il dispositivo elementare. L'elemento di informazione più piccolo che viene acquisito in ingresso è come sempre il bit. Per fornire due diversi livelli di tensione, immaginiamo un circuito in grado di commutare 0 volt con il circuito aperto e +5 volt con il circuito chiuso. ▶

Il segnale di attivazione non può essere connesso direttamente al bus dati in quanto andrebbe a imporre il proprio livello sul filo conduttore impedendo agli altri dispositivi di usare il bus. Dobbiamo pertanto aggiungere una circuiteria in grado di far memorizzare e indicare la direzione, se in uscita o in ingresso: un buffer tri-state. Nello schema che segue viene utilizzato un circuito in grado di simulare un solo bit di ingresso; nella realtà dobbiamo immaginare la presenza di altri sette elementi, per formare almeno un byte di ingresso, collegati allo stesso circuito di selezione:



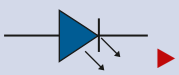
Come possiamo notare l'elemento di ingresso è rappresentato dal buffer tri-state che mantiene la sua uscita ad alta impedenza fino a quando non riceve il segnale di Select, in modo da lasciare libero il bus dati quando non è occupato da operazioni che coinvolgono il dispositivo di I/O per essere usato dagli altri elementi (memoria e CPU).

Il circuito di selezione è formato da una porta logica AND che riceve il segnale Read dal bus di controllo e il segnale di selezione Select proveniente dall'address decoder. Quando la CPU decide di acquisire un valore proveniente dal dispositivo di I/O (operazione di IN) seleziona la porta attivando anche il segnale Read che la informa che si tratta di una lettura da quel dispositivo. A quel punto il buffer tri-state esce dallo stato di alta impedenza e pone in uscita il valore logico sul piedino del bus dati (D_i). La CPU può a questo punto leggere il bit dal bus dati selezionando successivamente i segnali di controllo che riportano il buffer tri-state in alta impedenza liberando il tal modo il bus dati.

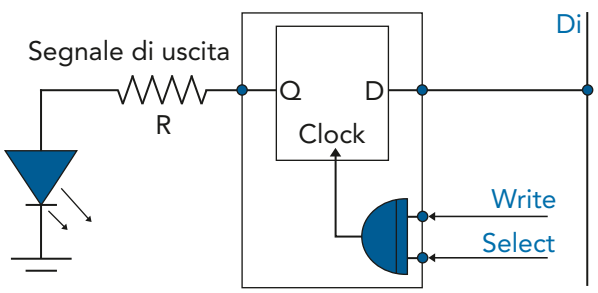
Il segnale di Select proviene dal circuito di decodifica degli indirizzi (address decoder) che riceve in ingresso l'indirizzo dal bus indirizzi e il segnale di controllo di I/O, quindi decodifica questi segnali generando l'indirizzo che costituisce il segnale di selezione per la porta con l'indirizzo corrispondente.

L'elemento di uscita dell'I/O

Consideriamo sempre il bit come elemento minimo di informazione che può essere emesso in uscita: anche in questo caso prendiamo in esame un dispositivo di uscita che emuli il funzionamento di un qualsiasi dispositivo simile; utilizziamo un circuito elementare formato da un ◀ LED ▶ che emette un segnale luminoso quando è presente un valore logico pari a 1 e si spegne quando il valore logico è 0. Il filo conduttore che riceve il segnale logico non può essere collegato direttamente con il bus dati in quanto la continua commutazione dei segnali presenti sul bus farebbe accendere e spegnere il led secondo una sequenza senza senso.

◀ LED Si tratta di un particolare diodo a emissione luminosa. LED infatti significa light emitting diode ed è un dispositivo optoelettronico che sfrutta le proprietà ottiche di alcuni materiali semiconduttori per produrre fotoni attraverso il fenomeno dell'emissione spontanea ovvero a partire dalla ricombinazione di coppie elettrone-lacuna. Il simbolo elettronico è riportato a lato:  ▶

Il circuito seguente mostra come mantenere l'accensione del LED inalterata fino a quando il sistema (tipicamente la CPU) non invia un dato diverso al dispositivo di I/O. Per fare questo usiamo un flip-flop di tipo D in grado di conservare l'informazione scritta fino alla scrittura successiva sulla stessa porta. Possiamo pertanto paragonare il funzionamento di una porta di uscita con quello di una cella di memoria con la sola differenza che in questo caso avviene la sola scrittura, mentre la lettura avviene con un circuito differente. Nello schema a lato viene utilizzato un circuito in grado di simulare un solo bit di uscita; nella realtà dobbiamo immaginare la presenza di altri sette elementi per formare almeno un byte di uscita collegato allo stesso circuito di selezione. ▶

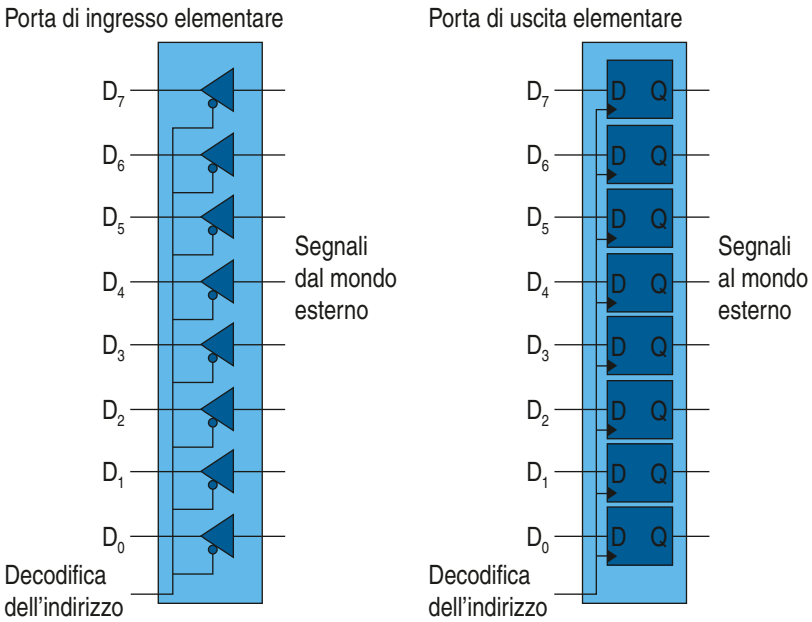


Come possiamo notare l'elemento di uscita è un flip-flop di tipo D che mantiene inalterato il valore Q di uscita indipendentemente dal valore del suo ingresso D. Il circuito di selezione è formato da una porta logica AND che riceve il segnale Write dal bus di controllo e il segnale di selezione Select dall'address decoder. Quando la CPU decide di inviare un dato in uscita colloca il dato da emettere sul bus dati, quindi imposta la porta prescelta e attiva il segnale di controllo Write. La commutazione che avviene sul segnale CLK provoca il caricamento del segnale contenuto nel filo del bus dati nella memoria del flip-flop provocando di conseguenza l'emissione del dato in uscita che resterà inalterato fino alla successiva selezione e scrittura.

Il segnale di Select proviene dal circuito di decodifica degli indirizzi (address decoder) il quale riceve in ingresso l'indirizzo dal bus indirizzi e il segnale di controllo di I/O, quindi decodifica questi segnali generando l'indirizzo che costituisce il segnale di selezione per la porta con l'indirizzo corrispondente.

Le porte di I/O

Ogni dispositivo di I/O non è accessibile singolarmente ma solo come parte di un blocco di elementi che vengono gestiti insieme. Il più piccolo blocco di memoria accessibile singolarmente, come accade per la memoria, è rappresentato da un byte. Ciascun bit della porta è collegato a un diverso filo conduttore del bus dati in maniera tale che ciascun bit sia trasferito parallelamente agli altri.



A ogni lettura da I/O (IN) tutti i valori presenti nella porta vengono scritti sul bus dati e conseguentemente a ogni scrittura su I/O (OUT) tutti i valori presenti sul bus dati vengono caricati sulla rispettiva porta.

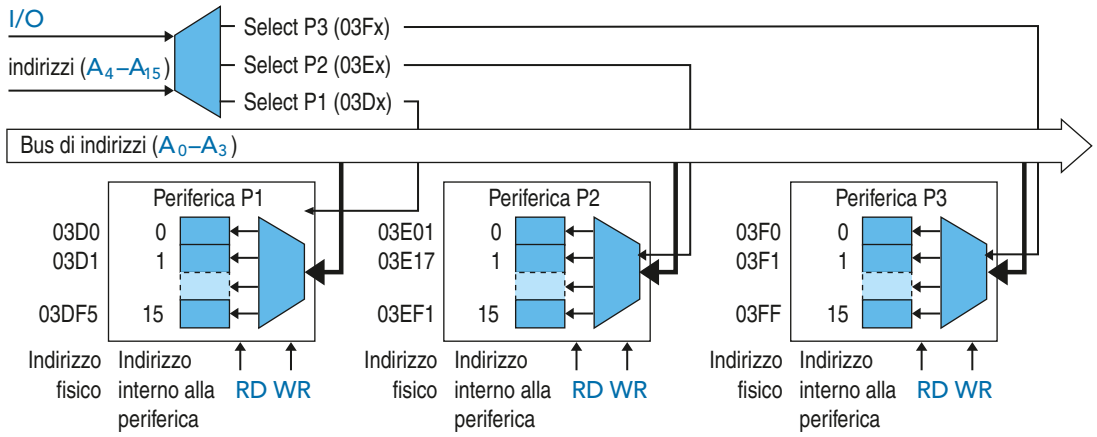
Le porte di I/O si trovano integrate nei componenti elettronici utilizzati per la realizzazione dei controller di periferiche. Sono piccoli processori dotati di memoria ROM che conservano un software che ne definisce le funzioni. Le porte vengono indirizzate seguendo lo stesso modello degli indirizzi di memoria, quindi possiamo immaginare una lista di porte di I/O poste logicamente una dopo l'altra, dove la posizione di ciascuna porta è definita univocamente dal suo indirizzo.

Il numero massimo di porte che possono essere indirizzate in un sistema viene chiamato spazio di I/O.

Il bus indirizzi viene usato anche per indirizzare le porte di I/O, anche se la dimensione dei bit utilizzati per l'indirizzamento delle porte di I/O è sensibilmente minore.

Il circuito di decodifica degli indirizzi di I/O

Nella figura che segue è indicato lo schema di un ipotetico circuito che rappresenta una struttura di porte di I/O formata da tre periferiche chiamate P1, P2 e P3:



Ciascuna ipotetica periferica è formata da 16 porte con indirizzo interno che va da 0 a 15. I quattro bit meno significativi del bus indirizzi (A_0 - A_3) servono per indirizzare le 16 porte presenti all'interno di ciascuna periferica ($2^4 = 16$) e sono collegate al circuito di decodifica dell'indirizzo di I/O posto all'interno di ciascuna periferica. I 12 bit più significativi (A_4 - A_{15}) presenti sul bus indirizzi consentono di individuare ben $2^{12} = 4096$ periferiche diverse; in questo esempio ne vengono indicate tre ma come si può ben capire possiamo aggiungerne altre.

Lo schema proposto simula la situazione reale rappresentata dalla decodifica degli adattatori Com1, Com2, Lpt1, che rappresentano le porte seriali (Com1 e Com2) e parallele (Lpt1) del PC. La logica interna delle periferiche deve convertire i dati emessi dalla CPU e ricevuti dall'esterno, per esempio l'adattatore seriale deve convertire su un unico canale i dati provenienti in parallelo dalla CPU e viceversa.

> Il bus dati, che non appare nello schema per non appesantirlo troppo, raggiunge tutte le periferiche.

Le porte di I/O di un PC

Le porte di I/O in un PC sono rappresentate da una serie di prese, posizionate in genere sul pannello posteriore del computer, le quali vengono utilizzate per collegare alla macchina tutti i dispositivi esterni (tastiera, video, mouse ecc.). Alcune porte sono poste direttamente sulla scheda madre, ciascuna con un colore identificativo, secondo lo ◀ standard PC99 ▶:

◀ Standard PC99 Si tratta di uno standard appartenente al PC System Design Guide che identifica una serie di requisiti e raccomandazioni di progettazione hardware per i personal computer compilata da Microsoft e Intel per aiutare i produttori di hardware a usare meglio il sistema operativo Microsoft Windows e per semplificare l'installazione hardware sui computer. In tali requisiti vi sono per esempio i colori che devono avere le porte presenti sul pannello del computer. ▶

Vediamole in sintesi.

Porte PS/2
Utilizzate per il collegamento del mouse e della tastiera (una è dedicata al mouse e l'altra alla tastiera e non si possono invertire).

Interfaccia seriale
Porta di comunicazione per periferiche esterne tramite bus seriale, può trasportare un solo bit per volta. La sua velocità massima è infatti di soli 115.200 bps. Alla porta seriale vengono solitamente collegate periferiche lente, come mouse e modem esterni di qualche anno fa. Sostanzialmente è superata dalle porte USB e FireWire.

Interfaccia parallela
Normalmente i PC possiedono una porta stampante parallela a cui possono essere connesse la stampante e altri dispositivi che comunicano in modalità parallela (Iomega ZIP, scanner ecc.). Essa è conosciuta anche come LPT, porta stampante, printer port oppure ancora interfaccia Centronics. Ormai è montata solo su richiesta in quanto le periferiche più diffuse consentono di trasmettere dati in modalità seriale con USB.

Nel tempo la porta parallela montata nel PC si è evoluta anche se, con qualche eccezione, si è mantenuta la compatibilità con i primi modelli. Molti costruttori hanno in passato apportato modifiche e miglioramenti allo standard originario purtroppo con scarsa compatibilità tra modelli diversi: ciò ha reso di fatto inutilizzabili sulla maggior parte dei PC le caratteristiche avanzate disponibili solo per alcune porte parallele.

Le categorie che comprendono tutte le porte attuali sono sostanzialmente le tre descritte di seguito.

- Standard Parallel Port (SPP): è il modello originario di porta parallela, installato sul primo PC XT. Pensata originariamente solo per la connessione delle stampanti, con qualche accorgimento consente anche altri usi. Tutti gli altri tipi di porta parallela sono compatibili con questo e quindi il software e l'hardware sviluppati per questa porta sono praticamente universali.
- Enhanced Parallel Port (EPP): permette lo scambio bidirezionale dei dati fornendo un supporto hardware per l'handshaking. È possibile raggiungere alte velocità di trasferimento, in alcune applicazioni anche 10 volte maggiori della SPP. Questa modalità è descritta in un altro tutorial presente sul sito http//www.vincenzov.net.
- Extended Capabilities Port (ECP): permette la maggiore flessibilità di utilizzo, ha il supporto del DMA e una grande varietà di modalità di funzionamento. Come contropartita è piuttosto difficile progettare hardware in grado di usarla.

Porta di rete
Per collegare la macchina direttamente a una rete di computer, senza usare il modem. Ne esistono di vari tipi, ma ormai la presa RJ45, come vedremo nei Moduli successivi, ha di fatto rimpiazzato tutte le altre.

> Il termine porta indica una connsessione tra i cavi di un calcolatore o un qualunque altro dispositivo elettronico.
> Nel campo della trasmissione dei dati, durante una connessione, non esiste alcuna fisica oltre quella di aggancio del connettore del cavo. Nel campo delle reti di telecomunicazione il termine porta indica solamente una porta virtuale identificata da un'etichetta utile a discriminare il traffico dati di una connessione da quello di un'altra.

Zoom su...

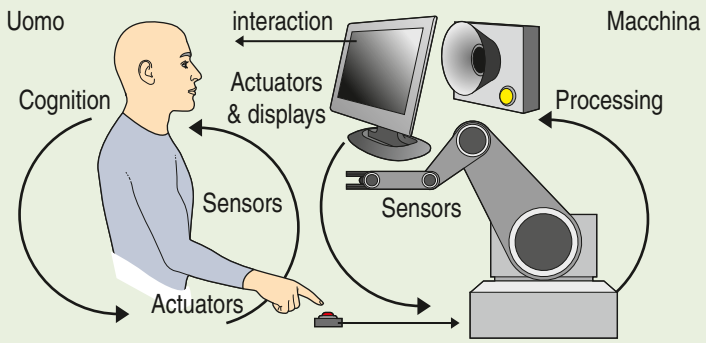
L'INTERFACCIA UMANO-MACCHINA (HMI)

L'interfaccia umano-macchina (in inglese Human-Machine Interface, HMI) si riferisce allo spazio che separa un essere umano che sta utilizzando una macchina dalla macchina stessa.

> La parola è un'evoluzione del termine originale, interfaccia uomo-macchina (in inglese Man-Machine Interface, MMI).

Un esempio di interfaccia umano-macchina sono sia l'hardware che il software di un computer che rendono possibile per esempio a un utente monitorare e controllare in remoto un grande macchinario.
L'interfaccia utente comprende il flusso di informazioni per il supporto delle decisioni attraverso:

- messaggi visivi, generalmente forniti da uno schermo o monitor;
- messaggi sonori, altoparlanti, sirene, ricetrasmittenti;
- azioni di controllo, tastiere, pulsanti, interruttori.



Un insieme di più monitor, dispositivi e superfici di controllo forma una console o stazione di controllo. Da una console, un operatore visiona le informazioni, riceve notifiche ed esegue azioni di controllo. Una sala di controllo può contenere una o più stazioni di controllo. La sale di controllo e le stazioni di controllo si possono incontrare in molte applicazioni diverse, come per esempio:

- torre di controllo di un aeroporto;
- centro di controllo di polizia e ambulanze;
- centrale nucleare.

La progettazione dell'abitacolo di aeroplani ed elicotteri comprende la progettazione di interfacce utente altamente specializzate. Ma anche applicazioni largamente diffuse come lavatrici o lavastoviglie necessitano di un'interfaccia utente.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 L'operazione di emissione di dati verso i dispositivi di I/O da parte della CPU prende il nome di:

- ○ store
- ○ OUT
- ○ load
- ○ IN

2 Il numero massimo di porte che possono essere indirizzate in un sistema viene chiamato:

- ○ espansione
- ○ spazio di memoria
- ○ spazio di indirizzamento
- ○ spazio di I/O

3 Le porte di I/O sono: (*due risposte*)

- ○ celle di memoria indirizzabili
- ○ processori dotati di memoria ROM
- ○ dispositivi asincroni
- ○ indirizzabili come la memoria

4 Lo standard PC99 definisce:

- ○ la dimensione dei bus
- ○ i colori delle porte sul pannello della scheda madre
- ○ lo spazio di indirizzamento dei dispositivi di I/O
- ○ la bit rate dei bus locali

5 La porta PS/2 viene usata per il collegamento:

- ○ delle videocamere
- ○ della stampante
- ○ dei dischi SATA
- ○ del mouse

6 La porta USB comunica in modo:

- ○ parallelo
- ○ seriale
- ○ asincrono
- ○ sincrono

7 Quale tra le seguenti categorie non appartengono agli standard delle porte parallele?

- ○ PPP
- ○ SPP
- ○ EPP
- ○ ECP

Esercizi di completamento

1 I dispositivi di I/O trasformano l'informazione rappresentata da una ..... in una ..... o viceversa.
2 Il segnale è una grandezza fisica che può assumere ..... diversi nel corso del tempo, a ciascun ..... viene associato un diverso significato.
3 Il video è un dispositivo di I/O formato dalla periferica chiamata ..... e dal controller chiamato .....
4 A ogni lettura da I/O ..... tutti i valori presenti nella porta vengono scritti sul bus dati e conseguentemente a ogni scrittura su I/O ..... tutti i valori presenti sul bus dati vengono caricati sulla rispettiva porta.

UNITÀ DIDATTICA 7

LE ARCHITETTURE NON VON NEUMANN

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere le principali tecniche che migliorano le prestazioni dei computer
- a capire come si sono evolute le tecniche di elaborazione
- a comprendere le metodologie di gestione evoluta della memoria
- ad approfondire lo sviluppo nella gestione dei dispositivi di I/O

L'evoluzione dei sistemi di elaborazione

Per migliorare le caratteristiche delle CPU e dei sistemi di elaborazione in generale esistono diverse tecniche. In primo luogo vi è l'aumento progressivo della frequenza di clock che a parità di comportamento rende più veloci le elaborazioni. Altre direzioni di evoluzione sono state l'aumento dell'ampiezza di parola e l'aumento dello spazio di indirizzamento, incrementando rispettivamente il numero di bit elaborati contemporaneamente e il numero di celle indirizzabili.
Entrambe le tecniche tuttavia, nell'evoluzione delle architetture, hanno raggiunto presto un punto nel quale non era più possibile ottenere prestazioni migliori. Nacque allora un'altra direzione di sviluppo, chiamata in generale con il termine di architetture non Von Neumann, nella quale sono state introdotte funzioni di elaborazione parallele che consentono di aumentare la velocità di elaborazione a parità di frequenza di clock.

Zoom su...

GRADO DI PARALLELISMO DELLE ARCHITETTURE

Le architetture possono essere classificate in base al grado di parallelismo che rendono possibile:
- SISD (*Single Instruction Single Data*), un unico processore esegue un'unica istruzione per volta e può prelevare e depositare in memoria un solo dato per volta;
- SIMD (*Single Instruction Multiple Data*), permette di eseguire contemporaneamente la stessa operazione su più dati (utile per le operazioni multimediali);

- MISD (*Multiple Instruction Single Data*), consente di iniziare l'esecuzione di diverse istruzioni, ognuna sui propri dati. Mentre si esegue un'istruzione si può cominciare a elaborare la successiva;
- MIMD (*Multiple Instruction Multiple Data*), è relativa ai sistemi multiprocessori.

Le evoluzioni che riguardano l'elaborazione

Le evoluzioni riguardanti l'elaborazione delle operazioni da parte della CPU sono descritte di seguito.

Esecuzione fuori ordine

Indica la capacità delle CPU di eseguire istruzioni senza rispettarne necessariamente l'ordine imposto dal codice che le contiene. Per fare questo la CPU analizza le istruzioni che dovrà eseguire, individuando le istruzioni non vincolate dalle altre. Quelle che non hanno vincolo sequenziale vengono fatte eseguire in parallelo.
Questo metodo non sempre può essere impiegato. Consideriamo un'istruzione che esegue una somma tra due registri, il risultato deve essere usato come operando per una divisione. In questo caso le due istruzioni non potranno essere eseguite in parallelo in quanto è necessario rispettare una sequenza operativa: non potremo in questo caso ricorrere all'esecuzione fuori ordine.
Il metodo prevede prima di tutto che il programma venga caricato con istruzioni seriali, quindi le istruzioni vengono analizzate e riordinate tenendo conto delle dipendenze, successivamente eseguite in parallelo e riordinate prima di provvedere al salvataggio dei dati in memoria.

Il prefetch (precaricamento)

In base a questa tecnica i processori implementano delle unità che analizzano il codice cercando di prevedere in anticipo quali dati o istruzioni serviranno al processore, provvedendo inoltre al loro caricamento in cache (o direttamente nel processore) prima della loro reale impiego. Alcune architetture prevedono apposite istruzioni per indicare quali blocchi precaricare ma la maggior parte delle architetture non sono dotate di queste istruzioni e quindi devono basarsi solamente sull'ispezione del codice in tempo reale.
Il precaricamento dei dati e delle istruzioni determina alcuni problemi: per esempio se il microprocessore carica delle istruzioni dipendenti da un salto e quel salto non viene eseguito il processore deve provvedere a eliminare le istruzioni caricate erroneamente prima di caricare le istruzioni da eseguire.
Problemi anche più seri si hanno nel caso di generazione di un'eccezione, per esempio l'accesso a una locazione non consentita genera un'eccezione che va segnalata al sistema operativo, ma la segnalazione va effettuata nel momento in cui dovrebbe effettivamente avere luogo e non prima, per via del precaricamento.
Il precaricamento dei dati deve invece garantire la coerenza e la validità dei dati stessi, quindi se un dato viene precaricato nessuna istruzione deve modificarlo prima del suo effettivo utilizzo da parte delle istruzioni che hanno forzato il precaricamento. Queste criticità rendono il precaricamento dei dati e delle istruzioni molto complesso da implementare in hardware senza un supporto diretto del set di istruzioni.
Se invece il set di istruzioni supporta nativamente questa caratteristica la sua gestione diventa molto più semplice.

Speculative execution

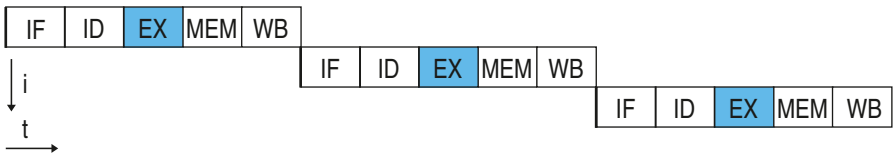
È una tecnica che consiste nell'eseguire entrambi i rami di un salto in modo da poter prevedere la direzione in una diramazione.

La pipeline

Il termine pipeline significa catena di montaggio e sta proprio a indicare che nella CPU esistono dei blocchi che elaborano parte delle fasi passandole al blocco successivo.

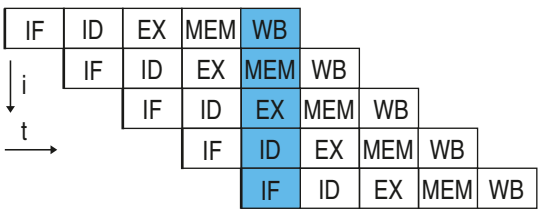
L'elaborazione di un'istruzione da parte di un processore si compone di cinque passaggi fondamentali:
- IF: lettura dell'istruzione da memoria (Instruction Fetch);
- ID: decodifica dell'istruzione e lettura degli operandi da registri (Instruction Decode);
- EX: esecuzione dell'istruzione (Execute);
- MEM: attivazione della memoria (Memory);
- WB: scrittura del risultato nel registro opportuno (Write Back).

Senza l'uso della pipeline abbiamo la seguente sequenza:



Le attività della CPU, come già illustrato nelle Unità didattiche precedenti, sono scandite da un clock: ciascuna operazione elementare richiede almeno un ciclo di clock per poter essere eseguita. Con il progresso della tecnologia si è potuto integrare un numero maggiore di transistor all'interno dello stesso microprocessore per aumentare il parallelismo delle operazioni riducendo in tal modo i tempi di esecuzione.

L'attività di una CPU con pipeline è formata da cinque stadi specializzati, capaci di eseguire ciascuno un ciclo macchina. La CPU lavora come in una catena di montaggio (pipeline) eseguendo a ogni stadio un solo compito specifico. Quando la catena è a regime, a ogni ciclo di clock viene completata l'esecuzione dei cinque stadi per un'istruzione. Nello stesso istante ogni unità elabora in parallelo i diversi stadi delle istruzioni successive:



Riassumendo, la pipeline consente di migliorare le prestazioni in termini di velocità di esecuzione, ma comporta una maggiore complessità circuitale della CPU che deve essere composta da almeno cinque unità specializzate in grado di collaborare tra loro.

Le principali problematiche che si possono manifestare riguardo all'impiego della pipeline sono le seguenti:
1. conflitti strutturali: quando due fasi utilizzano contemporaneamente la stessa risorsa, come per esempio i bus, l'ALU o la memoria RAM;
2. conflitti tra i dati: quando un'istruzione utilizza un dato che è ancora in fase di elaborazione;
3. conflitti di controllo: quando le istruzioni che devono essere eseguite da un salto si trovano in una zona di memoria non prevista. In questo caso la coda delle istruzioni deve essere caricata nuovamente con conseguente perdita di tempo.

La pipeline presenta una dipendenza dalle strutture (caso 1), dai dati (caso 2) e dai controlli del flusso del programma (caso 3).
Un metodo che viene applicato per la soluzione di queste criticità è dato dallo stallo: consiste nell'inserimento di alcuni cicli per bloccare temporaneamente una parte della pipeline in attesa che si risolva il conflitto. Se per esempio nella fase di Execute la seconda istruzione richiede il contenuto di un registro che dovrà essere aggiornato dalla fase successiva (Write Back), viene aggiunto un ciclo di stallo dopo la fase di Execute della seconda istruzione, come illustrato dalla tabella seguente:

I istruzione	Fetch	Decode	Execute	Write Back		
II istruzione	---	Fetch	Decode	Execute	Stallo	Write Back

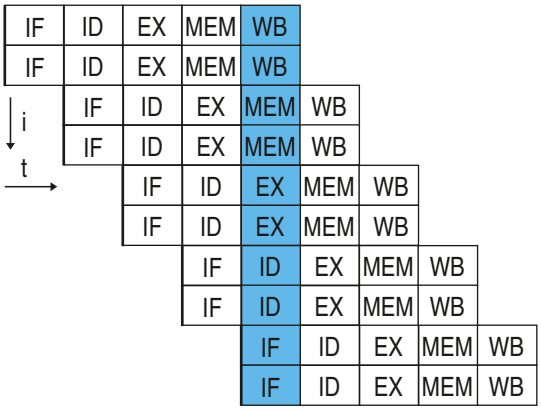
In generale, possiamo tuttavia affermare che l'inserimento degli stalli rallenta il funzionamento della pipeline.

Per i conflitti strutturali esiste una soluzione hardware: aggiungendo all'ALU l'unità FPU (*Floating Point Unit*), consentiamo al sistema l'uso di questa unità in alternativa all'ALU, in modo tale che due stadi della pipeline utilizzino contemporaneamente le due unità (ALU e FPU), la prima per calcoli su numeri interi e la seconda per calcoli con numeri espressi in virgola mobile. Oppure, può essere risolto l'accesso contemporaneo alla memoria RAM da parte di due istruzioni successive dividendo la memoria cache di primo livello in due parti: memoria dati e memoria istruzioni, secondo l'architettura Harvard.
Il conflitto tra i dati può essere semplicemente spostato a un livello software: il compilatore può individuare le istruzioni che generano il conflitto sui dati interponendo tra esse una serie di istruzioni assembly inutili chiamate NOP (*No Operation*) con il compito di ritardare l'esecuzione della seconda istruzione per consentire l'adeguamento delle pipeline. Possiamo paragonare l'uso di questo metodo all'inserimento di uno stallo a livello software.
I conflitti di controllo rappresentano il problema maggiore, in quanto i salti condizionati necessari alla selezione e all'iterazione non danno la possibilità di riconoscere a priori se il salto sarà effettuato, né tanto meno la locazione di memoria da cui riprendere l'esecuzione del programma. Solo dopo la fase di Execute si può conoscere se il salto sarà effettuato e conseguentemente la locazione dalla quale far proseguire il programma.

Tecnologie superscalari

◀ Tecnologia superscalare Identifica la tecnologia costruttiva di microprocessori che sono in grado di eseguire mediamente più di un'operazione per ciclo di clock. ▶

Per realizzare CPU con prestazioni sempre migliori si è affermata la ◀ tecnologia superscalare ▶ grazie alla quale vengono integrati in un unico microprocessore più pipeline che funzionano in parallelo.
Nelle CPU attuali, inoltre, le pipeline non sono composte da soli cinque stadi ma, in realtà, ne utilizzano molti di più. Questo si è reso necessario per potere innalzare la frequenza di clock: infatti spezzando le singole operazioni elementari necessarie per completare un'istruzione si può elevare la frequenza della CPU. Questa scelta di progettazione consente effettivamente di aumentare la frequenza di funzionamento delle CPU ma rende critico il problema dei salti condizionati. Nel caso di un salto condizionato non previsto la CPU può essere costretta a svuotare e ricaricare una pipeline di 30 stadi perdendo fino a 30 cicli di clock contro una classica CPU a pipeline a cinque stadi che avrebbe sprecato nella peggiore delle ipotesi cinque cicli di clock. Nella figura che segue si vede il diagramma delle operazioni elementari di una CPU superscalare a doppia pipeline.



Inoltre la continua richiesta di maggiore potenza di calcolo ha spinto le industrie produttrici di microprocessori a integrare in un unico chip più microprocessori. Ciò consente al computer di avere due CPU separate dal punto di vista logico ma fisicamente risiedenti nello stesso chip. Questa strategia progettuale attenua i problemi di coerenza e di predizione dei salti. Infatti ogni CPU logica esegue un programma separato e quindi tra i diversi programmi non si possono avere problemi di coerenza tra le istruzioni.

Branch prediction

L'uso della pipeline è influenzato negativamente dai salti condizionati, dato che questi possono costringere il processore a ripetere gli stessi cicli macchina. Un processore basato su architettura x86 incontra un salto condizionato mediamente ogni circa 6/7 istruzioni. Pertanto è assai importante cercare di prevedere il salto condizionato, così da caricare in anticipo il blocco di istruzioni corretto nella pipeline.

Le evoluzioni che riguardano la memoria centrale

Cache memory

La cache è una memoria di tipo statico (SRAM) molto veloce e di dimensioni contenute, inserita tra la memoria centrale di tipo SDRAM e la CPU. Come abbiamo visto nelle Unità precedenti, consente un accesso più rapido alla memoria. Tuttavia, essendo di dimensioni ridotte rispetto alla RAM, la memoria cache è in grado di memorizzare soltanto una minima parte dei dati contenuti nella RAM. L'uso della cache è legato strettamente al comportamento delle applicazioni durante la loro esecuzione secondo due principi, denominati principi di località, descritti di seguito.

- Località spaziale: i programmi in esecuzione possono accedere a celle di memoria presenti all'interno di una zona definita, oppure possono accedere alle stesse locazioni di memoria già utilizzate, come per esempio all'interno di un ciclo.
- Località temporale: il programma eseguirà molto probabilmente le istruzioni accedendo alle istruzioni immediatamente successive a quella appena eseguita.

La cache funziona proprio sulla base di questi principi: la memoria intorno a quella utilizzata dal programma viene trasferita nella cache, la CPU non vedrà fisicamente la cache ma la utilizzerà in modo trasparente. In genere, all'aumentare della sua dimensione otteniamo un miglioramento nelle prestazioni del sistema. Vediamo quali sono i livelli di cache disponibili:

- cache integrata direttamente nel processore, chiamata cache di primo livello (L1);
- cache esterna collegata direttamente al processore, chiamata cache di secondo livello (L2); è collegata alla CPU tramite il Back Side Bus, come abbiamo già visto nelle Unità precedenti;
- cache esterna collegata sulla motherboard, denominata cache di terzo livello (L3).

La cache viene usata per operazioni di lettura e scrittura ed è suddivisa in due sezioni, una per le istruzioni e una per i dati. È organizzata in blocchi di celle chiamate linee; a ogni accesso in memoria centrale le informazioni vengono copiate in una linea della cache. Durante ogni operazione di lettura da parte della CPU avviene una ricerca delle informazioni nella cache: se presenti (operazione di Cache Hit) non serve accedere alla memoria centrale, altrimenti (operazione di Cache Miss) viene effettuata la lettura in memoria e le informazioni prelevate vengono memorizzate anche nella cache. Per le operazioni di scrittura in memoria la cache può essere usata con due modalità diverse:

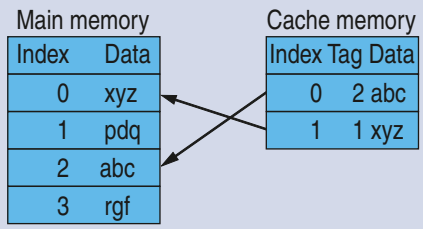
- cache Write Through: le operazioni di scrittura vengono effettuate sia nella cache che nella memoria centrale;
- cache Write Back: le operazioni di scrittura avvengono solo nella cache, mentre la linea di cache viene scritta in memoria solo quando viene sostituita.

La cache viene gestita attraverso tre possibili metodi:

- metodo diretto (o mappatura diretta);
- metodo completamente associativo;
- metodo associativo a N vie.

Nel metodo diretto a ciascuna linea o ◀ blocco ▶ di celle della memoria centrale, viene associato un blocco analogo nella cache che contiene gli stessi dati.
Nel metodo completamente associativo ogni blocco di memoria può essere allocato in qualsiasi linea della cache.
Nel metodo associativo a N vie la cache è suddivisa in gruppi di N linee ciascuno. Ogni blocco di memoria viene allocato in un preciso gruppo ma al suo interno può occupare qualsiasi linea.

◀ Blocco di memoria Ogni locazione della memoria nella memoria RAM (Main Memory) contiene un dato (Data) e possiede un indirizzo (Index). Ogni linea della cache contiene sia il dato della cella di memoria al quale fa riferimento sia l'indirizzo di memoria che lo contiene, quest'ultimo suddiviso in due valori: Index e Tag. Index rappresenta lo spiazzamento a partire dal Tag, mentre il Tag è l'indirizzo di partenza del blocco. Nelle cache dati questi valori sono chiamati blocchi di cache o linee di cache. ▶



Memoria virtuale

La memoria virtuale rappresenta un metodo per far apparire la memoria centrale di dimensione maggiore rispetto a quella effettivamente allocata. Di questo si occupa un componente chiamato ◀ MMU ▶ (*Memory Management Unit*).
Le MMU moderne generalmente suddividono lo spazio degli indirizzi virtuali (l'intervallo di indirizzi accessibili dal processore) in pagine di memoria di dimensione di pochi kB. Gli N bit meno significativi dell'indirizzo rappresentano l'offset interno della pagina e i bit restanti rappresentano il numero virtuale della pagina.
La MMU contiene una tabella delle pagine indicizzata dal numero della pagina. Ogni elemento di questa tabella si chiama PTE (*Page Table Entry*) e restituisce il numero fisico della pagina corrispondente a quello virtuale che, combinato con l'offset della pagina, forma l'idirizzo fisico completo.

◀ MMU Si tratta di un processore in grado di traslare o tradure gli indirizzi di memoria virtuali in indirizzi di memoria fisici, di effettuare la protezione della memoria, il controllo della cache della CPU, l'arbitraggio del bus e, in architetture più semplici, la commutazione di banchi di memoria. ▶

È possibile che non esista memoria fisica (RAM) allocata a una data pagina virtuale. In questo caso la MMU segnala una condizione di "pagina di memoria mancante" (*page fault*) alla CPU. Il sistema operativo interviene per gestire questa condizione, cerca di trovare una pagina libera nella RAM e di creare una nuova PTE nella quale mappare l'indirizzo virtuale richiesto nell'indirizzo fisico della pagina trovata.

Quando non c'è spazio disponibile nella RAM per una nuova pagina di memoria, può essere necessario scegliere una pagina esistente utilizzando un algoritmo di sostituzione, farne una copia su disco rigido e rimpiazzarla con quella nuova. Analogamente, quando non ci sono PTE inutilizzate a disposizione, il sistema operativo deve liberarne una.

Il sistema operativo assegna a ogni processo il suo spazio di indirizzamento virtuale.

Le evoluzioni che riguardano gli I/O

DMA (direct memory access)

Abbiamo visto in precedenza come i dispositivi periferici segnalino la disponibilità a emettere o accettare dati attraverso l'interrupt. Tuttavia la tecnica dell'interruzione è particolarmente inefficiente se il trasferimento riguarda grosse moli di dati (hard disk, scheda video, scheda di rete). In questi casi è opportuno che dopo la segnalazione la CPU deleghi le attività di trasferimento a un dispositivo accessorio che accede ai bus di sistema quando non sono usati dalla CPU. In questo modo la CPU è coinvolta solo nell'avvio delle operazioni e nella terminazione.
Il DMA controller consente l'impostazione dei trasferimenti tramite un insieme di porte di configurazione visibili dalla CPU. Il programma imposta l'indirizzo iniziale del blocco di memoria da trasferire e le dimensioni. Una volta iniziato il trasferimento il controller DMA provvederà autonomamente al trasferimento dei dati da o per la memoria. Il controller DMA condivide i bus con la CPU, quindi il DMA trasferisce quando la CPU non usa i bus (elaborazioni interne).

Coprocessori

Sono CPU secondarie dedicate a scopi specifici che scaricano la CPU principale da alcune elaborazioni consentendo un maggior parallelismo.
Di seguito alcuni esempi tipici:

- coprocessore numerico: realizza una grande quantità di elaborazioni numeriche soprattutto in virgola mobile non realizzate dalla CPU principale. Può anche essere integrato nello stesso chip della CPU principale;
- coprocessore grafico: mantiene una memoria immagine della presentazione video e realizza elaborazioni grafiche per la costruzione della presentazione.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Quale tra le seguenti metodologie non può garantire un miglioramento delle prestazioni secondo il modello di Von Neumann?

- ○ aumento della frequenza di Clock
- ○ aumentare l'ampiezza di parola
- ○ aumentare il numero di processori
- ○ aumentare lo spazio di indirizzamento

2 Collega il grado di parallelismo a sinistra con la relativa definizione posta a destra:

a) SISD	permette di eseguire contemporaneamente la stessa operazione su più dati	.....
b) SIMD	un unico processore esegue un'unica istruzione per volta e può prelevare e depositare in memoria un solo dato per volta	.....
c) MISD	è relativa ai sistemi multiprocessori	.....
d) MIMD	consente di iniziare l'esecuzione di diverse istruzioni, ognuna sui propri dati.	.....

3 La CPU individua le istruzioni non vincolate alle successive per essere eseguite in parallelo. Si tratta della tecnica chiamata:

- ○ esecuzione fuori ordine
- ○ branch prediction
- ○ prefecth
- ○ speculative execution

4 Quali problemi può generare il precaricamento?

- ○ si effettua una divisione e uno dei due operandi non è stato ancora calcolato dall'istruzione precedente
- ○ la memoria è troppo lenta rispetto alla CPU
- ○ la CPU carica delle istruzioni dipendenti da un salto e quel salto non viene eseguito
- ○ la memoria è troppo veloce rispetto alla CPU

5 Metti in ordine crescente le seguenti fasi eseguite da una CPU:

a) EX .....
b) ID .....
c) WB .....
d) MEM .....
e) IF .....

6 Una CPU con pipeline possiede:

- ○ una ALU ad alta velocità
- ○ cinque stadi, uno per ciclo macchina
- ○ cinque CPU, una per ciclo macchina
- ○ cinque fasi, eseguite in modo sequenziale

7 Nella tecnologia superscalare:

- ○ in una CPU vengono integrate più CPU che lavorano in parallelo
- ○ in una CPU vengono integrate più pipeline che lavorano in parallelo
- ○ in una CPU vengono integrati più stadi che lavorano in parallelo
- ○ nella motherboard vengono integrate più CPU che lavorano in parallelo

8 La cache è una memoria di tipo:

- ○ ROM
- ○ statico (DRAM)
- ○ dinamico (SRAM)
- ○ statico (SRAM)

9 Collega il tipo di cache a sinistra con il nome esteso posto a destra:

a) cache esterna collegata direttamente al processore	L1	.....
b) cache esterna collegata sulla motherboard	L2	.....
c) cache integrata	L3	.....

Esercizi di completamento

1 Un metodo per risolvere la criticità della pipeline è ..... che consiste nell'inserimento di alcuni ..... per bloccare temporaneamente una parte della pipeline in attesa che si risolva il conflitto.

2 Le principali problematiche che si possono manifestare riguardo all'impiego della pipeline sono: ....., ... e .....

3 Le criticità strutturali della pipeline si possono risolvere aggiungendo all' ..... una ..... in modo tale che due stadi della pipeline le utilizzino contemporaneamente.

4 Il conflitto tra i dati può essere risolto aggiungendo delle ..... che ..... l'esecuzione delle istruzioni per consentire l'adeguamento della .....

5 Tramite la tecnologia superscalare si integrano diverse ..... nella stessa .....

6 La tecnologia superscalare consente effettivamente di aumentare la ..... di funzionamento delle CPU ma rende critico il problema dei .....

7 La ..... indica che i programmi in esecuzione che possono accedere a celle di memoria presenti all'interno di una zona definita, oppure possono accedere alle stesse locazioni di memoria già utilizzate, come per esempio all'interno di un ciclo, mentre nel principio di ..... il programma eseguirà molto probabilmente le istruzioni accedendo a istruzioni immediatamente successive a quella appena eseguita.

8 Cache ..... indica che le operazioni di scrittura che vengono effettuate sia nella cache che nella memoria centrale, mentre con la cache ..... le operazioni di scrittura avvengono solo nella cache e la linea di cache viene scritta in memoria solo quando viene sostituita.

9 Il DMA controller, consente l'impostazione dei ..... tramite un insieme di ..... di configurazione visibili dalla CPU.

2 MODULO L'ISA X86 E IL LINGUAGGIO ASSEMBLY

OBIETTIVI

- Conoscere la struttura del processore 8086
- Conoscere il modello di programmazione x86 a 16 e 32 bit
- Riconoscere la struttura dello stack
- Saper distinguere gli elementi che concorrono all'assemblaggio
- Conoscere le istruzioni principali dell'ISA x86
- Conoscere la struttura di un programma assembly
- Conoscere i metodi di indirizzamento

ATTIVITÀ

- Scrivere programmi in assembly x86
- Usare istruzioni di salto condizionato e incondizionato
- Realizzare i cicli in assembly
- Utilizzare le principali istruzioni aritmetiche
- Utilizzare i principali servizi DOS di lettura e scrittura a video/tastiera
- Scrivere programmi assembly contenenti procedure
- Installare l'emulatore MS-DOS DOSBox e l'assemblatore TASM

UNITÀ DIDATTICA 1

IL PROCESSORE 8086

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere la struttura del processore 8086
- a comprendere il significato dei registri
- a riconoscere gli elementi che costituiscono l'architettura 8086

I microprocessori Intel

Come abbiamo visto il microprocessore è un circuito integrato dotato di una struttura circuitale in grado di attuare un prefissato ◀ set di istruzioni ▶.

◀ Set di istruzioni Si definisce set di istruzioni, dall'inglese ISA (*Instruction Set Architecture*) l'insieme delle istruzioni utilizzabili dal programmatore e legate strettamente all'architettura hardware di un calcolatore. Appartengono all'ISA la dimensione e il tipo dei vari registri della CPU, i tipi di dati, le istruzioni e le modalità di indirizzamento. ▶

Sul mercato sono disponibili diversi tipi di microprocessori, tra loro differenti sia a livello fisico sia a livello software; tuttavia possiamo distinguere i principali fattori che li differenziano:

- la dimensione delle celle di memoria (8, 16, 32 e 64 bit);
- il numero e il tipo di registri interni;
- l'ampiezza dei bus;
- la dimensione delle istruzioni.

Per quanto riguarda il processore 8086 i dati, che verranno ampiamente discussi in seguito, sono i seguenti:

- la dimensione delle celle di memoria (1 byte);
- il numero e il tipo di registri interni (8, 16 e 32 bit);
- l'ampiezza dei bus (bus dati: 16 bit, bus indirizzi: 20 bit);
- la dimensione delle istruzioni (da 1 a 6 byte).

Dal punto di vista strutturale il microprocessore risulta composto da tre blocchi:

1. l'ALU (*Arithmetic Logic Unit*);
2. i registri, divisibili in due categorie: general purpose e speciali;
3. l'UC e il temporizzatore.

L'ALU

L'unità aritmetico-logica è il componente che esegue le elaborazioni richieste alla CPU. Le operazioni di elaborazione eseguite dall'ALU sono di tipo aritmetico e logico. La CPU, mediante appositi segnali di controllo che variano a seconda dell'istruzione, indica all'ALU il tipo di operazione da eseguire. Tutti i segnali collegati in ingresso all'ALU sono ◀ bufferizzati ▶, mentre i segnali di uscita confluiscono in un registro chiamato accumulatore.

◀ Un segnale bufferizzato è un segnale che viene mantenuto costante fino alla variazione successiva del clock. A ogni variazione del clock viene memorizzato un valore di ingresso che rimane costante in uscita fino alla variazione successiva del clock. In questo caso il dato in ingresso all'ALU è mantenuto costante da un registro temporaneo che svolge il compito di buffer. ▶

I registri general purpose

I registri general purpose vengono usati principalmente per memorizzare temporaneamente i dati dei programmi e variano a seconda dell'ISA del sistema. Generalmente consentono di memorizzare tre categorie di dati:

- operandi;
- indici;
- indirizzi.

Questi registri, la cui dimensione è definita dall'ISA del sistema, in genere possono essere accoppiati per consentire una lunghezza maggiore.

I registri speciali

Anche il numero di registri speciali varia a seconda dell'ISA del sistema, tuttavia alcuni registri sono sempre presenti:

- IP (*Instruction Pointer*)
  Contiene l'indirizzo dell'istruzione che deve essere eseguita immediatamente dopo quella corrente.
- SP (*Stack Pointer*)
  Contiene l'indirizzo di una specifica zona di memoria organizzata secondo la filosofia ◀ LIFO ▶ utile all'esecuzione delle procedure, delle funzioni e dei parametri passati a esse.
- IR (*Instruction Register*)
  Contiene il codice operativo dell'istruzione che è stata prelevata durante la fase di Fetch. È collegato alla sezione di decodifica e, a seconda della logica della CPU, è formato da una coda di istruzioni.
- AR (*Address Register*)
  Sono registri che consentono la segmentazione della memoria e sono divisi in:
  – CS (*Code Segment*)
  – DS (*Data Segment*)
  – ES (*Extra Segment*)
  – SS (*Stack Segment*)
- Registro dei flag
  Consiste in un gruppo di bit che consentono di ottenere tutta una serie di informazioni sullo stato dei risultati dell'ultima operazione aritmetico-logica eseguita.

◀ LIFO significa Last In First Out e indica una struttura di memoria chiamata anche pila in cui l'ultimo elemento inserito rappresenta il primo a essere estratto. ▶

La tabella seguente riassume i principali registri del processore 8086, che verranno ampiamente trattati nelle unità successive:

	15 8	7 0	
Accumulator	AH	AL	AX
Base	BH	BL	BX
Counter	CH	CL	CX
Data	DH	DL	DX

	15 0
Instruction Pointer	IP
Stack Pointer	SP
Base Pointer	BP
Source Index	SI
Destination Index	DI

	15 0
Code Segment	CS
Data Segment	DS
Stack Segment	SS
Extra Segment	ES

	15 0
Flag Register	

L'UC e il temporizzatore

L'unità di controllo (UC o CU, *Control Unit*), unitamente al temporizzatore, definisce la cadenza con cui si susseguono in modo sincrono le varie operazioni del sistema. La sezione della CPU legata all'UC e al temporizzatore genera i comandi eseguibili e il segnale di clock, utile alla sincronizzazione del sistema. L'UC riceve direttamente dall'IR il codice operativo dell'istruzione traducendola in una prefissata sequenza di comandi. Oltre a questi tre blocchi funzionali esistono i bus che, come abbiamo visto nel modulo precedente, si occupano di trasferire i dati tra i vari blocchi funzionali, ed esternamente verso la memoria e i dispositivi di I/O.

Zoom su...

I PROCESSORI INTEL E I COMPUTER CHE LI OSPITANO


XT


AT

I computer che possiedono una CPU Intel vengono classificati nel modo seguente:

- XT (*Extended Technology*)
  Computer dotati di microprocessori 8088.
- AT (*Advanced Technology*)
  Computer dotati di microprocessori 80286, 80386, 80486, 80586 (Pentium).

Tali CPU possiedono anche una circuiteria di supporto rappresentata da altri componenti elettronici così riassunti:

Gestione CPU

- iAPx87 NDP (*Numeric Data Processor*), coprocessore numerico;
- 8089 IOP (*Input Output Processor*), gestore dei dispositivi di I/O.

Gestione interfacciamenti con i bus

- 8288 controllore del bus;
- 8284 circuito generatore del segnale di clock a 4.77 MHz.

Controllo del sistema

- 8253 PIT (*Programmable Interval Timer*), gestore dei segnali di temporizzazione;
- 8237 DMA (*Direct Memory Access*), gestore dell'accesso diretto alla memoria;
- 8259 PIC (*Programmable Interrupt Controller*), gestore delle interruzioni.

Interfacciamenti esterni

- 8250 UART (*Universal Asyncronous Receiver Transmitter*), gestore dell'interfaccia seriale;
- 8255 PPI (*Programmable Peripheral Interface*), gestore della comunicazione con dispositivi esterni.

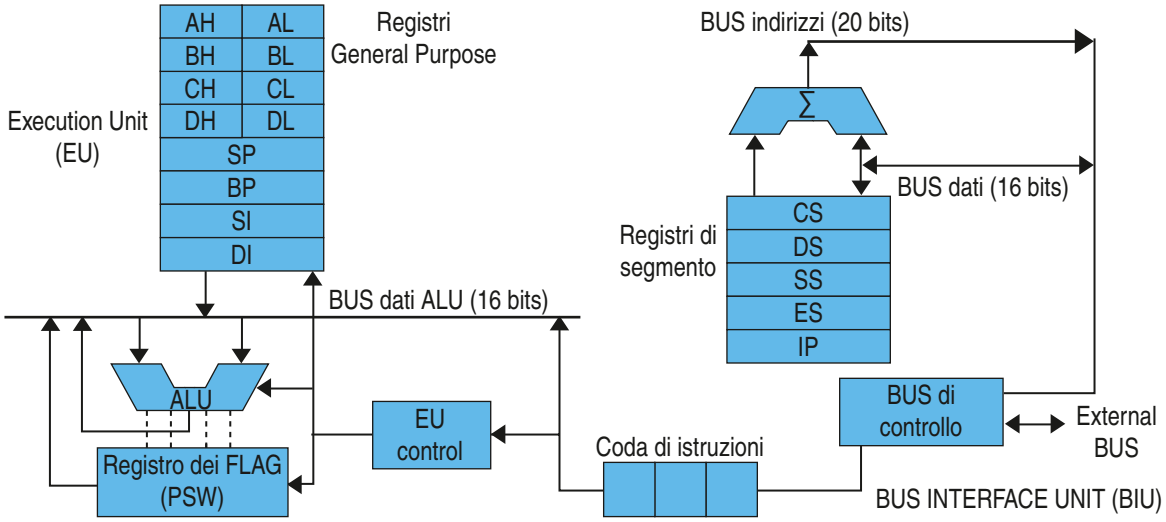
Il processore 8086

Il processore Intel 8086, chiamato anche iAPx86/10, introdotto sul mercato nel giugno 1978, è stato il primo microprocessore della seconda generazione a 16 bit. È contenuto in un chip al silicio a 40 pin i cui piedini sono disposti ordinatamente su due linee contenenti ognuna 20 piedini, a ciascuno dei quali fa capo un segnale specifico.

Ha avuto un'enorme diffusione soprattutto perché Microsoft lo ha utilizzato per costruire il sistema operativo MS-DOS, diventato subito uno standard grazie a IBM, che l'ha scelto per le sue prime linee di PC.

L'8086 è costituito da due sottosistemi che operano in modo parzialmente indipendente e asincrono:
- EU (*Execution Unit*);
- BIU (*Bus Interface Unit*).

La EU opera in parallelo con la BIU: in particolare, mentre il bus non è usato, per esempio durante la fase di execute delle istruzioni, la BIU esegue la Fetch delle istruzioni successive.



La EU costituisce la parte della CPU che elabora, ed è costituita da:
- registri general purpose (AX, BX, CX, DX);
- registri speciali (PSW o registro dei flag);
- unità di controllo (EU control);
- unità aritmetico-logica (ALU).

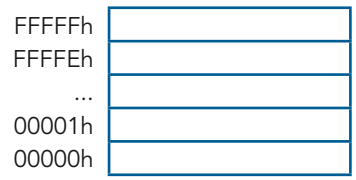
La BIU gestisce l'indirizzamento, il prelievo dei dati e delle istruzioni dalla memoria e gestisce il colloquio con i dispositivi esterni. È costituita da:
- logica di controllo dei bus;
- registri di segmento: CS, DS, SS, ES;
- registro contatore di programma: IP;
- registri puntatore: BP, SP, DI, SI;
- coda delle istruzioni, registro FIFO a 6 byte, in cui vengono memorizzate le istruzioni da eseguire.

Attraverso l'architettura a pipeline la BIU esegue le seguenti operazioni:

- preleva le istruzioni dalla memoria e le mette in coda nel registro Instruction Queue mediante la tecnica chiamata prefetching;
- preleva l'istruzione dalla coda, la pone nel registro IR e la esegue;
- in caso di interruzione, di istruzioni di salto (Jump) o di chiamata a procedura (Call), la coda delle istruzioni viene svuotata e riempita con le altre istruzioni presenti nella destinazione indicata.

L'organizzazione della memoria

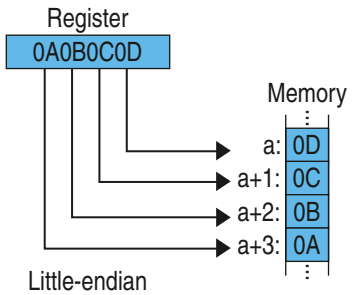
La CPU 8086 possiede un bus indirizzi a 20 bit, uno spazio di indirizzamento di 2^{20} bit = 1048576 byte = 1024 kB = 1 MB. Gli indirizzi validi vanno pertanto da: ▶



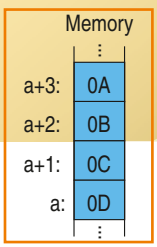
◀ Il termine little endian deriva dal famoso romanzo *I viaggi di Gulliver*, nel quale due popoli erano in guerra a causa del modo con cui venivano rotte le uova, dall'estremità più piccola (*little endian*) o dall'estremità più grossa (*big endian*). ▶

Come avete notato abbiamo indicato le celle di memoria partendo dal basso. Questo accade perchè è più conveniente disporre le celle secondo la notazione ◀ little endian ▶.

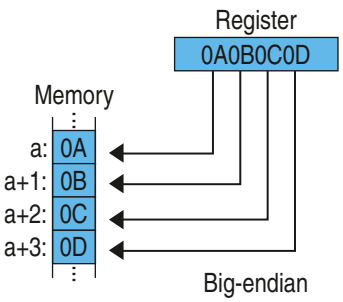
Si tratta di una tecnica usata dai processori Intel secondo la quale si presuppone che i dati memorizzati in una word abbiano il byte meno significativo memorizzato nella cella di indirizzo minore. Questo modo di operare viene anche chiamato backwords storage. Si contrappone alla tecnica big endian usata dai processori Motorola, secondo la quale invece i dati memorizzati in una word hanno il byte meno significativo memorizzato nella cella di indirizzo maggiore. Per esempio se memorizziamo a partire dalla cella di indirizzo a un numero formato da 4 byte (0A0B0C0D), quello meno significativo (0D) viene inserito nella cella di indirizzo più basso (a), mentre i byte più significativi vengono immessi nelle celle successive, cioè di indirizzo maggiore, come indicato dalla figura a lato. ▶



Come possiamo notare il numero scritto in memoria diventa tuttavia di difficile lettura, infatti sembra scritto al contrario. Proprio per facilitare la lettura dei numeri a più byte, viene disposta la memoria a rovescio, cioè dal basso verso l'alto. In tal modo il numero è leggibile più facilmente.



Secondo la tecnica big endian invece se memorizziamo a partire dalla cella di indirizzo a un numero formato da 4 byte (0A0B0C0D), quello più significativo (0A) viene inserito nella cella di indirizzo più basso (a), mentre i byte meno significativi vengono immessi nelle celle successive, cioè di indirizzo maggiore, come indicato dalla figura a lato. ▶



Fisicamente la memoria è organizzata in due banchi da 512 kB ciascuno: un banco di indirizzi pari trasferiti sui pin D_0-D_7 del bus dati e un banco di indirizzi dispari trasferiti sui pin D_8-D_{15} del bus dati. I due banchi sono indirizzati in parallelo dai segnali A_0-A_{19} del bus indirizzi. Il processore seleziona contemporaneamente la cella di indirizzo pari e la corrispondente di indirizzo

dispari, decodificando le linee di indirizzo $A_1..A_{19}$. La linea di indirizzo A_0 è usata per abilitare il banco pari, mentre quello dispari è abilitato da un segnale di controllo chiamato BHE (Bus High Enable). Quando la CPU deve indirizzare un singolo byte situato in una locazione pari porta BHE a 1, in quanto l'indirizzo pari possiede $A_0 = 0$. Se al contrario il byte è in posizione dispari, la CPU porta BHE a 0 e $A_0 = 1$.

Se invece il dato da indirizzare è una word localizzata in una coppia di celle a partire da un indirizzo pari, la CPU porta BHE a 0, essendo poi $A_0 = 0$ vengono attivate entrambe le celle.

A0	BHE	Significato
0	0	Trasferimento di una word intera
1	0	Trasferimento di un byte alto su indirizzo dispari
0	1	Trasferimento di un byte basso su indirizzo pari
1	1	Situazione impossibile

Quando la CPU deve indirizzare una word su indirizzo dispari, l'operazione viene suddivisa in due operazioni consecutive nelle quali avviene il trasferimento di un singolo byte. In sostanza quando vogliamo leggere e scrivere una word su indirizzo dispari otteniamo una penalizzazione in termini di tempo di esecuzione.

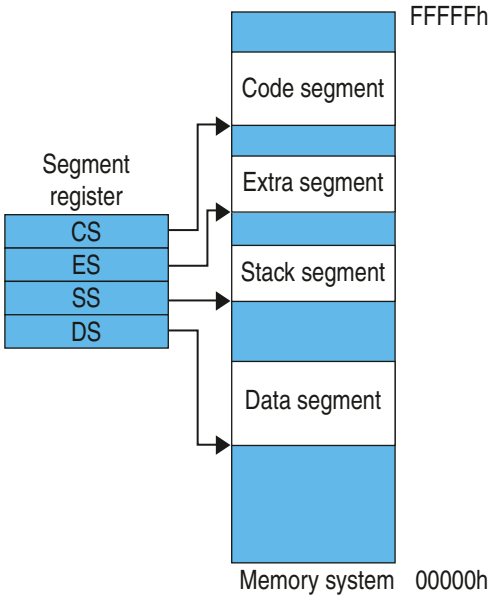
La memoria è logicamente suddivisa in quattro segmenti da 64 kB; ogni segmento può essere indirizzato al suo interno mediante 16 bit, infatti $2^{16} = 65536 = 64$ kB. Questi segmenti prendono il nome dei corrispondenti registri:

CS = *Code Segment*
DS = *Data Segment*
ES = *Extra Segment*
SS = *Stack Segment*

L'indirizzo logico della cella di memoria alla quale vogliamo accedere è dato dall'insieme del segmento e dell'offset rappresentati dai due registri:

CS : IP

Il registro di segmento punta alla cella iniziale, chiamata base, del corrispondente segmento.
L'offset invece stabilisce di quanto ci si deve spostare all'interno del segmento, dalla base stessa. ▶



Indirizzi fisici e logici

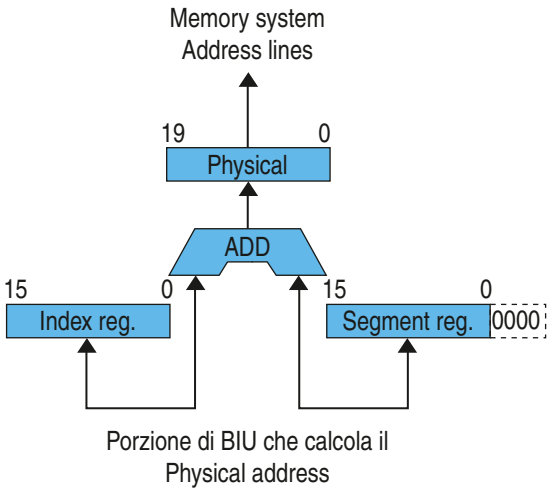
L'indirizzo fisico di una cella di memoria è espresso da 20 bit; non possiamo perciò indirizzare le celle solo mediante un unico registro a 16 bit. L'indirizzo fisico si ottiene infatti come somma data da due indirizzi chiamati segmento e offset.

Per calcolare l'indirizzo fisico a 20 bit, partendo dall'indirizzo logico, è necessario moltiplicare per 16 il contenuto del registro di segmento. Sommando al dato ottenuto l'offset si ottiene l'indirizzo fisico:

(SEGMENT * 16) + OFFSET

Calcoliamo l'indirizzo fisico avente come segmento B02Ah e come offset 120Ch:

```
B02A0  +
 120C  =
----------
B14AC
```

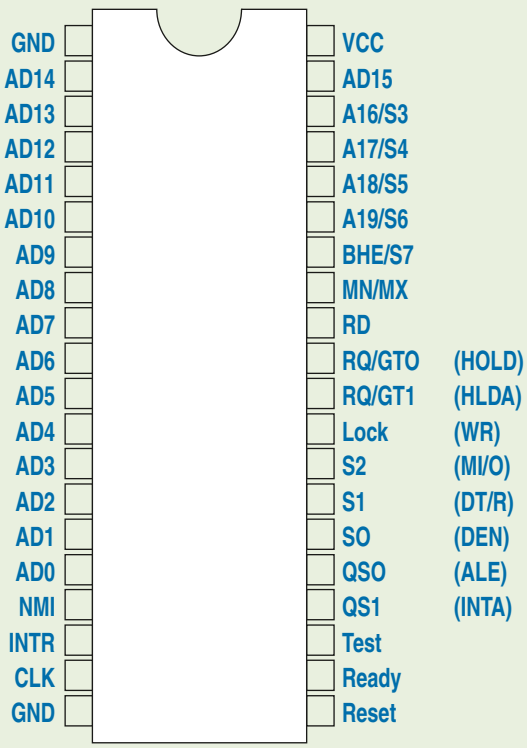


Durante l'esecuzione di un programma assembly non dovremo preoccuparci di tradurre indirizzi logici in indirizzi fisici: in sostanza il programmatore lavora soltanto con indirizzi logici.

Zoom su...

LA PIEDINATURA DELL'8086

La figura seguente illustra la piedinatura dell'8086. Tra parentesi vi sono i segnali in modo minimo:



GND
Massa

VCC
Alimentazione

CLK
Segnale di clock

AD_0-AD_{15}
Durante il primo periodo di clock T1, in un ciclo di accesso alla memoria o ai dispositivi di I/O, su questi pin transita l'indirizzo, mentre negli altri periodi di clock T2, T3, T4 transitano i dati.

A_{16}/S3, A_{17}/S4, A_{18}/S5, A_{19}/S6
Durante il T1 su queste linee sono presenti i 4 bit più significativi dell'indirizzo; durante gli altri stati T2, T3, T4 queste linee forniscono informazioni di stato:

- S3 e S4 individua il segmento utilizzato nel ciclo di bus;
- S5 è uguale al flag di interrupt;
- S6 = 0 indica il collegamento dell'8086 al bus;

BHE/S7
Bus High Enable, attivo durante lo stato T1, determina il trasferimento tra CPU e memoria di un singolo byte o di tutta la parola a 16 bit. S7 viene utilizzato quando è in corso un DMA o durante il primo ciclo di riconoscimento di un'interruzione.

RD
Read è attivo durante un ciclo di lettura dalla memoria o da un dispositivo di I/O.

READY
Segnale inviato dalla memoria o dal dispositivo di I/O per segnalare che l'operazione richiesta è stata terminata. Questo segnale deve essere sincronizzato con il clock.

TEST
Segnale di ingresso che viene esaminato dall'istruzione Wait. Se TEST è basso l'8086 continua normalmente, altrimenti si arresta e rimane in attesa fino a quando TEST non ritorna a 0.

NMI
Richiesta di interruzione non mascherabile.

INTR
Richiesta di interruzione mascherabile.

RESET
Fa terminare l'attività in corso della CPU. È attivo quando è alto, azzera alcuni registri, elimina la coda di pipeline e mette la CPU in condizione di leggere l'istruzione memorizzata nella locazione FFFF0h. Quando ritorna basso, il microprocessore riprende il suo funzionamento come se fosse stato acceso per la prima volta.

MN/MX
Indica in quale modo deve operare il microprocessore, definendo le funzioni dei piedini da 24 a 31.

Significato dei piedini in modo minimo

M/IO
Indica se l'operazione in corso si riferisce alla memoria o a un dispositivo di I/O.

WR
Indica che il microprocessore sta effettuando un ciclo di scrittura in memoria o in un dispositivo di I/O.

INTA
Riconoscimento di interruzione mascherabile.

ALE
Address Lach Enable è generato dal microprocessore per memorizzare esternamente un indirizzo. È attivo durante lo stato T1.

DT/R
Data Trasmit/Receive è un segnale usato per controllare il verso di trasmissione dei dati, quando l'8086 è collegato a un tranceiver (trasmettitore-ricevitore) tipo 8286.

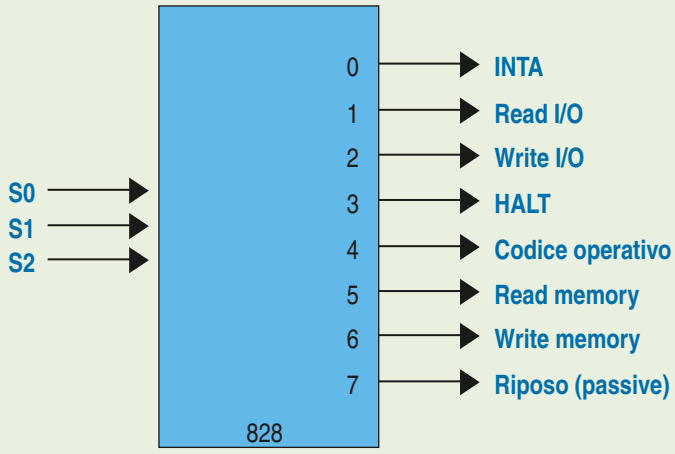
DEN
Data Enable è attivo durante i cicli di accesso alla memoria o ai dispositivi di I/O e nei cicli di riconoscimento di interrupt.

HOLD
Segnala che un altro master ha richiesto il bus. Il microprocessore riacquista il controllo del bus quando HOLD ritorna basso.

HLDA
Viene attivato quando l'8086 rilascia il bus a un altro master. In questo caso tutte le linee di indirizzo e quelle di controllo di tipo OUT (tranne ALE) vengono messe in alta impedenza.

Significato dei piedini in modo massimo

S2, S1, S0
In base alla tabella seguente, questi segnali di stato vengono usati dal controller 8288 per generare i segnali di controllo per l'accesso alla memoria e ai dispositivi di I/O:



S2	S1	S2	
0	0	0	INTA
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	HALT
1	0	0	Codice operativo

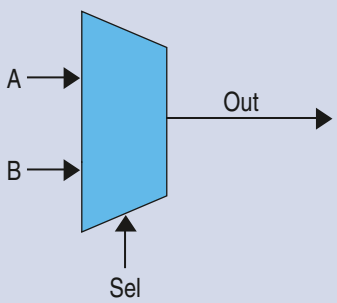
RQ/GT0, RQ/GT1
Request/Grand: utilizzati separatamente per ricevere la richiesta di un bus da parte di un altro master e per trasmettere il segnale di accettazione della richiesta.

LOCK
Quando è attivo impedisce che gli altri master prendano il controllo.

QS0, QS1
Queue status: indica all'esterno lo stato della coda di pipeline.

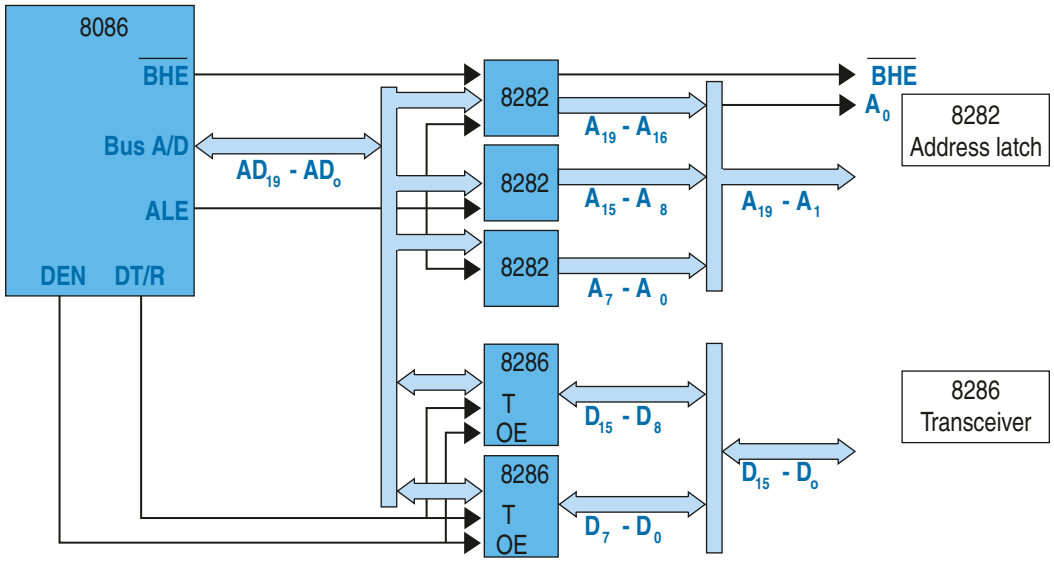
La configurazione del sistema

◀ Il multiplexaggio è una tecnica che consente di selezionare un singolo segnale elettrico tra diversi segnali in ingresso, in base al valore degli ingressi di selezione. Nella figura seguente si nota che l'uscita è multiplexata tra due linee (A e B). Il segnale sel seleziona quale linea deve essere collegata con l'uscita (out). ▶



Come abbiamo visto la CPU 8086 possiede 40 piedini: si tratta di una limitazione che impone il ◀ multiplexaggio ▶ delle linee indirizzi e dati. Il bus di sistema è organizzato in modo tale che alcune linee del bus dati e del bus indirizzi siano, per così dire, in comune.
Le linee del bus dati e indirizzi condividono gli stessi segnali AD_0-AD_{15}. Il bus indirizzi possiede inoltre altre 4 linee A_{16}-A_{19}.
L'8086 opera secondo due diverse configurazioni chiamate modo minimo e modo massimo a seconda del segnale inserito nel pin numero 33. Le due modalità agiscono direttamente sul bus di controllo, creando per così dire due sistemi diversi. Il significato dei piedini posti tra il numero 24 e il numero 31 cambia a seconda della configurazione scelta.
Il modo minimo si ottiene ponendo l'ingresso MN/MX = 1, mentre il modo massimo si ottiene ponendo il segnale in ingresso MN/MX = 0. In modo massimo la CPU 8086 viene interfacciata alle memorie e ai dispositivi di I/O mediante il controllore di bus 8288 e può lavorare in tandem con i processori NDP 8087 e IOP 8089.
In modo minimo il microprocessore genera direttamente i segnali di controllo necessari per interfacciare le memorie e i dispositivi di I/O.

La figura seguente mostra come avviene il multiplexaggio dei bus:



Il ciclo di bus è l'insieme delle fasi usate dalla CPU per comunicare con la memoria, con un dispositivo di I/O, con l'interrupt controller. Si compone di quattro fasi denominate T_1, T_2, T_3, T_4. Vediamo la sequenza di queste fasi:

T_1 La CPU pone l'indirizzo sul bus indirizzi (A_{19}-A_{16} e AD_{15}-AD_0).

T_2, T_3, T_4 Il dato viene successivamente posto sul bus dati (AD_{15}-AD_0).

◀ Un latch è un circuito elettronico formato da due stati stabili, in grado di memorizzare un bit di informazione nei sistemi a logica sequenziale asincrona. Il latch modifica lo stato logico dell'uscita al variare del segnale di ingresso, mentre il flip-flop, basato sulla struttura del latch, cambia lo stato logico dell'uscita solamente quando il segnale di clock è nel semiperiodo attivo. ▶

Durante la fase T_1 la CPU attiva il segnale di controllo ALE (*Address Latch Enable*) che consente di memorizzare l'indirizzo su registri ◀ latch ▶ (8282).

Durante le fasi T_3-T_4 vengono invece attivati i segnali di controllo DEN (*Data Enable*) e DT/R (*Data Tansmit/Receive*), per indicare rispettivamente la presenza di un dato e il suo verso.

Quando l'8086 è in modo minimo ALE, DEN e DT/R vengono forniti direttamente dalla CPU, mentre nel modo massimo vengono forniti dal bus controller 8288.

Se la CPU non deve accedere all'esterno, i segnali di controllo del bus sono inattivi e il bus si trova nello stato di idle o stato di alta impedenza.

Ciclo di lettura

T_1 L'indirizzo della cella da leggere viene messo dalla CPU sul bus indirizzi.

T_2 La CPU forza il bus dati in stato di alta impedenza.

T_3, T_4 La memoria invia il dato sul bus dati e la CPU lo legge.

Ciclo di scrittura

T_1 L'indirizzo della cella su cui scrivere viene messo dalla CPU sul bus indirizzi.

T_2 La CPU invia il dato da scrivere sul bus dati.

T_3, T_4 La memoria legge il dato proveniente dal bus dati e lo memorizza.

Ciclo di wait

Quando la memoria non è ancora pronta a inviare o ricevere un dato, lo segnala alla CPU inserendo un ciclo di attesa (wait) tra i periodi T_3 e T_4. Per comunicare all'8086 la necessità di uno o più cicli di wait, la memoria esterna invia un segnale sul pin READY del microprocessore.

Ciclo idle

Questi cicli di attesa vengono aggiunti automaticamente dalla CPU quando si verificano le seguenti situazioni:

- la CPU non necessita di nuovi dati;
- la coda interna delle istruzioni è piena e non può essere eseguita alcuna fase di prefetch.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Quali tra i seguenti elementi sono definiti nell'ISA? (3 risposte)**

- ○ dimensione dell'hard disk
- ○ insieme delle istruzioni assembly
- ○ registri della CPU
- ○ sistema operativo
- ○ metodi di indirizzamento
- ○ frequenza di lavoro della CPU

2 **Quale tra i seguenti fattori non è utile a differenziare tra loro i vari microprocessori presenti sul mercato in termini di architettura:**

- ○ il numero ed il tipo di registri
- ○ la dimensione delle celle di memoria
- ○ l'ampiezza dei bus
- ○ la dimensione della cache
- ○ la frequenza di lavoro
- ○ la dimensione delle istruzioni

3 **I registri general purpose non consentono di memorizzare:**

- ○ indirizzi
- ○ operandi
- ○ indici
- ○ parametri

4 **Seleziona il registro speciale che contiene l'indirizzo dell'istruzione successiva da eseguire:**

- ○ IP
- ○ IR
- ○ SP
- ○ AR

5 **Seleziona il registro che consente di memorizzare il codice operativo dell'istruzione da decodificare:**

- ○ SP
- ○ IR
- ○ AR
- ○ IP

6 **Quale tra i seguenti registri contiene un indirizzo di segmento?**

- ○ IP
- ○ IR
- ○ SP
- ○ AR

7 **Quale tra le seguenti è la sigla del processore 8086?**

- ○ 8087
- ○ iAPx86/10
- ○ iAPx87
- ○ 8089

8 **Quali sono i due sottoinsiemi che lavoro in modo coordinato secondo il processore 8086? (2 risposte)**

- ○ EU
- ○ UC
- ○ ALU
- ○ BIU

9 **L'ALU è contenuta:**

- ○ nella UC
- ○ nella BIU
- ○ nella EU
- ○ nella RAM

10 **Metti in ordine le seguenti operazioni svolte dalla BIU dell'8086:**

- ○ in caso di salto la coda delle istruzioni viene svuotata e riempita con le altre istruzioni presenti nella destinazione indicata
- ○ preleva le istruzioni dalla memoria e le mette in coda nel registro Instruction Queue mediante la tecnica chiamata prefetching
- ○ preleva l'istruzione dalla coda e la esegue

11 **Indica lo spazio di memoria con un bus indirizzi a 24 bit:**

- ○ 1MB
- ○ 8 MB
- ○ 16 MB
- ○ 1GB

12 **Avendo a disposizione un bus indirizzi a 20 bit, qual è l'indirizzo di memoria valido più alto?**

- ○ FFFFh
- ○ 0000h
- ○ 10000h
- ○ FFFFFh

13 **Se il segnale BHE=0 e il segnale A0=1 del BUS indirizzi otteniamo:**

- ○ un trasferimento di una word su indirizzo pari
- ○ un trasferimento di una word su indirizzo dispari
- ○ un trasferimento di un byte su indirizzo pari
- ○ un trasferimento di un byte su indirizzo dispari

14 **Quale tra i seguenti indirizzi rappresenta l'indirizzo fisico di C03A:103B?**

- ○ 1C3EA
- ○ C13DA
- ○ C13DB
- ○ 1C3EB

15 **Quando viene generato un ciclo IDLE? (2 risposte)**

- ○ quando la memoria non è pronta a inviare un dato
- ○ quando la coda interna delle istruzioni è piena
- ○ quando la memoria non è pronta a ricevere un dato
- ○ quando la CPU non necessita di nuovi dati

16 **Quando viene generato un ciclo WAIT? (2 risposte)**

- ○ quando la memoria non è pronta ad inviare un dato
- ○ quando la coda interna delle istruzioni è piena
- ○ quando la memoria non è pronta a ricevere un dato
- ○ quando la CPU non necessita di nuovi dati

Test vero/falso

1	Dal punto di vista strutturale il microprocessore è formato da due blocchi: ALU e registri.	V F
2	La dimensione delle celle di memoria si esprime in word.	V F
3	Tutti i segnali collegati in ingresso all'ALU vengono sempre bufferizzati.	V F
4	I registri general purpose non possono essere accoppiati per raggiungere dimensioni maggiori.	V F
5	L'UC riceve direttamente dal registro IR il codice operativo traducendolo in comandi.	V F
6	I registri general purpose appartengono alla EU.	V F
7	La EU è la parte della CPU che elabora.	V F
8	La BIU codifica le istruzioni.	V F

Esercizi di completamento

1 Il processore è un circuito ..... dotato di una struttura circuitale in grado di attuare un prefissato ..... chiamato .....

2 Tutti i segnali collegati in ingresso all'ALU sono ..... mentre i segnali di uscita confluiscono in un registro chiamato .....

3 Un segnale bufferizzato è ..... fino alla variazione successiva del ..... Ad ogni variazione del ..... viene memorizzato un valore di ingresso che rimane ..... in uscita fino alla variazione successiva del .....

4 L'UC definisce la cadenza con cui si susseguono in modo ..... le varie operazioni del sistema.

5 Il segnale di clock è utile alla ..... del sistema.

6 Nella memoria sono memorizzati i seguenti valori che vengono inseriti in un registro a 4 byte.

0100h	1A
0101h	2F
0102h	2B
0103h	7A

Scrivi cosa contiene il registro se i dati sono stati trasferiti mediante tecnica little endian?

UNITÀ DIDATTICA 2

IL MODELLO X86

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere il modello di programmazione x86 a 16 e 32 bit
- a definire i registri a 8, 16 e 32 bit
- a riconoscere la struttura dello stack

L'architettura x86

Esistono diversi modelli di architettura, con i relativi set di istruzioni. Tali modelli vengono anche chiamati ◀ modelli di programmazione ▶ e il modello di programmazione del processore 8086 è chiamato x86 o PC compatibile.

◀ Modello di programmazione Il modello di programmazione di una CPU definisce quella parte dell'architettura del microprocessore accessibile al programmatore, che è rappresentata dai registri interni general purpose e dal repertorio delle istruzioni. ▶

Tale modello è coerente con altri modelli di programmazione come per esempio quello di linguaggi più evoluti (C++, Pascal) oppure di linguaggi specifici per lo sviluppo di codice per il microcontrollore PIC 80186.

I registri x86

Il modello di programmazione x86 può cambiare a seconda del processore sul quale viene implementato, mentre in generale può essere a 16, 32 bit o 64 bit. I registri che illustriamo di seguito sono riferiti a un processore della famiglia ◀ 80386 ▶ per un sistema operativo a 32 bit.

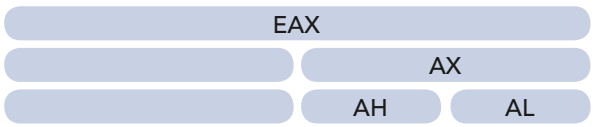
◀ 80386 Il modello 80386, o modello x86 a 32 bit, non verrà utilizzato in questo modulo in quanto programmeremo con l'assembly per x86 a 16 bit; tuttavia è importante conoscere la struttura di tale ISA per eventuali approfondimenti. I registri che possono essere utilizzati nella programmazione a 16 bit possiedono una dimensione massima di 16 bit: sono pertanto esclusi tutti i registri che iniziano con la denominazione E (Extended). ▶

Il modello di programmazione si compone fondamentalmente degli elementi descritti di seguito.

- Registri interni
  L'insieme dei registri per uso generale (general purpose) e dei registri specializzati forma l'ambiente di esecuzione delle istruzioni.
  Le istruzioni, come vedremo in seguito, possono eseguire trasferimenti, operazioni aritmetico/logiche e controllo di flusso del programma.
- Spazio di indirizzamento della memoria
  A ogni programma in esecuzione viene associato uno spazio di memoria che può arrivare a un massimo di 4 GB di memoria avendo a disposizione un bus indirizzi gestito a 32 bit (2^{32}).
- Spazio di indirizzamento degli I/O
  L'accesso ai dispositivi di I/O è un'operazione privilegiata che tratteremo successivamente.

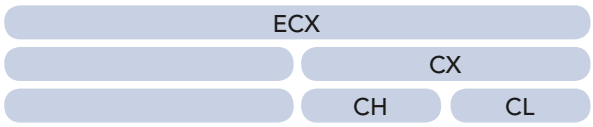
Vediamo i principali registri del modello x86 a 32 bit:

Registro **Accumulator**

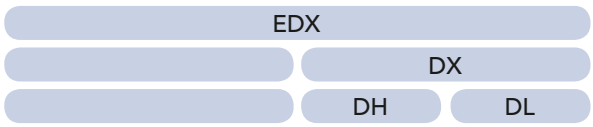


Registro **Base**

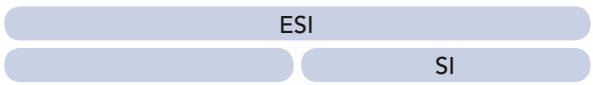
Registro **Counter**



Registro **Data**



Registro **Source Index**



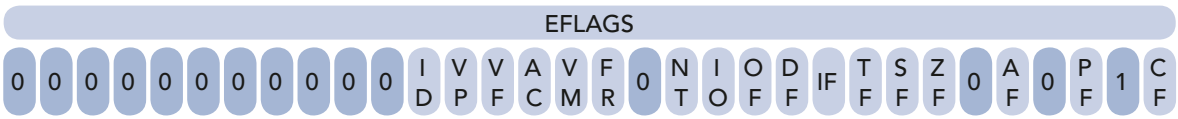
Registro **Destination Index**

Registro **Base Pointer**

Registro **Stack Pointer**

Registro **Instruction Pointer**

Registro **dei Flag**

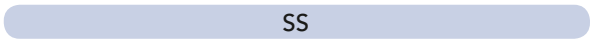


Registri di segmento

Registro **Code Segment**

Registro **Data Segment**

Registro **Stack Segment**



Registro **Extra Segment**

Registro **Extra Segment generico**

Registro **Extra Segment generico**

I registri EAX, EBX, ECX, EDX, ESI, EDI, EBP sono registri di tipo general purpose a 32 bit. Servono come sorgente o destinazione per qualunque operazione di trasferimento oppure di calcolo (aritmetico-logica).

> Ricorda che la CPU non è in grado di trasferire direttamente i dati da memoria a memoria; il dato deve sempre passare per un registro, come abbiamo visto precedentemente durante le fasi dei cicli macchina.

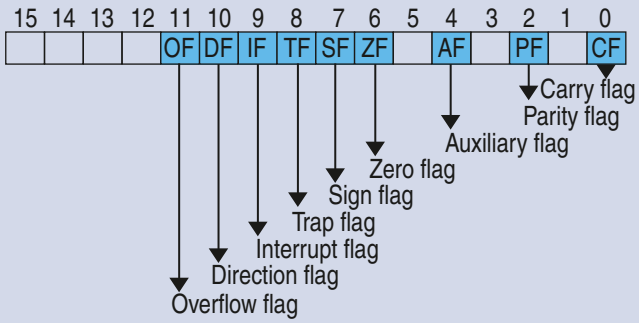
EAX significa *Extended AX*, dove AX rappresenta il registro accumulatore (Accumulator). EBX rappresenta il corrispondente a 32 bit di BX, il cui nome significa registro base. ECX rappresenta il corrispondente a 32 di CX, il cui nome significa registro contatore (Counter). EDX rappresenta il corrispondente a 32 bit del registro DX, il cui nome significa registro dei dati (Data).

ESI rappresenta il corrispondente a 32 bit di SI, il cui nome significa registro sorgente indice (Source Index).

ESI rappresenta il corrispondente a 32 bit di DI, il cui nome significa registro destinazione indice (Destination Index). EBP rappresenta il corrispondente a 32 bit di BP, il cui nome significa registro puntatore alla base (Base Pointer).

I 16 bit meno significativi di EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP, EIP, EFLAGS possono essere visti come i registri a 16 bit chiamati rispettivamente AX, BX, CX, DX, SI, DI, BP, SP, IP e ◀ FLAGS ▶.

◀ FLAGS Il registro dei flag che utilizzeremo nel volume è in realtà rappresentato dai 16 bit meno significativi di tale registro che possono essere così riassunti:



I flag possono essere suddivisi in flag di stato e flag di controllo. I flag di stato forniscono indicazioni sull'ultima istruzione eseguita:

- CARRY FLAG (CF) = 1 quando un'operazione provoca riporto; può essere una condizione per i salti.
- PARITY FLAG (PF) = 1 quando la somma dei bit a 1 presenti nel risultato di un'operazione è pari; viene usato per la trasmissione dei dati.
- AUXILIARY CARRY FLAG (AF) = 1 quando si riproduce un riporto tra il bit 3 e 4 di un operando.
- ZERO FLAG (ZF) = 1 quando il risultato di un'operazione è un valore nullo; può essere una condizione per i salti.
- SIGN FLAG (SF) ripete il valore del bit più significativo di un'operazione; come è noto 0 indica segno +, 1 indica –; può essere una condizione per i salti.
- OVERFLOW FLAG (OF) = 1 quando un'operazione aritmetica dà origine a overflow oppure quando una moltiplicazione tra due numeri ngativi dà luogo a un valore ancora negativo; può essere una condizione per i salti.

I flag di controllo forniscono indicazioni sull'ambiente di programmazione:

- TRAP FLAG (TF) usato in ambienti di debug può richiedere, se = 1, l'interruzione del programma.
- INTERRUPT FLAG (IF) se = 0, usato per disabilitare eventuali interruzioni esterne.
- DIRECTION FLAG (DF) usato nelle operazioni sulle stringhe (MOVS, LODS, STOS) per regolare l'incremento (DF=0) o il decremento (DF=1) dei registri indice. ▶

A sua volta gli 8 bit meno significativi di AX, BX, CX, DX possono essere visti come i registri a 8 bit AL, BL, CL, DL, dove la lettera "L" indica LOW (byte meno significativo). Di conseguenza gli 8 bit più significativi di AX, BX, CX, DX possono essere visti come i registri a 8 bit AH, BH, CH, DH, dove la lettera "H" indica HIGH (byte più significativo).
Il registro EIP (*Extended Instruction Pointer*) è un registro specializzato di tipo puntatore che serve alla CPU per conoscere l'indirizzo della prossima istruzione da prelevare dalla memoria durante una fase di Fetch. Si tratta della versione a 32 bit del registro IP discusso nelle unità precedenti. Il contenuto di tale registro viene implicitamente modificato dalle istruzioni di salto o di chiamata alle procedure oppure di chiamata alle interruzioni.
Il registro ESP (*Extended Stack Pointer*) è un registro specializzato di tipo puntatore che contiene l'indirizzo della cima dello stack, spazio di memoria indispensabile per la gestione delle procedure, del passaggio dei parametri e delle variabili locali.
Il registro dei flag è chiamato EFLAGS (*Extended Flags*) che contiene i segnali di flag modificati dal risultato delle ultime istruzioni aritmetico-logiche. Nella tabella vi sono alcuni flag non utilizzati riservati per usi futuri.
I registri di segmento CS, DS, SS, ES, FS, GS sono tutti a 16 bit e vengono usati nel modello di programmazione segmentata.

I registri dati general purpose

I registri AX, BX, CX, DX vengono chiamati registri dati e possono essere usati per operazioni aritmetiche e logiche sui dati.

Il registro accumulatore AX viene usato principalmente per le operazioni logiche e aritmetiche sui dati. Il suo concetto di uso è analogo al concetto di accumulatore dei linguaggi evoluti e molte operazioni vengono eseguite più velocemente sull'accumulatore.

> A volte il registro accumulatore AX si lega al registro dati DX per formare un numero a 32 bit. Per esempio durante una moltiplicazione a 16 bit il risultato viene memorizzato nella coppia di registri DX:AX, dove il registro AX rappresenta i 16 bit meno significativi e DX i 16 bit più significativi.

Il registro accumulatore viene usato principalmente nelle operazioni in cui è coinvolta un'estensione di segno (si aumenta il numero di bit di un numero senza cambiarne il segno) e nelle operazioni di I/O, in quanto è l'unico registro in grado di inviare o ricevere dati da una periferica.

Esempio di assegnazione

```
MOV AL,0105
MOV AX,'S'
```

Esempio di operazioni aritmetiche

```
INC AX
DEC AX
ADD AX,03
SUB AX,02
```

Esempio di operazioni logiche

```
CMP AX,01
ROL AL,01
AND AX,00FFh
```

Il registro base BX viene usato principalmente per memorizzare l'indirizzo di partenza (base) di una cella di memoria; a tale indirizzo viene sommato un offset in modo da ottenere un indirizzo finale per identificare una cella di memoria sulla quale leggere o scrivere. Gli altri registri base invece non possono essere usati per formare un indirizzamento base più offset.

Esempi di assegnazione

```
MOV BX,01
MOV BX,0FFFFh
MOV BX,CX
```

Esempio che copia in AL il contenuto della cella puntata da BX

```
MOV AX,[BX]
```

Esempio che copia in AH il contenuto della cella puntata da BX più offset rappresentato da SI

```
MOV AH,[BX+SI]
```

Il registro contatore CX viene usato principalmente per effettuare il conteggio all'interno dei cicli oppure per effettuare copie di blocchi di memoria, operazioni di rotazione e scorrimento o per le istruzioni di gestione dell'I/O.

Esempio di ciclo ripetuto CX volte (incrementa il contenuto del registro AL per 10 volte)

```
       MOV CX,0Ah
       MOV AL,00
ciclo: ADD AL,01
       LOOP ciclo
```

Il registro dati DX viene utilizzato in maniera simile al registro accumulatore, quindi principalmente per le operazioni aritmetico-logiche. Inoltre il registro dati viene impiegato per effettuare la lettura e la scrittura sui dispositivi di I/O.

Esempio di assegnazione

```
MOV DX,03
```

Esempio di operazioni aritmetiche

```
INC DX
DEC DX
ADD DX,03
SUB DX,02
```

Esempio di operazioni logiche

```
CMP DX,01
ROL DL,01
AND DH,0Fh
```

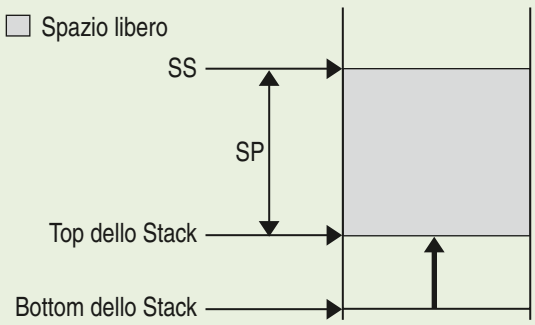
Esempio di lettura dato dalla prima porta seriale (COM1)

```
MOV DX,03F8h
IN AL,DX
```

Zoom su...

STACK

Il termine stack può essere tradotto in italiano come catasta e indica un'area di memoria collocata all'interno della memoria centrale.
I registri SS e SP definiscono rispettivamente l'indirizzo del segmento e l'offset della cima (top) dello stack.
La dimensione dello stack cresce andando dagli indirizzi di memoria più alti verso quelli più bassi. Non dobbiamo commettere l'errore di pensare che l'indirizzo di bottom dello stack sia rappresentato dal contenuto del registro SS: si tratta dell'indirizzo di segmento SS seguito dall'offset 0000h.
In pratica la base o bottom dello stack possiede l'indirizzo logico SS:0000, mentre la cima possiede l'indirizzo logico SS:SP, come illustrato dalla figura seguente:



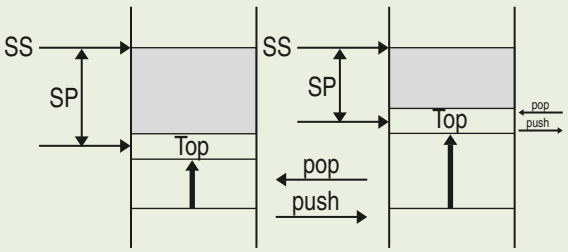
Le istruzioni assembly PUSH e POP agiscono sullo stack trasferendo 2 byte (1 word) per ciascuna operazione. In particolare l'operazione di PUSH scrive nella cima dello stack un nuovo valore mentre l'operazione di POP preleva il valore posto in cima allo stack.

In dettaglio l'operazione di PUSH esegue:

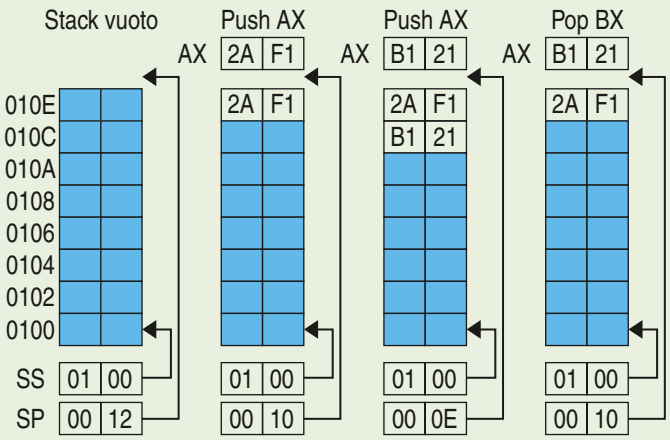
- SP = SP – 2
- Scrive una word alla posizione top dello stack.

In dettaglio l'operazione di POP esegue:

- SP = SP + 2
- Legge una word dalla posizione top dello stack.



L'esempio seguente mostra una situazione ipotetica nella quale l'indirizzo indicato è a 16 bit e coincide con il segmento per facilitare i calcoli. Ciascun elemento dello stack è formato da due celle: la cella a sinistra possiede l'indirizzo indicato mentre quella a destra possiede l'indirizzo + 1 rispetto a quella alla sua sinistra, quindi la cella più in alto possiede l'indirizzo 010F:



Come possiamo notare, nella situazione iniziale il registro SS punta alla base dello stack e tale valore rimane inalterato per tutte le operazioni di lettura e scrittura. Il registro SP punta alla cima dello stack, rappresentata dall'elemento di offset 0012 (12 in esadecimale equivale alla 18ª cella di memoria). Nella seconda situazione viene effettuata una PUSH del contenuto del registro AX: il registro SP viene decrementato di due unità (0010) per puntare alla nuova cima dello stack e il contenuto del registro AX, cioè 2AF1 viene copiato nella cella dello stack puntata da SP.

Nella terza situazione viene effettuata un'altra operazione di PUSH del registro AX (B121): il registro SP viene decrementato di due unità (000E) per puntare alla nuova cima dello stack e il contenuto di AX viene copiato nella cella di offset 000E.

Nella quarta situazione viene effettuata un'operazione di POP nel registro BX: la cella puntata da SP (000E) viene copiata nel registro BX e il contenuto del registro SP viene incrementato di due unità (0010).

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Quali tra i seguenti sono registri base? (2 risposte)**
- ○ BP
- ○ SP
- ○ BX
- ○ AX

2 **I registri AX e BX sono a:**
- ○ 8 bit
- ○ 16 bit
- ○ 32 bit
- ○ 64 bit

3 **I registri che iniziano per E sono sempre a:**
- ○ 16 bit
- ○ 32 bit
- ○ 64 bit
- ○ 8 bit

4 **I registri DI e SI sono registri di tipo: (2 risposte)**
- ○ indice
- ○ puntatore
- ○ accumulatore
- ○ segmento

5 **Il sign flag contiene:**
- ○ il segno dell'ultima operazione aritmetica, 1 se positiva, 0 se negativa
- ○ il bit meno significativo dell'ultima operazione aritmetica
- ○ il segno dell'ultima operazione aritmetica, 0 se positiva, 1 se negativa
- ○ il riporto tra i 4 bit meno significativi e più significativi

6 **Quali tra i seguenti sono registri dati (2 risposte):**
- ○ BX
- ○ DI
- ○ SI
- ○ DX
- ○ SP
- ○ DS

7 **Quali tra le seguenti sono operazioni aritmetiche: (2 risposte)**
- ○ DEC AX
- ○ MOV AX,01
- ○ ROL AL,1
- ○ SUB AX,01
- ○ AND AL,01Fh
- ○ MOV [SI],AL
- ○ IN AL,DX

8 **Associa il nome del registro posto a destra con il relativo registro della CPU posto a sinistra:**

a) CX ____ Registro accumulatore
b) BX ____ Registro dati
c) DX ____ Registro base
d) AX ____ Registro contatore

9 **A ogni istruzione POP il registro SP viene:**
- ○ decrementato di due
- ○ incrementato di due
- ○ decrementato di uno
- ○ incrementato di uno

10 **Quale istruzione delle seguenti è di tipo logico?**
- ○ OUT
- ○ ADD
- ○ MOV
- ○ AND

Test vero/falso

1 Gli 8 bit meno significativi di AX sono contenuti nel registro AL. V F
2 Il carry flag si attiva quando il risultato dell'ultima operazione aritmetica è diverso da zero. V F
3 Il flag di overflow è attivato dal traboccamento di un'operazione aritmetica con segno. V F
4 Il flag di parity è attivo quando il risultato di un'operazione aritmetica è pari. V F
5 A ogni istruzione di PUSH il registro SP viene incremenentato di due unità. V F
6 L'istruzione MOV DL,01FFh è corretta. V F
7 L'istruzione MOV AX,01FFh è corretta. V F
8 L'istruzione ADD è di tipo logico. V F

UNITÀ DIDATTICA 3

IL LINGUAGGIO ASSEMBLY E L'ASSEMBLER

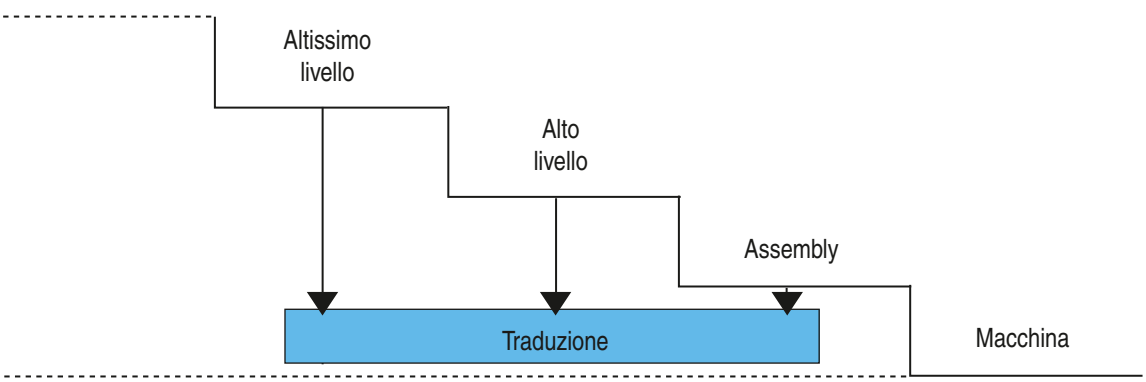
IN QUESTA UNITÀ IMPAREREMO...

- a distinguere gli elementi di un assemblatore
- a conoscere le istruzioni principali di diverse ISA

Il linguaggio assembly

◀ Il termine assembly deriva dal nome del programma traduttore in linguaggio macchina: assembler, che significa assemblatore. L'assemblatore non fa altro che assemblare il codice macchina partendo dalle istruzioni scritte in linguaggio mnemonico assembly. Con mnemonico intendiamo cha a ciascun codice operativo in linguaggio macchina è associato un codice più facilmente comprensibile. Non dobbiamo confondere i termini assembly e assembler: il primo indica il linguaggio di programmazione mentre il secondo il programma che traduce il sorgente in linguaggio macchina. ▶

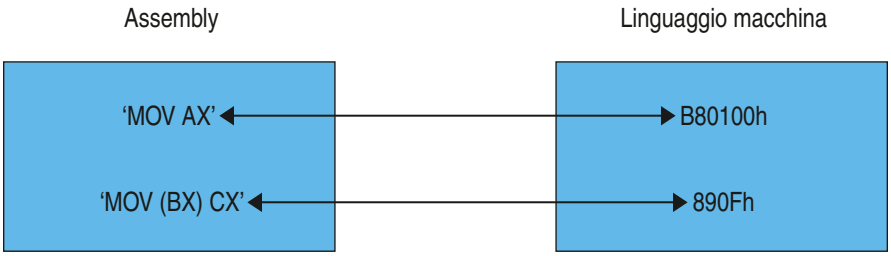
Come abbiamo visto precedentemente affinché la CPU possa compiere il proprio lavoro, cioè elaborare, è necessario che in memoria sia presente un programma che indichi al processore che cosa deve fare. Tuttavia il processore deve ricevere le istruzioni formattate in modo opportuno, secondo un preciso codice. Un programma deve contenere, per ciascun comando impartito al sistema, una precisa combinazione di cifre binarie. La figura seguente mostra la collocazione del linguaggio ◀ assembly ▶ che si trova a metà strada tra il linguaggio evoluto e il linguaggio macchina.



Per molti aspetti l'assembly è un linguaggio di programmazione come gli altri, ma quello che lo distingue è il fatto che possiede una corrispondenza uno a uno con il linguaggio macchina. Infatti per ogni codice operativo delle istruzioni di una CPU esiste una corrispondente istruzione in assembly.
La seguente tabella mostra come viene tradotta ogni istruzione dal linguaggio assembly in linguaggio macchina:

Codice mnemonico Assembly	Linguaggio macchina Esadecimale	Linguaggio macchina Binario
XOR AX,AX	33 C0	00110011 11000000
INC AX	40	01000000
ADD AL,01	41	01000001

Ciascuna istruzione assembly può essere tradotta in un'istruzione di macchina, lo stretto legame tra linguaggio assembly e la CPU appare evidente, come abbiamo visto finora parlando di ISA x86:



In commercio esistono diversi tipi di assembly che differiscono per la CPU sulla quale devono lavorare, tuttavia i concetti fondamentali della programmazione in linguaggio assembly rimangono fondanti per cimentarsi anche in ISA diverse.

Proprio per questa caratteristica, quella cioè di essere strettamente dipendente dal tipo di architettura, possiamo affermare che il linguaggio assembly non è un linguaggio portabile.

Il set di istruzioni del linguaggio assembly è una rappresentazione mnemonica dell'ISA del processore, chiamato appunto linguaggio macchina, in cui viene associato a ciascun comando in linguaggio macchina un breve codice, formato generalmente da tre lettere, chiamato codice mnemonico che ne rappresenta il significato.
Di seguito alcuni esempi:

Istruzione assembly	Comando	Significato
MOV	Move	Sposta o copia da una registro a una variabile o viceversa
ADD	Addiction	Somma il contenuto del registro o della variabile indicata
INC	Increment	Incrementa di una unità il contenuto del registro o della variabile indicata
JMP	Jump	Passa il controllo a un'istruzione non consecutiva indicata
CMP	Compare	Confronta un registro o una variabile con un dato indicato

Zoom su...

VANTAGGI E SVANTAGGI DELLA PROGRAMMAZIONE IN ASSEMBLY

Vediamo alcuni vantaggi della programmazione in linguaggio assembly:

- è il linguaggio più facile da utilizzare per creare applicazioni in diretto contatto con l'hardware (interfacciamenti); inoltre, è per molti un ottimo strumento che può essere applicato in contesti diversi, quali per esempio la programmazione di sistemi cablati o computer embedded;
- è un linguaggio molto veloce in fase di esecuzione in quanto esegue istruzioni che possono essere elaborate direttamente dalla CPU senza dover essere tradotte in un numero assai elevato di altre istruzioni, spesso inutili;
- consente al programmatore il pieno controllo della funzionalità della CPU, della memoria e dei dispositivi di I/O. I linguaggi evoluti non lasciano traccia di come vengono realizzate le istruzioni a basso livello, tale compito è demandato al compilatore;
- i programmi scritti in assembly occupano meno memoria rispetto ai linguaggi evoluti;
- consente la manipolazione dei singoli bit ricevuti o inviati dai dispositivi di I/O.

Vediamo i principali svantaggi di questo linguaggio:

- non è adatto alla realizzazione di programmi complessi che necessitano di moltissime righe di codice;
- necessita di una certa conoscenza dell'architettura di riferimento;
- è difficile da leggere, in quanto il flusso del programma consente una programmazione non strutturata e assai disordinata, essendo controllato da salti.
- il codice non è portabile su altre architetture.

Istruzioni di base assembly

Vediamo le istruzioni di base del linguaggio assembly per l'ISA x86, suddivise per tipo:

- **Istruzioni di spostamento o assegnazione:**
  – MOV, PUSH, POP, XCHG, LEA, LDS, LES;
- **Istruzioni aritmetiche:**
  – ADD, SUB, MUL, IMUL, DIV, IDIV, ADC, SBB, INC, DEC, NEG, BCD, DAA, DAS, AAA, AAS, AAM, AAD;
- **Istruzioni bitwise logiche:**
  – AND, OR, XOR, NOT;
- **Istruzioni bitwise di rotazione/traslazione:**
  – SAL, SAR, SHL, SHR, ROL, ROR, RCL, RCR;
- **Istruzione di confronto:**
  – TST, CMP;
- **Istruzioni di salto:**
  – JMP, Jx, JCXZ, CALL, RET, IRET, LOOP, INT, INTO;
- **Istruzioni di gestione delle stringhe:**
  – LODS, STOS, MOVS, CMPS, SCAS;
- **Istruzione di lettura/scrittura da e per I/O:**
  – IN, OUT;
- **Istruzioni di attesa:**
  – NOP, WAIT.

Zoom su...

LE NUOVE ISTRUZIONI ASSEMBLY

L'elenco seguente mostra alcune delle nuove istruzioni dell'ISA 80286:

PUSHA	Salva tutti i registri (AX, CX, DX, BX, SP, BP, SI, DI) sullo stack
POPA	Legge tutti i registri (DI, SI, BP, SP, BX, DX, CX, AX) salvati nello stack
ENTER	Predispone lo stack a eseguire una procedura
LEAVE	Svuota lo stack all'uscita di una procedura
BOUND	Controlla i limiti dei vettori

L'elenco seguente mostra alcune delle nuove istruzioni dell'ISA 80386 a 32 bit:

FS, GS	Nuovi registri di segmento
PUSHAD/POPAD	Salva e preleva dallo stack tutti i registri estesi
PUSHFD/POPFD	Salva e preleva dallo stack il registro dei flag esteso
SHLD/SHRD	Trasla i dati a 32 bit a sinistra o a destra
IRETD	Ritorno da un interrupt a 32 bit

L'elenco seguente mostra alcune delle nuove istruzioni dell'ISA 80486:

BSWAP	Scambia il contenuto di due registri a 16 bit
XADD	Scambia e somma
CMPXCHG	Confronta e scambia

L'elenco seguente mostra alcune delle nuove istruzioni dell'ISA 80586 (Pentium):

CMPXCHG8B	Confronta e scambia 8 byte
CPUID	Identifica la CPU

L'elenco seguente mostra alcune delle nuove istruzioni dell'ISA Pentium Pro:

FXSAVE/FXRSTOR	Salva/ripristina lo stato dei registri floating point
SYSENTER	Chiamata di sistema veloce
SYSEXIT	Ritorno da chiamata di sistema veloce

L'elenco seguente mostra alcune delle nuove istruzioni dell'ISA ◀ MMX ▶:

◀ MMX (*MultiMedia eXtension*) è un ISA formato da 57 nuove istruzioni mirate a velocizzare alcune operazioni grafiche e di comunicazione. Utilizza un approccio SIMD (*Single Instruction Multiple Data*) che permette di eseguire la stessa operazione su 2, 4 o 8 operandi interi contemporaneamente usando operandi chiamati packet byte formati da 8 operandi da 8 bit oppure packed word formati da 4 operandi da 16 bit, oppure packet double word formati da pacchetti di 2 operandi da 32 bit, oppure un unico operando a 64 bit detto quad word. Tutti gli operandi vengono elaborati in parallelo garantendo in tal modo prestazioni di velocità di elaborazione assai elevate. Possiamo per esempio caricare in un registro 8 diversi operandi a 8 bit (per le variabili di tipo char) e sommare, in un solo ciclo di clock, a tutti lo stesso valore contemporaneamente. ▶

- **Istruzioni di spostamento o assegnazione:**
  MOVD, MOVQ
- **Istruzioni aritmetiche:**
  PADD, PADDS, PADDUS, PMADD, PMULH, PMULL, PSUB, PSUBS
- **Istruzioni logiche bitwise:**
  PAND, PANDN, POR, PXOR

- **Istruzioni bitwise di rotazione/traslazione:**
  PSLL, PSRA, PSRL
- **Istruzione di confronto:**
  PCMPEQ, PCMPGT
- **Istruzione di confronto:**
  PACKSSDW, PACKSSWB, PACKUSWB, PUNPCHK, PUNPCKL

Esistono inoltre molte altre istruzioni legate alle nuove tecnologie architetturali del nuovi processori, e tra queste spicca per importanza l'ISA ◀ SSE ▶ del Pentium III.

◀ SSE (*Streaming SIMD Extension*) è un ISA composto da 70 nuove istruzioni, di cui ben 50 operano in modalità SIMD sui numeri espressi in virgola mobile a singola precisione, accelerando alcune operazioni effettuate in campo grafico di tipo tridimensionale e in campo di gestione del suono per elaborazioni audio. Sono state introdotte nuove istruzioni per gli interi (in particolare, sommatoria di differenze assolute e calcolo della media) così da velocizzare le funzioni di compensazione e stima di moto, caratteristiche della codifica MPEG. ▶

In seguito sono state sviluppate nuove ISA per processori di microarchitetture più recenti: tra esse spicca per importanza la SSE4.2 creata per processori di generazione Nehalem. L'intero set prevede 47 nuove istruzioni, divise nei seguenti sottogruppi:

- FPDP (*Floating Point Dot Product*);
- FPR (*Floating Point Round*).

Sono coinvolte nell'ottimizzazione delle scene 2D e 3D.
L'ambito di utilizzo è prevalentemente quello dei videogiochi, insieme a tutti quelli che richiedono un uso massiccio dei calcoli in virgola mobile.
Consente inoltre di eseguire operazioni a 128 bit in un solo ciclo di clock velocizzando alcune tipiche operazioni legate all'utilizzo delle istruzioni SSE.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Assembly è un linguaggio basato su istruzioni scritte dal programmatore secondo un alfabeto:**

- ○ evoluto
- ○ macchina
- ○ mnemonico
- ○ esadecimale

2 **Che differenza c'è tra assembler e assembly?**

- ○ il primo indica il linguaggio di programmazione e il secondo programma traduttore
- ○ tutti e due i termini indicano il linguaggio di programmazione
- ○ entrambi indicano il programma traduttore
- ○ il primo indica il programma traduttore e il secondo il linguaggio di programmazione

3 **Quali tra i seguenti sono vantaggi legati all'uso del linguaggio assembly? (3 risposte)**

- ○ è un linguaggio portabile su architetture diverse
- ○ è un linguaggio che genera codice eseguibile molto velocemente dalla CPU
- ○ è facile da programmare
- ○ è un linguaggio molto evoluto
- ○ è un linguaggio che genera codice eseguibile che occupa poca memoria
- ○ consente al programmatore pieno controllo della CPU
- ○ non necessita, da parte del programmatore, di conoscenze dell'architettura

4 **Quali tra i seguenti sono svantaggi legati all'uso del linguaggio assembly? (2 risposte)**

- ○ è molto lento in fase di esecuzione
- ○ non è adatto alla realizzazione di programmi complessi
- ○ è un linguaggio che genera codice eseguibile che occupa molta memoria
- ○ necessita da parte del programmatore una conoscenza dell'architettura
- ○ non consente il pieno controllo delle funzionalità della CPU

5 **Quali tra le seguenti sono istruzioni di assegnazione? (3 risposte)**

- ○ SAL
- ○ ADD
- ○ MUL
- ○ MOV
- ○ CMP
- ○ LEA
- ○ TEST
- ○ LOOP
- ○ POP

6 **Quali tra le seguenti sono istruzioni di salto? (3 risposte)**

- ○ Jx
- ○ MOV
- ○ MUL
- ○ SUB
- ○ SAR
- ○ CALL
- ○ OR
- ○ AND
- ○ JMP

7 **Quali tra le seguenti sono istruzioni di rotazione/traslazione? (2 risposte)**

- ○ ADD
- ○ SUB
- ○ SHL
- ○ SUB
- ○ SAR
- ○ CMP
- ○ AND
- ○ ROR
- ○ JMP

8 **Che cosa effettua l'istruzione POPA (ISA 80286)?**

- ○ scrive in cima allo stack
- ○ estrae dallo stack
- ○ salva tutti i registri nello stack
- ○ estrae tutti i registri dallo stack

9 **Che cosa effettua l'istruzione BSWAP (ISA 80486)?**

- ○ scambia il contenuto di due registri a 16 bit
- ○ scambia il contenuto di due registri a 32 bit
- ○ salva il registro BX nello stack
- ○ riavvia il programma dalla cella 0100h

Test vero/falso

1 Assembly e assembler sono la stessa cosa, cioè un linguaggio di programmazione. V F

2 A ogni codice operativo delle istruzioni CPU esiste una corrispondente istruzione in assembly. V F

3 Ogni istruzione assembly può essere espressa mediante un codice sadecimale o binario. V F

4 L'MMX è un ISA che opera con istruzioni di tipo. V F

UNITÀ DIDATTICA 4

LA STRUTTURA DI UN PROGRAMMA ASSEMBLY

IN QUESTA UNITÀ IMPAREREMO...

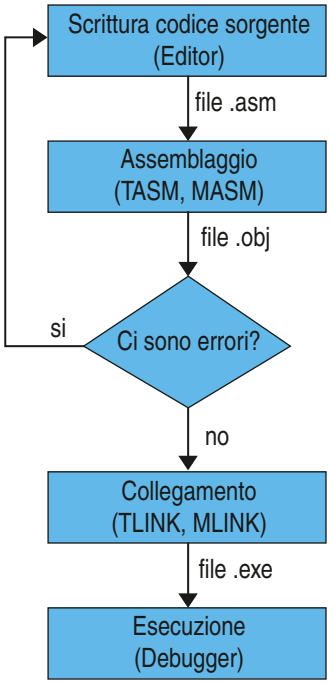
- a conoscere la struttura di un programma
- a distinguere le fasi dell'assemblaggio
- a conoscere le direttive e il formato istruzioni
- a conoscere come rappresentare i metodi di indirizzamento

L'assemblaggio di un programma

Per ottenere un programma eseguibile partendo da un programma sorgente in assembly dobbiamo effettuare due operazioni:

- assemblaggio mediante il programma assemblatore (assembler);
- collegamento mediante il programma collegatore (linker).

La scrittura del codice sorgente può avvenire con un qualunque editor di tipo ASCII, l'importante è che venga assegnata l'estensione corretta al file, per esempio .asm per i file assembly secondo la sintassi TASM. Esistono in commercio numerosi assemblatori, uno dei più diffusi è il TASM (*Turbo Assembler*) di Borland che funziona con processori fino a 32 bit (vedi il laboratorio alla fine di questo modulo per l'installazione). Una volta che abbiamo scritto il file sorgente deve essere effettuato l'assemblaggio, e se non vengono segnalati errori significa che possiamo passare all'operazione successiva, cioè il collegamento, altrimenti dobbiamo apportare le modifiche al programma sorgente mediante l'editor. Una volta assemblato il programma il file che viene generato assume l'estensione .obj.



Dopo aver collegato il file (TLINK), otteniamo un file con estensione .exe che rappresenta il programma eseguibile. ▶
L'assemblatore effettua due passi fondamentali:

1. Analisi
   In questa fase effettua il riconoscimento delle macro, delle direttive e della correttezza delle istruzioni ammesse, quindi realizza

la creazione della tabella dei simboli (*Symbol Table*) associando l'indirizzo fisico alle etichette e alle variabili.

2 Sintesi

In questa seconda fase l'assemblatore effettua la traduzione del programma utilizzando la tabella contenente i codici operativi, e sostituisce gli operandi simbolici con i corrispondenti ◀ indirizzi rilocabili ▶, segnalando gli eventuali identificatori non definiti e generando, di conseguenza, il listato e le istruzioni speciali necessarie per il linker, rappresentate da simboli pubblici o esterni.

◀ Indirizzi rilocabili È un indirizzo che una cella di memoria avrebbe se il programma iniziasse in memoria dalla posizione assoluta 0 (zero). Per passare dall'indirizzo rilocabile a quello assoluto basta quindi sommare al primo l'indirizzo di impianto del programma:

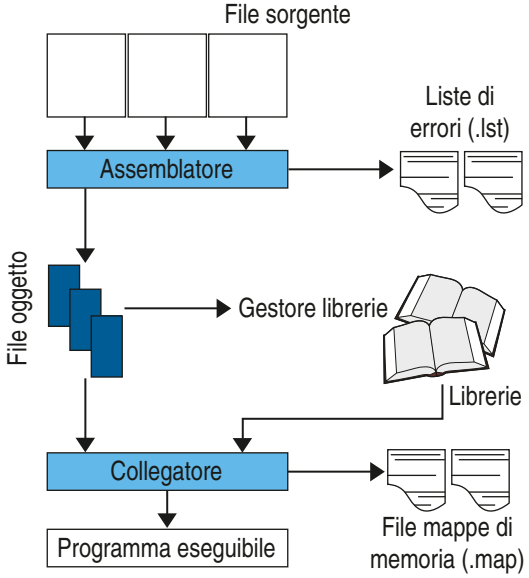
indirizzo assoluto = indirizzo rilocabile + indirizzo di impianto

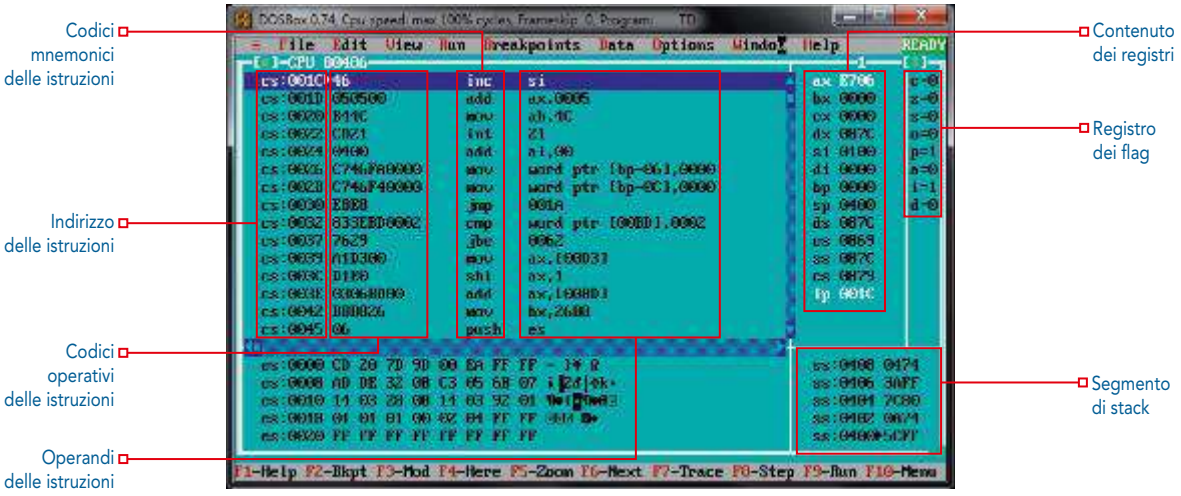
Un programma si dice rilocabile quando può essere eseguito in un punto qualsiasi della memoria, in quanto formato da indirizzi rilocabili. ▶

I file oggetto generati dall'assemblatore contengono le informazioni necessarie al collegamento con altri moduli come per esempio le librerie di sistema e le informazioni necessarie alla rilocazione con le istruzioni macchina e i dati generati dalla traduzione del programma sorgente.

La fase di collegamento (linker) genera un file eseguibile unico collegando più file oggetto tra loro, il quale contiene il codice e i dati rilocabili nonché le direttive per il loader, il programma caricatore che verrà usato dal sistema operativo per collocare in memoria il file eseguibile durante l'esecuzione dello stesso. ▶

La fase di esecuzione può essere eseguita anche mediante il debugger, un programma che consente di "spulciare" tutte le istruzioni per verificarne il funzionamento, come si vede nella finestra seguente che mostra l'esecuzione del Turbo Debugger di Borland: ▼





Zoom su...

SVILUPPO DI UN PROGRAMMA

In generale possiamo suddividere tutte le fasi che portano allo sviluppo di un programma fino alla sua esecuzione:

- **Compilatore**
  Traduce un file sorgente scritto in linguaggio ad alto livello (per esempio Visual Basic, C, Pascal) in codice rilocabile ma non eseguibile, chiamato modulo oggetto.
- **Assemblatore**
  Traduce un file sorgente scritto in linguaggio assembly in codice rilocabile ma non eseguibile chiamato modulo oggetto.
- **Linker** (Collegatore)
  Collega tra loro i vari moduli oggetto e crea un modulo eseguibile.
- **Loader** (Caricatore)
  Carica in memoria il modulo eseguibile.
- **Debugger**
  Controlla l'esecuzione del programma in fase di messa a punto.

Struttura di un programma assembly

Un programma, secondo la sintassi assembly standard, deve dichiarare innanzi tutto come deve essere utilizzata la memoria e come viene suddivisa nei quattro segmenti fondamentali (dati, stack, codice, extra). I segmenti possono essere dichiarati secondo due diverse direttive, chiamate:

- direttive standard;
- direttive semplificate.

Le direttive standard

Le istruzioni comprese tra le direttive SEGMENT ed ENDS vengono associate al segmento indicato.

```
primo SEGMENT
          ...
          dichiarazione di dati
          ...
primo ENDS                        ;termina il segmento chiamato primo
secondo SEGMENT
          ...
          istruzioni
          ...
secondo ENDS                      ;termina il segmento chiamato secondo
terzo SEGMENT
          ...
          Dati o istruzioni
          ...
terzo ENDS                        ;termina il segmento chiamato terzo
```

In questo esempio sono stati associati tre segmenti chiamati rispettivamente primo, secondo e terzo, tuttavia i nomi possono essere diversi.

La direttiva ASSUME comunica all'assemblatore a quale segmento punta ogni registro di segmento secondo la seguente sintassi.

Il codice di esempio seguente mostra come associare al segmento primo il segmento dei dati rappresentato dal registro DS, come associare al segmento secondo il segmento che conterrà le istruzioni del codice rappresentato dal registro CS, e infine come associare al segmento terzo il segmento dello stack rappresentato dal registro SS.

```
ASSUME DS:primo, CS:secondo, SS:terzo
```

Si tratta di una direttiva che non fa eseguire nessuna istruzione ma è necessaria per il calcolo degli offset degli indirizzi in memoria a cui fa riferimento ciascun segmento.

Un programma assembly deve possedere almeno una direttiva per il segmento CS, altrimenti non potrebbe contenere il codice eseguibile.

Le direttive semplificate

L'assembler TASM ha introdotto direttive di segmentazione semplificate che consentono al programmatore di scrivere il codice risparmiando alcune righe di direttive. Si tratta di un formalismo che consente una maggior concisione nella stesura dei programmi; le direttive semplificate si scrivono facendole iniziare con il punto (.). Il codice seguente mostra la sintassi e la struttura tipica di un programma con queste direttive:

```
TITLE programma
.MODEL SMALL
.STACK
;qui vanno le istruzioni o i dati del segmento puntato dal registro SS
.DATA
;qui vanno le istruzioni o i dati del segmento dei dati puntato dal registro DS
.CODE
.STARTUP      ;direttiva che determina i valori per DS e ES
;qui vanno le istruzioni del segmento del codice puntato dal registro CS


;servizio DOS di chiusura programma
MOV AH,4Ch
INT 21H
;qui vanno definite le procedure

END
```

La chiusura del programma avviene mediante un servizio del sistema operativo DOS che chiude effettivamente l'esecuzione liberando la memoria in uso dal programma:

```
MOV AH,4Ch
INT 21h
```

Tale servizio deve essere usato anche nei sistemi operativi Windows.

La direttiva .MODEL identifica le dimensioni dei segmenti in uso. In questo caso abbiamo indicato SMALL ma ne esistono anche altri:

- TINY
  Il codice del programma consente salti di tipo NEAR, cioè effettuabili solo all'interno del segmento di codice. Inoltre, poiché i segmenti dati e codice coincidono, viene usato per creare file eseguibili con estensione .com, cioè non rilocabili.
- SMALL
  Il codice del programma consente salti di tipo NEAR, cioè effettuabili solo all'interno del segmento di codice. I segmenti dati e codice sono diversi, cioè in sostanza i registri CS e DS puntano a indirizzi diversi.
- MEDIUM
  Il codice del programma consente salti di tipo FAR, cioè effettuabili anche al di fuori del segmento di codice. I segmenti dati e codice sono diversi, cioè in sostanza i registri CS e DS puntano a indirizzi diversi.
- COMPACT
  Il codice del programma consente salti di tipo NEAR, cioè effettuabili solo all'interno del segmento di codice, mentre il segmento dati consente salti di tipo FAR, cioè effettuabili anche al di fuori del segmento. I segmenti dati e codice sono diversi, cioè in sostanza i registri CS e DS puntano a indirizzi diversi.
- LARGE
  Il codice del programma consente salti di tipo NEAR, cioè effettuabili solo all'interno del segmento di codice, mentre il segmento dati consente salti di tipo FAR, cioè effettuabili anche al di fuori del segmento. I segmenti dati e codice sono diversi, cioè in sostanza i registri CS e DS puntano a indirizzi diversi. Gli array possono avere una dimensione inferiore a 64 k.
- HUGE
  Il codice del programma e il segmento dati consentono salti di tipo FAR, cioè effettuabili anche al di fuori del segmento di codice. I segmenti dati e codice sono diversi, cioè in sostanza i registri CS e DS puntano a indirizzi diversi. Gli array possono avere una dimensione superiore a 64 k.

In sintesi: ▶

Modello di memoria	Codice	Dati	Dati e codice combinati
TINY	NEAR	NEAR	SÌ
SMALL	NEAR	NEAR	NO
MEDIUM	FAR	NEAR	NO
COMPACT	NEAR	FAR	NO
LARGE o HUGE	FAR	FAR	NO

Formato delle istruzioni

Ciascuna istruzione assembly deve essere scritta cercando di rispettare il seguente incolonnamento che prevede **4 colonne**: una per le etichette (label) che devono terminare con i due punti (:), una per la parola chiave dell'istruzione o per la direttiva, una per l'operando o gli operandi separati dalla virgola, e infine una per i commenti che devono iniziare con il punto e virgola (;).

```
[Etichetta]    Istruzione | Direttiva    [Operando, operando]    [; Commento]
```

Etichetta
Chiamata anche *label*, consente di dare un nome simbolico alle istruzioni, da utilizzare come operando nelle istruzioni di salto o nelle procedure. Termina con i due punti (per esempio ciclo:) e può essere scritta indifferentemente in minuscolo o maiuscolo. È peraltro diffuso lo stile che prevede le etichette scritte interamente in minuscolo.

Istruzione o direttiva
Rappresenta la parola chiave dell'istruzione da eseguire (ADD, MOV ecc.) e può essere scritta indifferentemente in minuscolo o maiuscolo. È peraltro diffuso lo stile che prevede le parole chiave scritte interamente in maiuscolo. Le direttive sono istruzioni particolari che dialogano con l'assemblatore e iniziano con il carattere punto (per esempio .CODE).

Operandi
Sono una combinazione di tre tipi di dati:
- registro;
- memoria;
- ◀ dato immediato ▶.

Per registro si intende un registro della CPU (per esempio AX, CL, BL ecc.). Per memoria si intende una cella di memoria identificata da un registro puntatore scritto tra parentesi quadre (per esempio [SI]) oppure una variabile espressa con un ◀ nome simbolico ▶.
Per dato immediato si intende un valore numerico costante (per esempio 0FFh) oppure di tipo carattere compreso tra gli apici (per esempio 'a'). Quando un'istruzione ammette due operandi il primo rappresenta sempre la destinazione mentre il secondo la sorgente, per esempio:

◀ Dato immediato Sono costanti numeriche rappresentate di default in base 10. Possiamo anche esprimere le costanti numeriche in basi diverse:
in base 16 (esadecimale) mediante il suffisso h alla fine del numero (per esempio 0F1h);
in base 8 (octale) mediante il suffisso o alla fine del numero (per esempio 023o);
in base 2 (binario) mediante il suffisso b alla fine del numero (per esempio 01010100b);
in base 10 (decimale di default) mediante il suffisso d alla fine del numero (per esempio 123d). ▶

◀ Nome simbolico I nomi simbolici ammessi nel linguaggio sono:
- caratteri: A-Z a-z 0-9 $? @;
- il primo carattere di un nome non può essere un numero (0-9);
- ogni nome deve essere univoco;
- non deve mai coincidere con una parola riservata, come per esempio il nome di un registro o di un operatore. ▶

```
MOV CX,AX
```

copia il contenuto del registro AX nel registro CX.

Commenti
I commenti servono per rendere più leggibile il programma; tra l'altro essendo il linguaggio assembly un linguaggio poco leggibile, dobbiamo cercare di spiegare il significato dei blocchi di istruzioni.

Metodi di indirizzamento

Prima di tutto è necessario introdurre i ◀ metodi di indirizzamento ▶ offerti dal linguaggio assembly.

◀ Metodi di indirizzamento Si chiamano metodi o modalità di indirizzamento i sistemi usati nelle istruzioni per accedere agli operandi. Gli operandi vengono recuperati dall'UC (unità di controllo) in base al loro indirizzo che può essere specificato in vari modi, detti appunto modalità o metodi di indirizzamento. In generale i metodi di indirizzamento per l'ISA x86 a 16 bit possono essere facilmente ricondotti allo schema seguente:

$$\begin{Bmatrix} CS: \\ DS: \\ SS: \\ ES: \end{Bmatrix} \left[\begin{Bmatrix} BX \\ BP \end{Bmatrix} + \begin{Bmatrix} SI \\ DI \end{Bmatrix} + [spiazzamento] \right]$$

▶

Le istruzioni utilizzabili nell'ISA x86 sono raggruppabili in tre categorie:
- istruzioni di trasferimento;
- istruzioni aritmetico-logiche;
- istruzioni di controllo di flusso.

Le prime vengono usate per trasferire dati tra i registri interni, la memoria centrale e le porte di I/O, le seconde servono per manipolare i dati attraverso calcoli aritmetici e logici, detti anche ◀ bitwise ▶, eseguiti dall'ALU che può accedere agli operandi presenti nei registri interni o nella memoria centrale ma non nell'I/O, che è accessibile solo con i trasferimenti. Le terze consentono di variare il flusso sequenziale del programma tramite salti a indirizzi specifici, condizionati o non condizionati, realizzando in tal modo i costrutti della programmazione strutturata (selezione, iterazione).

◀ Bitwise Bitwise significa letteralmente "orientati ai bit": si tratta di operatori che consentono di manipolare direttamente i bit di un numero. Tra di essi distinguiamo quelli di scorrimento (shift) che consentono di far scorrere i bit, quelli di rotazione (rotate) che consentono di ruotare i bit di un numero binario di un certo numero di posizioni, e quelli booleani, come AND, OR, NOT e XOR. ▶

Le operazioni svolte dalle istruzioni possono essere unarie o binarie: un'operazione unaria è quella che ha un unico operando; un tipico esempio di operazione unaria è il cambio di segno o negazione oppure la complementazione o NOT in cui l'operazione avviene su un unico operatore. Un'operazione binaria consente per esempio di trasferire dati indicando espressamente due elementi (sorgente e destinazione).

Ciascuna istruzione deve specificare alla CPU il metodo di accesso con cui fare le operazioni, cioè il modo in cui può ottenere il dato (nel caso unario) o i dati (nel caso binario). Il metodo di accesso è identificato dal codice operativo dell'istruzione: esistono molte istruzioni simili ma con codice operativo diverso a seconda del metodo di indirizzamento e del registro sul quale operano. Di seguito vengono illustrati i principali metodi di indirizzamento.

Indirizzamento immediato

In questa modalità di indirizzamento l'operando dell'istruzione è contenuto nel codice operativo dell'istruzione stessa ed è quindi una costante. Il dato viene ottenuto tramite letture successive al Fetch del codice operativo. Questo metodo di indirizzamento non può evidentemente essere usato per gli operandi destinazione che sono oggetto di modifica.

Indirizzamento a registro

In questa modalità di indirizzamento l'operando dell'istruzione è contenuto all'interno di un registro. La CPU non deve compiere alcuna operazione attraverso il bus in quanto l'operando è già disponibile internamente nel registro indicato. Quest'ultimo è una variabile e quindi può essere sia sorgente (operando) che destinazione (risultato) di un'operazione.

Indirizzamento diretto

In questa modalità di indirizzamento l'operando dell'istruzione è contenuto a partire dalla cella di memoria il cui indirizzo è contenuto nel codice dell'istruzione stessa. Il dato è quindi una variabile di posizione costante. Esso viene ottenuto tramite letture successive al Fetch del codice operativo. Questo metodo di accesso può essere usato sia per la sorgente (operando) che per la destinazione (risultato) di un'operazione in quanto ciò che è costante è il suo indirizzo e non il suo valore, che può essere letto o scritto in base alla funzione svolta dall'istruzione.

Indirizzamento indiretto

In questa modalità di indirizzamento l'operando dell'istruzione è contenuto a partire dalla cella di memoria il cui indirizzo è contenuto in un registro puntatore. Il dato è quindi una variabile e la sua posizione è variabile proprio perché può variare il contenuto del registro puntatore. La CPU non deve compiere alcuna operazione attraverso il bus in quanto l'operando è già disponibile internamente nel registro indicato. Questo metodo di accesso può essere usato sia per la sorgente (operando) che per la destinazione (risultato) di un'operazione facendo letture o scritture all'indirizzo specificato dal registro puntatore. Il vantaggio di questo tipo di accesso rispetto al precedente è che l'indirizzo del dato non è fissato a priori e può cambiare durante l'esecuzione.

Indirizzamento indicizzato

In questa modalità di indirizzamento l'operando dell'istruzione è contenuto a partire dalla cella di memoria il cui indirizzo è calcolato dalla somma di due registri in cui il primo è considerato puntatore alla base di un blocco di memoria mentre il secondo è considerato spiazzamento, cioè spostamento all'interno del blocco di memoria. Il calcolo di questa somma tra registri è effettuato internamente, pertanto il dato è una variabile e la sua posizione è doppiamente variabile perché può variare sia il contenuto del registro puntatore sia quello dell'indice. La CPU non deve compiere alcuna operazione attraverso il bus in quanto l'operando è già disponibile internamente nel registro indicato. Questo metodo di accesso può essere usato sia per la sorgente (operando) che per la destinazione (risultato) di un'operazione facendo letture o scritture all'indirizzo specificato dal registro puntatore. Questo metodo di accesso viene tipicamente usato per accedere agli array.

Indirizzamento implicito

In questa modalità di indirizzamento l'operando non viene specificato in alcun modo in quanto implicitamente dichiarato nel codice operativo dell'istruzione stessa.

Alcuni esempi

Vediamo adesso alcuni esempi di istruzioni assembly che coprono l'intera gamma dei metodi di indirizzamento con indicati a fianco il metodo di indirizzamento utilizzato, il codice operativo espresso in base esadecimale e la descrizione del comportamento in base ai metodi di indirizzamento utilizzati. Nella prima colonna si trova il nome simbolico dell'istruzione, nella seconda il metodo di indirizzamento del primo operando, nella terza il metodo di indirizzamento del secondo operando se l'operazione è binaria, mentre nella quarta colonna si trovano le spiegazioni.

Istruzioni di trasferimento

Istruzione	Indirizzamento primo operando	Indirizzamento secondo operando	Spiegazione
`MOV BX,AX`	A registro	A registro	Copia il contenuto del registro AX nel registro BX
`MOV AX,05`	A registro	Immediato	Copia il dato immediato (numero 5) nel registro AX
`MOV BX,variabile1`	A registro	Diretto	Il contenuto della variabile è copiato nel registro BX
`MOV variabile1,AX`	Diretto	A registro	Il contenuto del registro BX è copiato nella variabile
`MOV AX,[BX]`	A registro	Indiretto	Il contenuto delle due celle puntate dal registro BX viene copiato nel registro AX

Istruzione	Indirizzamento primo operando	Indirizzamento secondo operando	Spiegazione
`MOV AL,[BX]`	A registro	Indiretto	Il contenuto della cella puntata dal registro BX viene copiato nel registro AL
`MOV [BX],AX`	Indiretto	A registro	Il contenuto del registro AX viene copiato nelle due celle puntate dal registro BX
`MOV [BX],AL`	Indiretto	A registro	Il contenuto del registro AH viene copiato nella cella puntata dal registro BX
`MOV AL,[BX+SI]`	A registro	Indicizzato	Il contenuto della cella puntata dalla somma dei registri BX e SI viene copiato in AL
`MOV AX,[BX+SI]`	A registro	Indicizzato	Il contenuto delle due celle puntate dalla somma dei registri BX e SI viene copiato nel registro AX
`MOV [BX+SI],CH`	Indicizzato	A registro	Il contenuto del registro CH viene copiato nella cella puntata dalla somma dei registri BX e SI
`MOV [BX+SI],CX`	Indicizzato	A registro	Il contenuto del registro CX viene copiato nelle due celle puntate dalla somma dei registri BX e SI
`MOV ES:[SI],AL`	Indicizzato	A registro	Il contenuto del registro AL viene copiato nell'indirizzo assoluto rappresentato dalla coppia di registri ES e SI che insieme formano un indirizzo a 32 bit dove la parte più significativa di tale indirizzo è contenuta nel registro SI
`MOV ES:[SI],CX`	Indicizzato	A registro	Il contenuto del registro CX viene copiato nelle due celle di indirizzo assoluto rappresentato dalla coppia di registri ES e SI che insieme formano un indirizzo a 32 bit dove la parte più significativa di tale indirizzo è contenuta nel registro SI
`DIV CL`	Implicito	A registro	Il dividendo è sempre AX (implicito) e il divisore è il registro CL (a 8 bit). Il quoziente viene copiato in AL e il resto nel registro AH
`DIV CX`	Implicito	A registro	Il dividendo è sempre la coppia di registri DX:AX (considerato come un numero a 32 bit) e il divisore è il registro CX (a 16 bit). Il quoziente viene copiato in AX e il resto nel registro DX
`PUSH DX`	Implicito		Copia in cima allo stack (implicito) il contenuto di DX
`POP AX`	Implicito		Copia in AX il contenuto della cima dello stack

Istruzione	Indirizzamento primo operando	Indirizzamento secondo operando	Spiegazione
`IN AL,0FEh`	A registro	Diretto	Copia il contenuto della porta di ingresso 2FEh nel registro AL
`IN AL,DX`	A registro	A registro	Copia il contenuto della porta puntata da DX (16 bit) nel registro AL
`OUT 20h,AL`	Diretto	A registro	Copia il contenuto di AL nella porta di uscita 20h
`OUT DX,AL`	Indiretto	A registro	Copia il contenuto di AL nella porta di uscita puntata da DX (16 bit)
`NEG BL`	A registro		Effettua il complemento a due del registro BL
`NEG BX`	A registro		Effettua il complemento a due del registro BX
`NEG variabile1`	Diretto		Effettua il complemento a due della variabile
`NEG [SI]`	Indiretto		Effettua il complemento a due della cella di memoria puntata dal registro SI
`JMP etichetta`	Immediato		Somma al registro IP lo spiazzamento necessario per raggiungere l'istruzione accanto all'etichetta indicata. Lo spiazzamento è un valore a 8 bit con segno compreso tra –128/+127
`JMP BX`	A registro		Copia in IP il contenuto di BX, quindi esegue l'istruzione il cui indirizzo è contenuto nel registro BX
`JMP [BX]`	Indiretto		Copia in IP il contenuto delle due celle puntate da BX, quindi esegue l'istruzione il cui indirizzo è contenuto nelle celle puntate da BX

Adesso vogliamo mostrare alcune istruzioni errate che violano le regole di sintassi oppure di indirizzamento e pertanto non possono essere in nessun caso utilizzate dal programmatore in quanto il programma traduttore (assemblatore o assembler) genererebbe un errore:

Istruzione	Indirizzamento primo operando	Indirizzamento secondo operando	Spiegazione
`MOV 09h,DX`	Immediato	A registro	Una costante non può essere destinazione
`MOV CL,CX`	A registro	A registro	I due operandi hanno dimensione diversa
`MOVvariabile1,variabile2`	Diretto	Diretto	In questo caso viene violato un principio fondamentale dell'assembly: non esiste l'indirizzamento memoria memoria, cioè si deve sempre passare per un registro

Istruzione	Indirizzamento primo operando	Indirizzamento secondo operando	Spiegazione
`MOV AL,[BL]`	A registro	Indiretto	Il registro puntatore deve essere a 16 bit
`MOV [AX],[BX]`	Indiretto	Indiretto	In questo caso viene ancora violato un principio fondamentale dell'assembly: non esiste l'indirizzamento memoria memoria, cioè si deve sempre passare per un registro
`MOV [BX+SI],[CX+DI]`	Indicizzato	Indicizzato	Anche in questo caso viene violato un principio fondamentale dell'assembly: non esiste l'indirizzamento memoria memoria, cioè si deve sempre passare per un registro
`IN AL,200h`	A registro	Diretto	Gli indirizzi di porta sono espressi a 8 bit, pertanto il valore 200h non è ammesso
`MOV AL,F1h`	A registro	Diretto	I valori esadecimali che iniziano con una lettera dell'alfabeto (tra A e F) devono essere preceduti dallo 0 (0F1h)
`OUT 20h,1`	Diretto	Immediato	L'operando non può essere una costante ma solo AL o AX
`OUT 200h,AL`	Diretto	A registro	Il metodo diretto funziona solo per gli indirizzi di porta a 8 bit
`NEG 05h`	Immediato		L'operando non può essere rappresentato da un dato immediato

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Quale estensione deve avere il file sorgente secondo l'assembler TASM?**

- ○ .ass
- ○ .asm
- ○ .bly
- ○ .tasm

2 **Quale estensione possiede un file dopo che è stato assemblato secondo l'assembler TASM?**

- ○ .asm
- ○ .obj
- ○ .exe
- ○ .dll

3 **Quando si ottiene il file eseguibile?**

- ○ dopo l'assemblaggio del file
- ○ dopo il collegamento del file
- ○ dopo il debug del file
- ○ dopo l'editing del file

4 **Come si chiamano le due fasi che esegue l'assemblatore?**

- ○ analisi e collegamento
- ○ sintesi e collegamento
- ○ analisi e debug
- ○ analisi e sintesi

5 **Metti in ordine logico crescente l'uso dei seguenti programmi che consentono di sviluppare un programma assembly e di eseguirlo:**

- ○ debug ____
- ○ linker ____
- ○ editor ____
- ○ assembler ____
- ○ loader ____

6 **Quali tra i seguenti non è un segmento tipico di un programma in esecuzione?**

- ○ stack
- ○ codice
- ○ extra
- ○ memory

7 **L'istruzione che segue:**

```
MOV AH,4Ch
INT 21h
```

- ○ termina l'esecuzione del programma liberando la memoria in uso
- ○ termina l'esecuzione del programma ma non libera la memoria in uso
- ○ indica la fine del segmento codice
- ○ indica la fine del segmento dati

8 **Associa al modello di memoria posto a sinistra la relativa descrizione posta a destra:**

a)	MODEL TINY	____ salti FAR, segmenti dati e codice diversi, array maggiori di 64k
b)	MODEL SMALL	____ salti FAR, segmenti dati e codice diversi
c)	MODEL MEDIUM	____ salti di tipo FAR nel codice NEAR nei dati, segmenti codice e dati diversi
d)	MODEL COMPACT	____ salti di tipo NEAR nel codice FAR nei dati, segmenti codice e dati diversi array fino a 64k
e)	MODEL LARGE	____ salti NEAR, segmenti codice e dati coincidono
f)	MODEL HUGE	____ salti NEAR, segmenti dati e codice diversi

9 **Per indicare un commento è necessario anteporre a ogni riga:**

- ○ il simbolo doppio slash: //
- ○ il simbolo della virgola: ,
- ○ il simbolo del punto e virgola: ;
- ○ il simbolo della parentesi e asterisco: (*

10 **Quale numero tra i seguenti è espresso in ottale?**

- ○ 0100o
- ○ 0100h
- ○ o0100
- ○ h0100
- ○ 0100oct
- ○ 0100d

Test vero/falso

1 Per ottenere un programma eseguibile è necessario collegare e quindi assemblare il programma sorgente. V F
2 L'indirizzo rilocabile è un indirizzo che una cella di memoria avrebbe se il programma iniziasse dalla cella di indirizzo 0. V F
3 Le direttive semplificate consentono di utilizzare la notazione punto. V F
4 Il segmento extra contiene gli elementi della catasta LIFO. V F
5 Il segmento dati è indirizzato dal registro DS. V F
6 Un programma assembly può anche non possedere una direttiva per il segmento CS. V F
7 La direttiva ASSUME comunica all'assemblatore a quale segmento punta ogni registro di segmento. V F
8 Le etichette incominciano sempre con i due punti. V F
9 Le istruzioni si scrivono dopo la direttiva .CODE. V F
10 Il primo carattere di un nome di variabile non può essere un numero. V F
11 Le istruzioni di controllo di flusso consentono di variare l'esecuzione del programma. V F
12 Non tutte le istruzioni specificano il metodo di accesso ai dati. V F
13 Il codice operativo identifica il metodo di accesso delle istruzioni. V F
14 Nell'indirizzamento immediato il codice operativo contiene l'operando che è quindi costante. V F

Esercizi di completamento

1 L'assemblaggio avviene mediante il programma ..... mentre il collegamento mediante il programma .....
2 L'assembler durante la fase di ..... effettua il riconoscimento delle macro, delle direttive e della correttezza delle istruzioni ammesse, quindi effettua la creazione della tabella dei ____ associando l'indirizzo fisico alle etichette ed alle variabili.
3 L'assembler durante la fase di ..... effettua la ..... del programma, utilizzando la tabella contenente i ..... e sostituisce gli .... con i corrispondenti .....
4 Per passare dall'indirizzo rilocabile a quello assoluto:
Indirizzo assoluto = ... + ...
5 Riempi le parti mancanti con le direttive corrette:
TITLE programma
.....
.....
; qui vanno le istruzioni o i dati del segmento puntato dal registro SS
.....
; qui vanno le istruzioni o i dati del segmento dei dati puntato dal registro DS
.....
.....
;direttiva che determina i valori per DS e ES
; qui vanno le istruzioni del segmento del codice puntato dal registro CS

6 Gli operandi sono una combinazione di ....., ... e .....

7 Si chiamano metodi di indirizzamento i sistemi usati nelle istruzioni per accedere agli .... che vengono recuperati dall' .... in base al loro ..... che può essere specificato in vari modi.

8 Nell'indirizzamento ..... l'operando dell'istruzione è contenuto a partire dalla cella di memoria il cui indirizzo è contenuto in un registro puntatore.

9 Nell'indirizzamento ..... l'operando dell'istruzione è contenuto a partire dalla cella di memoria il cui indirizzo è calcolato dalla somma di due registri in cui il primo è considerato puntatore alla base di un blocco di memoria ed il secondo è considerato spiazzamento cioè spostamento all'interno del blocco di memoria.

10 Nell'indirizzamento ..... l'operando non viene specificato in alcun modo in quanto implicitamente dichiarato nel codice operativo dell'istruzione stessa.

11 Completa la seguente tabella indicando il metodo di indirizzamento usato negli operandi indicati:

Istruzione	Indirizzamento primo operando	Indirizzamento secondo
MOV BL,DL		
MOV DL,01		
MOV CL,variabile1		
MOV variabile1,BL		
MOV BL,[SI]		
MOV [SI],CH		
MOV AL,[BX+SI]		
MOV CX,[BX+DI]		
MOV ES:[DI],CX		
DIV DH		
PUSH AX		
POP DX		
IN BL,01Bh		
IN AL,DX		
OUT 20h,AL		
OUT DX,AL		
NEG DX		
NEG CL		
NEG variabile1		
NEG [DI]		

UNITÀ DIDATTICA 5

LE ISTRUZIONI DI ASSEGNAZIONE ASSEMBLY

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere la sintassi delle istruzioni di trasferimento
- a conoscere il significato di stack e le istruzioni a esso collegate

La sintassi

Le istruzioni utilizzabili in assembly x86 sono moltissime (oltre 300); abbiamo deciso di illustrare le più significative spiegandone le seguenti caratteristiche:

- Nome simbolico: è una parola chiave riconosciuta dall'assemblatore che individua il tipo di istruzione ed è seguita da nessuno, uno o due operandi. In realtà a uno stesso nome simbolico corrispondono diversi codici operativi perché una stessa istruzione può essere realizzata con diversi metodi di accesso.
- Modalità di indirizzamento: la sintassi indicata per i vari metodi di indirizzamento è la seguente:
  – immediato a 8/16/32 bit;
  – registro a 8/16/32 bit;
  – memoria a 8/16/32 bit.
- Flag alterati: flag alterati dalle istruzioni.
- Esempi di utilizzo: elenco di possibili metodi di accesso e brevi sequenze in cui l'istruzione è utilizzata.

L'assegnazione con MOV

L'istruzione MOV possiede due operandi: l'operando di destra viene copiato in quello di sinistra. L'operando sorgente può essere un valore immediato, un registro o una cella di memoria a cui accedere in modo diretto, indiretto o indicizzato.
L'operando destinazione può essere un registro, una cella di memoria a cui accedere in modo diretto, indiretto o indicizzato ma non un valore immediato. I due operandi devono avere le stesse dimensioni.

Sintassi:

```
MOV destinazione,sorgente
```

Modalità di indirizzamento:
MOV memoria,registro
MOV registro,memoria
MOV registro,immediato
MOV memoria,immediato
MOV registro,registro

Nell'assegnazione di un dato immediato alla memoria è sempre necessario indicare esplicitamente la dimensione della cella di memoria indirizzata. Per fare questo usiamo le parole riservate Word Ptr e Byte Ptr, a seconda che il dato immediato venga assegnato a una word (coppia di celle) oppure a una cella singola (byte). Tale indicazione deve essere posta prima della parentesi quadra: l'esempio che segue mostra come assegnare una word e successivamente un byte a due locazioni di memoria puntate dal registro SI e DI:

```
MOV    Word Ptr[SI],05h
MOV    Byte Ptr[DI],00h
```

Il problema non si pone nel trasferimento dei dati tra memoria e registri in quanto la dimensione del registro è esplicita; pertanto se effettuiamo le seguente istruzione

```
MOV    AX,[SI]
```

le celle che vengono trasferite nel registro AX sono due, in quanto la dimensione del registro rende esplicita l'assegnazione. In pratica è come se avessimo indicato:

```
MOV    AX,Word Ptr[SI]
```

Flag alterati:
nessuno

ESEMPIO 1 *Uso dell'istruzione MOV*

```
MOV BL,AL                ;copia il contenuto del registro AL nel registro BL (8 bit)
MOV DX,CX                ;copia il contenuto del registro CX nel registro DX (16 bit)
MOV CX,0F1B1h            ;copia nel registro CX un dato immediato a 16 bit
MOV CL,05h               ;copia nel registro CL un dato immediato a 8 bit
MOV AL,[SI]              ;copia nel registro AL il contenuto della cella puntata
                         ;da SI (8 bit)
MOV AX,[SI]              ;copia nel registro AX il contenuto della cella puntata
                         ;da SI (16 bit)
MOV AL,variabile1        ;copia nel registro AL il contenuto della variabile di tipo
                         ;byte (8 bit)
MOV AX,variabile2        ;copia nel registro AX il contenuto della variabile di tipo
                         ;word (16 bit)
MOV Byte Ptr[SI],01h     ;copia nella cella puntata dal registro SI il dato
                         ;immediato (8 bit)
MOV Word Ptr[BX],0125h   ;copia nelle due celle puntate dal registro BX il dato
                         ;immediato (16 bit)
MOV [BX+SI],AL           ;copia nella cella puntata dall'indirizzo formato dalla somma
                         ;dei registri BX e SI il contenuto del registro AL (8 bit)
MOV [BX+SI],AX           ;copia nelle due celle puntate dall'indirizzo formato
;dalla somma dei registri BX e SI il contenuto del registro AL (16 bit)
;esempio di accesso indiretto a una cella (8 bit)
MOV SI,0100h
```

```
MOV AL,[SI]
;esempio di accesso indiretto a due celle (16 bit)
MOV SI,0100h
MOV AX,[SI]
;esempio di inizializzazione di una variabile di tipo word (16 bit)
MOV AX,0
MOV variabile1,AX
;esempio di inizializzazione a zero alla cella precedente a quella puntata da DI (8 bit)
MOV Byte Ptr[DI-1],0
;esempio di inizializzazione a zero alla word successiva puntata da SI (16 bit)
MOV Word Ptr[SI+2],0      ;l'incremento delle celle come Word è +2
;esempio di scambio di due variabili (var1 e var2) a 8 bit
MOV AL,var1               ;per scambiare due variabili usiamo due registri temporanei
                          ;in quanto
MOV AH,var2               ;non abbiamo a disposizione l'indirizzamento memoria
                          ;con memoria
MOV var1,AH
MOV var2,AL
```

Dobbiamo sempre tenere presente la regola generale di organizzazione di memoria Intel: little endian. In base a tale regola se le celle dalla 0100h contengono i dati a lato ▶ l'istruzione che segue

0100h	15
0101h	27

```
MOV SI,0100h
MOV BX,[SI]
```

assegna al registro BX il contenuto delle celle puntate da SI. Il registro BX contiene adesso: ▶

BH	BL
27	15

Prova adesso!

- Utilizzare l'istruzione MOV
- Comprendere i metodi di indirizzamento

CREA IL FILE esempio1.asm

1. Scrivi un numero a 8 bit nel registro BL.
2. Sposta il contenuto di BL nella cella di indirizzo 00FEh usando un indirizzamento base + spiazzamento.
3. Scrivi un numero a 16 bit nella cella 0100h e spostalo nel registro DX.
4. Leggi il numero presente nella cella 0101h e scrivilo nel registro AL.
5. Scrivi un numero a 8 bit nella cella 0200h e spostalo nella cella successiva e precedente.
6. Scrivi un numero a 16 nella cella 0200h e spostalo nella cella 0100h.

Le variabili in assembly

Le variabili sono i simboli con cui il programmatore fa riferimento ai dati. Gli attributi di una variabile sono:

- l'indirizzo;
- il tipo;
- la visibilità.

L'indirizzo possiede due componenti che sono il segmento e lo spiazzamento (offset).

Ogni variabile possiede un tipo che viene verificato a ogni istruzione che coinvolge la variabile stessa; per esempio se la variabile numero viene definita di tipo byte tramite la direttiva DB, l'istruzione seguente

```
MOV AX,numero
```

restituisce un errore durante la fase di assemblaggio in quanto lo statement viola le dimensioni degli operandi che devono essere congrui (AX è a 16 bit mentre numero è a 8 bit).

La visibilità specifica se la variabile deve essere nota solo all'interno, oppure anche all'esterno, del file in cui è definita, o se è stata definita in un altro file attraverso i seguenti valori: INTERNAL, PUBLIC, EXTERN.

Per dichiarare una variabile esistono alcuni tipi primitivi che corrispondono sostanzialmente a quelli offerti direttamente dall'architettura (byte, word, double word). I tipi predefiniti in TASM sono: Byte, Word, DWord.
La sintassi per la dichiarazione di una variabile è la seguente

```
variabile     DB | DW | DDW   valore iniziale
```

dove variabile può essere un nome qualunque lungo al massimo 8 caratteri che deve sottostare ai nomi simbolici visti nelle unità precedenti. La definizione di una variabile causa l'allocazione in memoria del numero di byte corrispondente al tipo all'interno del Data Segment e deve essere effettuata all'interno della direttiva .DATA, anche se non è obbligatorio. Il valore iniziale definisce il valore attraverso il quale la variabile viene inizializzata.

L'esempio seguente dichiara una variabile e la inizializza contestualmente a zero:

```
.DATA
numero        DB              00h
```

È possibile lasciare inalterato il valore della cella o delle celle che formano la variabile indicando come valore iniziale il punto di domanda (?):

```
numero    DB              ?
```

Il codice seguente definisce una variabile di tipo byte di nome numeri che è formata da tre elementi, di cui il terzo non ha un valore iniziale specificato, mentre i valori iniziali degli altri sono nell'ordine 5 e 10, memorizzati in indirizzi di memoria contigui crescenti:

```
.DATA
numeri        DB              5,10,?
```

In tal modo l'istruzione MOV AL, numeri+1 copia in AL il contenuto della cella posizionata all'indirizzo numeri+1, cioè 10.

L'operatore DUP permette di duplicare il valore di inizializzazione per più celle in modo da formare un array. La sintassi è la seguente:

```
.DATA
vettore   DB 100          DUP(0)
```

L'esempio dichiara un vettore di 100 celle (ciascuna formata da un byte) dove ciascun elemento viene inizializzato a zero (0).

ESEMPIO 2

```
var1        DB   0                          ;1 byte inizializzato a 0
var2        DB   1,2,3,4                    ;array di 4 byte inizializzati a 1,2,3,4
stringa     DB   'a','b','c','d'            ;stringa di 4 caratteri o array
stringa1    DB   'abcd'                     ;identico al precedente
stringa2    DB   'salve mondo',0Dh,0Ah,     ;stringa che contiene una coppia
;di caratteri alla fine per andare a capo
vettore     DW   100  DUP(?)                ;array di 100 elementi ciascuno è una
                                            ;word (2 byte)
                                            ;ogni elemento non viene inizializzato (?)
matrice     DB   4    DUP(1,3 DUP(0))       ;matrice 4 per 4
```

Zoom su...

LE COSTANTI

La direttiva EQU permette di assegnare un nome simbolico a una costante attraverso la sintassi:

```
nome      EQU     espressione
```

Il nome della costante può essere utilizzato nel codice al posto della costante stessa; in genere si usano nomi maiuscoli per differenziarle dalle variabili. L'espressione fornisce un valore costante. Per esempio:

```
CR        EQU     0dh                   ;carattere Carriage Return (Invio)
LF        EQU     0ah                   ;carattere Line Feed (Inizio riga)
titolo    DB      'salve mondo',CR,LF   ;variabile che contiene una stringa che
                                        ;va a capo
```

Lo scambio con XCHG

L'istruzione XCHG (Exchange) possiede due operandi il cui contenuto viene scambiato in un'unica operazione.
Per effettuare lo scambio viene usato un registro interno invisibile al programmatore. Gli operandi possono essere rappresentati da un registro, da una cella di memoria con indirizzamento diretto, indiretto o indicizzato ma non da un valore immediato.

Sintassi:

```
XCHG operando1,operando2
```

Anche in questo caso i due operandi devono avere le stesse dimensioni, che possono essere: Byte, Word o DWord.

Modalità di indirizzamento:
XCHG registro,registro
XCHG registro,memoria
XCHG memoria,registro

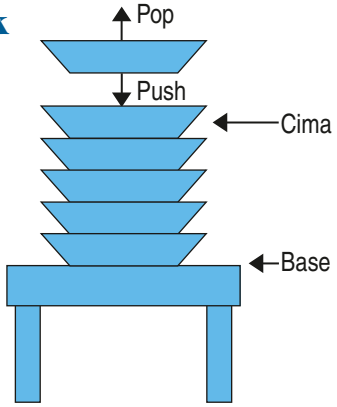
Flag alterati:
nessuno

ESEMPIO 3 ***Uso dell'istruzione XCHG***

```
XCHG AL,AH          ;scambia il contenuto dei registri AL e AH (8 bit)
XCHG AX,BX          ;scambia il contenuto dei registro AX e BX (16 bit)
XCHG AL,AH          ;scambia il contenuto della parte alta e bassa del registro AX
                    ;(8 bit)
XCHG CL,variabile1  ;scambia il contenuto del registro CL e della variabile di tipo
                    ;Byte (8 bit)
XCHG DX,variabile1  ;scambia il contenuto del registro DX e della variabile di tipo
                    ;Word (16 bit)
XCHG BL,[SI]        ;scambia il contenuto del registro BL e la cella puntata
                    ;dal registro SI (8 bit)
XCHG BX,[DI]        ;scambia il contenuto del registro BX e le due celle puntate
                    ;dal registro DI (16 bit)
XCHG AL,[BX+SI]     ;scambia il contenuto del registro AL e la cella puntata
                    ;dall'indirizzo il cui valore è pari alla somma del ;contenuto
                    ;del registro BX e SI (8 bit)
XCHG AX,[BX+SI]     ;scambia il contenuto del registro AX e le due celle ;puntate
                    ;dall'indirizzo il cui valore è pari alla somma ;del contenuto
                    ;del registro BX e SI (16 bit)
;scambia il contenuto di due registri a 16 bit
XCHG CX,DX
;equivale alle seguenti istruzioni che usano il registro AX come variabile temporanea
MOV AX,CX
MOV CX,DX
MOV DX,AX
```

Le istruzioni di trasferimento mediante stack

Lo stack, come abbiamo visto, è una struttura dati formata da una base a partire dalla quale si inserisce il primo dato e da una cima in cui si inserisce ogni volta un nuovo dato. Quando lo stack è vuoto la cima coincide con la base mentre l'estrazione di un dato può avvenire solo dalla cima. L'inserimento di un dato viene chiamato PUSH (che significa "spingere") mentre l'estrazione viene chiamata POP (da *pop out* che significa "saltare fuori").



La figura a lato illustra il principio di funzionamento: pensiamo a un insieme di piatti disposti su un tavolo, l'estrazione del piatto avviene levando l'ultimo piatto inserito, dalla cima: ▶

L'istruzione PUSH

Poiché le celle dello stack sono a 16 bit l'operando dell'istruzione PUSH deve essere sempre a 16 bit. L'istruzione PUSH copia l'operando indicato in cima allo stack e la destinazione del trasferimento è implicita ed è rappresentata dalla cella di memoria puntata dal registro SP. L'operando può essere un registro, una cella di memoria con indirizzamento diretto, indiretto o indicizzato oppure un dato immediato che viene sempre considerato a 16 bit. L'istruzione decrementa il registro SP di 2, mentre nei sistemi a 32 bit di 4.

Sintassi:

```
PUSH operando1
```

Modalità di indirizzamento:
PUSH registro
PUSH immediato
PUSH memoria

Flag alterati:
nessuno

L'istruzione POP

L'istruzione POP trasferisce le due celle presenti nella cima dello stack nella destinazione indicata come operando. La sorgente del trasferimento è implicita ed è la cella di memoria puntata da SP cioè la cima dello stack. Dopo aver copiato il contenuto di quest'ultima incrementa il registro SP di 2 o 4 (nei sistemi a 32 bit). L'operando destinazione è rappresentato da un registro, da una cella di memoria con indirizzamento diretto, indiretto o indicizzato ma non da un valore immediato.

Sintassi:

```
POP operando1
```

Modalità di indirizzamento:
POP registro
POP memoria

Flag alterati:

nessuno

ESEMPIO 4 *Uso delle istruzioni PUSH e POP*

```
PUSH AX                ;copia il contenuto del registro nello stack (16 bit)
PUSH Word Ptr[SI]      ;copia nello stack il contenuto delle 2 celle puntate
                       ;dal registro SI (16 bit)
PUSH Word Ptr[BX+DI]   ;copia nello stack il contenuto delle 2 celle puntate
                       ;dall'indirizzo dato dalla somma dei registri BX e DI (16 bit)
PUSH variabile1        ;copia nello stack il contenuto della variabile di tipo Word
                       ;(16 bit)
PUSH 05h               ;copia nello stack il dato immediato 05 espresso come Word
                       ;(16 bit)
POP AX                 ;copia la cima dello stack nel registro AX (16 bit)
POP Word Ptr[SI]       ;copia la cima dello stack nelle due celle puntate
                       ;dal registro SI (16 bit)
```

```
POP Word Ptr[BX+SI]   ;copia la cima dello stack nelle due celle puntate dall'indirizzo
                      ;dato dalla somma dei registri BX e DI (16 bit)
;salvataggio di alcuni registri nello stack
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH SI
PUSH DI
;... qui potrebbero essere alterati i contenuti dei registri salvati...
;ripristino dei registri salvati nello stack
;devono essere indicati nell'ordine invertito rispetto a quello usato da PUSH
POP DI
POP SI
POP CX
POP DX
POP BX
POP AX
```

Zoom su...

LE ISTRUZIONI DI TRASFERIMENTO CON L'I/O

L'istruzione IN copia nel registro AL, AX, EAX il contenuto della porta il cui indirizzo è specificato come operando sorgente. L'operando sorgente rappresenta l'indirizzo della porta e, in alternativa al dato immediato, deve sempre essere usato il registro DX. L'ampiezza della porta deve corrispondere alla dimensione del registro indicato, quindi rispettivamente 8, 16 o 32 bit. La versione con indirizzamento immediato è limitata alle prime 256 porte dello spazio di I/O. Per le porte da 256 a 65336 si deve usare la versione a indirizzamento indiretto che possiede una sintassi piuttosto irregolare in quanto non devono essere esplicitate le parentesi quadre attorno al registro indice.

Sintassi:

```
IN destinazione,sorgente
```

Modalità di indirizzamento:
IN AL,immediato
IN AX,immediato
IN EAX,immediato
IN AL,DX
IN AX,DX
IN EAX,DX

Flag alterati:
nessuno

L'istruzione OUT copia nella porta indicata come operando destinazione il contenuto dell'operando sorgente rappresentato dal registro AL, AX, EAX. L'operando destinazione rappresen-

ta l'indirizzo della porta e, in alternativa al dato immediato, deve sempre essere usato il registro DX. L'ampiezza della porta deve essere rispettivamente 8, 16 o 32 bit. La versione con indirizzamento immediato è limitata alle prime 256 porte dello spazio di I/O. Per le porte da 256 a 65336 si deve usare la versione a indirizzamento indiretto che possiede una sintassi piuttosto irregolare in quanto non devono essere esplicitate le parentesi quadre attorno al registro indice.

Modalità di indirizzamento:
OUT immediato,AL
OUT immediato,AX
OUT immediato,EAX
OUT DX,AL
OUT DX,AX
OUT DX,EAX

Flag alterati:
nessuno

ESEMPIO 5 *Uso delle istruzioni IN e OUT*

```
IN AL,20h       ;lettura dalla porta di indirizzo 20h nel registro AL (8 bit)
IN AX,20h       ;lettura dalla porta di indirizzo 20h nel registro AX (16 bit)
IN AL,DX        ;lettura dalla porta di indirizzo contenuto nel registro DX
                ;nel registro AL (8 bit)
IN AX,DX        ;lettura dalla porta di indirizzo contenuto nel registro DX
                ;nel registro AX (16 bit)
OUT 20h,AL      ;invio alla porta di uscita di indirizzo 20h del dato presente
                ;nel registro AL (8 bit)
OUT 20h,AX      ;invio alla porta di uscita di indirizzo 20h del dato presente
                ;nel registro AX (16 bit)
OUT DX,AL       ;invio alla porta di uscita di indirizzo contenuto in DX
                ;del dato presente nel registro AL (8 bit)
OUT DX,AX       ;invio alla porta di uscita di indirizzo contenuto in DX
                ;del dato presente nel registro AX (16 bit)
;lettura di una porta oltre l'indirizzo 0FFh
MOV DX,3F8h
IN AL,DX
;inizializzazione di una porta di uscita al valore 0
MOV AL,0
OUT 20H,AL
```

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **L'istruzione MOV operando1,operando2 consente di:**

- ○ trasferire il contenuto dell'operando1 nell'operando2, lasciando inalterato il contenuto dell'operando1
- ○ trasferire il contenuto dell'operando2 nell'operando1, lasciando inalterato il contenuto dell'operando1
- ○ trasferire il contenuto dell'operando2 nell'operando1, lasciando inalterato il contenuto dell'operando2
- ○ trasferire il contenuto dell'operando1 nell'operando2, lasciando inalterato il contenuto dell'operando2

2 **La seguente istruzione:**

```
MOV [SI],[BX]
```

- ○ trasferisce il contenuto della cella puntata dal registro BX nella cella puntata dal registro SI
- ○ trasferisce il contenuto della cella puntata dal registro SI nella cella puntata dal registro BX
- ○ è errata in quanto fa riferimento a un indirizzamento tra memoria e memoria che non è consentito
- ○ è errata in quanto il registro BX non può essere usato come registro puntatore

3 **La seguente istruzione:**

```
MOV SI,0100h
MOV Word Ptr[SI],0
```

- ○ azzera la cella 0100h
- ○ azzera il registro SI
- ○ azzera sia la cella 0100h che la cella 0101h
- ○ azzera la cella 0101h

4 **L'istruzione seguente:**

```
MOV [DI],AX
```

- ○ trasferisce il valore contenuto nel registro AX in due celle puntate da DI
- ○ trasferisce il valore contenuto nel registro AX nella cella puntata da DI
- ○ trasferisce il valore contenuto nel registro AX in due celle puntate da DI
- ○ trasferisce il valore contenuto nel registro AX in due celle puntate da DI

5 **La memoria contiene i seguenti dati:**

0200h	01h
0201h	FFh

Indica che cosa conterrà il registro CX dopo le istruzioni che seguono:

```
MOV DI,0200h
MOV CX,[DI]
```

- ○ FFh
- ○ 01h
- ○ 01FFh
- ○ FF01h

6 **Il registro DX contiene 0232h. Indica che cosa conterrà la memoria dopo le istruzioni che seguono:**

```
MOV DI,0200h
MOV [DI],DX
```

a)

0200h	02h
0201h	32h

b)

0200h	32h
0201h	02h

c)

0200h	02h
0201h	02h

d)

0200h	32h
0201h	32h

7 **La memoria contiene i seguenti dati:**

0200h	03h
0201h	15h

Indica cosa conterrà il registro AX dopo le istruzioni che seguono:

```
MOV DI,0201h
MOV AL,[DI]
MOV AH,[DI-1]
```

- ○ 0315h
- ○ 1503h
- ○ 03h
- ○ 15h

8 **Viene dichiarata una variabile numero:**

```
.DATA
numero      DW      10h
```

Quindi viene assegnata al registro BX:

```
MOV BX,numero
```

Che cosa contiene BX?

- ○ 1000h
- ○ 1010h
- ○ 10h
- ○ 0

Verifichiamo le competenze

Esprimi la tua creatività

Esegui i seguenti esercizi in assembly.

1. Scrivi un programma assembler che, dato un numero a 8 bit memorizzato nella cella 0100h lo sposti nella cella successiva usando un indirizzamento indiretto tramite registro.
2. Scrivi un programma assembler che, dato un numero a 8 bit memorizzato nella cella 0100h lo sposti nella cella successiva usando un indirizzamento diretto.
3. Scrivi un programma assembler che, dato un numero a 16 bit memorizzato nella cella 0100h lo sposti nella cella successiva usando un indirizzamento indiretto tramite registro.
4. Scrivi un programma assembler che, dati due numeri a 8 bit memorizzati a partire dalla cella 0100h li copi invertiti nelle celle successive tramite indirizzamento indicizzato.
5. Scrivi un programma assembler che, dato un numero a 8 bit memorizzato nella cella 0100h lo sposti nella cella successiva usando un indirizzamento indicizzato.
6. Scrivi un programma assembler che, dati due numeri a 8 bit memorizzati a partire dalla cella 0100h li copi nello stack e quindi nel registro CX.
7. Scrivi un programma assembler che, dato un numero a 16 bit memorizzato nella cella 0100h lo sposti nella cella successiva.
8. Scrivi un programma assembler che, dato un numero a 8 bit memorizzato nella cella 0100h lo copi nella cella successiva.
9. Scrivi un programma assembler che, dato un numero a 16 bit memorizzato nella cella 0100h lo copi nella cella successiva.
10. Scrivi un programma assembler che, dati due numeri a 8 bit memorizzati a partire dalla cella 0100h li copi invertiti nelle celle successive.
11. Scrivi un programma assembler che, dati due numeri a 16 bit memorizzati a partire dalla cella 0100h li copi invertiti nelle celle successive.
12. Scrivi un programma assembler che, dati due numeri a 8 bit memorizzati a partire dalla cella 0100h li copi nello stack e quindi nel registro BX.
13. Scrivi un programma assembler che, dati due numeri a 16 bit memorizzati a partire dalla cella 0100h li trasferisca nello stack e da quest'ultimo nei registri AX e BX, quindi in due variabili di tipo word dichiarate ad inizio programma.
14. Scrivi un programma Assembler che, dati due numeri a 16 bit memorizzati a partire dalla cella 0200h li trasferisca nei registri AX e BX, e da questi ultimi nello stack, infine in due variabili di tipo word dichiarate ad inizio programma
15. Scrivi un programma assembly che legga il contenuto di una cella e lo trasferisca nella parte alta del registro BX, quindi colloca una costante che hai dichiarato byte nella parte bassa del registro, infine trasferisci tale registro nello stack.

UNITÀ DIDATTICA 6

LE ISTRUZIONI DI SALTO

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere e utilizzare le istruzioni di salto condizionato e incondizionato
- a saper realizzare i cicli in assembly

Le istruzioni che controllano il flusso

In questa unità studieremo le istruzioni di controllo del flusso di un programma. La selezione e l'iterazione sono controllate dai salti condizionati e incondizionati che, insieme con il controllo dei flag, forniscono al programmatore un metodo per ottenere cicli e selezioni, analogamente a quanto avviene con i linguaggi evoluti. Innanzitutto però è necessario comprendere come operano le istruzioni di salto e pertanto esaminare l'istruzione CMP, che permette di confrontare due valori.

L'istruzione di confronto CMP

L'istruzione CMP, che permette di confrontare tra loro due valori espressi dai due operandi, effettua una sottrazione tra il secondo operando e il primo senza memorizzare il risultato. Il suo scopo è infatti quello di alterare i flag per utilizzarli nelle istruzioni di salto condizionato che seguono l'istruzione CMP stessa. Il primo operando può essere un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato; il secondo operando può consistere in un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato o un dato immediato. È importante sottolineare che anche in questo caso i due operandi devono avere le stesse dimensioni, che possono essere Byte, Word o DWord.

Sintassi:

```
CMP operando1,operando2
```

Sebbene tutti i flag vengano alterati, i più significativi sono ZF e CF:
ZF = 0 quando i due operandi sono diversi, cioè quando operando1 <> operando2
ZF = 1 quando i due operandi sono uguali, cioè quando operando1 = operando2
CF = 0 quando il secondo operando è minore o uguale al primo, cioè quando operando1 >= operando2.
CF = 1 quando il secondo operando è maggiore del primo, cioè quando operando1 < operando2.

Modalità di indirizzamento:
CMP registro,registro
CMP registro,memoria
CMP registro,immediato
CMP memoria,registro
CMP memoria,immediato

Flag alterati:
ZF, CF, SF, OF, PF

ESEMPIO 6 ***Uso di CMP***

```
CMP AL,BL               ;confronto tra due registri (8 bit)
                        ;se ZF = 1 e CF = 1 allora significa che AL = BL
                        ;se ZF = 1 e CF = 0 allora significa che AL = BL
                        ;se ZF = 0 e CF = 1 allora significa che AL < BL
                        ;se ZF = 0 e CF = 0 allora significa che AL > BL
CMP CX,BX               ;confronto il registro CX con BX (16 bit) identiche modalità
                        ;viste sopra (16 bit)
CMP DL,02h              ;confronto il registro DL con il dato immediato 02h (8 bit)
CMP CX,05h              ;confronto il registro CX con il dato immediato 05h (16 bit)
CMP BH,[SI]             ;confronto il registro BH con la cella puntate dal registro
                        ;SI (8 bit)
CMP BX,[SI]             ;confronto il registro BX con le due celle puntate
                        ;dal registro SI (16 bit)
CMP AL,variabile1       ;confronto il registro AL con la variabile di tipo Byte (8 bit)
CMP BX,variabile2       ;confronto il registro BX con la variabile di tipo Word (16 bit)
CMP Byte Ptr[SI],0FFh   ;confronto la cella puntata da SI con il dato immediato 0FFh
                        ;(8 bit)
CMP Word Ptr[SI],0FFFh  ;confronto le 2 celle puntate da SI con il dato immediato
                        ;0FFFh (16 bit)
CMP variabile1,BL       ;confronto la variabile di tipo Byte con il registro BL (8 bit)
CMP variabile2,CX       ;confronto la variabile di tipo Word con il registro CX (16 bit)
CMP variabile1,01h      ;confronto la variabile di tipo Byte con il dato immediato
                        ;01h (8 bit)
CMP variabile2,01h      ;confronto la variabile di tipo Word con il dato immediato
                        ;01h (16 bit)
;verifica se una variabile di tipo byte contiene 0
CMP variabile1,0
;verifica se il registro SI contiene un numero maggiore di 10
CMP SI,0Ah
;verifica se la cella di indirizzo formato dalla somma di BX e DI contiene un numero
minore di 20
CMP Byte Ptr[BX+DI],14h
;verifica se il valore contenuto nelle 2 celle puntate da DI contiene 0
CMP Word Ptr[DI],0
```

L'istruzione di salto incondizionato JMP

Le istruzioni di salto si dividono in istruzioni di salto condizionato e incondizionato. I salti incondizionati saltano all'etichetta indicata senza verificare il contenuto dei flag (potremmo dire che saltano qualunque cosa accada).
Tecnicamente un'istruzione di salto copia l'operando indicato, rappresentato da un'etichetta tradotta in un indirizzo, nel registro IP.

L'operando è l'indirizzo in cui si trova l'istruzione successiva da eseguire; pertanto l'esecuzione del programma non procederà sequenzialmente, come avviene nella maggior parte delle altre istruzioni, ma a partire dall'istruzione specificata dall'etichetta.

Le etichette o label possono essere poste in un punto qualunque del programma e devono terminare con i due punti, per esempio "ciclo:". Un'etichetta non può essere distante più di 127 byte dall'indirizzo della chiamata. Mediante un'etichetta si realizza un salto relativo, in quanto il salto riguarderà un numero di celle successive o precedenti indicate dall'etichetta. Possiamo anche effettuare un salto assoluto indicando espressamente l'indirizzo di Offset della cella da raggiungere mediante una cella di memoria oppure un registro a 16 bit.

Sintassi:

```
           JMP etichetta oppure spiazzamento
           ...
etichetta:
```

Modalità di indirizzamento:
JMP etichetta
JMP [registro]
JMP memoria

Flag alterati:
nessuno

ESEMPIO 7 ***Uso di JMP***

```
JMP ciclo          ;salto incondizionato relativo all'etichetta chiamata ciclo:
JMP SI             ;salto incondizionato assoluto all'istruzione di indirizzo indicato
                   ;dal registro SI
JMP Word Ptr[DI]   ;salto incondizionato assoluto all'istruzione di indirizzo indicato
                   ;dal registro DI
;salto incondizionato di un ciclo infinito
infinito:  NOP
           JMP infinito
```

L'istruzione di salto condizionato J

L'istruzione di salto condizionato determina una modifica nella sequenza di esecuzione del programma a seconda dell'esito di una determinata condizione: se quest'ultima è soddisfatta il controllo passa alla riga in cui è presente l'etichetta indicata come operando dell'istruzione; in caso contrario il controllo passa all'istruzione successiva. Le condizioni valutate dall'istruzione sono relative al contenuto del flag indicato; esistono infatti numerose istruzioni di salto condizionato, una per ogni flag che deve essere verificato. Quando vogliamo effettuare un salto condizionato dobbiamo necessariamente effettuare un'istruzione CMP prima del salto per ottenere una variazione dei flag in relazione allo stato degli elementi che formano la condizione. Per effettuare una condizione che verifichi, per esempio, se un registro contiene zero è necessario anteporre all'istruzione di salto condizionato un'istruzione CMP tra i due valori, in modo che il flag di zero venga settato se i due valori sono uguali:

```
CMP AL,0h        ;se AL = 0 viene settato il flag di zero
JZ salto         ;se il flag di zero è settato il controllo passa all'etichetta salto:
```

Vediamo ora la sintassi e il significato delle istruzioni di salto condizionato, tenendo presente che abbiamo effettuato preventivamente un'istruzione di confronto: CMP op1,op2.

JA ;Jump if above (salta se CF=0 e ZF=0, cioè se op1>op2) Opera su numeri senza segno
JAE ;Jump if above or equal (salta se CF=0, cioè se op1>=op2) Opera su numeri senza segno
JB ;Jump if below (salta se CF=1, cioè se op1<op2) Opera su numeri senza segno
JBE ;Jump if below or equal (salta se CF=1 or ZF=1, cioè se op1<=op2) Opera su numeri senza segno
JC ;Jump if carry (salta se CF=1)
JE ;Jump if equal (salta se ZF=1, cioè se op1=op2)
JG ;Jump if greater (salta se ZF=0 and SF=OF, cioè se op1>op2) Opera su numeri con segno
JGE ;Jump if greater or equal (salta se SF=OF, cioè se op1>=op2) Opera su numeri con segno
JL ;Jump if less (salta se SF<>OF, cioè se op1<op2) Opera su numeri con segno
JLE ;Jump if less or equal (salta se ZF=1 or SF<>OF, cioè se op1<=op2) Opera su numeri con segno
JNA ;Jump if not above (salta se CF=1 or ZF=1, cioè se op1 non è > op2) Opera su numeri senza segno
JNAE ;Jump if not above or equal (salta se CF=1, cioè se op1 non è >= op2) Opera su numeri senza segno
JNB ;Jump if not below (salta se CF=0, cioè se op1 non è < op2) Opera su numeri senza segno
JNBE ;Jump if not below or equal (salta se CF=0 and ZF=0, cioè se op1 non è <= op2) Opera su numeri ;senza segno
JNC ;Jump if not carry (salta se CF=0)
JNE ;Jump if not equal (salta se ZF=0, cioè se op1<>op2)
JNG ;Jump if not greater (salta se ZF=1 or SF<>OF, cioè se op1 non è > op2) Opera su numeri senza segno
JNGE ;Jump if not greater or equal (salta se SF<>OF, cioè se op1 non è >= op2) Opera su numeri senza ;segno
JNL ;Jump if not less (salta se SF=OF, cioè se op1 non è < op2) Opera su numeri senza segno
JNLE ;Jump if not less or equal (salta se ZF=0 and SF=OF, cioè se op1 non è <= op2) Opera su numeri ;senza segno
JNO ;Jump if not overflow (salta se OF=0)
JNP ;Jump if not parity (salta se PF=0)
JNS ;Jump if not sign (salta se SF=0)
JNZ ;Jump if not zero (salta se ZF=0)
JO ;Jump if overflow (salta se OF=1)
JP ;Jump if parity (salta se PF=1)
JPE ;Jump if parity even (salta se PF=1)
JPO ;Jump if parity odd (salta se PF=0)
JS ;Jump if sign (salta se SF=1)
JZ ;Jump if zero (salta se ZF = 1)

Modalità di indirizzamento:
Jx etichetta

Flag alterati:
nessuno

ESEMPIO 8 *Uso dei salti condizionati*

```
;se la cella di indirizzo 0100h contiene un valore diverso da zero poniamo 'a' in AL
;altrimenti poniamo 'z' in AL
            MOV SI,0100h
            CMP Byte Ptr[SI],0
            JNZ non_zero
            MOV AL,'z'              ;istruzione eseguita se la cella contiene 0
            JMP dopo
non_zero:   MOV AL,'a'              ;istruzione eseguita se la cella non contiene 0
dopo:       ...
```

```
;se la cella di indirizzo 0100h contiene un valore maggiore della cella di indirizzo 0101h
;poniamo 'M' in AL, altrimenti 'm' in AL
            MOV SI,0100h
            MOV BL,[SI]
            CMP BL,[SI+1],
            JA maggiore
            MOV AL,'m'              ;istruzione eseguita se la cella 0100 non contiene
                                    ;un numero > della cella 0101
            JMP dopo
maggiore:   MOV AL,'M'              ;istruzione eseguita se la cella 0100 contiene
                                    ;un numero > della cella 0101
dopo:       ...
```

Cicli con istruzione LOOP

L'istruzione LOOP consente di effettuare un ciclo a contatore ripetuto un numero di volte pari al contenuto del registro CX. A ogni istruzione LOOP avviene un decremento di CX; se CX è zero il ciclo termina, altrimenti effettua un salto all'etichetta indicata. La sintassi è le seguente:

```
LOOP ciclo      ;salta se CX <> 0
LOOPZ ciclo     ;salta se CX <> 0 e ZF = 1
LOOPNZ ciclo    ;salta se CX <> 0 e ZF = 0
```

Questa istruzione, se posta al termine di una sequenza, consente di realizzare un ciclo a uscita per falso di tipo post-condizionato di (Do ... While).

Modalità di indirizzamento:
LOOP etichetta

Flag alterati:
nessuno

ESEMPIO 9 *Uso dei cicli LOOP*

In questo esempio vediamo come applicare l'istruzione LOOP per ripetere un ciclo 20 volte. La seconda parte effettua lo stesso ciclo usando un'istruzione aritmetica (che verrà illustrata in seguito), DEC, che decrementa di 1 il contenuto di un registro, per cui se il flag di zero non viene settato salta all'etichetta. In questo caso possiamo notare che non abbiamo usato l'istruzione CMP, perché l'istruzione DEC setta automaticamente il flag di zero quando il risultato è effettivamente zero.

```
;ciclo ripetuto per 20 volte
            MOV CX,14h          ;inizializzazione del contatore CX
ciclo:      ...                 ;istruzioni da ripetere
            LOOP ciclo          ;salta se CX <> 0

;il ciclo sopra equivale alle seguenti istruzioni realizzate senza costrutto LOOP
            MOV CX,14h          ;inizializzazione del contatore CX
ciclo:      ...                 ;istruzioni da ripetere
            DEC CX              ;decrementa il contenuto di CX di 1 unità
            JNZ ciclo           ;salta se CX <> 0
```

La selezione semplice in assembly

I costrutti di selezione semplice consentono di alterare il flusso sequenziale del programma eseguendo segmenti di programma in modo condizionato in base alla valutazione di un'espressione. La selezione semplice permette di eseguire o non eseguire una sequenza in base al risultato della valutazione di un'espressione:

se (condizione)
  Istruzioni
fine se

ESEMPIO 10 *Selezione semplice*

Il codice che segue compie una selezione semplice in assembly (collocata a destra) paragonata alla selezione semplice scritta in un linguaggio evoluto (collocata a sinistra). Il programma esegue il blocco di istruzioni se la condizione è verificata, cioè se la variabile contiene zero:

Codifica in linguaggio C	Codifica in assembly
`int var;`	`.DATA`
`...`	`var  DB  ?  ;dichiarazione variabile di tipo Byte`
`if (var==0)`	`.CODE`
`  {`	`  CMP var,00h  ;confronto con zero`
`  ...`	`  JNZ esci  ;se ZF=0 salta all'etichetta esci, cioè se var<>0`
`  istruzioni`	`  ...  ;se var==0 esegue questo blocco`
`  ...`	`  istruzioni`
`  }`	`  ...`
	`esci:  ...  ;prosecuzione programma`

Il programma esegue il blocco di istruzioni se la condizione è verificata, cioè se la variabile è maggiore di 10:

Codifica in linguaggio C	Codifica in assembly
`int var;`	`.DATA`
`...`	`var  DB  ?  ;dichiarazione variabile di tipo Byte`
`if (var>0)`	`.CODE`
`  {`	`  CMP var,0Ah  ;confronto con 10`
`  ...`	`  JNA esci  ;se var non è > 10 salta all'etichetta esci`
`  istruzioni`	`  ...  ;se var>10 esegue questo blocco`
`  ...`	`  istruzioni`
`  }`	`  ...`
	`esci  ...  ;prosecuzione programma`

Il programma esegue il blocco di istruzioni se la condizione è verificata, cioè se la variabile è compresa tra 10 e 20:

Codifica in linguaggio C	Codifica in assembly
`int var;`	`.DATA`
`...`	`var  DB  ?  ;dichiarazione variabile di tipo Byte`
`if (var>=10) && (var<=20)`	`.CODE`
`  {`	`  CMP var,0Ah  ;confronto con 10`
`  ...`	`  JB esci  ;se var<10 salta all'etichetta esci`
`  istruzioni`	`  CMP var,14h  ;confronto con 20`

Codifica in linguaggio C	Codifica in assembly
...	JA esci ;se var>20 salta all'etichetta esci
}	... ;se var>=10 e <=20 esegue questo blocco
	istruzioni
	...
	esci: ... ;prosecuzione programma

La selezione doppia in assembly

Il costrutto di selezione doppia consente di eseguire in alternativa una delle due sequenze che formano il costrutto in base al risultato della valutazione di una condizione:

se (condizione)
  Istruzioni blocco vero
  altrimenti
  Istruzioni blocco falso
fine se

ESEMPIO 11 *Selezione doppia*

Il codice che segue compie una selezione doppia in assembly (collocata a destra) paragonata alla selezione semplice scritta in un linguaggio evoluto (collocata a sinistra). Il programma esegue il blocco1 di istruzioni se la condizione è verificata, cioè se la variabile contiene zero, altrimenti esegue il blocco2:

Codifica in linguaggio C	Codifica in assembly
int var;	.DATA
...	var DB ? ;dichiarazione variabile di tipo Byte
if (var==0)	.CODE
{	CMP var,00h ;confronto con zero
...	JNZ blocco2 ;se ZF=0 salta all'etichetta blocco2, cioè se var<>0
istruzioni blocco1	... ;se var==0 esegue il blocco1
...	istruzioni blocco1
}	...
else	JMP fine ;salto incondizionato utile per saltare ripetere blocco2
{	blocco2: ... ;blocco eseguito se var<>0
...	istruzioni blocco 2
istruzioni blocco2	...
...	fine: ... ;prosecuzione programma
}	

Il programma è molto simile a quello visto in precedenza per la selezione semplice; è importante sottolineare che l'istruzione JMP fine serve per evitare di eseguire il blocco2 al termine del blocco1.

Come potete notare, il linguaggio evoluto non ha bisogno di tali stratagemmi per uscire dal blocco in quanto la sua struttura prevede l'uscita dal ramo del blocco1 al termine delle istruzioni.

La selezione multipla in assembly

Il costrutto di selezione multipla consente di eseguire una sequenza tra le molte che formano il costrutto in base al risultato della valutazione di un'espressione che viene sempre considerato intero:

scelta (valutazione di una espressione)
  caso 1:
    Istruzioni blocco1
  caso 2:
    Istruzioni blocco2
  ...
  altrimenti:
    Istruzioni blocco altrimenti
fine scelta

ESEMPIO 12 *La selezione multipla*

Il codice che segue compie una selezione doppia in assembly (collocata a destra) paragonata alla selezione semplice scritta in un linguaggio evoluto (collocata a sinistra). Il programma esegue il blocco1 di istruzioni se la condizione è verificata, cioè se la variabile contiene zero, altrimenti esegue il blocco2:

Codifica in linguaggio C	Codifica in assembly
int var;	.DATA
...	var DB ? ;dichiarazione variabile di tipo Byte
switch var {	.CODE
case 0:	CMP var,00h ;confronto con 0
istruzioni blocco1	JZ blocco1 ;se ZF=1 salta all'etichetta blocco1
break;	CMP var,01h ;confronto con 1
case 1:	JZ blocco2 ;se ZF=1 salta all'etichetta blocco2
istruzioni blocco2	...
break;	CMP var,n ;confronto con "n" valore
...	JZ bloccoN ;se ZF=1 salta all'etichetta bloccoN
case "n":	JMP altrimenti ;salto incondizionato per saltare a blocco default
istruzioni blocco"n"	blocco1:... ;blocco eseguito se var=0
default:	istruzioni blocco 1
istruzioni blocco alternativo	...
	JMP fine
}	blocco2:... ;blocco eseguito se var=1
	istruzioni blocco 2
	...
	JMP fine
	bloccoN:... ;blocco eseguito se var = "n"
	istruzioni blocco "n"
	...
	JMP fine
	altrimenti:... ;blocco eseguito se var<>da tutti i valori
	istruzioni blocco "default"
	...
	fine: ... ;prosecuzione programma

La condizione è costituita da una sequenza di controlli sulla stessa variabile; il primo test che soddisfa la condizione determina il salto. In questo esempio emerge l'importanza del fatto che CMP

non altera gli operandi, cosicché non è necessario ricaricare ogni volta il dato. Se nessun caso è soddisfatto, un salto incondizionato salta all'etichetta di default. Ogni blocco deve terminare con un salto incondizionato perché altrimenti verrebbe eseguito anche il caso successivo.

I costrutti iterativi in assembly

I costrutti iterativi consentono di ripetere un blocco di istruzioni più volte modificando il flusso sequenziale del programma attraverso dei salti condizionati in base a una condizione.

Il ciclo precondizionato per vero

Il costrutto precondizionato per vero consente di ripetere una sequenza per una quantità indeterminata di volte uscendo dal ciclo solo quando la condizione è falsa. Essendo posto all'inizio del ciclo, la condizione prevede che la variabile sulla quale opera debba essere necessariamente dichiarata e inizializzata prima di entrare nel ciclo ed è possibile che il ciclo non venga del tutto eseguito se la condizione si dimostra subito falsa.

mentre (condizione)
  Istruzioni
fine mentre

ESEMPIO 13 ***Il ciclo Mentre Fai***

Il codice che segue compie un ciclo precondizionato per vero ripetuto mentre la variabile è maggiore di 10; pertanto deve essere inizializzata a un valore maggiore o uguale a 10 affinché il ciclo venga ripetuto almeno una volta. Il costrutto precondizionato svolto in linguaggio assembly (collocato a destra) viene paragonato al relativo costrutto scritto in un linguaggio evoluto (collocato a sinistra). Il programma esegue il blocco di istruzioni mentre la condizione è verificata, cioè se la variabile è maggiore di 10, altrimenti esce.

Codifica in linguaggio C	Codifica in assembly
`int var;`	`.DATA`
`...`	`var DB 14h ;dichiarazione variabile di tipo Byte e inizializzazione`
`var=20;`	`;a 20`
`while (var>10)`	`.CODE`
`{`	`ciclo: CMP var,0Ah ;confronto con 0`
`...`	`JBE fine ;se var<=10 salta all'etichetta esci`
`istruzioni blocco`	`DEC var ;incremento variabile di 1 unità`
`var=var-1;`	`...`
`}`	`istruzioni blocco`
	`...`
	`JMP ciclo ;salto incondizionato che ritorna a "While"`
	`fine: ... ;prosecuzione programma`

Il ciclo viene ripetuto fino a quando la variabile non risulta essere minore o uguale a 10. Il controllo iniziale prevede una condizione invertita rispetto a quella presentata nella codifica in linguaggio C; infatti il salto condizionato JBE opera un salto quando la variabile contiene un numero minore o uguale a 10, esattamente al contrario rispetto al linguaggio evoluto.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **L'istruzione CMP operando1,operando2 effettua:**

- ○ un'addizione tra operando1 e operando2 senza salvare il risultato
- ○ una sottrazione del tipo operando2-operando1 senza salvare il risultato
- ○ una sottrazione del tipo operando1-operando2 senza salvare il risultato
- ○ una AND logica tra operando1 e operando2 senza salvare il risultato

2 **Il flag di zero (ZF) vale zero quando:**

- ○ il risultato dell'ultima operazione è diverso da zero
- ○ l'ultima operazione ha provocato un cambio di segno inaspettato
- ○ l'ultima operazione ha provocato un riporto
- ○ il risultato dell'ultima operazione è zero

3 **Il flag di carry (CF) vale uno quando:**

- ○ il risultato dell'ultima operazione è diverso da zero
- ○ l'ultima operazione ha provocato un cambio di segno inaspettato
- ○ l'ultima operazione ha provocato un riporto
- ○ il risultato dell'ultima operazione è zero

4 **Se il primo operando di un'istruzione CMP operando1, operando2 è maggiore del secondo abbiamo che:**

- ○ ZF=1
- ○ CF=1
- ○ CF=0 e ZF=0
- ○ CF=1 e ZF=0

5 **Se il primo operando di un'istruzione CMP operando1,operando2 è minore del secondo abbiamo che:**

- ○ CF=1 e ZF=0
- ○ CF=1 e ZF=1
- ○ CF=0 e ZF=0
- ○ CF=0 e ZF=1

6 **L'istruzione JMP etichetta può saltare:**

- ○ in avanti o indietro di uno spiazzamento di 8 bit (-128 +127) con condizione
- ○ in avanti o indietro di uno spiazzamento di 8 bit (-128 +127) senza condizione
- ○ in avanti o indietro di uno spiazzamento di 16 bit (-32768 +32767) senza condizione
- ○ in qualsiasi punto del programma senza condizione

7 **Indica in quale condizione il seguente frammento di codice effettuerà il salto all'etichetta ciclo:**

```
CMP CX,20h
JBE salto
```

- ○ salta se CX è minore di 20h
- ○ salta se CX è maggiore di 20h
- ○ salta se CX è maggiore o uguale a 20h
- ○ salta se CX è minore o uguale a 20h
- ○ salta se CX è diverso da 20h
- ○ salta se CX è uguale a 20h

8 **L'istruzione di salto JBE opera su numeri senza segno e salta quando:**

- ○ se CF=1 e ZF=0
- ○ se CF=1 e ZF=1
- ○ se CF=0 e ZF=0
- ○ se CF=0 e ZF=1

9 **L'istruzione di salto JG opera su numeri con segno e salta quando:**

- ○ se SF=1 e ZF=0
- ○ se SF=OF e ZF=1
- ○ se SF=OF e ZF=0
- ○ se SF=OF e ZF=OF

10 **L'istruzione di salto JA opera su numeri senza segno e salta quando:**

- ○ se CF=1 e ZF=0
- ○ se CF=1 e ZF=1
- ○ se CF=0 e ZF=0
- ○ se CF=0 e ZF=1

11 **L'istruzione LOOP ciclo equivale a:**

- ○
```
INC CX
CMP CX,n
JNE ciclo
```
- ○
```
DEC CX
JZ ciclo
```
- ○
```
DEC CX
JNZ ciclo
```
- ○
```
INC CX
JNZ ciclo
```

12 **Indica a quale costrutto equivale il seguente frammento di codice:**

```
     CMP CL,00h
     JNZ qua
     MOV AX,10h
     INC CL
     JMP fine
qua: MOV DL,01
     INC AX
fine: ...
```

- ○ selezione multipla
- ○ selezione doppia
- ○ selezione semplice
- ○ ciclo precondizionato

Verifichiamo le competenze

Esprimi la tua creatività

Esegui i seguenti esercizi in assembly.

1. Scrivi un programma assembly che trasferisca 255 celle a 8 bit dall'indirizzo 0100h all'indirizzo 0200h.
2. Scrivi un programma assembly che decrementi il contenuto delle celle di indirizzo pari e incrementi il contenuto delle celle di indirizzo dispari per le prime 255 celle a partire dall'indirizzo 0100h.
3. Scrivi un programma assembly che scambi il contenuto di 10 celle di memoria dall'indirizzo 0100h all'indirizzo 0200h.
4. Scrivi un programma assembly che azzeri tutte le celle a partire dalla cella 0100h. Il numero di celle da azzerare è contenuto nella cella precedente alla 0100h
5. Scrivi un programma assembly che azzeri tutte le celle a partire dalla cella 0200h fino a quando non incontra una cella che contiene il carattere ASCII '$'.
6. Scrivi un programma assembly che incrementi il contenuto delle celle di indirizzo pari e decrementi il contenuto delle celle di indirizzo dispari di tutte le celle a partire dall'indirizzo 0100h fino a quando non incontra una cella che contiene il carattere ASCII '?'.
7. Scrivi un programma assembly che decrementi il contenuto delle celle di indirizzo pari e incrementi il contenuto delle celle di indirizzo dispari a partire dall'indirizzo 0300h. La cella precedente alla 0300h contiene il numero di celle da controllare.
8. Scrivi un programma assembly che decrementi il contenuto delle celle a 16 bit di indirizzo pari e incrementi il contenuto delle celle di indirizzo dispari a partire dall'indirizzo 0200h. La cella precedente alla 0200h contiene il numero di celle da controllare.
9. Scrivi un programma assembly che scambi il contenuto delle celle di memoria dall'indirizzo 0100h all'indirizzo 0200h per le sole celle che contengono un numero maggiore di 100.
10. Scrivi un programma assembly che trasferisca le celle di memoria che iniziano all'indirizzo 0200h all'indirizzo 0300h per le sole celle che contengono un numero compreso tra 10 e 50.
11. Scrivi un programma assembly che incrementi il contenuto delle celle che contengono un numero compreso tra 100 e 200 e decrementi il contenuto delle celle che contengono un valore maggiore di 200, a partire dall'indirizzo 0300h. La cella precedente alla 0300h contiene il numero di celle da controllare.
12. Scrivi un programma assembly che scambi il contenuto delle celle di memoria poste a partire dall'indirizzo 0100h con quelle poste a partire dall'indirizzo 0300h, solo per le celle che contengono un valore minore di 100. Il numero di celle da verificare è contenuto nella cella precedente alla 0100h.
13. Scrivi un programma assembly che azzeri tutte le celle che contengono un carattere alfabetico minuscolo (carattere ASCII tra 97 e 122) a partire dalla cella 0100h per un totale di 20 celle.
14. Scrivi un programma assembly che porti a 1 tutte le celle che contengono un carattere alfabetico maiuscolo (carattere ASCII tra 65 e 90) a partire dalla cella 0200h. La cella precedente alla 0200h contiene la lunghezza della tabella.
15. Scrivi un programma assembly che incrementi il contenuto delle celle di indirizzo pari e decrementi il contenuto delle celle di indirizzo dispari di tutte le celle a partire dall'indirizzo 0100h fino a quando non incontra una cella che contiene un numero maggiore di 100.
16. Scrivere un programma assembly che memorizzi in una tabella, a partire dall'indirizzo 0100h, un numero imprecisato di caratteri ASCII terminanti con il simbolo '$'. Il programma deve determinare se i caratteri formano una parola palindroma, come per esempio:
    otto
    radar
    ingegni
    anilina
    itopinonavevanonipoti

UNITÀ DIDATTICA 7

LE ISTRUZIONI ARITMETICHE

IN QUESTA UNITÀ IMPAREREMO...

- a utilizzare le principali istruzioni aritmetiche
- a utilizzare i principali servizi DOS di lettura e scrittura a video/tastiera

L'incremento con l'istruzione INC

L'istruzione INC effettua un'addizione tra l'operando e la costante implicita 1 copiando il risultato nell'operando. Il precedente contenuto del primo operando viene quindi sostituito dalla sua somma con la costante 1. L'operando può essere un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato. Negli ultimi due casi è necessario specificare la dimensione dell'operando che non può essere dedotta dall'istruzione. Il Carry flag non viene in alcun modo alterato da questa istruzione.

Sintassi:

```
INC operando
```

Modalità di indirizzamento:
INC registro
INC memoria

Flag alterati:
ZF, SF, OF, PF

ESEMPIO 14 *Uso dell'istruzione INC*

L'esempio che segue effettua un ciclo per contare, all'interno di una lista di celle che inizia dalla cella di indirizzo 0100, quante celle contengono un valore uguale a zero, inserendo il risultato nel registro DL. Nel ciclo l'istruzione INC incrementa il puntatore SI alla cella successiva e per contare quante celle, che risultano essere pari a zero, sono state trovate. Il ciclo termina quando sono state lette "n" celle.

```
.DATA
n         DB                  0FFh
.CODE
.STARTUP
```

```
        MOV CX,00           ;azzeramento registri usati
        MOV DX,00
        MOV SI,0100h        ;inizializzazione puntatore alla cella iniziale
                            ;della tabella (0100)
        MOV CL,n            ;assegnazione di n a CX per effettuare un ciclo LOOP
ciclo:  CMP Byte Ptr[SI],0  ;confronto tra contenuto cella puntata da SI e 0 (8 bit)
        JNZ dopo            ;se la cella è <> 0 salta all'etichetta dopo
        INC DL              ;se la cella = 0 incremento contatore DL
dopo:   INC SI              ;incremento contenuto del puntatore alla cella (SI)
        LOOP ciclo          ;se CX <> 0 salta a ciclo e decrementa CX
```

Prova adesso!

- Uso di cicli
- Uso dell'istruzione INC
- Uso di istruzioni aritmetiche e puntatori

1 Modifica l'esempio 14 in modo tale da cercare anche quante celle sono diverse da zero nel registro DH.

L'addizione con l'istruzione ADD

L'istruzione ADD permette di sommare il primo operando con il secondo operando, quindi colloca il risultato nel primo operando. Il precedente contenuto del primo operando viene in questo modo sostituito dalla sua somma con il secondo operando (accumulo) mentre il secondo operando rimane inalterato. Il primo operando può essere un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato; il secondo operando può consistere in un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato o un dato immediato. Anche in questo caso i due operandi devono avere le stesse dimensioni, che possono essere Byte, Word o DWord.

Sintassi:

```
ADD operando1,operando2
```

Modalità di indirizzamento:
ADD registro,registro
ADD registro,memoria
ADD memoria,registro
ADD registro,immediato
ADD memoria,immediato

Flag alterati:
ZF, CF, SF, OF, PF

ESEMPIO 15 *Uso dell'istruzione ADD per sommare due celle*

L'esempio che segue mostra come sommare il contenuto di due variabili a 8 bit, ponendo il risultato in un registro a 16 bit (AX). È necessario fare attenzione in quanto la somma di due valori a 8 bit potrebbe risultare maggiore di 8 bit, con conseguente perdita di dati. Per risolvere questo problema utilizziamo due registri a 16 bit che contengono nella parte bassa le variabili a 8 bit. La parte alta dei registri viene azzerata:

```
.DATA
numero1         DB      0Ah
```

```
numero2         DB      0Bh
.CODE
.STARTUP
MOV AL,numero1          ;copia della variabile a 8 bit nella parte bassa di AX (AL)
MOV AH,00h              ;azzeramento della parte alta di AX (AH)
MOV DL,numero2          ;copia della seconda variabile nella parte bassa di DX (DL)
MOV DH,00h              ;azzeramento della parte alta di DX (DH)
ADD AX,DX               ;somma tra AX e DX con risultato in AX
```

ESEMPIO 16 *Uso dell'istruzione ADD per effettuare la somma a 16 bit*

L'esempio che segue mostra come accumulare nel registro DX il contenuto degli elementi della tabella formata dalle "n" celle che incominciano dalla cella di indirizzo 0100h. In questo esempio useremo uno stratagemma per sommare un valore a 8 bit, rappresentato dal singolo elemento della cella, con il registro DX che è invece a 16 bit. Per fare questo collochiamo nella parte bassa di AX (registro AL) il dato letto dalla cella; azzerando la parte alta di AX (registro AH) otteniamo il risultato voluto, cioè un dato a 8 bit in un registro a 16. In tal modo il registro AX può essere indirizzato al registro DX nell'istruzione ADD. Il ciclo di lettura della tabella utilizza un'istruzione LOOP ripetuta CX volte.

```
.DATA
n               DB              0Ah
.CODE
.STARTUP
                MOV CL,n        ;caricamento in CX di un valore a 8 bit assegnandolo
                                ;alla parte bassa del registro
                MOV CH,00h      ;la parte alta di CX viene azzerata
                MOV SI,0100h    ;inizializzazione puntatore a inizio tabella
                MOV DX,00h      ;azzeramento accumulatore DX
ciclo:          MOV AL,[SI]     ;copia della cella nella parte bassa di AX
                MOV AH,00h      ;azzeramento di AH per trasferire il dato a 8 bit letto
                                ;dalla cella su 16 bit (AX)
                ADD DX,AX       ;accumulo del dato letto in AX (16 bit)
                INC SI          ;passiamo a puntare alla cella successiva
                LOOP ciclo      ;decremento di CX e se CX <> 0 salta a ciclo
```

Prova adesso!

- Uso di cicli
- Uso dell'istruzione ADD
- Uso di istruzioni aritmetiche e puntatori

1 Modifica l'esempio 16 in modo tale che vengano sommati solo i valori contenuti nelle celle se maggiori di 10.

Zoom su...

SOMMA CON RIPORTO

Esiste un'istruzione che effettua la somma tra due operandi tenendo conto anche del riporto precedente presente nel Carry flag: si tratta di ADC (Add with Carry). Il precedente contenuto

del primo operando viene quindi sostituito dalla sua somma con il secondo operando (accumulo) mentre il secondo operando rimane inalterato. Il primo operando può essere un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato, come per l'istruzione ADD; il secondo operando può consistere in un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato o un dato immediato come l'istruzione ADD.
Il codice che segue mostra come utilizzare questa istruzione per realizzare una somma a 32 bit tra 4 celle (DWord) di indirizzo iniziale 0100h e la coppia di registri DX:AX. Il risultato viene posto sempre nella coppia di registri formati dalla sequenza DX:AX, dove la parte alta è rappresentata da DX mentre la parte bassa da AX:

```
MOV SI,0100h              ;inizializzazione del registro puntatore alla DWord
MOV AX,01FEh              ;inizializzazione del registro AX (parte bassa del numero
                          ;a 32 bit) con un valore qualunque
MOV DX,00h                ;inizializzazione del registro DX (parte alta del numero
                          ;a 32 bit) con un valore qualunque
ADD AX,Word Ptr[SI]       ;somma tra le parti basse del registro e delle 2 celle
                          ;di memoria puntate da SI (16 bit)
ADC DX,Word Ptr[SI+2]     ;somma tra le parti alte del registro e delle 2 celle
                          ;successive di memoria puntate da SI+2
                          ;(16 bit), inoltre viene considerato anche l'eventuale
                          ;riporto
                          ;della parte bassa presente nel Carry flag
```

La sottrazione con l'istruzione SUB

L'istruzione SUB effettua una sottrazione tra il secondo e il primo operando copiando il risultato nel primo operando. Il precedente contenuto del primo operando viene quindi sostituito dalla sua sottrazione con il secondo operando mentre il secondo operando rimane inalterato. Il primo operando può essere un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato; il secondo operando può consistere in un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato o un dato immediato. I due operandi devono avere le stesse dimensioni, che possono essere Byte, Word o DWord.

Sintassi:

```
SUB operando1,operando2
```

Modalità di indirizzamento:
SUB registro,registro
SUB registro,memoria
SUB registro,immediato
SUB memoria,registro
SUB memoria,immediato

Flag alterati:
ZF, CF, SF, OF, PF

ESEMPIO 17 *Uso dell'istruzione SUB per convertire un carattere ASCII in un numero*

L'esempio che segue legge da tastiera un valore ASCII attraverso il servizio DOS 01h. Il servizio genera nel registro AL un codice ASCII che rappresenta il tasto premuto ed è formato da due istruzioni:

```
;Legge da tastiera e mette il valore ASCII nel registro AL
MOV AH,01h
INT 21h
;Scrive a video il contenuto del registro DL
MOV AH,02h
INT 21h
```

Come abbiamo visto esiste anche un servizio DOS in grado di effettuare l'operazione contraria, cioè stampare a video un carattere ASCII che deve essere preventivamente collocato nel registro DL.

Se l'utente per esempio preme il tasto "0" il carattere ASCII restituito è 30h. Se il numero è il "9" il carattere è 39h. Pertanto per effettuare la conversione da ASCII a numero dobbiamo sottrarre 30h al carattere letto. Il ciclo termina quando l'utente preme INVIO, dato dal carattere ASCII 13 (0Dh). I numeri letti da tastiera vengono memorizzati nelle celle a partire dall'indirizzo 0100h. L'esempio conta nella variabile totale quanti numeri sono stati letti.

```
.DATA
totale  DB            00h
.CODE
.STARTUP
        MOV CX,00h    ;azzeramento registro CX che viene usato come contatore
        MOV SI,0100h  ;inizializzazione puntatore a 0100h
        MOV AH,01h    ;servizio DOS che legge un carattere ASCII da tastiera
        INT 21h       ;e lo pone nel registro AL
ciclo:  CMP AL,0Dh    ;verifica se il carattere è INVIO (0Dh)
        JE fine       ;salto condizionato che salta all'etichetta fine se AL contiene 0Dh
        SUB AL,30h    ;trasformazione da carattere ASCII a numero sottraendo 30h (48)
        MOV [SI],AL   ;inserimento numero nella memoria all'indirizzo puntato da SI
                      ;(8 bit)
        INC SI        ;incremento puntatore a cella successiva
        INC CL        ;contatore di numeri letti
        JMP ciclo     ;salto incondizionato all'etichetta ciclo per inserire un altro
                      ;numero
fine:   MOV totale,CL ;assegnazione alla variabile totale del numero di celle lette
```

ESEMPIO 18 *Stampa a video di numeri*

L'esempio che segue stampa a video i numeri letti dalla memoria come caratteri ASCII attraverso il servizio DOS 02h. Il servizio legge il contenuto del registro DL, lo converte in ASCII sommando 48 (30h) e lo stampa. L'esercizio completa il precedente che aveva memorizzato nella variabile totale il numero di celle di memoria lette da tastiera.

```
.CODE
.STARTUP
        MOV SI,0100h  ;inizializzazione puntatore a 0100h
ciclo:  SUB totale,1  ;decremento della variabile totale
        CMP totale,0  ;verifica se siamo arrivati alla fine della tabella
        JE fine       ;salto condizionato che salta all'etichetta fine quando totale=0
        MOV DL,[SI]   ;copia la cella da stampare nel registro DL
        ADD DL,30h    ;trasformazione da numero a carattere ASCII sommando 30h (48)
        MOV AH,02h    ;servizio DOS che stampa un carattere ASCII a video
        INT 21h       ;presente nel registro DL
        INC SI        ;incremento puntatore a cella successiva
```

```
        JMP ciclo      ;salto incondizionato all'etichetta ciclo per passare al numero
                       ;successivo
fine:   ...
```

Prova adesso!

- Uso di cicli
- Uso dell'istruzione SUB
- Uso di istruzioni aritmetiche e puntatori
- Uso dei servizi DOS 01h e 02h

1 Modifica l'esempio 18 in modo tale che vengano stampati solo i numeri diversi da zero.

Il decremento con l'istruzione DEC

L'istruzione DEC effettua una sottrazione tra l'operando e la costante implicita 1 copiando il risultato nell'operando. Il precedente contenuto del primo operando viene quindi sostituito dalla sua differenza con il dato immediato 1. L'operando può essere un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato. Negli ultimi due casi è necessario specificare la dimensione dell'operando che non può essere dedotta dall'istruzione. Il Carry flag non viene in alcun modo alterato da questa istruzione.

Sintassi:

```
INC operando
```

Modalità di indirizzamento:
DEC registro
DEC memoria

Flag alterati:
ZF, SF, OF, PF

ESEMPIO 19 *Uso dell'istruzione DEC*

L'esempio che segue effettua un ciclo per stampare a video i caratteri ASCII contenuti a partire dalla cella 0100h. La variabile "n" indica quante celle stampare. I caratteri devono tuttavia essere stampati al contrario, cioè partendo dall'ultimo della tabella: per fare questo utilizziamo un'istruzione aritmetica che permetta di far puntare al registro DI la fine della tabella, sommandone il contenuto alla variabile "n". Il ciclo termina quando il registro DI diventa uguale al valore 0100h.

```
.DATA
n       DB          02Ch
.CODE
.STARTUP
        MOV AL,n    ;assegnazione del totale delle celle da stampare alla parte
                    ;bassa di AX (AL)
        MOV AH,0    ;azzeramento della parte alta di AX (AH)
        MOV DI,0100h ;inizializzazione puntatore alla cella 0100 di inizio tabella
        ADD DI,AX   ;somma tra DI e AX che contiene il totale delle celle da stampare
                    ;in tal modo DI punta all'ultima cella della tabella
ciclo:  CMP DI,0100h ;verifica se DI punta alla prima cella
        JB fine     ;Se DI<0100 salta alla fine del ciclo
```

```
        MOV DL,[DI]    ;assegnazione del carattere ascii da stampare al registro DL
        MOV AH,02h     ;servizio DOS di stampa a video
        INT 21h        ;servizio DOS
        DEC DI         ;decremento del contenuto del puntatore DI, così punta
                       ;alla cella precedente
        JMP ciclo      ;salto incondizionato a inizio ciclo
fine:   ...            ;prosecuzione programma
```

Prova adesso!

- Uso di cicli
- Uso dell'istruzione DEC
- Uso di istruzioni aritmetiche e puntatori
- Uso dei servizi DOS 01h e 02h

1 Modifica l'esempio 19 in modo tale che vengano stampate le celle partendo dalla 0100 secondo un ordine sequenziale crescente.

La divisione con l'istruzione DIV

L'istruzione DIV effettua una divisione ponendo a denominatore l'operando indicato. Se l'operando è a 8 bit effettua la divisione tra AX e l'operando. Se l'operando è a 16 bit effettua la divisione tra il numero a 32 bit formato dalla coppia di registri DX:AX e l'operando. Nella divisione a 8 bit il quoziente viene collocato nel registro AL e il resto nel registro AH. Nella divisione a 16 bit il quoziente viene collocato nel registro AX e il resto nel registro DX. Nella divisione a 8 bit il registro DX deve essere sempre azzerato prima di effettuare la divisione, a causa di un bug dell'assemblatore Intel che non è stato ancora risolto. I flag CF e OF sono settati se il valore occupa la parte alta del risultato.

◀ ISR (*Interrupt Service Routine*) Le Interrupt Service Routine sono particolari procedure che vengono attivate quando si verifica un'interruzione. Le ISR sono routine di sistema che possono tuttavia essere scritte anche dall'utente, anche se la loro gestione è relativa al sistema operativo DOS; pertanto con versioni di Windows a partire da XP non possono più essere gestite. ▶

Nella versione IDIV viene applicata la regola del segno, cioè se i segni degli operandi sono concordi il risultato è positivo mentre se sono discordi è negativo.

L'operazione di divisione non può essere effettuata per denominatori pari a zero e inoltre il quoziente della divisione potrebbe eccedere la dimensione del registro implicito; in entrambi i casi la CPU lancia un interrupt interno (INT 0) che esegue una ◀ ISR ▶, cioè una procedura che consente una gestione dell'errore da parte del sistema operativo.

Sintassi:

```
DIV operando
```

Modalità di indirizzamento:
DIV registro
DIV memoria

Flag alterati:
CF, OF

ESEMPIO 20 *Determinazione di numeri pari e dispari*

L'esempio che segue effettua un ciclo per contare quanti numeri pari sono presenti in una tabella di elementi a partire dall'indirizzo di memoria 0100h. La variabile "n" indica la lunghezza della tabella. Il ciclo innanzitutto legge i numeri, quindi effettua una divisione a 8 bit per due usando la variabile di tipo Byte due che contiene proprio la costante 2. Se il resto è uguale a zero viene incrementato il contatore di numeri pari (registro BL).

```
.DATA
n       DB          0FFh
due     DB          02h
.CODE
.STARTUP
        MOV SI,0100h  ;inizializzazione puntatore SI a inizio tabella (0100)
        MOV CX,0
        MOV CL,n      ;inizializzazione contatore CX alla dimensione della tabella
ciclo:  MOV AL,[SI]   ;copia del valore da verificare nel registro AL (8 bit) dalla
                      ;cella puntata da SI
        MOV AH,0      ;azzeramento parte alta di AX
        MOV DX,0      ;azzeramento di DX
        DIV due       ;divisione per due (8 bit)
        CMP AH,0      ;confronto del resto (AH) con zero
        JNZ dopo      ;se resto <> 0 numero dispari quindi salta a dopo
        INC BL        ;se resto = 0 incrementa il contatore di numeri pari (BL)
dopo:   INC SI        ;incremento puntatore celle di una unità
        LOOP ciclo    ;decremento di CX. Se CX <> 0 salta a ciclo
```

Prova adesso!

- Uso di cicli
- Uso dell'istruzione DIV
- Uso di istruzioni aritmetiche e puntatori

1. Modifica l'esempio 20 in modo tale che la ricerca dei numeri pari avvenga su celle da 16 bit.
2. Modifica l'esempio 20 contando anche quanti numeri dispari sono presenti nel registro BL.

ESEMPIO 21 *Conversione di un numero da decimale a binario*

L'esempio che segue mostra il codice necessario per la conversione di un numero in binario e la stampa dello stesso a video. Il numero è presente in memoria nella cella 0100h. L'algoritmo prevede che il numero venga diviso per due fino a quando il quoziente è pari a zero. I resti verranno stampati a video mediante il servizio DOS 02h. Il ciclo si ripete fino a quando il quoziente (AL) è zero. All'interno del ciclo vengono salvati nello stack tutti i resti, in modo che possano essere estratti al contrario dalla catasta per essere stampati. La variabile conta viene utilizzata per conoscere le cifre del numero in binario, utili al ciclo di stampa che deve essere ripetuto un numero corretto di volte, per non incorrere in un'istruzione POP a vuoto che provocherebbe un errore di esecuzione.

```
.DATA
due       DB          02h
conta     DB          0
.CODE
.STARTUP
```

```
          MOV SI,0100h ;inizializzazione puntatore alla cella che contiene numero
                       ;da stampare
          MOV AX,0     ;azzeramento registro
          MOV AL,[SI]  ;AL contiene numero da convertire
ciclo:    MOV AH,0     ;azzeramento della parte alta di AX necessario per fare
                       ;la divisione a 8 bit
          DIV due      ;divisione a 8 bit tra AX e variabile due (2)
          MOV BL,AL    ;salvataggio del quoziente in BL
          CMP AH,0     ;verifica se resto = 0
          JE zero      ;se resto = 0 salto a etichetta zero
          MOV AL,'1'   ;se resto <> 0 collochiamo '1' come carattere ASCII in AL
          PUSH AX      ;salvataggio del resto
          JMP dopo     ;salto incondizionato all'etichetta successiva
zero:     MOV AL,'0'   ;se resto = 0 collochiamo '0' come carattere ASCII in AL
          PUSH AX      ;salvataggio del resto
dopo:     MOV AX,0     ;azzeramento di AX
          MOV AL,BL    ;recupero quoziente salvato in BL
          INC conta    ;incremento contatore di cifre binarie
          CMP AL,0     ;confronto se quoziente = 0
          JE stampa    ;se quoziente = 0 allora termine divisioni e salto a stampa
          JMP ciclo    ;se quoziente <> 0 salto a ciclo per continuare le divisioni per 2
stampa:   MOV CL,conta ;colloco il conta cifre binarie in CX
          MOV CH,0
ciclo_s:  POP AX       ;recupero resto da stack
          MOV DL,AL    ;copio resto in DL per servizio DOS
          MOV AH,02h   ;servizio DOS 02 che stampa a video DL
          INT 21h
          LOOP ciclo_s ;decremento CX, se CX <> 0 salta a ciclo_s
          MOV AH,4Ch   ;servizio DOS che termina programma
          INT 21h
END
```

Prova adesso!

- Uso di cicli
- Uso dell'istruzione DIV
- Uso di istruzioni aritmetiche e puntatori

1 Modifica l'esempio 21 in modo tale che il numero da convertire sia a 16 bit e sia presente nella cella puntata dal registro DI.

La moltiplicazione con l'istruzione MUL

L'istruzione MUL effettua una moltiplicazione tra l'operando indicato e il registro implicito AL o AX a seconda della dimensione dell'operando. Se l'operando è a 8 bit viene moltiplicato il registro AL per l'operando e il prodotto viene collocato nel registro AX. Se l'operando è a 16 bit viene moltiplicato il registro AX per l'operando e il prodotto viene collocato nella coppia di registri DX:AX, dove DX rappresenta i 16 bit più significativi e AX i 16 bit meno significativi del prodotto (32 bit). I flag CF e OF sono settati se il valore occupa la parte alta del risultato.
Nella versione IMUL viene applicata la regola del segno, cioè se i segni degli operandi sono concordi il risultato è positivo mentre se sono discordi è negativo. Per esempio, se con MUL di tipo Byte si moltiplica 0FFh per 0FFh si ottiene 0FE01H, mentre facendo la stessa operazione con la IMUL si ottiene:

−1 (FFh) * −1 (FFh) = 1 (0001h)

Sintassi:

```
MUL operando
```

Flag alterati:
CF, OF

Modalità di indirizzamento:
MUL registro
MUL memoria

ESEMPIO 22 *Lettura da tastiera di un numero a più cifre*

L'esempio che segue mostra il codice necessario per leggere da tastiera un numero a più cifre (fino a 16 bit) da collocare nel registro BX. Il problema non è di banale interpretazione in quanto il servizio DOS di lettura da tastiera consente l'immissione di un solo carattere ASCII per volta. L'algoritmo di soluzione prevede che il ciclo termini quando l'utente preme il tasto INVIO (carattere ASCII 13 o 0Dh). Nel ciclo innanzitutto viene moltiplicato per 10 il contenuto del registro di accumulo del risultato (BX), quindi viene sommato al registro il valore numerico della cifra letta da tastiera.
Per esempio, per il numero 62341 il programma compie i seguenti passi:

Legge il carattere 6
- se diverso da INVIO
  BX = BX * 10 + 6 = 6

Legge il carattere 2
- se diverso da INVIO
  BX = BX * 10 + 2 = 6 * 10 + 2 = 62

Legge il carattere 3
- se diverso da INVIO
  BX = BX * 10 + 3 = 62 * 10 + 3= 623

Legge il carattere 4
- se diverso da INVIO
  BX = BX * 10 + 6 = 623 * 10 + 4= 6234

Legge il carattere 1
- se diverso da INVIO
  BX = BX * 10 + 6 = 6234 * 10 + 1 = 62341

Legge il carattere INVIO
- se = INVIO
  BX contiene il risultato = 62341

```
.DATA
dieci    DW          0Ah
.CODE
.STARTUP
         MOV BX,0    ;azzero registro di accumulo della cifra a 16 bit
ciclo:   MOV AH,01h  ;servizio DOS di lettura da tastiera
         INT 21h
         CMP AL,0Dh  ;il carattere ASCII letto viene confrontato con INVIO (carattere 13)
         JE fine     ;se INVIO premuto la cifra è terminata salto a fine
         PUSH AX     ;salvo il numero letto nello stack
         MOV AX,BX   ;moltiplico per 10 il numero accumulato in BX
         MUL dieci   ;AX * 10
         MOV BX,AX   ;colloco in BX il risultato della moltiplicazione per 10
```

```
        POP AX      ;riprendiamo in AX l'ultima cifra letta
        SUB AL,30h  ;la cifra viene aggiustata da ASCII a numero sottraendo 30h
        MOV AH,0    ;azzero parte alta di AX
        ADD BX,AX   ;somma del valore di accumulato in BX con l'ultima cifra letta
        JMP ciclo   ;salto incondizionato al ciclo di lettura da tastiera
fine:   ...         ;prosecuzione programma
```

Prova adesso!

- Uso di cicli
- Uso dell'istruzione DIV
- Uso di istruzioni aritmetiche e puntatori

1. Modifica l'esempio 22 in modo tale che il numero presente in BX venga stampato a video usando il servizio DOS 02h.
2. Confronta la tua soluzione con quella proposta di seguito:

```
.DATA
dieci   DW          0Ah
.CODE
.STARTUP
        MOV AX,BX   ;poniamo in AX il numero da stampare
        MOV CL,00   ;azzeramento contatore di cifre
sta:    MOV DX,0    ;azzeramento parte alta della cifra a 32 bit che viene
                    ;divisa per 10
        DIV dieci   ;divisione a 16 bit per fare DX:AX/10. Resto in DX.
                    ;Quoziente in AX
        PUSH DX     ;salvataggio del resto nello stack
        INC CL      ;aggiornamento contatore cifre
        CMP AX,0    ;verifica se quoziente = 0
        JNE sta     ;salta a etichetta sta se quoziente <> 0
rip:    POP DX      ;recupero resto da stampare
        ADD DL,30h  ;aggiustamento cifra come carattere ASCII sommando 48 (30h)
        MOV AH,02h  ;servizio DOS di stampa a video
        INT 21h
        DEC CL      ;decremento contatore cifre
        JNZ rip     ;se diverso da zero ripeto la stampa di un'altra cifra
        ...         ;altrimenti termine programma
```

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Indica la funzione dell'istruzione seguente:**

```
INC [SI]
```

- ○ incrementa di un'unità il registro SI
- ○ incrementa di un'unità la cella puntata dal registro SI
- ○ È errata in quanto non indica la dimensione della memoria coinvolta
- ○ È errata in quanto la cella non può essere indirizzata con la parentesi quadra, bensì con la parentesi tonda

2 **Che cosa contiene il registro BH al termine del seguente programma?**

```
        MOV BX,1001h
        MOV CX,0h
ciclo:  DEC BX
        INC CX
        CMP CX,05h
        JNZ ciclo
```

- ○ 01h
- ○ 0Fh
- ○ 10h
- ○ FCh
- ○ 0Eh

3 **Che cosa contiene il registro BL al termine del seguente programma?**

```
        MOV BX,01FDh
        MOV CX,03h
ciclo:  INC BX
        DEC CX
        CMP CX,0h
        JNZ ciclo
```

- ○ 00h
- ○ 01h
- ○ FFh
- ○ 02h

4 **Quale tra le seguenti affermazioni riguardo all'istruzione ADC è errata?**

- ○ se si verifica un riporto esso viene collocato nel carry flag
- ○ esegue la somma tra il registro AX e l'operando indicato
- ○ esegue la somma tra gli operandi indicati e il risultato viene posto nel primo operando
- ○ esegue la somma tra due operandi

5 **Indica il funzionamento del seguente programma:**

```
MOV DI,0100h
MOV BX,1E00h
MOV DX,00h
ADD BX,Word Ptr[DI]
ADC DX,Word Ptr[DI+2]
```

- ○ somma a 32 bit tra la DWord di indirizzo 0100, il carry flag e DX:BX
- ○ somma a 32 bit tra la DWord di indirizzo 0100 e DX:BX
- ○ somma a 32 bit tra la DWord di indirizzo 0100 e il carry flag
- ○ somma a 16 bit tra la Word di indirizzo 0100, il carry flag e il registro BX

6 **Che cosa contiene il registro BL al termine del seguente programma?**

```
        MOV AX,0h
        MOV DX,0FFFCh
        MOV BL,10h
        MOV CX,04h
ciclo:  ADD DX,02h
        ADC AX,0h
        DEC CX
        SUB BL,AL
        CMP CX,0h
        JNZ ciclo
```

- ○ FFh
- ○ 0Eh
- ○ 0Fh
- ○ 0Dh

7 **Il servizio DOS usato nell'istruzione seguente:**

```
MOV DL,'A'
MOV AH,02h
INT 21h
```

- ○ legge un carattere da tastiera e lo pone nel registro AL
- ○ visualizza il carattere ASCII A
- ○ visualizza il numero 65d
- ○ visualizza il numero 48h

8 **Il servizio DOS usato nell'istruzione seguente:**

```
MOV DL,'A'
MOV AH,01h
INT 21h
```

- ○ legge un carattere da tastiera e lo pone nel registro AL
- ○ visualizza il carattere ASCII A
- ○ visualizza il numero 65d
- ○ visualizza il numero 48h

Verifichiamo le competenze

Esprimi la tua creatività

Esegui i seguenti esercizi in assembly.

1. Scrivi un programma assembly che sommi i contenuti di due celle contigue (0100h e 0101h) e ponga il risultato nella cella successiva. Ripeti l'esercizio per due valori a 16 bit.
2. Scrivi un programma assembly che sottragga i contenuti di due celle contigue (0100h e 0101h) e ponga il risultato nella cella successiva. Ripeti l'esercizio per due valori a 16 bit.
3. Scrivi un programma assembly che effettui la divisione tra il contenuto di due celle contigue (0100h e 0101h), ponendo il quoziente e il resto nelle due celle successive. Ripeti l'esercizio per due valori a 16 bit.
4. Data una tabella di 100 elementi, scrivi un programma assembly che conti quanti elementi sono pari e quanti sono dispari scrivendone il totale alla fine della tabella.
5. Date due tabelle di indirizzo 0100h e 0200h e di dimensione "n", scrivi un programma assembly che cerchi la posizione degli elementi dispari scrivendone la posizione nella seconda tabella. Le posizioni relative sono numeri a 8 bit che indicano la posizione dell'elemento all'interno della tabella (0 per il primo, 1 per il secondo, ecc.).
6. Data una tabella a partire della cella 0100h e di dimensione "n", scrivi un programma assembly che moltiplichi per due gli elementi di indirizzo pari e divida per due gli elementi di indirizzo dispari.
7. Data una tabella a partire della cella 0100h e di dimensione "n", scrivi un programma assembly che moltiplichi per due gli elementi di indirizzo dispari e divida per due gli elementi di indirizzo pari.
8. Data una tabella a partire della cella 0100h e di dimensione "n", scrivi un programma assembly che moltiplichi per due gli elementi di indirizzo relativo pari e divida per due gli elementi di indirizzo relativo dispari.
9. Scrivi un programma assembly che legga una serie di caratteri ASCII terminanti con il carattere '0' e li memorizzi a partire dall'indirizzo 0100h. Calcola, ponendo il risultato alla fine della tabella, quante volte viene ripetuto un determinato carattere.
10. Scrivi un programma assembly che legga un numero a più cifre e ponga il risultato nella cella 0100h. Dopo aver diviso tale numero per 2 visualizza il quoziente a video usando l'algoritmo di scrittura a video di un numero vista nell'unità.
11. Scrivi un programma assembly che legga un numero a più cifre e ponga il risultato nella cella 0200h. Dopo aver moltiplicato tale numero per 2 visualizza il risultato a video usando l'algoritmo di scrittura a video di un numero visto nell'unità.
12. Data una tabella di 100 elementi scritta in memoria a partire dall'indirizzo 0200h, scrivi un programma assembly che calcoli quanti numeri primi (divisibili per 1 o per se stessi) vi sono tra loro, inserendo il totale alla fine della tabella stessa.
13. Data una tabella di 100 elementi scritta in memoria a partire dall'indirizzo 0200h, scrivi un programma assembly che stampi a video un elemento della tabella stessa, la cui posizione relativa è data dal numero letto da tastiera.
14. Scrivi un programma assembly che stampi a video la posizione relativa del primo valore diverso da zero presente in una tabella di 100 elementi posta a partire dall'indirizzo 0100h.
15. Scrivi un programma assembly contenente due tabelle lunghe 250 elementi, delle quali una inizia dalla cella di indirizzo 0100h mentre l'altra dall'indirizzo 0300h. Il programma deve trasferire dalla tabella 1 tutti i numeri moltiplicando per due solo quelli pari.
16. Scrivi un programma assembly contenente due tabelle lunghe "n" elementi, delle quali una inizia dalla cella di indirizzo 0100h mentre l'altra dall'indirizzo 0200h. Il programma deve trasferire dalla tabella 1 tutti i numeri dividendo per due solo quelli pari.
17. Scrivi un programma assembly contenente tre tabelle lunghe 100 elementi, che iniziano dagli indirizzi 0100, 0200 e 0300. Il programma deve dividere i valori contenuti nella tabella 1 per 3, la tabella 2 conterrà i quozienti mentre la tabella 3 i resti.

UNITÀ DIDATTICA 8

LE ISTRUZIONI LOGICHE E DI MANIPOLAZIONE DEI BIT

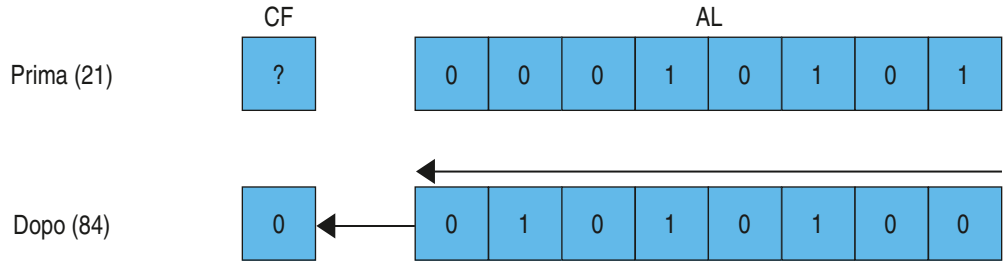
IN QUESTA UNITÀ IMPAREREMO...

- a utilizzare le principali istruzioni bit wise e logiche
- a utilizzare le principali istruzioni di scorrimento e rotazione
- a utilizzare i principali servizi DOS di stampa stringa a video

Lo scorrimento aritmetico con le istruzioni SAL e SAR

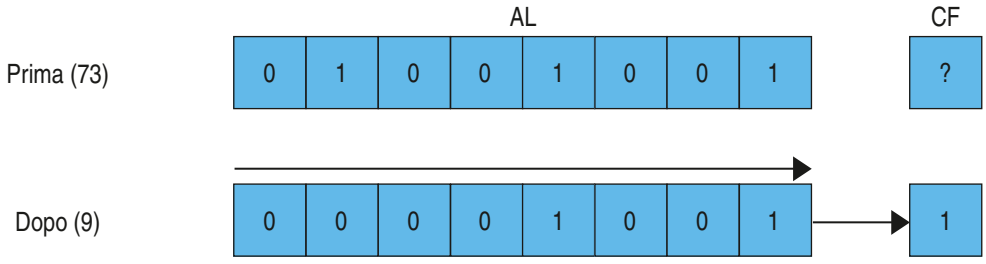
L'istruzione SAL effettua uno scorrimento (shift) a sinistra dei bit dell'operando. Il bit più significativo dell'operando viene copiato nel CF, mentre il bit meno significativo viene inizializzato a zero. Gli scorrimenti, il cui numero è determinato dal dato immediato o dal contenuto del registro CL, sono 1-8 per operandi di tipo Byte, 1-16 per operandi di tipo Word, 1-32 per operandi di tipo DWord. L'operazione coincide con una moltiplicazione per una potenza del 2 (2, 4, 8, 16, 32, 64, 128 ecc.). Vediamo che cosa accade eseguendo la seguente istruzione:

```
SAL AL,2            ;equivale a moltiplicare AL per 4
```



L'istruzione SAR effettua uno scorrimento (shift) a destra dei bit dell'operando. Il bit meno significativo dell'operando viene copiato nel CF, mentre il bit più significativo viene inizializzato a zero. Gli scorrimenti, il cui numero è determinato dal dato immediato o dal contenuto del registro CL, sono 1-8 per operandi di tipo Byte, 1-16 per operandi di tipo Word, 1-32 per operandi di tipo DWord. L'operazione coincide con una divisione per una potenza del 2 (2, 4, 8, 16, 32, 64, 128 ecc.). Vediamo che cosa accade eseguendo la seguente istruzione:

```
SAR AL,3            ;equivale a dividere AL per 8
```



Sintassi:

```
SAL operando,bit
SAR operando,bit
```

Modalità di indirizzamento:
SAL registro,immediato
SAL registro,CL
SAR registro,immediato
SAR registro,CL

Flag alterati:
ZF, CF, SF, OF, PF

ESEMPIO 23 *Uso dello scorrimento per verificare la parità di un numero*

L'esempio seguente utilizza le istruzioni di scorrimento per verificare se un numero è pari o dispari. Effettuando infatti uno scorrimento a destra di 1 bit (SAR) possiamo valutare in base al contenuto del CF se il numero sia pari oppure no. Se il Carry flag è attivo significa che il numero è dispari, dal momento che il bit meno significativo dei numeri dispari è sempre pari a 1. In questo caso vogliamo conoscere quante celle contengono numeri pari utilizzando questo metodo. Il totale deve essere poi memorizzato alla fine della tabella. La dimensione della tabella è contenuta nella variabile "n".

```
.DATA
n       DB          01Fh
.CODE
        MOV BX,0h     ;azzeramento registro contatore numeri pari
        MOV SI,0100h  ;inizializzazione registro puntatore a inizio tabella
ciclo:  MOV AL,[SI]   ;copia del valore da controllare in AL
        SAR AL,1      ;scorrimento a destra
        JC dispari    ;se CF attivo allora il numero è dispari
        INC BL        ;altrimenti se è pari incremento contatore numeri pari (BL)
dispari:INC SI        ;incremento puntatore
        DEC n         ;decremento dimensione tabella
        JNZ ciclo     ;se dimensione <>0 salto a ciclo per prossimo numero da controllare
        MOV [SI],BL   ;copia del risultato alla fine della tabella
```

Lo scorrimento a sinistra può essere usato invece per valutare se un numero espresso con segno è positivo oppure negativo: se CF=1 numero negativo, se CF=0 numero positivo.

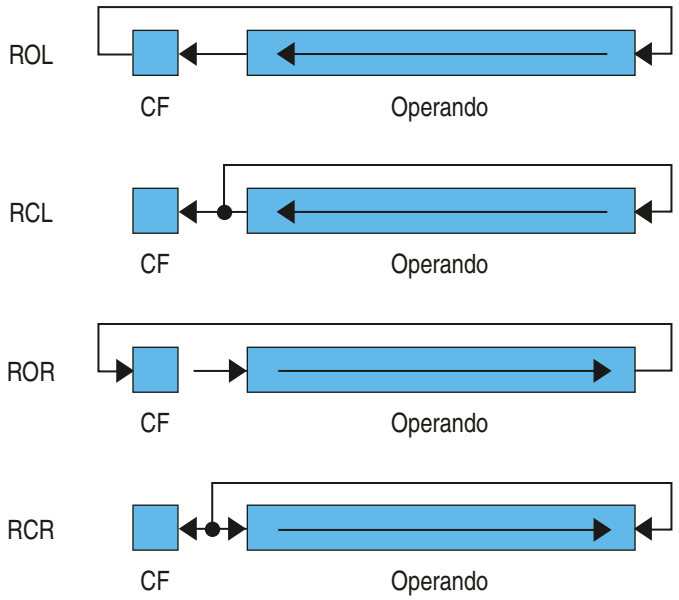
Prova adesso!

- Uso delle istruzioni SAL e SAR
- Uso dei puntatori
- Uso del servizio DOS di stampa

1. Modifica l'esempio 23 in modo da moltiplicare per due i risultati usando lo shift a sinistra. Rispondi alla seguente domanda: tutti i numeri possono essere moltiplicati per due con questo metodo?
2. Modifica infine il codice in modo che possa contare quanti numeri minori di 128 ci sono nella tabella (possiedono il bit più significativo a zero).

La rotazione attraverso le istruzioni ROL, ROR, RCL, RCR

L'istruzione ROL effettua una rotazione (rotate) a sinistra dei bit dell'operando interpretando il CF come bit più significativo dell'operando. A ogni scorrimento il CF passa nel bit meno significativo e il bit più significativo passa nel CF.
L'istruzione RCL effettua una rotazione (rotate) a sinistra dei bit dell'operando. A ogni scorrimento il bit più significativo passa nel bit meno significativo e viene anche copiato nel CF.
L'istruzione ROR effettua una rotazione (rotate) a destra dei bit dell'operando interpretando il CF come bit meno significativo dell'operando. A ogni scorrimento il CF passa nel bit più significativo e il bit meno significativo passa nel CF.
L'istruzione RCR effettua una rotazione (rotate) a destra dei bit dell'operando. A ogni scorrimento il bit meno significativo passa nel bit più significativo e viene anche copiato nel CF.
Il numero di rotazioni è determinato dalla costante immediata o dal contenuto del registro CL (1-8 per operazioni Byte, 1-16 per operazioni Word, da 1-32 per operazioni DWord).



L'istruzione di rotazione non è distruttiva come un'istruzione di scorrimento. Infatti, procedendo con una rotate nella direzione opposta dello stesso numero di bit, possiamo ricostruire la cifra originale che è stata ruotata.

Sintassi:

```
ROR operando,bit
RCR operando,bit
ROL operando,bit
RCL operando,bit
```

Modalità di indirizzamento:
ROR registro,immediato
ROR registro,CL
RCR registro,immediato
RCR registro,CL
ROL registro,immediato
ROL registro,CL
RCL registro,immediato
RCL registro,CL

Flag alterati:
CF

ESEMPIO 24 *Uso della rotazione per scambiare i nibble di un byte*

L'esempio seguente utilizza le istruzioni di rotazione per scambiare il nibble più significativo con il nibble meno significativo di un byte. Un nibble è la metà di un byte, quindi corrisponde a 4 bit. Il registro AL contiene ABh e al termine del programma deve contenere BAh.

```
.CODE
      MOV AL,0ABh  ;numero da scambiare nei nibble meno significativo e più significativo
      MOV CL,04h   ;CL viene inizializzato a 4
      ROL AL,CL    ;rotazione di 4 bit a sinistra (AL contiene ora BAh)
```

Prova adesso!

- Uso delle istruzioni ROL e ROR
- Uso dei puntatori
- Uso del servizio DOS di stampa

1. Modifica l'esempio 24 in modo da scambiare i 2 byte contenuti nel registro DX.
2. Modifica il codice stampando a video il risultato finale.

La congiunzione logica con l'istruzione AND

L'istruzione AND esegue l'And logico bit a bit tra il primo e il secondo operando, chiamato maschera, e copia il risultato nel primo operando. Il precedente contenuto del primo operando viene quindi sostituito dal risultato che contiene 1 nei bit che erano a 1 tanto nel primo quanto nel secondo operando, mentre il secondo operando rimane inalterato. Il primo operando può essere un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato; il secondo operando può consistere in un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato o un dato immediato. I due operandi devono avere le stesse dimensioni, che possono essere di tipo Byte, Word o DWord. La tecnica di mascheramento dei bit mediante l'istruzione AND consente di stabilire il valore di un bit posto in una determinata posizione. Per

esempio, per verificare se il bit posto alla posizione 7 di un numero binario è zero basta effettuare un'istruzione AND con una maschera rappresentata da una sequenza di zeri eccetto il bit da verificare. Se il flag di zero viene settato il bit alla posizione 7 è 1, altrimenti è zero. L'esempio seguente stampa 1 oppure zero a seconda del valore assunto dal bit di posizione 7 presente nella variabile numero:

```
      MOV AX,numero              ;numero da verificare
      AND AX,0000000001000000b   ;il flag di zero dipende solo dal bit di posizione 7
      JZ uno                     ;se flag di zero attivo significa che bit = 1
      MOV DL,'0'                 ;altrimenti bit = 0. Metto '0' nel registro DL per
                                 ;stamparlo
      JMP dopo                   ;salto all'etichetta dopo
uno:  MOV DL,'1'                 ;se flag di zero attivo metto '1' nel registro DL per
                                 ;stamparlo
dopo: MOV AH,02h                 ;servizio DOS che stampa a video il contenuto del
                                 ;registro DL
      INT 21h
```

Sintassi:

```
AND operando,maschera
```

Modalità di indirizzamento:
AND registro,immediato
AND registro,memoria
AND registro,immediato
AND memoria,immediato
AND memoria,registro

Flag alterati:
ZF, CF = 0, SF, OF, PF

ESEMPIO 25 *Conversione da ASCII in numero con AND logico*

L'esempio che segue mostra come ottenere il valore numerico di un carattere ASCII compreso fra 30h e 39h. Tali valori rappresentano l'alternativa alla più classica sottrazione per 30h. Mascherando infatti i 4 bit più significativi di un carattere ASCII mediante un'istruzione AND si ottiene il numero. Consideriamo l'esempio seguente:

Carattere ASCII del numero 9 (39h)
0 1 1 1 1 0 0 1

Maschera (0Fh)
0 0 0 0 1 1 1 1

AND tra carattere ASCII e maschera (9)
0 0 0 0 1 0 0 1

```
.DATA
carattere  DB   '9'
numero     DB   0
.CODE
.STARTUP
```

```
MOV AL,carattere
AND AL,0Fh
MOV numero,AL
```

ESEMPIO 26 *Generazione di un valore casuale in base all'ora del Timer*

In questo esempio utilizziamo l'istruzione TEST che equivale in tutto e per tutto a un'istruzione AND, con la sola differenza che viene settato il registro dei flag, ma il risultato non viene posto nell'operando. Per simulare la generazione di un numero casuale si utilizza il seguente stratagemma: attraverso il servizio DOS 2Ch si ottiene l'ora di sistema, che viene memorizzata nel registro DH. Leggendo il bit meno significativo di tale registro (DH) è possibile ottenere un numero abbastanza casuale (0 o 1). A seconda del valore letto si stampa la stringa "Testa" oppure "Croce" attraverso il servizio 09h.

Il servizio DOS 09h consente di stampare una stringa che deve essere memorizzata in una variabile terminante con il simbolo del dollaro ('$'). Per dichiarare una stringa procediamo nel modo seguente:

```
.DATA
stringa         DB      'Salve Mondo',0Dh,0Ah,'$'
```

Come possiamo notare alla fine della stringa sono stati aggiunti i caratteri 10 e 13: si tratta di CR (*Carriage Return*) e LF (*Line Feed*), che servono per andare a capo dopo la scrittura della stringa stessa.
Per stampare tale stringa dobbiamo usare il servizio DOS 09h, formato da tre istruzioni:

```
LEA DX,stringa  ;inserisce nel registro DX l'indirizzo in cui è situata la
                ;stringa in memoria
MOV AH,09h      ;indicazione del servizio DOS richiesto, in questo caso 09
INT 21h         ;interrupt DOS che esegue il servizio
```

La prima istruzione consente di conoscere l'indirizzo della stringa (Offset) in memoria, che andrà posto nel registro DX; la seconda e la terza invece attivano il servizio di stampa.

Vediamo il codice completo dell'esempio proposto:

```
.DATA
stringa1 DB              'Testa',0Dh,0Ah,'$'
stringa2 DB              'Croce',0Dh,0Ah,'$'
.CODE
.STARTUP
         MOV AH,2Ch      ;servizio DOS 2Ch che genera in DH i centesimi di sec
                         ;dell'ora di sistema
         INT 21h
         TEST DH,01h     ;AND tra DH e 1 per mascherare il bit meno significativo
         JZ testa        ;se flag di zero settato allora testa
         LEA DX,stringa2 ;altrimenti croce, quindi assegnazione a DX dell'indirizzo
                         ;della stringa ('Croce')
         JMP dopo        ;salto per evitare conflitto di assegnazioni
testa:   LEA DX,stringa1 ;se flag di zero settato assegnazione in DX dell'indirizzo
                        di stringa1 ('testa')
```

```
dopo:    MOV AH,09h       ,servizio DOS di stampa stringa a video
         INT 21h
         MOV AH,4Ch       ;chiusura programma
         INT 21h
END
```

Prova adesso!

- Uso di cicli
- Uso dell'istruzione LEA
- Uso dell'istruzione AND
- Uso dei servizi DOS 2Ch e 09h

1 Modifica l'esempio 26 in modo tale che i valori casuali siano 4 (Quadri, Cuori, Fiori, Picche) a seconda del valore dei 2 bit meno significativi generati dal servizio DOS 2Ch che genera l'ora di sistema, mascherati con AND.

ESEMPIO 27 *Trasformazione da minuscolo a maiuscolo*

L'esempio seguente mostra come utilizzare una maschera di AND (BFh) per convertire un carattere da minuscolo a maiuscolo. Per fare questo è necessario usare la maschera 01001111 con il carattere per sottrarre 30h al valore originale. Il programma infine stampa a video il carattere ASCII con il servizio DOS 02h.

```
.DATA
carattere  DB   'a'
.CODE
           MOV AL,carattere  ;copio il carattere ASCII da convertire in AL (61h, cioè
                             ;01100001b)
           AND AL,1011111b   ;maschera di AND (BFh)
           MOV DL,AL         ;copio il carattere nel registro DL per la stampa a video
           MOV AH,02h        ;servizio DOS di stampa a video
           INT 21h
           MOV AH,4Ch        ;chiusura del programma
           INT 21h
END
```

La disgiunzione logica con l'istruzione OR

L'istruzione OR esegue l'or logico bit a bit tra il primo e il secondo operando, chiamato maschera, e copia il risultato nel primo operando. Il precedente contenuto del primo operando viene quindi sostituito dal risultato che contiene 1 nei bit che erano a 1 nel primo o nel secondo operando, mentre il secondo operando rimane inalterato. Il primo operando può essere un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato; il secondo operando può consistere in un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato o un dato immediato. I due operandi devono avere le stesse dimensioni, che possono essere di tipo Byte, Word o DWord.

Sintassi:

```
OR operando,maschera
```

Modalità di indirizzamento:
OR registro,immediato
OR registro,memoria
OR registro,immediato
OR memoria,immediato
OR memoria,registro

Flag alterati:
ZF, CF = 0, SF, OF, PF

ESEMPIO 28 *Mascheramento dei 4 bit più significativi di un numero per ottenere il carattere ASCII*

L'esempio che segue mostra come ottenere il carattere ASCII di un numero compreso tra 0 e 9. Tali valori rappresentano l'alternativa alla più classica addizione per 30h. Mascherando infatti i 2 bit meno significativi del nibble più significativo (30h) si ottiene il carattere ASCII del numero mediante un'istruzione OR. Consideriamo l'esempio seguente:

Numero da trasformare: 7
0 0 0 0 0 1 1 1

Maschera (30h)
0 0 1 1 0 0 0 0

OR tra numero (7) e maschera: 37h
0 0 1 1 0 1 1 1

```
.DATA
carattere  DB              0
numero     DB              7
.CODE
.STARTUP
           MOV AL,numero
           OR AL,30h
           MOV carattere,AL
```

ESEMPIO 29 *Trasformazione da maiuscolo a minuscolo*

L'esempio seguente mostra come utilizzare una maschera di OR (40h) per convertire un carattere da maiuscolo a minuscolo. Per fare questo è necessario usare la maschera 01000000 con il carattere per ottenere il relativo carattere minuscolo. Il programma infine stampa a video il carattere ASCII con servizio DOS 02h.

```
.DATA
carattere  DB              'A'
.CODE
           MOV AL,carattere  ;copio il carattere ASCII da convertire in AL (41h, cioè
                             ;01000001b)
           OR AL,01000000b   ;maschera di OR (40h)
           MOV DL,AL         ;copio il carattere nel registro DL per la stampa a video
           MOV AH,02h        ;servizio DOS di stampa a video
           INT 21h
           MOV AH,4Ch        ;chiusura del programma
           INT 21h
END
```

L'istruzione XOR consente di eseguire una Or esclusiva bit a bit tra il primo e il secondo operando copiando il risultato nel primo operando. Il precedente contenuto del primo operando viene quindi sostituito dal risultato che contiene 0 per ciascun bit uguale nei due operandi. Pertanto un'operazione di Or esclusivo su di uno stesso operando provoca un azzeramento dell'operando stesso. Il primo operando può essere un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato; il secondo operando può consistere in un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato o un dato immediato. I due operandi devono avere le stesse dimensioni, che possono essere di tipo Byte, Word o DWord. Il codice che segue mostra come utilizzare l'istruzione XOR per azzerare il contenuto di un registro:

```
XOR DX,DX   ;azzeramento del registro DX
```

Il codice che segue mostra come utilizzare l'istruzione XOR per verificare se due variabili contengono dati diversi:

```
MOV AL,variabile1
XOR AL,variabile2          ;se AL = 0 le due variabili sono uguali
```

La negazione logica con l'istruzione NOT

L'istruzione NOT effettua una negazione o complemento a 1 bit a bit dell'operando. Il risultato viene posto nell'operando, che può essere rappresentato da un registro, una cella di memoria indirizzata in modo diretto, indiretto o indicizzato. Negli ultimi due casi è necessario specificare la dimensione dell'operando che non può essere dedotta dall'istruzione.

Sintassi:

```
NOT operando
```

Flag alterati:
nessuno

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 L'istruzione SAL effettua:
- ○ uno shift a destra
- ○ uno shift a destra senza copiare il bit meno significativo nel carry flag
- ○ uno shift a sinistra
- ○ uno shift a sinistra senza copiare il bit più significativo nel carry flag

2 A che cosa equivale un'operazione di shift a destra?
- ○ a una divisione per 2
- ○ a una divisione per 10
- ○ a una moltiplicazione per 10
- ○ a una moltiplicazione per 2

3 Che cosa effettua il seguente frammento di codice?

```
     MOV AX,n
     SAL AX,1
     JNC qua
     MOV DL,'p'
     MOV AH,02h
     INT 21h
     JMP fine
qua: MOV DL,'n'
     MOV AH,02h
     INT 21h
fine: ...
```

- ○ se il numero presente in AX è pari stampa 'n', se dispari stampa 'p'
- ○ se il numero presente in AX è pari stampa 'p', se dispari stampa 'n'
- ○ se il numero presente in AX è negativo stampa 'p', se positivo stampa 'n'
- ○ se il numero presente in AX è positivo stampa 'p', se negativo stampa 'n'

4 È possibile usare un'istruzione di scorrimento per: (due risposte)
- ○ verificare se un numero è uguale a zero
- ○ verificare se un numero è divisibile per 2
- ○ verificare se un numero è divisibile per 3
- ○ verificare se un numero in complemento a due è positivo o negativo
- ○ verificare se un numero è maggiore di 2

5 Le istruzioni di rotazione rispetto alle istruzioni di scorrimento:
- ○ non sono distruttive, infatti consentono di recuperare il numero iniziale
- ○ sono distruttive, infatti non consentono di recuperare il numero iniziale
- ○ consentono sempre di dividere un valore per 2

6 Che cosa contiene il registro AL al termine del frammento di codice seguente?

```
MOV BL,0FEh
MOV AL,03h
ADD BL,02h
RCL AL,1
```

- ○ 5 ○ 6
- ○ 7 ○ 8

7 Che cosa contiene il registro AL al termine del frammento di codice seguente?

```
MOV BL,0FEh
MOV AL,0F0h
ADD BL,02h
RCR AL,1
```

- ○ 0Eh ○ F7h
- ○ 08h ○ F8h

8 Che cosa contiene il registro AL al termine del frammento di codice seguente?

```
MOV AL,0F1h
MOV DL,03h
SAR DL,1
ROR AL,1
```

- ○ F9h ○ 09h
- ○ F8h ○ 0Eh

9 Che cosa contiene il registro AX al termine del frammento di codice seguente?

```
MOV BL,01h
MOV AX,00FFh
SAR BL,1
RCL AX,1
```

- ○ 01FFh ○ 02FEh
- ○ 0FFh ○ 01FEh

10 Che cosa contengono i registri AX e BX al termine del frammento di codice seguente?

```
MOV BX,0FE1h
MOV AX,0Fh
SAL BL,1
RCR AX,1
```

- ○ AX=8008h BX=0FC3h ○ AX=8007h BX=0FC2h
- ○ AX=8006h BX=0FC1h ○ AX=8007h BX=0FC1h

Verifichiamo le competenze

Esprimi la tua creatività

Esegui i seguenti esercizi in assembly.

1. Scrivi un programma assembly in grado di convertire un carattere ASCII (lettera minuscola) inserito da tastiera in una lettera maiuscola usando la maschera logica.
2. Come l'esercizio precedente ma eseguendo l'inserimento di 10 caratteri da tastiera.
3. Scrivi un programma assembly in grado di convertire un carattere ASCII (lettera maiuscola) inserito da tastiera in una lettera minuscola, effettuando il controllo sulla correttezza durante l'inserimento. Se il carattere inserito è già minuscola ripeti l'inserimento.
4. Data una tabella di caratteri ASCII che inizia dalla cella di indirizzo 0100h, di dimensione "n", scrivi un programma assembly in grado di contare quanti caratteri maiuscoli vi sono.
5. Data una tabella di numeri a 16 bit che inizia dalla cella di indirizzo 0100h, di dimensione "n", scrivi un programma assembly in grado di contare quanti numeri negativi vi sono.
6. Data una tabella di numeri che inizia dalla cella di indirizzo 0100h, di dimensione "n", scrivi un programma assembly in grado di contare quanti numeri pari e dispari vi sono inserendo il risultato nelle celle successive alla fine della tabella.
7. Dato un numero a 4 byte inserito da tastiera, che rappresenta una classe di indirizzo IP, scrivi un programma assembly che verifichi a quale classe appartiene utilizzando la tecnica del mascheramento, sapendo che i bit più significativi del primo byte sono:

   CLASSE A |0|
   CLASSE B |10|
   CLASSE C |110|
   CLASSE D |1110|
   CLASSE E |1111|

   Il programma deve stampare a video una stringa che specifichi la classe usando il servizio DOS 09h.
8. Scrivi un programma assembly in grado giocare con il computer. Il programma genera un numero da 1 a 4, l'utente deve indovinare il numero inventato dal programma. Poi inserisce il numero con il servizio DOS di lettura e il sistema comunica una stringa, mediante il servizio DOS 09h, che dice se l'utente ha vinto oppure no.
9. Data una tabella a partire dalla cella di indirizzo 0200h, di dimensione pari a "n", scrivi un programma che mascheri con AND a 0Fh tutti i numeri della tabella.
10. Data una tabella a partire dalla cella di indirizzo 0100h, di dimensione pari a "n", scrivi un programma che mascheri con OR a F1h tutti i numeri della tabella.

UNITÀ DIDATTICA 9

LE PROCEDURE ASSEMBLY

IN QUESTA UNITÀ IMPAREREMO...

- a saper gestire le procedure in assembly
- ad applicare le procedure a diverse situazioni operative
- a utilizzare il passaggio dei parametri in assembly

La definizione delle procedure

PROCEDURE

Le procedure sono un insieme di istruzioni che possono essere richiamate in altri punti del programma. Vengono anche chiamate routine o sottoprogrammi.

Per dichiarare una procedura è necessario racchiuderne il codice tra le parole chiave seguenti:

```
nome    PROC    tipo      ;inizio procedura
        ...               ;corpo della procedura
        Istruzioni
        ...
        RET               ;ritorno all'istruzione successiva alla chiamata
nome    ENDP
```

Il tipo identifica il metodo di chiamata, che può essere di tipo NEAR oppure di tipo FAR. Il metodo NEAR indica che la chiamata avviene all'interno dello stesso segmento, mentre FAR indica che la chiamata può avvenire da istruzioni presenti in altri segmenti.
Se la procedura è di tipo NEAR l'istruzione di ritorno dovrà effettuare solo il prelievo di IP dallo stack, mentre se è di tipo FAR dovrà prelevare anche CS. Se non specifichiamo nulla la chiamata è di tipo NEAR.

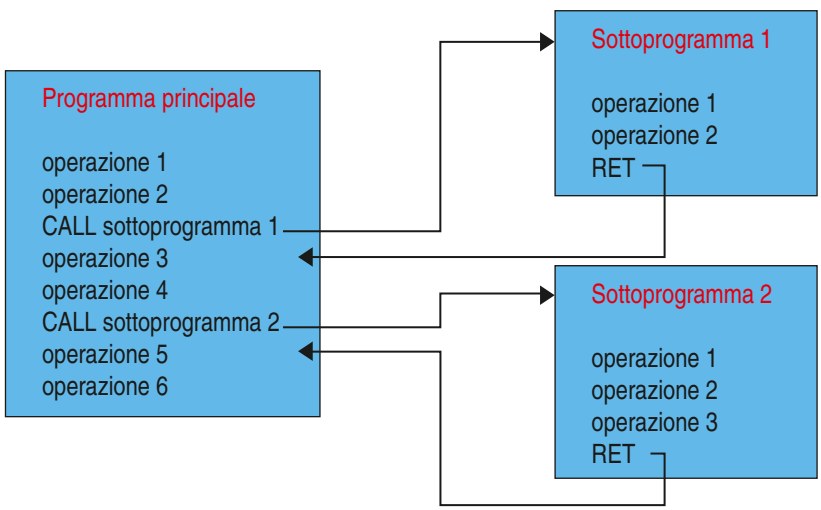
L'istruzione RET restituisce il controllo all'istruzione successiva a quella che ha effettuato il salto alla procedura, prelevando dallo stack l'indirizzo di ritorno che viene collocato nei registri CS e IP se la procedura è di tipo FAR, soltanto IP se la procedura è di tipo NEAR.

È importante sottolineare che una procedura deve essere dichiarata in una zona del codice che non viene eseguita durante il normale svolgimento del programma. Tale sezione esiste in modo esplicito nei linguaggi evoluti mentre l'assembly non la prevede: per questo motivo le procedure devono essere definite tra la chiamata del sistema operativo per la chiusura del programma (servizio DOS 4Ch) e l'istruzione END:

```
.CODE
.STARTUP
        ...
        Istruzioni
        ...
        MOV AH,4Ch
        INT 21h
        ...             ;sezione per le procedure
        nome  PROC

        nome  ENDP
        ...
        END
```

La chiamata alle procedure

L'istruzione CALL effettua un salto incondizionato all'etichetta indicata che identifica una procedura. Prima che avvenga il salto l'indirizzo dell'istruzione successiva alla chiamata viene salvato nella cima dello stack, mentre il registro SP viene decrementato di 4. Se la chiamata è di tipo FAR viene anche salvato il contenuto del registro di segmento CS: in questo caso SP verrà decrementato di 6. L'indirizzo dell'etichetta che rappresenta la procedura viene copiato nel registro IP per passare così il controllo alla prima istruzione del sottoprogramma. Al termine dell'esecuzione della procedura l'istruzione RET estrae dalla cima dello stack l'indirizzo di ritorno copiandolo nel registro IP, incrementando quindi il registro puntatore della cima dello stack di 4. Se la procedura è di tipo FAR l'istruzione RET estrae anche il valore di CS salvato precedentemente, incrementando pertanto il registro SP di 6. In questo modo possiamo anche richiamare una procedura quando siamo all'interno di un'altra procedura; la cima dello stack conterrà sempre l'indirizzo di rientro alla procedura chiamante.



In assembly non esiste una distinzione tra procedure e funzioni, che vengono chiamate sempre procedure; è infatti compito del programmatore costruirle in modo tale che realizzino una procedura o una funzione.

L'esempio seguente mostra come richiamare una procedura da un punto qualsiasi del programma; come di può vedere, dopo la parola chiave CALL viene indicata l'etichetta che indica il nome della procedura:

```
.CODE
.STARTUP
        ...
        CALL proce     ;chiamata procedura con conseguente salvataggio di IP nello stack
        ...
        MOV AH,4Ch
        INT 21h
proce   PROC           ;inizio procedura
        ...
        RET            ;uscita dalla procedura con ripristino di IP dallo stack
proce   ENDP
END
```

All'ingresso di una procedura è buona norma salvare nello stack tutti i registri utilizzati, in modo che all'uscita della stessa vengano ripristinati al valore corrente prima della chiamata. In alternativa si possono utilizzare le istruzioni PUSHA e POPA che salvano e ripristinano tutti i registri. La regola da rispettare è che le procedure non devono alterare il contesto attuale del programma, cioè lo stato delle variabili e dei registri usati nel programma chiamante.

ESEMPIO 30 *Lettura e stampa di un numero a 16 bit*

In questo esempio vengono usate due procedure, una per la lettura di un numero a 16 bit, chiamata leggi, e l'altra per la stampa a video di un numero a 16 bit. Alla prima procedura viene passato il numero da leggere attraverso il registro BX. Il registro BX viene restituito modificato dalla prima procedura al programma principale. Lo stesso registro (BX) viene usato come parametro per passare alla procedura scrivi il valore da stampare a video. Il programma principale dopo aver ricevuto il numero da tastiera lo divide per due e quindi lo passa alla seconda procedura per la stampa a video.

```
.DATA
numero    DW             0h
stringa   DB             'inserisci un numero da dividere per due','$'
due       DW             02h
temp      DW             0h
.CODE
.STARTUP
          MOV AH,09h     ;stampa a video stringa per l'utente
          LEA DX,stringa ;con servizio DOS 09h
          INT 21h
          CALL leggi     ;chiamata procedura che legge il numero da tastiera in BX
          MOV AX,BX      ;copia di BX in AX per la divisione per due
          MOV DX,0h      ;azzeramento registro DX per divisione a 16 bit
          DIV due        ;divide AX per 2
          MOV BX,AX      ;inserimento quoziente in BX
```

```
          CALL scrivi     ;chiamata procedura che stampa a video il numero passato
                          ;tramite BX
          MOV AH,4Ch      ;chiusura programma principale
          INT 21h

;sezione delle procedure

;procedura di lettura numero da tastiera
leggi     PROC
          PUSHA           ;salvataggio di tutti i registri
          MOV BX,0        ;azzero registro di accumulo della cifra a 16 bit
ciclo:    MOV AH,01h      ;servizio DOS di lettura da tastiera
          INT 21h
          CMP AL,0Dh      ;il carattere ASCII letto viene confrontato con INVIO
                          ;(carattere 13)
          JE fine         ;se INVIO premuto la cifra è terminata salto a fine
          PUSH AX         ;salvo il numero letto nello stack
          MOV AX,BX       ;moltiplico per 10 il numero accumulato in BX
          MUL dieci       ;AX * 10
          MOV BX,AX       ;colloco in BX il risultato della moltiplicazione per 10
          POP AX          ;riprendiamo in AX l'ultima cifra letta
          SUB AL,30h      ;la cifra viene aggiustata da ASCII a numero sottraendo 30h
          MOV AH,0        ;azzero parte alta di AX
          ADD BX,AX       ;somma del valore di accumulato in BX con l'ultima cifra letta
          JMP ciclo       ;salto incondizionato al ciclo di lettura da tastiera
fine:     MOV temp,BX     ;salvataggio del registro da restituire in uscita nella
                          ;variabile temporanea
          POPA            ;ripristino di tutti i registri
          MOV BX,temp     ;al registro BX viene di nuovo copiato il valore da restituire
                          ;salvato nella
                          ;istruzione precedente alla POPA nella variabile temp
          RET             ;uscita dalla procedura e ritorno a programma chiamante
leggi ENDP                ;fine procedura leggi

;procedura di stampa a video
scrivi    PROC
          PUSHA           ;salvataggio di tutti i registri
          MOV AX,BX       ;poniamo in AX il numero da stampare prelevandolo da BX
          MOV CL,00       ;azzeramento contatore di cifre
sta:      MOV DX,0        ;azzeramento parte alta della cifra a 32 bit che viene divisa
                          ;per 10
          DIV dieci       ;divisione a 16 bit per fare DX:AX/10. Resto in DX. Quoziente
                          ;in AX
          PUSH DX         ;salvataggio del resto nello stack
          INC CL          ;aggiornamento contatore cifre
          CMP AX,0        ;verifica se quoziente = 0
          JNE sta         ;salta a etichetta sta se quoziente <> 0
rip:      POP DX          ;recupero resto da stampare
          ADD DL,30h      ;aggiustamento cifra come carattere ASCII sommando 48 (30h)
          MOV AH,02h      ;servizio DOS di stampa a video
          INT 21h
          DEC CL          ;decremento contatore cifre
          JNZ rip         ;se diverso da zero ripeto la stampa di un'altra cifra
          POPA            ;ripristino di tutti i registri
          RET             ;uscita dalla procedura e ritorno a programma chiamante
scrivi    ENDP            ;fine procedura scrivi
```

Prova adesso!

- Uso di procedure
- Uso dei servizi DOS 01h, 02h e 09h

1. Modifica l'esempio 30 aggiungendo una procedura che stampi la stringa.
2. Modifica l'esempio 30 aggiungendo la stampa del quoziente separato dal resto della divisione con una stringa che indichi il tipo di dato stampato.

Il passaggio dei parametri

Il principio di funzionamento di una procedura prevede che possa essere utilizzata più volte nello stesso programma o richiamata da altri programmi per formare quella che viene generalmente chiamata una libreria di procedure. I parametri passati a una procedura la rendono svincolata dal contesto nel quale si trova. *Possiamo paragonare una procedura a una grande scatola contenente un esecutore al quale passiamo una serie di dati scritti su un foglietto attraverso una fessura di ingresso; l'esecutore effettua i calcoli e ci restituisce il foglietto con i risultati tramite la fessura di uscita.* La scatola (o l'oggetto in questione) può essere utilizzata da chiunque, a patto che inserisca i dati nel modo corretto. Dobbiamo ricordarci del concetto appena enunciato ogniqualvolta definiamo una nuova procedura, cercando quindi il più possibile di renderla svincolata dal contesto.

L'uso dei parametri svincola una procedura dal contesto e ne accresce l'utilità, in quanto sarà possibile utilizzarla in qualunque programma variando i valori dei parametri in ingresso.
Il passaggio dei parametri può avvenire secondo due metodi:

- usando i registri;
- usando lo stack.

Il primo metodo è adatto a situazioni nelle quali i parametri sono pochi; infatti, non solo i registri della CPU sono in numero ridotto, ma dobbiamo sempre tenere conto della necessità di limitarne l'impiego allo stretto indispensabile per non rischiare di ridurne l'utilità. Inoltre bisogna ricordarsi di assegnare al registro il valore appena precedente alla chiamata della procedura. Nell'esempio 30 è stato usato proprio questo metodo: per esempio il numero da leggere è stato scritto nel registro BX, prima dell'istruzione RET.

> Quando una procedura restituisce un parametro tramite un registro deve collocarne il valore dopo aver ripristinato i registri mediante l'istruzione POPA e prima dell'uscita con RET: l'istruzione POPA infatti ripristina tutti i registri al valore iniziale.

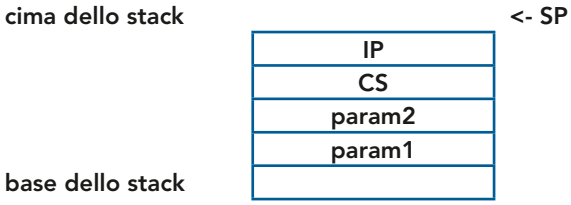
Il metodo che si serve dello stack per il passaggio dei parametri consente di passare un numero di parametri assai elevato e non pone limitazioni al programmatore nell'uso dei registri. Chiameremo parametri in input quelli passati alla procedura e parametri di output quelli restituiti dalla procedura stessa.

I parametri da passare alla procedura (parametri di input) devono essere caricati nello stack prima della chiamata della procedura stessa. Se una procedura ammette parametri di output dobbiamo necessariamente riservare dello spazio nello stack prima di chiamare la procedura. Al termine della procedura il programma chiamante deve estrarre dallo stack tutti i valori inseriti prima della chiamata.

Dobbiamo tenere presente che se anche il parametro è lungo soltanto un byte bisogna codificarlo su 16 bit perché sia possibile salvarlo nello stack. Per fare questo si associa il valore alla parte bassa di un registro o di una cella di memoria, azzerandone la parte alta.

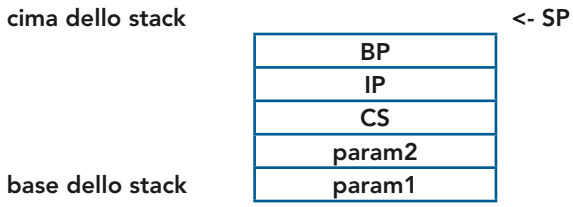
Come abbiamo visto al momento della chiamata della procedura, con l'istruzione CALL vengono salvati nello stack gli indirizzi che consentono alla CPU di riprendere a eseguire il programma all'istruzione successiva all'ultima CALL. Tali valori vengono copiati nello stack dopo i parametri passati e pertanto si posizionano in cima allo stack. Vediamo che cosa contiene lo stack in seguito a una chiamata di tipo FAR con due parametri passati:

```
PUSH param1              ;primo parametro nello stack
PUSH param2              ;secondo parametro nello stack
CALL procedura    FAR    ;chiamata alla procedura
```



Per estrarre i parametri dallo stack la procedura non può usare l'istruzione POP, che andrebbe a creare conflitti di sistema. Per accedere ai parametri direttamente nello stack, la procedura sfrutta l'indirizzamento base + spiazzamento, dove la base è rappresentata dal registro BP che finora non abbiamo mai usato.
Per fare questo all'inizio della procedura salviamo il contenuto precedente di BP nello stack (PUSH BP) e quindi associamo al registro BP il puntatore alla cima dello stack (MOV BP,SP). In riferimento all'esempio precedente, al termine di queste due istruzioni, lo stack contiene i seguenti dati:

```
procedura  PROC    FAR
           PUSH BP
           MOV BP,SP
           ...
```



Per indirizzare il primo parametro dobbiamo sommare 6 (4 nel caso di procedura di tipo NEAR) al contenuto di BP. Nell'esempio il secondo parametro viene copiato nel registro AX, mentre il primo nel registro BX:

```
procedura    PROC    FAR
             PUSH BP          ;salvataggio vecchio BP
             MOV BP,SP        ;adesso il registro BP punta alla cima dello stack
             PUSHA            ;salvataggio di tutti i registri
```

```
            MOV AX,[BP + 6]  ;copia del secondo parametro in AX
            MOV BX,[BP + 8]  ;copia del primo parametro in BX
            ...
            POPA             ;ripristino di tutti i registri
            POP BP           ;ripristino registro BP
            RET              ;fine procedura salto a CS e IP e liberazione stack
procedura   ENDP
```

Come abbiamo visto, l'accesso ai parametri successivi avviene incrementando lo spiazzamento, [BP + 6] per l'ultimo, [BP + 8] per il penultimo e così via.

Per evitare di dover usare in ogni istruzione il riferimento all'indirizzo dello stack, possiamo utilizzare l'istruzione EQU che permette di associare un nome simbolico a un'espressione:

```
nome EQU espressione
```

SI può usare EQU per assegnare un nome simbolico all'espressione che individua l'indirizzo di ciascun parametro per poi accedere ai parametri stessi tramite il nome associato:

```
procedura   PROC
            PUSH BP
            MOV BP,SP
            parametro EQU [BP+6]
            ..
            MOV AX,parametro        ;equivale a MOV AX,[BP + 6]
            ...
```

ESEMPIO 31 *Procedura che somma due numeri*

In questo esempio viene usata una procedura che prevede che nello stack vengono passati due parametri da sommare, e un terzo parametro che conterrà il risultato. L'esempio simula un passaggio di parametri per riferimento: infatti, all'uscita della procedura, il terzo parametro verrà modificato nel contenuto in quanto conterrà il risultato della somma.

```
.DATA
risultato   DW   0h
numero1     DB   100h
numero2     DB   78h
.CODE
.STARTUP
            MOV BX,0         ;passo nello stack il primo parametro che conterrà
                             ;il risultato
                             ;(il registro può essere uno qualunque a 16 bit
            PUSH BX          ;serve solo per assegnare zero al primo parametro nello stack)
            MOV AL,numero1   ;copia nella parte bassa di AX il primo parametro
            MOV AH,0h        ;azzero la parte alta di AX
            PUSH AX          ;passo nello stack il secondo parametro (il primo numero
                             ;da sommare)
            MOV AL,numero2   ;ripeto le stesse operazioni per il secondo parametro
            MOV AH,0h        ;azzero la parte alta di AX
            PUSH AX          ;passo nello stack il terzo parametro (il secondo numero
                             ;da sommare)
```

```
          CALL addizione   ;chiamata NEAR della procedura con salvataggio dello stack di IP
          POP AX           ;recupero primo parametro non modificato
          POP AX           ;recupero secondo parametro non modificato
          POP tot          ;recupero terzo parametro (risultato della somma) modificato
          MOV AH,4Ch
          INT 21h
;inizio procedura
addizione PROC
          PUSH BP          ;salvataggio vecchio valore di BP nello stack
          MOV BP,SP        ;assegnazione a BP della cima dello stack
          PUSHA            ;salvataggio di tutti i registri
          n3 EQU [BP + 4]  ;assegnazione nome simbolico al terzo parametro
          n2 EQU [BP + 6]  ;assegnazione nome simbolico al secondo parametro
          n1 EQU [BP + 8]  ;assegnazione nome simbolico al primo parametro
          MOV AX,n2        ;copio il primo parametro in AX per fare la somma a 16 bit
          ADD AX,n1        ;somma tra primo parametro (AX) e il secondo parametro,
                           ;risultato in AX
          MOV n3,AX        ;trasferimento del risultato nel terzo parametro
          POPA             ;ripristino registri prima di uscita
          POP BP           ;ripristino di BP
          RET              ;ritorno a programma chiamante
addizione ENDP
```

Prova adesso!

- Uso di procedure
- Passaggio dei parametri
- Indirizzamento diretto nello stack

1. Modifica l'esempio 31 aggiungendo una procedura che stampi il risultato passato usando lo stack.
2. Modifica l'esempio 31 aggiungendo l'inserimento dei numeri da sommare mediante una procedura alla quale vengono passati i parametri per riferimento usando lo stack.

ESEMPIO 32 *Procedura che legge un vettore*

In questo esempio viene usata una procedura alla quale viene passato nello stack un indirizzo di memoria e una dimensione di un vettore da caricare di numeri a 8 bit da tastiera. L'esempio simula un passaggio di parametri per riferimento in quanto al termine della procedura il vettore di elementi che è stato passato risulta modificato.

```
.DATA
dimensione DB 64h
dieci      DB 0Ah
.CODE
.STARTUP
           MOV DL,dimensione     ;nel primo parametro inseriamo la dimensione
                                 ;del vettore
           ;(il registro può essere uno qualunque a 16 bit)
           MOV DH,0H             ;serve solo per azzerare la parte alta di DX
           PUSH DX               ;copio il primo parametro nello stack (dimensione)
           MOV DX,0100h          ;copio in DX l'indirizzo iniziale del vettore in memoria
```

```
          PUSH DX                 ;passo nello stack il secondo parametro (indirizzo
                                  ;vettore in memoria)
          CALL carica             ;chiamata NEAR della procedura con salvataggio dello
                                  ;stack di IP
          POP AX                  ;recupero primo parametro non modificato
          POP AX                  ;recupero secondo parametro non modificato
          MOV AH,4Ch
          INT 21h
;inizio procedura
carica    PROC
          PUSH BP                 ;salvataggio vecchio valore di BP nello stack
          MOV BP,SP               ;assegnazione a BP della cima dello stack
          PUSHA                   ;salvataggio di tutti i registri
          indirizzo EQU [BP +4 ]  ;assegnazione nome simbolico al primo parametro
                                  ;(indirizzo)
          dimensione EQU [BP + 6] ;assegnazione nome simbolico al secondo parametro
                                  ;(dimensione)
          MOV SI,indirizzo        ;il registro SI punta alla prima locazione del vettore
          MOV CX,dimensione       ;il registro contatore contiene il numero di celle
                                  ;da leggere
ciclo:    MOV AH,01h              ;servizio DOS di lettura da tastiera
          INT 21h
          CMP AL,0Dh              ;il carattere ASCII letto viene confrontato con INVIO
                                  ;(carattere 13)
          JE fine                 ;se INVIO premuto la cifra è terminata salto a fine
          PUSH AX                 ;salvo il numero letto nello stack
          MOV AL,BL               ;moltiplico per 10 il numero accumulato in BL
          MUL dieci               ;AL * 10
          MOV BL,AL               ;colloco in BL il risultato della moltiplicazione per 10
          POP AX                  ;riprendiamo in AX l'ultima cifra letta
          SUB AL,30h              ;la cifra viene aggiustata da ASCII a numero
                                  ;sottraendo 30h
          ADD BL,AL               ;somma del valore di accumulato in BL con l'ultima
                                  ;cifra letta
          JMP ciclo               ;salto incondizionato al ciclo di lettura da tastiera
fine:     MOV[SI],BL              ;scrittura nel vettore del nuovo numero
          INC SI                  ;incremento SI per passare alla cella del vettore
                                  ;successiva
          MOV BL,0h               ;azzero BL per accumulare cifre nuovo numero
          LOOP ciclo              ;decremento CX e se CX <>0 torno a ciclo
          POPA                    ;ripristino registri prima di uscita
          POP BP                  ;ripristino di BP
          RET                     ;ritorno a programma chiamante
addizione ENDP
```

Prova adesso!

- Uso di procedure
- Passaggio dei parametri
- Indirizzamento diretto nello stack

1 Modifica l'esempio 32 aggiungendo una procedura che stampi il vettore passandone l'indirizzo iniziale e la dimensione mediante lo stack.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Quale istruzione tra le seguenti chiude una procedura riportando l'esecuzione alla riga successiva alla chiamata?**

- ○ ENDP
- ○ RET
- ○ END
- ○ POP BP

2 **Nelle chiamate a procedure di tipo FAR l'assemblatore memorizza:**

- ○ i registri CS e IP
- ○ i registri DS e SS
- ○ il solo registro IP
- ○ il solo registro CS
- ○ i registri CS, IP e FLAGS

3 **In quale punto del programma è conveniente dichiarare le procedure?**

- ○ nella sezione identificata dalla direttiva .CODE prima di .STARTUP
- ○ nella sezione identificata dalla direttiva .CODE prima di END
- ○ nella sezione identificata dalla direttiva .CODE prima di MOV AH,4Ch
- ○ nella sezione identificata dalla direttiva .DATA

4 **Associa l'istruzione CALL proc al frammento di codice corretto tra i seguenti:**

○
```
PUSH IP
JMP proc
```

○
```
PUSH BP
PUSH IP
JMP proc
```

○
```
PUSH IP
JNC proc
```

○
```
PUSH IP
JZ proc
```

5 **Indica quale tipo di passaggio parametri è ammesso in linguaggio assembly: (due risposte)**

- ○ tramite stack
- ○ per registri
- ○ tramite variabili
- ○ tramite dato immediato

6 **Prima di chiamare una procedura di tipo NEAR vengono salvati nello stack i seguenti parametri:**

```
PUSH param1
PUSH param2
CALL proc
```

Indica tra le seguenti quale risposta è quella che consente di estrarre in AX dallo stack il secondo parametro (param2):

- ○ MOV AX,[BP+4]
- ○ Ripetendo due volte POP AX
- ○ MOV AX,[BP+6]
- ○ MOV AX,[BP+8]

7 **Quale tra le seguenti istruzioni deve essere inserito all'inizio di una procedura che ha la necessità di estrarre i parametri dallo stack:**

○
```
PUSH BP
MOV BP,SP
```

○
```
PUSHA
```

○
```
PUSH BP
MOV BP,SP
```

○
```
POP BP
MOV BP,SP
```

8 **Come può essere passato un vettore di 100 elementi a una procedura:**

- ○ per valore, inserendo 100 elementi nello stack
- ○ per riferimento, inserendo nello stack l'indirizzo della prima cella del vettore e la dimensione dello stesso
- ○ per riferimento, inserendo nello stack solo l'indirizzo della prima cella del vettore
- ○ per registro passando l'indirizzo della prima cella del vettore nel registro SI

9 **Che cosa contengono i registri BX e AX al termine della procedura indicata?**

```
MOV AX,03h
MOV BX,05h
PUSH AX
PUSH BX
CALL sottoprog
...
sottoprog   PROC
            PUSH BP
            MOV BP,SP
            MOV CX,[BP+4]
            MOV DX,[BP+6]
            ADD CX,DX
            POP BP
            RET
sottoprog   ENDP
```

- ○ AX=3, BX=5
- ○ AX=8, BX=5
- ○ AX=3, BX=8
- ○ AX=3, BX=3

Verifichiamo le competenze

Esprimi la tua creatività

Esegui i seguenti esercizi in assembly.

1. Scrivi un programma assembler che, data una tabella di 100 elementi memorizzata a partire dalla cella di indirizzo 0200h, effettui una copia di tali elementi nella tabella posta a partire dalla cella di indirizzo 0300h usando una procedura che copi gli elementi passati uno a uno come parametri.
2. Scrivi un programma assembly che data una tabella di numeri in memoria a partire dall'indirizzo 0100h costruisca a partire dall'indirizzo 0200h la tabella dei soli numeri pari usando una procedura.
3. Scrivi un programma assembly che data una tabella di numeri in memoria a partire dall'indirizzo 0100h costruisca a partire dall'indirizzo 0200h la tabella dei soli numeri negativi usando una procedura.
4. Scrivi un programma assembly che data una tabella di numeri in memoria a partire dall'indirizzo 0100h costruisca a partire dall'indirizzo passato come parametro la tabella dei soli numeri pari usando una procedura.
5. Scrivi un programma assembly che data una tabella di numeri in memoria a partire dall'indirizzo 0100h ne calcoli la media aritmetica e la restituisca per registro usando una procedura.
6. Scrivi un programma assembly che data una tabella di numeri in memoria a partire dall'indirizzo 0100h passi a una procedura l'indirizzo iniziale e la lunghezza di tale tabella, quindi proceda a ordinare gli elementi usando un algoritmo conosciuto.
7. Scrivi un programma assembly che data una tabella di numeri in memoria a partire dall'indirizzo 0100h passi a una procedura la dimensione, l'indirizzo di inizio della tabella e un numero da cercare e restituisca per registro il numero di occorrenze trovate.
8. Scrivi un programma assembly che data una tabella di numeri in memoria a partire dall'indirizzo 0100h costruisca a partire dall'indirizzo 0200h la tabella dei valori divisi per il valore passato alla procedura solo se compresi tra 100 e 200.
9. Scrivi un programma assembly che data una tabella di numeri in memoria a partire dall'indirizzo 0100h costruisca a partire dall'indirizzo 0200h la tabella dei valori moltiplicati per il valore passato alla procedura solo se compresi tra 10 e 20.
10. Scrivi un programma assembly che utilizzi due procedure, una per ricevere una serie di valori da inserire in una tabella, e una che stampi i numeri presenti in una tabella. Una terza tabella deve dividere tutti i numeri per 2 se maggiori di 150, mentre moltiplicare per due tutti i numeri maggiori o uguali a 150.
11. Scrivi un programma assembly che utilizzi una procedura in grado di leggere un nome da tastiera e un'altra che determini se la tabella di caratteri ASCII è palindroma o meno.
12. Scrivi un programma assembly che utilizzi una procedura in grado di mostrare a video i primi 20 numeri primi.
13. Scrivi un programma assembly in grado di far giocare un utente a testa o croce. Una procedura deve generare il valore casuale, utilizzando il servizio DOS 2Ch. Il giocatore a ogni vittoria raddoppia il capitale e a ogni sconfitta dimezza il capitale.
14. Scrivi un programma assembly che data una tabella di numeri in memoria a partire dall'indirizzo 0100h ne calcoli il minimo e il massimo restituendoli come parametri usando lo stack mediante una procedura.
15. Scrivi un programma assembly che data una tabella di numeri in memoria a partire dall'indirizzo 0100h ne restituisca i soli numeri primi come parametri usando lo stack mediante una procedura.
16. Scrivi un programma assembly che utilizzi una procedura in grado di generare in una tabella restituita tramite lo stack (in realtà indirizzo iniziale e dimensione) le sequenza dei primi quattro numeri perfetti.

    Un numero si dice perfetto quando è uguale alla somma di tutti i suoi divisori escluso sé stesso. Per esempio, il numero 496, divisibile per 1, 2, 4, 8, 16, 31, 62, 124 e 248 è un numero perfetto:

    496 = 1, 2, 4, 8, 16, 31, 62, 124 e 248

    Lo stesso vale per il 6 che è divisibile per 1, 2 e 3.

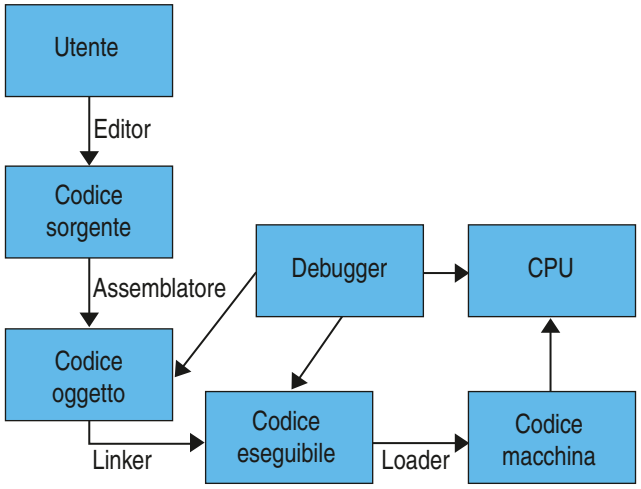
    6 = 1 + 2 + 3

ESERCITAZIONI DI LABORATORIO 1

USARE L'AMBIENTE TURBO ASSEMBLER

Il ciclo di vita di un programma

Il ciclo di vita di un programma assembly può essere così schematizzato:



Come possiamo notare la CPU svolge un ruolo effettivo essendo a diretto contatto con l'esecuzione del Codice macchina. Inoltre l'esecuzione del debug del programma è strettamente legata alla struttura e all'architettura fisica del processore, come vedremo in seguito: il Turbo Debugger, lo strumento usato da noi per effettuare il debug del programma, dispone infatti gli elementi della CPU rivolti al programmatore (Registri e Flag).

Le fasi della programmazione in assembly secondo la sintassi TASM sono le seguenti:

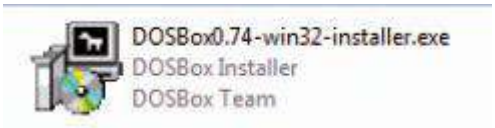
1. scrittura del codice sorgente con un Editor scelto tra i tanti disponibili in commercio (da Notepad++ a Scite);
2. salvataggio del file sorgente con estensione .asm;
3. assemblaggio del file sorgente con Turbo Assembler (TASM.exe);
4. linking del file oggetto con Turbo Linker (TLINK.exe);
5. verifica del codice eseguibile con Turbo Debugger (TD.exe) o in alternativa esecuzione.

Turbo Assembler (TASM) è un ambiente gratuito che consente l'assemblaggio, il linking e infine il debug di un sorgente in linguaggio assembly. Tuttavia, perché possa funzionare con sistemi operativi superiori a Windows XP è necessario usare degli emulatori MS-DOS, come per esempio DOSBox.

Installare DosBox

DOSBox è un emulatore che ricrea una macchina virtuale per MS-DOS compatibile e completa, in grado di gestire le schede video e audio, la grafica e anche un'interfaccia di rete di base. Grazie all'emulatore è possibile eseguire programmi per MS-DOS senza doverli modificare. Innanzitutto occorre scaricare DosBox: in questo caso utilizziamo la versione 0.74 per Windows sia a 32 bit che a 64 bit, disponibile all'indirizzo web http://www.dosbox.com/wiki/Releases.

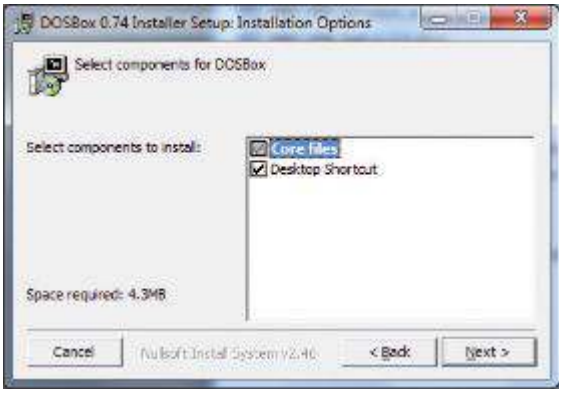
La procedura riportata a lato illustra come installare il software DosBox. ▶



❶ Dopo aver scaricato il software apposito, fai doppio clic sul file di installazione e successivamente, nella finestra che appare, fai clic su Esegui: ▶

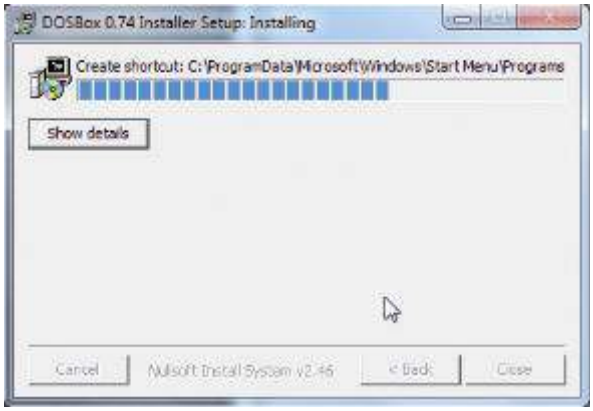
❷ Leggi il contratto di licenza e fai clic su Next per sottoscriverlo: ▼

❸ La finestra a lato richiede se si desidera collocare sul desktop un collegamento a DOSBox; è consigliabile fare clic su Next lasciando attivate le opzioni proposte, contrassegnate dai segni di spunta: ▶



❹ In questo caso la directory di installazione è quella indicata per un sistema a 64 bit. Fai clic su Installa per procedere alla copia dei file necessari all'esecuzione: ▶

5 Ha inizio la fase d'installazione. A operazione ultimata fai clic su Close: ▶

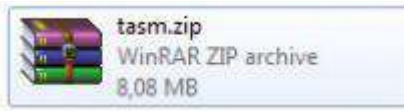


6 Per utilizzare DOSBox dovrai fare doppio clic sull'icona di collegamento che è stata creata sul desktop: ▶

Installare e usare Turbo Assembler (TASM)

Il linguaggio assembly possiede diversi assemblatori, disponibili quasi tutti gratuitamente (VASM, BASM, TAS, MASM ecc.). In questo volume i codici presentati sono scritti con la codifica più diffusa, TASM (Turbo Assembler). La procedura riportata di seguito descriveillustra come installare e usare questo assemblatore.

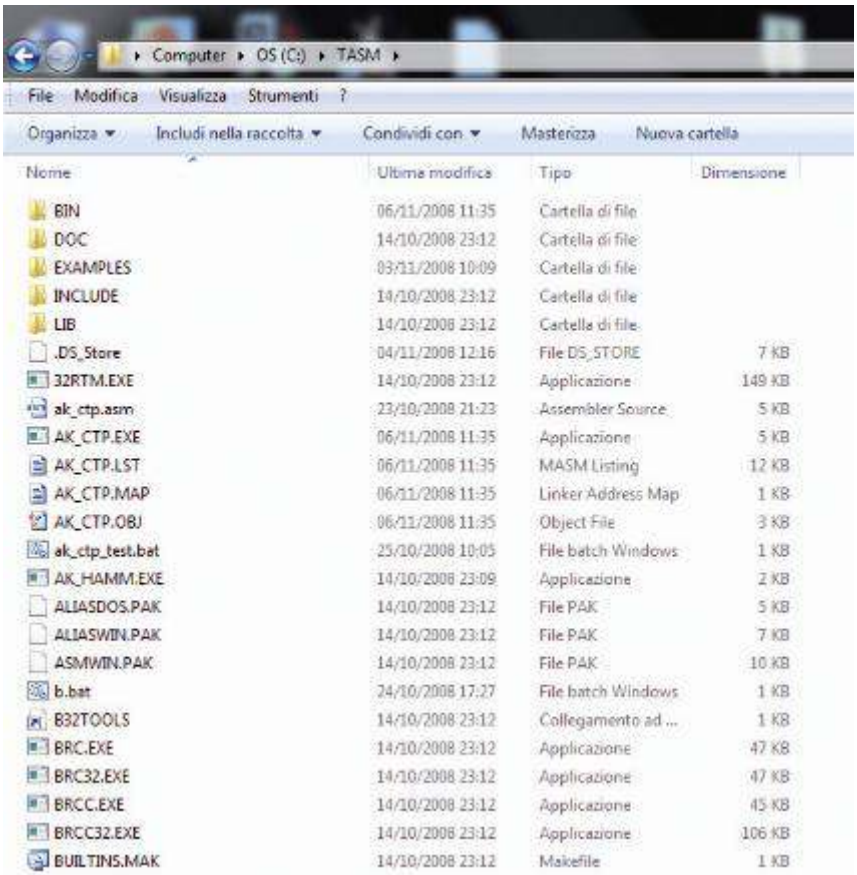
1 Crea una cartella nel disco C: di nome TASM e scompatta al suo interno il file compresso che contiene Turbo Assembler: ▶



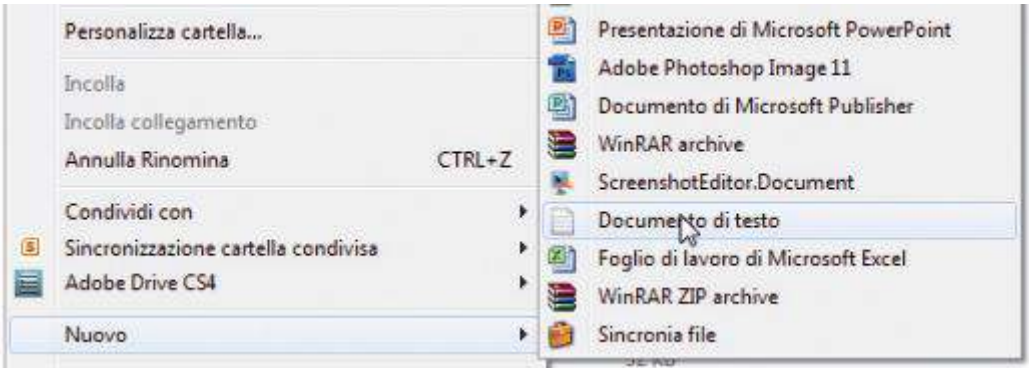
Per maggiore comodità, si raccomanda di creare una cartella posizionata nella radice del disco fisso in quanto consente un più facile accesso ai file in ambiente DOSBox.

2 A questo punto copia dalla sottocartella BIN i tre file principali (TASM.exe, TLINK.exe e TD.exe) nella cartella superiore (C:\TASM).

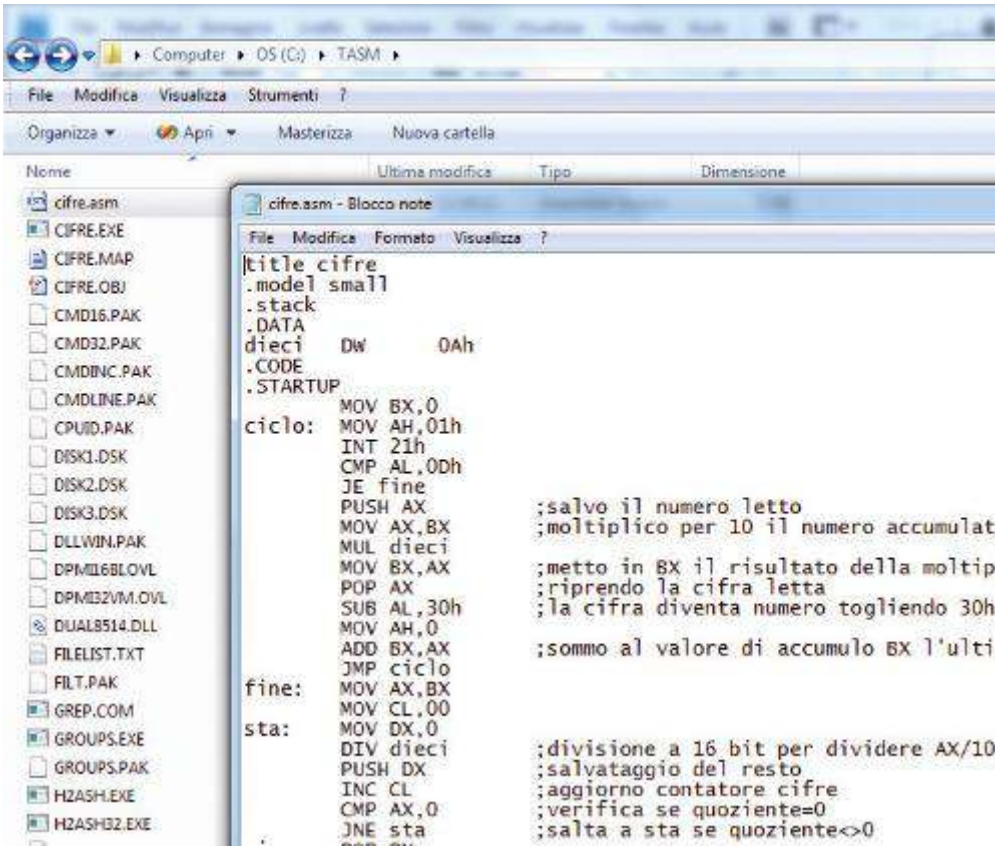
3 Adesso per creare un nuovo programma in assembly non devi fare altro che posizionarti nella directory C:\TASM usando Esplora risorse oppure Computer (schermata a lato). ▶



4 Per creare un nuovo file senza utilizzare un editor, devi fare clic con il tasto destro del mouse, quindi selezionare le voci Nuovo e Documento di testo (figura sottostante). ▶



5 Scrivi il codice sorgente e salvalo, sempre all'interno della cartella C:\TASM, con il nome che preferisci, tenendo comunque presente che non deve superare gli 8 caratteri e che l'estensione deve essere .asm: ▼

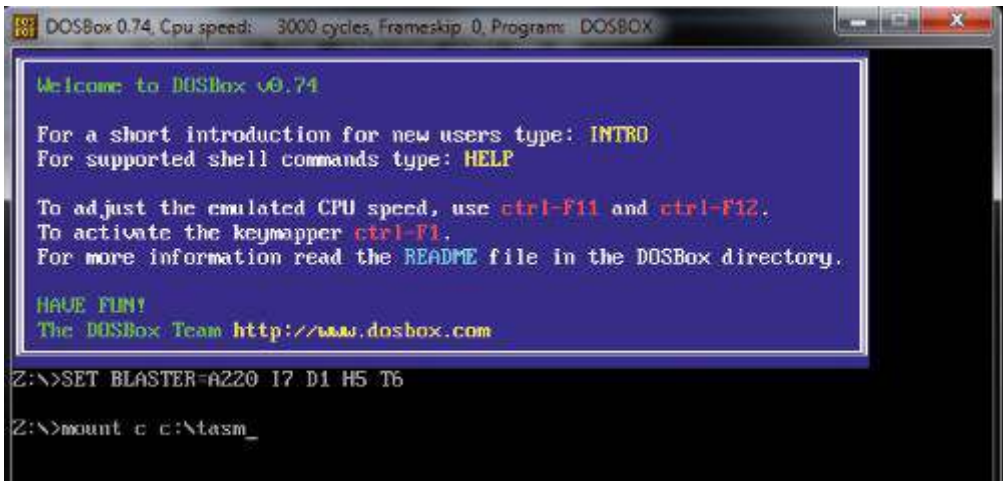


6 Per assemblare il file devi ricorrere all'ambiente DOSBox che consente di eseguire i file eseguibili dell'assemblatore. Fai clic sull'icona di DOSBox sul desktop. ▶

7 Appare la finestra della macchina virtuale MS-DOS (DOSBox):

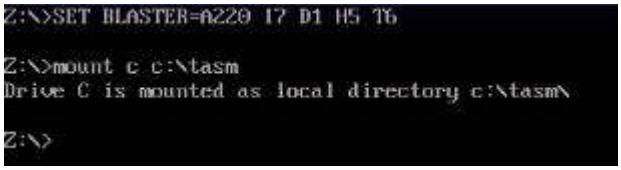
8 Adesso è necessario montare la directory che si intende usare. Per fare questo digita la seguente istruzione dal prompt dei comandi:

```
Z:>mount c c:\tasm
```

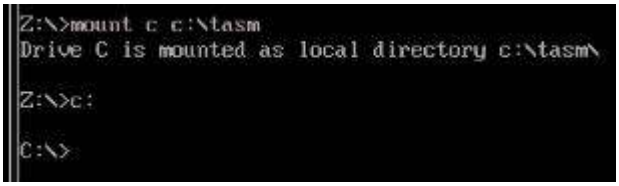


Per attivare l'emulatore a tutto schermo puoi premere ALT + INVIO; premendo una seconda volta la stessa combinazione di tasti ritornerai alla dimensione normale.

❾ Dopo aver premuto INVIO ottieni la schermata riportata a lato, che conferma l'attivazione della directory. ▶



❿ A questo punto digita C: e premi INVIO per posizionarti nella directory dell'assemblatore. ▶



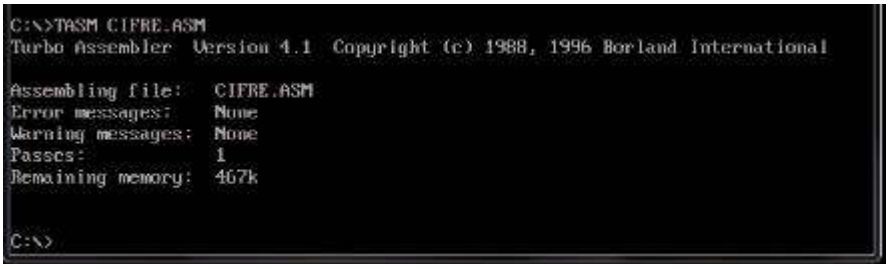
L'operazione di mount della directory deve essere ripetuta ogniqualvolta si intende utilizzare l'emulatore. Tuttavia aprendo il file dosbox.conf con un editor qualsiasi puoi fare in modo che la directory venga montata automaticamente all'apertura di DOSBBox. Devi aggiungere nella sezione [autoexec] (posta alla fine del documento) le righe riportate di seguito.

```
[autoexec]
mount c c:\tasm
c:
```

A questo punto salva il file dosbox.conf ed esegui DOSBox. Successivamente verificane il funzionamento.

⓫ Adesso vediamo come assemblare il file sorgente. In questo caso il file sorgente è stato salvato con il nome cifre.asm: l'assemblaggio avviene con il comando TASM, da digitare al prompt dei comandi:

```
TASM cifre
```



⓬ Come si può notare, nella schermata di cui sopra, appaiono alcune righe che indicano gli eventuali errori rilevati. In questo caso il TASM è esente da errori, tuttavia se così non fosse dovremmo prima modificare il codice sorgente, poi salvarlo nuovamente per poterlo di nuovo assemblare. La fase successiva è il linking (collegamento) del file oggetto che è stato creato automaticamente dall'assemblatore (cifre.obj). Per ottenere questo risultato digita il comando TLINK al prompt dei comandi:

```
TLINK cifre
```

```
C:\>TLINK CIFRE
Turbo Link  Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

C:\>
```

La fase di link genera il file eseguibile con estensione .exe. Per generare file con estensione .com devi scrivere il codice sorgente senza la direttiva .STACK e con .MODEL TINY (vedi Unità didattica 3 del presente modulo). La sintassi per generare un file con estensione .com è la seguente (TLINK /t):

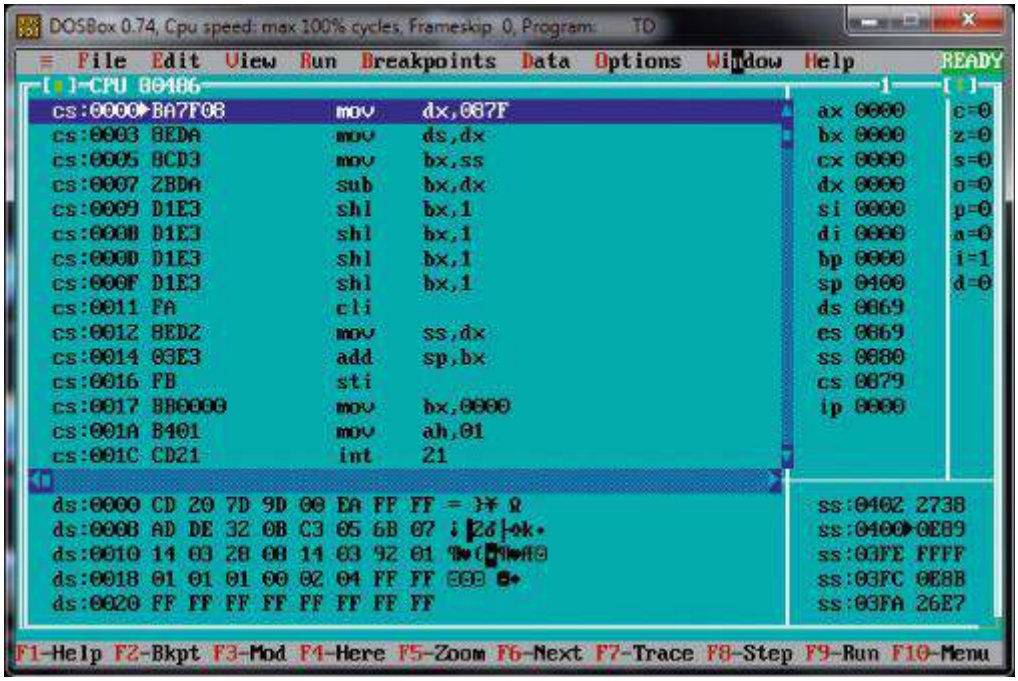
```
C:\>tlink /t cifre2
Turbo Link  Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

C:\>dir *.com
Directory of C:\.
CIFRE2   COM                66 15-03-2012 16:03
```

13. Infine non ti resta che eseguire il programma scrivendone semplicemente il nome al prompt, oppure eseguire il Turbo Debugger per verificarne il funzionamento passo passo con il comando TD al prompt dei comandi:

```
TD cifre.exe
```

```
C:\>TD CIFRE_
```

14. Come possiamo notare si apre la finestra del Turbo Debugger:



3 MODULO FONDAMENTI DI NETWORKING

OBIETTIVI

- Conoscere gli elementi fondamentali di una rete
- Conoscere le topologie di rete
- Acquisire il concetto di protocollo
- Apprendere le tecniche di multiplazione
- Apprendere le tecniche di commutazione
- Comprendere il concetto di architettura stratificata
- Conoscere i compiti dei livelli ISO-OSI e TCP-IP

ATTIVITÀ

- Classificazione delle reti in base alla topologia
- Riconoscere i dispositivi di rete
- Saper classificare le reti in base all'uso dei mezzi trasmissivi
- Classificare le tecniche di trasferimento dell'informazione
- Saper collocare le funzioni ai diversi livelli protocollari
- Saper confrontare il modello ISO-OSI con il modello TCP-IP

UNITÀ DIDATTICA 1

INTRODUZIONE AL NETWORKING

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere gli elementi fondamentali di una rete
- a classificare le reti
- a individuare le topologie di rete

Introduzione

"Non c'è nessun buon motivo per cui una persona dovrebbe tenersi in casa un computer."
Con questa frase nel 1977 Kenneth Henry Olsen, fondatore della Digital Equipe Corporation (DEC), prevedeva un'era in cui i computer sarebbero stati utilizzati unicamente in ambito scientifico o aziendale. Mai una previsione fu più errata. Olsen, che è scomparso lo scorso anno all'età di 84 anni, pur essendo tra gli ideatori del ◀ minicomputer ▶, non poteva certo immaginare quale impatto sconvolgente avrebbe avuto la tecnologia informatica nella vita di tutti i giorni.

◀ Minicomputer È un tipo di computer con capacità e costi intermedi tra i mainframe e i microcomputer: inizia a diffondersi negli anni '60 con l'avvento dei circuiti integrati. Molte innovazioni fondamentali dell'informatica moderna sono state concepite e sperimentate per la prima volta su di esso (per esempio il sistema operativo Unix, il linguaggio C, il mouse). Come primo minicomputer viene oggi riconosciuto il PDP-8, realizzato dalla DEC nel 1964, grazie alle intuizioni di Olsen. ▶

Oggi le tecniche e la velocità di comunicazione giocano un ruolo fondamentale e la possibilità che viene offerta ai calcolatori di comunicare tra distanze considerevoli come se fossero nello stesso locale ha impresso una svolta determinante allo sviluppo di reti di comunicazione informatiche.

L'ultimo decennio ha evidenziato la necessità fondamentale di mobilità e connettività, cioè l'esigenza di connettere i propri dispositivi, che siano personal computer o altri oggetti come gli smatphone, in reti diverse dalla propria "casalinga" o "aziendale": l'utente vuole sempre avere la possibilità di "connettersi" con la propria organizzazione o con il resto del mondo in qualunque posto si trovi.

Per offrire queste possibilità le reti informatiche si sono evolute cercando soluzioni alternative all'utilizzo dei cavi di collegamento: si è realizzata la connessione wireless, cioè una comunicazione senza l'utilizzo del cavo, che permette ai dispositivi mobili di connettersi semplicemente a opportuni punti di accesso alla rete (*access point*).

Con la diffusione delle connessioni, e quindi del trasferimento di informazioni, è sorto un insieme di problematiche connesse alla sicurezza delle informazioni stesse: l'introduzione di sistemi distribuiti, l'uso delle reti e delle apparecchiature di comunicazione per il trasferimento dei dati fra terminale utente e computer centrale o tra computer, via cavo e soprattutto via etere, richiedono di adottare le necessarie misure di sicurezza per proteggere i dati durante la trasmissione.

Lo standard per le reti wireless venne introdotto nel 1990 quando l'◀ IEEE ▶ costituì un apposito comitato (l'802.11) per definirne l'omonimo standard: esso si manifestò carente soprattutto in termini di sicurezza e venne ignorato per molto tempo al punto che non vi furono investimenti significativi nella tecnologia wireless per reti di dati per oltre un decennio dalla sua pubblicazione.

◀ IEEE Si tratta di una sigla che identifica l'associazione internazionale di scienziati professionisti con l'obiettivo di cercare nuove applicazioni e teorie in diverse discipline scientifiche (informatica, telecomunicazioni, biomedica ecc.). È l'acronimo di Institute of Electrical and Electronic Engineers (Istituto degli ingegneri elettrici ed elettronici).
Si occupa di definire e pubblicare gli standard delle discipline indicate per migliorare la qualità della vita dell'uomo favorendo la conoscenza e l'applicazione delle nuove tecnologie. Le pubblicazioni dello IEEE rappresentano una buona parte della documentazione ingegneristica mondiale; infatti coprono quasi tutti gli aspetti dell'informatica e delle telecomunicazioni moderne, avendo definito oltre mille standard industriali. ▶

Un ulteriore motivo che rallentò la diffusione dei dispositivi wireless era il costo elevato delle interfacce, per cui questa tecnologia venne utilizzata solo laddove l'uso di cavi era difficile oppure impossibile. Con il passare degli anni il drastico crollo dei prezzi ha fatto sì che tali tipologie di connessione entrassero prima nelle aziende e in un secondo tempo anche nelle case, permettendo la condivisione dei dati e della connessione a Internet.

Reti: definizioni e concetti di base

La sempre crescente diffusione di strumenti tecnologici, ricchi di molteplici funzionalità, necessita di una solida struttura che permetta loro di comunicare potendo facilmente scambiarsi informazioni.

- Prima di addentrarci a fondo in contenuti tecnici complicati definiamo in modo preciso che cosa si intende per rete informatica.

RETE INFORMATICA
Combinazione di hardware, software e cablaggio che, assieme, permettono a più dispositivi di elaborazione di comunicare tra loro (W. Odom).

Questa definizione si può applicare a molti tipi di rete diversi tra loro: le reti hanno lo scopo di fornire un servizio di trasferimento delle informazioni a una popolazione di utenti distribuiti su un'area geografica più o meno ampia.

Tra le reti informatiche rivestono particolare importanza quelle che hanno una modesta estensione territoriale dove i computer collegati tra loro sono nello stesso luogo, per esempio all'interno di un'area aziendale, di un'abitazione privata o semplicemente di un edificio: queste reti prendono il nome di reti locali, o ◀ LAN ▶.

◀ LAN Una rete locale (o LAN, Local Area Network) è un sistema di comunicazione che permette ad apparecchiature indipendenti di comunicare tra loro, entro un'area delimitata, utilizzando un canale fisico a velocità elevata e con basso tasso d'errore. ▶

La connessione tra le reti prende il nome di ◀ networking ▶.
La comunicazione tra i dispositivi di una rete avviene mediante lo scambio di segnali codificati in binario sotto forma di onde elettromagnetiche, come in ogni altro sistema di telecomunicazioni (TLC).

◀ Networking L'insieme dei sistemi di rete, ovvero le connessioni, di solito permanenti, tra i computer di tutto il mondo. ▶

SISTEMA TLC

Un sistema di TLC è costituito da un insieme di nodi (elaboratori di dati) e di collegamenti (canali di interconnessione tra nodi) che consentono a una (o più) entità sorgenti di inviare dati (informazioni) a una (o più) entità destinazioni.

Le entità sorgenti e destinazioni sono chiamate nodi terminali (end systems oppure hosts) del sistema mentre tutti gli altri nodi del sistema sono chiamati nodi di commutazione (switching nodes): per raggiungere l'host di destinazione il dato inviato dall'host sorgente "viaggia" sul canale fisico e viene ritrasmesso dai nodi di commutazione (inoltro dei dati).

L'informazione generata dalle sorgenti viene trasferita alle destinazioni sotto forma di segnali: il segnale è l'entità fisica che trasporta l'informazione dalle sorgenti alle destinazioni.

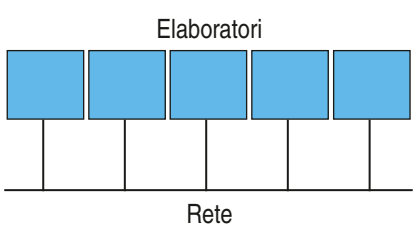
Aspetti hardware delle reti

Possiamo iniziare lo studio delle reti classificandole secondo due criteri:
- tecnologia trasmissiva;
- scala dimensionale.

Tecnologia trasmissiva

A seconda della modalità di trasmissione dei dati, le reti si possono dividere in due tipologie, elencate di seguito.

1. Reti broadcast: nelle reti broadcast gli host sono direttamente connessi al canale di comunicazione, condiviso da tutti. Colui che deve trasmettere un messaggio, che di solito consiste in un breve pacchetto di dati, inserisce l'indirizzo del destinatario e lo spedisce a tutti; naturalmente solo il destinatario lo leggerà mentre verrà ignorato dagli altri host.
   Si potrebbe però verificare la situazione nella quale esiste la necessità di inviare un messaggio a più destinatari contemporaneamente: in questo caso si chiama comunicazione multicast ed è il caso tipico di trasmissioni radio, televisive o videoconferenze. ▶



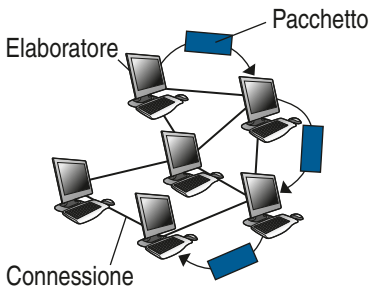
Zoom su...

BROADCAST

Nelle reti broadcast, oltre alla comunicazione tra terminali conosciuti, esiste anche la possibilità di richiedere o di offrire un servizio agli altri utenti, secondo le seguenti modalità:

- solicitation: la stazione che necessita di un servizio invia un messaggio broadcast per chiedere se esiste una stazione adiacente che lo eroga (e questa risponde);
- advertisement: la stazione che offre un servizio trasmette periodicamente un messaggio broadcast per informare tutte le stazioni adiacenti che quel servizio è disponibile.

2 Reti punto a punto: nelle reti punto a punto gli host sono connessi tra loro a coppie, mediante un insieme di canali di trasmissione. Per arrivare alla corretta destinazione un messaggio viene inviato dal mittente all'host al quale è direttamente connesso (oppure a uno degli host nel caso di connessione multipla), che ritrasmette il messaggio alla ricerca del destinatario.
Quindi il percorso che il messaggio deve compiere non è noto a priori, anzi, dato che le connessioni sono multiple e generano un grafo, sicuramente esisteranno più percorsi alternativi per raggiungere la destinazione. La scelta dei percorsi viene fatta da appositi algoritmi di instradamento (◀ routing ▶) che rivestono un ruolo determinante per il funzionamento della rete. ▶



◀ Routing Con il termine routing si intende il reinstradamento automatico del "messaggio" in base alla rilevazione e all'analisi della situazione corrente della rete: un algoritmo di routing decide quale connessione usare per instradare il "messaggio" dalla macchina sorgente alla macchina destinazione in base al "traffico" presente e ad altri parametri che saranno oggetto di studio nel prossimo anno di corso. ▶

Non è una regola ma nella pratica, per le reti geograficamente localizzate, si privilegia la struttura a broadcast mentre per quelle molto estese geograficamente è preferibile il punto a punto.
Come eccezione possiamo ricordare le reti satellitari, che sono broadcast, e le reti locali ATM, che sono punto a punto.

Scala dimensionale

Un criterio alternativo di classificazione è la dimensione dell'area geografica in cui sono dislocati i PC: le reti possono essere classificate in reti locali LAN, reti metropolitane MAN e reti geografiche WAN (e GAN), come mostrato nella tabella a lato. ▶

Classificazione delle reti in base alla distanza		
Area coperta	**Distanza**	**Tipo di rete**
Stanza	10 metri	LAN
Edificio	100 metri	LAN
Campus	1 chilometro	LAN
Città	10 chilometri	MAN
Area metropolitana	100 chilometri	MAN
Stato o nazione	1000 chilometri	WAN
Continente	5000 chilometri	WAN
Pianeta	10.000 chilometri	GAN

- LAN (Local Area Network): si tratta di un insieme di computer collegati tra loro e ubicati fisicamente nello stesso luogo, per esempio all'interno di un'area aziendale, di un'abitazione privata o semplicemente di un edificio.
- MAN (Metropolitan Area Network): in questo caso i computer si trovano all'interno di un'area urbana di grandi dimensioni oppure sono dislocati in più comuni limitrofi. Tra i vari esempi, prendiamo in considerazione quello riguardante più computer, interconnessi tra loro e collegati a un server centrale, dislocati nell'intero territorio comunale, e quello relativo ai PC delle segreterie delle facoltà universitarie dislocate in una determinata area metropolitana.
- WAN (Wide Area Network): in questo caso l'area geografica può comprendere diverse città, fino a interessare l'intero territorio nazionale o addirittura gli Stati con esso confinanti.
- GAN (Global Area Network): è facile intuire, dalla traduzione stessa dell'acronimo, che si tratta di reti che collegano computer dislocati in tutti i continenti. Internet, la Rete delle reti, è un tipico esempio di GAN.

Diverse sono le tecnologie impiegate per interconnettere le macchine: dal cavo in rame del comune doppino telefonico agli avanzati sistemi satellitari.

Reti locali

Le reti locali LAN sono generalmente delle reti private che connettono i PC di un'organizzazione, sia pubblica che privata, come un'azienda, una fabbrica, una scuola ecc.
Hanno un'estensione limitata (locale) che può raggiungere anche qualche chilometro e, di norma, si estendono in un singolo edificio o in un piccolo gruppo di immobili adiacenti (campus).

Hanno tre caratteristiche che le differenziano dagli altri tipi di rete:
- dimensione: come vedremo in seguito, esistono dei limiti dimensionali massimi dipendenti dalla natura dei mezzi trasmissivi utilizzati e dalle modalità di connessione dei dispositivi utilizzati;
- tecnologia trasmissiva: la trasmissione nella LAN avviene generalmente in broadcast, tranne in particolari casi di reti dedicate a speciali applicazioni. La velocità tipica di trasmissione è da 10 a 100 Mps, anche se oggi si vanno diffondendo le Giganet, reti con velocità di trasmissione dell'ordine dei 1000 Mps;
- topologia: con topologia si intende la "forma geometrica" con la quale i calcolatori (host) sono tra loro connessi, cioè la "disposizione geometrica dei nodi".

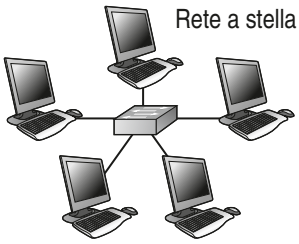
Le diverse topologie, descritte di seguito, sono raggruppabili in due tipi:
- topologia a bus: gli host sono connessi direttamente su "un pezzo di cavo" comune (segmento) e solo un calcolatore alla volta può trasmettere. Nel caso in cui due o più PC vogliano trasmettere contemporaneamente si contendono il segmento di cavo e si "sincronizzano" mediante un sistema chiamato "a collisione" (arbitraggio distribuito). Il protocollo standard per le topologie a bus broadcast con arbitraggio distribuito è l'IEEE 802.3, descritto nel seguito della trattazione;
- topologia ad anello: i calcolatori sono collegati in un anello e i dati vengono trasmessi "circolarmente" a tutti i PC; anche in questo caso i dati vengono trasmessi uno alla volta. Per poter trasmettere è necessario possedere il gettone (token), unico per tutta la rete, che abilita alla trasmissione. Lo standard per questo tipo di reti è l'IEEE 802.5 che deriva dalle "vecchie" reti proprietarie dell'IBM (la Token Ring).

Topologia delle reti locali

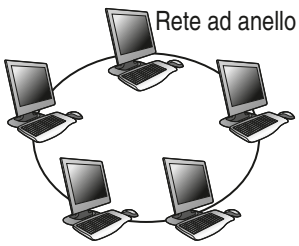
L'aspetto più evidente di una rete è sicuramente la topologia delle connessioni, ovvero la disposizione geometrica dei vari elementi (nodi) di cui è composta. Al crescere del numero dei nodi aumentano le geometrie possibili: ciascun tipo sarà più o meno adatto a una particolare esigenza; non è quindi possibile una classificazione rigorosa delle topologie di rete, possiamo solo individuare alcune tipologie fondamentali di riferimento.

Alcune reti utilizzano i collegamenti punto a punto, cioè la connessione diretta tra coppie di nodi, con canali riservati, senza stazioni intermedie. Altre prevedono i collegamenti multipunto, in altre parole utilizzano un canale comune su cui possono accedere più nodi.

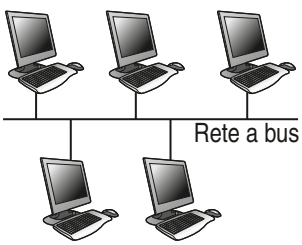
Le topologie più importanti sono descritte di seguito.



- Reti a stella: il numero di canali è uguale al numero di nodi meno uno ($c = n - 1$). In questo tipo di rete la *fault tolerance*, cioè la capacità di sopperire automaticamente a una situazione di errore, è inesistente: nel caso in cui il canale si guasti, la funzionalità della rete viene compromessa. Al centro della stella si trova il dispositivo concentratore hub o switch. ▶

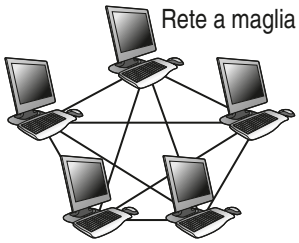


- Reti ad anello: nella topologia ad anello ogni nodo è collegato con altri nodi in modo da formare una struttura circolare. Ogni informazione da trasferire deve percorrere l'anello fino al destinatario: per esempio, se consideriamo il messaggio di risposta per la conferma, ogni scambio di informazioni coinvolge tutti i nodi della rete, che devono cooperare alla comunicazione anche se non sono direttamente interessati al messaggio. Il guasto di un nodo quindi causa la "caduta" dell'intera rete anche se è facile "ponticellare" l'ingresso e l'uscita di un nodo per escluderlo (inserendo un segmento di cavo di rete che faccia "saltare" il nodo guasto, cortocircuitandolo). ▶

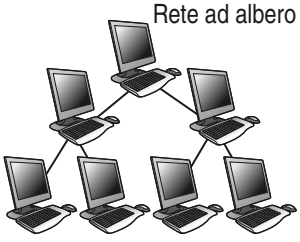


- Reti a bus: i computer sono connessi a un unico canale comune, condiviso da tutte le comunicazioni. I messaggi, organizzati in pacchetti, vengono inviati sul canale e tutti i nodi possono valutarli, ma solo il nodo che riconosce di essere il destinatario acquisisce il messaggio. Ovviamente solo un nodo alla volta può trasmettere sul canale. ▶

- Reti a maglia: una topologia a maglia completamente connessa è caratterizzata dal fatto che ogni nodo è collegato con tutti gli altri; ciò significa che esistono tutte le possibili connessioni tra i vari nodi della rete. È impensabile adottare una topologia di questo tipo per una rete medio-grande, proprio per il fatto che il numero di collegamenti tra i vari nodi diverrebbe troppo elevato: già nel caso di una rete con mille nodi sono necessari circa un milione di collegamenti. La topologia a maglia completamente connessa offre però un significativo vantaggio: in questo caso non esistono, nella pratica, problemi di commutazione, in quanto l'individuazione del canale che collega due interlocutori qualsiasi si risolve andando a scegliere proprio "il cavo" che collega direttamente i due utenti. Inoltre, la presenza di un così alto numero di canali rende questa topologia molto adatta a tollerare la presenza temporanea di eventuali guasti. ▶



- Reti ad albero: eliminando da una topologia a maglia completamente connessa tutti i canali che non sono indispensabili per permettere ai nodi di comunicare, si arriva a una topologia ad albero. In questo caso tra due nodi qualsiasi esiste un unico percorso fisico, quindi se un canale si satura o si guasta la rete non è più in grado di funzionare. I vantaggi della topologia ad albero derivano dalla semplicità della topologia e della commutazione, perché l'esistenza di un unico cammino tra due nodi rende molto semplici le procedure per la costruzione di un collegamento tra due utenti. Il ridotto numero di canali implica anche bassi costi. ▶



Reti geografiche

Le reti geografiche WAN (*Wide Area Network*) si estendono a livello di una nazione oppure di un continente fino al limite dell'intero pianeta (GAN, *Global Area Network*), come la rete Internet.

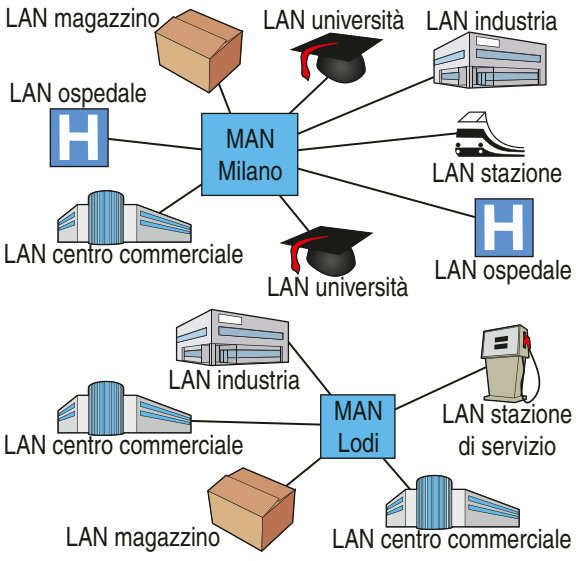
In una WAN possiamo individuare due componenti:

- un insieme di host (ospiti) sui quali vengono eseguiti i programmi degli utenti (end system);
- un insieme di connessioni (subnet) che li collega tra loro e permette il trasferimento delle informazioni da un host sorgente (il mittente) a un host ricevente (il destinatario) attraverso altri host o dispositivi che trasmettono i dati fino a che non sono giunti a destinazione. Questo meccanismo prende anche il nome di comunication subnet.

Ogni subnet può essere vista come composta da due elementi:

- la linea di trasmissione: è il canale di comunicazione, che può essere costituito dal filo di rame, da fibra ottica oppure dall'etere, come nelle comunicazioni wireless o negli avanzati sistemi satellitari;
- i dispositivi di commutazione: sono i componenti che permettono al messaggio di "spostarsi" attraverso subnet diverse, cioè il dispositivo che riceve il segnale lo analizza e decide su quale altra subnet lo deve trasmettere affinché possa giungere a destinazione. Questi dispositivi sono i repeater, i bridge e i router.

Nella figura è rappresentata una tipica MAN che interconnette più LAN e WAN. ▶



Reti wireless

Le reti wireless rappresentano oggi una forma di connessione sempre più importante per molte attività e per la vita domestica, sia per il lavoro che per l'home entertainment (videogame, home theatre ecc.): praticamente ogni dispositivo oggi presente sul mercato, dal PC portatile al telefono cellulare, ha un'interfaccia wireless che gli permette di connettersi in rete.

L'ideologia fondamentale che ha aperto una nuova era di comunicazione con la tecnologia wireless è la connessione sempre e ovunque (*anytime & anywhere*).

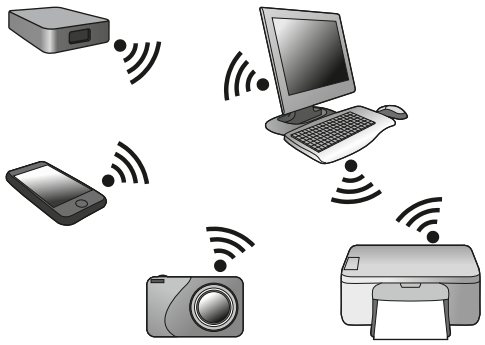
Le reti senza fili prendono genericamente il nome di WLAN (*Wireless Local Area Network*), cioè reti che sfruttano la tecnologia wireless.

Più precisamente, le tipologie di reti wireless sono due e si classificano in base alle dimensioni:

- PAN (*Personal Area Network*);
- WLAN propriamente dette, cioè *wireless LAN*.

Le reti PAN sono composte da collegamenti a portata ridotta, tipicamente limitata agli oggetti indossati da una persona o a quelli presenti in un'automobile o su una scrivania. Bluetooth è la tecnologia più utilizzata per questo tipo di collegamento, specificamente progettata per realizzare la comunicazione senza fili per apparecchi di piccole dimensioni. Il concetto chiave ispiratore di

questa tecnologia è quello di eliminare completamente i cavi necessari alla comunicazione tra apparecchi. Questi collegamenti senza fili sono effettuati usando un ricetrasmettitore che opera nella frequenza di 2.4 GHz, ma le frequenze utilizzate variano da Paese a Paese, in relazione alle normative nazionali. ▶



Le LAN aziendali, invece, sono spesso sostituite o integrate da reti wireless e prendono il nome di WLAN: questo sistema di connessione è vantaggioso negli edifici più vecchi, dove non esiste o non è possibile installare un impianto di cablaggio strutturato.
La tecnologia WLAN più diffusa è nota con il nome commerciale Wi-Fi (*Wireless Fidelity*): con questo termine si indicano i dispositivi che possono collegarsi alle WLAN basate sullo specifico standard IEEE 802.11 (parecchio rielaborato e completato rispetto alla sua prima versione del 1990).

Generalmente i sistemi wireless utilizzano per comunicare onde radio a bassa potenza in radiofrequenza (RF). Lo strumento che consente l'accesso alla rete tramite linea wireless prende il nome di access point (AP). ▶

Zoom su...

UN ESEMPIO DI GAN: LA RETE INTERNET

La rete Internet potrebbe essere paragonata a una ragnatela formata da risorse collegate tra loro mediante link (collegamenti ipertestuali), che costituiscono ciò che si potrebbe definire un ipertesto di dimensioni potenzialmente infinite.
Le risorse situate all'interno della rete Internet risiedono su computer tra loro collegati e che fungono da server, mentre i computer che quotidianamente vengono utilizzati per navigare svolgono la funzione di client.
Per comprendere meglio il significato di client e server si può ricorrere a un paragone. Immaginiamo di entrare in un ristorante e di chiedere una particolare pietanza al cameriere. Il client può essere individuato in colui che effettua la richiesta di una risorsa, in questo caso nel cliente che si è seduto al tavolo. Il cameriere è il server, ovvero colui che fornisce la risorsa. La risorsa richiesta potrebbe essere disponibile, indisponibile momentaneamente (tempi lunghi di preparazione perché richiesta da molti altri clienti) oppure terminata. Analogamente il server non sempre può fornire la risorsa richiesta, oppure, se la risorsa viene richiesta da molti client, si allungano i tempi di risposta.
Nella rete Internet le risorse sono molto spesso di tipo software e, nel caso della navigazione, si tratta di siti web. In particolare, ogni volta che un client web fa clic su un link, invia una richiesta a un server web, che risponde inviando la risorsa corrispondente all'indirizzo richiesto.

> Un server web è un programma, individuato in un computer, che si occupa di fornire, su richiesta del browser, una pagina web. Le informazioni inviate dal server web viaggiano in rete trasportate dal protocollo HTTP. L'insieme dei server web dà vita al World Wide Web, uno dei servizi più utilizzati di Internet.

Normalmente un server web risiede su sistemi dedicati, ma può essere eseguito su computer dove risiedano altri server o in cui vengano utilizzati anche per altri scopi. Per esempio, si può installare un server web su un normale personal computer allo scopo di provare il proprio sito web.

Tra le varie risorse che possono essere scambiate in rete mediante il protocollo HTTP vi sono documenti, comunemente chiamati pagine web, scritte in un linguaggio di contrassegno (per esempio HTML). I documenti HTML contengono informazioni e si collegano a loro volta ad altre risorse (per esempio, immagini, suoni, animazioni), costituendo di fatto la spina dorsale degli ipertesti.
Il browser (letteralmente "sfogliatore") è il programma usato dal client per effettuare richieste HTTP a un server web. Permette di individuare la posizione di una pagina in Internet e di interpretare le righe di codice sorgente scritte in HTML, in modo da presentare la pagina web all'utente finale.
Il codice sorgente HTML che viene interpretato dal browser nelle ultime versioni è anche in grado di mandare in esecuzione programmi scritti in Java (Applet) o di interpretare programmi incapsulati internamente – detti script –, quali per esempio VBScript, JScript o il più conosciuto JavaScript.
I browser più diffusi sono certamente Microsoft Internet Explorer (a volte denominato IE per brevità), Mozilla Firefox e Google Chrome, oltre ad altri meno noti (Safari, Opera, Galeon, SlimBrowser, Lynx).
Essendo per propria natura una rete mondiale e quindi una GAN, Internet è composta da nodi che possono essere suddivisi in tre categorie principali.

1. Router: sono gli "instradatori" della rete, che permettono ai pacchetti di dati di essere "rimbalzati" da una sottorete all'altra fino a raggiungere la destinazione, mediante particolari algoritmi, detti appunto di routing.
2. *Internet Service Provider* (ISP): grazie ai molteplici POP (*Point Of Presence*), è un server in grado di distribuire la connessione ai client web che ne fanno richiesta.
3. Host: contengono le informazioni che devono essere memorizzate in Internet oppure sono gli "ospiti" che cercano le informazioni sulla rete.

Nella figura a lato è riportata una rappresentazione scherzosa del World Wide Web. ▶

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Un minicomputer è:

- ○ più piccolo di un mainframe
- ○ più piccolo di un microcomputer
- ○ più piccolo di un home computer
- ○ più piccolo di un personal computer

2 L'acronimo IEEE significa:

- ○ *Internet Electrical and Electronic Engineers*
- ○ *Industrial Electrical and Electronic Engineers*
- ○ *Institute of Electrical and Electronic Engineers*
- ○ *International Electrical and Electronic Engineers*

3 LAN è l'acronimo di:

- ○ *Limited Area Network*
- ○ *Local Area Network*
- ○ *Limited Access Network*
- ○ *Local Access Network*

4 A seconda della modalità di trasmissione dei dati, le reti si possono dividere in:

- ○ reti broadcast
- ○ reti multicast
- ○ reti unicast
- ○ reti punto a punto

5 Indica per ciascuna area il tipo di rete:

- ○ locale .....
- ○ nazione .....
- ○ pianeta .....
- ○ campus .....
- ○ immobile .....
- ○ continente .....
- ○ città .....

6 WAN è l'acronimo di

- ○ *White Area Network*
- ○ *Wide Area Network*
- ○ *Wire Area Network*
- ○ *Wine Area Network*

7 Quale, tra le seguenti, non è una caratteristica che differenzia le LAN dagli altri tipi di rete?

- ○ dimensione
- ○ velocità
- ○ tecnologia trasmissiva
- ○ topologia

8 Indica lo standard IEEE per ciascuna tipologia:

- ○ rete ad anello .....
- ○ rete a bus .....
- ○ rete wireless .....

Test vero/falso

1	Le LAN sono reti informatiche che hanno una modesta estensione territoriale.	V F
2	Le entità sorgenti e destinazioni sono chiamate nodi terminali.	V F
3	Con *end system* si intendono solo gli host di destinazione.	V F
4	I nodi di commutazione sono anche chiamati *switching nodes*.	V F
5	Il segnale è l'entità fisica che trasporta l'informazione dalle sorgenti alle destinazioni.	V F
6	Nelle reti broadcast gli host sono connessi al canale di comunicazione mediante un router.	V F
7	Nelle reti broadcast i messaggi arrivano a tutti gli host.	V F
8	Nelle reti broadcast solicitation, la stazione che necessita di un servizio invia un messaggio.	V F
9	Nelle reti broadcast advertisement, la stazione che offre un servizio trasmette un solo messaggio a tutti.	V F
10	Nelle reti multicast la scelta dei percorsi viene fatta da appositi algoritmi di instradamento.	V F
11	Il termine campus indica un piccolo gruppo di immobili adiacenti.	V F
12	La trasmissione nella LAN avviene generalmente in multicast.	V F

UNITÀ DIDATTICA 2

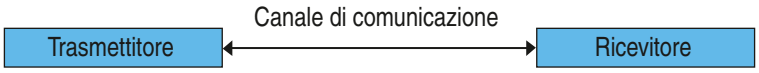
IL TRASFERIMENTO DELL'INFORMAZIONE

IN QUESTA UNITÀ IMPAREREMO...

- le generalità sui protocolli
- le tecniche di multiplazione
- le tecniche di accesso
- le tecniche di commutazione

La trasmissione delle informazioni

Il sistema più semplice per realizzare la trasmissione di informazioni può essere schematizzato come nella figura seguente:



La comunicazione tra due calcolatori si realizza mediante lo scambio di dati sul canale di comunicazione: per poter essere trasmessi i dati sono stati opportunamente codificati e trasformati in un segnale che può essere elettrico (una tensione oppure una corrente) o una qualsiasi grandezza fisica capace di convogliare un'informazione (per esempio un'onda luminosa).

Le informazioni che vengono trasmesse attraverso una rete vengono denominate "traffico", utilizzando come metafora la circolazione dei mezzi di trasporto sulla rete stradale delle nostre città.

Modalità di comunicazione

Le modalità di comunicazione tra entità si dividono in due grandi classi:
- modalità a connessione (*connection-oriented*);
- modalità senza connessione (*connectionless*).

Nella modalità connection-oriented, prima dell'inizio della trasmissione delle informazioni, si deve stabilire una connessione tra il mittente e il destinatario, che viene "rilasciata" al termine delle comunicazioni.

Possiamo quindi individuare tre fasi distinte:
- apertura della connessione: è la fase più delicata durante la quale si "cerca l'accordo" tra i due host e si stabilisce la connessione;
- trasferimento dell'informazione: è la parte centrale durante la quale avviene la comunicazione;
- chiusura della sessione: viene rilasciata la risorsa e terminata la comunicazione.

Un esempio tipico di funzionamento con modalità connection-oriented è la telefonata.

Nella modalità connectionless il mittente effettua il trasferimento delle informazioni in modo autonomo, senza che avvenga un preventivo accordo con il destinatario. Non si verifica quindi una "sincronizzazione" tra mittente e destinatario e il mittente ignora se la sua trasmissione sia andata o meno a buon fine, a meno che successivamente il destinatario non inoltri un messaggio di conferma ricezione.
Un esempio tipico di funzionamento con modalità connectionless è il servizio postale.

Modalità di utilizzo del canale

La trasmissione dei dati, a seconda della modalità di utilizzo dei canali di comunicazione, può essere classificata in:
- simplex: la trasmissione può avvenire solo in un senso; è il caso tipico delle trasmissioni televisive o radiofoniche dove i ricevitori sono i televisori o le radio. Tranne in casi particolari (televisione interattiva), non è possibile che un apparecchio ricevitore invii messaggi al trasmettitore;
- half-duplex: la trasmissione può avvenire nei due sensi, ma in tempi diversi; l'esempio tipico è il "walkie-talkie", dov'è possibile parlare solo uno alla volta. Il canale di comunicazione è unico e viene utilizzato in modo alternato tra le due stazioni che si scambiano anch'esse la modalità di funzionamento, passando da trasmettitore a ricevitore;
- full-duplex: la trasmissione può avvenire nei due sensi contemporaneamente (per esempio nella trasmissione telefonica). Viene realizzata mediante due canali di trasmissione, uno per la trasmissione in un senso e l'altro per quella in senso contrario.

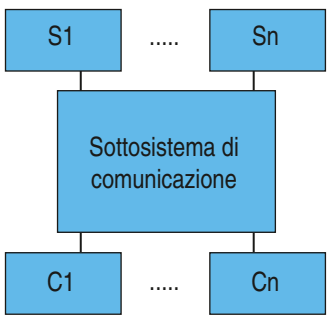
> Nelle reti vengono utilizzate generalmente le ultime due modalità, in quanto a ogni trasmissione è sempre richiesta una risposta e quindi la comunicazione deve sempre essere nei due sensi.

Nelle comunicazioni tra calcolatori le tecnologie utilizzate sono di vario tipo e impiegano metodologie e componenti diversi anche sulla stessa rete: vengono introdotti dei meccanismi che permettono l'interazione tra dispositivi e computer eterogenei, indipendentemente dall'hardware utilizzato, dal tipo e dalla natura dei computer connessi, e dai sistemi operativi che sono installati su di essi. Il primo che analizziamo è il concetto di protocollo.

Generalità sui protocolli

Qualunque entità che intenda comunicare con un'altra deve stabilire una serie di regole e convenzioni, tra le quali assume importanza fondamentale il protocollo di comunicazione.

Prendiamo per esempio una rete di telecomunicazione e rappresentiamola nella figura a lato come un sottosistema di comunicazione composto da elementi di connessione, di trasferimento dati e di commutazione al quale sono connesse due tipologie di elementi: ▶
Ⓐ i fornitori di servizi (o server);
Ⓑ i fruitori di servizi (o client).



Nelle prime reti degli anni '70 i server erano i ◀ **mainframe** ▶ e i client i terminali, ma con l'evoluzione e lo sviluppo informatico oggi non ci sono differenze hardware tra server e client: la differenza viene fatta solo dal software e quindi un calcolatore può essere contemporaneamente server per alcuni servizi e client per altri.

◀ Mainframe Il termine mainframe è legato alla storia dell'informatica: con esso venivano indicati i sistemi centrali costituiti da un unico grosso elaboratore al quale erano collegati terminali non intelligenti: oggi il termine viene usato per distinguere gli elaboratori di fascia alta da quelli meno potenti. ▶

Il sottosistema di comunicazione si è trasformato da una semplice connessione realizzata per mezzo di un cavo a un complesso sistema composto da apparecchiature e risorse di telecomunicazione (sistema telematico) con cui si possono fornire servizi a distanza.

Il sistema telematico spesso coincide con quella che abbiamo detto essere la rete di calcolatori.

Per trasmettere informazioni attraverso la rete tutti i dispositivi devono condividere un insieme di regole e di specifiche in quanto solo se utilizzano un "linguaggio" comune possono comprendere le comunicazioni che si scambiano.

Abbiamo già visto che sulla stessa rete coesistono tipologie di dispositivi diversi che funzionano con sistemi operativi diversi: le regole per comunicare devono essere comuni per tutti, indipendentemente dalla diversa natura hardware e software.

Per prima cosa ogni computer deve connettersi alla rete con un dispositivo opportuno (interfaccia) che, utilizzando particolari moduli software, nasconda la sua vera natura rendendolo "virtualmente" uguale agli altri dispositivi di rete durante le comunicazioni.

Per realizzare queste funzionalità complesse si è progettata un'architettura organizzata a livelli delegando parti di competenze a ogni livello e stabilendo per ciascuno di essi compiti, funzionalità e meccanismi di funzionamento.

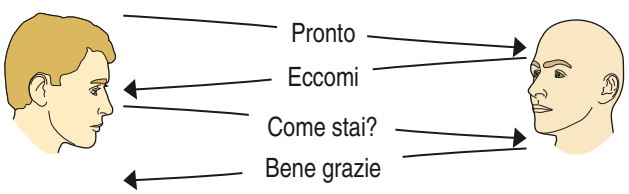
Con i protocolli sono stati definiti i meccanismi di comunicazione all'interno di ogni singolo livello e le modalità di trasferimento dai dati tra i singoli livelli.
Una semplice definizione di protocollo è la seguente.

PROTOCOLLO
Un protocollo è un insieme di regole e di azioni che due applicazioni che vogliono comunicare devono applicare e seguire per scambiarsi i dati.

All'interno del protocollo devono essere specificati:
- la tipologia dei messaggi che le applicazioni devono scambiarsi;
- le azioni che esse devono intraprendere per inviare e ricevere correttamente i messaggi.

ESEMPIO

Un semplice protocollo è quello che giornalmente viene utilizzato tra due esseri umani quando per esempio si scambiano una comunicazione telefonica come quella raffigurata a lato: ▶



Le regole sono semplici e possono essere così sintetizzate:
- si utilizza la stessa lingua;
- si parla uno alla volta;
- si inizia la conversazione con una frase convenzionale;
- si termina la conversazione con una frase convenzionale;
- si intraprendono azioni specifiche quando i messaggi sono stati ricevuti.

Possiamo osservare come le regole vengano definite sia sulla struttura del messaggio

<inizio> <messaggio>< fine>

che sulla sua natura

<lingua> <alternanza>

I protocolli di comunicazione usati nelle reti, chiamati anche protocolli di linea, hanno sostanzialmente le stesse caratteristiche ma devono essere assolutamente rigidi in quanto la comunicazione avviene tra macchine e non tra uomini: devono definire il formato e la natura fisica del messaggio, la sua struttura e il significato di ogni sua parte in modo che possa essere compreso in modo univoco e senza possibilità di interpretazioni soggettive.

Vedremo nel dettaglio il protocollo Ethernet, analizzandolo livello per livello a partire dalla sua "forma fisica" in termini di segnali elettrici e/o ottici fino al più alto livello, quello delle applicazioni.

Tecniche di trasferimento dell'informazione

Nelle reti informatiche possiamo individuare due tipi di risorse utilizzate per effettuare la comunicazione del messaggio da un mittente a un destinatario:
- risorse trasmissive: il mezzo trasmissivo utilizzato per il collegamento e quindi la trasmissione del segnale;
- risorse elaborative: i dispositivi necessari per effettuare le operazioni di indirizzamento e utilizzo del mezzo trasmissivo.

La maggior parte delle elaborazioni avviene nei nodi interni della rete che, in base alla destinazione del messaggio, "instradano" i dati sui diversi "segmenti" della rete, gestendo di fatto la trasmissione e l'utilizzo delle connessioni.
Come spesso accade, le risorse non sono mai disponibili "in abbondanza", anche perché ogni utente gradirebbe avere una linea completamente dedicata, veloce ed efficiente. È quindi necessario introdurre alcune tecniche che permettano di gestire al meglio le risorse disponibili per massimizzare l'efficienza del funzionamento del sistema.

Analizziamo quindi come avviene il trasferimento dei dati sulla rete da parte di due applicazioni che vogliono scambiarsi dati.

MODI DI TRASFERIMENTO

Con modo di trasferimento si intendono le modalità (strategie) adottate dalla rete per permettere lo scambio di dati tra due applicazioni.

Un modo di trasferimento è individuato quando sono specificate le tre componenti seguenti:

1. la tecnica di multiplazione adottata dalla rete:
   - multiplazione statica;
   - multiplazione dinamica;
2. le modalità di accesso al canale (o architettura protocollare):
   - centralizzato;
   - distribuito (o multiplo):
     - protocollo deterministico senza contesa:
       - protocolli di accesso multiplo a divisione di tempo (TDMA);
       - protocolli di accesso multiplo a divisione di frequenza (FDMA);
       - tecniche di passaggio del testimone;
     - protocolli ad accesso casuale o contesa:
       - Aloha:
         - puro;
         - slotted;
       - CSMA/CD (*Carrier Sense Multiple Access/Collision Detection*);
     - CDMA (*Code Division Multiple Access*);
3. la tecnica di commutazione:
   - a commutazione di circuito;
   - a commutazione di messaggio;
   - a commutazione di pacchetto.

Multiplazione (multiplexing)

Nelle reti la connessione fisica tra due elaboratori (linee bifilari, collegamenti in fibra ottica, collegamenti radio ecc.) deve essere utilizzata in modo tale che più applicazioni possano contemporaneamente trasmettere informazioni sullo stesso canale: si pone quindi il problema di sfruttare la capacità del mezzo a disposizione per ottenere più canali di livello fisico (o logico) sullo stesso mezzo trasmissivo.

MULTIPLAZIONE

La multiplazione (multiplexing) è l'implementazione del concetto di condivisione di risorse trasmissive.

Lo schema di multiplazione definisce le modalità secondo cui i dati generati da più applicazioni distinte possono essere trasmessi ordinatamente attraverso uno stesso canale presente nella rete, cioè il modo in cui condividono le risorse di trasmissione (banda e tempo) messe a disposizione da ogni singolo canale.

In caso di multiplazione il canale prende appunto il nome di canale multiplato.

Una prima classificazione delle modalità di multiplazione è quella che viene fatta tra:

- multiplazione statica;
- multiplazione dinamica.

Nella multiplazione statica la suddivisione del canale avviene in parti fisse, in tanti canali fisici quanti sono noti a priori, come per esempio nelle trasmissioni via satellite o televisive.

Nella multiplazione dinamica la suddivisione del canale avviene in base alle richieste, quindi in tempo reale, e a seconda delle condizioni di traffico, in canali logici che vengono assegnati temporaneamente a un coppia di host che li rilasciano non appena hanno terminato di comunicare.

Esistono due modalità di assegnamento dinamico:

- assegnazione a domanda: è la modalità che permette di massimizzare le prestazioni nel caso in cui un'applicazione alterni l'utilizzo della risorsa con momenti di pausa: durante il periodo di non utilizzo il canale viene rilasciato per altre attività e viene riassegnato a seguito di una successiva richiesta. In questo caso non avviene un *inizio attività* e una *fine attività*, ma esiste un frazionamento in sottoparti del messaggio, in "unità informative autonome" che volta per volta si contendono il canale;

> Questa modalità è conveniente per un utente con bassa utilizzazione del canale rispetto al tempo di vita dell'attività.

- preassegnazione: la risorsa viene assegnata all'inizio dell'attività, su richiesta, e viene riservata per tutta la durata dell'attività stessa (per esempio, una comunicazione telefonica).

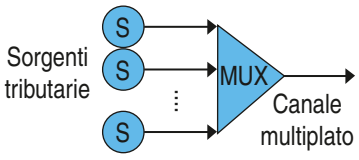
Tecniche di accesso o protocolli di accesso

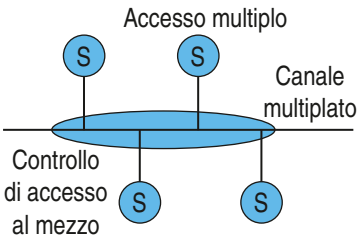
Dopo aver determinato la modalità di assegnazione del canale il problema più importante è quello della formazione del flusso multiplato in base alla modalità di accesso al canale.

Le tecniche di accesso descrivono le modalità con le quali i nodi terminali utilizzano il mezzo trasmissivo al fine di realizzare una corretta trasmissione delle informazioni e una gestione ottimale del traffico all'interno di una rete locale: l'obiettivo è quello di smaltire velocemente il traffico dati.

Esistono due alternative:

- accesso centralizzato: esiste un multiplatore che prima acquisisce (e memorizza) tutte le richieste e successivamente assegna alle sorgenti, in modo dinamico o statico, la banda richiesta del canale trasmissivo attraverso un'opportuna politica di scheduling; ▶
- accesso distribuito o multiplo: sono le sorgenti che generano il flusso multiplato accedendo direttamente alla rete; questa tecnica richiede l'utilizzo di particolari protocolli per assegnare la banda a disposizione a tutti i richiedenti, cercando di minimizzare le attese delle sorgenti e di ottimizzare l'utilizzo della banda disponibile. ▶



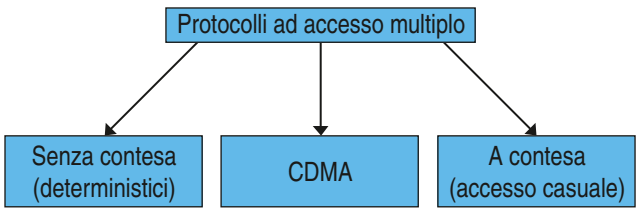


> In questo volume analizziamo solo protocolli di multiplazione con accesso multiplo.

Classificazione delle tecniche di accesso multiplo

Le tecniche di accesso multiplo si dividono in tre classi:

- protocolli deterministici senza contesa: evitano la possibilità che due utenti accedano al canale contemporaneamente (collisione) programmando l'accesso di ogni utente;
- protocolli ad accesso casuale o contesa: si possono avere interferenze tra le stazioni trasmittenti che sono gestite con particolari procedure;
- CDMA (*Code Division Multiple Access*): le stazioni operano in continua sovrapposizione e interferenza ma codificando appositamente i segnali. ▶



Accesso multiplo senza contesa: protocolli deterministici

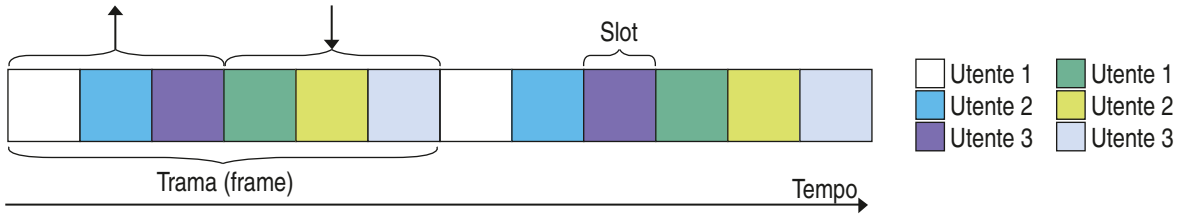
I principali protocolli di accesso multiplo di tipo deterministico sono:

- a divisione di tempo TDMA (*Time Division Multiple Access*): gli utenti trasmettono in intervalli di tempo diversi;
- a divisione di frequenza FDMA (*Frequency Division Multiple Access*): gli utenti utilizzano frequenze diverse per trasmettere;
- a passaggio del testimone (a gettone).

Accesso multiplo a divisione di tempo TDMA

Nel protocollo a divisione di tempo TDMA viene effettuata una doppia suddivisione del tempo di trasmissione: il primo livello è il frame, suddiviso in un numero fisso di intervalli più piccoli chiamati slot.

A ogni utente viene assegnato un particolare slot all'interno del frame sia per la trasmissione che per la ricezione.

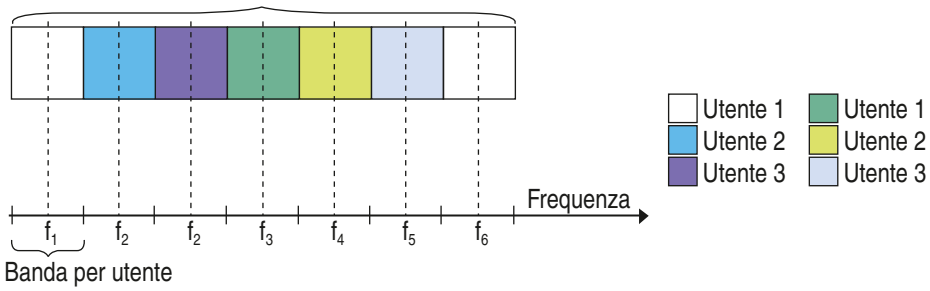


Ogni utente può quindi trasmettere per un intervallo di tempo pari a uno slot per ogni frame e il trasmittente deve essere sincronizzato con il ricevente.

Un esempio di trasmissione che usa questa tecnica è quella telefonica GSM (la durata del frame è di 4.62 ms e con n. slot/frame = 8 si ottiene che ogni slot ha una durata di 0,577 ms).

Accesso multiplo a divisione di frequenza FDMA

Nel protocollo a divisione di frequenza FDMA attorno alla frequenza caratterizzante del canale vengono create delle sottobande uguali, concesse in modo esclusivo a un utente per tutto il periodo necessario alla trasmissione; una è per la trasmissione e l'altra per la ricezione.



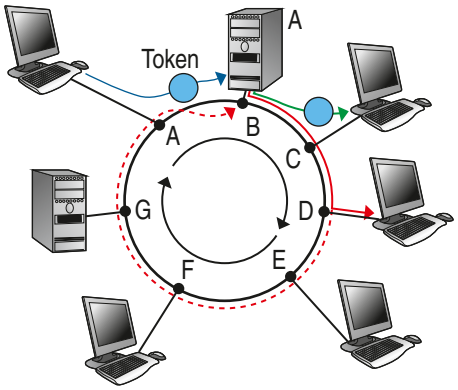
In questo protocollo non è necessaria la sincronizzazione tra le varie stazioni in quanto hanno un "canale dedicato".

Tecniche di passaggio del testimone (token passing)

Nella tecnica di passaggio del testimone le stazioni compongono un circolo, dove ciascuna conosce solo l'indirizzo delle due a essa adiacenti, cioè quella che precede e quella che segue.

L'ordine logico con cui si dispongono le stazioni prende il nome di anello (ring).

Per poter comunicare, una stazione deve essere in possesso del gettone (o token), consistente in un apposito pacchetto che gli viene consegnato dalla stazione che la precede. ▶
Se la stazione che possiede il token ha necessità di trasmettere, immetterà sul canale comune un pacchetto nel quale indicherà l'indirizzo della stazione destinataria: nell'esempio A è abilitato a trasmettere (ha il token) e invia un messaggio a B, dopodiché aspetta che il pacchetto percorra tutto l'anello, in modo da essere sicuro che giunga a destinazione: quando ha terminato di trasmettere tutti i dati o dopo un certo tempo massimo, cede il token al suo vicino, cioè B, che inizierà a sua volta la trasmissione del suo messaggio.



Le stazioni sono perennemente in lettura sull'anello e quando riconoscono in un pacchetto il loro indirizzo se ne impossessano.

Accesso multiplo con contesa: metodi casuali

Nelle tecniche con contesa l'accesso del canale non viene regolato da nessuna gestione centralizzata o suddivisione preventiva del canale: le stazioni che devono trasmettere accedono al canale e "provano" a trasmettere immettendo il loro messaggio, senza preoccuparsi delle altre stazioni. Inevitabilmente se due stazioni trasmettono contemporaneamente avvengono delle sovrapposizioni di segnale che prendono il nome di collisioni: entrambe si sospendono, attendono un intervallo di tempo casuale (generato da un apposito algoritmo) e successivamente riprovano a trasmettere "sperando" di trovare libero il canale.
In alcuni protocolli la stazione che deve trasmettere prima di immettere il suo messaggio "testa" la linea per vedere se è libera o occupata; in tal caso sospende la sua attività e riprova successivamente.

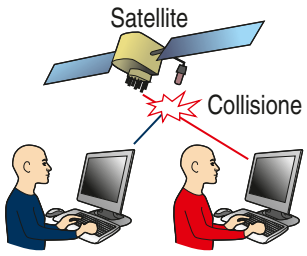
Dato che gli accessi sono casuali, per la realizzazione di questa tecnica è sempre necessario garantire un meccanismo che escluda la possibilità di blocco e permettere che, prima o poi, tutte le stazioni possano comunicare.

Le tecniche di accesso multiplo casuali più note sono:
- Aloha (puro e slotted);
- CSMA/CD (*Carrier Sense Multiple Access/Collision Detection*).

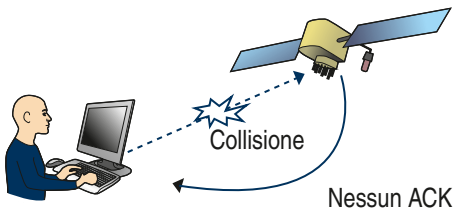
Aloha

Questo protocollo prende il nome dalla rete Aloha, realizzata presso l'Università delle Hawaii nel 1971 (aloha in hawaiano significa hello) per coprire le esigenze di comunicazione tra le isole dell'arcipelago delle Hawaii.

Il meccanismo che sta alla base del protocollo Aloha è quello di trasmissione con attesa di conferma di ricezione da parte del ricevente.
Il mittente inizia a trasmettere senza curarsi degli altri e quindi senza controllare la disponibilità del canale: se la trasmissione si sovrappone anche solo parzialmente a quella di un altro host avviene la collisione. ▶
In questo caso il destinatario non riceverà il messaggio e quindi non potrà inviare nessuna conferma di ricezione (◀ ACK ▶) al mittente che, al termine del tempo di attesa (pari al tempo di andata e ritorno), si accinge a ripetere la trasmissione, eventualmente introducendo un



tempo di attesa affinché si liberi il canale trasmissivo (il tempo di ritardo viene generato da un apposito algoritmo, chiamato di backoff). ▶

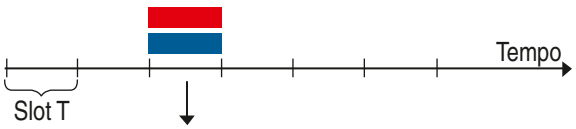


◀ ACK è l'abbreviazione di acknowledge (riconoscere) ed è il simbolo che identifica un segnale emesso in risposta alla ricezione di un'informazione. ▶

Aloha slotted

Un miglioramento del protocollo Aloha viene realizzato introducendo una suddivisione del tempo (*time slot*) in parti di lunghezza uguale al tempo necessario per effettuare la trasmissione di un pacchetto: una stazione inizia a trasmettere in corrispondenza all'inizio del tempo di slot e quindi la collisione avviene solo se due stazioni provano a trasmettere nello stesso slot.

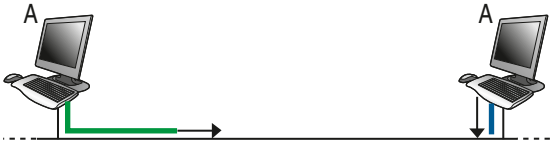
Questo protocollo è molto più efficiente dell'Aloha puro, ma richiede un meccanismo di sincronizzazione tra le stazioni che comporta una complessità e un costo maggiore delle stesse.



CSMA

Il CSMA, o accesso multiplo con ascolto della portante, richiede a ogni stazione di ascoltare il canale prima di iniziare una trasmissione per verificare se è libero: in questo modo non vengono completamente eliminate le collisioni in quanto è necessario tener conto dei tempi di propagazione del segnale tra due stazioni. Il nuovo host B che inizia a trasmettere potrebbe sentire la linea libera mentre invece, come si vede dal seguente disegno, è già occupata da A anche se B non ha ancora ricevuto il messaggio inviato da A.

A sta trasmettendo: B ascolta il canale e, sentendolo libero perché il segnale di A non gli è ancora arrivato, inizia a sua volta a trasmettere. ▶

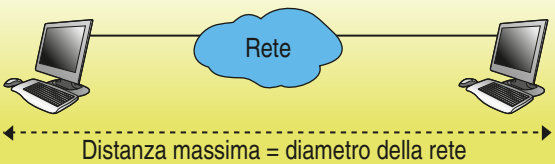


Nelle reti che utilizzano questo protocollo acquistano fondamentale importanza i ritardi di propagazione: infatti una stazione si accorge che la linea è occupata solo quando il segnale gli arriva, cioè trascorso il tempo necessario alla propagazione sulla linea.

Nella tecnica CSMA esiste una "finestra temporale" di collisione che corrisponde al tempo necessario per percorrere la distanza massima tra due stazioni: questo è il motivo per cui il CSMA viene utilizzato in reti di piccole dimensioni e con lunghezza di pacchetti elevata.

DIAMETRO DELLA RETE

Con diametro della rete si indica la massima distanza esistente tra due host presenti sulla rete stessa. ▶



Esistono diverse varianti del CSMA:
a) 1-persistent CSMA;
b) Non-persistent CSMA;
c) P-persistent CSMA;
d) CSMA/CD (Collision Detection) CSMA con rivelazione delle collisioni.

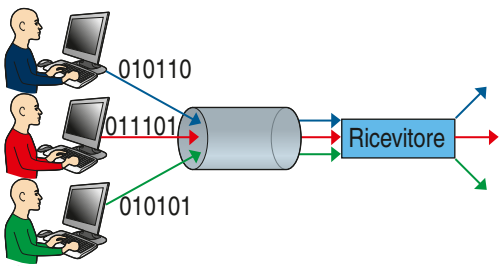
Descriviamo brevemente solo l'ultima, che viene utilizzata nella rete Ethernet e alla quale dedichiamo un'intera Unità didattica.

Nel CSMA/CD, a differenza delle prime tre varianti dove le stazioni non si accorgono quando avviene una collisione e quindi continuano a trasmettere tutto il pacchetto inutilmente, viene eliminato questo spreco di banda dotando ogni stazione di un hardware che permetta di rilevare una collisione. In sintesi il meccanismo è il seguente:
a) ogni stazione ascolta il canale anche quando sta trasmettendo;
b) se si accorge di una collisione, cessa immediatamente la trasmissione e invia sul mezzo un segnale di disturbo (segnale di jamming) per occupare la linea e comunicare alle altre stazioni che il canale è occupato;
c) si mette in attesa per un periodo di tempo casuale (backoff) prima di ritentare l'accesso.

Come vedremo, con un opportuno dimensionamento della linea e del pacchetto trasmesso in funzione della velocità di trasferimento dei dati, è possibile garantire il corretto trasferimento dei dati in ogni situazione di collisione.

CDMA

Nel protocollo CDMA a ogni terminale viene associato un opportuno codice (chiamato chip) che identifica l'utente. Con esso vengono codificati e decodificati messaggi che le stazioni continuano a inviare in sovrapposizione e interferenza. ▶



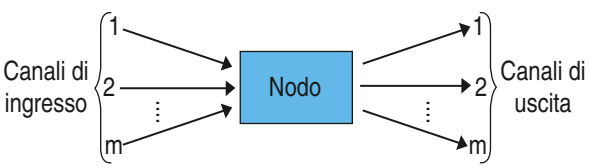
La commutazione (switching)

Per trasferire i dati dal nodo sorgente al nodo di destinazione vengono utilizzate particolari tecniche di commutazione (switching) nei nodi interni alla rete in modo da gestire la trasmissione delle singole unità di informazione (UI). La commutazione ha come fine quello di trasferire (inoltrare) i dati che sono presenti sui canali di ingresso di un nodo verso uno specifico canale di uscita, in modo tale che l'UI possa procedere verso la destinazione finale.

COMMUTAZIONE

La commutazione definisce la *strategia* secondo la quale i dati provenienti da ciascun specifico canale di ingresso al nodo sono trasferiti a uno specifico canale di uscita del nodo. Si definisce nodo di commutazione un qualsiasi nodo di rete che realizzi l'operazione di commutazione.

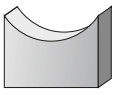
Nell'esempio della figura a lato il nodo ha m canali in ingresso e n canali in uscita: deve quindi trasferire l'UI in arrivo da una linea di IN verso la giusta linea di OUT effettuando la commutazione: ▶



Questa operazione viene realizzata in due "momenti" che vengono chiamati:
- inoltro: con questo termine si intendono le operazioni che vengono effettuate sul dato in ingresso per decidere su quale canale di uscita deve essere inoltrato;
- attraversamento: è la funzione che "fisicamente" trasferisce il dato dal canale di ingresso allo specifico canale di uscita.

I nodi di commutazione vengono classificati, a seconda delle modalità con cui effettuano l'operazione di commutazione, in:
- bridge;
- router.

Graficamente vengono rappresentati negli schemi con i seguenti simboli:


BRIDGE

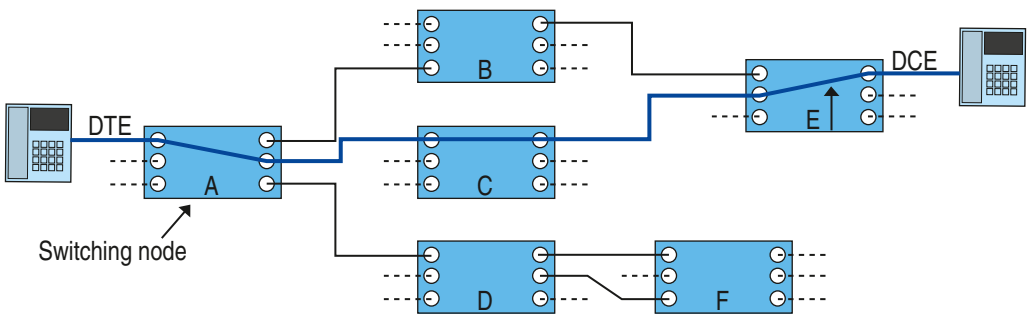

ROUTER

Esistono tre tecniche di commutazione, in base alle quali le reti sono state suddivise in:
- reti a commutazione di circuito (circuit switching);
- reti a commutazione di messaggio (message switching);
- reti a commutazione di pacchetto (packet switching).

Commutazione di circuito

Il circuit switching è la tecnica usata nella rete telefonica: consiste nel creare un percorso tra il nodo chiamante e il nodo chiamato prima di iniziare la comunicazione per poi mantenerlo per tutta la durata della comunicazione.
Una rete che utilizza il modo di trasferimento "a circuito" mette a disposizione di ciascuna coppia di applicazioni comunicanti un circuito fisico che rimane a esse dedicato per tutta la durata della comunicazione.

Nella figura seguente è riportato lo schema di una rete a commutazione di circuito dove sono indicati con:
- **DTE** (*Data Terminal Equipment*): la porta della stazione dati che si interfaccia con la rete;
- **DCE** (*Data Circuit Termination Equipment*): la porta della rete che si interfaccia verso la stazione dati;
- **A**, **B**, **C**, **D**, **E**, **F**: i nodi di commutazione;
- ——: la connessione fisica.



I nodi interni alla rete sono semplicemente degli autocommutatori che cortocircuitano una porta di ingresso con una porta di uscita.

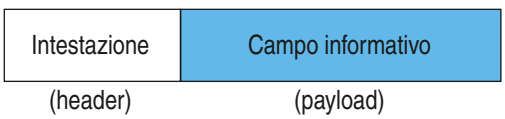
Il modo di trasferimento "a circuito" offre un servizio di trasferimento con una strategia di multiplazione di tipo "statico" dove possiamo individuare le specifiche fasi che caratterizzano il servizio:
- fase di instaurazione della connessione;
- fase di trasferimento dei dati;
- fase di abbattimento della connessione.

Commutazione di messaggio

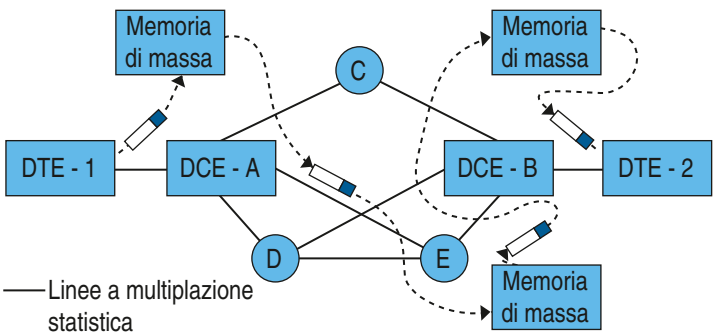
Nel message switching tra trasmettitore e ricevitore viene stabilita una connessione logica e non fisica in quanto il collegamento fisico può variare a seconda dello stato della rete: può quindi cambiare ogni volta il percorso che viene compiuto dai dati.
Le comunicazione viene "spezzata" in componenti elementari autonomi (messaggi) che vengono instradati volta per volta dai nodi di commutazione: quindi "un particolare circuito" dura solo il tempo necessario per la trasmissione del singolo messaggio; questa tecnica è vantaggiosa quando è necessario trasmettere messaggi brevi rispetto ai periodi di assenza di trasmissione.

Il messaggio è composto da due parti:
- un'intestazione (header): ha una lunghezza fissa e contiene gli indirizzi della stazione trasmittente e ricevente e tutte le informazioni necessarie per trasmettere il messaggio;
- un campo informativo (payload): ha una lunghezza variabile e contiene i dati da trasferire.

Il terminale dati DTE invia il messaggio contenente l'indirizzo della stazione chiamante, della stazione chiamata e i dati da trasmettere ai nodi di commutazione, che dapprima lo memorizzano in memorie permanenti e successivamente lo trasmettono.



Questa modalità di funzionamento prende anche il nome di *store and forward* (accumulo e invio).



I messaggi tra due DTE possono seguire percorsi diversi: inoltre nella commutazione di messaggio non è necessaria una fase preventiva di instaurazione della connessione e quindi neppure una fase di abbattimento della connessione.

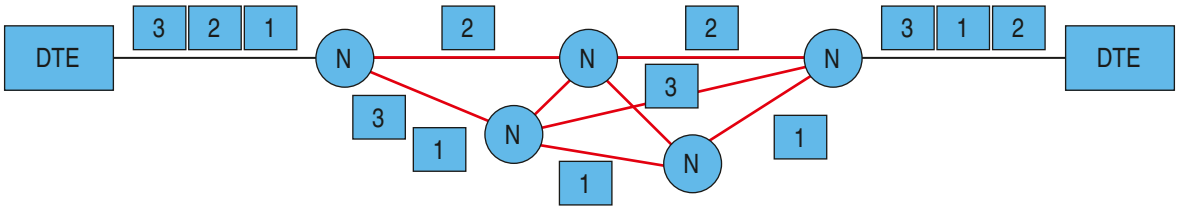
Commutazione di pacchetto

Le packet switching sono un'evoluzione di quelle a commutazione di messaggio dove ogni messaggio viene suddiviso in pacchetti numerati progressivamente e questo numero di sequenza viene aggiunto all'intestazione assieme agli indirizzi della stazione trasmittente e ricevente.

Possiamo distinguere due tipologie di reti a commutazione di pacchetto:
- di tipo datagram;
- a canali virtuali.

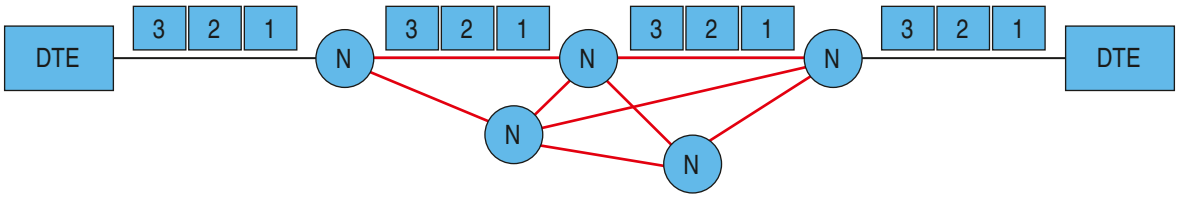
Nella commutazione di pacchetto di tipo datagram i pacchetti ottenuti da un messaggio possono seguire percorsi diversi, dato che ogni singolo pacchetto è instradato indipendentemente dagli

altri e può quindi arrivare al ricevitore in ordine diverso rispetto a quello con cui è stato spedito: sarà compito del ricevitore ristabilire il messaggio ordinando i pacchetti una volta che li ha acquisiti tutti.



Tra gli esempi di reti a commutazione di pacchetto di tipo datagram ricordiamo ARPANET (USA) e DATAPAC (Canada).

Nella commutazione di pacchetto a canali virtuali i pacchetti ottenuti da un messaggio seguono tutti *lo stesso percorso*, che viene stabilito prima della trasmissione: quindi vengono ricevuti nello stesso ordine con cui sono stati trasmessi.



A differenza del circuit switching, il circuito che si viene a creare tra mittente e destinatario è virtuale e non fisico e quindi non è dedicato a una sola connessione: le risorse trasmissive vengono utilizzate solo al momento del bisogno attraverso una politica di gestione dinamica, molto utile in condizioni di congestione della rete (multiplexing statistico).

Esempi di reti pubbliche che utilizzano i circuiti virtuali sono ITAPAC in Italia, TRANSPAC in Francia, TELENET e TYMNET in USA.

Zoom su...

RETI DATAGRAMMI "CONTRO" RETI A CIRCUITI VIRTUALI

Vantaggi offerti dalla rete di tipo datagram:

1. non è necessario stabilire un collegamento iniziale, quindi è preferibile per piccole comunicazioni;
2. consente di effettuare collegamenti senza connessioni;
3. è più sicura, in quanto se "cade" un nodo si instaura un percorso alternativo per i datagrammi.

Vantaggi offerti dalla rete a circuiti virtuali:

1. l'individuazione del percorso viene effettuata solo all'inizio del collegamento;
2. i pacchetti sono ricevuti sempre nello stesso ordine in cui sono stati generati;
3. fornisce un servizio orientato alla connessione.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Gli elementi di un sistema di trasmissione sono:
- ○ trasmettitore
- ○ messaggio
- ○ canale
- ○ ricevitore

2 A seconda della modalità di utilizzo dei canali di comunicazione la trasmissione dei dati è:
- ○ simplex
- ○ half-simplex
- ○ full-simplex
- ○ duplex
- ○ half-duplex
- ○ full-duplex

3 Un modo di trasferimento è individuato dalle seguenti componenti:
- ○ modalità di trasmissione
- ○ tecnica di multiplazione
- ○ modalità di accesso al canale
- ○ tecnica di commutazione

4 Le risorse di trasmissione sono:
- ○ banda
- ○ protocollo
- ○ tempo
- ○ canale

5 Esistono due modalità di assegnamento dinamico:
- ○ assegnazione a domanda
- ○ assegnazione a tempo
- ○ preassegnazione
- ○ postassegnazione

6 CSMA/CD significa:
- ○ *Carrier Signal Multiple Access/Collision Detection*
- ○ *Carrier Sense Multiple Access/Collision Detection*
- ○ *Common Signal Multiple Access/Collision Detection*
- ○ *Common Sense Multiple Access/Collision Detection*

7 Quale tra le seguenti non è una tecnica di accesso multiplo?
- ○ deterministico senza contesa
- ○ deterministico con contesa
- ○ ad accesso casuale
- ○ CDMA

8 Le tecniche di commutazione sono (*indica la risposta errata*):
- ○ a commutazione di circuito
- ○ a commutazione di dispositivo
- ○ a commutazione di messaggio
- ○ a commutazione di pacchetto

Test vero/falso

1	Le modalità di comunicazione con connessione si chiama anche connectionless.	V F
2	Un esempio di funzionamento con modalità connection-oriented è la telefonata.	V F
3	Un esempio di funzionamento con modalità connection-oriented è il servizio postale.	V F
4	La multiplazione è l'implementazione del concetto di condivisione di risorse trasmissive.	V F
5	Nella multiplazione dinamica la suddivisione del canale avviene in parti fisse in tanti canali fisici.	V F
6	Il CDMA è un protocollo ad accesso casuale a contesa.	V F
7	Nel TDMA il trasmittente deve essere sincronizzato con il ricevente.	V F
8	L'FDMA è un protocollo di accesso multiplo di tipo deterministico a divisione di tempo.	V F
9	La tecnica di passaggio del testimone è una tecnica con contesa.	V F
10	Lo *store and forward* è tipico della commutazione di circuito.	V F

UNITÀ DIDATTICA 3

L'ARCHITETTURA A STRATI ISO-OSI E TCP-IP

IN QUESTA UNITÀ IMPAREREMO...

- il concetto di architettura stratificata
- l'architettura ISO-OSI
- il modello Internet TCP-IP

Generalità

I sistemi di comunicazione sono composti da un insieme di dispositivi che devono cooperare, cioè collaborare per il conseguimento di uno scopo comune.
Negli anni '70 i dispositivi erano costruiti da un numero limitato di aziende che realizzavano hardware e software con l'obiettivo di utilizzare esclusivamente i loro prodotti senza curarsi della comunicazione con sistemi di terze parti: si realizzarono quelli che furono in seguito identificati come sistemi chiusi (*closed system*).
Con il passare del tempo le piccole reti private aziendali non bastarono più: nacque la necessità di collegare tra loro dispositivi anche a media distanza, per esempio tra due sedi delle stessa azienda poste in città differenti, e quindi i sistemi chiusi dovettero connettersi a sistemi e impianti di comunicazione, sia privati che pubblici, di altri produttori e/o gestori.

Affinché sistemi diversi possano colloquiare per cooperare è necessario che utilizzino le stesse regole procedurali per effettuare il trasferimento delle informazioni.

L'ISO (*International Standards Organization*) iniziò un processo di standardizzazione proponendo un modello di riferimento chiamato *Open System Interconnection reference model* (modello di riferimento OSI) con l'obiettivo di definire le regole comuni affinché i processi applicativi, residenti su computer di case costruttrici diverse, potessero comunicare tra loro:

- venne data una descrizione astratta delle modalità di comunicazione tra due o più dispositivi fisicamente posti in rete in luoghi remoti, realizzando un modello funzionale che è di riferimento per rappresentare l'*ambiente di comunicazione*;
- venne definita l'architettura della comunicazione del tipo a strati (*layered architecture*) stabilendo i compiti e le attività di pertinenza di ciascuno stato e le modalità di interazione tra i diversi strati, sia in orizzontale, cioè sullo stesso livello, sia in verticale, tra strati inferiori e superiori.

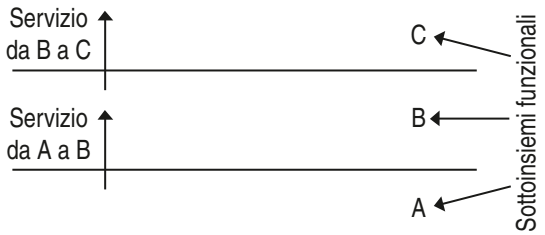
L'architettura a strati

In ogni architettura a strati le funzioni svolte dal sistema vengono organizzate in modo da creare un'*indipendenza tecnologica* tra i diversi strati, per permettere l'evoluzione indipendente di ciascun processo di comunicazione.

Ogni strato raggruppa alcune attività che sono organizzate secondo criteri gerarchici: allo stesso sottosistema appartengono funzioni simili per logica e per tecnologia realizzativa, che possono interagire tra loro con il minimo utilizzo di risorse in modo da ottenere la semplificazione delle interazioni tra sottosistemi diversi.

ESEMPIO

Nel sistema a lato sono stati individuati tre sottosistemi funzionali, identificati nei tre strati A B e C: ▶



Viene definita una gerarchia tra gli strati.

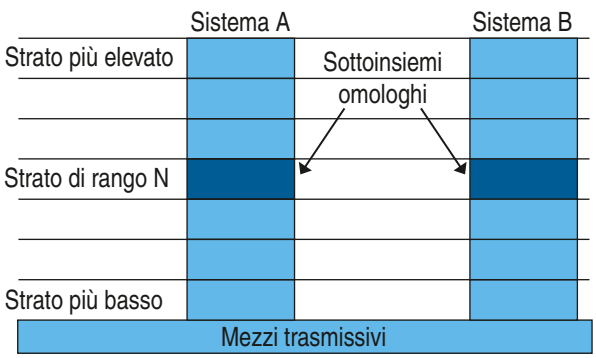
GERARCHIA

Gli strati sono gerarchicamente in ordine crescente se lo svolgimento delle funzioni di B richiede preventivamente lo svolgimento delle funzioni di A e l'insieme di A e B è il presupposto per la realizzazione dei compiti dello strato C.

La funzione (o servizio) offerta dal livello B si basa su un'elaborazione preventiva fatta dal livello A e gli aggiunge funzionalità di sua competenza, arricchendolo, per poi restituirlo come "base" al livello C.

Secondo questo schema di funzionamento ogni livello scambia informazioni con gli elementi dello stesso livello appartenenti al proprio sistema, indipendentemente dagli altri componenti dello stesso livello; interagisce quindi con i livelli gerarchicamente adiacenti non appena termina l'elaborazione di sua competenza, cioè quando ha "arricchito il servizio" mediante le funzioni che gli competono.
Lo schema di un sistema architetturale a livelli è quello raffigurato a lato: ▶

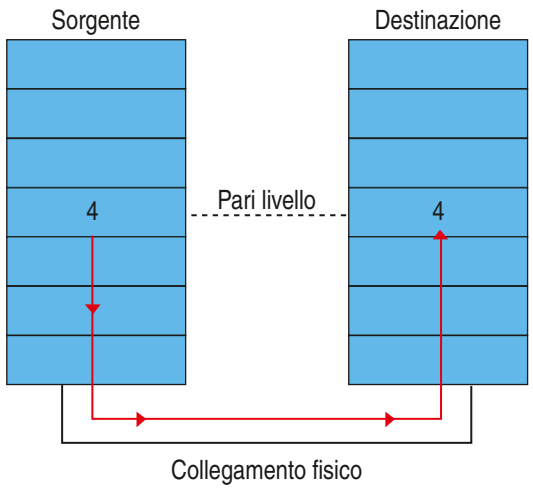
Questa metodologia di lavoro prende anche il nome di principio del valore aggiunto.



Possiamo individuare tre elementi fondamentali:
a) allo strato più basso è collocato il sistema fisico, cioè il mezzo trasmissivo di interconnessione tra i due sistemi;
b) ogni sistema ha la stessa suddivisione, cioè lo stesso numero e tipo di strati;
c) sottosistemi omologhi occupano lo stesso livello n-esimo nelle due pile architetturali.

Per definire un'architettura a livelli è necessario definire:
- gli *strati architetturali*;
- i *servizi di strato*;
- le *primitive di servizio* e i *protocolli di strato*;
- la *funzione di indirizzamento*.

Affinché i dati possano arrivare dalla sorgente alla destinazione è necessario che i due sistemi usino lo stesso protocollo: se per esempio una funzione del livello 4 del primo dispositivo (sorgente) deve comunicare con la stessa funzione del dispositivo destinazione, dapprima svolge le funzioni di sua competenza e successivamente con il protocollo specifico di livello "passa" i dati al livello inferiore a lui adiacente e così via, livello per livello, fino a giungere al livello fisico: nel dispositivo di destinazione le operazioni sono svolte in ordine inverso, dal basso verso l'alto. ▶
Regole e servizi non sono statici, in quanto la tecnologia è in continua evoluzione: nuove apparecchiature si devono integrare con architetture esistenti e nuovi servizi devono essere offerti agli utenti sia su vecchie piattaforme che su nuovi dispositivi.



Il modello ISO-OSI viene "mantenuto" e aggiornato costantemente da diverse organizzazioni, per esempio IEEE (*Institute of Electrical and Electronic Engineers*), ANSI (*American National Standards Institute*), TIA (*Telecommunications Industry Association*), EIA (*Electronic Industries Alliance*), ITU (*International Telecommunication Union*), CCITT (*Comité Consultatif International Telephonique et Telegraphique*).

Il modello OSI

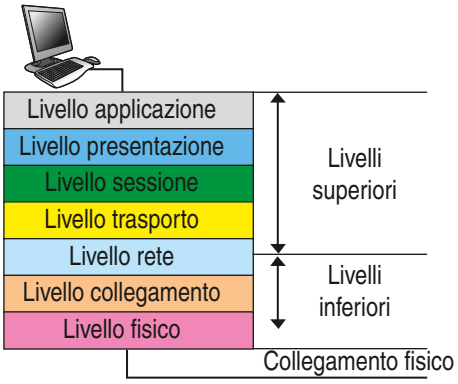
Lo standard OSI definisce un modello di riferimento per lo scambio di informazioni tra due calcolatori, e dal nome del progetto si individuano gli obiettivi: *Open System Interconnection*, cioè interconnessione di sistemi aperti.

È stato realizzato negli anni '70 affinché tutti i produttori di tecnologie proprietarie potessero riconoscersi in questa architettura e iniziassero ad "aprirsi" verso altri produttori per realizzare sistemi "distribuiti ibridi".

Il modello OSI è nato quindi per fornire una base comune per la realizzazione di standard nel settore dell'interconnessione di sistemi informatici e di telecomunicazione, in modo da facilitare il dialogo tra apparati prodotti da aziende diverse al fine di realizzare una rete aperta e trasparente per l'utente.

Lo standard OSI utilizza una struttura composta da sette livelli o strati che sono tra loro suddivisi logicamente in due gruppi: ▶

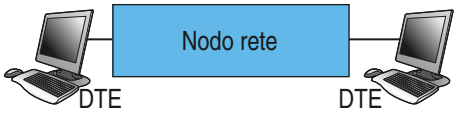
- i livelli inferiori: sono quelli che si occupano della comunicazione e del trasferimento dell'informazione tra i due sistemi e sono presenti all'interno dei nodi di rete (dispositivi intermedi);
- i livelli superiori: sono quelli orientati all'applicazione.



In un sistema di comunicazione un dispositivo può essere terminale o intermedio:

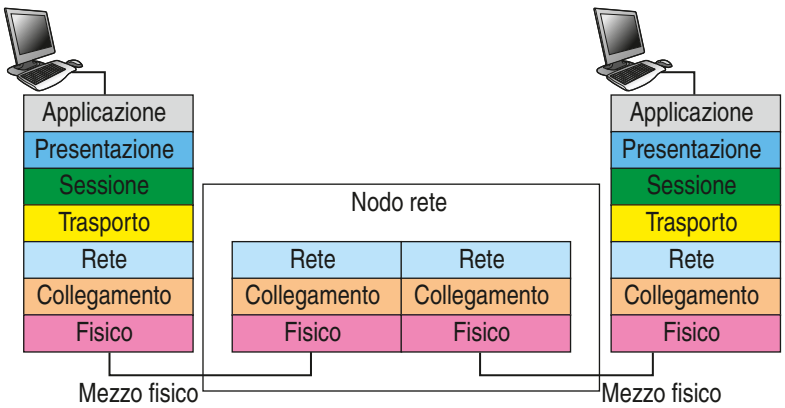
- un sistema terminale (*end system*) è origine o destinazione finale delle informazioni;
- un sistema intermedio (*relay system*) provvede ad assicurare il collegamento fisico o logico tra due o più sistemi terminali.

Se prendiamo per esempio uno schema di trasmissione tra due PC (DTE, *Data Terminal Equipment*) sul mezzo trasmissivo sono presenti dei nodi (dispositivi intermedi). ▶



Nel nodo di rete sono presenti solo i primi tre livelli OSI mentre i DTE devono contenere tutti i livelli OSI.

Quindi a livello funzionale lo schema precedente viene così "trasformato" e sarà il nostro modello di riferimento: descriviamo brevemente i compiti di ogni livello, che saranno poi ciascuno oggetto di una (o più) Unità didattiche.



Livello fisico o physical layer

Il livello fisico definisce le caratteristiche dei segnali e dei dispositivi necessari per connettere due o più DTE mediante il mezzo trasmissivo, visto come un canale pronto a trasportare segnali elettrici (oppure ottici), prodotti della "trasformazione" fisica dell'informazione che i due DTE devono trasmettersi. ▶

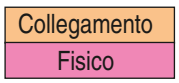
Fisico

I messaggi che i DTE devono trasmettere sono una sequenza di bit e per essere immessi sul sistema fisico necessitano di un dispositivo di interfacciamento (DCE, *Data Communication Equipment*), che generalmente è una porta seriale.

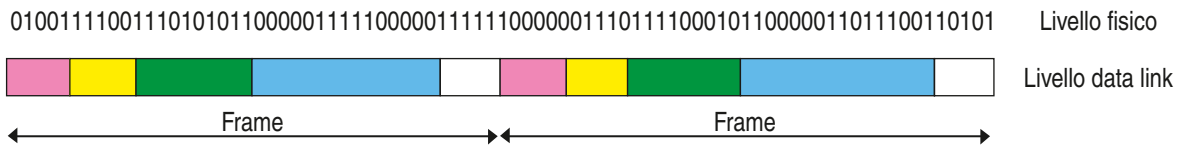
È quindi necessario che vengano definite:
- le caratteristiche meccaniche: l'interfaccia tra il DTE e il DCE con cui si può realizzare l'interconnessione tra un sistema di calcolo e il canale;
- le caratteristiche elettriche dei segnali: le forme d'onda utilizzate per trasmettere i simboli, la durata temporale di ciascun simbolo;
- le regole per l'attivazione e la disattivazione del collegamento fisico tra due punti;
- le caratteristiche dei cavi e dei connettori;
- le operazioni di multiplazione e modulazione dei segnali.

Livello di collegamento o data link

Il servizio fornito dal livello fisico è quello di trasferire sequenze di bit da una parte all'altra del collegamento, senza curarsi di risolvere i problemi legati ai disturbi e quindi agli eventuali errori dovuti all'interferenza dei campi magnetici ed elettrici: non viene dato alcun significato ai bit, che risultano essere l'unità informativa del livello 1. ▶



Il livello immediatamente superiore, il livello di data link, all'interno della sequenza di bit trasmessa dal livello fisico si occupa di definire la struttura del messaggio dividendolo in frame (o trame), individuando dove queste iniziano e dove finiscono, scomponendo ogni frame in campi, ciascuno con un proprio significato e un proprio compito.



A questo livello si inizia a dare al messaggio un senso compiuto per poter poi effettuare le operazioni che il protocollo di collegamento (anche chiamato protocollo di linea) intende svolgere sul messaggio stesso.

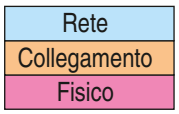
Il protocollo di linea, dopo aver individuato il messaggio, deve fornire un insieme di procedure che permettano di capire se tale messaggio è stato "danneggiato" e se quindi siano presenti degli errori dovuti alla trasmissione: in questo caso deve prevedere dei meccanismi tali per cui le due stazioni possano ritrasmettere il messaggio e correggere gli errori.

Il "valore aggiunto" offerto da questo livello è quello di trasformare bit privi di significato in "messaggi affidabili", cioè in messaggi liberati dagli errori, pronti per essere trasferiti al livello superiore.

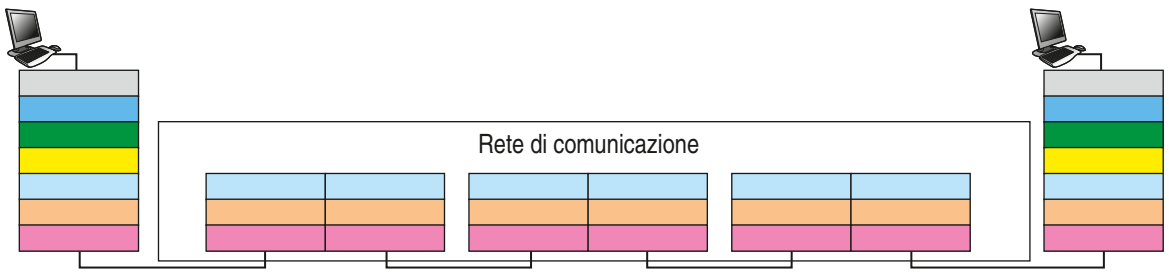
In sintesi, le principali funzioni svolte dal livello di collegamento sono le seguenti:
- suddividere i bit forniti dal livello fisico in frame;
- individuare la presenza di errori nei frame e gestire i meccanismi per correggerli;
- definire l'accesso multiplo da parte di diversi utenti allo stesso canale di comunicazione;
- regolare la trasmissione tra dispositivi che lavorano a velocità diverse;
- … oltre a fornire servizi al livello superiore (livello di rete).

Il protocollo più utilizzato a livello di collegamento è l'HDLC (*High Level Data Link Control*).

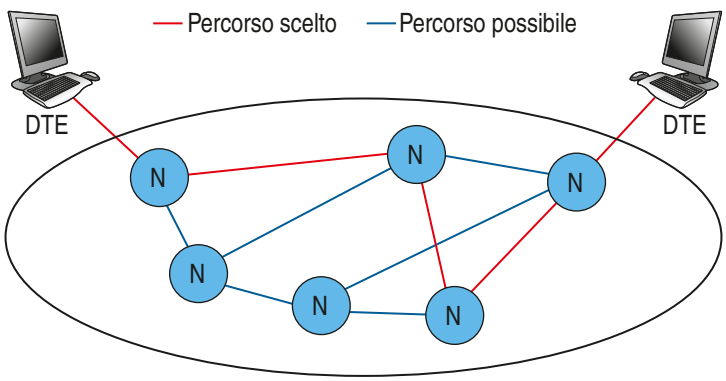
Livello di rete o network layer



Il messaggio deve essere trasferito da un mittente a un destinatario: è quindi necessario che, in base all'indirizzo dei due DTE interessati, il messaggio attraversi tutta la rete per giungere a destinazione creando un canale "di transito" mediante tecniche di instradamento e commutazione. ▶
Il livello di rete permette il colloquio tra una stazione host e un dispositivo intermedio di rete e tra coppie di nodi intermedi: l'utente vede la rete di comunicazione come un unico canale che collega la sorgente alla destinazione.



All'aumentare del numero di reti connesse che si frappongono tra sorgente e destinazione cresce la possibilità di realizzare più percorsi alternativi per mettere in collegamento i due host: è compito del livello di rete definire il percorso che i dati devono percorrere nella rete di comunicazione per andare dal DTE di trasmissione al DTE in ricezione tra tutti quelli possibili.

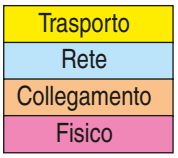


Questi sono i principali compiti del livello di rete e prendono il nome di routing, una funzionalità necessaria quando trasmittente e destinatario non appartengono alla stessa rete locale LAN ma sono connessi tra loro da una maglia di nodi intermedi.

Inoltre i percorsi potrebbero anche essere realizzati su reti di natura diversa che utilizzino diversi protocolli di linea: è necessario avere un protocollo a un livello superiore in grado di uniformare tutte le situazioni provenienti dagli strati inferiori per avere un unico tipo di messaggio indipendente dalle caratteristiche della singola rete che viene utilizzata. Questa funzionalità è detta internetworking.

In sintesi, le principali funzioni svolte dal livello di rete sono le seguenti:
- multiplazione di due o più flussi di dati sullo stesso circuito fisico;
- instradamento (routing) dei dati dalla stazione di partenza a quella di arrivo;
- controllo della congestione della rete;
- interconnessione di reti (internetworking);
- … oltre a fornire servizi al livello superiore (di trasporto).

Livello di trasporto o transport layer



Il messaggio che l'host sorgente deve trasmettere generalmente viene scomposto in segmenti che, numerati progressivamente, vengono messi in sequenza sulle rete: a causa della dinamicità del traffico non è detto che tutti percorrano lo stesso canale e quindi arrivino a destinazione nello stesso ordine con cui sono partiti. ▶

È compito del livello di trasporto effettuare la ricostruzione esatta dei dati dell'utente rimuovendo le cause di possibile errore: esso si occupa quindi di rendere affidabile la trasmissione accorgendosi di eventuali guasti sulla rete, in quanto, al momento della ricostruzione del messaggio, è in grado di accorgersi dell'eventuale perdita di una sua parte e quindi di intervenire.

> Il transport layer è un protocollo end-to-end che non vede elementi di rete e di fatto può essere definito come un protocollo di cooperazione.

In sintesi, le principali funzioni svolte dal livello di trasporto sono le seguenti:

- segmentazione e assemblaggio dei dati;
- controllo end-to-end dei dati per prevenire errori e malfunzionamenti e fornire perciò un servizio affidabile al livello di sessione;
- definizione della qualità del servizio;
- … oltre ai servizi offerti al livello superiore (di sessione).

Livello di sessione o session layer

Nella comunicazione tra due sistemi non è sufficiente garantire che il messaggio trasmesso dal mittente giunga a destinazione, ma è necessario gestire completamente il dialogo: vengono create applicazioni che prendono il nome di sessioni di lavoro, le quali possono anche coinvolgere più host. ▶

Supponiamo per esempio di dover memorizzare dei dati su un database distribuito: è necessario che nessun altro host acceda all'archivio finché non sia terminato l'aggiornamento dei dati e quindi il DB non sia nuovamente disponibile: si parla quindi di necessità di sincronizzazione (che viene realizzata con meccanismi a token).

> Un altro esempio di sessione è quello che avviene ogni volta che un utente si collega alla rete effettuando il login (fornendo account e password): ogni singolo login viene anche chiamato sessione di lavoro, e prosegue fino alla disconnessione dell'utente dalla rete.

La gestione delle sessioni viene fatta dal session layer che, nel caso in cui i dispositivi siano di natura diversa, si occupa di negoziare tra i due nodi per fissare i parametri del colloquio (velocità, controllo, errori, tipo di trasferimento simplex, half-duplex o full-duplex ecc.) verificando periodicamente l'efficienza e modificandoli in caso di perdita di prestazioni.

> Una sessione deve essere individuata, eventualmente interrotta e ripresa per far fronte a vari eventi imprevisti indesiderati: perdita di dati, caduta della linea, momentaneo crash di uno dei due interlocutori ecc.

In sintesi, le principali funzioni svolte dal livello di sessione sono le seguenti:

- suddividere il dialogo tra le applicazioni in unità logiche (dette appunto sessioni);
- gestire la chiusura ordinata (*soft*) del dialogo;
- introdurre i cosiddetti punti di sincronizzazione;
- … oltre ai servizi offerti al livello superiore (livello di presentazione).

Livello di presentazione o presentation layer

Il presentation layer si occupa della sintassi e della semantica delle informazioni da trasferire: se due interlocutori utilizzano linguaggi diversi è possibile che interpretino diversamente i dati sia nel tipo sia nel formato. ▶
È stato definito un formato di riferimento (la sintassi dei dati definita dall'*Abstract Syntax Notation*), utilizzato soprattutto nelle prime reti, quando si era "all'inizio" del processo di standardizzazione e c'erano ancora PC che lavoravano con sequenze di 8 bit, altri con sequenze di 16 bit, fino a sequenze strane di 36 bit. Con questo linguaggio comune si effettuava una descrizione per presentare i propri dati a tutti gli altri componenti della rete e metterli in condizione di interpretarli correttamente.

Oggi questo problema è stato superato ma ci possono essere nuovi tipi di problemi, soprattutto se si vuole mandare una struttura dati complessa di tipo multimediale, che contiene il formato testo, il formato immagine e il formato suono: l'informazione deve essere consegnata nella sua interezza ed è quindi necessario far precedere tale struttura dati da un'intestazione nella quale si indichi com'è strutturato il record, cioè la quantità, la natura e la lunghezza di ogni singolo campo che compone la struttura dati.

Il presentation layer si occupa inoltre di svolgere una funzione di sicurezza effettuando il criptaggio dei dati.

In sintesi, le principali funzioni svolte dal livello di presentazione sono le seguenti:
- rappresentazione dei dati;
- compressione dei dati;
- cifratura dei dati;
- … oltre ai servizi offerti al livello superiore (livello applicativo).

Livello applicativo o application layer

Lo strato di applicazione è l'ultimo della "pila", ossia quello a contatto con l'utente della rete di calcolatori che manda in esecuzione un'applicazione e pertanto non deve offrire servizi a nessuno: deve fornire agli utenti il mezzo per accedere alle reti, fungendo da interfaccia tra il sistema informativo e il mondo reale. ▶
Per svolgere i propri compiti il programma applicativo dell'utente ha bisogno di comunicare con altre applicazioni remote e quindi il modello è un processo distribuito: sono stati definiti protocolli specifici per un insieme di applicazioni universali, come il *File Transfer*, l'*accesso a database* ecc., che persino i "difensori delle applicazioni proprietarie" sono costretti a utilizzare se vogliono rimanere in contatto con il resto del sistema.

> Il livello di applicazione contiene quindi tutti i programmi utente o programmi applicativi che consentono all'end user di svolgere le sue attività in rete.

In sintesi, le principali funzioni fornite dal livello di applicativo sono le seguenti:
- trasferimento, accesso e gestione dei file;
- posta elettronica;
- terminale virtuale;
- gestione di messaggi (come la posta elettronica);
- scambio risultati tra programmi (applicazioni client-server).

Il modello Internet o TCP/IP

L'architettura ISO-OSI è il risultato di un lavoro svolto per oltre dieci anni e ha definito un'enorme quantità di protocolli, alcuni dei quali hanno avuto successo e sono effettivamente divenuti degli standard mentre altri sono stati dimenticati.
L'importanza che ha avuto e ha tuttora questo tipo di architettura consiste nell'aver introdotto il concetto di stratificazione e di interfacciamento tra gli stati: la definizione dei protocolli gioca un ruolo fondamentale nel processo di indipendenza tecnologica tra gli strati e anche all'interno degli strati stessi.

Prendendo come riferimento il modello ISO-OSI un gruppo di ricercatori ha definito un modello architetturale semplificato, sempre a strati, cercando di avvicinare la teoria alla realizzazione pratica, per interconnettere alcune reti che erano già esistenti.

Questo modello prende il nome da due dei suoi protocolli ed è l'architettura TCP/IP.

DA ARPANET A TCP/IP

Si è arrivati al modello TCP/IP dall'evoluzione della rete ARPANET, un progetto avviato dal Dipartimento della difesa degli Stati Uniti nel 1964, che dal 1972 divenne un'architettura aperta grazie al contributo di due ricercatori, Kahn e Cerf, che applicarono il modello stratificato definendo nel 1973 il protocollo TCP *Trasmission Control Protocol*. Inizialmente esso includeva IP *Interconnection Protocol* ma successivamente, quando ci si rese conto che non sempre era necessario avere una connessione tra due host, IP venne separato definendo così due protocolli di trasporto:

- TCP *Trasmission Control Protocol*: protocollo orientato alla connessione;
- UDP *User Datagram Protocol*: protocollo non orientato alla connessione;

e delegando al protocollo IP i compiti di livello rete, anche detto livello Internet.
Nel 1983 la rete ARPANET adottò il protocollo TCP/IP e venne separata in due parti: una civile, che mantenne il nome ARPANET, e una militare, con il nome MILNET.

Il modello TCP/IP ha avuto un successo immediato sia per la sua semplicità che per i bassi costi dei suoi dispositivi. Mettiamo a confronto gli strati del modello OSI e quelli del modello TCP/IP: ▶

Non esistono corrispondenze ben definite tra gli strati, come è possibile vedere dalla tabella, ma abbiamo la corrispondenza tra livello inferiore e livello superiore.

Modello OSI	Modello Internet
Applicazione	Applicativo
Presentazione	
Sessione	
Trasporto	Da estremo a estremo
Rete	Internet
Collegamento	Accesso in rete
Fisico	

Accesso in rete

Questo livello non è in realtà specificato rigorosamente dal modello di riferimento TCP/IP, in quanto il protocollo utilizzato viene definito e varia da host a host e da rete a rete: l'importante è che in questo strato "esista un modo" per recapitare i pacchetti prodotti dal livello superiore, il livello Internet; alcune volte a questo livello viene dato il nome di livello di host to network oppure di accesso alla sottorete (SAL, *Subnet Access Layer*).

Internet layer

Il livello di rete Internet layer corrisponde in parte al network layer dell'architettura OSI e ha il compito principale di spedire i pacchetti di informazione verso ogni nodo destinazione presente sulla rete.

Per motivi di robustezza e di affidabilità per il TCP/IP è stato scelto il modello di rete a commutazione di pacchetto del tipo connectionless.

Transport layer

Essendo il protocollo di tipo connectionless è possibile che i pacchetti arrivino a destinazione in ordine differente da come sono stati inviati: è compito di questo livello ricomporre correttamente l'intero messaggio.
Abbiamo già detto in precedenza che sono presenti due protocolli a questo livello:

- il protocollo TCP *Trasmission Control Protocol*, orientato alla connessione, è basato su una comunicazione affidabile a circuito virtuale che effettua il riassemblaggio dei pacchetti e il controllo di flusso per regolare le diverse velocità di elaborazione che possono essere presenti tra mittente e destinatario;
- il protocollo UDP *User Datagram Protocol*, non orientato alla connessione, è basato su una comunicazione a datagramma inaffidabile che non effettua il flow control e il riassemblaggio dei pacchetti: esso viene utilizzato quando è trascurabile la perdita di qualche pacchetto a vantaggio della velocità, come per esempio nelle trasmissioni video in broadcast.

Application layer

Nel livello di applicazione sono inseriti gli applicativi ad alto livello che permettono di risolvere i problemi concreti relativi all'utilizzo della rete: ne riportiamo brevemente di seguito i più conosciuti, dato che saranno oggetto di trattazione più approfondita in seguito:

- TELNET: permette di accedere al proprio PC in remoto, creando un terminale virtuale che "dà l'impressione" di essere a casa propria, sul proprio PC, e di lavorare con i proprio dati e programmi;
- SMTP (*Simple Mail Transfer Protocol*): gestisce i servizi di posta elettronica;
- FTP (*File Transfer Protocol*): permette di scambiare o di copiare file presenti nei server di rete;
- HTTP: protocollo usato per caricare le pagine del World Wide Web;
- DNS (*Domain Name Server*): è un servizio utile a "tradurre" i nomi simbolici in indirizzi IP; digitando per esempio il nome di un sito web permette di trasformarlo nell'indirizzo fisico necessario affinché il PC sul quale sono residenti i file di quel sito possa essere individuato nella rete.

La tabella seguente riporta alcuni esempi di "pile" protocollari OSI e Internet:

<table>
<tr><th>Modello OSI</th><th>Protocolli</th><th colspan="2">Protocolli</th></tr>
<tr><td>Applicazione</td><td rowspan="2">SMB</td><td colspan="2" rowspan="2">SMB</td></tr>
<tr><td>Presentazione</td></tr>
<tr><td>Sessione</td><td rowspan="3">NetBEUI</td><td rowspan="3">NWLINK</td><td>NetBios*</td></tr>
<tr><td>Trasporto</td><td>SPX</td></tr>
<tr><td>Rete</td><td>IPX</td></tr>
<tr><td>Collegamento</td><td rowspan="2">Ethernet</td><td colspan="2">LLC</td></tr>
<tr><td>Fisico</td><td colspan="2">IEEE 802.3</td></tr>
</table>

<table>
<tr><th>Modello Internet</th><th>Protocolli</th></tr>
<tr><td>Applicativo</td><td>HTTP</td></tr>
<tr><td>Da estremo a estremo</td><td>TCP</td></tr>
<tr><td>Internet</td><td>IP</td></tr>
<tr><td rowspan="2">Accesso in rete</td><td>PPP</td></tr>
<tr><td>V.35</td></tr>
</table>

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Qual è il significato di OSI?

- ○ *Open System Internet reference model*
- ○ *Open System Interconnection reference model*
- ○ *Open Standards Interconnection reference model*
- ○ *Open Standards Internet reference model*

2 Per definire un'architettura a livelli è necessario definire:

- ○ gli strati architetturali
- ○ i servizi di strato
- ○ il mezzo trasmissivo
- ○ le primitive di servizio
- ○ i protocolli di strato
- ○ le tipologie dei dispositivi
- ○ la funzione di indirizzamento

3 Disponi in ordine i livelli ISO-OSI:

..... Presentazione
..... Fisico
..... Applicazione
..... Sessione
..... Data link
..... Trasporto
..... Rete

4 Indica quale tra i seguenti acronimi è errato:

- ○ DCN *Data Comunication Network*
- ○ DTE *Data Terminal Equipment*
- ○ ISO *Industrial Standards Organization*
- ○ DCE *Data Communication Equipment*
- ○ IEEE *Institute of Electrical and Electronic Engineers*
- ○ TIA *Telecommunications Industry Alliance*
- ○ EIA *Electronic Industries Alliance*

5 Metti in corrispondenza strato e unità informativa:

1 physical layer .....
2 data link .....
3 network layer .....

a frame
b pacchetto
c bit

6 Metti in corrispondenza gli strati tra ISO-OSI e TCP-IP:

.....	Presentazione	A Transport layer
.....	Fisico	B Application layer
.....	Applicazione	C Accesso in rete
.....	Sessione	D Internet layer
.....	Data link	
.....	Trasporto	
.....	Rete	

Test vero/falso

1	Il modello ISO-OSI è di tipo gerarchico.	V F
2	Nel modello ISO-OSI si realizza il principio del valore aggiunto.	V F
3	Nella pila ISO-OSI lo strato di trasporto è tra i livelli inferiori.	V F
4	Nella pila ISO-OSI lo strato di rete è tra i livelli inferiori.	V F
5	Un relay system provvede ad assicurare il collegamento fisico o logico.	V F
6	Nei relay system sono presenti solo funzionalità dei primi tre livelli.	V F
7	Il DCE è generalmente una porta seriale.	V F
8	A livello 1 l'unità informativa è il byte.	V F
9	Il protocollo di linea è in grado di individuare gli errori di trasmissione.	V F
10	Il transport layer è un protocollo end-to-end che vede elementi di rete.	V F

4 MODULO DISPOSITIVI PER LA REALIZZAZIONE DI RETI LOCALI

OBIETTIVI

- Conoscere la modalità di trasmissione di segnali elettrici via cavo
- Apprendere gli strumenti e le tecniche di test sui cavi
- Conoscere la modalità di trasmissione di segnali ottici in fibra
- Apprendere gli strumenti e le tecniche di test sulle fibre
- Conoscere la modalità di trasmissione dei segnali wireless
- Individuare le problematiche connesse alla sicurezza nelle comunicazioni wireless
- Conoscere la normativa americana standard EIA/TIA 568
- Conoscere la normativa europea ISO/IEC DIS 11801

ATTIVITÀ

- Crimpare un cavo diretto e un cavo incrociato
- Trasformate un cavo diretto in un cavo incrociato
- Effettuare i principali test sui cavi in rame
- Effettuare i principali test sulle fibre ottiche
- Utilizzare la terminologia dei componenti dei cablaggi strutturati
- Progettare il cablaggio strutturato di un edificio
- Progettare il cablaggio strutturato di un campus

UNITÀ DIDATTICA 1

LA CONNESSIONE CON I CAVI IN RAME

IN QUESTA UNITÀ IMPAREREMO...

- gli elementi fondamentali di elettrologia
- gli elementi fondamentali della trasmissione di segnali elettrici via cavo

Generalità sulle connessioni

In ogni tipo di sistema i dispositivi presenti devono scambiarsi dei dati in modo che ogni componente possa eseguire il proprio compito producendo i risultati desiderati.
I dispositivi possono essere comunque complessi ed essere fisicamente vicini, anche posti all'interno dello stesso contenitore (◀ chassis ▶) oppure possono trovarsi a distanze anche considerevoli, da centinaia di metri fino a centinaia di chilometri.

◀ Il termine francese chassis identifica un telaio. Nel caso di un computer si tratta in genere del case o comunque dell'involucro che raggruppa i vari componenti di cui è dotato (scheda madre, alimentatore, dischi ecc.). ▶

A seconda della distanza o della tipologia di sistema, i meccanismi di comunicazione possono essere completamente diversi, ma la loro struttura deve essere composta da tre componenti come vediamo nel seguente schema:

emittente => mezzo trasmissivo >= ricevente

In questo Modulo didattico ci occuperemo del mezzo trasmissivo, cioè di quali sono gli strumenti, le tecniche e le modalità con le quali due (o più) dispositivi vengono connessi in modo tale da poter comunicare scambiandosi informazioni.

Il mezzo trasmissivo è dipendente dalla distanza alla quale sono posti i dispositivi da connettere e proprio in base a essa vengono usate differenti tecnologie per rappresentare il messaggio, che può essere:

- di natura elettrica per piccole distanze, utilizzando un conduttore come mezzo trasmissivo;
- di natura ottica per distanze medio-lunghe, utilizzando una fibra particolare (fibra ottica);
- di tipo elettromagnetico per ogni tipo di distanza senza bisogno di connessioni fisiche ma utilizzando l'etere come mezzo trasmissivo (connessioni wireless).

Prima di affrontare lo studio dei mezzi trasmissivi è utile richiamare alcuni elementi di fisica necessari per comprendere come avviene la trasmissione del segnale e quali sono le problematiche connesse alle diverse tecnologie per poter garantire il corretto funzionamento dei sistemi di comunicazione.

SISTEMA

Un sistema è un insieme di componenti che interagiscono fra loro per realizzare uno scopo comune.

Trasmissione di segnali elettrici via cavo

La trasmissione di segnali elettrici via cavo sfrutta una connessione fisica tra due dispositivi effettuata mediante un cavo (filo) conduttore che garantisce al segnale di natura elettrica, generato dal dispositivo trasmettitore, di giungere pressoché inalterato al dispositivo ricevitore.

Sul conduttore avviene quindi il trasferimento di un messaggio, per esempio composto da una sequenza di uno e di zero, trasformato in segnali elettrici da un dispositivo presente nel trasmettitore. Gli stessi segnali elettrici "viaggiano" sul cavo e vengono trasferiti fino al ricevitore che li ritrasforma in segnali interpretabili dal destinatario.

Affinché il cavo possa trasmettere un segnale elettrico deve avere particolari caratteristiche tra le quali la più importante è una buona conducibilità, cioè l'attitudine a essere attraversato da corrente elettrica.

La corrente elettrica: richiami di elettrologia

Per spiegare la corrente elettrica è necessario ricorrere alla definizione di carica elettrica partendo dalla legge di Coulomb.

Carica elettrica

CARICA ELETTRICA

L'unità di misura di carica elettrica è chiamata coulomb ed è la quantità di carica Q che in un secondo attraversa una sezione qualunque di un conduttore percorso dalla corrente di 1 ampere.

Nel Sistema Internazionale (SI), dato che la corrente I è la grandezza fondamentale utilizzata come riferimento per le altre grandezze, si ricavano le dimensioni di Q dalla relazione:

$$Q = IT \quad (1)$$

dall'analisi dimensionale ne segue che:

$$1C = 1A \times 1s \quad (2)$$

cioè

$$[1 \text{ coulomb}] = [1 \text{ ampere} \times 1 \text{ secondo}] \quad (3)$$

Per avere un'idea delle dimensioni in gioco si ricorda che la carica dell'elettrone è:

$$e = 1{,}60 * 10^{-19} \text{ C} \quad (4)$$

quindi per avere la carica di 1C sono necessari circa $6{,}25 * 10^{18}$ elettroni.

Campo elettrico e potenziale

Il campo elettrico **E** presente in una regione dello spazio è definito operativamente come *il rapporto tra la forza **F** che agisce su una carica elettrica Q e la carica stessa*:

$$E = F/Q \quad (5)$$

da cui si ricava l'unità di misura del campo elettrico:

$$[\text{newton/coulomb}] \quad (6)$$

Come per il campo gravitazionale terrestre il campo elettrico in ogni suo punto può essere descritto da una funzione detta potenziale elettrico: a un punto del campo elettrico associamo un valore V_P che chiamiamo potenziale elettrico del punto p. Possiamo ora definire l'energia potenziale.

ENERGIA POTENZIALE DI UNA CARICA

Si definisce energia potenziale U di una carica Q posta in un punto P del campo il prodotto della carica Q per il potenziale V_P in quel punto:

$$U = QV_P \quad (7)$$

Dalla definizione si ottiene la formula inversa che esprime il potenziale elettrico V_P:

$$V_P = U/Q \quad (8)$$

Un concetto fondamentale è quello di differenza di potenziale elettrico tra due punti, A e B, del campo elettrico (indicato con *d.d.p.*$_{AB}$ oppure con ΔV_{AB}), che viene calcolato semplicemente con:

$$\Delta V = V_A - V_B = U_A/Q - U_B/Q = \Delta U/Q \quad (9)$$

dove ΔU è la differenza di energia tra i due punti del campo elettrico e ha il significato fisico del lavoro che è necessario per spostare la carica Q da un punto all'altro

$$L = \Delta U \quad (10)$$

La formula (9) può essere riscritta in questi termini:

$$\Delta V = V_A - V_B = L/Q \quad (11)$$

Alla sua unità di misura è stato dato il nome di *volt* che viene rappresentato con il simbolo **V** e risulta essere

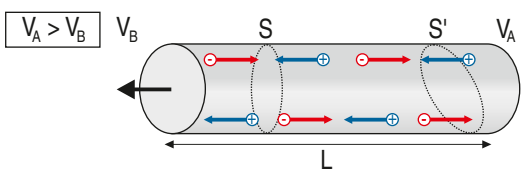
$$1V = 1J \times 1C^{-1} \quad (12)$$

cioè

$$[1 \text{ volt}] = [1 \text{ joule}/1 \text{ coulomb}] \quad (13)$$

Corrente elettrica

Si supponga di avere due punti A e B con diverso potenziale elettrico (rispettivamente V_A e V_B): quando i due punti vengono connessi tra loro con un filo metallico conduttore avviene uno spostamento delle cariche elettriche da A verso B (o viceversa a seconda dei valori dei potenziali). I potenziali elettrici diversi V_A e V_B tendono a eguagliarsi: pertanto si dice che la loro *differenza di potenziale* (d.d.p.) tende a zero.
Un moto di cariche elettriche fra A e B, cioè lo spostamento delle cariche elettriche che equilibrano la differenza di potenziale, costituisce una *corrente elettrica*. ▶



Se poniamo attenzione a una sezione S' del filo conduttore, essa sarà attraversata da una certa quantità di carica q in un intervallo di tempo di osservazione t.

CORRENTE ELETTRICA

Si definisce intensità della corrente elettrica, indicata con I, la quantità di carica che attraversa una data sezione S di un conduttore nell'unità di tempo:

$$I = \Delta Q/\Delta t \quad (14)$$

All'unità di misura dell'intensità di corrente, nel Sistema Internazionale (SI) in cui essa è una grandezza fondamentale, è stato dato il nome di *ampere* (*A*) e si rappresenta con il simbolo I.

Dalla definizione di intensità di corrente si deduce che:

$$1A = 1C \times 1s^{-1} \quad (15)$$

cioè

$$[1 \text{ ampere}] = [1 \text{ coulomb} / 1 \text{ secondo}] \quad (16)$$

Nella pratica vengono usati i sottomultipli dell'ampere:

$$1 \text{ mA (milliampere)} = 10^{-3} \text{ A} \quad (14)$$
$$1 \text{ µA (microampere)} = 10^{-6} \text{ A} \quad (15)$$

Diamo un'idea del numero di cariche elementari (elettroni) trasportate nell'unità di tempo da una corrente d'intensità pari a

$$1 \text{ µA} = 1 \text{ µC/1s} \quad (16)$$

Sappiamo quanti elettroni sono necessari per ottenere una carica pari a 1 C e lo dividiamo per 10^6 per ottenere

$$1 \text{ µC} = 6{,}25 * 10^{18}/10^6 = 6{,}25 * 10^{12} \quad (17)$$

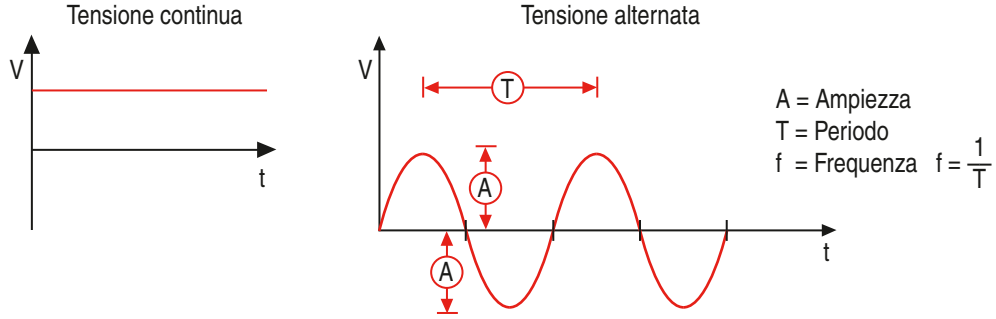
risulta quindi che la corrente di

$$1 \text{ µA} = 6{,}25 * 10^{12} \quad (18)$$

cariche elementari/secondo.

La corrente elettrica può essere di due tipi:

- continua: se la differenza di potenziale si mantiene sempre costante (DC, *Direct Current*);
- alternata: se la d.d.p. viene fatta variare nel tempo in modo che passi da una polarità all'altra secondo una legge tipicamente sinusoidale (AC, *Alternating Current*).



Tutte le apparecchiature e gli strumenti elettrici funzionano proprio perché esiste una differenza di potenziale elettrico tra due punti e si genera un flusso di corrente: nelle nostre case le "prese di corrente" collocate nei muri non sono altro che due punti del campo elettrico che il fornitore dell'energia (per esempio Enel) provvede a mantenere a una d.d.p. pari a 220 V alternato (fra i poli delle prese esiste una *tensione di 220 V* che garantisce un flusso di corrente per un tempo indefinito con la quale è possibile alimentare un utilizzatore come il frigorifero, il ferro da stiro, la TV, il videoregistratore, l'HiFi ecc.).

Conduttore elettrico

Non tutti gli elementi presenti in natura sono adatti a essere utilizzati come conduttori di elettricità: possiamo raggruppare i materiali in tre categorie proprio in base al loro comportamento:

- conduttori;
- semiconduttori;
- isolanti.

Il comportamento dei materiali è legato alla loro struttura fisica, cioè alla particolare costituzione degli atomi che la compongono.
Ogni atomo è costituito da:

- un nucleo, cioè la parte centrale dell'atomo, formato da:
  – protoni: cariche positive che stanno nel nucleo;
  – neutroni: non hanno carica e stanno nel nucleo;
- elettroni: hanno carica negativa e ruotano attorno al nucleo disposti su diverse orbite. Gli elettroni presenti sull'ultima orbita vengono chiamati elettroni liberi (e sono gli elettroni di valenza).

È proprio in base al numero di elettroni liberi che un materiale favorisce o meno il passaggio della corrente:

- senza elettroni liberi: materiali isolanti, come la plastica, la carta, il legno, l'aria, l'acqua;
- con quattro elettroni liberi: materiali semiconduttori, come il silicio, il germanio;
- con almeno un elettrone libero: materiale conduttore, come l'argento, l'oro, il rame, lo stagno, il corpo umano ecc.

A seconda del materiale anche i conduttori offrono un modesto ostacolo al passaggio della corrente che si esprime in termini di impedenza e viene indicato con la lettera Z: essa indica la tipologia di ostacolo che un conduttore offre, e può essere di tre tipi:

- resistivo: viene indicato con R (resistenza) e non è influenzato dal valore di frequenza della corrente;
- induttivo: viene indicato con X_L (impedenza reattiva) e aumenta proporzionalmente con la frequenza;
- capacitivo: viene indicato con X_C (impedenza capacitiva) e aumenta in modo inversamente proporzionale con la frequenza.

In generale l'impedenza di un conduttore è responsabile del degradamento del segnale che avviene quando questo attraversa un conduttore ed è sempre legato alla sua lunghezza: quindi in un conduttore è importante conoscere quanto attenua un segnale per dimensionare correttamente il mezzo trasmissivo.

Tipologie di cavi

Per ogni tipologia di applicazione sono previsti cavi specifici che permettono di ottenere le prestazioni desiderate: innanzitutto la prima differenza fondamentale, necessaria per effettuare la scelta di un cavo, è quella legata alla natura del segnale. Per i segnali digitali vengono utilizzati

cavi differenti da quelli dei segnali analogici, in quanto nella trasmissione digitale (◀ baseband ▶) sono necessarie caratteristiche diverse dei conduttori rispetto alla trasmissione analogica (broadband).

◀ Baseband È una tecnica che consente di trasmettere in modo diretto segnali digitali senza l'utilizzo della modulazione. ▶

L'attenuazione che un segnale subisce è in funzione della velocità di trasmissione (e quindi della sua frequenza), della lunghezza del conduttore e della tipologia di cavo utilizzato.

Abbiamo due tipologie di cavi:
- coassiale
  - thinnet (o *Thin Ethernet*): cavo sottile;
  - thicknet (o *Thick Ethernet*): cavo spesso;
- doppino
  - STP (*Shielded Twisted Pair*): a coppie di fili doppiamente schermati;
  - FTP (*Foiled Twisted Pair*): a coppie di fili con un'unica schermatura;
  - UTP (*Unshielded Twisted Pair*): a coppie di fili non schermati.

I più utilizzati per le trasmissioni digitali sulle reti Ethernet con velocità di 10 Mb sono tre:
- 10 BASE 2, dove il 2 indica che il segnale può viaggiare per circa 200 m, chiamato Thinnet, ed è un cavo coassiale sottile;
- 10 BASE 5, dove il 5 indica che il segnale può viaggiare per circa 500 m, chiamato Thicknet dato che è un cavo coassiale spesso;
- 10 BASE T, dove T indica twisted pairs, cioè cavi con coppie di fili attorcigliati.

Cavo coassiale

Il cavo coassiale è costituito da un filo conduttore centrale di rame ricoperto da un cilindro di plastica isolante sul quale viene avvolta una maglia costituita da filamenti di rame, e il tutto è esternamente ricoperto da una guaina.

Il foglio (o la calza metallica) che avvolge il cavo coassiale oltre a essere il conduttore per il ritorno della corrente del segnale realizza una gabbia di Faraday per il conduttore interno rendendolo praticamente immune ai disturbi elettromagnetici. ▶

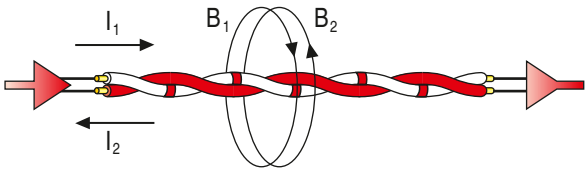
Agli estremi (o capocorda) vengono crimpati (oppure avvitati o saldati) dei connettori particolari, i connettori BNC, e il circuito tra i due connettori viene chiuso mediante un terminatore a 50 ohm che permettere inoltre di connettere a massa la maglia esterna (anche chiamata calza). ▶

Essi vengono realizzati con due diversi diametri del conduttore interno:
- il più spesso, o thicknet, è oggi utilizzato nei backbone Ethernet, ed è costituito da:
  - un conduttore centrale in rame;
  - un cilindro isolante (generalmente realizzato in teflon);
  - due schermi in foglio di alluminio;
  - due schermi in calza;
- il più sottile, o thinnet, è stato largamente utilizzato per le "vecchie" reti Ethernet degli anni '90 (tipo RG58): ha un'attenuazione circa tre volte superiore al cavo Thick ed è costituito da:
  - un conduttore centrale in rame;
  - un cilindro isolante (generalmente realizzato in teflon);
  - uno schermo in foglio di alluminio;
  - uno schermo in calza.

Può arrivare a distanze maggiori delle connessioni STP e UTP ed è meno costoso delle fibre ottiche.

Doppino

Il doppino è costituito da una o più coppie (pair) di conduttori di rame "attorcigliati tra loro" (twisted) per ridurre gli effetti di disturbi originati dai campi magnetici e dalla corrente che circola nei conduttori stessi. ▶



Le correnti I_1 e I_2 che attraversano i due conduttori sono di uguale intensità e verso opposto, e quindi generano campi magnetici B_1 e B_2 opposti che tendono ad annullarsi.

Per migliorare la tolleranza ai disturbi elettromagnetici (EMI) e quindi migliorare le caratteristiche di un cavo si introduce la schermatura.
Questa si realizza avvolgendo il doppino o l'intero cavo:

- con un foglio di alluminio molto sottile (da 0,05 mm a 0,2 mm);
- con una calza costituita da un intreccio "a reticolo" di due trecce di fili di rame che avvolgono il filo in due direzioni opposte.

I cavi composti da doppini hanno costi inferiori rispetto ai cavi coassiali e per questo motivo sono utilizzati normalmente sia in telefonia sia per la trasmissione dati nelle reti locali e nei cablaggi strutturati. A seconda del livello di schermatura abbiamo diverse tipologie di cavi.

- Cavo UTP

  Il cavo UTP (*Unshielded Twisted Pair*) è costituito da un insieme di quattro coppie di fili attorcigliati e avvolti da una guaina di isolante. A differenza del STP ogni coppia di cavi non è singolarmente schermata. ▶

  L'impedenza in questo tipo di conduttori è generalmente di 100 Ω: esso viene utilizzato per distanze fino a 100 metri ed è sensibile ai disturbi elettromagnetici, ma rispetto al cavo STP è più economico e semplice da installare. Come connettori vengono utilizzati gli RJ45. ▶

Il connettore RJ45 può essere utilizzato collegando una coppia o due coppie per fonia mentre si collegano tutte e quattro le coppie nel cablaggio strutturato.

- Cavo FTP

  Il cavo FTP (*Foiled Twisted Pair*) è costituito da un insieme di quattro coppie di fili attorcigliati con un'unica schermatura globale costituita da un foglio di alluminio. ▶

  L'impedenza in questo tipo di conduttori è generalmente di 100 Ω.

- Cavo STP

  Il cavo STP (*Shielded Twisted Pair*) è costituito da un insieme di quattro coppie di fili attorcigliati e schermati sia per ogni singola coppia che tutti insieme, e avvolti da una guaina di isolante. ▶

  L'impedenza in questo tipo di conduttori è generalmente di 150 Ω: esso viene utilizzato per distanze fino a 100 metri ed è in grado di ridurre maggiormente il rumore rispetto al cavo UTP ma è più costoso. La schermatura metallica viene connessa a massa a entrambe le estremità.

Zoom su...

CAVO IBM

Il cavo STP a 150 W è anche conosciuto come cavo di Tipo 1 IBM: esistono anche altri tipi di cavi con il nome IBM, per esempio il Tipo 2 IBM, costituito da un cavo di Tipo 1 IBM con l'aggiunta di quattro coppie non schermate da 100 Ω, nato come soluzione integrata per il Cabling System IBM dati (due coppie a 150 W) e fonia (quattro coppie a 100 W), e il cavo di Tipo 6 IBM a 150 W con due coppie singolarmente schermate in foglio e schermo globale in calza di rame.

Classificazione dei doppini

Tutti i tipi di cavi elettrici sono sensibili alla frequenza del segnale che devono trasmettere e quindi è necessario scegliere opportunamente il cavo in funzione della specifica applicazione (in particolare l'attenuazione e la diafonia sono fortemente dipendenti dalla frequenza, come vedremo in seguito).

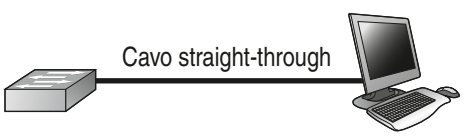
Per semplificare la scelta è stata fatta una classificazione che cataloga i cavi in sette categorie in base alle applicazioni per le quali essi sono idonei: al crescere del numero di categoria aumentano le prestazioni del cavo e quindi le possibilità di impiego del cavo stesso.

- Categoria 1: solo per telefonia analogica (*Telecommunication*).
- Categoria 2: per telefonia digitale ISDN e trasmissione di dati a bassa velocità (linee seriali, *Low Speed Data*).
- Categoria 3: per reti locali che non producono frequenze fondamentali superiori a 12.5 MHz, adatti a realizzare reti locali fino a 10 Mb/s (Ethernet 10BaseT e 10BaseT4, Token Ring 4 Mb/s).
- Categoria 4: per reti locali che non producono frequenze fondamentali superiori a 20 MHz (Token Ring 16 Mb/s).
- Categoria 5: per reti locali che non producono frequenze fondamentali superiori a 32 MHz; comprende i cavi disponibili per applicazioni fino a 100 Mb/s, su distanze di 100 metri (FDDI MLT-3, Ethernet 100BaseTX, ATM. Sono i cavi normalmente utilizzati per reti locali).
- Categoria 6: per reti locali con una banda passante fino a 200 MHz; comprende i cavi disponibili per applicazioni fino a 1 Gb/s su distanze di 100 metri.
- Categoria 7: per reti locali con una banda passante fino a 600 MHz; comprende i cavi disponibili per applicazioni fino a 10 Gb/s su distanze di 100 metri.

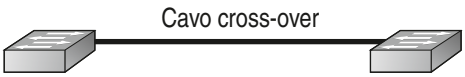
Cavi: collegamento dei pin

In generale possiamo affermare che esistono tre tipologie di cavi, in relazione a come essi vengono connessi ai due capi:

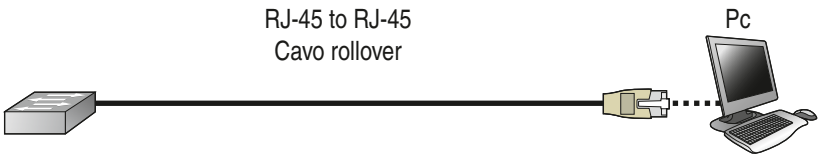
- Straight-through o dritto: permette di effettuare il collegamento tra dispositivi di tipo diverso, quali:
  - Switch e router
  - Switch e PC o server
  - Hub e PC o server



- Cross-over o incrociato: permette di effettuare il collegamento tra dispositivi dello stesso tipo, quali:
  - Switch e switch
  - Switch e hub
  - Hub e hub
  - Router e router
  - PC e PC
  - Router e PC



- Rollover: è il cavo che collega l'adattatore RJ45 posto sulla porta seriale del computer alla porta console di un router o uno switch.



Nella tabella seguente sono messi a confronto i colori delle connessioni tra il cavo dritto e quello incrociato nello standard ◀ EIA/TIA-568B ▶ dove è possibile vedere che solo metà dei pin vengono "crossati" mentre gli altri quattro sono identici nelle due diverse tipologie.

◀ EIA/TIA TIA (*Telecommunications Industry Association*) ed EIA (*Electronic Industries Alliance*) sono enti leader nello sviluppo di standard di ingegneria. Le specifiche EIA/TIA-568B riguardano gli standard per il cablaggio dei sistemi di comunicazione per gli edifici commerciali e prevedono due insiemi di cavi, uno per la voce e uno per i dati: quello più frequentemente raccomandato per entrambi è il CAT 5 UTP (categoria 5 UTP). ▶

Straight-through			Cross-over		
Pin 1	_____	Pin 1 Bianco Arancio	Bianco Arancio	Pin 1	_____ Pin 3 Bianco Verde
Pin 2	_____	Pin 2 Arancio	Arancio	Pin 2	_____ Pin 6 Verde
Pin 3	_____	Pin 3 Bianco Verde	Bianco Verde	Pin 3	_____ Pin 1 Bianco Arancio
Pin 4	_____	Pin 4 Blu	Blu	Pin 4	_____ Pin 4 Blu
Pin 5	_____	Pin 5 Bianco Blu	Bianco Blu	Pin 5	_____ Pin 5 Bianco Blu
Pin 6	_____	Pin 6 Verde	Verde	Pin 6	_____ Pin 2 Arancio
Pin 7	_____	Pin 7 Bianco Marrone	Bianco Marrone	Pin 7	_____ Pin 7 Bianco Marrone
Pin 8	_____	Pin 8 Marrone	Marrone	Pin 8	_____ Pin 8 Marrone

È possibile riconoscere i diversi cavi confrontando i due connettori RJ45 presenti alle due estremità:
- nel straight-through la sequenza dei fili è identica;
- nel cross-over vengono incrociati i pin 1 con 3 e 2 con 6 dato che i fili di ricezione e di trasmissione sono tra loro invertiti;
- nel rollover i colori sono esattamente in senso opposto.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Quali, tra le differenti tecnologie, sono usate per rappresentare il messaggio? *(indica quella errata)***

- O di tipo elettrico
- O di tipo ottico
- O di tipo magnetico
- O di tipo elettromagnetico

2 **I materiali possono essere di tre categorie: *(indica quella errata)***

- O conduttori
- O isolanti
- O semiconduttori
- O semi-isolanti

3 **STP è l'acronimo di:**

- O *Shielded Twisted Pair*
- O *Shared Twisted Pair*
- O *Shared Twiced Pair*
- O *Shielded Twiced Pair*
- O nessuna delle precedenti

4 **Il cavo thicknet è costituito da:**

- O un conduttore centrale in rame
- O due conduttori centrali in rame
- O un cilindro isolante
- O due cilindri isolanti;
- O uno schermo in foglio di alluminio
- O due schermi in foglio di alluminio
- O uno schermo in calza
- O due schermi in calza

5 **Indica a quale categoria appartengono i seguenti cavi:**

- O Categoria .....: 12,5 MHz
- O Categoria .....: solo per telefonia analogica
- O Categoria .....: 32 MHz
- O Categoria .....: 20 MHz
- O Categoria .....: per telefonia digitale ISDN

6 **Indica quale tipo di cavo RJ45 viene utilizzato per i seguenti collegamenti:**

- O Switch e router .....
- O Hub e hub .....
- O Switch e server .....
- O Hub e PC .....
- O Router e router .....
- O Hub e server .....
- O Switch e switch .....
- O Switch e PC .....
- O Switch e hub .....
- O PC e PC .....
- O Router e PC .....

Test vero/falso

1	Con il termine "elettroni liberi" in un materiale si intendono gli elettroni di valenza.	V F
2	Un materiale semiconduttore ha quattro elettroni liberi nell'ultima orbita.	V F
3	La trasmissione digitale è anche chiamata broadband.	V F
4	Il cavo thicknet (o Thick Ethernet) è un doppino di tipo spesso.	V F
5	I connettori RJ45 vengono utilizzati sui cavi UTP.	V F
6	I connettori RJ45 vengono utilizzati sui cavi FTP.	V F
7	Il cavo UTP è costituito da quattro coppie di fili attorcigliati e avvolti da una guaina di isolante.	V F
8	Il cavo FTP è costituito da quattro coppie di fili attorcigliati e avvolti da una guaina di isolante.	V F

Verifichiamo le competenze

Esprimi la tua creatività

Domande a risposta aperta

1 Vengono usate differenti tecnologie per rappresentare il messaggio che può essere di natura .....

.....

2 L'impedenza di un materiale è dovuta a .....

.....

3 Sono disponibili tre tipologie di cavi a coppie, ...

.....

4 Il cavo coassiale è costituito da un filo .....

.....

5 Il cavo UTP è costituito da un insieme .....

.....

6 Il cavo FTP è costituito da un insieme .....

.....

7 Il cavo STP a 150 W è anche conosciuto come cavo di .....

.....

8 Il cavo di Tipo 6 IBM a 150 W è costituito da .....

.....

9 La categoria 5 di cavi viene utilizzata per .....

.....

10 La categoria 6 viene utilizzata per .....

.....

Esercizi di completamento

isolanti • thicknet • emittente • mezzo trasmissivo • semiconduttori • isolanti • thinnet • thicknet • straight-through • cross-over • rollover • ricevente • conduttori

1 La struttura dei sistema di comunicazione è formata da tre componenti: .....

2 In base al loro comportamento elettrico possiamo raggruppare i materiali in tre categorie: .....

3 Abbiamo due tipologie di cavi coassiali:

o ..... cavo sottile

o ..... cavo spesso

4 Il cavo ..... è oggi utilizzato nei backbone Ethernet.

5 Esistono tre tipologie di cavi, in relazione a come essi vengono connessi ai due capi: .....

.....

UNITÀ DIDATTICA 2

LE MISURE SUI CAVI IN RAME

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere le caratteristiche elettriche dei cavi
- a effettuare i test sui cavi

Caratteristiche elettriche

È abbastanza complesso individuare una relazione che metta in collegamento le prestazioni elettriche con le caratteristiche fisiche dei cavi.
Bisognerebbe trovare una relazione tra la geometria dei materiali impiegati, il tipo e le modalità del loro raggruppamento, le tipologie e il numero di schermature utilizzate ecc. e legarle mediante una funzione con la frequenza del segnale, l'impedenza e l'attenuazione. Si preferisce studiare singolarmente come questi parametri elettrici sono influenzati dai campi elettromagnetici nelle diverse tipologie di cavi.

Impedenza

Abbiamo detto che l'impedenza è l'ostacolo che il conduttore oppone al passaggio della corrente: si misura in ohm (Ω) e si esprime con la relazione

$$(Z = R + jX)$$

dove il primo termine tiene conto degli effetti resistivi del circuito (R) e il secondo termine di quelli reattivi, come somma algebrica degli effetti induttivi (X_L) e capacitivi (X_C).

Al variare della frequenza il valore dell'impedenza varia in quanto le reattanze X_C e X_L sono legate al valore della pulsazione $\omega = 2\pi f$ dalle seguenti relazioni:

$$X_L = j\omega L \text{ e } X_C = 1/j\omega C$$

L'obiettivo è quello di ottenere cavi con valore di impedenza poco sensibile alla frequenza, in modo da avere prestazioni sempre costanti e valori di impedenza stabili per effettuare accoppiamenti tra emettitore e ricevitore che garantiscano il massimo trasferimento di segnale (adattamento di impedenza).

Oggi si realizzano cavi con impedenze comprese tra 50 e 150 ohm, per lavorare a un intervallo di frequenze tra 100 Hz e 350 MHz.

Zoom su...

AWG

Le dimensioni dei conduttori sono state tabulate secondo un'unità di misura apposita (AWG, *American Wire Gage*) che in funzione della resistività permette di calcolare l'impedenza del conduttore.

AWG	mm (∅)	mm²	kg/km	Ω/km
22	0,6438	0,3255	2,894	52,96
23	0,5733	0,2582	1,820	84,21
24	0,5106	0,2047	1,746	87,82
25	0,4547	0,1624	1,414	108,4
26	0,4049	0,1288	1,145	133,9
28	0,3638	0,0823	0,894	145,3

Per i cablaggi strutturati si utilizzano cavi 22 o 24 AWG mentre per i cavetti di permutazione si utilizzano i cavi 26 AWG.

Velocità di propagazione

I segnali si propagano nei conduttori con velocità ridotta rispetto a quella della luce nel vuoto (circa $3 \cdot 10^8$ m/s), soprattutto nel caso di frequenze elevate. Quando un trasmettitore ha finito di inviare il segnale, se la distanza con il ricevitore è notevole, questo potrebbe non avere ancora iniziato a riceverlo.

VELOCITÀ DI PROPAGAZIONE

Si definisce velocità di propagazione v_p la percentuale della velocità della luce nel vuoto dalla quale si propaga un segnale elettrico sul cavo.

ESEMPIO 1

Proviamo a calcolare "come si muovono i bit" nel caso di velocità di trasmissione di 10 Mb/s.
Nei cavi di rame la velocità di propagazione media è del 65% quindi si ottiene che la velocità del segnale è di:

$$V_s = 0{,}65 \times 3 \cdot 10^8 \text{ m/s} \approx 200.000 \text{ km/s}$$

Dalla velocità di trasmissione (10 Mb/s) ricaviamo per quanto tempo un bit viene "posto" sulla linea (bit time), cioè:

$$t_s = 1/10 * 10^6 = 10^{-7}$$

Otteniamo lo spazio percorso dalla relazione fisica che lega lo spazio con la velocità

$$s = V * t = 200 * 10^6 \times 10^{-7} = 20 \text{ m}$$

Al termine del tempo dedicato alla trasmissione di un singolo bit quest'ultimo ha percorso solo 20 m.

Viene definito propagation delay (ritardo di propagazione) il tempo che il segnale impiega ad attraversare un cavo; esso viene sfruttato dai tester che misurano la lunghezza dei cavi, il *Time Domain Reflectometer* (TDR) che, in base al tempo che un segnale impiega a percorrere un conduttore, ne ricava la sua misura.
Questo test viene anche effettuato per individuare a quale distanza è presente un guasto di un conduttore, cioè la presenza di un circuito aperto o di un cortocircuito. ▶

Zoom su...

VELOCITÀ E PARAMETRI DI FUNZIONAMENTO

I parametri di funzionamento di alcuni protocolli di livello MAC per le LAN (per esempio CSMA/CD, il protocollo della rete Ethernet 802.3, come vedremo in seguito) sono stati calcolati avendo come riferimento proprio la velocità di trasmissione.

Attenuazione

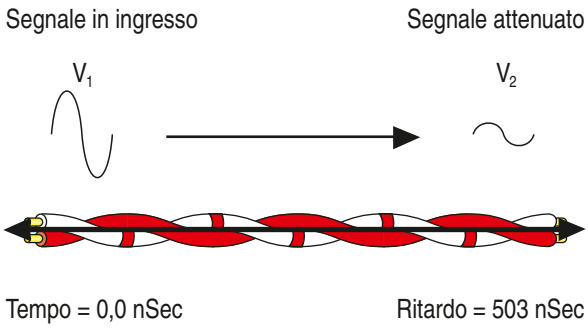
Per effetto dell'impedenza il segnale si degrada man mano che attraversa un conduttore.

ATTENUAZIONE

Con attenuazione si definisce il rapporto tra la tensione del segnale in ingresso al cavo e la tensione misurabile all'altra estremità.

Prende anche il nome di Insertion Loss (perdita di inserzione). Di solito è espressa in decibel (dB) e, poiché si tratta del rapporto tra un segnale minore e uno maggiore, avrà segno negativo.

Questa perdita è dovuta all'impedenza elettrica del cavo di rame, alla perdita di energia attraverso l'isolamento dei cavi e all'impedenza causata dai connettori; inoltre la perdita di inserzione aumenta con la distanza e con la radice quadrata della frequenza.



Per misurare l'attenuazione, il cavo viene testato con un cable tester che usa le frequenze più alte ammesse su quel cavo in modo da calcolare il massimo valore di attenuazione.

Il cable tester della figura a lato permette di effettuare parecchie misure oltre all'Insertion Loss: il Wire Map, Delay, Delay Skew, Length, DC Resistance, Return Loss, NEXT, Power Sum NEXT (PSNEXT), Equal Level Far-End Cross Talk (ELFEXT), Power Sum ELFEXT (PSELFEXT), Attenuation-to-Cross Talk Ratio (ACR), Power Sum ACR (PSACR). Ha un costo di circa 5000,00 euro).

Zoom su...

DECIBEL

Per confrontare potenze o ampiezze relative si utilizza una misura del loro rapporto in scala logaritmica, detta decibel:

- le onde di luce nelle fibre ottiche e le onde radio usano i dB in potenza

$$dB = 10 \cdot \log\left(\frac{P_2}{P_1}\right)$$

- le onde elettromagnetiche usano i dB in tensione moltiplicati per 20 invece che per 10

$$dB = 20 \cdot \log\left(\frac{V_2}{V_1}\right)$$

Se il risultato è negativo si parla di attenuazione, altrimenti di amplificazione.

Esprimiamo in decibel il rapporto in queste tre situazioni:

a) $\left|\frac{V_2}{V_1}\right| = 10 \Rightarrow |A_V| = 20\ dB$ è un'amplificazione;

b) $\left|\frac{V_2}{V_1}\right| = 0{,}1 \Rightarrow |A_V| = -20\ dB$ è un'attenuazione;

c) $\left|\frac{V_2}{V_1}\right| = 0{,}5 \Rightarrow |A_V| = -3\ dB$ è un'attenuazione.

Rumore

RUMORE

Si definisce rumore (noise) un segnale non desiderato che si sovrappone al segnale trasmesso.

Le principali cause di rumore sono le seguenti:

- crosstalk (diafonia): il segnale presente su un filo è disturbato dal campo elettromagnetico generato dal segnale dei fili posti accanto a esso;
- RFI: disturbi in radiofrequenza dovuti ad altri dispositivi che utilizzano la stessa tecnologia (come per esempio i forni a microonde, le trasmissioni radio/TV, i telefoni cellulari ecc.);
- EMI: disturbi di natura elettromagnetica, come quelli generati dai cavi elettrici, dalle condizioni meteorologiche ecc.).

A seconda che il rumore agisca su tutte le frequenze del segnale oppure solo su una gamma delle stesse prende il nome di rumore bianco oppure narrow band.

La protezione dal rumore nei cavi viene fatta con la schermatura. Abbiamo visto tre tipi di schermatura:

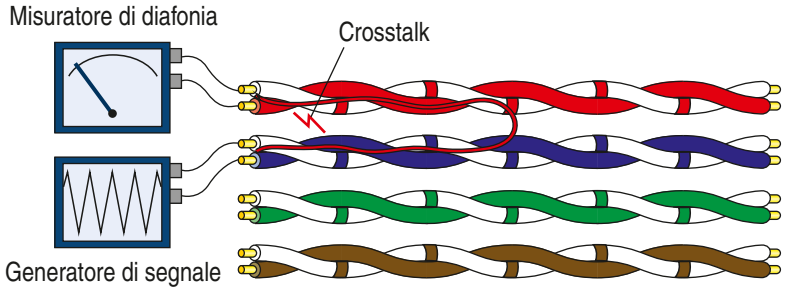
- foglio (foil);
- calza (braid);
- foglio più calza: uso combinato del foglio e della calza.

Diafonia

Se due cavi sono disposti uno vicino all'altro, il passaggio della corrente in essi genera dei campi elettromagnetici e ogni cavo produce un disturbo per l'altro. Questo fenomeno prende il nome di diafonia (in inglese crosstalk).
Come parametro di qualità di un cavo viene misurata "l'insensibilità" ai campi elettromagnetici indotti del cavo stesso e viene espresso in decibel il valore di attenuazione del segnale indotto: più un segnale indotto viene attenuato e più il cavo risulta essere insensibile ai disturbi generati dalla diafonia.

Esistono due tipi di diafonia: la paradiafonia e la telediafonia.
La paradiafonia indica il valore del segnale indotto nel cavo vicino quando questo viene misurato dalla stessa parte del trasmettitore (viene anche indicato con NEXT, *Near End Crosstalk*).

$$NEXT = V_{\text{segnale di test}} - V_{\text{segnale crosstalk}}$$



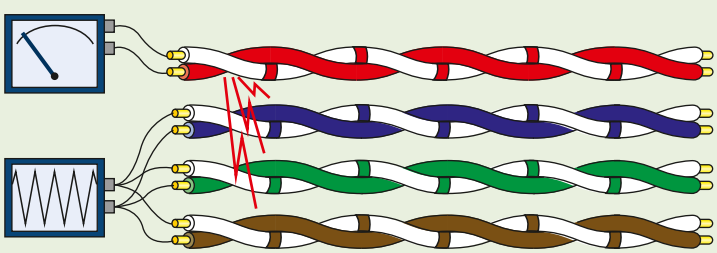
Il risultato di questa misurazione è sempre un valore negativo: poiché è espresso in decibel, quando il risultato viene letto con uno strumento la lettura sarà positiva e quindi bisogna aggiungere il segno meno.

Ogni coppia di fili deve essere controllata e la misura deve essere effettuata a ogni estremità del cavo: più il valore è alto, migliore sarà l'insensibilità del cavo (un cavo con NEXT = –20 dB è migliore di un cavo con NEXT = –10 dB).

Zoom su...

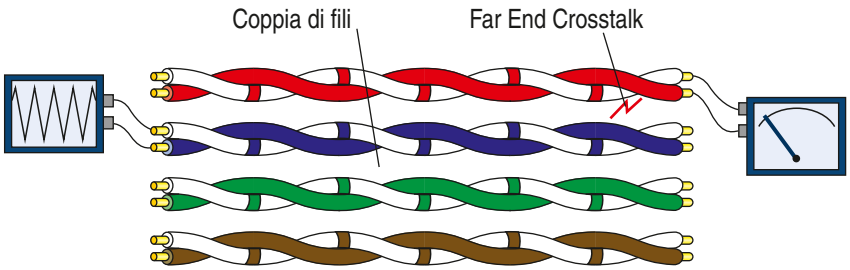
PSNEXT

A volte si misura contemporaneamente l'effetto cumulativo della paradiafonia su tutte le coppie di fili: nell'esempio rappresentato nella figura si trasmette un segnale su tre coppie di fili e si effettua la lettura del NEXT sulla quarta coppia. ▶



Le specifiche TIA/EIA-568B richiedono anche questo test.

La telediafonia indica il valore del segnale indotto nel cavo vicino quando questo viene misurato dall'estremo opposto del trasmettitore (viene anche indicato con FEXT, *Far End Crosstalk*).



Dato che sui fili il segnale viene attenuato, anche il segnale indotto dovuto al crosstalk subisce una attenuazione lontano dal trasmettitore; quindi il FEXT è un problema meno grave del NEXT.

Zoom su...

MISURA DELLA DIAFONIA

È abbastanza semplice effettuare la misura della diafonia in quanto è sufficiente trasmettere un segnale in un cavo e misurare la tensione indotta nell'altro cavo collegando lo strumento di misura a una sola estremità: vista l'importanza di questo parametro, soprattutto quando molte coppie di cavi "scorrono affiancati" nella stessa guaina, deve sempre essere indicato nella certificazione dei sistemi di cablaggio.
Nella pratica viene misurato solo il valore di paradiafonia e viene utilizzato genericamente come diafonia utilizzando un cable tester.
Nella figura a lato sono riportati due modelli di cable tester. ▶

ACR

L'attenuazione che abbiamo appena descritto deve però sottostare a due esigenze opposte: se da una parte si vuole un'alta attenuazione del segnale indotto, dall'altra si richiede una minima attenuazione per il segnale trasmesso.

È quindi necessario avere un parametro che esprima contemporaneamente questa caratteristica del cavo: è stata introdotta l'ACR, *Attenuation to Crosstalk Ratio*, che esprime il rapporto tra il segnale attenuato presente su una coppia e il segnale indotto dalla coppia vicina.
Per avere una buona trasmissione tale rapporto deve essere elevato in modo che il rumore risulti trascurabile rispetto al segnale e quindi il cavo trasmetta i dati in modo affidabile riducendo al minino la possibilità che vi siano errori di trasmissione.
Anch'esso viene misurato utilizzando un cable tester.

ACR è il risultato più importante per i test su un collegamento perché rappresenta la performance complessiva del cavo utilizzato.

ELFEXT (Equal Level Far End Crosstalk)

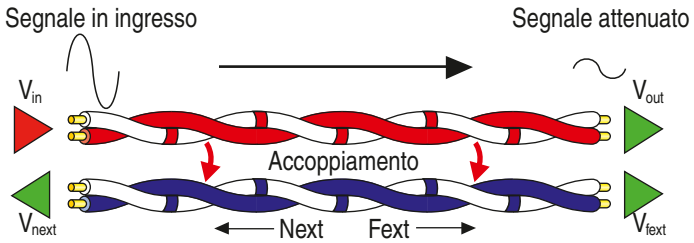
Un parametro significativo che permette di normalizzare i risultati delle prove rispetto alla lunghezza dei cavi è quello che prende il nome di ELFEXT, ed è ottenuto non da una misura ma da un'operazione algebrica:

- si calcola l'attenuazione in una coppia di cavi;
- si calcola il FEXT in una coppia di cavi adiacente, accoppiati con il segnale indotto dalla prima coppia;
- si ottiene il valore di ELFEXT come differenza delle due misurazioni.

Naturalmente le due coppie di cavi devono avere la stessa lunghezza e devono essere della stessa tipologia.

Vediamo un esempio con misurazioni fatte su un cavo di 50 metri: ▶

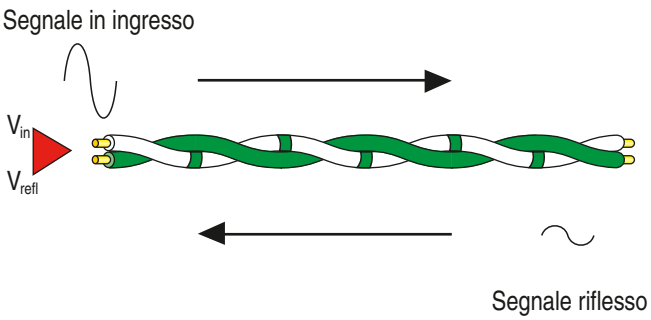
- Attenuazione = 11 dB
- FEXT = 45 dB
- ELFEXT = 45 – 11 = 34 dB



Per cercare di capire il significato del ELFEXT è utile pensarlo come il valore di ACR del limite estremo del cavo (*far end*).

Riflessione

Il segnale propagandosi lungo il cavo può incontrare delle discontinuità dovute per esempio a connettori, deformazioni, giunture che provocano variazioni di impedenza, facendo sì che una parte di esso venga riflesso e ritorni alla sorgente.
La misura accurata di questo valore prende il nome di perdita di ritorno o *Return Loss* (RL) e viene misurato in dB. ▶



La componente di RL dovuta al piccolo valore dell'impedenza caratteristica sul cavo prende il nome di *Structural Return Loss* (SRL) ed è un parametro che riassume l'omogeneità della costruzione del cavo e deve essere monitorato durante il processo di produzione dei cavi.

Questa misurazione è di particolare importanza nella realizzazione di Gigabit Ethernet dato che varia sensibilmente con la frequenza: ha valori superiori a basse frequenze e valori più bassi ad alte frequenze.

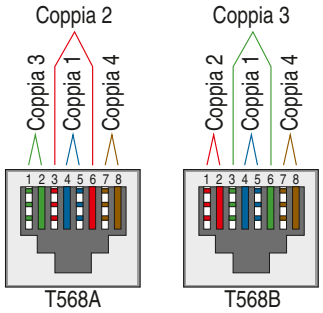
■ Test da effettuare sullo standard TIA/EIA-568B

Lo standard TIA/EIA-568B del 2001 considera un elenco di dieci test ai quali deve essere sottoposto un cavo di rame al fine di:

- determinare la sua lunghezza;
- individuare i collegamenti non effettuati correttamente;
- fornire la mappa dei cavi e di come sono "crossati" nel connettore;
- misurare l'attenuazione del segnale;
- misurare l'eventuale interferenza;
- misurare il livello di rumore.

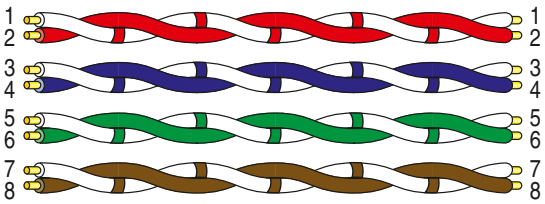
Vediamoli nel dettaglio.

1 Mappatura dei fili (*wire map*): i fili devono seguire una determinata sequenza di colori. ▶
Nel connettore Ethernet RJ45 ogni coppia di fili ha una funzione, e va ricordato come le coppie 1-2 effettuano la trasmissione mentre le coppie 3-6 sono quelle utilizzate per la ricezione.

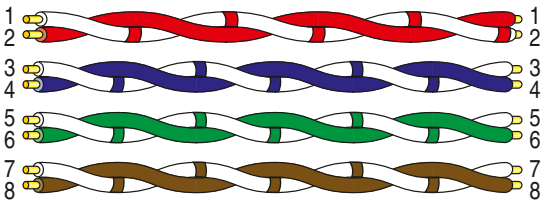


Ci sono vari tipi di errori nella disposizione dei fili (*wire fault*) e vanno confrontati con la disposizione corretta:

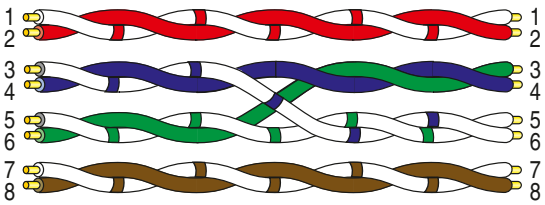
connessione corretta dei fili



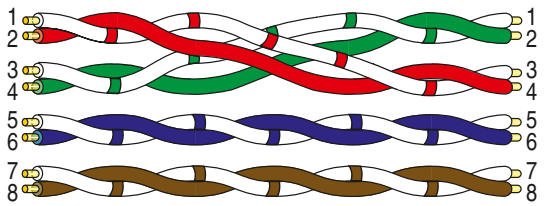
connessione errata
una coppia invertita



connessione errata
una coppia mischiata



connessione errata
trasposizione di due coppie



2 *Cable length*: è la misura della lunghezza dei cavi per cui bisogna assicurarsi che sia inferiore al valore massimo raccomandato di 100 metri in una rete 10BASE-T/100BASE-TX/1000BASE-T.

Due test sono legati alla discontinuità:

3 *Return Loss* (RL): è la misura della perdita di segnale di ritorno, riflesso verso il trasmettitore;

4 *Insertion Loss* (IL): è una misura in decibel che combina l'attenuazione alla discontinuità.

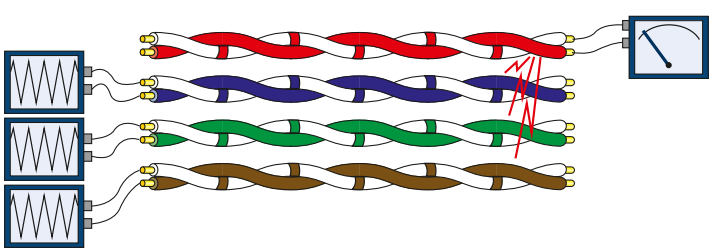
Quattro test per la valutazione della misura del crosstalk:

5 NEXT = $V_{\text{segnale di test}} - V_{\text{segnale crosstalk}}$ [in DB, negativo] (misurato alla stessa estremità);

6 PSNEXT: misura basata sugli effetti combinati di NEXT sulle quattro coppie di fili;

7 ELFEXT: è un calcolo ottenuto da (FEXT – Insertion Loss); è importante per reti 1000 baseT;

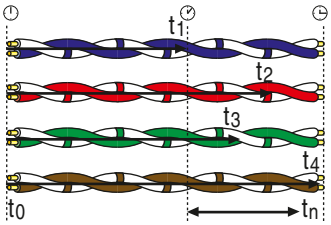
8 PSELFEXT (*Power Sum Equal Level Far End Crosstalk*): anch'esso è un calcolo, ed è l'effetto combinato di ELFEXT da tutte le coppie di fili. Si ottiene come somma algebrica degli effetti ELFEXT individuale su ogni coppia dalle altre tre coppie, quindi ci sono quattro risultati PSELFEXT per ogni estremità. Il valore tipico del PSELFEXT è inferiore di circa 3 dB al caso peggiore del risultato di ELFEXT a ogni estremità del collegamento. ▶



Gli ultimi due test sono riferiti al ritardo del segnale:

9 Propagation Delay: viene misurato il ritardo di propagazione.

10 Delay Skew: verifica la differenza di ritardo di propagazione tra la serie più veloce e quella più lenta delle coppie di fili. ▶
Il valore ideale è compreso tra 25 e 50 nanosecondi in un cavo lungo 100 metri: più il valore è basso migliore è il cavo.



La normativa ISO 11801/2002 aggiunge altri tre test a quelli della TIA-568B:

1 *Attenuation to Crosstalk Ratio* (ACR)
Viene richiesto un valore abbastanza grande affinché si possa avere un basso numero di errori. Se l'ACR non è sufficientemente grande è necessario passare da un tipo di cavo a un altro: per esempio può essere necessario per passare da un doppino schermato (UTP) a doppino ritorto schermato (STP) al fine di aumentare l'ACR.

2 *Power Sum ACR* (PSACR)
Il *Power Sum ACR* (PSACR) è simile all'ACR ma nel calcolo viene utilizzato il valore PSNEXT piuttosto che il NEXT.

3 Resistenza DC
È la misura dell'impedenza di cortocircuito del cavo, determinata mettendo una coppia di cavi per chiudere l'anello a un'estremità della connessione.
Il valore calcolato si esprime in ohm ed è l'impedenza caratteristica del cavo.

Ricapitoliamo nella tabella che segue l'insieme dei test confrontando quelli che devono essere superati (Pass/Fail) rispetto a quelli per i quali viene solo indicato il valore misurato a titolo informativo.

Tipo di prova	TIA-568-B	ISO/IEC 11801 (2002)
Mappatura	Pass / Fail	Pass / Fail
Lunghezza del cavo	Pass / Fail	Solo informativo
Attenuazione di inserzione (IL)	Pass / Fail	Pass / Fail
Perdita di ritorno (RL)	Pass / Fail	Pass / Fail
Near End Crosstalk (NEXT)	Pass / Fail	Pass / Fail
Power Sum NEXT (PSNEXT)	Pass / Fail	Pass / Fail
Equal Level Far End Crosstalk (ELFEXT)	Pass / Fail	Pass / Fail
Power Sum ELFEXT (PSELFEXT)	Pass / Fail	Pass / Fail
Ritardo di propagazione	Pass / Fail	Pass / Fail
Delay Skew	Pass / Fail	Pass / Fail
Attenuazione to Crosstalk Ratio (ACR)	Solo informativo	Pass / Fail
Power Sum ACR (PSACR)	Solo informativo	Pass / Fail
Resistenza DC	Solo informativo	Pass / Fail

Categorie e classi ISO

I cavi vengono classificati in categorie e vengono "marchiati" con sigle tipo CAT 5, CAT 5e, CAT 6 ecc.

Il primo parametro utilizzato per classificare cavi di rete riguarda la frequenza dei segnali che si possono propagare senza che le interferenze reciproche dei conduttori al loro interno producano alterazioni del segnale significative.

Parlare di frequenza significa anche parlare di capacità di trasporto dei dati (banda) dato che questi due fattori sono strettamente connessi tra loro.

TIA (*Telecommunications Industry Association*) e ISO (*International Standard for Organization*) sono enti leader nello sviluppo di standard di cablaggio strutturato. I componenti di questi comitati lavorano in collaborazione con gli enti di sviluppo delle applicazioni per garantire che le nuove categorie di cablaggio siano in grado di supportare le più recenti innovazioni della tecnologia di trasmissione del segnale. Gli standard TIA sono specificati per il mercato nordamericano, mentre gli standard ISO si riferiscono più comunemente al mercato globale.

CABLAGGIO STRUTTURATO

Il cablaggio strutturato di un edificio è l'impianto che permette il collegamento dei computer in rete locale e dei telefoni alla centrale telefonica dell'edificio stesso: a ogni postazione di lavoro l'utente collega il computer e/o il telefono e il fax utilizzando specifiche prese che sono poste di solito a muro o in apposite colonne.

Classificazioni equivalenti TIA e ISO				
Banda di frequenza	**TIA (Componenti)**	**TIA (Canale/Link)**	**ISO (Componenti)**	**ISO (Canale/Link)**
1 - 100 MHz	categoria 5	categoria 5	categoria 5	classe C
1 - 100 MHz	categoria 5e	categoria 5e	categoria 5e	classe D
1 - 250 MHz	categoria 6	categoria 6	categoria 6	classe E
1 - 500 MHz	categoria 6_A	categoria 6_A	categoria 6_A	classe E_A
1 - 600 MHz	n/s	n/s	categoria 7	classe F
1 - 1,000 MHz	n/s	n/s	categoria 7_A	classe F_A
1 - 10 GHz	n/s	n/s	categoria 8	classe G

La CAT 5e è una versione "rinforzata" (e = enhanced) della 5, presenta le stesse specifiche ma garantisce una maggiore affidabilità alle frequenze più alte della classe di certificazione.

Le prime tre categorie sono classificate in modo univoco dai vari istituti degli standard mentre le ultime sono ancora in fase di definizione e standardizzazione (6_A, 7, 8) per cui è facile trovare in commercio cavi di questo tipo con caratteristiche anche molto differenti tra loro: l'unica indicazione certa è che questi saranno orientati alle applicazioni di rete che richiedono frequenze da 600 MHz (si parla di cavi per applicazioni Gigabit Ethernet).

Zoom su...

MISURE SUI CABLAGGI: NOMENCLATURA

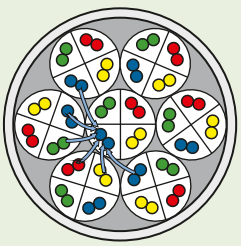
Sono state anche aggiornate le specifiche tecniche sulle misure da eseguire per certificare i cablaggi di tipo 6A: sostanzialmente i parametri di prova sono gli stessi già usati nella certificazione dei cablaggi in categoria 5e e categoria 6 (sono solamente stati rinominati alcuni test). Riportiamo di seguito una tabella dove è possibile confrontare i nomi vecchi e nuovi dei parametri di misura insieme ai loro acronimi.

Parametro di test – "Vecchio" nome	Parametro di test – "Nuovo" nome
Insertion Loss (IL)	*Insertion Loss* (IL)
Near End Crosstalk (NEXT)	*Near End Crosstalk* (NEXT)
Power Sum Near End Crosstalk (PSNEXT)	*Power Sum Near End Crosstalk* (PSNEXT)

Parametro di test – "Vecchio" nome	Parametro di test – "Nuovo" nome
Attenuation to Crosstalk Ratio (ACR)	*Attenuation to Crosstalk Ratio – Near End* (ACR-N)
Power Sum Attenuation to Crosstalk Ratio (PSACR)	*Power Sum Attenuation to Crosstalk Ratio – Near End* (PSACR-N)
Far End Crosstalk (FEXT)	*Far End Crosstalk* (FEXT)
Equal Level Far End Crosstalk (ELFEXT)	*Attenuation to Crosstalk Ratio – Far end* (ACR-F)
Power Sum Equal Level Far End Crosstalk (PSELFEXT)	*Power Sum Attenuation to Crosstalk Ratio – Far End* (PSACR-F)
Return Loss (RL)	*Return Loss* (RL)
Wire Map	*Wire Map*
Propagation Delay	*Propagation Delay*
Delay Skew	*Delay Skew*

Zoom su...

DIAFONIA ALIENA



Per le trasmissioni dati ad alta velocità come Ethernet 10GBase-T a 10 Gbps si richiede di controllare anche l'accoppiamento tra cavi adiacenti e cavi all'interno del pannello di permutazione (*patch panel*).

Si introduce inoltre il concetto di diafonia aliena (*Alien Crosstalk*, AC), un fenomeno di disturbo molto complesso da caratterizzare in quanto è legato alla diafonia osservata su una coppia che viene disturbata da altre coppie facenti parte di altri cavi (fasci) posti nelle vicinanze. ▶

Questa misura deve essere fatta anche per la certificazione di un cablaggio in categoria 6A:

- dapprima si eseguono i dieci test richiesti per la CAT 5A delle prestazioni di ogni singolo collegamento;
- deve essere poi aggiunta una prova sulla diafonia aliena.

Non è possibile realizzare il test della diafonia aliena su tutti i collegamenti del cablaggio sia per il tempo necessario che per la scarsa significatività del test stesso, dato che sperimentalmente si è dimostrato che i cavi appartenenti a fasci di cavi diversi non interagiscono in modo particolare tra loro.
Praticamente vengono effettuati due test:

- si analizza la diafonia aliena limitata ai cavi che appartengono allo stesso fascio o che sono terminati sul patch panel in posizioni adiacenti al cavo che si sta esaminando;
- si esegue un ulteriore test a campione sulla diafonia aliena per un sottoinsieme significativo dei collegamenti.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 L'acronimo AWG indica:
- ○ *Alternate Wire Gadget*
- ○ *American Wire Gage*
- ○ *American Wine Gage*
- ○ *Alternate Wire Gadget*

2 L'acronimo TDR indica:
- ○ *Time Data Reflectometer*
- ○ *Trasmit Data Realy*
- ○ *Time Domain Realy*
- ○ *Time Domain Reflectometer*

3 Le principali cause di rumore sui cavi sono le seguenti: (*indicare quella errata*)
- ○ crosstalk
- ○ RFI
- ○ NEXT
- ○ EMI

4 La schermatura dei cavi viene fatta con:
- ○ foglio
- ○ calza
- ○ maglia
- ○ foglio + calza
- ○ foglio + maglia
- ○ calza + maglia

5 Che cosa significa la sigla NEXT?
- ○ *New Est Crosstalk*
- ○ *Near End Crosstalk*
- ○ *Near Est Crosstalk*
- ○ *New End Crosstalk*

6 L'ACR, Attenuation to Crosstalk Ratio, misura:
- ○ il rapporto tra il segnale sorgente e il segnale indotto dalla coppia vicina
- ○ il rapporto tra il segnale attenuato e il segnale indotto dalla coppia vicina
- ○ il rapporto tra il segnale sorgente e il segnale indotto nella stessa coppia
- ○ il rapporto tra il segnale attenuato e il segnale indotto nella stessa coppia

7 La diafonia aliena è un fenomeno di disturbo dato:
- ○ da coppie facenti parte di altri cavi posti nelle vicinanze
- ○ da segnali esterni ai cavi che trasportano il segnale
- ○ da interferenze di campi elettromagnetici dovuti all'alimentazione
- ○ da tutte le interferenze non bene definite

Test vero/falso

1	Per i cablaggi strutturati si utilizzano cavi 22 o 24 AWG.	V F
2	Al diminuire dell'AWG diminuisce la sezione del conduttore.	V F
3	Con il TDR è possibile individuare se è presente un guasto in un conduttore.	V F
4	Con Insertion Loss si intende la perdita di segnale dovuta alla giuntura di due cavi.	V F
5	Il valore dell'attenuazione espresso in decibel è sempre negativo.	V F
6	La diafonia (crosstalk) viene anche indicata con la sigla NEXT.	V F
7	Esistono due tipi di diafonia: la paradiafonia e la telediafonia.	V F
8	La telediafonia viene anche indicata con la siglia FEXT.	V F
9	La categoria 5 TIA corrisponde alla classe D ISO.	V F
10	L'Alien Crosstalk (AC) è un fenomeno di disturbo dovuto soprattutto alle alte frequenze.	V F

Verifichiamo le competenze

Esprimi la tua creatività

Domande a risposta aperta

1 Con impedenza elettrica si intende .....

2 Il ritardo di propagazione è .....

3 Si definisce attenuazione .....

4 La schermatura nei cavi viene fatta con .....

5 La diafonia è .....

6 L'ELFEXT è ottenuto da .....

7 Insertion Loss è una misura in decibel che combina .....

8 La normativa ISO 11801/2002 aggiunge altri tre test a quelli della TIA-568B: .....

Esercizi di completamento

NEXT • Alien Crosstalk • rumore • Attenuation to Crosstalk Ratio • return loss • discontinuità • Insertion Loss • FEXT

1 Si definisce con ..... il rapporto tra la tensione del segnale in ingresso al cavo e la tensione misurabile all'altra estremità.

2 La protezione dal ..... nei cavi viene fatta con la schermatura.

3 Esistono due tipi di diafonia misurate direttamente: la ..... e la .....

4 L'..... esprime il rapporto tra il segnale attenuato presente su una coppia e il segnale indotto dalla coppia vicina.

5 La perdita di ritorno o ..... del segnale e causato dalla ..... nel cavo.

6 La ..... è legata alla diafonia osservata su una coppia che viene disturbata da coppie facenti parte di altri cavi posti nelle vicinanze.

UNITÀ DIDATTICA 3

LA CONNESSIONE OTTICA

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere le modalità di trasmissione di segnali ottici in fibra
- a effettuare i test sulle fibre ottiche

La trasmissione di segnali ottici in fibra

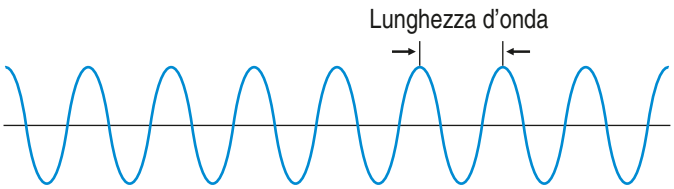
Oltre alla trasmissione di segnali elettrici si può utilizzare un secondo fenomeno fisico per trasmettere comunicazioni, la luce, che viene propagata utilizzando un particolare conduttore chiamato fibra ottica.

L'idea che sta alla base dello sfruttamento della luce per trasmettere informazioni binarie è stata quella di associare direttamente l'informazione all'accensione e allo spegnimento della luce facendo corrispondere per esempio allo 0 la presenza della luce e all'1 il buio.

I primi ostacoli incontrati nella realizzazione di fibre ottiche furono la loro elevata attenuazione del segnale (negli anni '50 era di circa 1000 dB/km) che le rendeva praticamente inutilizzabili.
La rivoluzione nelle telecomunicazioni avvenne quando nel 1970 i ricercatori della Corning Glass Works riuscirono a perfezionare una fibra ottica con attenuazione di "soli" 20 dB/km alla lunghezza d'onda di 633 nm (nanometri, 10^{-9} m): iniziò quindi la "sostituzione" dei cavi elettrici con cavi in fibra che a partire dagli anni '90 hanno trovato impiego anche per le reti locali: oggi le fibre ottiche hanno attenuazione di circa 0,2 dB/km.

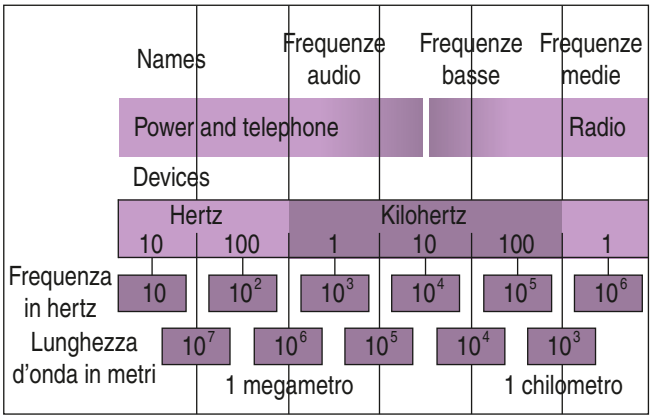
La trasmissione della luce

La luce è una particolare onda elettromagnetica e quindi, come tutte le onde elettromagnetiche, si propaga a una velocità di 300.000 km/s nel vuoto.
Le onde si classificano in base ai valori di frequenza f (o di lunghezza d'onda $\lambda = 1/f$). ▶

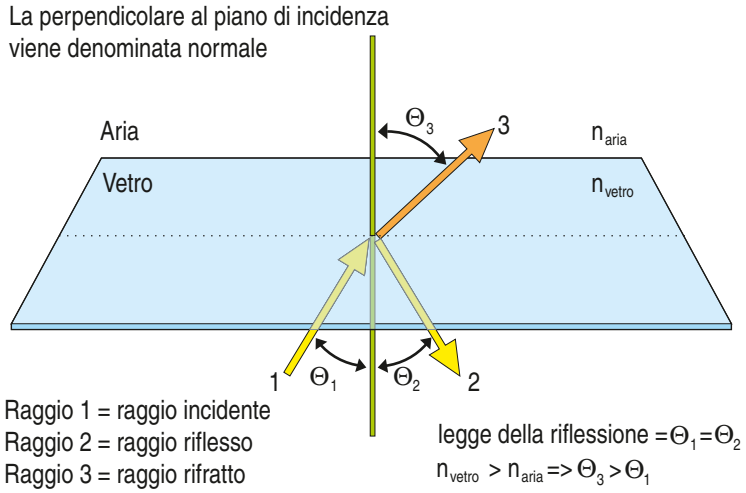


Per esempio, la gamma dei valori visibili dall'uomo è compresa tra 0,4 a 0,7 µm mentre la gamma utilizzata per la trasmissione nelle fibre ottiche è la luce infrarossa da 0,85, 1,35, 1,55 µm. ▶

Quando le dimensioni della fibra sono molto maggiori di quelle della lunghezza d'onda, le proprietà e i modi di propagazione dell'energia luminosa in una fibra ottica possono essere studiati semplicemente applicando le leggi dell'ottica geometrica. In particolare vengono utilizzate le leggi di Snell che studiano la riflessione e la rifrazione di un raggio luminoso incidente sulla superficie di separazione di due materiali.



Quando un raggio di luce attraversa due materiali con un angolo θ diverso da 90° una parte di energia viene riflessa mentre l'altra entra nel materiale con un angolo diverso da quello di ingresso e viene rifratta: siamo in presenza di un'onda incidente, di un raggio riflesso e di un raggio rifratto.



L'angolo di incidenza θ_1 è uguale all'angolo di riflessione θ_2 mentre l'angolo di rifrazione è in relazione al tipo di materiale, secondo un parametro specifico di ogni materiale chiamato indice di rifrazione, indicato con n.

L'indice di rifrazione di un materiale è dato dal rapporto tra la velocità della luce nel vuoto e la velocità della luce nel materiale stesso:

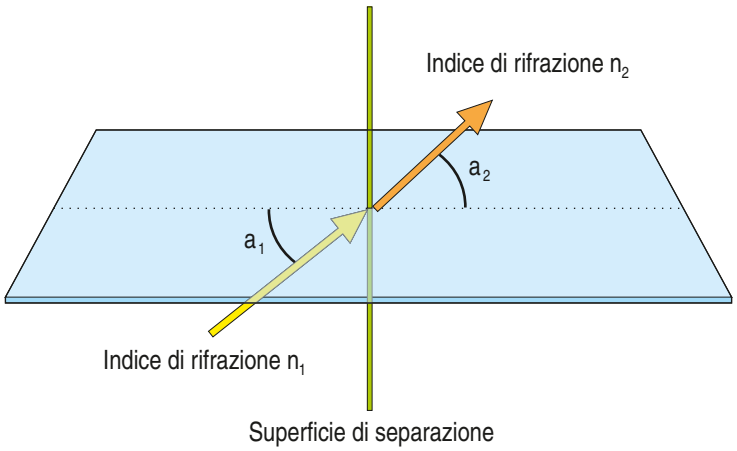
$$\text{indice di rifrazione } n = \frac{\text{velocità della luce nel vuoto}}{\text{velocità della luce nel materiale}}$$

Naturalmente l'indice di rifrazione nell'aria ha valore 1 mentre per tutti i materiali ha valore >1, essendo la velocità della luce nei materiali più densi sempre inferiore a quella nel vuoto. Nella tabella sono riportati i valori dell'indice di rifrazione di alcuni materiali.

Materiale	Indice di rifrazione
Aria	1
Acqua	1,333
Plastica	1,47
Vetro	1,523
Diamante	2,418

Come si può vedere dalla figura a lato, la legge di Snell lega l'angolo di incidenza α_1 di un raggio luminoso che attraversa un materiale con indice di rifrazione n_1 e "passa attraverso" la superficie di separazione di un secondo materiale con indice n_2 formando un angolo α_2. ▶
La relazione tra i due angoli è:

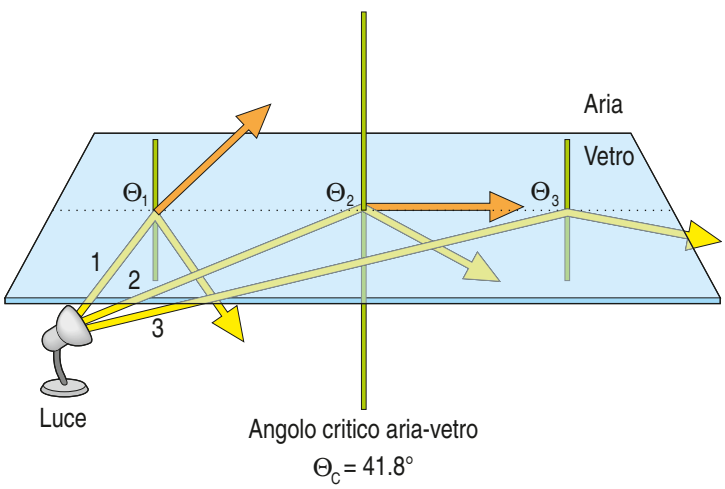
$$n_1 \text{ sen } \alpha_1 = n_2 \text{ sen } \alpha_2$$



La parte di luce che viene rifratta "sottrae" energia a quella che viene riflessa: il nostro obiettivo è quello di ottenere un sistema in grado di riflettere totalmente la luce in modo da trasferire il segnale più lontano possibile.

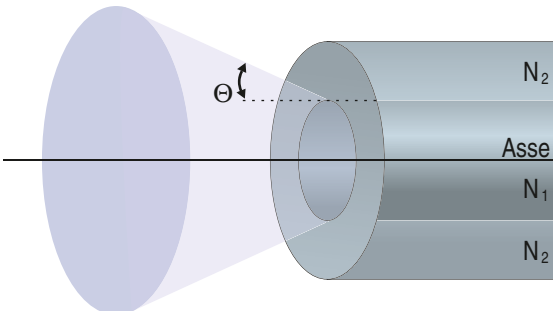
Dalla legge di Snell si può ricavare un valore particolare dell'angolo di incidenza che prende il nome di angolo critico, α_c, tale che per i valori dell'angolo di incidenza superiori a esso si ha la riflessione totale.

$$\alpha_c = \sin^{-1}\left(\frac{n_2}{n_1}\right)$$



Raggio 1: $\theta_1 < \theta_c$: parte della luce viene riflessa e parte viene rifratta
Raggio 2: $\theta_1 = \theta_c$: parte della luce viene riflessa e parte viene rifratta
Raggio 3: $\theta_1 > \theta_c$: viene totalmente riflessa

Per trasmettere la luce si realizzano quindi dei conduttori particolari che prendono il nome di fibre ottiche composti da due materiali disposti in modo coassiale come nella figura a lato. ▶
Al centro viene posto un filo di vetro di dimensioni micrometriche, che prende il nome di core (o nucleo). Esso viene avvolto da un mantello esterno (cladding) che ha un indice di rifrazione diverso dal vetro (generalmente ha valori n_2 = 1,475 per il cladding e n_1 = 1,52 per il core).



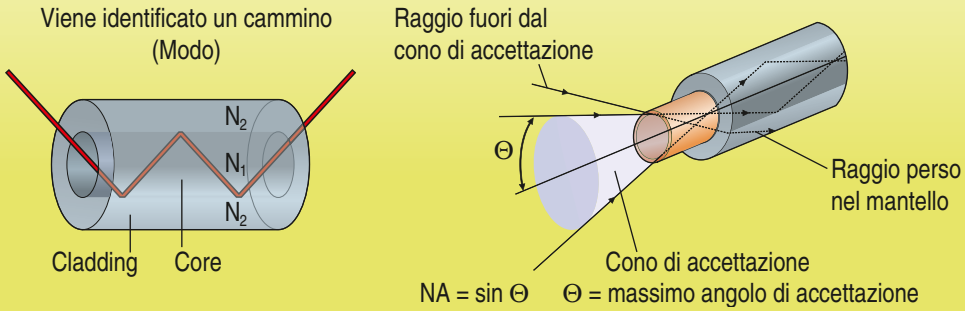
L'angolo critico è quindi $\alpha_c = \sin^{-1}(1{,}475/1{,}52) \cong 79{,}5$ gradi e al di sotto di questo angolo di incidenza la luce viene totalmente riflessa, quindi si mantiene tutta all'interno del core.

CONO DI ACCETTAZIONE, APERTURA NUMERICA, MODO

Si definisce cono di accettazione l'insieme degli angoli di incidenza per i quali avviene la riflessione totale del segnale nella fibra.
Si definisce apertura numerica (NA) di una fibra il "range" di angoli di luce incidente che saranno completamente riflessi (il suo valore varia in genere tra 0,1 e 0,3).

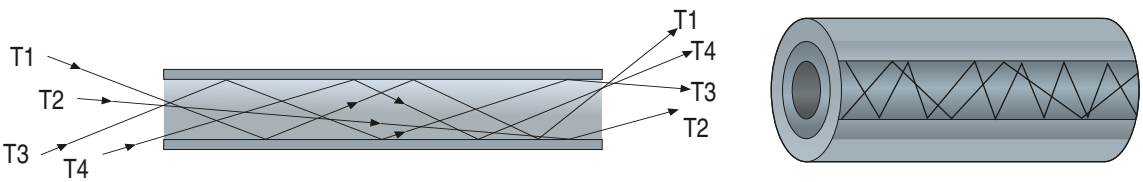
$$NA = (n_2^2 - n_1^2)^{1/2}$$

Si definisce modo il percorso che un raggio di luce segue quando attraversa una fibra.



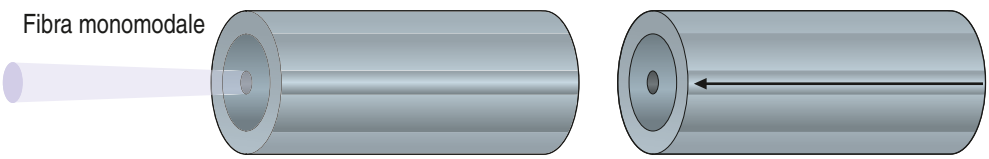
Se in una fibra viene trasmesso un unico segnale luminoso prende il nome di fibra monomodale (single mode) mentre le fibre ottiche che ammettono più modi di propagazione vengono dette multimodali (multimode).

Nella figura seguente viene riportato un esempio di fibra multimodale con 4 modi.



Si può osservare come all'aumentare del diametro del core i cammini percorsi nei diversi modi hanno lunghezza diverse: dal disegno si vede immediatamente che il percorso T2 è molto più breve del percorso T3. Di conseguenza avremo tempi di propagazione diversi, per cui segnali inviati contemporaneamente giungeranno a destinazione in tempi diversi (fenomeno della dispersione): il segnale che arriva per ultimo pone un limite inferiore di permanenza alla durata dell'impulso (valore 0 o valore 1) limitando di fatto la velocità di trasmissione.

L'eliminazione del problema della dispersione modale si ottiene solo con le fibre monomodali: in esse si riesce a ridurre la dimensione del core fino a circa 4-10 μm e la propagazione dei raggi avviene in un unico modo.



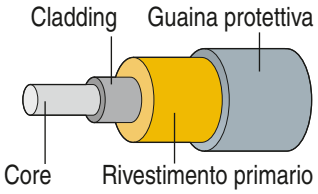
Nelle fibre monomodali è però necessario avere una sorgente di segnale che generi un fascio preciso di luce monocromatica. Generalmente viene utilizzato un laser che genera una luce monocromatica e coerente mentre nelle fibre multimodali è possibile trasmettere con nomali LED, più economici e semplici da utilizzare.

I vantaggi di questo tipo di fibra sono:
- elevato tempo di vita;
- assenza di dispersione;
- minima perdita della potenza ottica;
- bassa attenuazione;
- ampia larghezza di banda.

Gli svantaggi si limitano fondamentalmente a due punti:
- problemi di connessione a causa del piccolissimo diametro del core;
- elevata potenza ottica richiesta.

La struttura di una fibra ottica

Abbiamo detto che una fibra ottica si presenta come un cavo coassiale dove al centro, al posto del cavo di rame, viene posto un filo di vetro di dimensioni micrometriche che prende il nome di core (o nucleo). Questo viene avvolto da un mantello esterno (cladding) che ha un indice di rifrazione molto diverso dal vetro: in questo modo si mantiene la luce all'interno del core. ▶
Core e cladding vengono poi ricoperti da due strati di materiali protettivi.



I vantaggi delle fibre ottiche rispetto ai cavi elettrici sono i seguenti:
- sono completamente immuni dai disturbi elettromagnetici: il segnale trasportato consiste in fotoni, che sono elettricamente neutri;
- permettono trasmissioni ad alta velocità: esistono fibre ottiche operative a 2 Gb/s;
- hanno una bassa attenuazione del segnale: alcuni decimi di dB/km;
- sono di dimensioni molto ridotte e quindi molto comode per i cablaggi strutturati;
- hanno costi contenuti.

Lo svantaggio più evidente per le fibre ottiche è il fatto che possono essere utilizzate solo per connessioni punto-punto dato che non è possibile prelevare il segnale in un punto intermedio ma solo alla sua estremità.

Oggi le dimensioni caratteristiche di una fibra ottica sono le seguenti:
- il diametro delle fibre è di 125 μm;
- il diametro completo del cavo, compreso i rivestimenti, è di circa 0,25 mm.

Caratteristiche costruttive dei cavi in fibra ottica

Ogni cavo in fibra ottica è composto da due fibre, una per il trasmettitore (Tx) e una per il ricevitore (Rx) per cui la trasmissione è full duplex e le due fibre sono messe in un singolo contenitore fino ad arrivare ai connettori.
Le coppie di fibre ottiche sono raggruppate all'interno di un cavo che può arrivare fino a 24 coppie.

Il connettore più comune per le fibre multimode è SC (*Subscriber Connector*), mentre per le single mode è ST (*Straight Tip*). ▶

A seconda della diversa tecnologia costruttiva possiamo avere tre tipologie di cavi:

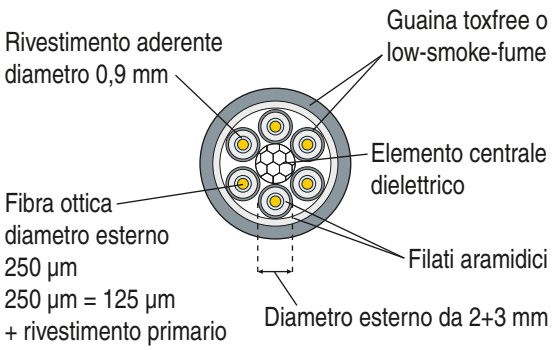
- cavi di tipo tight;
  - cavi multimonofibra
  - cavi multifibra
- cavi di tipo loose;
- cavi di tipo slotted core.

I cavi di tipo tight sono usati principalmente per installazioni in luoghi interni e si suddividono in:

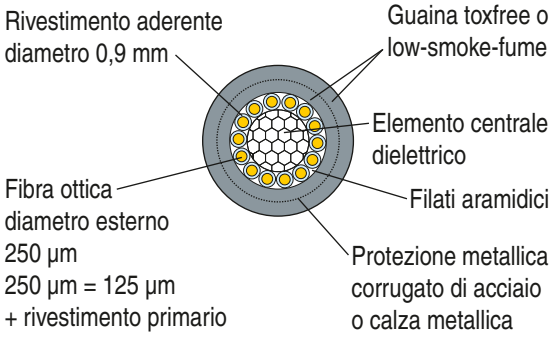
- cavi multimonofibra che sono costituti al massimo da otto fibre disposte attorno a un nucleo centrale di materiale dielettrico. Sono particolarmente robusti in quanto ogni singola fibra viene rivestita con una guaina protettiva fino ad arrivare a un diametro di 2-3 mm. Vengono per esempio utilizzati per realizzare le bretelle ottiche che assicurano le permutazioni tra gli apparati attivi e i pannelli di terminazione (multifibra a due fibre, chiamati bifibra);
- cavi multifibra che sono meno robusti dei cavi multimonofibra dato che il rivestimento della singola fibra arriva al massimo a un diametro di 0,9 mm: per questo motivo, però, possono contenere con lo stesso diametro globale del cavo un numero maggiore di fibre ottiche (fino a 32) e quindi sono maggiormente indicati per la realizzazione delle dorsali.

I cavi di tipo loose sono usati principalmente per installazioni in luoghi esterni e possono contenere fino a 100 fibre: sono particolarmente indicati per la connessione in ambienti umidi e in presenza di acqua in quanto hanno più strati di protezione con particolari gel che tamponano il cavo.
Non sono invece indicati per cablaggi verticali e devono essere giuntati utilizzando cavetti monofibra.
I cavi di tipo slotted core sono usati principalmente per installazioni in luoghi esterni e possono contenere fino a 400 fibre: sono particolarmente indicati per la connessione in ambienti umidi e in presenza di acqua in quanto hanno anch'essi una protezione tamponante con gel.
Sono costituiti da un elemento centrale scanalato entro le cui cavità vengono riposte le fibre nude; anch'essi devono essere giuntati utilizzando cavetti monofibra.
Per le LAN in genere si usano le tight all'interno degli edifici mentre le loose si usano per installazioni esterne agli edifici.

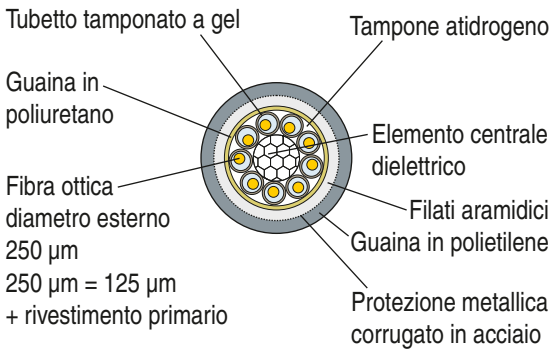
Cavo multimonofibra



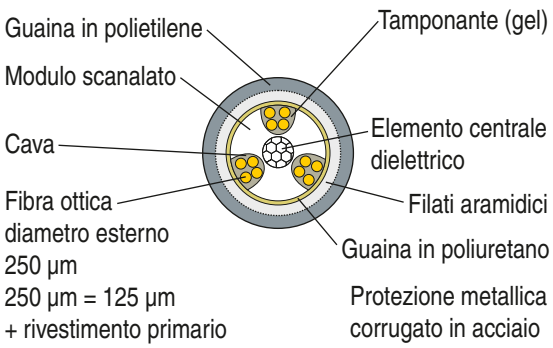
Cavo multifibra



Cavo di tipo loose



Cavo di tipo slotted core



Installazione, rumore e test sulle fibre ottiche

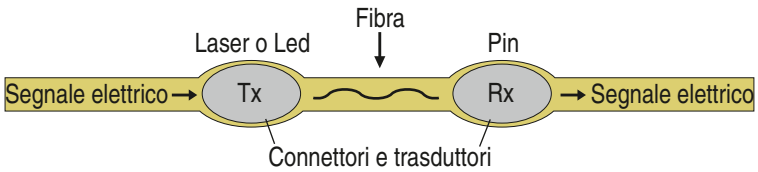
Un collegamento in fibra ottica viene realizzato con tre componenti, come ogni altro sistema di comunicazione:

- il trasmettitore (Tx);
- il cavo;
- il ricevitore (Rx).

Il segnale da trasmettere è binario, quindi costituito da una sequenza di zero e di uno, trasformati in presenza o assenza di segnale luminoso. Il segnale luminoso viene generato da due tipi di sorgenti che ricevono il segnale elettrico da trasmettere e lo convertono in segnali ottici:

1. i LED (diodi a emissione luminosa) usati nelle fibre multimode (producono una luce infrarossa di lunghezza d'onda di 850 nm o di 1310 nm);
2. i laser, usati nelle fibre single mode (producono un raggio di luce infrarossa con lunghezza d'onda di 1310 nm o di 1550 nm).

Come ricevitore viene utilizzato un fotodiodo PIN, che reagisce a una specifica lunghezza d'onda.



L'istallazione dei connettori sulle fibre ottiche è un'operazione molto delicata ed è la prima causa della perdita di segnale: in presenza di una discontinuità del conduttore parte della luce viene riflessa e quindi viene persa energia del segnale.
Anche la corretta stesura del cavo è un'operazione fondamentale per avere le massime prestazioni nella comunicazione: la fibra viene inserita in una guida (interducting) che permette di evitare curve troppo strette, piegature o schiacciature dannose. Agli estremi la fibra deve essere tagliata perfettamente, smussata e accuratamente pulita, così come è necessario avere particolare accortezza quando vengono effettuate delle giunture.

Rumore nelle fibre ottiche

A differenza dei cavi in rame le fibre non sono soggette a rumore esterno e crosstalk in quanto non sono presenti campi elettromagnetici indotti, ma sono presenti altre cause di perdita di energia luminosa, descritte di seguito.

Dispersione

Il principale problema che crea l'attenuazione della luce in una fibra ottica è causato da microscopiche discontinuità della fibra (distortions) che causano riflessioni indesiderate: questa prima causa di perdita di energia prende il nome di scattering (dispersione).

Assorbimento

La seconda causa di perdita di energia è dovuta alla presenza nella fibra di alcune impurità chimiche che assorbono (absorption) parte dell'energia riscaldando quel punto e confondendo i segnali.

Attenuazione per rifrazione

Le irregolarità presenti nella fibra tra il core e il cladding causano piccole rifrazioni e quindi perdite di energia per attenuazione.

Bending
Durante la stesura della fibra, se questa viene tesa più del dovuto, si possono creare microfratture nel core che creano dispersione di raggi luminosi; un analogo problema può essere causato da raggi di curvatura tropo stretti che possono determinare una modifica del valore dell'angolo di incidenza dei raggi.

Terminazione
Creano attenuazione tutte le irregolarità e i difetti dovuti alla terminazione non perfetta alle estremità.

Test e strumentazione
È indispensabile effettuare un insieme di test sui cavi ottici, a partire dalla verifica del corretto assemblaggio dei connettori. Gli strumenti utilizzati per i test su fibra ottica sono i seguenti:
- sorgenti luminose (*Visual Fault Locator*, VFL);
- microscopi per la fibra;
- *Optical Loss Test Kits* (OLTS) o *Power Meter*;
- *Optical Time Domain Reflectometers* (OTDR);
- piattaforme integrate di test.

Il VFL emette mediante un laser una luce bianca o rossa nello spettro del visibile; è tascabile, leggero e viene utilizzato per verificare la continuità della fibra e la polarità di una coppia di fibre oppure per individuare il punto di rottura della fibra con guaine di cavo sottili (in questo caso viene emessa luce lampeggiante o incandescente facilmente visibile). ▶
Durante l'installazione della fibra, una causa di criticità è dovuta alla pulizia delle terminazioni: la presenza di polvere, di residui di imballo o anche delle impronte digitali lasciate "maldestramente" dall'installatore portano a un'elevata attenuazione ottica.

> Statisticamente circa il 90% dei problemi di trasmissione sulle fibre ottiche è dovuto all'attenuazione causata da questo motivo.

In commercio sono disponibili appositi kit in valigette dedicate per la pulizia della fibra, che contengono tutto il necessario per rimuovere le sedimentazioni dello sporco nelle connessioni delle fibre (contengono un solvente speciale sviluppato per i cavi di fibra di vetro che viene applicato mediante un apposito spinotto in modo da avere sempre il preciso dosaggio, tamponi appositi per le diverse porte dei dispositivi attivi, cleaning card specifiche). ▶

È necessario ispezionare con una sonda video (microscopio per la fibra) anche le porte degli apparati attivi (router, switch, NICs) che possono sporcarsi. ▶
Inoltre il microscopio per la fibra viene utilizzato per verificare la presenza di righe o scheggiature sui connettori, la pulizia della faccia delle terminazioni dopo l'installazione, la pulizia dei connettori delle bretelle (*patch cord*) prima dell'uso, delle porte ottiche, delle porte patch panel e in generale la pulizia delle porte in fibra degli apparati.

La verifica della continuità di una fibra e la misura della perdita di segnale per attenuazione all'interno viene effettuata con un *Optical Loss Test Set* (OLTS) che, mediante una sorgente di luce sia monomodale che multimodale, a LED o a laser, misura la perdita di inserzione (attenuazione) di un link in fibra.
È molto semplice da usare ed è la soluzione meno costosa per il test di attenuazione della fibra ottica da utilizzare nella certificazione dei cablaggi. ▶

Per i collegamenti su lunga distanza si introduce un parametro, il power budget, che sta a indicare la massima quantità di potenza che può essere dissipata dal collegamento: viene calcolato come differenza tra il valore di potenza del trasmettitore e la sensibilità del ricevitore.
È sempre necessario misurare o stimare la perdita complessiva del collegamento, o link loss, che deve essere inferiore al power budget.
La misura viene effettuata utilizzando un OTDR (*Optical Time Domain Reflectometry*), cioè un riflettometro ottico nel dominio del tempo, che è in grado di calcolare le attenuazioni e le riflessioni per ogni componente del "canale ottico" e di "fotografare" il comportamento di ogni componente visualizzando la traccia del segnale.

Nella figura a lato è riportato un esemplare di riflettometro dove è possibile visualizzare la traccia del segnale. ▶

Zoom su...

CALCOLO DEL POWER BUDGET

Generalmente viene lasciato come power budget aggiuntivo un margine di 3-6 dB per precauzione, contro la perdita dovuta all'invecchiamento e allo sporco.
Vediamo a titolo di esempio il conteggio per un collegamento di 30 km, dove sono presenti otto giunzioni lungo il collegamento e i connettori posti solo all'estremità:

- attenuazione della fibra 0,24 dB/km * 30 km = 7,2 dB;
- perdita nelle giunzioni 0,1 dB/giunzione * 8 = 0,8 dB;
- perdita nei connettori 0,5 dB/connettore è 4 = 2 dB.

In totale si ottiene un link loss di 10 dB al quale dobbiamo aggiungere un margine di sicurezza di 5 dB ottenendo quindi un power budget minimo di 15 dB.

Riassumiamo nella tabella che segue i test che vengono effettuati sulle fibre ottiche un e gli strumenti utilizzati.

Tipo di test	Strumento
Accettazione cavi	OTDR, OLTS
Verifica di continuità	OTDR, OLTS, Laser Visible
Verifica stress cavo	OTDR
Riflessione	OTDR
Analisi terminazione	Microscopio
Difetti su pigtail/jumper	Laser Visible
Individuazione interruzioni	OTDR, Fiber break locato
Test del sistema	OLTS
Verifica attenuazione	OTDR

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Le fibre ottiche hanno oggi un'attenuazione di circa:

- ○ 20 dB/km
- ○ 2 dB/km
- ○ 0,2 dB/km
- ○ 2 dB/m
- ○ 0,2 dB/m

2 L'indice di rifrazione di un materiale si ottiene dal rapporto tra:

- ○ la velocità della luce nel materiale e la velocità della luce nel vetro
- ○ la velocità della luce nel vetro e la velocità della luce nel materiale
- ○ la velocità della luce nel vuoto e la velocità della luce nel materiale
- ○ la velocità della luce nel materiale e la velocità della luce nel vuoto

3 Quale, tra i seguenti, non è un vantaggio per una fibra monomodale:

- ○ elevato tempo di vita
- ○ assenza di dispersione
- ○ bassa potenza ottica richiesta
- ○ minima perdita della potenza ottica
- ○ bassa attenuazione
- ○ ampia larghezza di banda

4 Per realizzare le bretelle ottiche vengono utilizzati:

- ○ cavi di tipo tight multifibra
- ○ cavi di tipo tight multimonofibra
- ○ cavi di tipo loose
- ○ cavi di tipo slotted core

5 Per le LAN in genere si usano: (due scelte)

- ○ tight all'interno degli edifici
- ○ tight per le installazioni esterne agli edifici
- ○ loose all'interno degli edifici
- ○ loose per le installazioni esterne agli edifici
- ○ slotted all'interno degli edifici
- ○ slotted per installazioni esterne agli edifici

6 Per misurare l'attenuazione in un fibra ottica si utilizza:

- ○ VLS (*Visual Fault Locator*)
- ○ microscopio per la fibra
- ○ OLTS (*Optical Loss Test Set*)
- ○ OTDR (*Optical Time Domain Reflectometers*)

Test vero/falso

1	L'indice di rifrazione nell'aria ha valore 1.	V F
2	L'indice di rifrazione per tutti i materiali ha valore <1.	V F
3	In una fibra monomodale viene trasmesso un unico segnale luminoso.	V F
4	Nelle fibre monomodali il core viene ridotto fino a circa 4-10 mm.	V F
5	I cavi di tipo loose sono usati principalmente per installazioni in luoghi esterni.	V F
6	I cavi di tight multifibra possono contenere fino a 100 fibre.	V F
7	I cavi di tipo loose possono contenere fino a 100 fibre.	V F
8	I cavi di slotted loose possono contenere non più di 300 fibre.	V F
9	I LED sono usati nelle fibre multimode.	V F
10	I laser producono una luce infrarossa di lunghezza d'onda di 850 nm o di 1310 nm.	V F
11	L'analisi della terminazione di una fibra viene effettuata mediante il microscopio.	V F
12	La verifica dell'attenuazione in una fibra viene effettuata con l'OLTS.	V F

Verifichiamo le competenze

Esprimi la tua creatività

Domande a risposta aperta

1 L'indice di rifrazione di un materiale si ottiene dal rapporto tra .....

2 L'angolo critico α_c è particolare perché per i valori dell'angolo .....

3 Si definisce apertura numerica (NA) di una fibra .....

4 In una fibra monomodale .....

5 I cavi di tipo loose sono usati principalmente .....

6 Il segnale luminoso viene generato da due tipi di sorgenti:

1) .....: .....

2) .....: .....

7 Con il termine di "scattering" si intende .....

8 La verifica della continuità di una fibra e la misura della perdita di segnale per attenuazione al suo interno viene effettuata .....

Esercizi di completamento

slotted core • core • OTDR • onda elettromagnetica • vetro • VFL • cladding • link loss • tight • multimonofibra • multifibra

1 La luce è una particolare .....
e quindi si propaga a una velocità di ..... km/s nel vuoto.

2 Compila la seguente tabella con i rispettivi indici di rifrazione:

Aria	
Acqua	
Plastica	
Vetro	
Diamante	

3 Al centro viene posto un filo di vetro di dimensioni micrometriche, il .....
(o nucleo) che viene avvolto da un mantello esterno (.....)
che ha un indice di rifrazione diverso dal .....

4 I cavi di tipo ..... sono usati principalmente per installazioni all'interno di edifici e si suddividono in .....
e .....

5 I cavi di tipo ..... possono contenere fino a 400 fibre.

6 Il ..... emette mediante un laser una luce bianca o rossa nello spettro del visibile.

7 La misura del ..... viene effettuata utilizzando un .....

UNITÀ DIDATTICA 4

LA CONNESSIONE WIRELESS

IN QUESTA UNITÀ IMPAREREMO...

- la modalità di trasmissione dei segnali wireless
- la sicurezza nelle comunicazioni wireless

La trasmissione di segnali wireless

La caratteristica delle reti wireless è quella di essere "senza cavi", come dice il termine stesso: l'utilizzo di dispositivi in grado di comunicare i messaggi utilizzando come mezzo conduttore l'aria permette di realizzare reti senza fili.

> Le reti wireless da una parte soddisfano le esigenze in situazioni in cui il cablaggio può risultare non conveniente, dall'altra vengono incontro all'esigenza di connettività da parte di dispositivi elettronici mobili, come palmari, cellulari, laptop ecc. che comunicano tra loro attraverso interfacce aeree.

Al primo standard per le comunicazioni wireless creato dalla IEEE negli Stati Uniti nel 1995 fu attribuito il codice 802.11, che consiste in un famiglia di protocolli definiti per realizzare una versione radio del protocollo 802.3/Ethernet.

Le tecnologie più usate per realizzare reti senza fili sono:
- raggi infrarossi;
- onde radio.

La tecnologia a raggi infrarossi viene utilizzata da oltre vent'anni: è quindi "matura" e molto semplice, ma presenta il grosso svantaggio di essere estremamente sensibile agli ostacoli. Anche una piccola parete in cartongesso rappresenta un impedimento considerevole dal punto di vista elettromagnetico. Può quindi essere utilizzata solo in uno spazio aperto o all'interno di un singolo locale.
Per l'utilizzo in una rete locale che si estende all'interno di edifici l'unica soluzione è quella che utilizza le onde radio: il protocollo 802.11 prevede infatti che la propagazione dei segnali avvenga utilizzando questo tipo di onde.
La trasmissione dei bit viene effettuata generalmente in modulazione di fase differenziale, dove la portante viene modulata con uno sfasamento di 180° per la codifica dell'1 rispetto a quella dello 0.

Le trasmissioni occupano una banda di frequenze che viene suddivisa in tanti canali tra loro separati; per trasmettere su un particolare canale è necessario possedere il rispettivo codice di autorizzazione.

Il protocollo 802.11 è stato pensato:
- per reti non cablate: per gestire il colloquio fra tante stazioni che comunicano tra loro in modo paritetico;
- per reti cablate: per l'accesso a una rete più articolata da parte di un dispositivo per la realizzazione di Distributed System (Sistema di distribuzione dell'informazione).

I dispositivi comunicano tra loro a 2.4 GHz con velocità da 1 a 2 Mbps.

Nel protocollo sono definite:
1. le modalità in cui diverse stazioni costituiscono un'Ad Hoc Network, cioè una rete *ad hoc* wireless in ambito locale;
2. le modalità per cui le stazioni appartenenti a singole Ad Hoc Network possono colloquiare con un Distributed System attraverso dei punti di accesso, gli *Access Point* (AP).

La successiva evoluzione, l'802.11b, arriva fino a 11 Mbps ed è chiamata Wi-Fi o wireless ad alta velocità: di fatto, però, la maggior parte delle apparecchiature lavora a 2 o a 4 Mbps.
La diffusione a livello mondiale di dispositivi predisposti alla connessione Wi-Fi sta incrementando la richiesta di connettività "mobile" ed è in costante aumento la disponibilità di connessione offerta dalle LAN aziendali di alberghi, ristoranti, caffè, infopoint, centri commerciali, aeroporti ecc. per offrire servizi (come la connessione a Internet) o informazioni ai proprio clienti.

Zoom su...

Esistono altri protocolli nella famiglia 802.11, per esempio:
- 802.11a: per i dispositivi wireless che operano a 5 GHz e arrivano fino a 54 Mbps;
- 802.11g: utilizza tecniche di modulazione OFDM.

Realizzazione di una rete wireless

Per realizzare una rete wireless sono necessari almeno due dispositivi wireless, entrambi con possibilità di connessione radio. Per garantire la compatibilità tra i diversi dispositivi, viene utilizzato un *Access Point* (AP) che ha funzioni di concentratore e "collegatore" tra la rete wireless e la LAN cablata.
L'area coperta da un AP viene chiamata cella e ha un raggio massimo di circa 150 metri: per coprire spazi di dimensioni maggiori è necessario predisporre più AP connessi alla stessa LAN per ricoprire tutte le aree da raggiungere (è necessario che le aree raggiungibili da ogni AP siano parzialmente sovrapposte per almeno il 20%).
La LAN con accesso wireless prende anche il nome di WLAN.

Quando un dispositivo vuole connettersi a una WLAN esegue uno scanning alla ricerca di un AP compatibile con il quale eseguire l'associazione (associate).
Lo scanning può essere:
- attivo (*active*): il client lancia una richiesta per unirsi alla LAN contenente un SSID (*Service Set Identifier*) e, se viene accettata analizzando le credenziali del richiedente e controllando gli eventuali diritti di accesso, l'AP gli risponde permettendogli il collegamento;
- passivo (*passive*): il client si pone in ascolto di messaggi (bacon) trasmessi dall'AP in attesa di riceverne uno contente l'SSID della rete alla quale desidera connettersi.

SSID (SERVICE SET IDENTIFIER)

L'SSID è il nome con cui una rete Wi-Fi si presenta ai suoi utenti: consiste in genere in una serie di caratteri ASCII stampabili e viene continuamente trasmesso in modo che gli utenti possano individuare la presenza della rete di loro interesse alla quale connettersi.

L'autenticazione per l'accesso a una WLAN è a livello 2 e quindi viene autenticato il dispositivo e non l'utente: se la LAN offre un servizio di "sistema aperto", cioè permette libero accesso ad alcuni servizi (tipo la connessione Internet) da parte di chiunque ne faccia esplicita richiesta: è sufficiente che coincidano il SSID della LAN e del client. Per accessi a rete privata o LAN aziendali l'AP viene configurato per inviare la richiesta a un server di autenticazione oppure può effettuare direttamente la validazione mediante una chiave condivisa: in questo caso è richiesta la crittografia WEP (*Wireless Equivalent Protocol*) a 64 o 128 bit della comunicazione per evitare intrusioni indesiderate: all'AP e a tutti i nodi viene assegnata staticamente la chiave di accesso, e dal semplice confronto di questa si determina l'autenticazione del client. A seguito dell'autenticazione l'AP effettua l'associazione e quindi autorizza il client a connettersi alla rete e a trasferire i dati.

Comunicazione wireless

Il primo problema che si incontra utilizzando le trasmissioni in Radio Frequenza (RF) è la condivisione del mezzo trasmissivo da parte di più soggetti che utilizzano l'etere come mezzo di comunicazione. Sono presenti molteplici fonti di disturbo, come i telefoni wireless, il forno a microonde, i dispositivi che utilizzano la tecnologia Bluetooth (che usa l'intero spettro a 2.4 GHz); anche le condizioni atmosferiche interferiscono con i segnali, soprattutto la nebbia e i temporali.

Il secondo problema è legato al fatto che più client possono contemporaneamente cercare un trasmettere dati allo stesso AP generando quindi delle collisioni; nei paragrafi che seguono spiegheremo come avviene la comunicazione e quali sono i meccanismi che ci permettrono di gestire le collisioni.

Accesso al canale

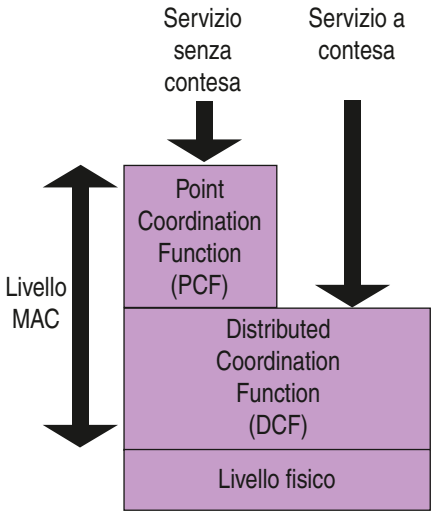
Sostanzialmente sono due le modalità di trasmissione wireless:

- ad hoc, o IBSS (*Independent Basic Service Set*): senza infrastruttura, è una connessione peer to peer utilizzata per sale conferenze, condivisione di periferiche, allineamento archivi ecc.;
- infrastrutturata, o EES (*Extended Service Set*): con un AP connesso alla rete cablata o a Internet, che richiede autenticazione e permessi di accesso, è utilizzata nelle reti aziendali, in ufficio, in reti domestiche, in hotspot ecc.

In entrambe per poter comunicare è necessario avere l'accesso a un canale, e questo può effettuarsi in due diverse modalità:

- DFC: *Distributed Coordination Function*;
- PFC: *Point Coordination Function*.

Senza entrare nel dettaglio dei due meccanismi, il primo può essere utilizzato sia per connessioni *ad hoc* che strutturate, poiché non gestisce priorità ma gestisce un meccanismo equo di acceso al canale; il secondo può essere utilizzato solo per connessioni strutturate, e dà all'AP la massima priorità nell'interrogare gli host in polling garantendo un servizio per ogni client. ▶



Nella figura è rappresentato il modello a strati del 802.11 sul livello fisico: la PCF opera al di sopra della DCF e ne sfrutta le caratteristiche per assicurare un accesso privilegiato ai suoi utenti.

Elusione delle collisioni

◀ Il MAC è un sotto-livello del livello Data Link, cioè il livello 2 della pila ISO/OSI, che si occupa della connessione dei dispositivi di collegamento dei dati: sarà dettagliatamente descritto nel prossimo modulo didattico. ▶

Per risolvere il problema di collisioni le WLAN utilizzano il protocollo ◀ livello MAC ▶ chiamato CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*), cioè accesso multiplo con ascolto della portante ed elusione delle collisioni (che sarà dettagliatamente descritto nel seguito della trattazione).

Dato che il client non si accorge delle collisioni, è il ricevente che deve gestire la comunicazione:

- il trasmettitore quando trova un canale libero invia un pacchetto di dati;
- il ricevente invia un segnale di conferma ricezione ACK;
- il trasmittente, se non riceve l'OK dal ricevente, considera il pacchetto perso e ripete la trasmissione.

La collisione avviene quando due emittenti sentono che un canale è libero e iniziano contemporaneamente a trasmettere; per eludere le collisione si utilizza il CA:

- il mittente inizia la comunicazione trasmettendo un pacchetto molto corto RTS (*Request To Send*);
- il destinatario risponde con un pacchetto CTS (*Clear To Send*);
- le altre emittenti prima di trasmettere si mettono in ascolto (carrier sensing), ricevono questi pacchetti e li interpretano come segnali di canale occupato e quindi evitano di trasmettere in intervalli successivi di tempo.

Zoom su...

FRAME 802.11

La comunicazione in una WLAN non utilizza i frame standard 802.3. Esistono tre tipi di frame: control, management e data, e solo il frame data è simile a 802.3, con una larghezza che da 1500 byte può arrivare fino a 2346 byte.

La sicurezza nelle comunicazioni wireless

SICUREZZA DI UN SI

Con il termine "sicurezza" di un sistema informatico si intende l'insieme dei meccanismi che vengono adottati per garantire:

- confidenzialità: terze parti non possono accedere al sistema;
- integrità: chi non è autorizzato non deve modificare le risorse (in particolare i dati);
- autenticazione: solo gli utenti autorizzati possono accedere alle risorse dopo che il sistema ha effettuato un'affidabile identificazione.

Per le reti cablate la sicurezza è importante ma per poter accedere a esse è necessario avere "fisicamente" una connessione, questo significa che un malintenzionato deve connettere un cavo alla rete per poter accedere; per le reti wireless la facilità di connessione è favorita dalla mancanza di questo ostacolo. Se noi "chiudiamo in una stanza" la nostra rete LAN e non facciamo accedere nessuno al locale possiamo stare abbastanza tranquilli che nessun estraneo accederà alla nostra

rete: invece con accessi Wi-Fi non abbiamo alcun controllo fisico e quindi chiunque si trovi entro il raggio di copertura del trasmettitore dell'AP può ascoltare, intervenire e immettere traffico interferendo nella rete.

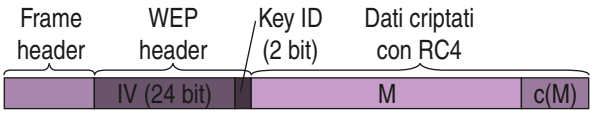
> È anche difficile delimitare la zona di copertura perché risulta essere molto influenzata dalla geografia dell'ambiente e dalla tipologia di materiali presenti: inoltre le onde radio passano attraverso i muri, e sono state preferite proprio per questo motivo, rendendo di fatto impossibile isolare la zona di copertura Wi-Fi.

Possiamo classificare le possibili minacce alla sicurezza in cinque gruppi:
- ascolto passivo del traffico di rete;
- uso non autorizzato delle risorse di rete, quali la connessione veloce a Internet;
- interferenza intenzionale per il danneggiamento delle funzionalità del sistema wireless;
- introduzione di spamming indesiderato sulla rete;
- accesso temporaneo da parte di utenti di passaggio (vampire connection).

Non tutte le minacce sono pericolose, anche se in qualche modo sono dannose poiché utilizzano impropriamente il sistema e quindi riducono l'efficienza delle risorse della rete; possiamo classificare due tipologie di "attacco":
- attacchi passivi: sono effettuati per "leggere" i dati semplicemente "ascoltando" quanto circola nei canali di comunicazione Wi-Fi;
- attacchi attivi: a loro volta possono essere di due tipi:
  – per immettere traffico: per accedere a tutti gli effetti all'utilizzo della rete;
  – per decrittare traffico: per ricevere informazioni inviando all'AP pacchetti "trappola".

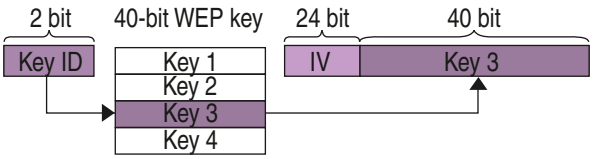
Dato che gli indirizzi MAC delle schede Wi-Fi sono unici, come tutti gli indirizzi delle schede Ethernet, è possibile pensare di inserire una tabella di indirizzi autorizzati all'interno degli AP in modo che questi possano accettare solo i pacchetti trasmessi da indirizzi conosciuti.
Questa soluzione non sempre è efficace in quanto per un malintenzionato è possibile "mascherare" da software l'indirizzo MAC camuffandolo e sostituendolo con indirizzi autorizzati.

Le specifiche 802.11 introducono il WEP (*Wired Equivalent Privacy*) per proteggere la rete e impedire che utenti non autorizzati ascoltino o immettano traffico sulla rete.
Si definisce WEP un frame così costituito:



- il WEP header contiene 24 bit chiamati di *Initialization Vector* (IV) e 2 bit di Key-ID: vengono utilizzati per la determinazione della chiave RC4 di 40 bit (WEP key) tra le quattro definite;
- i dati, criptati con la chiave RC4;
- un gruppo di 32 bit di parità (c(M)).

Questa tecnica può andare bene per respingere attacchi da parte di malintenzionati inesperti, ma dato che la chiave segreta del RC4 deve essere inserita in ogni stazione e non cambia praticamente mai, e il Key-ID è trasmesso in chiaro, è possibile superare questa protezione.



Anche i primi 24 bit della chiave sono trasmessi in chiaro, quindi non è difficile per un malintenzionato intercettare questi dati per ricostruire la chiave completa di 40 bit.

Con le velocità attuali dei personal computer è anche possibile "forzare" brutalmente i 40 bit provando tutte le combinazioni in tempi accettabili.

La prima contromisura è stata quella di passare a chiavi di 128 bit, che vengono utilizzate con meccanismi di VPN (*Virtual Private Networking*) e EAP (*Extensible Authentication Protocol*) che hanno sistemi di autenticazione più complessi attuati da appositi server.

◀ RC4 è un famoso algoritmo di cifratura a chiave simmetrica dove il testo viene cifrato con una chiave pseudocasuale mediante un'operazione di XOR. ▶

Per eliminare le debolezze del WEP, sia a 40 sia a 104 bit, si è anche introdotto il protocollo WPA (*Wi-Fi Protected Access*), che utilizza una variante dell'algoritmo WEP dal quale sono state rimosse le principali debolezze (effettua una cifratura a 128 bit mediante l'algoritmo ◀ RC4 ▶).
Ma anche questo protocollo si è dimostrato vulnerabile: alcuni ricercatori utilizzando un attacco con il metodo WPA-PSK hanno dimostrato di potere violare in 60 secondi una connessione WPA.
Si è passati al protocollo WPA2, ratificato nel 2008, che utilizza un diverso algoritmi di cifratura, l'AES (*Advanced Encryption Standard*), un algoritmo di cifratura a blocchi, che garantisce un'elevata sicurezza ma non è compatibile con le apparecchiature della generazione precedente.

La sicurezza informatica e delle reti è un argomento estremamente importante al quale verrà dato ampio spazio in una sezione dedicata all'interno di questo corso.

Zoom su...

VANTAGGI DELLA RETE WIRELESS

La realizzazione di una rete wireless, rispetto a una tradizionale LAN cablata, offre notevoli vantaggi:

- richiede un minor impiego di manodopera e tempi di realizzazione più contenuti;
- risulta essere la soluzione ideale per quelle realtà che operano in ambienti nei quali il cablaggio tradizionale è reso impossibile da barriere architettoniche (uffici storici, uffici ubicati in edifici diversi ecc.);
- permette di aggiungere sempre nuove postazioni di lavoro senza richiedere la posa di cavi aggiuntivi;
- computer portatili e desktop possono connettersi alla rete senza la necessità di un collegamento fisico: l'utente può scegliere se stare nel proprio ufficio o spostarsi in un'altra stanza senza perdere la connettività;
- è la soluzione ideale per implementazioni rapide come: eventi, fiere, manifestazioni;
- grazie alla tecnologia wireless è possibile effettuare collegamenti radio a brevi-medie distanze (fino a qualche decina di chilometri), realizzare connessioni punto-punto (tra due edifici) o connessioni punto-multipunto (tra più edifici).

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Le tecnologie più usate per realizzare reti senza fili sono:

- ○ raggi infrarossi
- ○ raggi ultravioletti
- ○ onde radio
- ○ onde ottiche

2 La propagazione delle onde senza fili è descritta nel:

- ○ protocollo 802.09
- ○ protocollo 802.10
- ○ protocollo 802.11
- ○ protocollo 802.12

3 L'acronimo SSID significa:

- ○ *Sistem Set Identifier*
- ○ *Sistem Service Identifier*
- ○ *Service Set Identifier*
- ○ *Service Set Identifier*

4 A che livello avviene l'autenticazione per l'accesso a una WLAN?

- ○ livello 1
- ○ livello 2
- ○ livello 3
- ○ livello di applicazione

5 Il WEP adotta una tecnica di:

- ○ crittografia a chiave multipla
- ○ crittografia a blocchi
- ○ crittografia con chiave RC4 di 40 bit
- ○ crittografia a 128 bit

6 WPA significa:

- ○ *Wireless Protected Area*
- ○ *Wireless Protected Access*
- ○ *Wi-Fi Protected Access*
- ○ *Wi-Fi Protected Area*

Test vero/falso

1	Al primo standard per le comunicazioni wireless creato dalla IEEE fu dato codice 802.11	V F
2	I dispositivi wireless comunicano tra loro a frequenze di 2.4 GHz.	V F
3	I dispositivi wireless comunicano tra loro con velocità dell'ordine dei Mbps.	V F
4	Nello scanning attivo il client si pone in ascolto di messaggi (bacon) trasmessi dall'AP.	V F
5	La DCF opera al di sopra della PCF e ne sfrutta le caratteristiche.	V F
6	Una modalità di trasmissione wireless è quella *ad hoc*, o EES.	V F
7	La connessione BSS (*Independent Basic Service Set*) è una connessione peer to peer.	V F
8	Nelle reti aziendali, in ufficio, nelle reti domestiche si utilizza la EES.	V F
9	WEP è l'acronimo di Wireless Equivalent Privacy.	V F
10	Nel WEP il key-ID viene trasmesso cifrato.	V F
11	Nel frame WEP è presente un gruppo di 32 bit di parità.	V F
12	Il WEP è un'evoluzione del WPA.	V F
13	Il protocollo WPA utilizza una cifratura a 128 bit.	V F
14	Il WPA2 utilizza per la cifratura l'algoritmo AES.	V F

Verifichiamo le competenze

Esprimi la tua creatività

Domande a risposta aperta

1 Le tecnologie più usate per realizzare reti senza fili sono .....

.....

2 I dispositivi in una rete wireless comunicano tra loro a .....

.....

3 L'evoluzione dello standard 802.11 è .....

.....

4 Quando un dispositivo vuole connettersi a una WLAN esegue uno scanning che può essere:

1) .....

2) .....

5 L'accesso a un canale, per poter comunicare con tecnologia wireless, può essere fatto:

1) .....

2) .....

6 La comunicazione in una WLAN non utilizza i frame standard 802.3 ma .....

.....

7 Il protocollo WPA (*Wi-Fi Protected Access*) è stato introdotto per .....

.....

Esercizi di completamento

raggi infrarossi • onde radio • Access Point • SSID • Wi-Fi • PFC • WEP • RC4 • Initialization Vector • AES

1 La tecnologia a ..... ha lo svantaggio di essere estremamente sensibile agli ostacoli.

2 Il protocollo 802.11 prevede infatti che la propagazione dei segnali avvenga utilizzando .....

3 Le funzioni di concentratore e di "collegatore" tra la rete wireless e la LAN cablata vengono fatte mediante un dispositivo che prende il nome di .....

4 L' ..... è il nome con cui una rete ..... si identifica verso i suoi utenti.

5 Per effettuare connessioni strutturate l'accesso a un canale si ottiene con il .....

6 Per accessi a rete privata o LAN aziendali l'AP con chiave condivisa usa la crittografia .....

7 Il WEP per proteggere la rete utilizza un codice cifrato mediante chiave .....

8 Il WEP header contiene 24 bit chiamati di .....

9 Il protocollo WPA2 utilizza come algoritmo di cifrature .....

UNITÀ DIDATTICA 5

IL CABLAGGIO STRUTTURATO DEGLI EDIFICI

IN QUESTA UNITÀ IMPAREREMO...

- la terminologia dei componenti dei cablaggi strutturati
- la normativa americana standard EIA/TIA-568
- la normativa europea ISO/IEC DIS 11801

Generalità

La comunicazione tra due dispositivi attivi avviene mediante una connessione che il modello OSI definisce essere a livello 1, o strato fisico, e che ha come compito proprio quello di effettuare il trasferimento delle cifre binarie tra i due sistemi che devono comunicare.

Precise norme stabiliscono e regolamentano le modalità con cui viene realizzata l'interconnessione fisica di due dispositivi, o nodi, nella rete.

Vengono specificate le caratteristiche:
- meccaniche: la forma e la tipologia di prese e spine, il numero di contatti;
- elettriche: il voltaggio e le caratteristiche elettriche dei segnali associati all'interfaccia;
- funzionali: il significato dei vari segnali;
- procedurali: la combinazione e le sequenze dei segnali necessarie per il corretto funzionamento dei dispositivi.

La realizzazione degli impianti che permettono la comunicazione dei dati prende il nome di cablaggio.

CABLAGGIO

Per cablaggio si intende l'insieme di componenti passivi come cavi, prese, connettori, permutatori ecc., installati e predisposti per poter interconnettere i componenti attivi dei sistemi di elaborazione.

Il continuo aumento di richiesta di comunicazione tra dispositivi presenti anche in ambiente domestico ha portato alla standardizzazione della realizzazione degli edifici di nuova costruzione con predisposizioni (canalizzazioni) specifiche che ne permettono il cablaggio strutturato.

L'elevato numero degli apparecchi e le diverse funzionalità dei sistemi richiedono la posa di grandi quantità di cavi in modo capillare in ogni parte dell'edificio e ogni diverso sistema richiede un impianto distinto (quindi un impianto per la telefonia, un impianto per la LAN, un impianto per la videosorveglianza ecc.).

Anche negli appartamenti residenziali si richiedono sempre più spesso infrastrutture simili a quelle degli uffici: la ◀ **domotica** ▶, i sistemi di sicurezza, il videocontrollo, le videoconferenze, la telefonia, la televisione e i collegamenti tra computer necessitano di una connessione permanente, realizzata mediante un sistema specifico di cablaggio.

◀ La domotica, detta anche home automation dalla definizione inglese, è la disciplina che si occupa di studiare le tecnologie atte a migliorare la qualità della vita nella casa mediante l'automazione dei processi e l'utilizzo delle tecnologie informatiche. ▶

Le normative sui sistemi di cablaggio definiscono i metodi per cablare sia singoli edifici industriali e/o gruppi di edifici costruiti su un comprensorio (campus), sia per cablare un singolo appezzamento di suolo privato: prevedono inoltre le regole per cablare edifici su campus diversi anche non direttamente a contatto ma separati da strade o da altri ostacoli permanenti.

In sintesi, nelle normative di cablaggio sono presenti le descrizioni:
- delle caratteristiche dei mezzi trasmissivi e dei componenti passivi;
- delle geometrie delle topologie di cablaggio (stella, anello, *bus*, maglia), dove vengono indicate le distanze massime, le modalità di interconnessione, le modalità di definizione degli eventuali livelli di gerarchia;
- delle regole di installazione, di test e di collaudo, e le modalità di come documentare il progetto in ogni sua parte.

I sistemi di cablaggio possono essere di tipo proprietario, per esempio il Cabling System IBM o il DECconnect Digital, oppure standard internazionali, che attualmente sono di due tipi:
- lo standard americano (ANSI) definito dall'◀ EIA/TIA ▶;
- lo standard internazionale definito dall'ISO.

◀ EIA, TIA e ISO sono gli acronimi di tre organizzazioni, due americane e una europea:
- EIA: *Electronic Industries Alliance*;
- TIA: *Telecommunications Industry Association*;
- ISO: *International Organization for Standardization*. ▶

Standard internazionali

In ambiente industriale l'esigenza di disporre di sistemi di cablaggio è nata con la diffusione agli inizi degli anni '80 dell'informatica in azienda e la nascita di sistemi informativi che richiedevano in un primo tempo la connettività dei terminali operativi e in seguito dei personal computer da affiancarsi ai normali sistemi di telecomunicazione: i primi sistemi realizzati cercavano di creare una sinergia proprio con i sistemi di cablaggio per telefonia e nacquero i cablaggi "fonia-dati".

I primi standard di cablaggio sono nati negli Stati Uniti dalla collaborazione tra due comitati: l'EIA (associazione delle industrie elettroniche) e la TIA (associazione delle industrie di telecomunicazioni):
- EIA/TIA-568: è uno standard americano definito nel 1991 per il cablaggio di edifici commerciali ed è tra gli standard più applicati in tutto il mondo;
- EIA/TIA-570: è uno standard americano per il cablaggio di edifici prevalentemente residenziali ma anche con la presenza di qualche ufficio commerciale.

Con il passare degli anni questi standard vennero modificati e integrati per adeguarli all'evoluzione tecnologica. Vennero quindi definiti:

- ISO/IEC DIS 11801: è uno standard internazionale approvato nel 1994 per i cablaggi di edifici commerciali divenuto in Europa un requisito base per la realizzazione di cablaggi strutturati (IEC significa *International Electrotechnical Commission* e DIS significa *Draft International Standard*);
- SP-2840-A: consiste in un adeguamento dello standard EIA/TIA-568 per includere le nuove tecnologie con maggiori velocità di trasmissione;
- prEN 50173 Final Draft (European Norms emesse dal Comitato Tecnico TC 115 CENELEC): riprende e fa propria a livello CEE/UE la proposta ISO/IEC DIS 11801.

Il CENELEC è l'organismo di coordinamento dei Paesi membri dell'Unione Europea, che ha come scopo principale quello di far adottare le norme IEC e di preparare bozze di norme.

Zoom su...

EIA/TIA A CONFRONTO CON ISO/IEC DIS 11801

Le differenze sostanziali della proposta europea rispetto allo standard americano EIA/TIA sono:

- gli elementi costituenti il cablaggio hanno una nomenclatura leggermente diversa;
- è stato introdotto il concetto di "classi di lavoro" utilizzato per definire i requisiti minimi di un collegamento;
- è stata ampliata la gamma dei tipi di cavo che possono essere utilizzati, sia a livello di rame sia di fibra ottica, fornendo specifiche più complete e dettagliate e abolendo l'uso di cavi coassiali;
- sono stati introdotti test più rigorosi sui cavi in rame: ogni cablaggio, una volta realizzato, deve essere testato e validato con un insieme di prove effettuate con idonei strumenti di misura, e i risultati delle prove devono essere riportati sulla certificazione;
- dato che viene introdotto l'utilizzo di doppini schermati vengono definiti e meglio dettagliati gli aspetti della messa a terra (le specifiche sono descritte nello standard TIA/EIA 607).

È necessario sottolineare che la realizzazione di un cablaggio strutturato richiede interventi sulle strutture edili e meccaniche per la realizzazione di infrastrutture permanenti descritte nello standard americano EIA/TIA-569 che però non sempre sono facilmente realizzabili.

Esistono sostanzialmente due momenti in cui viene realizzato un cablaggio strutturato:

- alla costruzione dell'edificio o nella fase di ristrutturazione;
- in edifici/locali preesistenti per esigenze sopraggiunte in un secondo tempo o come adeguamento alle normative internazionali.

Naturalmente nel primo caso esiste una completa libertà di intervento e la realizzazione del cablaggio strutturato risulta essere operativamente abbastanza semplice: al contrario, nel secondo caso spesso è difficoltoso se non impossibile realizzare opere murarie che richiederebbero l'interruzione delle attività esistenti e quindi si deve ripiegare su alternative che spesso rischiano di non aderire completamente alle normative di riferimento e in generale ci si limita a interventi per rendere a norma la telefonia e la trasmissione dati.

Il cablaggio secondo lo standard EIA/TIA-568

La prima normativa approvata per il cablaggio strutturato è stata definita negli Stati Uniti nel 1991; resta tutt'oggi una base di riferimento per le normative europee più moderne e definisce i requisiti minimi richiesti per il cablaggio di un edificio o un gruppo di edifici facenti parte di uno stesso ◀ comprensorio ▶.

◀ Con comprensorio si intende una regione geografica con estensione massima di 3000 m e una superficie calpestabile per gli edifici di 1.000.000 m^2; viene anche indicato in 50.000 il numero massimo persone che ne possono fruire. ▶

Nella fase di progetto di un cablaggio strutturato è opportuno tener presente che la validità dello stesso deve essere almeno di dieci anni: quindi oltre alle esigenze specifiche del momento è necessario "prevedere" anche quelle che potrebbero sopraggiungere negli anni futuri, per poterle poi realizzare all'occorrenza con minimi interventi aggiuntivi.

Dieci anni sono da considerarsi un'"eternità" per l'informatica, in considerazione dei tempi dell'evoluzione tecnologica previsti dalla legge di Moore: è quindi necessario organizzare già nelle prime fasi del progetto un gruppo di studio formato dai rappresentanti delle differenti parti in causa in modo da definire le soluzioni nel rispetto della normativa con i seguenti obiettivi:

- miglior rapporto tra costi e benefici;
- massima flessibilità di aggiornamento tecnologico;
- tecnologie trasmissive aggiornate;
- offrire la migliore flessibilità di utilizzo;
- soddisfare tutti i servizi richiesti dall'utenza;
- garantire una realizzazione nel rispetto ambientale.

Zoom su...

LEGGE DI MOORE

Gordon Moore, uno dei fondatori di Intel, nel 1965 azzardò una previsione sullo sviluppo futuro dell'elettronica che si è poi mostrata corretta per i quaranta anni seguenti e ancora oltre: la formulazione originaria riguardava la densità di componenti nei circuiti integrati, ma la deduzione che da essa ne deriva e che oggi ricordiamo appunto come *legge di Moore* è la seguente: "*la potenza di calcolo dei computer raddoppia grosso modo ogni 18 mesi*".

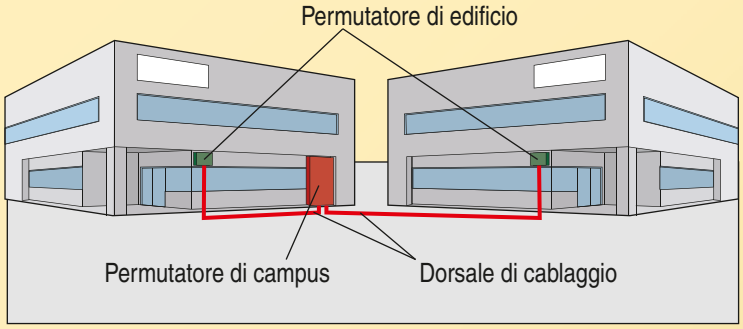
Lo standard EIA/TIA-568 prevede che il cablaggio venga realizzato contestualmente alla costruzione o ristrutturazione organica di un edificio, dove sono previste opere murarie sia verticali (interventi nelle pareti) che orizzontali (interventi nella pavimentazione).
È comunque anche applicabile nel caso di semplice rinnovamento della rete locale o di fonia dove però sussistano le necessarie infrastrutture sulle quali "appoggiare" i nuovi cavi.

Le specifiche dello standard riguardano:

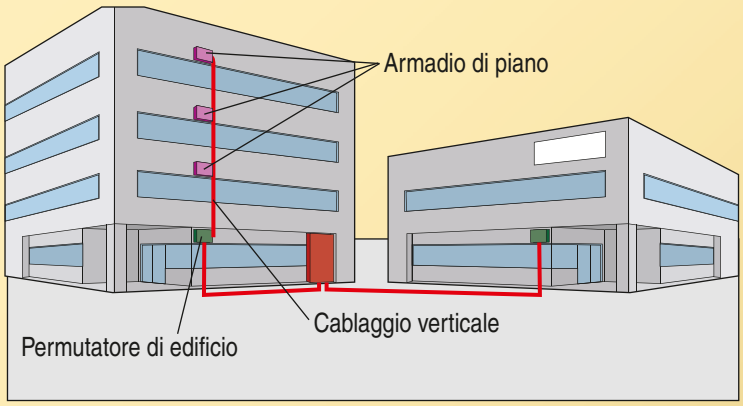
- la topologia: la topologia del cablaggio è di tipo stellare gerarchica, dove si possono individuare tre livelli nella gerarchia:
  - centro stella di comprensorio o *Main Crossconnect* MC =>unico per il comprensorio;
  - centro stella di edificio o *Intermediate Crossconnect* IC =>uno per ogni edificio;
  - armadio di piano o *Telecommunication Closet* TC => uno per ogni piano;

La gerarchia descritta assegna anche a ogni livello un nome per il cablaggio:

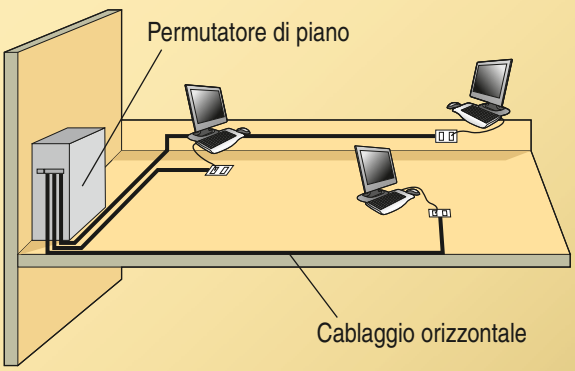
- 1° livello = cablaggio di campus: interconnette tra loro diversi edifici; ▶



- 2° livello = cablaggio verticale: interconnette tra loro i piani di un singolo edificio; ▶

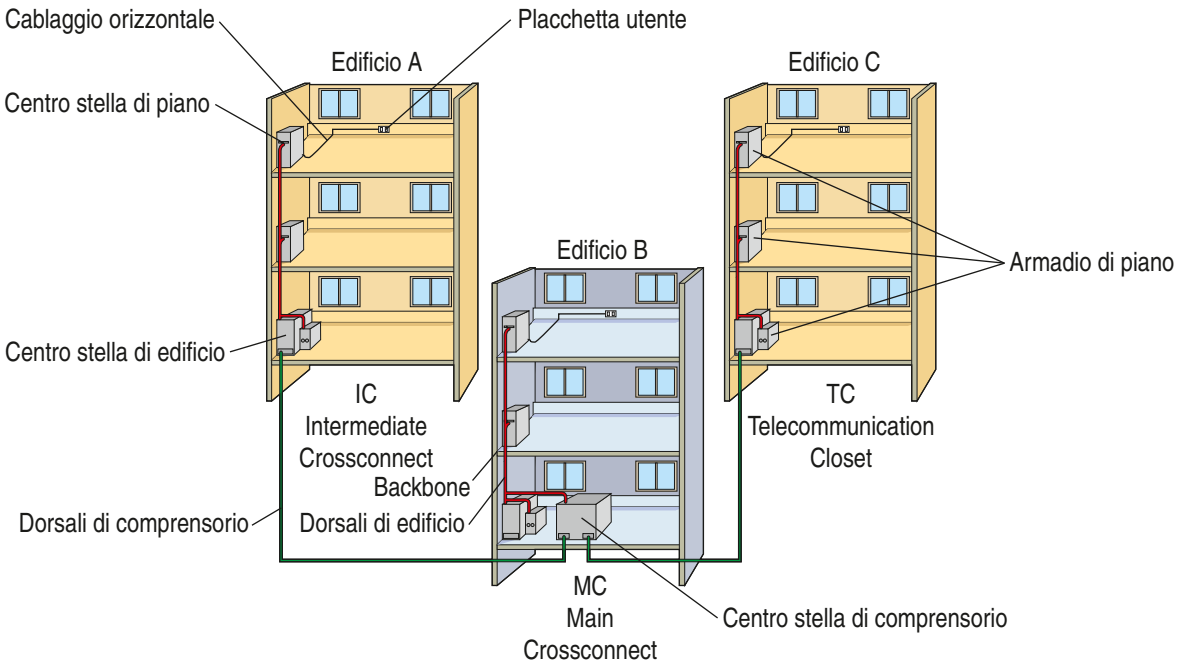


- 3° livello = cablaggio orizzontale: è riferito al cablaggi dei locali di un singolo piano. ▶



- gli elementi facenti parte del cablaggio:
  - il *Main Crossconnect* MC: è il permutatore principale situato nell'edificio centrale di un comprensorio da cui vengono distribuiti i cavi di dorsale agli altri edifici (centro stella di primo livello);
  - l'*Intermediate Crossconnect* IC: è il locale o l'armadio di distribuzione presente in ogni edificio del comprensorio dal quale vengono distribuiti i cavi di dorsale di edificio ai vari piani (centro stella di secondo livello);
  - i *Telecommunication Closet* TC: a ogni piano di ogni edificio è presente un armadio dal quale vengono distribuiti i cavi che raggiungono l'utenza (centro stella di terzo livello): a volte vi è dedicato un apposito locale contenente gli armadi e questo prende il nome di *Telecommunication Room* TR).
  - l'*interbuilding backbone* o dorsale di comprensorio: è la dorsale di interconnessione tra l'edificio centro stella di comprensorio e un altro edificio che mette in comunicazione l'MC con ogni IC dei singoli edifici.
  - l'*intrabuilding backbone* o dorsale di edificio: è la dorsale di interconnessione tra il locale tecnologico di edificio IC e l'armadio di piano TC.

Possiamo rappresentare con un disegno gli elementi fino ad ora descritti che riguardano sia il cablaggio di campus che il cablaggio verticale:



Per le dorsali (backbone) devono essere impiegati cavi in fibra ottica (generalmente di tipo loose, idoneo all'installazione sia in esterno che in interno) con protezione antiroditore, che devono essere posati nelle tubazioni e canalizzazioni di distribuzione dedicate, fino all'armadio di piano TC: i cavi devono essere tutti dotati di connettori e attestati ai rispettivi pannelli di permutazione.
Per esempio, un tipo di fibra utilizzata per i backbone ha le caratteristiche riportate nella seguente tabella:

Tipo cavo	Fibra ottica multimodale 62,5 / 125 μm
Numero fibre	≥12
Caratteristiche costruttive	*loose tube* con gel idrorepellente, protezione antiroditore completamente dielettrica
Caratteristiche guaina esterna	non propagante l'incendio, a basso contenuto di gas alogeni, secondo la normativa CEI 20-22 e CEI 20-37
Attenuazione massima per ciascuna fibra	3,5 dB/km a 850 nm e 1 dB/km a 1300 nm
Banda passante	200 MHz a 850 nm e 500 MHz a 1300 nm per ciascuna fibra

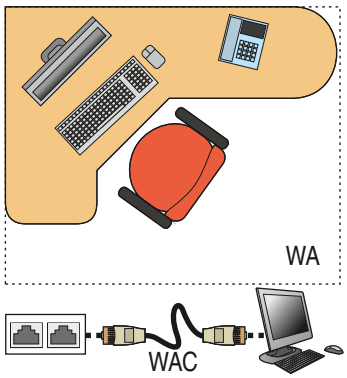
Durante la posa dei cavi si deve avere la massima cura di non superare sia la tensione di tiro che il raggio di curvatura minimo, onde evitare il degradamento delle loro caratteristiche tecniche.

Quando si è completata la realizzazione degli impianti verticali si procede con il di cablaggio orizzontale (o di piano), secondo quanto previsto dalla normativa.

Il cablaggio orizzontale contiene la più grossa quantità di cavi da stendere per l'intero sistema e deve essere progettato con particolare cura. I cavi di interconnessione sono alloggiati in canaline

predisposte in maniera da escludere raccordi di curvatura o percorsi con curvatura tanto spinta da rendere disagevole il passaggio dei cavi sia in fase di installazione sia per i successivi interventi di manutenzione e di ampliamento.

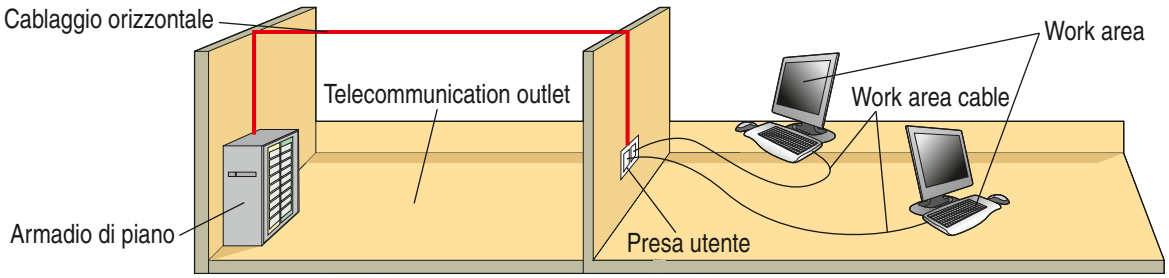
La norma inizia con il descrivere il posto di lavoro, o *Work Area* (WA) che, come possiamo vedere dalla figura, deve prevedere due prese, rispettivamente una per i dati e una per le telecomunicazioni: il cavo di collegamento alla presa a muro, la *Telecommunication Outlet* (TO) prende il nome di *Work Area Cable* (WAC) che mediante il cablaggio orizzontale lo connette con l'armadio di piano TC posto possibilmente in un apposito locale. ▶



La presa a muro *Telecommunication Outlet* è generalmente composta da due o tre connettori, rispettivamente di tipo:

- RJ45 per cavi a quattro coppie UTP;
- Ermafrodita 802.5 per cavi a due coppie STP;
- SC per fibra ottica.

Dal connettore a muro i cavi raggiungono l'armadio di piano con collegamenti orizzontali di lunghezza non superiore ai 90 m, indipendentemente dal mezzo impiegato: questa distanza deve essere "coperta" con un filo senza giunture dall'armadio di piano (centro stella) alla presa a muro (da IC a TO). La distanza tra la presa a muro e la postazione di lavoro non deve inoltre superare i 3 m.



La rete di distribuzione orizzontale deve presentare caratteristiche uniformi di cui gli unici parametri specifici sono il numero e la dislocazione delle prese utente, utilizzando cavi UTP di categoria 5e per dati e fonia fino a 100 MHz.

A titolo di esempio riportiamo le caratteristiche di un cavo UTP Cat 5e nella tabella che segue.

Tipo di cavo	UTP 4×2 AWG24 Cat 5e
Impedenza	100 Ohm ± 15
Attenuazione massima ammessa	22 db/100 m a 100 MHz
NEXT (minimo valore ammesso)	32 db/100 m a 100 MHz
Rivestimento	guaina non propagante l'incendio e a basso contenuto di gas alogeni
Rispondenza norme	EIA/TIA-568A, ISO/IEC 11801

I cavi devono essere posati nelle tubazioni e nelle canalizzazioni di distribuzione dedicate che verranno implementate all'interno dei locali a partire dai locali tecnici (dal centro stella di piano) su permutatori per rame e per fibra.

PERMUTATORE

Con il termine permutatore (o patch panel) si intende un dispositivo che opera commutazioni, conversioni o collegamenti in una rete di trasferimento di segnali elettrici, siano essi dati che telecomunicazioni.

Per esempio, un pannello di permutazione fibra/rame contiene un convertitore che trasforma la luce in un segnale elettrico.

A seconda del tipo di cavi, il patch panel può assumere due forme:
- per cavi in rame può contenere uno o più blocchi di terminazione;
- per fibre ottiche può contenere una serie di connettori passanti, chiamati barrel o bussole, che servono a permutare le fibre tra pannelli diversi oppure tra un pannello e un apparato attivo.

Il cavetto di fili di rame che viene utilizzato per realizzare le connessioni al suo interno viene chiamato patch cord: in caso di fibre ottiche prende il nome di bretella ottica.

Le figure seguenti riportano un telaio a 24 porte per cavi Cat. 6, un patch panel a 24 porte per fibra ottica e un patch panel completo di connessioni e due patch cord.

I permutatori generalmente sono alloggiati all'interno dell'armadio rack (per esempio rack 1 9"), anche insieme a elementi attivi (alimentatori, switch, modem ecc.). ▶
All'interno dell'armadio i cavi vengono numerati e fascettati (oppure posti negli appositi passacavi da rack) e legati ai montanti del rack (è necessario prestare attenzione, in fase di raggruppamento dei cavi, a non fascettarli in modo stretto, per non incorrere nelle problematiche di degradamento.

Riassumiamo in una tabella le specifiche che la norma definisce per i mezzi trasmissivi:

fibre ottiche	multimodali di dimensioni 62,5/125 μm
cavi UTP a 4 coppie	dimensione 24 AWG
cavi UTP multicoppia	possono essere di 24 AWG oppure di 22 AWG e contengono uno o più gruppi da 25 coppie, ciascuno protetto con un guaina isolante
cavi STP	previsti dalle specifiche "tipo 1" IBM, da 150 Ω
cavi coassiali da 50 Ω	devono soddisfare lo standard IEEE 802.3, 10Base5 e 10Base2

Lo standard ISO/IEC DIS 11801

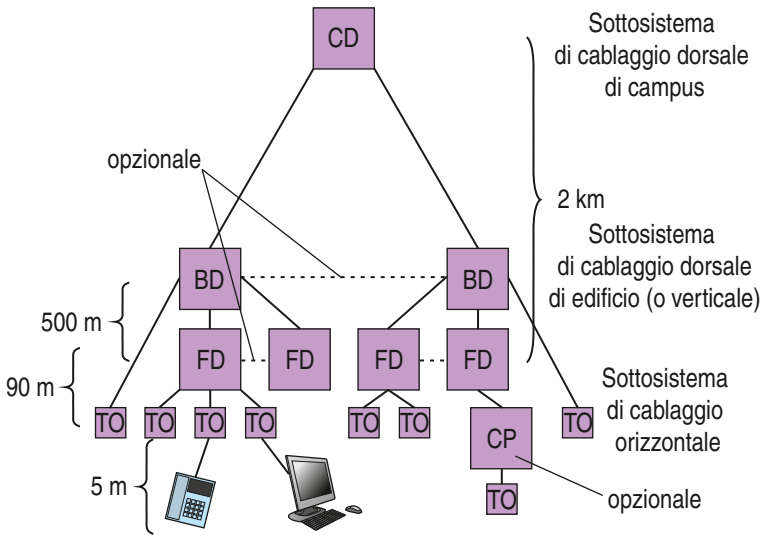
L'ISO/IEC DIS 11801 è una proposta di standard internazionale per i cablaggi, approvata nel 1995 e molto simile allo standard americano EIA/TIA-568, dal quale si differenza per le seguenti particolarità:

- ha una nomenclatura leggermente diversa:
  - *Campus Distributor* CD => centro stella primo livello;
  - *Building Distributor* BD => centro stella secondo livello;
  - *Floor Distributor* FD => centro stella terzo livello;
- introduce un ulteriore livello nella gerarchia, collocato tra il BD e l'FD:
  - *Consolidation Point* CP (o *Transition Point* TP). Si tratta di un punto di amministrazione interposto tra il distributore di piano e la presa telematica, da utilizzarsi soprattutto in ambienti open space dove si richiede una maggiore flessibilità nella rilocazione delle prese.

La figura seguente riporta la topologia completa della rete e le massime distanze ammesse. ▼

Il CP è di solito realizzato con componenti passivi in un patch panel di piccole dimensioni e serve al massimo dodici aree di lavoro.

L'introduzione del Consolidation Point implica alcuni accorgimenti sulle distanze di cui tener conto, in quanto la sua distanza minima dal distributore deve essere di 15 metri per ridurre l'effetto delle connessioni multiple in spazi ristretti sul NEXT.



- fornisce un maggior numero di dati sulle caratteristiche dei mezzi trasmissivi;
- permette l'utilizzo di un maggior numero di tipi di doppini e fibre ottiche, ma non ammette l'uso di cavi coassiali;

Cavi ammessi per le dorsali:

fibre ottiche	sia monomodale che multimodale di dimensioni 62.5/125 μm
cavi UTP	cavi di tipo bilanciato (doppino) da 100 Ω (preferito), 120 Ω o 150 Ω che possono essere di tipo schermato o non schermato e possono essere composti da 2 o più coppie, ciascuno protetto con un guaina a bassa emissione di gas zero-alogeni

Cavi ammessi per la distribuzione orizzontale:

fibre ottiche	sia monomodale che multimodale di dimensioni 62.5/125 μm
cavi UTP	cavi di tipo bilanciato (doppino) da 100 Ω (preferito), 120 Ω o 150 Ω che possono essere di tipo schermato o non schermato e possono essere composti da 2 o più coppie
cavi ibridi	ovvero cavi composti da elementi di diverso tipo o categoria, per esempio: 4 coppie UTP da 100 Ω di Cat. 5 e due fibre ottiche
cavi coassiali	non più ammessi

Per il cablaggio orizzontale devono essere previsti almeno due cavi per ogni posto di lavoro, che partono dall'armadio di piano e terminano nella presa a muro:
- il primo cavo deve essere di categoria 3 o superiore;
- il secondo cavo deve essere di categoria 5 o in alternativa può essere una fibra ottica multimodale.

La presa a muro o placchetta utente deve avere delle targhette permanenti, visibili esternamente dall'utente, che servono per identificare i cavi.

- introduce test più rigorosi per controllare le categorie dei cavi in rame: le misure di collaudo andrebbero eseguite a temperatura ambiente (definito in 20 °C) ed esistono delle indicazioni circa gli aggiustamenti da effettuare nel caso di sensibili differenze da tale valore;
- tratta in modo leggermente più approfondito gli aspetti della messa a terra;
- non si occupa di aspetti relativi alla documentazione.

Nella seguente tabella sono riportate le sigle relative alla terminologia standard utilizzate nelle due diverse normative, in modo da favorirne il confronto.

Elemento del cablaggio	Norma CEI EN 50173	Standard TIA/EIA 568-B
Distributore di comprensorio	CD *Campus Distributor*	MD *Main Cross Connect*
Distributore di edificio	BD *Building Distributor*	IC *Intermediate Distributor*
Distributore di piano	FD *Floor Distributor*	HC *Horizontal Cross Connect*
Presa di telecomunicazione	TO *Telecommunication Outlet*	TO
Punto di transizione	TP *Transition Point*	TP
Punto di consolidamento	Punto di consolidamento	CP *Consolidation Point*
Dorsale di comprensorio	CB *Campus Backbone*	*Interbuilding backbone*
Dorsale di edificio	BB *Building Backbone*	*Intrabuilding backcone*
Sala apparati	ER *Equipment Room*	ER
Armadio di piano	TC *Telecommunication Closet*	TC
Punto di ingresso dei servizi	EF *Entrance Facility*	EF
Cordoni di collegamento per i dispositivi	EC *Equipment Cord*	EC
Pannelli di distribuzione	PP *Patch Panel*	PP
Cordoni relativi ai pannelli di distribuzione	PC *Patch Cords*	PC

La tabella che segue riporta a confronto le caratteristiche dei cavi utilizzati nelle due normative descritte e nel documento CENELEC EN50173: CENELEC è European Committee for Electrotechnical Standardization ed è l'organismo responsabile delle standardizzazione europea in campo elettrico.

Standard	EIA/TIA 658 A	ISO 11801	CENELEC EN 50173
Cavi in rame per dorsali o cablaggi orizzontali	Cavo UTP, 4 coppie, 100, guaina in PVC	Cavo UTP, 4 coppie, 100 o 200, guaina a bassa emissione di gas zero-alogeni; cavo FTP o STP opzionale	Cavo UTP, 4 coppie, 100-200, guaina a bassa emissione di gas zero-alogeni; cavo STP opzionale
Raggio di curvatura del cavo posato in dorsale	> 10 * (diametro esterno del cavo)	> 6 * (diametro esterno del cavo)	> 6 * (diametro esterno del cavo)
Raggio di curvatura del cavo posato in orizzontale	> 4 * (diametro esterno del cavo)	> 4 * (diametro esterno del cavo)	> 4 * (diametro esterno del cavo)

Standard	EIA/TIA 658 A	ISO 11801	CENELEC EN 50173
Prestazioni	Categoria 3 fino a 16 MHz	Classe C fino a 16 MHz	Classe C fino a 16 MHz
	Categoria 4 fino a 20 MHz	Classe D fino a 100 MHz	Classe D fino a 100 MHz
	Categoria 5 fino a 100 MHz		
Return Loss (perdita per riflessione)	–	10 dB a 100 MHz	10 dB a 100 MHz
Attenuazione massima	23,2 dB a 100 MHz	23,6 dB a 100 MHz	23,6 dB a 100 MHz
ACR minimo	–	4 dB a 100 MHz	4 dB a 100 MHz

Sviluppi tecnologici e normativi

Il notevole progresso tecnologico e l'aumento del numero di dispositivi che devono scambiarsi informazioni hanno fatto sì che le caratteristiche tecniche dei vari componenti di cablaggio siano state spinte al loro limite prestazionale.

Il mercato offre oggi soluzioni di prodotti sia in rame sia in fibra ottica che sono in grado di realizzare collegamenti end-to-end dalle prestazioni impensabili fino a pochi anni fa: recentemente è stato ratificato da parte dell'IEEE (*Institute of Electrical and Electronic Engineers*), lo standard Ethernet per la trasmissione dati a velocità pari a 10 Gbit/s su cavo in rame di tipo UTP con distanze fino a 100 metri (standard IEEE 802.3an 10GBASE-T) che richiede caratteristiche trasmissive del canale molto particolari per tenere sotto controllo i fenomeni di alien crosstalk che si manifestano ad alte frequenze (si richiede un cablaggio di tipo completamente schermato sia per quanto concerne il cavo sia per gli altri componenti che costituiscono il canale).
Per queste velocità di trasmissione è anche necessario sostituire il connettore standard RJ45 con un connettore specifico che riduca le dispersioni e le riflessioni.

Nuovi standard si stanno diffondendo anche per i componenti in fibra ottica, grazie all'elevato livello tecnologico raggiunto: lo standard ISO/IEC 11801 2nd Edition e successivamente quello EN 50173 2nd Edition hanno riconosciuto la necessità di prendere in esame applicazioni di velocità superiore al gigabit, definendo le seguenti classi:

- OF 300 per applicazioni su collegamenti in f.o. fino a 300 m;
- OF 500 per applicazioni su collegamenti in f.o. fino a 500 m;
- OF 2000 per applicazioni su collegamenti in f.o. fino a 2000 m;

indicando per ognuno di essi valori di attenuazione ottica da non superare a seconda del tipo di fibra impiegata, multimodale o monomodale, e della lunghezza d'onda di trasmissione (finestra ottica).

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Il mezzo trasmissivo attualmente più diffuso sulle LAN è:

- O il cavo coassiale grosso
- O il cavo coassiale sottile
- O la coppia simmetrica
- O la fibra ottica

2 La fibra ottica nelle LAN è soprattutto utilizzata:

- O sulle dorsali per l'esigenza di una banda passante più elevata
- O sulle dorsali per consentire connessioni tra apparati a distanza superiore a 100 m
- O ovunque siano necessarie interfacce con velocità superiore a 10 Mbps
- O solo se l'ambiente è rumoroso dal punto di vista elettromagnetico

3 Nella maggior parte dei casi, il cablaggio di una LAN oggi è costituito da:

- O una rete dorsale e una di distribuzione, entrambe a stella, con armadi di permutazione intermedi per dare flessibilità di connessione agli apparati di rete (HUB e switch) e ai terminali
- O una rete dorsale a bus e una rete di distribuzione ad anello interconnesse da ripetitori Ethernet/FDDI
- O una rete dorsale a stella e una di distribuzione a bus, interconnnesse da switch
- O una rete dorsale in cavo coassiale grosso, per garantire maggiore affidabilità, una rete di distribuzione in cavo coassiale sottile, per ridurre i costi e aumentare la flessibilità di interconnessione

4 La topologia del cablaggio EIA/TIA-568 è di tipo stellare gerarchica su tre livelli:

A centro stella di comprensorio .....
B centro stella di edificio .....
C armadio di piano .....

1 Intermediate Crossconnect
2 Telecommunication Closet
3 Main Crossconnect

5 Indica, nella tabella seguente, le corrispondenze tra EIA/TIA-568 e ISO/IEC DIS 11801

EIA/TIA 568	ISO/IEC DIS 11801
centro stella primo livello	
centro stella secondo livello	
centro stella terzo livello	

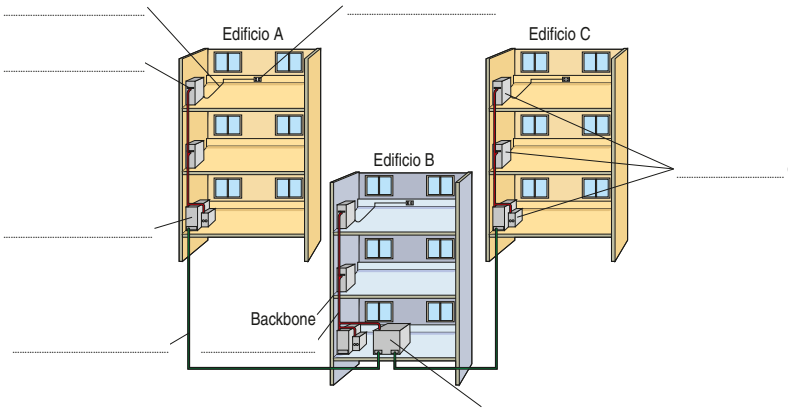
Test vero/falso

1	Lo standard internazionale per il cablaggio è definito dalla EIA/TIA.	V F
2	Lo standard europeo per il cablaggio è definito dall'ISO.	V F
3	Il *Main Crossconnect* (MC) è situato nell'edificio centrale di un comprensorio.	V F
4	L'*Intermediate Crossconnect* (IC) è il locale o l'armadio di distribuzione presente in ogni edificio.	V F
5	I *Telecommunication Closet* (TC) sono presenti a ogni piano.	V F
6	L'*Intermediate Crossconnect* (IC) coincide con il centro stella di primo livello.	V F
7	I *Telecommunication Closet* (TC) coincidono con il centro stella di secondo livello.	V F
8	*Telecommunication Room* (TR) è un locale contenente gli armadi di piano.	V F
9	L'interbuilding backbone interconnette gli edifici nel comprensorio.	V F
10	L'intrabuilding backbone è anche chiamata dorsale di campus.	V F
11	L'intrabuilding backbone connette il locale tecnologico di edificio (IC) e l'armadio di piano (TC).	V F
12	Il patch cord viene utilizzato per realizzare le connessioni all'interno del patch panel.	V F
13	Lo standard ISO/IEC DIS 11801 prevede il CP come livello intermedio tra BD e TO.	V F
14	Per i cavi UTP a 4 coppie la dimensione richiesta è 24 AWG.	V F

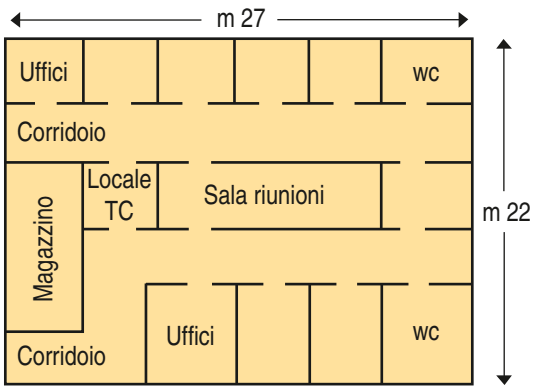
Verifichiamo le competenze

Esprimi la tua creatività

1 Indica sullo schema della figura che segue la terminologia di ogni elemento secondo la normativa EIA/TIA-568.



2 Data la planimetria di un piano di un edificio, progetta il cablaggio orizzontale effettuando una stima di massima dei costi dei materiali utilizzati.



Domande a risposta aperta

1 Qual è stato il primo standard americano in materia di cablaggio? .....

2 Quali sono le principali differenze tra EIA/TIA e ISO/IEC DIS 11801? .....

3 Quali sono i tre livelli della gerarchia previsti dallo standard EIA/TIA-568? .....

4 La norma descrive il posto di lavoro, o *Work Area* (WA) che deve vedere .....

5 In che cosa consiste, dove si trova e a che cosa serve il patch panel? .....

6 Quali sono i quattro livelli della gerarchia previsti dallo standard ISO/IEC DIS 11801? .....

7 Che cos'è il CENELEC? .....

Esercizi di completamento

dorsale di edificio • DECconnect Digital • bretella ottica • canalizzazioni • patch cord • intrabuilding backbone • proprietario • EIA/TIA-568 • cablaggio • dorsale di comprensorio • Cabling System IBM • patch panel • stellare gerarchica • interbuilding backbone • IC • TC

1 La realizzazione degli impianti che permettono la comunicazione dei dati prende il nome di ..... .

2 Negli edifici di nuova costruzione vengono realizzate predisposizioni specifiche per il cablaggio strutturato che denominate .....

3 I sistemi di cablaggio possono essere di tipo ....., per esempio il ..... o il .....

4 Lo standard ..... prevede che il cablaggio abbia una topologia .....

5 L'..... o .....: è la dorsale di interconnessione tra l'edificio centro stella di comprensorio e un altro edificio.

6 L'..... o .....: è la dorsale di interconnessione tra il locale tecnologico di edificio ..... e l'armadio di piano .....

7 All'interno del ..... vengono utilizzati per i collegamenti ..... e .....

Domande a risposta aperta

1 Che cosa si intende per "cablaggio strutturato"? .....

2 Quali sono gli standard internazionali per il cablaggio? .....

3 Che cos'è il CENELEC? .....

4 Che cosa si intende per "cablaggio comprensorio"? .....

5 Quale tipologia prevede lo standard EIA/TIA-568? .....

6 Quali sono gli elementi facenti parti il cablaggio nello standard EIA/TIA-568? .....

7 Che cosa viene utilizzato per le dorsali nello standard EIA/TIA-568? .....

8 Descrivi la work-area come indicata nello standard EIA/TIA-568. .....

9 Che cos'è e a che cosa serve un permutatore? .....

10 Quali sono le differenze introdotte nello standard ISO/IEC DIS 11801? .....

11 Quali cavi sono ammessi per le dorsali nello standard ISO/IEC DIS 11801? .....

12 Quali cavi sono ammessi per la distribuzione orizzontale nello standard ISO/IEC DIS 11801? .....

ESERCITAZIONI DI LABORATORIO 1

IL CRIMPAGGIO DI UN CAVO UTP RJ45 CAT 5

Realizzazione di un cavo di rete UTP RJ45

Una volta stabilito il tipo di cavo che si vuole realizzare si procede procurandosi i componenti e gli strumenti necessari per la ◀ crimpatura ▶ e il collaudo.

◀ Con crimpatura (o crimpaggio) si intende l'operazione che consiste nel "fissare" su un'estremità di un cavo elettrico un connettore (o un terminatore). L'operazione di crimpatura viene eseguita con un apposito attrezzo, chiamato *crimpatrice*, e consiste nello schiacciamento meccanico del connettore sul cavo che vi rimane fissato. ▶

Noi utilizzeremo un cavo UTP categoria 5 secondo la specifica EIA/TIA-568A.

Elenco dei componenti

- Due plug di rete maschio di tipo RJ45.
- Un cavo UTP Cat 5 della lunghezza desiderata.


Elenco della strumentazione

- Una pinza a crimpare o crimpatrice (due modelli sono riportati nella figura). ▶
- Un tester per cavi RJ45 (o un normale tester).


Le fotografie mostrano due modelli di crimpatrice.

Le fotografie mostrano due modelli di tester per cavi. ▶

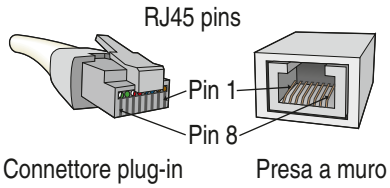

Tester come quelli riportati nella figura hanno un costo inferiore a 10,00 euro e permettono di effettuare le seguenti misure sui cavi:

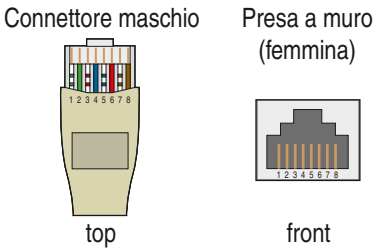
- verifica continuità dei cavi;
- inversioni;
- circuiti aperti;
- corretta crimpatura delle coppie.

Assemblaggio di un cavo tipo diretto secondo lo standard EIA-568A

Prima di procedere con il crimpaggio dobbiamo sempre verificare lo standard del cavo che ci accingiamo a utilizzare, in quanto le specifiche 568A e 568B dello standard EIA/TIA differiscono tra loro per il colore dei cavi. È necessario verificare la specifica del cavo anche in caso di sostituzione di un plug di rete RJ45.

1. Per procedere con la crimpatura si deve innanzitutto avere ben chiara la disposizione dei pin del plug di rete RJ45, illustrati nella figura a lato. ▶





Le estremità di un cavo prendono anche il nome di capocorda. Uno schema pratico di collegamento è quello riportato nella figura a lato. ▶

È buona norma, durante la fase di spellatura, scegliere spezzoni di cavo di lunghezza adeguata (almeno 2 m) per evitare la separazione del fili dalla guaina!

2. Per prima cosa si procedere alla rimozione della guaina esterna del cavo: la maggior parte delle pinze a crimpare possiede una taglierina spellafili chiamata stripper: ▶ Se la pinza di cui disponete non offre questa possibilità basta utilizzare una forbice da elettricista, con la quale incidere la parte esterna della guaina e rimuoverne una sezione di circa 1,5 centimetri, ottenendo una situazione come quella riportata nella figura a lato. ▶

È necessario prestare molta attenzione durante l'incisione della guaina: se vengono incisi anche gli isolanti dei fili posti all'interno, il cavo non sarà in grado di funzionare adeguatamente. Nel dubbio è sempre meglio ripetere l'operazione.

3. Come si può vedere dalla figura precedente, ci troviamo con quattro coppie di file avvolti tra loro a due a due e di colore diverso: ora devi svolgere le coppie di fili in modo da avere otto fili separati e ben appaiati.

Bisogna fare attenzione a distinguerli nel caso non avessero colori diversi: la maggior parte dei cavi ha quattro fili bicolore bianchi/colorati ma esistono in commercio anche cavi con quattro fili completamente bianchi.

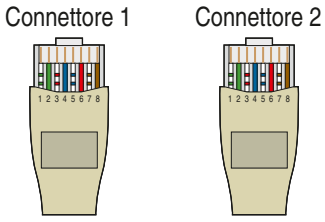
4. Ora è sufficiente aprire i fili "a ventaglio" disponendoli nell'ordine in cui devi crimparli, e ordinandoli da sinistra a destra: l'ordine con cui disporre i file è riportato nella tabella a lato. ▶

Configurazione di un cavo tipo diretto

	Connettore 1		Connettore 2	
TX +	Bianco/Verde	1	Bianco/Verde	TX +
TX –	Verde	2	Verde	TX –
RX +	Bianco/Arancio	3	Bianco/Arancio	RX +
Riservato	Blu	4	Blu	Riservato
Riservato	Bianco/Blu	5	Bianco/Blu	Riservato
RX –	Arancio	6	Arancio	RX –
Riservato	Bianco/Marrone	7	Bianco/Marrone	Riservato
Riservato	Marrone	8	Marrone	Riservato

In generale possiamo affermare che è possibile usare fili di qualsiasi colore, l'importante è mantenere le corrispondenze giuste, eventualmente aiutandosi creando una tabella *ad hoc* dove riportare le corrispondenze dei colori.

Nella figura a lato si vede come devono essere configurati entrambi i connettori del cavo diretto nei due standard T568A e B. ▶



Tenendo il connettore nella posizione indicata e con i fili che entrano dal basso, l'aletta a scatto si trova dietro (per questo motivo nella figura è stata tratteggiata).

Con il connettore nella posizione indicata, i fili entrano dal basso e l'aletta si trova sotto.

5. Avvicina i fili tenendoli stretti tra due dita "appiattendoli" in modo che siano ben affiancati tra loro, avendo cura di togliere le eventuali leggere curvature presenti: occorre fare attenzione a mantenere l'ordine previsto dallo schema della figura.

6. Taglia ora con la pinza (o con le forbici) le parti eccedenti i due centimetri per ottenerli tutti della stessa lunghezza.

7. A questo punto i fili sono pronti per essere inseriti nel connettore RJ45: come riportato nella figura, il connettore deve essere posizionato con la linguetta in basso, non in vista. La guaina isolante deve arrivare giusto vicino al bordo del connettore.

8. Prima di inserire il connettore nella pinza è necessario effettuare un ultimo controllo sul lato trasparente del connettore stesso per verificare che i fili siano giunti "sino in fondo".
Ora inserite il connettore nella pinza e, stringendo la pinza con due mani, crimpate il connettore in modo da fermare i fili: la pinza si apre automaticamente solo quando siete "arrivati" fino in fondo, cioè a fine corsa, avendo terminato la crimpatura del connettore.

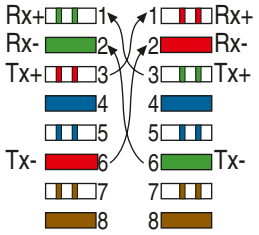
9. Lo stesso procedimento deve essere ripetuto per crimpare il cavo dall'altro lato.

Realizzazione di un cavo di rete UTP RJ45 incrociato (Crossover)

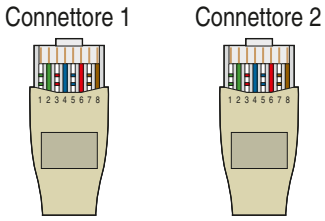
Questo è un cavo di rete particolare che viene usato quando si vogliono connettere tra loro due computer direttamente senza passare attraverso apparecchi come router, switch, hub o simili. Un'altra applicazione che utilizza questa tipologia di cavo, nel campo delle reti di computer, è quella che permette di collegare più hub (e solo hub) in cascata, in modo da fare condividere la stessa linea a più elementi senza limiti virtuali. Il cavo di rete Crossed (o Crossover) si differenzia da un cavo normale in quanto presenta quattro elementi invertiti ai due estremi (cioè in corrispondenza dei due capicorda).

Configurazione di un cavo tipo incrociato				
	Connettore 1		Connettore 2	
RX +	Bianco/Verde	1	Bianco/Arancio	RX +
RX –	Verde	2	Arancio	RX –
TX +	Bianco/Arancio	3	Bianco/Verde	TX +
Riservato	Blu	4	Blu	Riservato
Riservato	Bianco/Blu	5	Bianco/Blu	Riservato
TX –	Arancio	6	Verde	TX –
Riservato	Bianco/Marrone	7	Bianco/Marrone	Riservato
Riservato	Marrone	8	Marrone	Riservato

Si può osservare che alcuni fili sono "incrociati" nello schema di collegamento proprio per permettere lo scambio diretto dei segnali di ricetrasmissione tra i due terminali senza dispositivi intermediari che si occupino di fare questo lavoro. ▶



Lo schema di cablaggio è raffigurato a lato. ▶



Per realizzare questo cavo basta procedere con la stessa sequenza di operazioni eseguite per la crimpatura di un cavo diritto; cambia solo la disposizione dei fili nel secondo connettore.

Se questo cavo vi serve per applicazioni seriali, allora non vanno collegati gli elementi corrispondenti ai numeri (4, 5) e (7, 8).

Zoom su...

DAL CAVO DIRETTO AL CAVO CABLATO

Supponiamo di avere a disposizione un cavo diretto "in eccedenza", di non avere connettori da crimpare oppure di non essere in possesso della pinza crimpatrice e di avere urgenza di realizzare un cavo crossato.
È possibile trasformare il cavo RJ45 diritto e farlo diventare crossover seguendo una procedura "artigianale" (direi di emergenza e poco professionale):

1. tagliate al centro il cavo dividendolo in due parti e spellate gli otto fili sia da una parte che dall'altra;
2. unite con il nastro isolante il filo 1 di un'estremità al filo 3 dell'altra estremità;
3. unite con il nastro isolante il filo 2 di un'estremità al filo 6 dell'altra estremità;
4. unite con il nastro isolante il filo 3 di un'estremità al filo 1 dell'altra estremità;
5. unite con il nastro isolante il filo 6 di un'estremità al filo 2 dell'altra estremità.

I rimanenti fili si devono lasciare invariati. Naturalmente se avete un saldatore a stagno per "giuntare" i fili è preferibile realizzare delle saldature piuttosto che utilizzare il nastro isolante, che utilizzerete invece per isolare i singoli cavi.

MODULO 5 LE RETI ETHERNET E LO STRATO DI COLLEGAMENTO

OBIETTIVI

- Conoscere l'evoluzione di Ethernet
- Conoscere il formato dell'indirizzo MAC
- Conoscere il formato di una trama Ethernet
- Comparare il modello OSI ed Ethernet
- Apprendere la nomenclatura e la struttura del frame
- Conoscere le caratteristiche del CSMA/CD
- Apprendere il concetto di timing, interframe spacing e tempo di backoff
- Conoscere Ethernet ad alta velocità: Fast e Giga Ethernet
- Spiegare il livello MAC e il formato del frame Ethernet
- Conoscere la differenza tra repeater, bridge, hub, switch
- Apprendere il concetto di dominio di collisione

ATTIVITÀ

- Classificare le tipologie di Ethernet
- Saper decodificare un indirizzo MAC
- Saper individuare i campi del frame Ethernet
- Calcolare lo slot time alle diverse velocità di funzionamento
- Calcolare il Round Trip Delay alle diverse velocità di funzionamento
- Individuare le collisioni
- Saper distinguere i diversi errori in Ethernet
- Saper individuare i campi di un frame in formato esadecimale
- Saper realizzare una tabella di filtering
- Saper segmentare una rete

UNITÀ DIDATTICA 1

LA TECNOLOGIA ETHERNET

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere l'evoluzione di Ethernet
- a interpretare l'indirizzo MAC
- a individuare il formato del frame

Generalità

◀ LAN Si tratta di un sistema di comunicazione che permette ad apparecchiature indipendenti di comunicare tra loro entro un'area delimitata, utilizzando un canale fisico a velocità elevata e con basso tasso d'errore. ▶

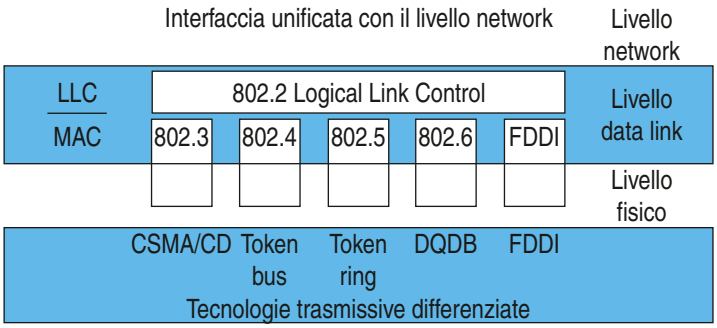
Negli anni '60 l'elaborazione dei dati in azienda era delegata ai mainframe, con l'elaborazione centralizzata in un'unica unità di elaborazione e un insieme di terminali "stupidi" collegato a essa. Con l'avvento dei PC a partire dagli anni '70 iniziarono a presentarsi sul mercato americano le prime ◀ LAN ▶, cioè le prime reti realizzate per connettere più calcolatori.
A quel punto, iniziarono a diffondersi diverse tipologie di LAN, differenti per architetture, topologie e strumenti trasmissivi utilizzati. L'organismo mondiale ◀ IEEE ▶ avviò un progetto identificato con la sigla 802 per cercare di studiare, classificare e standardizzare le LAN, partendo dall'idea che tutte le reti dovessero avere una comune interfaccia verso il terzo livello di organizzazione ISO/OSI, cioè verso il livello network.

◀ L'acronimo IEEE identifica l'associazione internazionale di scienziati professionisti con l'obiettivo di cercare nuove applicazioni e teorie in varie discipline scientifiche (informatica, telecomunicazioni, biomedica ecc.). È l'acronimo di Institute of Electrical and Electronic Engineers (Istituto degli ingegneri elettrici ed elettronici). Si occupa di definire e pubblicare gli standard delle discipline indicate per migliorare la qualità della vita dell'uomo favorendo la conoscenza e l'applicazione delle nuove tecnologie. Le pubblicazioni dello IEEE rappresentano una buona parte della documentazione ingegneristica mondiale, coprendo quasi tutti gli aspetti dell'informatica e delle telecomunicazioni moderne, avendo definito oltre 1000 standard industriali. ▶

Il progetto IEEE 802 si colloca al livello 2 della pila ISO/OSI chiamato data link (collegamento). Il livello data link è suddiviso in due sottolivelli:

- l'LLC, o Logical Link Control, che costituisce la parte superiore del livello di collegamento dati verso il livello network, mascherando la peculiarità della rete utilizzata;
- il MAC, o Media Access Control, che costituisce la parte inferiore del livello di collegamento dati verso il livello fisico, per risolvere il problema dell'accesso a un unico mezzo trasmissivo.

Dato che al livello fisico della pila ISO/OSI è necessario gestire tecnologie trasmissive di tipo diverso, a ciascuna LAN viene associato un MAC specifico. In tal modo avremo un LLC comune a tutte le LAN e un MAC specifico per ogni LAN, come indicato nello standard IEEE, il cui schema logico è riportato nella figura a lato: ▶



Il progetto 802 prevede 6 sezioni, indicate come segue:

- 802.2 Logical Link Control
- 802.3 CSMA/CD (Carrier Sense, Multiple Access with Collision Detection)
- 802.4 Token Bus
- 802.5 Token Ring
- 802.6 Metropolitan Area Networks - DQDB (Distributed Queue, Dual Bus)

Successivamente sono state aggiunte altre sezioni del progetto OSI, per seguire l'evoluzione tecnologica dei mezzi trasmissivi. Ricordiamo per esempio:

- 802.8 Fiber-optic technical advisory group;
- 802.11 Wireless network.

Ethernet

Nei primi anni '70 alcuni produttori mondiali, tra i quali Digital, Intel e Xerox, formarono un consorzio (DIX) per lo sviluppo di una rete locale lavorando sulle prime due versioni di Ethernet: la 1.0, operante a 10 Mb/s, e successivamente la 2.0. Contemporaneamente l'IEEE, basandosi proprio su Ethernet, iniziò lo sviluppo dello standard 802.3, che venne pubblicato nel 1985, fu approvato dall'ISO come DIS (*Draft International Standard*) ISO/DIS 8802.3 e nel 1989 divenne lo standard ISO 8802.3.

Oggi si identificano con Ethernet tutti i dispositivi conformi alle specifiche 802.3.

La grande diffusione di Ethernet è stata facilitata dai costi ridotti degli apparati e dalla grande facilità di progettare e realizzare reti di piccole dimensioni; queste caratteristiche l'hanno resa, in pochi anni, la tecnologia di rete più diffusa per le reti locali. Possiamo affermare, in estrema sintesi, che per utilizzare la tecnologia Ethernet per la connessione alla rete è necessario avere collocato nel computer o nel dispositivo una scheda di comunicazione che da essa prende il nome: la scheda Ethernet.
La scheda di rete spesso viene indicata con l'acronimo NIC (*Network Interface Card*), che genericamente contrassegna i dispositivi che forniscono l'interfaccia hardware tra un computer e una

rete. Tutte le schede prodotte attualmente supportano la Plug and Play (PnP), non necessitano cioè di installazione specifica, ma vengono riconosciute come periferiche standard e quindi configurate e installate automaticamente.

IL MODELLO TOKEN RING

Il modello Token Ring fu un competitore agguerrito rispetto al modello Ethernet. Le reti di tipo Token Ring vennero progettate nei laboratori IBM nel 1976, utilizzando una topografia ad anello, contrapposta a quella a stella di Ethernet. Negli anni successivi Ethernet diventò assai più diffusa grazie a caratteristiche trasmissive di qualità superiore in termini di velocità e grazie all'introduzione degli switch.

La larghezza di banda (broadband) o la capacità di trasmissione dei dati (throughput) di Ethernet era inizialmente di 10 Mbps. Nel 1995 venne introdotto un nuovo standard, la Fast Ethernet, che operava a una velocità di 100 Mbps. Attualmente esiste la tecnologia Gigabit Ethernet, in grado di operare alla velocità di 1000 Mbps (1 Gbps).
I diversi standard Ethernet sono individuati da una sigla composta da tre elementi, come quelli elencati di seguito:

- 10 base-2,
- 10 base-5,
- 10 base-T,
- 100 base-T

dove

- 10\100: indica la velocità massima di trasferimento dati espressa in Mbps;
- *base*: si riferisce al modo dell'onda portante del segnale;
- 2\T\F\U\TX,T2,...: indica la massima distanza raggiungibile da un segmento, o il tipo di conduttore F = fibra, T = cavo UTP, TX = cavo UTP intrecciato, T2 cavo UTD a 2 coppie ecc.

◀ Il termine broadcast indica una modalità di instradamento per la quale un pacchetto viene consegnato a tutti gli host collegati alla rete, a differenza di unicast che indica l'invio a un solo host e multicast che indica l'invio solo a un gruppo di host. ▶

L'accesso alla rete avviene in modo non deterministico, seguendo il modulo CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*), il quale ha come struttura fisica quella a bus o a stella, e come struttura logica quella a ◀ broadcast ▶ (cioè con spedizione simultanea dello stesso messaggio a molteplici destinatari).

La comunicazione tra due PC avviene mediante uno scambio di dati suddivisi in pacchetti e per inviarne uno è necessario che nessun altro pacchetto stia viaggiando sulla rete: se due o più stazioni iniziano a trasmettere i dati contemporaneamente si possono verificare delle collisioni e una stazione deve riprovare a trasmettere il pacchetto dopo una periodo di attesa.

La modalità di comunicazione è simile a quella che intercorre tra due interlocutori al telefono quando, durante una conversazione, una persona deve attendere che l'altra taccia prima di poter parlare.

Naturalmente all'aumentare del numero di PC che cercano di utilizzare la rete aumenta contestualmente anche il numero di collisioni, errori e ritrasmissioni, pregiudicando le prestazioni della rete: se si supera il 50% della larghezza di banda si ha un drastico calo delle prestazioni, dovuto alle collisioni, che può portare a un rallentamento estremo della rete se non al blocco totale.

Indirizzo MAC

◀ Indirizzo IP È un numero composto da 4 byte (indirizzo logico) univoco, assegnato a ogni host presente nella rete dal livello di rete (livello 3). ▶

Ogni dispositivo che si connette a una rete Ethernet può essere individuato univocamente grazie all'indirizzo MAC. Mentre l'◀ indirizzo IP ▶ può essere ripetuto all'interno di reti LAN o sottoreti diverse, l'indirizzo MAC è unico al mondo perché assegnato dal produttore a ogni scheda di rete esistente.
Ogni scheda NIC possiede un indirizzo MAC (MAC address) composto da un insieme di 6 byte (48 bit) separati da due punti:

- i primi 3 byte (24 bit) si chiamano OUI (Organizational Unique Identifier) e sono il codice identificativo dell'azienda produttrice;
- i secondi 3 byte sono il numero seriale progressivo che identifica il prodotto di quell'azienda.

Un esempio di codice MAC è il seguente:

02-60-8C:E9:8B:01

dove

- 02-60-8C: sta a indicare il produttore, in questo caso la 3Com;
- E9:8B:01: è il numero di matricola che la 3Com ha assegnato a questo prodotto.

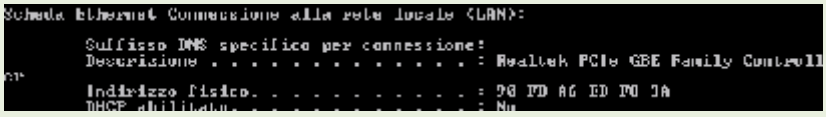
L'indirizzo MAC, anche detto indirizzo fisico (physical address o hardware address) o BIA (*Burn In Address*), risiede nella ROM della NIC e viene copiato in RAM quando la scheda NIC viene inizializzata.

Zoom su...

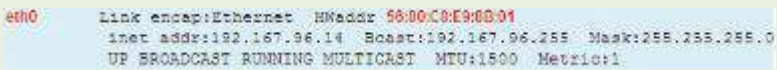
INDIRIZZO MAC

Per individuare l'indirizzo MAC di una scheda presente in un qualsiasi personal computer possiamo procedere nel modo seguente.

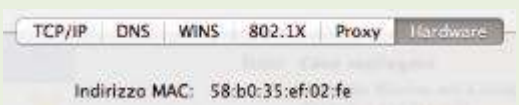
- Per l'ambiente Windows: al prompt dei comandi digitando ipconfig/all viene visualizzato un insieme di dati, tra cui l'indirizzo fisico, come illustrato dalla figura seguente:



- Per l'ambiente Linux: avviare una shell e digitare il comando ifconfig -a. La prima riga riporta il MAC address relativo alla scheda di rete (generalmente identificata con eth0) nella voce HWaddr:



- Per l'ambiente Macintosh: nel menu Mela individuare l'elemento Preferenze di sistema e fare clic nella finestra sull'icona relativa a Network. Nel successivo pannello bisogna selezionare l'opzione Ethernet Integrata (oppure Airport se si deve individuare il MAC address della connessione wireless) e fare clic sulla voce Ethernet, che riporterà il MAC address in questione:



Protocol Data Unit (PDU)

PDU

Con l'acronimo PDU (*Protocol Data Unit*) si intende l'unità di informazione che viene scambiata tra due host nel ◀ **protocollo** ▶ di comunicazione di un'architettura di rete a strati.

◀ Il protocollo è l'insieme delle regole sul formato dei messaggi, delle informazioni di servizio, degli algoritmi di trasferimento che guidano il colloquio tra entità dello stesso livello per trasmettere unità di trasferimento. ▶

Riprendiamo la struttura della pila ISO/OSI e analizziamo l'organizzazione dei dati partendo dal livello fisico. Il livello 1 si occupa della trasmissione dei dati trasformandoli in segnali fisici (elettrici, ottici, elettromagnetici): a questo livello l'unità elementare di informazione è di tipo binario e il messaggio che viene trasferito si limita a una sequenza di bit codificata in modo Manchester.
La sequenza di bit è il risultato finale delle trasformazioni che "subisce" il messaggio destinato alla trasmessione, a partire dal livello più alto della pila fino ad arrivare a quello più basso: ogni livello deve aggiungere informazioni di controllo relative al protocollo utilizzato dal livello stesso, sia per poter effettuare gli specifici compiti a lui assegnati sia per trasferire i dati al livello inferiore. Si dice che il messaggio viene imbustato, cioè con "busta di livello N" si indica il dato modificato al livello N della pila ISO/OSI: quest'operazione è chiamata framing.

Zoom su...

CODIFICA MANCHESTER

Nella codifica Manchester i bit vengono codificati attraverso la variazione del segnale che può assumere valori compresi tra –0,85V (Basso) e +0,85V (Alto). La codifica avviene alla commutazione tra i due valori: il valore logico 0 viene codificato attraverso il passaggio del segnale da Basso a Alto, il valore logico 1 viene invece codificato con il passaggio da Alto a Basso.

FRAMING

Il termine framing fa riferimento alla seguenti operazioni:
- incapsulamento dei dati con un'intestazione (header) e una eventuale coda (trailer);
- interpretazione dei bit presenti nelle intestazioni (ed eventualmente nelle code).

Seguiamo i passaggi fondamentali che avvengono sul messaggio iniziale a partire dai dati del mittente al livello più alto fino alla formazione della "sequenza binaria" trasmessa al livello fisico.

1. Ai dati dell'utente viene preposta un'intestazione (header app) dallo strato di applicazione e, strato dopo strato, vengono eseguite alcune operazioni e/o aggiunte altre intestazioni fino a raggiungere lo strato di trasporto.

L'unità di dati a livello di trasporto prende il nome di segmento.

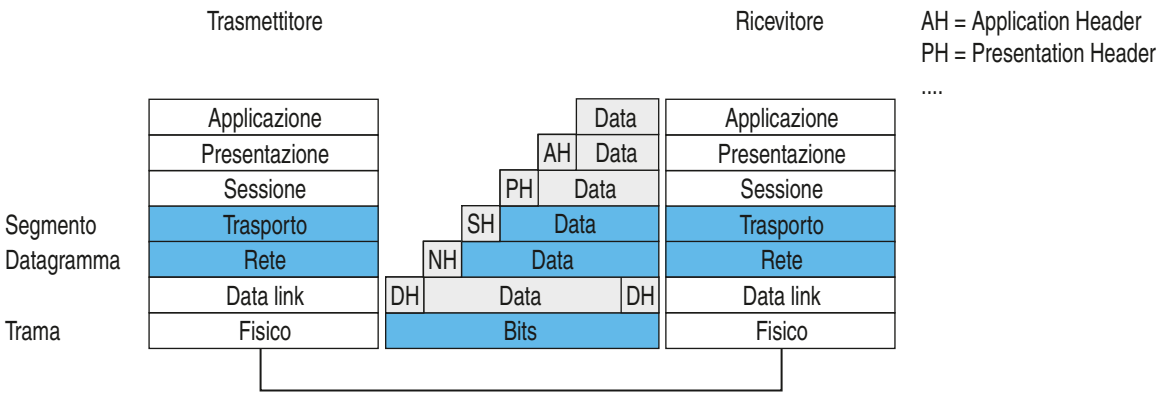
2. Lo strato di trasporto fornisce quindi il segmento allo strato di rete, che presta anch'esso servizi specifici e aggiunge un'intestazione che vedremo nelle prossime unità didattiche.

L'unità di dati allo strato di rete prende il nome di pacchetto (o datagramma IP).

3 Il pacchetto giunge quindi ai livelli inferiori, dove lo strato di collegamento del livello 2 aggiunge la propria intestazione e la propria coda: in Ethernet è la scheda NIC che compie questa operazione per incapsulare i dati da trasmettere.

L'unità di dati allo strato di collegamento prende il nome di frame (o trama).

La trama viene poi trasmessa in rete dallo strato fisico.



Nella figura in alto è mostrato un esempio di "imbustamento" dei dati, nell'ipotesi che la sottorete sia una LAN di tipo Ethernet.

Trama o frame

A livello fisico i dati vengono trasmessi sotto forma di frame (in italiano trama), piccoli contenitori (o pacchetti) che permettono di separare il flusso di dati in unità facilmente controllabili. Il frame nello standard originario 802.3 è lungo almeno 64 byte e non oltre 1518 byte: alcuni standard successivi ne hanno modificato le lunghezze, come per esempio lo standard Gigabit Ethernet 802.3z che ha portato la lunghezza minima del frame a 512 byte.

Nello standard originario 802.3 il frame è composto da:

- preambolo: sequenza di 7 byte, dove ciascun byte è impostato a 10101010 ed è usato per sincronizzare il ricevente prima che i dati effettivi vengano inviati;
- Start Frame Field (SFD): delimitatore di inizio frame (8 bit: 10101011) usato per ottenere l'allineamento al byte;
- indirizzo MAC di destinazione (48 bit), che può essere:
  - unicast: indirizzo MAC univoco dell'adattatore NIC del destinatario;
  - broadcast: tutti i bit a 1;
  - multicast: il primo bit è 1;
- indirizzo MAC di origine: l'indirizzo univoco della stazione mittente (48 bit);
- lunghezza/tipo: due byte che
  - in origine erano 0x05DC (1500_{10}) e indicavano la lunghezza del campo dati;
  - oggi spesso contengono 0x0800 (2048_{10}) per indicare il tipo di pacchetto (protocollo IP);
- corpo del messaggio: da 46 a 1500 byte di dati;
- pad (riempitivo) di zeri, qualora i dati siano meno di 46 byte (così da arrivare a 46 byte tra dati e pad);
- Frame Check Sequence (FCS): sequenza di controllo del frame, ovvero 4 byte per controllare l'integrità del frame tramite CRC-32 calcolato usando il polinomio AUTODIN II (controllo di ridondanza ciclica a 32 bit).

Uno schema del frame IEEE 802.3 è riportato nella seguente figura:

IEEE 802.3						
Preambolo	SFD	Destinatario	Mittente	Lunghezza/tipo	Dati/Pad	FCS
7	1	6	6	2	46-1500	4

Nel frame Ethernet v2, anche chiamato DIX Ethernet (DEC, Intel, Xerox):
- preambolo e SFD sono assieme e formano un preambolo di 8 bytes;
- lunghezza/tipo è un solo tipo.

Ethernet					
Preambolo	Destinatario	Mittente	Tipo	Dati/Pad	FCS
8	6	6	2	46-1500	4

Il calcolo del FCS viene effettuato senza tener conto dei primi 8 byte di preambolo:

		<-- ---------- calcolo del FCS ---------- -->				
Preambolo	SFD	Destinatario	Mittente	Lunghezza/tipo	Dati/Pad	FCS
7	1	6	6	2	46-1500	4

Nel corso degli anni, a più riprese, sono stati aggiunti nuovi campi, resi necessari da parte dei protocolli di strato superiore per l'utilizzo di Ethernet.

INTERFRAME SPACING

Si definisce interframe spacing il minimo spazio tra due frame che deve essere di almeno 96 bit: sotto questo valore un dispositivo continuerebbe a trasmettere sulla linea, monopolizzandola e impedendo agli altri dispositivi di trasmettere.

Nella trasmissione tra un frame e l'altro è quindi obbligatorio lasciare dello spazio, che viene tradotto in un tempo di pausa tra le due trasmissioni e che prende il nome di IFG (*Inter Frame Gap*).

Zoom su...

INTER FRAME GAP

Viene definito IFG l'intervallo di tempo che deve trascorrere tra due pacchetti trasmessi in Ethernet: poiché nell'intervallo devono essere trasmessi 96 bit (12 byte) è immediato calcolare per ogni frequenza di trasmissione il valore IFG (a 10 Mbps deve essere almeno di 9,6 µs). Calcoliamo per le diverse frequenze il tempo di Inter Frame Gap:

Velocità	Interframe spacing	Inter Frame Gap
10 Mbps	96 bit	9,6 µs
100 Mbps	96 bit	0,96 µs
1 Gbps	96 bit	0,096 µs
10 Gbps	96 bit	0,0096 µs

I dispositivi Ethernet accettano solo frame distanziati da un intervallo di tempo maggiore o uguale all'IFG.

ESEMPIO 1 *Calcolo del massimo numero di frame che possono essere trasmessi in un secondo in una rete Ethernet a 10 Mbps*

Il massimo numero di frame al secondo viene calcolato ipotizzando un messaggio con lunghezza minima, cioè un frame di 84 byte così ottenuto: ▶

Componenti del frame	Dimensione minima
Inter Frame Gap	12 Byte
Preambolo + SFD	8 Bytes
Address destinatario	6 Byte
Address mittente	6 Byte
Tipo (o lunghezza)	2 Byte
Payload (Network PDU)	46 Byte
Check Sequence (CRC)	4 Byte
Dimensione fisica del frame	84 Byte

Il numero di frame al secondo si ottiene dal rapporto tra la velocità di trasmissione e la lunghezza del messaggio trasmesso, cioè

$$\frac{Ethernet\ Data\ Rate\ (bit\ /\ s)}{Dimensione\ del\ frame\ (bit)} = \frac{10.000.000}{84 \cdot 8} = 14.880\ frame\ /\ s$$

Questo valore supera di molto la capacità massima dei dispositivi di rete oggi in commercio (switch e router possono elaborare circa 1000 frame al secondo). Se invece calcoliamo il numero di frame con il messaggio lungo di 1500 byte, la frequenza massima è di circa 812 ed è al di sotto dei valori operativi.
Se in un determinato istante fosse comunque superata, questo non rappresenterebbe un problema, in quanto i dispositivi semplicemente "scarterebbero" i frame in eccedenza demandando il problema a un altro livello di servizio, come vedremo in seguito.

ESEMPIO 2 *Calcolo del massimo throughput dei servizi al secondo livello di Ethernet*

Consideriamo la situazione ideale, con dimensione massima del messaggio utile Ethernet di 1500 byte ed efficienza massima, cioè assenza di collisioni sul cavo. Per prima cosa calcoliamo la lunghezza massima del frame in questa situazione, che è 1538 byte:

La frame rate massima è:

$$\frac{Ethernet\ Data\ Rate\ (bit\ /\ s)}{Dimensione\ del\ frame\ (bit)} = \frac{10.000.000}{1538 \cdot 8} = 812{,}74\ frame\ /\ s$$

Il ◀ throughput ▶ massimo del livello si ottiene con la formula seguente:

$$Frame\ rate \cdot Payload = 812{,}74 \cdot (1500 \cdot 8) = 9.752.880\ bps$$

Abbiamo un valore di throughput con efficienza al 97,5% della capacità del link.

◀ Throughput è un indice che identifica la quantità effettiva di dati trasmessi in un'unità di tempo; si esprime in bit/s e tiene conto delle pause del canale. La traduzione in italiano potrebbe essere *efficienza trasmissiva* e, a differenza della capacità del link che indica esclusivamente la frequenza trasmissiva massima alla quale i dati possono viaggiare, il throughput la calcola al netto delle pause di attesa per il riscontro della corretta ricezione dei pacchetti. ▶

Verifichiamo le conoscenze

1 L'acronimo NIC deriva da:
- ○ *Network Interface Cable*
- ○ *Network Interface Card*
- ○ *Network Internet Card*
- ○ *Network Internet Cable*

2 La sigla MAC deriva da:
- ○ indirizzo macchina
- ○ macchina virtuale
- ○ media access control
- ○ media access card

3 L'indirizzo 01:08:AA:3D:4E:23 è di tipo:
- ○ multicast
- ○ singlecast
- ○ broadcast
- ○ non è un indirizzo valido

4 Quale delle seguenti affermazioni è vera?
- ○ Il protocollo CSMA/CD è utilizzato dalla rete Internet
- ○ Il protocollo CSMA/CD è utilizzato dalla rete Token Ring
- ○ Il protocollo CSMA/CD è utilizzato dalla rete Ethernet
- ○ Il protocollo CSMA/CD è utilizzato dalla rete FDDI

5 Il data link è diviso in due livelli: (*due risposte*)
- ○ il LLC
- ○ il LIC
- ○ il MAC
- ○ il TCP
- ○ l'IP

6 Nel broadcast un pacchetto viene consegnato:
- ○ a tutti gli host collegati alla rete
- ○ a un solo host
- ○ a un gruppo di host
- ○ al ripetitore

7 Quale tra i seguenti campi non è previsto dal frame definito dalla 802.3?
- ○ preambolo
- ○ SFD
- ○ PDU
- ○ indirizzo MAC di destinazione
- ○ indirizzo MAC di origine
- ○ lunghezza/tipo
- ○ PAD
- ○ FCS

8 L'interframe spacing deve essere lungo almeno:
- ○ 9 bit
- ○ 96 bit
- ○ 4 byte
- ○ 96 byte

Test vero/falso

1	La parte superiore del livello data link è il Logical Link Control.	V F
2	La parte inferiore del livello data link è il Media Access Control.	V F
3	La scheda di rete spesso viene indicata con l'acronimo NIC.	V F
4	Ethernet effettua l'accesso alla rete in modo non deterministico.	V F
5	L'unità di dati al livello di trasporto prende il nome di segmento.	V F
6	L'unità di dati allo strato di rete prende il nome di pacchetto (o datagramma IP).	V F
7	A livello di rete "pacchetto" è sinonimo di datagramma.	V F
8	L'unità di dati allo strato di collegamento prende il nome di frame (o trama).	V F
9	Gli indirizzi MAC di una rete Ethernet sono diversi dagli indirizzi di una rete Token Ring.	V F
10	Ethernet è un protocollo di livello MAC che prevede una topologia ad anello.	V F

Esercizi di completamento

1 **Assegna il corrispondente codice IEEE 802 alle seguenti tipologie di connessione fisica:**

Codice	Tecnologia trasmissiva
	Fiber-optic technical advisory group
	Token Bus
	Logical Link Control
	Metropolitan Area Networks - DQDB (Distributed Queue, Dual Bus)
	Wireless network.
	CSMA/CD (Carrier Sense, Multiple Access with Collision Detection)
	Token Ring

2 **L'unità di dati:**

- a livello di trasporto prende il nome di .....
- allo strato di rete prende il nome di ..... (o .....
- allo strato di collegamento prende il nome di ..... (o .....).

3 **Scrivi un indirizzo MAC:**

a) Singlecast .....

b) Broadcast .....

4 **Indica la lunghezza di ciascun campo per un frame IEEE 802.3:**

IEEE 802.3						
Preambolo	SFD	Destinatario	Mittente	Lunghezza/tipo	Dati/Pad	FCS

5 **Indica la lunghezza di ciascun campo per un frame Ethernet:**

Ethernet					
Preambolo	Destinatario	Mittente	Tipo	Dati/Pad	FCS

6 **Calcola il Maximum Frame Rate (Minimum Frame Size) e il Maximum Throughput (Maximum Frame Size) per una rete Ethernet a 1 Gb/s:**

maximum rate $\frac{1.000.000.000 \text{ (}bits\text{ / sec)}}{\text{.....}} = \frac{\text{\_\_\_\_\_}}{\text{.....}} =$ .....

minimum rate $\frac{1.000.000.000 \text{ (}bits\text{ / sec)}}{\text{.....}} = \frac{\text{\_\_\_\_\_}}{\text{.....}} =$ .....

UNITÀ DIDATTICA 2

LE COLLISIONI IN ETHERNET

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere i compiti del sottolivello MAC
- a individuare il meccanismo delle collisioni
- a conoscere i compiti del sottolivello LLC

Introduzione

Abbiamo detto che il progetto IEEE 802 si colloca al livello 2 della pila ISO/OSI chiamato livello data link, o collegamento.

Il livello data link è suddiviso in due sottolivelli:
- il MAC, o *Media Access Control*, che costituisce la parte inferiore del livello collegamento dati verso il livello fisico, per risolvere il problema dell'accesso a un unico mezzo trasmissivo;
- l'LLC, o *Logical Link Control*, che costituisce la parte superiore del livello collegamento dati verso il livello network, mascherando la peculiarità della rete utilizzata.

In questa unità li descriveremo dettagliatamente.

Il sottolivello MAC

Il sottolivello MAC si trova direttamente a contatto con il livello fisico (livello1, cablaggio) e indipendentemente dalla topologia di rete si occupa delle seguenti problematiche:
- regolamentazione dell'accesso al mezzo trasmissivo;
- verifica della lunghezza minima del pacchetto, scartando i pacchetti ricevuti che hanno una lunghezza inferiore al valore minimo ammesso (64 byte);
- riconoscimento dell'indirizzo di destinazione del pacchetto (frame);
- generazione del preambolo: in trasmissione il MAC prepone un preambolo al pacchetto che deve essere trasmesso;
- assemblaggio dei dati provenienti dal sottolivello superiore LLC in trame con l'indirizzo sorgente, di destinazione e il campo per il controllo degli errori;
- rimozione del preambolo: in ricezione il MAC rimuove il preambolo;
- disassemblaggio delle trame ricevute e consegna dei dati al sottolivello LLC;

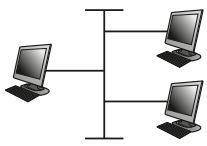
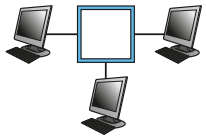
- individuazione degli errori ed eliminazione della trama ritenuta errata.

La criticità delle reti locali è dovuta al fatto che utilizzano un unico mezzo condiviso dai calcolatori connessi: reti di questo tipo sono dette reti broadcast, perché i dati trasmessi sul canale da una macchina vengono ricevuti da tutte le altre.

Per richiedere l'accesso al canale il livello MAC può usare due protocolli di accesso:

- deterministico o protocollo ad accesso controllato: prima di trasmettere è necessario entrare in possesso del controllo esclusivo della risorsa così da evitare qualsiasi tipo di collisione; ogni stazione deve aspettare il proprio turno, che le viene concesso secondo diverse politiche di gestione (tecnica token passing usata da reti Token Ring e FDDI);
- non deterministico o protocollo ad accesso casuale: il primo che arriva inizia a trasmettere senza acquisire il controllo esclusivo della risorsa canale e quindi esiste la possibilità di creare delle collisioni che provocano la perdita dell'informazione; il protocollo gestisce le collisioni con un apposito algoritmo, detto CRA (*Collision Resolution Algorithm*). Questa modalità viene utilizzata da Ethernet ed è chiamata CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*).

La tabella seguente riporta la modalità di accesso al canale per le diverse topologie di rete.

	Topologia logica	Topologia fisica		Protocollo
Ethernet	bus	stella o stella estesa		non deterministico
Token Ring	anello	stella		deterministico
FDDI	anello	doppio anello		deterministico

L'accesso al canale avviene mediante un protocollo di accesso chiamato CAP (*Channel Access Procedure*), una tecnica di *multiplazione statistica*: viene richiesto l'accesso finché il mezzo trasmissivo non diviene libero e se ne entra in possesso.
Non è presente un arbitro che regoli le richieste di accesso al canale e gestisca la risorsa condivisa: ogni PC continua a richiedere l'accesso fino a che "statisticamente" prima o poi troverà il canale libero.

Definizioni

Prima di descrivere il meccanismo con cui il protocollo CSMA/CD permette a una stazione di entrare in possesso del mezzo trasmissivo è necessario premettere alcune definizioni.

Per ridurre il tasso di collisioni si può suddividere la LAN in più porzioni dette "domini di collisione" in modo che la contesa del mezzo avvenga solo tra stazioni appartenenti a un singolo dominio.

BIT TIME

Con bit time si indica il tempo durante il quale un bit permane sulla linea che dipende, quindi, dalla frequenza di trasmissione.

ESEMPIO

Con trasmissioni a 10 Mbps la durata di un bit time è

$$t_{bit} = \frac{1}{f} = 10 \cdot 10^{-6} s$$

Il bit time è "l'unità moltiplicatore" per la determinazione dello slot time.

SLOT TIME

Con il termine slot time (ST) si indica il tempo che impiega uno slot a compiere il tragitto di massima distanza teorica tra due nodi.

Gli slot time attualmente in uso sono:
- per Ethernet da 100 o 10 Mbps è il tempo di 64 ottetti (512 bit time);
- per Ethernet da 1000 Mbps è quello di 512 ottetti (4096 bit time).

Moltiplicandoli per il bit time otteniamo la seguente tabella:

Velocità	Bit time	Slot time (in bit)	Slot time (in microsecondi)
10 Mbit/s	$1/10 * 10^{-6} = 100$ ns	512 × bit time	51,2 microsecondi
100 Mbit/s	$1/100 * 10^{-6} = 10$ ns	512 × bit time	5,12 microsecondi
1 Gbit/s	$1 * 10^{-9} = 1$ ns	4096 × bit time	4,096 microsecondi
10 Gbit/s	$1/10 * 10^{-9} = 0,1$ ns	Non applicabile	

A questi tempi vanno sommati i ritardi dovuti ai dispositivi presenti sulla rete.

ROUND TRIP DELAY (RTD)

Se indichiamo con T il tempo massimo di propagazione del segnale tra due stazioni qualsiasi presenti sulla rete, 2T rappresenta il tempo massimo di trasmissione per una stazione prima di accorgersi di una collisione (andata e ritorno).
Il tempo 2T prende il nome di round trip delay.

Lo slot time deve essere maggiore del round trip delay: è necessario che un segnale viaggi tra i due punti più lontani del dominio di collisione, collida eventualmente con un'altra trasmissione proprio all'ultimo istante possibile e i frammenti della collisione ritornino verso la stazione trasmittente; l'ampiezza dello slot time deve garantire questa situazione estrema.

Verifichiamo che l'ST sia "appena più lungo" dell'RTD calcolando i valori nel caso peggiore.
Se poniamo come distanza quella massima ammessa di 2,5 km, il round trip delay è il tempo che un impulso impiega per percorrere 5 km.

ESEMPIO

La velocità della luce nel conduttore è pari a circa 2/3 di quella nel vuoto:

$$v_c = 200 \cdot 10^6 \text{ km/s}$$

Il tempo impiegato per percorrere un chilometro è

$$t_{km} = \frac{1}{200 \cdot 10^6} = 5 \text{ µs}$$

La distanza teorica massima a cui ci si riferisce è la distanza del dominio di collisione che l'IEEE 802.3 per la 10 BASE 5 indica in 2500 m (500 m per 5 segmenti) alla velocità 10 Mbps, quindi:

$$t_{max} = 25 \text{ µs}$$

Per compiere il tragitto di andata e ritorno ci metterà il doppio, cioè 2T, e otteniamo

$$2T = 50 \text{ µs}$$

che risulta essere inferiore dei 51,2 microsecondi dello slot time.

SPAZIO INTERFRAME

Il minimo spazio tra due frame che non collidono è detto spazio interframe. Si misura dall'ultimo bit dell'FCS al primo bit del preambolo del secondo frame ed è fissato a 96 bit time.

Su una Ethernet a 10 Mbps dopo che un frame è stato spedito, tutte le stazioni devono aspettare un minimo di 96 bit time, cioè 9,6 microsecondi prima di essere autorizzate a trasmettere un frame.

Rilevamento delle collisioni

Parte delle tre funzioni svolte con il protocollo CSMA/CD, e cioè:

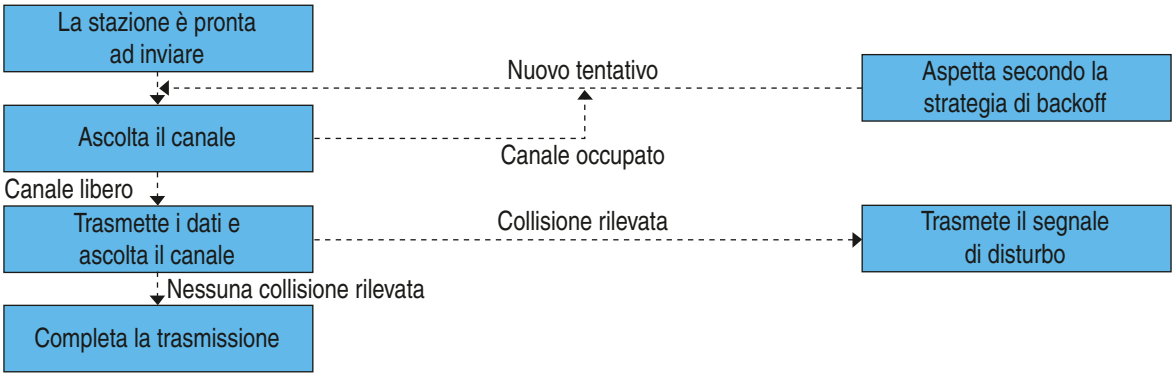
- trasmettere e ricevere i pacchetti;
- rilevare gli errori nei pacchetti;
- decodificare i pacchetti e controllare gli indirizzi prima di passarli ai livelli superiori;

è realizzata da due sottosistemi che operano a livello MAC:

1. CAP (*Channel Access Procedure*): l'insieme delle procedure che la stazione effettua per realizzare l'accesso al canale;
2. CRA (*Collision Resolution Algorithm*): l'insieme delle procedure che la stazione effettua per rilevare ed eventualmente recuperare situazioni di collisione.

Vediamo il flow chart delle operazioni eseguite per *effettuare la trasmissione*.

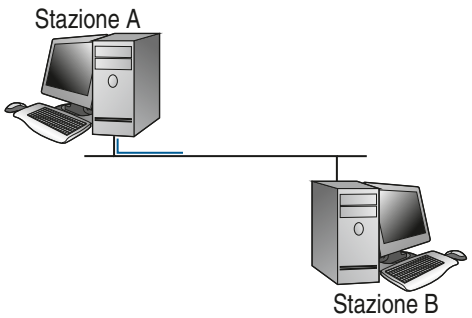
- Se la linea è inattiva (idle) non viene rilevata alcuna portante, quindi:
  - viene immediatamente inviato il pacchetto di dimensione massima 1500 byte;
  - si ascolta che nessun altro inizi a trasmettere;
  - si deve aspettare 9,6 microsecondi tra i frame back to back.
- Se la linea è occupata (busy) viene rilevata una portante e quindi:
  - si aspetta un tempo casuale prima di riprovare (*strategia di backoff*);
  - si riprova fino a quando la linea ritorna libera per procedere all'immediata trasmissione del pacchetto.
- Se viene rilevata la collisione:
  - si interrompe l'invio e si inserisce il segnale di disturbo (jam signal);
  - si riprova successivamente.



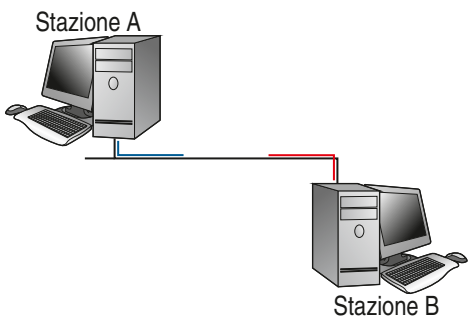
Ci si accorge della presenza di una collisione analizzando la dimensione del segnale (valore di tensione) che diventa maggiore rispetto allo stato normale: il primo adattatore che si accorge di una collisione, invece di sospendere la trasmissione, cerca di aumentare le collisioni trasmettendo un segnale di disturbo (un jamming signal composto da un'alternanza di 0 e di 1) in modo che tutti gli utilizzatori del mezzo trasmissivo si accorgano al più presto della collisione.

Vediamo graficamente quanto descritto:

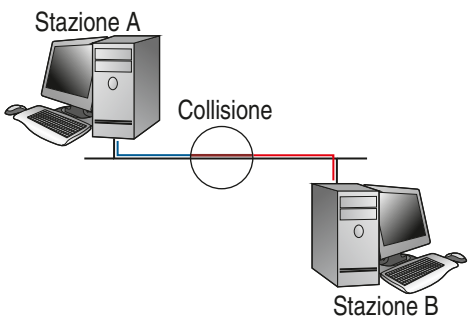
A La stazione A, che deve iniziare a trasmettere, si mette in ascolto sul filo e se non percepisce nessun segnale nel tempo stabilito dallo spazio interframe di 9,6 microsecondi, significa che il canale è libero e che può iniziare a trasmettere il suo frame di dati a partire dal preambolo 64 bit (vedi figura a lato), rimanendo però sempre in ascolto per percepire eventuali valori di tensione anomala sul filo, che indicherebbero la presenza di una collisione generata dall'inizio della trasmissione di una seconda stazione.



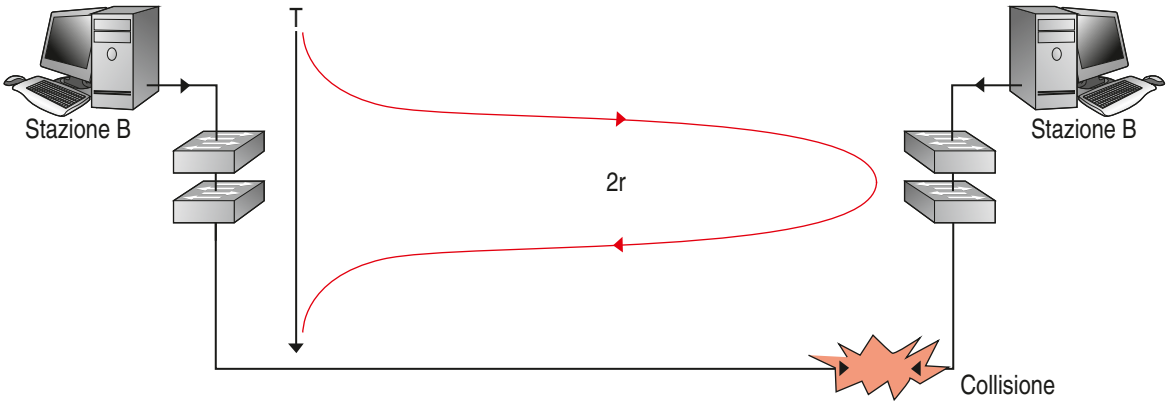
B Supponiamo che la stazione B, dopo aver atteso anch'essa 9,6 microsecondi senza sentire segnali sulla linea, inizi a trasmettere prima che il segnale inviato dalla stazione A si propaghi sulla linea e arrivi a destinazione: siamo nella situazione che entrambe le stazioni stanno trasmettendo contemporaneamente sullo stesso conduttore, come indicato nella figura a lato:



C Quando i due segnali collidono si sovrappongono e generano una situazione con *valori anomali* di tensione elettrica che si propaga sul filo in entrambe le direzioni (vedi figura a lato).



Nel caso peggiore, quando B inizia a trasmettere appena prima che gli arrivi il messaggio di A, quindi dopo un tempo T da quando il messaggio è stato trasmesso da A, la stazione A continua a trasmettere fino a che si accorge della collisione, cioè fino a quando il segnale alterato le arriva da B in un tempo T; nel caso peggiore, quindi, solo dopo un tempo 2T (round trip delay) possiamo essere certi che non ci sono state collisioni.
Ricordiamo che a 10 Mbps la distanza di 2T = 5 km viene coperta in 50 microsecondi, tempo che corrisponde al round trip delay.



ESEMPIO

Conoscendo il round trip delay, calcoliamo la lunghezza minima che deve avere un frame per occupare la linea in tutta la sua durata (imponiamo che il tempo di trasmissione deve essere uguale o maggiore del round trip).
Con velocità di trasmissione a 10 Mbps, imponendo che il tempo di trasmissione sia almeno 2T = 50 microsecondi, si ottiene la dimensione minima F_{min} che deve avere un frame affinché la stazione emittente, durante la sua spedizione, possa accorgersi che è avvenuta una collisione e possa intervenire per "eliminare" il dato trasmesso, che sicuramente risulta danneggiato:

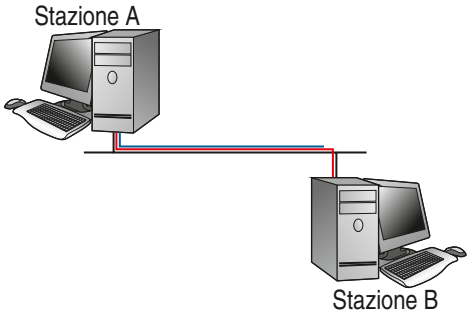
$$\frac{F_{min}}{10^7} > 50 \cdot 10^6$$

da cui si ottiene

$$F_{min} > 10^7 \cdot 50 \cdot 10^6 = 500 \text{ bit}$$

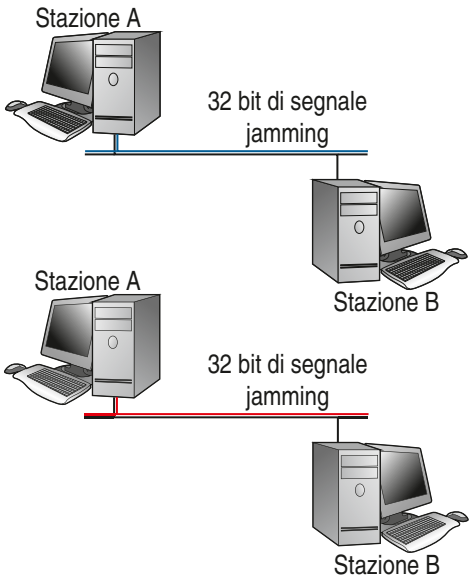
che viene arrotondato a 512 bit, cioè a 64 byte.
A 10 Mbps i 64 byte si ottengono semplicemente dalla somma dei 14 byte di intestazione (destinatario + mittente + tipo), dei 46 byte dei dati, dei 4 byte CRC.

D Dato che ogni stazione rimane sempre in ascolto sulla linea, entrambe si accorgono dell'avvenuta collisione ma in tempi diversi, quindi sospendono la trasmissione del messaggio e iniziano a trasmettere un segnale di disturbo (un jamming signal lungo 48 bit, composto da 0 e 1 alternati): la prima a trasmetterlo nel nostro esempio è la stazione A.



E Il segnale di jam viene subito individuato da tutte le stazioni che sono in ascolto sul filo; con esso riconoscono che il filo è occupato e quindi non possono parlare. Inoltre, dato che il messaggio ricevuto non ha un valido CRC, viene scartato perché riconosciuto errato da tutti i riceventi.

F In ogni stazione viene mandato in esecuzione un algoritmo che determina il tempo che questa stazione dovrà attendere prima di iniziare una nuova trasmissione (questo algoritmo si chiama Truncated Binary Exponential Backoff Algorithm, noto anche semplicemente con il nome di algoritmo di backoff).
In un secondo tempo anche la stazione B individua la collisione, trasmette il suo segnale di jamming e si pone in attesa.



In una rete Ethernet le collisioni fanno parte proprio del protocollo di comunicazione, quindi non sono un problema ma uno strumento per regolare la gestione del mezzo trasmissivo.

Gli errori Ethernet

La trama Ethernet è costituita da un preambolo di 8 byte, che ha lo scopo di assicurarsi dell'effettiva disponibilità della linea: se avvengono delle collisioni, queste hanno luogo generalmente sul preambolo, quindi prima che inizi la trasmissione del segnale SFD di inizio messaggio.

Non appena la collisione viene rilevata, le schede NIC d'interfaccia di entrambe le stazioni non interrompono immediatamente la trasmissione, ma continuano a inviare bit fino a raggiungere la dimensione minima di un pacchetto di 64 byte; questo accorgimento è necessario per fare in modo che le altre eventuali macchine presenti sulla rete si accorgano che la collisione è in corso e che la rete è momentaneamente bloccata.

Successivamente interrompono la trasmissione e attivano un timer nell'algoritmo di backoff di durata casuale prima di ritentare la trasmissione. Il fatto che il timer sia casuale impedisce che entrambe ripartano nello stesso istante, causando una nuova collisione. Se, nonostante l'uso dei timer, la collisione si verificasse ancora, il timer verrebbe allungato progressivamente fino a un punto in cui il continuare delle collisioni indicherebbe un guasto fisico sulla rete e le singole schede d'interfaccia comunicherebbero al rispettivo computer l'impossibilità di trasmettere.

ESEMPIO

Vediamo un esempio con due stazioni poste a 60 metri di distanza:

A alla velocità di 10 Mbit per secondo ci vogliono 100 nanosecondi per inviare un singolo bit:

$$t_s = 1/10*10^6 = 100*10^{-9}$$

B l'impulso elettrico viaggia alla velocità della luce, ma su un cavo di rame la propagazione non è istantanea per effetto del "ritardo di propagazione" (65%), quindi ≈ 200.000 km/s.
Otteniamo lo spazio percorso dalla relazione fisica che lega lo spazio con la velocità:

$$s = v*t = 200*10^6 \text{ x } 100*10^{-9} = 20 \text{ m}$$

- in un nanosecondo il primo bit ha quindi percorso 20 centimetri e prima che il secondo bit sia "uscito dalla scheda di rete" che sta trasmettendo (100 nanosecondi), il primo bit ha percorso circa 20 metri.
  Prima che tutto il preambolo sia stato trasmesso, cioè 48 bit in 4,8 millisecondi, il primo bit ha percorso 960 metri!

Se una seconda stazione, a 60 di metri distanza dalla prima, deve iniziare a trasmettere, lo può fare solo 3 nanosecondi dopo l'inizio della prima, altrimenti le arriva il segnale del preambolo che le fa sentire la linea occupata; se le due stazioni iniziassero contemporaneamente la trasmissione dopo la metà del tempo si troverebbero in collisione.
Se contemporaneamente una terza stazione, ancora più distante, iniziasse a trasmettere, questa collide rebbe con entrambe, e via di seguito.

Al crescere del numero di stazioni presenti sulla rete aumenta il numero di collisioni: anche per questo motivo viene imposto il limite massimo di lunghezza di un segmento di rete Ethernet.

Per poter superare i limiti fisici di lunghezza del cavo, del numero massimo delle stazioni per tratta di cavo e di numero massimo di ripetitori (cioè superare il RTD che è di 50 microsecondi) è necessario creare una seconda rete e collegarla alla prima attraverso un dispositivo "ponte" (chiamato bridge) che memorizza ogni messaggio in arrivo da una parte e lo ritrasmette alla rete successiva solo se è destinato a questa, oppure lo scarta se si tratta di un messaggio che deve rimanere all'interno della prima rete.

Così facendo svincoliamo le temporizzazioni della prima rete (che dal punto di vista del bridge diventa un segmento) dalla temporizzazione della seconda.

Zoom su...

L'ALGORITMO DI BACKOFF

Quando una stazione rileva una collisione, aspetta a ritrasmettere avviando l'algoritmo di backoff che decide quanti slot time deve attendere: se aumenta il numero di collisioni aumenta esponenzialmente il range temporale entro cui viene sorteggiato il backoff.
Quando riconosce la prima collisione l'algoritmo sceglie un valore compreso tra 0 e 1; alla seconda collisione il range diventa tra 0 e 3 slot time, alla terza tra 0 e 7 e cresce fino alla decima collisione che è il valore massimo previsto (0-1023 slot time di attesa). Se dopo 16 tentativi l'invio fallisce, il frame viene scartato e sarà un problema dei livelli superiori decidere come comportarsi.

Collision window e late collision

Quando la stazione trasmittente rileva il segnale, essa sospende la trasmissione del frame e inizia a trasmettere i 48 bit di jam.

FRAME DI COLLISIONE

L'insieme dei bit trasmessi, cioè il frame parziale + il jam, prende il nome di frame di collisione.

Determiniamo la lunghezza massima che deve avere questo frame di collisione: definiamo finestra di collisione (collision window) l'intervallo di tempo entro il quale la stazione trasmittente deve accorgersi della collisione che deve avvenire prima di aver finito di trasmettere il pacchetto minimo lungo 64 byte + 8 byte = 72 byte (pacchetto + preambolo), cioè 512 + 64 = 576 bit.
La durata della trasmissione del pacchetto minimo a 10 Mbps è quindi di almeno 57,6 ms che è il massimo valore ammissibile per il round trip delay (alla velocità 10 Mbps su una tratta di 2500 metri sappiamo essere $2T$ = 50 μs).

Il frammento di collisione, costituito dalla somma della parte di pacchetto trasmessa più la sequenza di jamming che viene posta in coda, deve avere una lunghezza inferiore al frame più piccolo, quindi deve durare meno di 576 bit time. Ciò significa che l'ultimo bit di jamming deve essere trasmesso entro 575 bit time dall'inizio della trasmissione.
La lunghezza massima del frammento di collisione pone però dei limiti più restrittivi sul tempo di ritardo della rete rispetto a quelli della collision window: se la collisione venisse rilevata verso la fine del frame, per esempio dopo 530 bit time dall'inizio della trasmissione e si iniziasse a trasmettere la sequenza di jamming di 48 bit, ne risulterebbe un frammento di collisione superiore a 576 bit time (530 + 48 = 578 > 576).

Diamo ora una definizione più dettagliata di round trip delay.

ROUND TRIP DELAY

Il tempo di ritardo massimo, che può intercorrere da quando la stazione ha trasmesso il primo bit del preambolo a quando viene propagato l'ultimo bit di jamming nel segmento di rete su cui essa è collegata, viene chiamato Round Trip Delay (RTD). Esso è quindi il tempo di propagazione del frammento di collisione.

I frammenti di collisione con lunghezza superiore al massimo consentito non vengono visti come tali dalle stazioni in ascolto, ma sono recepiti come pacchetti contenenti errori quali CRC error o alignment error. Se la stazione trasmittente rileva la collisione entro un tempo superiore alla collision window incrementa il contatore delle late collision.

Zoom su...

LATE COLLISION

Con late collision si definiscono le collisioni che avvengono dopo i primi 64 ottetti: la scheda di rete Ethernet ritrasmette automaticamente un frame se la collisione è avvenuta all'interno della collision window, altrimenti considera la trasmissione andata a buon fine. Questo frame viene comunque scartato perché risulta errato il CRC.

Tipologia di errori in Ethernet

Si è detto che le collisioni non sono degli errori ma costituiscono il meccanismo di sincronizzazione di Ethernet; tuttavia sulla rete può verificarsi un insieme di errori di diversa natura, che riassumiamo brevemente:

- **A** errori dovuti alla lunghezza "illegale" del frame:
  - frame troppo corto, con un numero di byte inferiore a 84;
  - frame troppo lungo, con un numero di byte superiore a 1518;

Ⓑ errori di allineamento:
- numero di bit insufficienti o eccessivi per essere raggruppati in ottetti;

Ⓒ errori di inconsistenza:
- differenza tra il numero di byte trasmessi e il numero presente nel campo length;
- lunghezza non corretta del preambolo;
- SFD errato o mancante: in questo caso il frame viene chiamato ghost;

Ⓓ errori di parità:
- FCS error: se una stazione ha un alto numero di errori di parità probabilmente la scheda NIC è difettosa, il cavo presenta dei problemi o si sono rovinati i driver;

Ⓔ errori di collisione:
- late collision: sono le collisioni rilevate fuori dalla collision window e il NIC non effettua la ritrasmissione del frame.

Il sottolivello LLC

Sempre al livello 2, sopra al sottolivello MAC, si colloca il sottolivello LLC, che ha due scopi principali:
- essere il "ponte" tra le diverse tipologie di sottolivello MAC e il livello di rete: in questo modo i diversi tipi di LAN fisica hanno un'interfaccia unificata e svincolata dalle loro caratteristiche hardware;
- mettere a disposizione del livello superiore un servizio più sofisticato sulla gestione della trasmissione di quello offerto dai vari sottolivelli MAC.

Il livello LLC quindi "trasferisce" i dati dal livello superiore (di rete) al livello MAC e viceversa:
Ⓐ incapsula i dati provenienti dal livello superiore in un'unità di trasmissione che prende il nome di LLC-PDU e li trasferisce al livello MAC;
Ⓑ elabora le unità di trasmissione provenienti dal livello MAC e le inoltra al livello superiore.

Il livello MAC offre un servizio non affidabile in quanto il mittente non viene a conoscenza del buon fine della trasmissione (servizio senza connessione); su richiesta dello strato superiore, il sottolivello LLC può invece offrire due servizi aggiuntivi:
- servizio a datagramma confermato: introduce un meccanismo di scambio di messaggi tra destinatario e trasmittente, facendo in modo che al momento della ricezione di una trama venga spedita al mittente la conferma della corretta ricezione. Dopo un intervallo di tempo prefissato, se il mittente non riceve questa conferma, provvede alla ritrasmissione della trama non confermata;
- servizio affidabile orientato alla connessione: in questo caso si crea una connessione stabile tra mittente e destinatario per tutto il tempo necessario alla trasmissione del messaggio, garantendo così che ogni trama sia consegnata correttamente e nell'ordine giusto.

Quindi nel sottolivello LLC vengono offerti tre diversi servizi:
- servizio senza connessione (servizio datagramma);
- servizio senza connessione con accettazione;
- servizio orientato alla connessione.

SERVIZIO DATAGRAMMA

Un servizio di trasferimento dati senza connessione è chiamato servizio datagramma.

Il *servizio senza connessione* è quello che maggiormente viene utilizzato nelle LAN: nello standard LLC sono contemplati tutti e tre i servizi sopra elencati, ma spesso nella realizzazione sono implementati solo uno o due di essi e il servizio senza connessione è *sempre presente*.

Mentre il sottolivello MAC si limita a rilevare gli errori e a scartare le trame errate, l'LLC effettua la gestione degli errori, diversa a seconda del servizio offerto.

A Servizio senza connessione

In questo caso non è previsto alcun controllo di flusso o recupero di errore:

- ogni LLC-PDU inviata viene processata indipendentemente dalle altre e non viene eseguito alcun controllo di sequenza per assicurarsi che le LLC-PDU siano ricevute nella stessa sequenza con la quale sono state inviate;
- il sistema in ricezione non invia alcun messaggio di accettazione come prova dell'avvenuta ricezione.

Tutti i controlli di flusso necessari e il recupero di errori sono a carico degli strati superiori all'LLC.

B Servizio orientato alla connessione

In questo caso è previsto un doppio controllo, sia di flusso che di errore:

- a ogni LLC-PDU viene assegnato un numero progressivo in fase di trasmissione e al ricevimento viene controllato che sia pervenuto nel giusto ordine;
- viene controllata la correttezza di ogni singolo LLC-PDU e, nel caso si individui un frame danneggiato, ne viene richiesta la ritrasmissione.

Se la trasmissione è andata a buon fine viene inviato un messaggio di accettazione in modo da informare il mittente che gli LLC-PDU sono giunti a destinazione: in caso di errori non recuperabili viene rilasciata la connessione e si informano i due utenti del problema che si è verificato.

C Servizio senza connessione con accettazione

Questo tipo di servizio si comporta come "una via di mezzo" rispetto ai due precedentemente descritti:

- viene garantita la consegna di ogni singolo LLC-PDU in quanto il mittente viene informato della consegna e quindi del buon fine o meno della trasmissione;
- ogni LLC-PDU inviato viene processato indipendentemente dagli altri e non viene eseguito alcun controllo di sequenza per assicurarsi che gli LLC-PDU siano ricevuti nella stessa sequenza con la quale sono stati inviati.

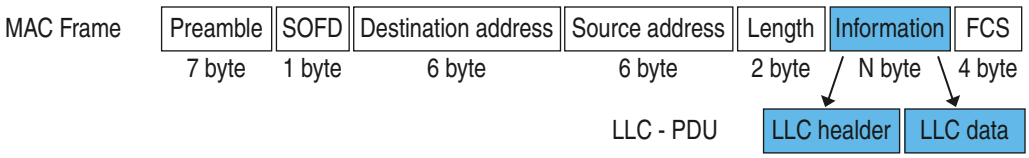
Come già abbiamo visto, il controllo degli errori di trasmissione avviene mediante il calcolo del controllo ciclico di ridondanza CRC sul campo FCS che fa parte del frame MAC.
Per effettuare il controllo di sequenza viene utilizzato il campo Send Count N(S), che contiene il numero di sequenza dell'LLC-PDU che l'entità LLC sta inviando, e il campo Receive Count N(R), che contiene il numero di sequenza che l'entità LLC si aspetta di trovare nel successivo LLC-PDU che riceverà. N(S) e N(R) vengono aggiunti alla trama da ogni sottostrato LLC: quando un mittente invia una LLC-PDU questo aggiorna il suo contatore di invio e il destinatario, quando riceve il frame, confronta il valore del contatore di invio N(S) contenuto nell'LLC-PDU con il suo contatore di ricezione. Se i due valori sono diversi il destinatario rileva un errore di sequenza e quindi respinge il frame.

LLC Protocol Data Unit

Analizziamo ora la struttura dell'LLC-PDU, ovvero come viene modificata la trama che giunge dal terzo livello per poi essere passata al sottolivello MAC.

Una LLC-PDU è organizzata in quattro campi:

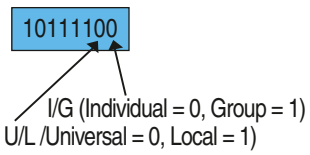
- DSAP (*Destination Service Access Point*): è composto da un byte;
- SSAP (*Source Service Access Point*): è composto da un byte;
- Control: un campo di uno o due ottetti che descrive il tipo di LLC-PDU e contiene le informazioni di controllo, tra cui i contatori N(S) e N(R);
- Data: un campo di lunghezza variabile che contiene i pacchetti contenenti le informazioni che provengono dallo strato superiore.



Dove l'LLC-PDU è così composta:

I SAP di LLC hanno la dimensione di un byte e vengono utilizzati per inserire la codifica del protocollo che a livello superiore ha generato il pacchetto: dato che è composto da 8 bit è possibile codificare fino a 63 tipi di protocolli. Gli ultimi due bit indicano se:

- l'indirizzo è stato assegnato ufficialmente o in base locale (bit U/L);
- l'indirizzo è di un singolo sistema o di un gruppo di sistemi (bit I/G).



La tabella a lato riporta alcuni LLC SAP definiti da IEEE: ▶

Esadecimale	Significato
00	Null LSAP
02	Individual LLC Sublayer Management
x2	Network Management Function
03	Group LLC Sublayer Management
x6	National Body Standard
06	DOD IP
0E	PROWAY-LAN
42	IEEE 802.ID (MAC bridge)
4E	EIA-RS 511
5E	ISI IP
8E	PROWAY-LAN
AA	SNAP
FE	ISO CLNS
FF	Global DSAP

mentre i seguenti sono definiti da IBM: ▼

Esadecimale	Significato
04	SNA Path Control
05	SNA Path Control Group
F0	IBM Netbios
F4	LAN Management Individual
F5	LAN Management Group

Un valore della codifica è FF che indica che la trasmissione è di broadcast, quindi destinata a tutte le stazioni.

Nella figura seguente è riportato nel dettaglio un esempio di LLC Header:

```
LLC:  ----- LLC Header -----
LLC:
LLC:  DSAP = F0, SSAP = F0, Command, I frame, N(R) = 29, N(S) = 109
```

mentre di seguito è riportato un esempio completo di frame:

```
DLC:  ----- DLC Header -----
DLC:
DLC:  Frame 1 arrived at 09:18:37.3543; frame size is 143 (008F hex) bytes.
DLC:  Destination = Station 3Com 676974
DLC:  Source      = Station 3Com 741178
DLC:  802.3 length = 129
DLC:
LLC:  ----- LLC Header -----
LLC:
LLC:  DSAP = F0, SSAP = F0, Command, I frame, N(R) = 29, N(S) = 109
LLC:
NETB: ----- NETBIOS Data Only Last -----

ADDR  HEX                                             ASCII
0000  02 60 8C 67 69 74 02 60   8C 74 11 78 00 81 F0 F0  .`.git.`.t.x....
0010  DA 3A 0E 00 FF EF 16 00   00 00 00 00 63 16 19 01  .:..........k...
```

Verifichiamo le conoscenze

1 **Il livello data link è composto dai seguenti sottolivelli:**

- ○ MAC
- ○ ARP
- ○ LLC
- ○ TCP
- ○ IP

2 **Qual è il tempo minimo che viene stimato dall'IEEE 802.3 sulla distanza di 2500 m?**

- ○ 0,512 microsecondi
- ○ 5,12 microsecondi
- ○ 51,2 microsecondi
- ○ 512 microsecondi

3 **Un jamming signal è lungo:**

- ○ 48 bit alternando 0 e 1
- ○ 64 bit alternando 0 e 1
- ○ 48 byte alternando 0 e 1
- ○ 64 byte alternando 0 e 1

4 **Un preambolo è lungo:**

- ○ 48 bit alternando 0 e 1
- ○ 64 bit alternando 0 e 1
- ○ 48 byte alternando 0 e 1
- ○ 64 byte alternando 0 e 1

5 **Gli errori in Ethernet possono essere:** (*indica la risposta errata*)

- ○ dovuti alla lunghezza del frame
- ○ di allineamento
- ○ di ritardo
- ○ di inconsistenza
- ○ di parità
- ○ di collisione

6 **Il sottolivello LLC offre due servizi aggiuntivi:**

- ○ servizio a datagramma confermato
- ○ servizio a datagramma ritrasmesso
- ○ servizio orientato alla connessione
- ○ servizio orientato alla collisione

7 **Il sottolivello LLC:**

- ○ non effettua funzioni di rilevamento e correzione di errori
- ○ rileva gli errori di sequenza errata
- ○ rileva e corregge gli errori
- ○ rileva gli errori e chiede la ritrasmissione del segmento errato

Test vero/falso

1	Il sottolivello Media Access Control costituisce la parte inferiore del livello 2.	V F
2	Il sottolivello Logical Link Control costituisce la parte superiore del livello 2.	V F
3	Il protocollo deterministico è anche detto protocollo ad accesso controllato.	V F
4	Il protocollo di accesso CAP significa Channel Access Protocol.	V F
5	La CAP è una tecnica di multiplazione statistica.	V F
6	Con bit time si indica il tempo durante il quale un bit permane sulla linea.	V F
7	L'ST deve essere più lungo dell'RTD.	V F
8	Con jam signal si intende un segnale di disturbo.	V F
9	Un jamming signal è composto da un'alternanza di 0 e di 1.	V F
10	Una collisione genera valori anomali di tensione elettrica sul canale.	V F
11	Nel frammento di collisione è compreso anche il jamming signal.	V F
12	Un servizio di trasferimento dati senza connessione è chiamato servizio datagramma.	V F
13	Il servizio a datagramma confermato introduce un meccanismo di scambio di messaggi.	V F
14	Nel servizio senza connessione non è previsto alcun controllo di flusso o recupero di errore.	V F
15	Nel servizio orientato alla connessione è previsto un doppio controllo.	V F

Esercizi di completamento

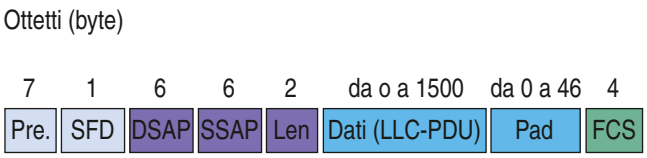
1 **Scrivi il significato dei seguenti acronimi:**

CSMA .....
MAC .....
LLC .....
CAP .....
CRA .....
RTT .....
ST .....
RTD .....
DSAP .....
SSAP .....

2 **Indica per ciascuna topologia di rete il protocollo utilizzato:**

Ethernet .....
Token Ring .....
FDDI .....

3 **Considera la struttura del seguente pacchetto Ethernet:**



a) a che cosa serve il campo PAD?

.....

b) scrivi il contenuto del campo DSAP in caso di trasmissione broadcast:

.....

Esercizi

1 **Calcola la lunghezza minima del frame per una rete che operi a una velocità di un 1 Gbit/s volendo mantenere una distanza massima di 2,5 km.**

2 **Calcola la distanza massima in una rete che operi a una velocità di un 1 Gbit/s volendo una lunghezza minima del frame a 640 kB.**

UNITÀ DIDATTICA 3

TIPOLOGIE DI RETE ETHERNET

IN QUESTA UNITÀ IMPAREREMO...

- a individuare i tipi di rete Ethernet
- a conoscere la Fast Ethernet
- a conoscere la rete Gigabit Ethernet

Ethernet a 10 Mbps

Ethernet ha avuto successo sostanzialmente per due motivazioni fondamentali:
- semplicità di implementazione;
- flessibilità in quanto si presta facilmente a utilizzare nuovi mezzi fisici per le connessioni.

Ci sono quattro tipi di reti a 10 Mbps:
- 10Base2;
- 10Base5;
- 10BaseT;
- 10BaseFL.

Costituiscono quella che viene chiamata Legacy Ethernet, cioè sistemi oramai obsoleti ma che continuano a rimanere "in servizio" in quanto la loro sostituzione non può essere effettuata oppure il buon livello di stabilità di funzionamento non rende necessaria la sostituzione, che comporterebbe costi apparentemente non giustificati.

Hanno quattro caratteristiche comuni:
1. "viaggiano" tutte a 10 Mbps, dove un bit time ha la durata di 0,1 µs;
2. hanno lo stesso formato del frame;
3. adottano i *processi di trasmissione* (codifica dei dati):
   - i dati sono convertiti da hex a binario a livello 2 (MAC);
   - dal sottolivello MAC i dati sono codificati in modo Manchester;
4. utilizzano le stesse *regole di base* per la progettazione della rete:
   - 96 bit di interframe spacing;
   - 32 bit dimensione massima del jamming;

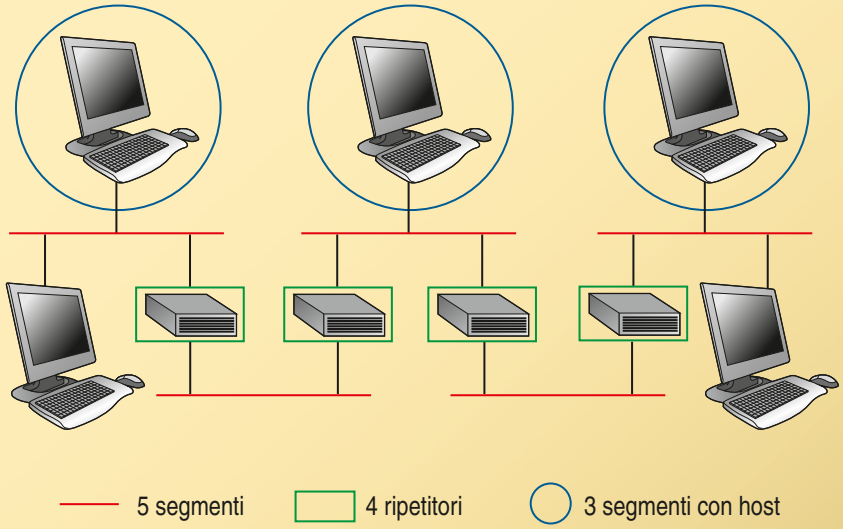
- dimensione minima del frame 512 bit;
- dimensione massima del frame 1508 bit;
- slot time di 512 bit time;

I limiti dei tempi di ritardo dipendono sostanzialmente da quattro fattori, cioè dalla lunghezza cavo e dal ritardo di propagazione sul mezzo fisico, dal ritardo introdotto dai repeater e transceiver, dal ritardo introdotto dal nodo che usa il mezzo.

Praticamente è stata introdotta una regola per rispettare i limiti sui tempi di ritardo (timing), chiamata la regola del 5-4-3:

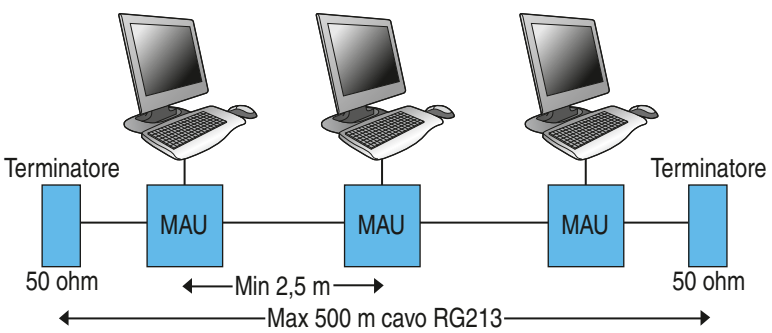
- 5: numero massimo di segmenti;
- 4: numero massimo di ripetitori;
- 3: numero massimo di segmenti popolati da host.

Quindi due segmenti possono essere utilizzati solo per collegare tra loro i tre segmenti che hanno connessi i PC.



10Base5

La prima rete Ethernet nasce nel 1980 ed è la 10Base5 che utilizza un cavo coassiale spesso RG213 (thick) con topografia a bus in modo half duplex: ha avuto una grande diffusione anche per i limitati costi dei componenti, ma oggi le schede sono difficilmente reperibili ed è stata deprecata dalle recenti normative.



Max 100 MAU (ricetrasmettitori) su un solo segmento

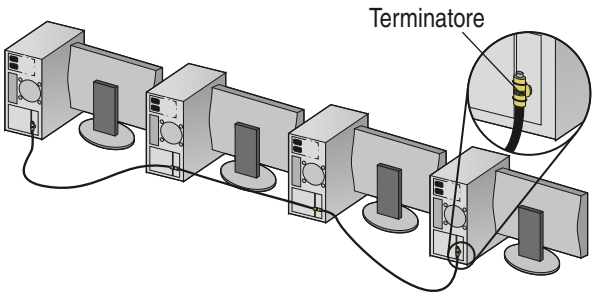
La rete, la cui lunghezza massima è di 2500 m, è organizzata in 5 segmenti di 500 m ciascuno.
La connessione 10Base5, senza la presenza di ripetitori, prevede l'uso di un cavo coassiale RG213, da 50 ohm, con una lunghezza massima di 500 m, terminante alle due estremità in una resistenza da 50 ohm.
Lungo il cavo possono essere inseriti i ricetrasmettitori, o MAU (Medium Attachment Unit), che si collegano al cavo attraverso il vampire tap (una sorta di ago che si insinua nell'anima del cavo, senza creare cortocircuiti) e a loro volta sono collegati alla scheda di rete tramite un cavo apposito.
I vari ricetrasmettitori possono essere al massimo 100 e la distanza sul cavo, tra uno qualunque di questi e il successivo, è al minimo di 2,5 m.

Riassumiamo le caratteristiche nella tabella che segue.

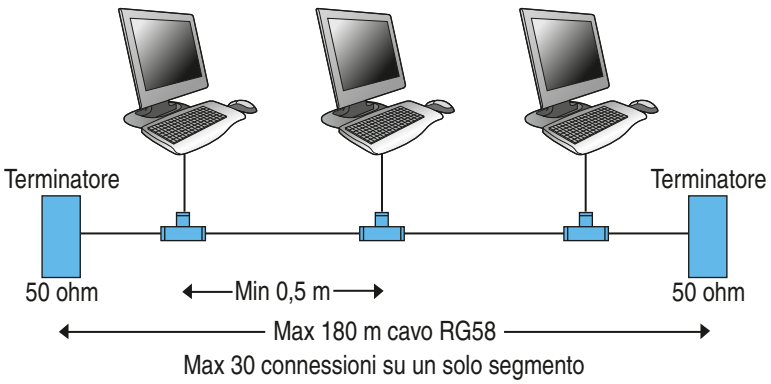
Cavo	Cavo coassiale Thicknet
Lunghezza massima del segmento	500 m
Distanza massima tra computer e trasmettitore-ricevitore	50 m
Segmenti di collegamento e ripetitori	È possibile unire 5 segmenti utilizzando 4 ripetitori, ma solo 3 dei 5 segmenti possono contenere computer (regola 5-4-3)
Computer per segmento	100 (in base alla specifica)
Lunghezza massima totale della rete	2500 m

10Base2

È stata introdotta nel 1985 e utilizza il cavo coassiale thin RG58, che è più piccolo e più leggero: anch'essa ha bassi costi ed è semplice da installare.
Ogni scheda NIC ha un connettore BNC sul quale viene montato un connettore a T per la connessione con il cavo, che agli estremi deve avere un apposito terminatore a 50 ohm.



Base2 indica la massima lunghezza di un segmento, che è circa di 185 m e, come la precedente, ha un limite massimo di 5 segmenti: su di esso possono essere connessi fino a 30 PC distanziati di almeno 0,5 m tra loro.



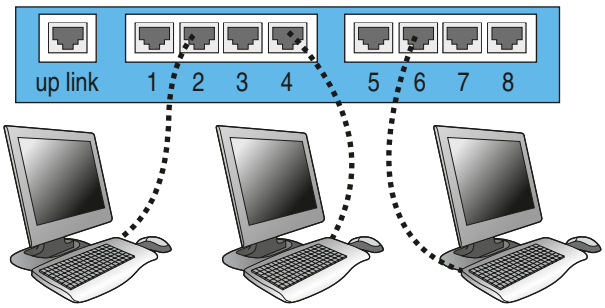
Riassumiamo le caratteristiche nella tabella che segue.

Cavo	Cavo coassiale Thicknet
Lunghezza massima del segmento	185 m
Connettori	Connettori cilindrici di tipo BNC (per estendere la lunghezza del cavo); connettori di tipo BNC a T (per collegare le schede di rete al cavo coassiale); terminatori di tipo BNC
Segmenti di collegamento e ripetitori	È possibile unire 5 segmenti utilizzando 4 ripetitori, ma solo 3 dei 5 segmenti possono contenere computer (regola 5-4-3)
Computer per segmento	30 (in base alla specifica)
Lunghezza massima totale della rete	925 m

10BaseT

Dagli anni '90 si è diffusa questa tipologia di rete che utilizza il cavo Twister UTP a partire dalla categoria 3 e con connettori RJ45 a 8 pin: a differenza delle prime due, la topologia di tipo di rete è a stella con un concentratore (◀ hub ▶) centrale.

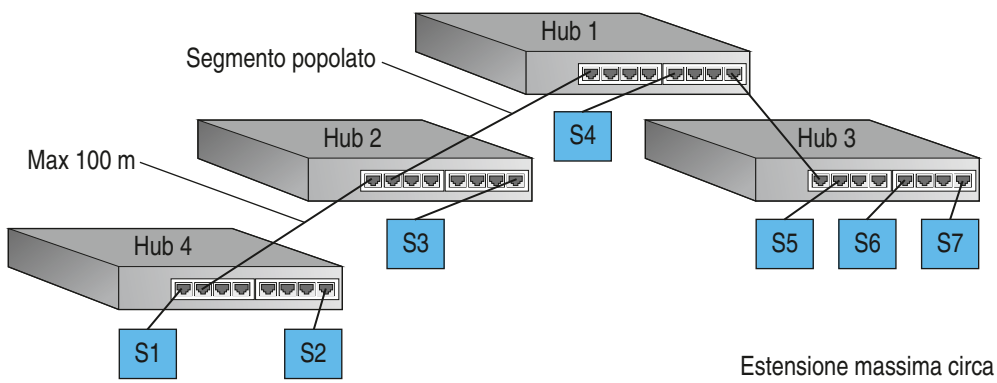
◀ L'hub è un dispositivo concentratore che opera a livello di strato fisico, amplificando e ritrasmettendo il segnale verso tutte le stazioni. ▶



Il limite del segmento è di 100 m, ma può essere superato introducendo apparati di rete come:
- ripetitori e/o hub: estendono la lunghezza del segmento senza separare i domini di collisione;
- bridge e/o switch: separano i domini di collisione.

Il limite di 100 m rimane per la lunghezza del cavo tra PC e hub oppure switch.

Vale la regola 5-4-3 e viene così interpretata:



Estensione massima circa 500 metri

È tra le reti più diffuse e oggi come cavi vengono utilizzati gli UTP cat. 5 o superiori, che offrono maggiori prestazioni e affidabilità.

Vedremo nella prossima unità didattica la differenza tra l'utilizzo di hub e switch.

Riassumiamo le caratteristiche nella tabella che segue.

Cavo	UTP di categoria 3, 4 o 5 o STP
Connettori	RJ-45 ai capi del cavo
Distanza dal computer all'hub	Massimo 100 m, ma può essere estesa utilizzando i ripetitori
Dorsale per hub	Coassiale a fibre ottiche per LAN di dimensioni maggiori
Numero totale di computer per LAN senza componenti di connettività	1024 (in base alla specifica)

Ethernet a 10BaseF

Questa tipologia di rete effettua la trasmissione dei dati a 10 Mbps utilizzando come mezzo trasmissivo un cavo in fibre ottiche multimode con fibre di vetro del tipo 62,5/125 micron.
Può raggiungere una distanza di 2000 m e generalmente viene utilizzata sia come dorsale di campus sia per la connessione di edifici che si trovano anche in località differenti.
I collegamenti con cavi a fibre ottiche sono sostanzialmente collegamenti di terminali tra due componenti elettrici attivi, quindi da un componente della rete viene condotto sempre un cavo diretto a una porta di un altro componente di rete.
Un componente di rete ha il compito di rigenerare e ripartire i segnali ricevuti emettendoli di nuovo a tutte le porte di uscita.

Il cablaggio 10BaseF è diviso in tre standard:
- 10BaseFB (Fiber Backbone);
- 10BaseFL (Fiber Link);
- 10BaseFP (Fiber Passive).

10BaseFB

Viene generalmente utilizzato per la connessione punto-punto di due ripetitori in un dorsale oppure in una struttura stellare mediante ripetitori multiporta.
Il segmento di 10Base-FB può avere una lunghezza massima di 2 km e non può connettere stazioni terminali ma solamente due ripetitori,

10BaseFL

Viene utilizzato per la connessione punto-punto tra un concentratore e una stazione periferica, e per collegare più stazioni terminali richiede necessariamente una connessione a stella.
Il segmento FL collega il concentratore al ◀ FOMAU ▶ della stazione host e può avere una lunghezza massima di 2 km.

◀ Il MAU (*Medium Attachment Unit*) della scheda di connessione presente all'interno dell'host che permette il collegamento di una fibra ottica prende anche il nome di FOMAU (*Fiber Optic MAU*). ▶

10BaseFP

Viene utilizzato per la topologia di rete stellare, costituita da vari segmenti in fibra ottica connessi a un concentratore passivo. La massima lunghezza di un segmento 10Base-FP è 500 m.

Ethernet a 100 Mbps

La rapida crescita delle reti locali e lo sviluppo continuo di applicazioni e servizi multimediali hanno portato all'esigenza di realizzare reti LAN con velocità superiori ai 10 Mb/s.

Le prime proposte per reti a 100 Mb/s furono presentate nel 1992, la prima basata sempre sul CSMA/CD e la seconda basata invece su un nuovo metodo di accesso multiplo indicato come *Demand Priority* (proposta da HP e AT&T).
Ne nacquero due diversi standard:
- Fast Ethernet o IEEE 802.3u (giugno 1995);
- 100 VG AnyLAN o IEEE 802.12.

Fast Ethernet

La proposta IEEE 802.3u o Fast Ethernet può essere vista come la naturale evoluzione della 10BaseX in quanto ne conserva tutte le caratteristiche:
- utilizza lo stesso protocollo di accesso multiplo CSMA/CD di Ethernet;
- utilizza lo stesso meccanismo di gestione delle collisioni;
- utilizza lo stesso formato del frame e la stessa lunghezza minima delle reti a 10 Mbps.

> Poiché la velocità è 10 volte maggiore, per rispettare queste caratteristiche è necessario ridurre di 10 volte il round trip delay, e di conseguenza la distanza massima, cioè la lunghezza del cavo.

Infatti abbiamo visto che nel protocollo CSMA/CD la velocità di trasmissione è legata alla lunghezza minima del pacchetto e al round trip delay: questo incide di conseguenza sulla massima distanza tra le stazioni sulla stessa rete.

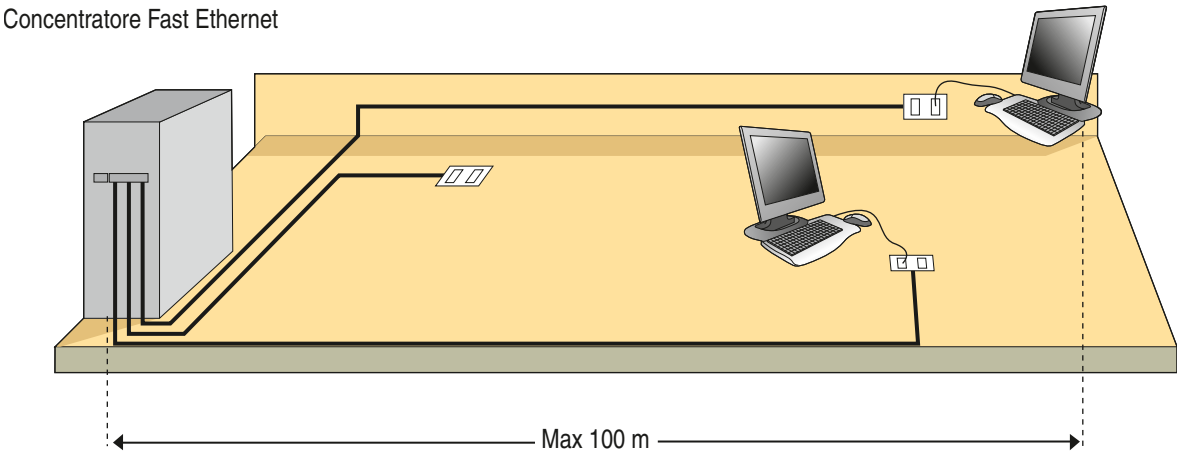
Quindi, per mantenere la compatibilità con le reti esistenti a 10 Mbps si è scelta la strada più facilmente realizzabile, cioè quella di ridurre la distanza massima teorica che, di fatto, non comporta quasi mai interventi sulle reti esistenti e permette gradualmente la migrazione verso le nuove velocità.

Il cablaggio di una rete Fast Ethernet è caratterizzato dai tre standard seguenti:
- 100Base-T4: il segnale è trasmesso su cavi Cat. 3-4 coppie;
- 100Base-TX: il segnale è trasmesso su cavi UTP Cat. 5, STP di tipo 1;
- 100Base-FX: il segnale è trasmesso su fibra ottica monomodale e multimodale.

> Dato che il segnale è trasmesso a una frequenza maggiore dei 10 Mbps, nelle prime due realizzazioni esso è più suscettibile al rumore.

Riassumiamo in una tabella le caratteristiche comuni per ogni realizzazione.

Bit rate	100 MB/s
Bit time	10 ns
Interpacket gap	0,96 µs
Durata minima del pacchetto	5,12 µs
Estensione massima della rete	circa 200 m
Cablaggio stellare con concentratore al centro e 100 m di raggio	
Il bridge o lo switch collegano i concentratori di diversi piani	



Repeater

Per estendere un segmento di una rete Fast Ethernet sono utilizzati i ripetitori (repeater). Esistono due classi di ripetitori.

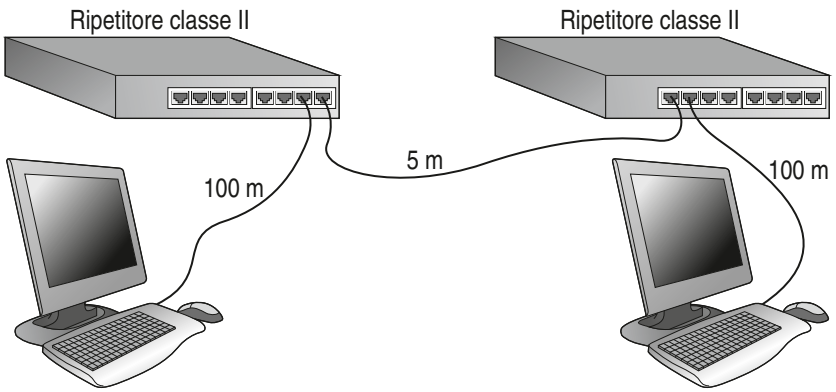
Repeater di classe I

I repeater di classe I trasformano il segnale analogico alla porta di ingresso in digitale, rigenerandolo e ritrasformandolo in analogico perché possa essere trasmesso sulla porta di uscita; questa operazione introduce un ritardo maggiore di quelli in classe II.
Vengono utilizzati tra i segmenti Fast Ethernet che impiegano tecnologie diverse, per esempio nel connettere segmenti 100Base-TX/-FX con segmenti 100Base-T4.

Repeater di classe II

I repeater di classe II introducono un ritardo più piccolo dato che amplificano il segnale ricevuto su una porta di ingresso e lo inoltrano senza alcuna trasformazione sulla porta di uscita: lo svantaggio è che questi ripetitori possono essere utilizzati solo su segmenti con le stesse caratteristiche.
Il ritardo che introducono è al massimo di 140 bit time e la massima distanza di funzionamento tra repeater di classe II è di *soli 5 metri*.

Lo standard Fast Ethernet impone le seguenti regole:
- è ammesso un solo ripetitore di classe I tra due qualunque DTE;
- sono ammessi al massimo due ripetitori di classe II tra due qualunque DTE.



100Base-T4

La trasmissione avviene in half duplex su quattro coppie di cavi bilanciati della categoria 3 (UTP), come definito dall'ISO/IEC 11801: due coppie sono utilizzate alternativamente in un senso o nel senso opposto.

> Vengono utilizzate quattro coppie di cavi in quanto la categoria 3 sopra i 25 Mhz è sensibile al rumore e quindi non sarebbe compatibile con le specifiche degli standard europei.

La codifica Manchester a 100 Mbps richiede 200 Mbaud e di fatto è impossibile utilizzarla per i doppini alle distanze richieste da questa tecnica. Si è cambiata la codifica e un ottetto di dati viene trasformato in 6 simboli ternari a 25 MHz: i gruppi di 6 simboli sono inviati separatamente su tre canali mentre il quarto canale (la quarta coppia) viene utilizzato per rilevare le collisioni e trasmetterle nella direzione opposta del segnale.

100Base-TX

Il segmento 100Base-TX utilizza il doppino telefonico UTP o STP dove una coppia è utilizzata per trasmettere e l'altra per ricevere, come nel 10BaseT.

Viene utilizzata una codifica a due livelli detta 4B/5B:

- ogni gruppo di cinque periodi di clock contiene 32 combinazioni, di cui 16 sono utilizzate per trasmettere 4 bit di dati e alcune delle altre per funzioni di controllo: le 16 combinazioni dedicate ai dati sono state scelte opportunamente per garantire un adeguato numero di transizioni del segnale allo scopo di facilitare la sincronizzazione in ricezione;
- trasmettendo 4 bit ogni cinque periodi di clock a 125 MHz si ottengono i 100 Mbps desiderati.

Nel caso di utilizzo di cavi UTP le due coppie sono di categoria 5, capace di supportare una frequenza di 125 MHz, mentre nel caso di cavi schermati si utilizzano i cavi STP di tipo 1 con impedenza caratteristica di 150 Ω.

100Base-FX

La trasmissione avviene in full duplex su due canali in fibra ottica multimodale con segmenti di lunghezza massima 412 m: come per la 100Base-TX si utilizza la codifica 4B/5B a 125 MHz, ma convertita in segnale ottico.
Se viene utilizzata una coppia di ripetitori di classe 2 la distanza si riduce a 320 m: è quindi necessario rivedere le massime distanze per ogni caso specifico, a seconda del numero e del tipo di ripetitori usati nel link (con ripetitori di classe 1 la distanza massima scende ulteriormente a 272 m).

Migrare da 10 a 100 Mbps

Lo standard IEEE 802.3u è stato appositamente definito con le stesse regole del protocollo Ethernet per poter far coesistere i vecchi segmenti a 10 Mbps presenti nelle LAN esistenti con i nuovi segmenti a 100 Mbps.
È necessario quindi avere switch con porte sia da 10 sia da 100 Mbps così da poter connettere segmenti con velocità diverse ottenendo LAN a velocità miste.
Questa possibilità permette alle aziende di pianificare una migrazione della rete da Ethernet a Fast Ethernet senza dover cambiare contemporaneamente tutti gli apparati di commutazione e le interfacce.

Nella figura a lato è riportato uno switch a 26 porte 10/100.

In questo modello ogni porta può essere connessa sia a 10 Mbps che a 100 Mbps: la velocità della porta viene *negoziata* dalle due interfacce all'atto dell'accensione degli host, così come la modalità di trasmissione (half duplex o full duplex).

Gli switch possono capire da soli in quale modalità devono operare e sono in grado di configurare automaticamente la porta in modo opportuno.

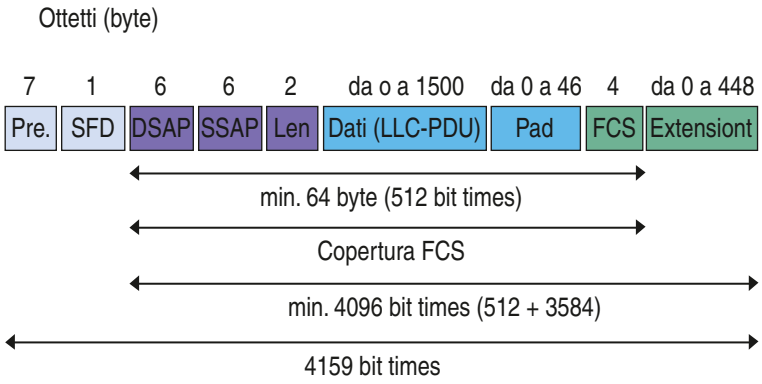
Ethernet a 1 e 10 Gigabit

La rete Gigabit Ethernet nacque nel novembre 1995 quando Compaq propose al comitato IEEE 802 l'architettura base di una rete Ethernet a 1 Gbit/s: IEEE formò il gruppo IEEE 802.3z con lo scopo di definire uno standard per questa rete.

Gigabit Ethernet opera a una velocità di 1 Gbit/s ed è compatibile con Ethernet e Fast Ethernet mantenendo lo stesso formato e la stessa ampiezza dei loro pacchetti: in questo modo si ha il vantaggio di mantenere in servizio le LAN Ethernet e Fast Ethernet e avviare il processo di migrazione in modo graduale e senza costi eccessivi.

Gigabit Ethernet usa il protocollo CSMA/CD come Ethernet e Fast Ethernet e quindi ha la medesima dimensione minima del pacchetto: di conseguenza la durata minima della trasmissione si deve ridurre di 10 volte, portando di fatto l'estensione massima della rete a soli 20 m! Per poter mantenere una lunghezza di 200 m viene quindi necessariamente allungata la dimensione minima del pacchetto.

Per aumentare la lunghezza del pacchetto più corto viene modificata la struttura del frame Ethernet aggiungendo un ulteriore campo, chiamato appunto extension, di dimensione massima 448 byte (3584 bit):



Quando si è in presenza di un pacchetto di lunghezza minima che può "viaggiare" sulle altre tipologie Ethernet, cioè 512 bit, si aggiungono 3584 bit per ottenere i 4096 che sommati al preambolo compongono un frame che occupa 4159 bit time, il tempo utile per rilevare la collisione.
Ricapitoliamo in una tabella i parametri usati dalle tre diverse tipologie di Ethernet:

	Ethernet	Fast Ethernet	Gigabit Ethernet
bit rate	10 Mb/s	100 Mb/s	1 Gb/s
bit time	100 ns	10 ns	1 ns
inter-packet gap	9,6 μs	0,96 μs	96 ns
slot time	51,2 μs	5,12 μs	4,096 μs

Le Gigabit Ethernet utilizzano mezzi sia in rame che in fibra e sono impiegate sia per connessioni ad alta velocità tra infrastrutture di rete sia per l'installazione di dorsali (backbone).
A livello fisico vengono utilizzati due livelli di codifica (chiamata 8B/10B) e il flusso di bit viene convertito dal livello MAC in simboli di dati e simboli di controllo per garantire la trasmissione di un numero di transizioni sufficienti a sincronizzare il ricevitore.

Conduttori di rame

Il segnale è trasmesso a una frequenza maggiore di 100 Mbps ed è quindi ancora più suscettibile al rumore. Sono possibili due diverse tecnologie:

- 1000Base-CX: utilizza un cavo a due coppie di 150 ohm su distanze massime di 25 m;
- 1000Base-TX: utilizza un doppino telefonico UTP a 4 coppie di categoria 5, revisionato in un nuova categoria 5e, alla quale si sono successivamente aggiunte le categorie 6 e 7. La massima distanza raggiungibile è 100 m.

Conduttori in fibra

Sono possibili due diverse tecnologie:

- 1000Base-SX: per il cablaggio utilizza fibre multimodali che operano tra 770-860 nm (normalmente definita come 850 nm) e per il segnale un laser del tipo SWL (Short Wavelength Laser con LED a 850 nm). Le massime distanze che possono essere raggiunte dipendono dal tipo di fibra (diametro della fibra, banda per km ecc.): si passa da 500 m per le fibre 50/125 a 220 m per quelle 62,5/125;
- 1000Base-LX: utilizza per il cablaggio fibre che operano tra 1270 e 1355 nm (normalmente definite come 1350 nm) che possono essere sia multimodali che monomodali e per il segnale un laser del tipo LWL (Long Wavelength Laser con LED a 1300 nm).

Le massime distanze per la connessione con fibre ottiche sono riportate nella seguente tabella:

Standard	Tipo fibra	Diametro fibra (μM)	Banda (MHZ * KM)	Massima distanza (M)
1000 Base SX	Multimodale	62,5	160	220
	Multimodale	62,5	200	275
	Multimodale	50	400	500
	Multimodale	50	500	550
1000 Base LX	Multimodale	62,5	500	550
	Multimodale	50	400	550
	Multimodale	50	500	550
	Multimodale	9	NA	5000

Esistono anche soluzioni "proprietarie" che permettono di andare ben al di là della distanza massima di 5 km prevista dallo standard: usando laser a 1550 nm su fibra SMF si può arrivare fino a circa 150 km (senza amplificazione ottica né compensazione di dispersione).

Verifichiamo le conoscenze

Esercizi a risposta multipla

1 Ethernet ha avuto successo soprattutto per:

- ○ semplicità di implementazione
- ○ velocità
- ○ flessibilità
- ○ sicurezza

2 Ci sono quattro tipi reti a 10 Mps (*indica la risposta errata*):

- ○ 10Base2
- ○ 10Base5
- ○ 10BaseT
- ○ 10BaseTL

3 I limiti dei tempi di ritardo dipendono da:

- ○ lunghezza cavo
- ○ ritardo di propagazione del cavo
- ○ ritardo introdotto da repeater e transceiver
- ○ ritardo introdotto dal nodo che usa il mezzo
- ○ ritardo della scheda NIC

4 La regola 5-4-3 indica:

- ○ 5 segmenti
- ○ 5 ripetitori
- ○ 4 ripetitori
- ○ 4 segmenti

5 Il cablaggio 10BaseF è diviso in tre standard (*indicare la risposta errata*):

- ○ 10BaseFB (Fiber Backbone)
- ○ 10BaseFL (Fiber Link)
- ○ 10BaseFX (Fiber Extended)
- ○ 10BaseFP (Fiber Passive)

6 Il cablaggio di una rete Fast Ethernet è caratterizzato da tre standard (*indicare la risposta errata*):

- ○ 100Base-T4
- ○ 100Base-TX
- ○ 100Base-FX
- ○ 100Base-CX

7 Indica quale affermazione per la Fast Ethernet è errata:

- ○ utilizza lo stesso protocollo di accesso multiplo CSMA/CD di Ethernet
- ○ utilizza lo stesso meccanismo di gestione delle collisioni 10 Mbps
- ○ ha la round trip delay delle reti a 10 Mbps
- ○ ha lo stesso formato del frame del 10 Mbps

8 Indica quale affermazione per i repeater 100 Mbps è errata:

- ○ è ammesso un solo ripetitore di classe I tra due qualunque DTE
- ○ sono ammessi al massimo due ripetitori di classe II tra due qualunque DTE
- ○ la massima distanza tra due repeater di classe II è di 5 m
- ○ nel 100Base-FX una coppia di ripetitori di classe I riduce la distanza a 320 m

Test vero/falso

1 Nella Ethernet a 10 Mbps i dati sono convertiti da hex a binario a livello 2 (MAC). V F
2 Nella Ethernet a 10 Mbps possono esserci più di 3 segmenti con host. V F
3 Nella Ethernet a 10 Mbps possono esserci al massimo 4 ripetitori. V F
4 Base2 indica la lunghezza massima di un segmento che è circa di 205 m. V F
5 In una rete 10BaseT il limite del segmento è di 100 m. V F
6 In una rete 10BaseT il limite di 100 m rimane per la lunghezza del cavo tra PC e hub. V F
7 In una rete 10BaseT con cavo STP categoria 3 si utilizzano i connettori RJ45 a 8 pin. V F
8 Con FOMAU si intende il connettore della scheda NIC che permette il collegamento di una fibra ottica. V F
9 Le prime proposte per reti a 100 Mb/s furono presentate nel 1989. V F
10 La proposta IEEE 802.3u è anche chiamata Fast Ethernet. V F
11 Il segnale a 100 Mbps ha una tolleranza maggiore al rumore di quello a 10 Mbps. V F
12 I repeater di classe I introducono un ritardo minore di quelli in classe II. V F

Esercizi di completamento

1 Con il termine ..... si intendono quei sistemi ormai obsoleti ma che continuano a rimanere "in servizio" in quanto la loro sostituzione ..... oppure il buon livello di ..... non rende necessario sostituirli perché prevedrebbe ..... non giustificati.

2 Il limite del segmento di 100 m nelle 10BaseT può essere superato introducendo apparati di rete come:
- .....: estendono la lunghezza del segmento senza separare i domini di collisione;
- .....: separano i domini di collisione.

3 Nella Fast Ethernet poiché la velocità è ..... rispetto alla 10BaseT, per rispettare le stesse te caratteristiche è ..... il round trip delay, e di conseguenza .....

4 Lo standard Fast Ethernet impone le seguenti regole per i repeater:
- è ammesso ..... di classe I tra due .....;
- sono ammessi al massimo ..... di classe 2 tra .....

5 Gli ..... possono capire da soli in che modalità devono operare e sono in grado di configurare automaticamente ..... in modo opportuno.

6 Gigabit Ethernet ha quindi la medesima dimensione minima del pacchetto di .....: di conseguenza la durata minima della trasmissione si deve ....., portando di fatto l'estensione massima della rete a soli 20 m: per poter mantenere una lunghezza di 200 m viene quindi necessariamente ..... del pacchetto.

UNITÀ DIDATTICA 4

DISPOSITIVI DI RETE A LIVELLO 2

IN QUESTA UNITÀ IMPAREREMO...

- a conoscere i dispositivi di rete a livello 2
- a individuare la differenza tra hub e bridge
- ad apprendere il concetto di dominio di collisione

Premessa

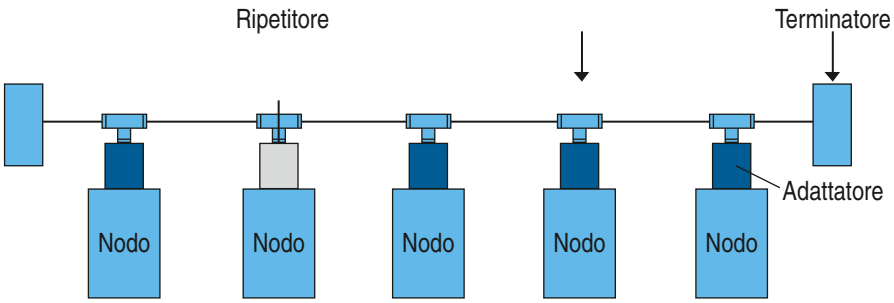
Negli ultimi anni Ethernet si è fortemente evoluta, spinta dalla tecnologia e dalle esigenze sempre più diffuse di connessione tra dispositivi, sia fissi che mobili.
Le prime reti Ethernet degli anni '80 erano realizzate con la connessione di pochi elaboratori, generalmente nello stesso ufficio, utilizzando un singolo pezzo di cavo coassiale ai cui estremi, per chiudere il circuito, vi erano i terminatori di 50 ohm connessi a massa, sul quale erano presenti i connettori a "T" per connettere i PC.
Successivamente la rete veniva connessa a un altro "pezzo" di rete simile mediante un ripetitore: era possibile collegare un massimo di cinque segmenti di rete, ma solo tre di essi potevano avere host connessi mentre gli altri due potevano essere utilizzati solamente per collegare tra loro i ripetitori (segmenti di collegamento).

Zoom su...

REPEATER

Il ripetitore (repeater) è un apparato che opera unicamente al livello 1 del modello di riferimento OSI il quale riceve, amplifica e sincronizza i pacchetti ricevuti da un segmento e li ritrasmette sull'altro: esso viene impiegato per prolungare un segmento di rete oltre i limiti fisici del tipo di medium utilizzato.

Nel decennio scorso i ripetitori erano costosi e, inoltre, prima di ricorrere al loro utilizzo, si cercava di aggiungere host sullo stesso segmento fino al limite estremo di funzionamento: all'aumentare del



numero di nodi connessi sullo stesso segmento aumenta la probabilità di collisione e, di conseguenza, aumenta la percentuale di trame ritrasmesse, con conseguente diminuzione dell'efficienza della rete.

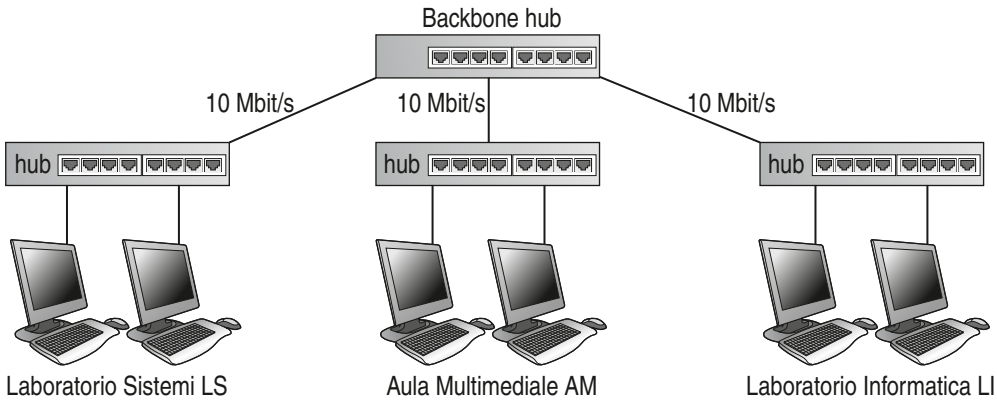
Per migliorare le prestazioni la soluzione è stata quella di "spezzare" la rete in segmenti aventi ciascuno un proprio dominio di collisione: l'avvento del doppino e la conseguente diminuzione dei costi dei dispositivi hanno portato all'introduzione dei bridge.

Avvicinamento al bridging

Nella topologia 10BaseT si utilizzano doppini intrecciati e dispositivi concentratori per realizzare la struttura "a stella" tipica di tale tecnologia.

Hub

I primi dispositivi utilizzati nel corso degli anni come concentratori sono stati gli hub: sono sostanzialmente dei "ripetitori di bit" che riproducono i bit in ingresso a un'interfaccia su tutte le altre interfacce.
La distanza massima tra un nodo e un hub è 100 m e, nel caso di malfunzionamento di un adattatore, l'hub è in grado di rilevarlo e di disconnetterlo dalla rete.
Possiamo organizzare la rete in modo gerarchico ponendo un hub a livello più alto (backbone hub) al quale connettere i singoli hub, come nel seguente esempio, dove ogni singola LAN collegata può essere considerata un segmento di rete.



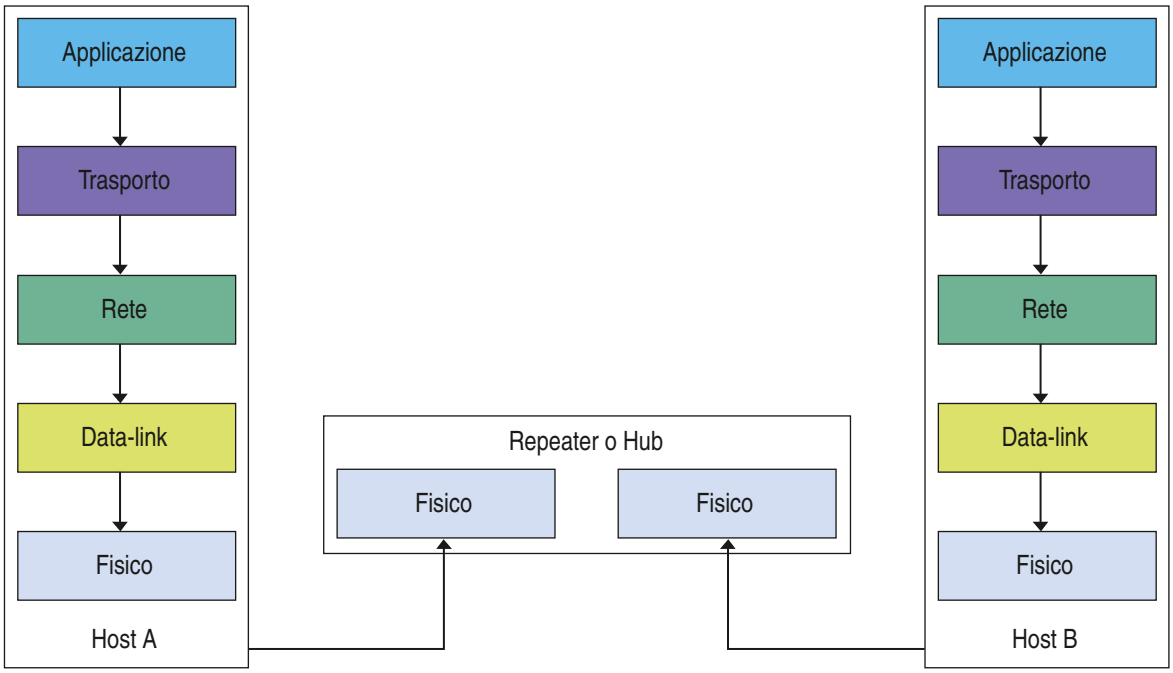
Il principale svantaggio degli hub è quello di non isolare i domini di collisione: le stazioni possono subire una collisione per una trasmissione simultanea con qualunque stazione presente su qualunque segmento.

Quindi la creazione di una rete multilivello a hub non comporta alcun aumento del throughput massimo, essendo un unico dominio di collisione e quindi identico a quello di una rete con un unico segmento; anche il numero massimo degli host rimane lo stesso.

Per contro, tra i vantaggi degli hub ricordiamo il costo e la possibilità di creare reti con buona tolleranza ai guasti introducendo diversi livelli nella struttura gerarchica, in modo da creare segmenti comprendenti pochi host o introducendo ridondanze che permettono il funzionamento della rete anche in presenza di rotture di uno o più hub stessi.
Possono inoltre "raccogliere dati" utili per realizzare statistiche e quindi dare all'amministratore preziose indicazioni per effettuare il monitoraggio della rete.

Nella tecnologia Gigabit Ethernet gli hub prendono il nome di buffered distributors.

Nella pila ISO/OSI gli hub si collocano al livello 1, così come i ripetitori di segnale.



Bridge

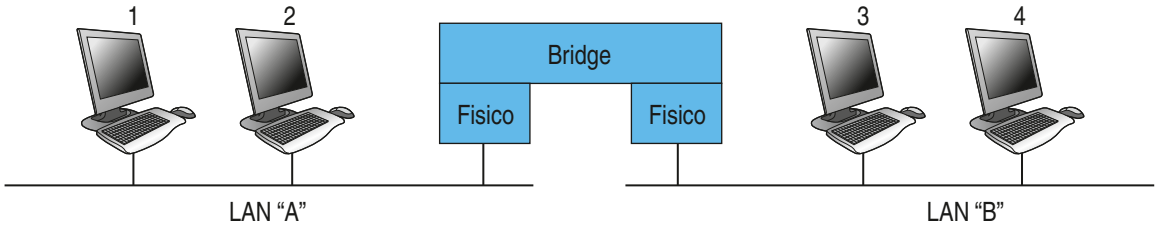
Successivamente si è passati a utilizzare dispositivi più complessi, i bridge, i quali permettono di spezzare i domini di collisione. Essi hanno un meccanismo di funzionamento diverso dagli hub che si limitano alla sola ritrasmissione del bit: i bridge ricevono il frame e lo memorizzano in appositi buffer, quindi ne leggono le intestazioni, individuando l'host di destinazione, e selezionano la ritrasmissione solo sulla linea interessata (link di uscita).
Essi applicano quindi un filtro sui pacchetti, bloccando i frame che sono destinati allo stesso segmento e inoltrando solo quelli destinati a host presenti su altri segmenti della LAN. Anch'essi utilizzano il protocollo CSMA/CD sul segmento LAN di uscita verificando lo stato della linea prima di trasmettere, in quanto l'host destinatario potrebbe avere iniziato a sua volta una trasmissione.

Questo meccanismo prende anche il nome di store and forward.

Tra i principali vantaggi dei bridge ricordiamo che isolano i domini di collisione, determinando un aumento complessivo del throughput massimo, non introducono limitazioni sul numero massimo delle stazioni, né sull'estensione geografica e non richiedono alcuna modifica per la loro installazione negli adattatori dei computer.

Una rete LAN nella quale sono presenti i bridge prende anche il nome di BLAN.

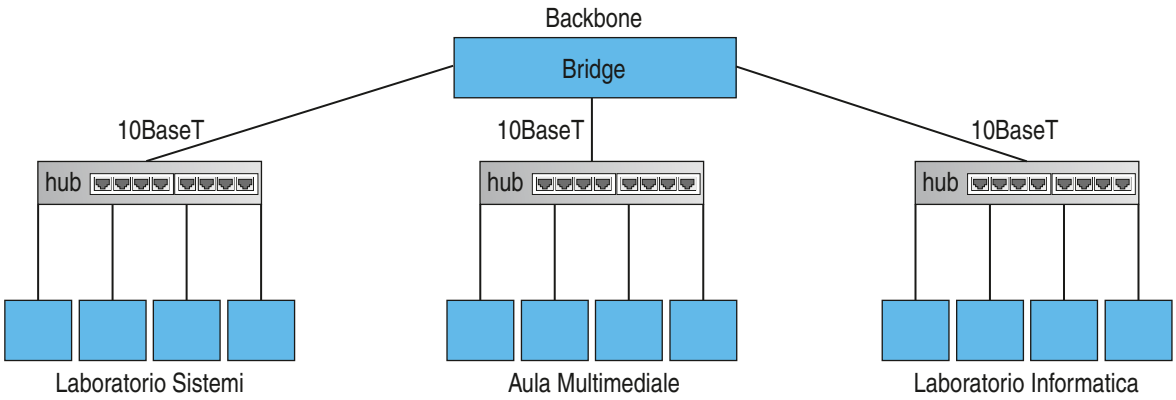
Il bridge si colloca nella pila di riferimento OSI al livello 2, sopra il livello fisico, a livello di data link: l'introduzione di un bridge, come si può vedere nella figura seguente, separa una LAN in due "sotto reti", generando di fatto una LAN A e una LAN B.



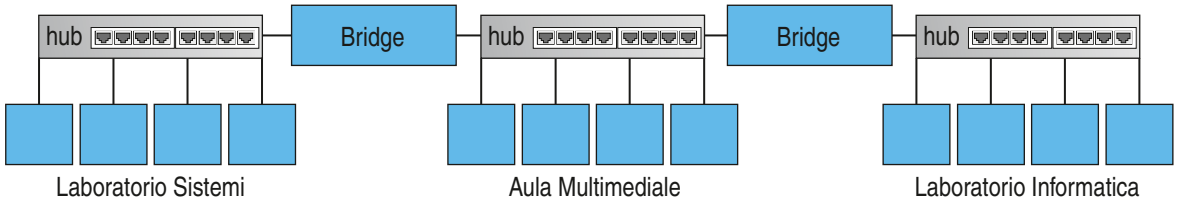
Inoltre, proprio grazie al meccanismo di funzionamento basato sullo store and forward, i bridge possono essere utilizzati anche in presenza di differenti tecnologie sulla stessa rete.

Nella figura seguente è riportato uno switch dove sono ben visibili i connettori sia per la tecnologia coassiale sia per quella twisted.

Riprendendo l'esempio precedente, se ora sostituiamo al livello gerarchico superiore il bridge otteniamo la seguente configurazione:



È anche possibile utilizzare il bridge direttamente all'interno del segmento, quindi scomponendolo di fatto in due o più segmenti, come riportato nell'esempio a pagina seguente:



Questa soluzione è però sconsigliata in quanto l'hub dell'aula multimediale diventerebbe un punto critico: la sua rottura isolerebbe i due laboratori.
Inoltre il traffico da e per i due laboratori dovrebbe necessariamente passare per l'hub dell'aula multimediale generando traffico inutile su quel segmento.

Bridge filtering

Solitamente un bridge ha solo due porte e divide il dominio di collisione in due parti; per effettuare la trasmissione o meno dei frame tra i due segmenti crea al suo interno una tabella eseguendo un algoritmo di autoapprendimento che individua e classifica le NIC collegate alle due porte: la tabella prende anche il nome di filtering table.

MAC address	Porta	Timestamp

La tabella è molto semplice ed è composta da tre campi: il bridge, da ogni frame che lo raggiunge, apprende il numero della porta sulla quale è connesso e l'indirizzo dell'host mittente, e lo colloca nella tabella aggiungendo il timestamp; ogni riga vi rimane per circa 5 minuti (ageing time) dopo di che viene rimossa.

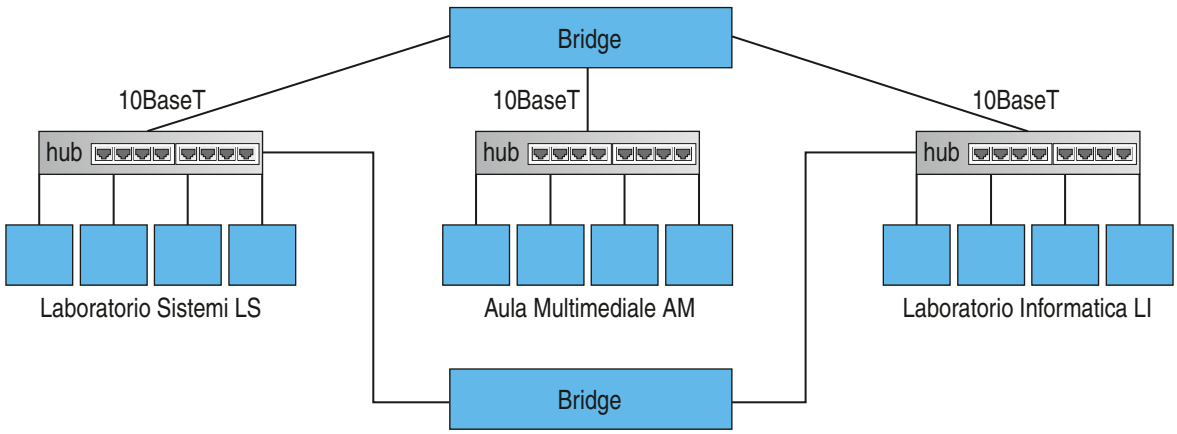
Alla prima trasmissione, cioè a tabella vuota, il bridge non conosce il segmento sul quale è presente il destinatario, ma lo spedisce ugualmente nel segmento opposto da dove è arrivato, infatti:
- se è il segmento giusto, riceverà la risposta e aggiornerà la tabella con questo indirizzo;
- se non riceve alcuna risposta, poco importa, perché il frame ha già raggiunto la destinazione sul segmento di partenza e da quel segmento arriverà la risposta e potrà scrivere la riga corrispondente nella tabella.

Quindi alla fine entrambi i PC saranno presenti nella tabella e alla successiva trasmissione il bridge saprà come comportarsi, cioè saprà se rispedire il frame oppure no: quest'operazione prende il nome di filtering.
Questo algoritmo prende anche il nome di routing isolato o backward learning.

FILTRAZIONE E INOLTRO

Con il termine filtrazione (filtering) si intende la capacità del bridge di decidere o meno se un frame deve essere inoltrato a qualche interfaccia.
L'inoltro (forwarding) è invece la capacità di determinare su quale porta del bridge bisogna inviare il frame ricevuto.

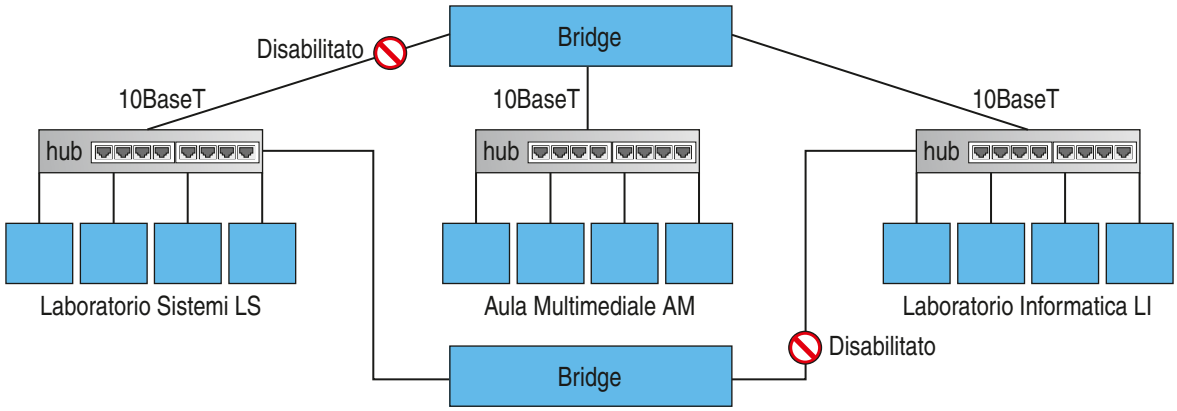
Per migliorare l'affidabilità della rete spesso si duplicano alcuni collegamenti introducendo bridge aggiuntivi: in questo modo si creano delle ridondanze di percorsi che potrebbero generare anche effetti indesiderati, come la duplicazione delle trame.
Per esempio, aggiungiamo un secondo bridge al caso precedentemente descritto:



Un frame inviato da LS verso LI raggiungerebbe in questo modo entrambi i bridge e verrebbe inoltrato due volte: si creerebbero delle tabelle con duplicazione di righe aventi lo stesso indirizzo ma porte diverse.
Per ovviare a questi problemi è necessario utilizzare un particolare algoritmo, lo spanning tree, che trasforma una situazione di rete a maglia in una topologia ad albero eleggendo un bridge come root, così da escludere la presenza di percorsi chiusi che genererebbero percorsi alternativi.

Questo algoritmo viene eseguito all'accensione dei bridge in modo da individuare subito la presenza di percorsi da isolare: la disconnessione viene fatta soltanto a livello logico bloccando temporaneamente alcune porte di qualche bridge.
Dato che una rete non è statica, un bridge può accorgersi che la sua topologia è stata modificata e richiedere la nuova esecuzione dell'algoritmo di spanning tree.

L'esecuzione dell'algoritmo sulla rete del nostro esempio potrebbe portare alla disabilitazione di due connessioni: alla successiva accensione dei dispositivi attivi potrebbe essere eletto come root il secondo bridge e operare in modo da trasmettere su entrambi i collegamenti disabilitati, isolando, come nell'esempio seguente, le porte di quelli abilitati.



Zoom su...

PROCEDURA DI FILTERING E LEARNING

Una pseudocodifica della procedura di filtering è la seguente:

```
if (destinazione frame coincide con LAN di partenza)
  then (ignora il frame)
  else {
        ricerca nella filtering table
        if (è presente il destinatario)
          then (invia il frame sulla porta indicata)
          else (invia il frame su tutte le porte tranne che da quella ricevuta)
        }
```

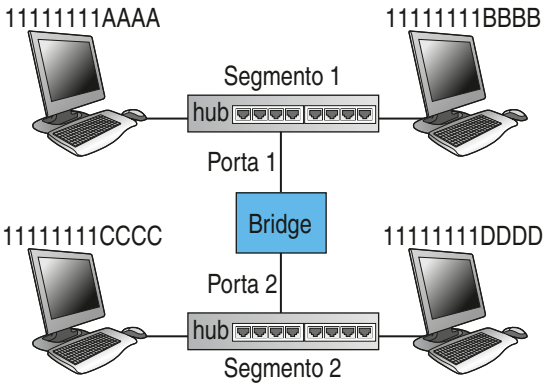
Una pseudocodifica della procedura di learning è la seguente:

```
if (indirizzo MAC mittente è presente nella tabella)
  then (ignora il frame)
      if (nr. porta di ricezione = nr porta in tabella)
        then (aggiorna il timestamp)
        else (aggiorna in tabella nr. porta, timestamp)
  else (aggiunge in tabella indirizzo MAC, nr. porta, timestamp)
```

Le due procedure sono eseguite a ogni frame che il bridge riceve.

ESEMPIO 3

Vediamo un esempio di come funziona l'algoritmo di learning sulla seguente rete composta da due segmenti, inizialmente con la seguente tabella dov'è presente solo una riga:



MAC address	Porta
11111111AAAA	1

Supponiamo ora che B invii un frame a D, e D produca un frame di risposta a B.

B invia il frame a D:

MAC address	Porta
11111111AAAA	1
11111111BBBB	1

Il bridge analizza la tabella e non trova nessun D, pertanto lo invia in flooding su tutte le sue porte tranne la 1 dalla quale lo ha ricevuto: contemporaneamente aggiunge il MAC del B nella sua tabella, dato che non è presente, e lo riconosce come appartenente al segmento connesso sulla porta 1.

Il frame viene ricevuto da D tramite la porta 2.

D invia il frame di risposta a B:

MAC address	Porta
11111111AAAA	1
11111111BBBB	1
11111111DDDD	2

Il bridge riceve il frame proveniente da D sulla porta 2 e, dato che non lo trova nella sua tabella, lo aggiunge.

Dalla tabella individua la porta dov'è connesso B e sulla quale deve trasmettere il frame: invia solo sulla porta 1.

Oltre al bridge esiste un altro dispositivo di tipo store and forward che compie "più o meno" le stesse attività: il router. Ha molti pregi e alcuni difetti rispetto al bridge e inoltre compie un insieme di operazioni aggiuntive che sono legate al livello 3, quindi verrà analizzato successivamente.

Switch Ethernet

Da alcuni anni è disponibile sul mercato un nuovo dispositivo che di fatto tende a sostituire sia l'hub che il bridge: questo dispositivo prende il nome di switch (commutatore).

In sostanza si tratta di un bridge ad alte prestazioni e dotato di un numero elevato di interfacce.

◀ La *Content Access Memory* (CAM) è una memoria associativa in cui, specificando come selettore l'indirizzo di destinazione, viene restituita l'informazione relativa alla porta su cui effettuare la trasmissione. Viene utilizzata con un apposito circuito integrato programmabile (ASIC, *Application Specific Integrated Circuit*) che svolge funzioni logiche ad alta velocità e passa i frame da una porta all'altra velocemente. ▶

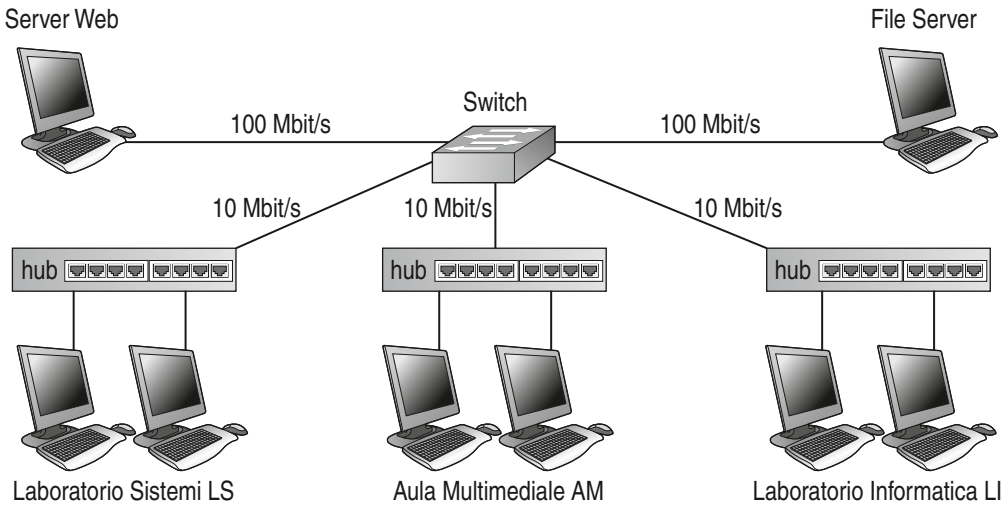
Essendo sostanzialmente un bridge, lo switch compie le stesse operazioni sopra descritte, cioè inoltra e filtra i frame usando l'indirizzo LAN di destinazione e costruisce in automatico la tabella degli indirizzi.

Le caratteristiche degli switch sono le seguenti:

1. sono dotati di un numero di interfacce elevato (nei bridge sono poche, 3 o 4 al massimo);
2. utilizzano memorie ad accesso veloce (◀ CAM ▶) per effettuare lo smistamento del frame;
3. garantiscono la banda per ogni stazione collegata alla porta;
4. permettono la trasmissione full duplex, cioè sulla stessa interfaccia possono essere spediti e ricevuti frame contemporaneamente.

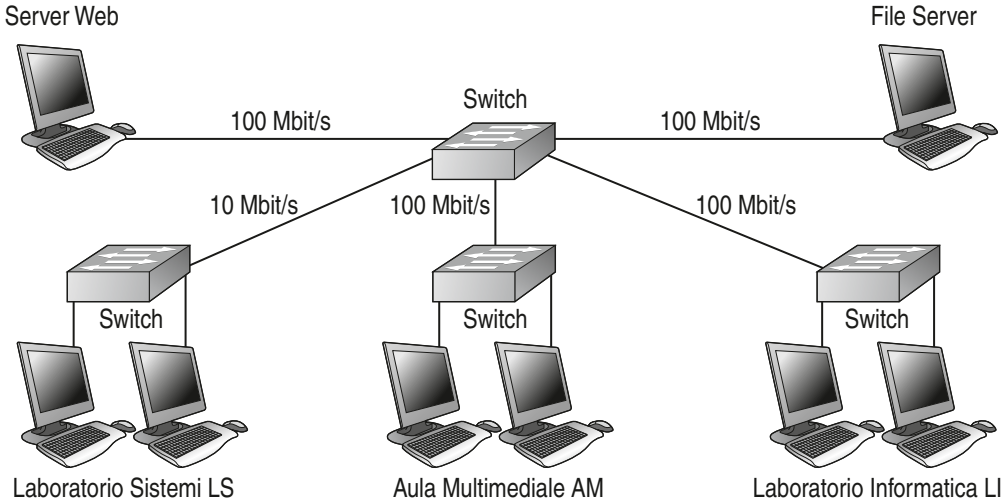
Gli switch segmentano la rete creando domini di collisione separati, garantendo la banda su ogni segmento.

Vediamo uno schema che ci permette meglio di capire i vantaggi dell'utilizzo di uno switch:

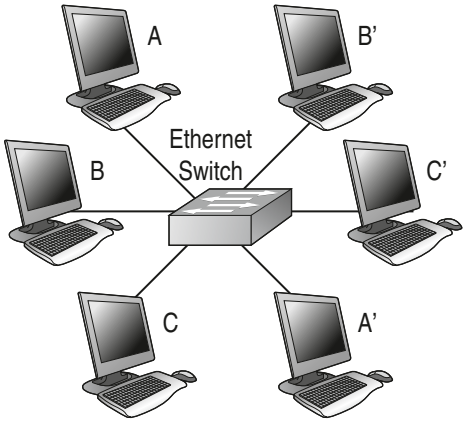


In questo caso avremo cinque segmenti (quindi cinque domini di collisione separati) e su ognuno di essi verrà garantita la banda completa: dato che nel livello inferiore ogni segmento utilizza un hub, la banda effettiva dipenderà dal numero di stazioni collegate all'hub stesso: per esempio, se nel laboratorio informatica ci sono 20 host la banda su quel segmento diviene (10 Mbit/s / 20).
I calcolatori che dovranno gestire grossi carichi di lavoro potranno essere collegati direttamente sulla porta dello switch e avranno tutta la banda dell'interfaccia a disposizione (come i due server nel nostro caso).

Un'evoluzione di quest'architettura potrebbe prevedere l'utilizzo di una rete completamente switched, come la seguente:



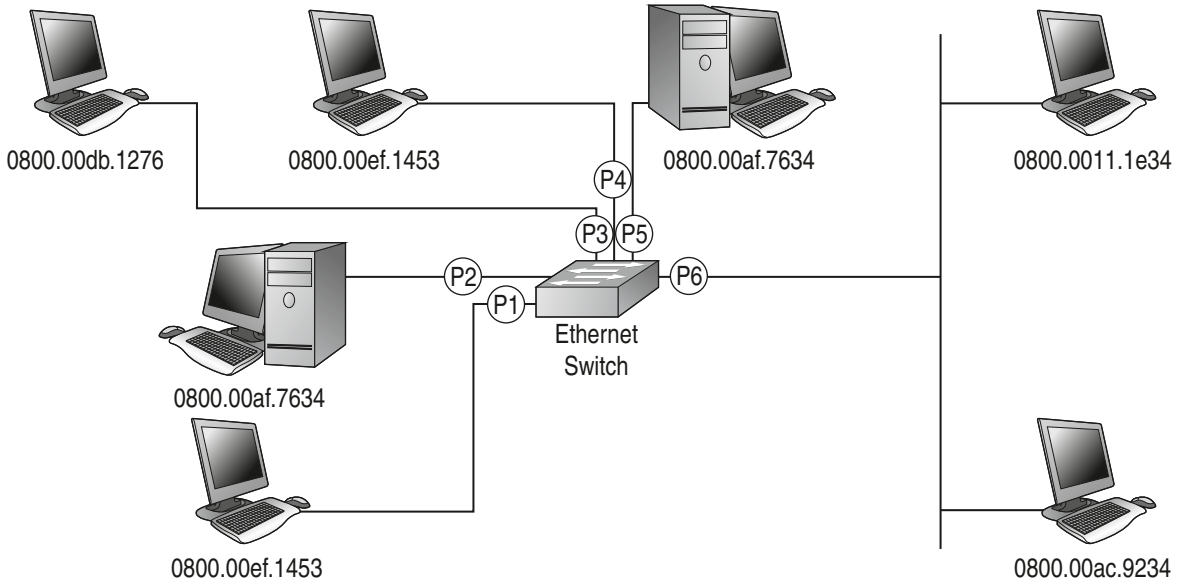
In questo caso si potrebbe utilizzare un unico dispositivo con un alto numero di interfacce per collegare singolarmente tutti gli host: la topologia diviene una stella con al centro lo switch, come schematizzato nella figura nella pagina seguente.



Ogni coppia di host viene direttamente connessa, quindi viene realizzata la trasmissione da A a B e da A' a B' simultaneamente: siamo ora in assenza di collisioni, pur utilizzando sempre il protocollo Ethernet!

ESEMPIO 4

Costruire la tabella di filtering per la rete riportata nella figura seguente:



Sulle porte P1-P5 è presente un solo host e solo sulla porta 6 abbiamo due host. La tabella è quindi quella a lato:

MAC Address	Porta
0800.00ab.1234	1
0800.00cd.5234	2
0800.00db.1276	3
0800.00ef.1453	4
0800.00af.7634	5
0800.0011.1e34	6
0800.00ac.9234	6

Il segmento fisico che collega una porta dello switch a un host è detto microsegmento: dato che l'UTP usa una coppia di fili per la trasmissione e un'altra per la ricezione, è possibile che i segnali passino contemporaneamente nelle due direzioni.
Questa possibilità di comunicazione è detta full duplex e teoricamente si raddoppia la banda dato che avviene la spedizione simultanea in entrambe le direzioni.

Nella figura a lato è riportato uno switch a 24 porte.

Uno switch ha sostanzialmente tre possibili modalità di funzionamento:

- cut through: viene iniziata la trasmissione del frame sulla porta di destinazione non appena riceve l'indirizzo MAC di destinazione: in questo caso non viene fatto nessun controllo di errore anche se il sistema di connessione è più veloce, ma sorgente e destinazione devono avere la stessa velocità di funzionamento (switching sincrono);
- store and forward: viene attesa la ricezione dell'intero frame prima di inviarlo sulla porta di destinazione (switching asincrono); in questo modo è possibile effettuare il controllo dell'FCS (*Frame Check Sum*) e quindi garantire che il frame sia corretto scartando i frame danneggiati; con questo meccanismo è anche possibile effettuare la trasmissione tra host che lavorano a velocità diversa;
- fragment free: è un compromesso tra i due metodi precedenti, in quanto legge solo i primi 64 byte prima di iniziare la trasmissione.

Osservazioni sul dominio di collisione

L'evoluzione che ha avuto Ethernet negli ultimi decenni ha portato a una completa trasformazione delle reti attuali rispetto a quelle degli anni '80: molti componenti sono, di fatto, "scomparsi", ne sono stati aggiunti di nuovi, anche la terminologia è cambiata e a volte si è trasformata.
Un elemento però ha sempre mantenuto la stessa importanza nel corso della sua evoluzione: è il concetto di segmento che, anche se nel corso degli anni ha ampliato e modificato il proprio significato, resta alla base dell'individuazione dei domini di collisione.

I segmenti nelle reti 10Base5 e 10Base2 si identificavano nella tratta di cavo coassiale che costituiva un *singolo percorso elettrico non interrotto*, un singolo lungo pezzo di cavo, chiuso alle due estremità dai terminatori.
Aggiungendo un ripetitore alla rete si creava un secondo segmento e, come sappiamo, il limite superiore massimo consentito era quello di cinque segmenti.

> Generalmente nelle reti realizzate su coassiale sottile esisteva un solo segmento e questa era la LAN!

Nelle reti connesse con doppino, con "segmento" si intende la parte di rete composta solo da due nodi: da una parte la stazione di lavoro, server o stampante, e dall'altra la porta del concentratore.
Se accettiamo come definizione di segmento la seguente:

SEGMENTO
Tratto di cavo elettricamente ininterrotto che collega i nodi.

è necessario passare dal concetto di segmento a quello di dominio di collisione per poter individuare un gruppo di nodi connessi tra loro.

Ogni singola macchina collegata alla rete 10BaseT costituisce un segmento elettrico a sé stante e, nello stesso tempo, fa parte di un unico dominio di collisione che la unisce a tutte le altre macchine collegate allo stesso concentratore.

DOMINIO DI COLLISIONE

I domini di collisione sono i segmenti fisici di rete dove si possono verificare collisioni. Tutte le macchine collegate allo stesso dominio di collisione condividono non solo il medesimo traffico ma anche le stesse collisioni, indipendentemente da quale sia il concentratore a cui sono collegate.

La crescita del numero di host ha portato le collisioni a livelli vertiginosi, compromettendo il funzionamento della rete, quindi si è deciso di "segmentarle" suddividendole in diversi domini di collisione, utilizzando i dispositivi che abbiamo descritto.

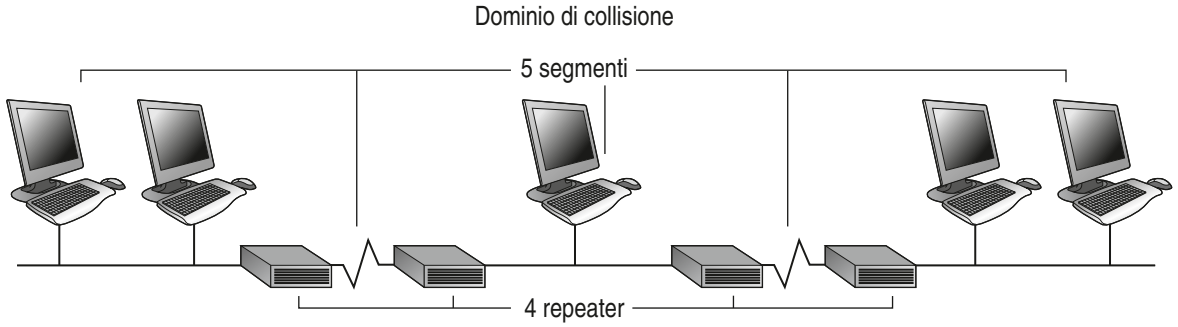
In particolare, riassumiamo il comportamento dei dispositivi che interconnettono i segmenti:
- hub e repeater: sono dispositivi di livello 1 che estendono il dominio di collisione;
- bridge e switch: sono dispositivi di livello 2 che spezzano il dominio di collisione;
- router: sono dispositivi di livello 3 che spezzano il dominio di collisione.

Più dispositivi ci sono in un dominio di collisione e maggiore è la probabilità di avere collisioni.

SEGMENTAZIONE

Con il temine segmentazione si intende l'operazione effettuata per spezzare il dominio di collisione in diverse parti aumentando il numero di domini di collisione ma riducendo il numero di host presenti in ciascun dominio.

Possiamo completare la regola del 5-4-3 aggiungendo alcuni dettagli:
- 5: numero massimo di segmenti;
- 4: numero di repeater o hub;
- 3: segmenti con host;
- 2: sezioni senza host;
- 1: dominio di collisione.



La seconda regola stabilisce che non si possano mettere più di quattro repeater tra due computer qualsiasi della rete, e se questa regola viene violata si rischia di superare il limite massimo di ritardo consentito, con il risultato di incrementare notevolmente il numero di collisioni.

Una delle conseguenze possibili è quella di introdurre collisioni riconosciute dopo i 64 byte, le late collision, che provocano la perdita del frame introducendo un ulteriore ritardo, dato che le NIC sono progettate per non ritrasmettere automaticamente dopo una late collision.

Messaggi di broadcast

Con il termine broadcast si intende la trasmissione di un pacchetto a tutte le stazioni presenti in ogni dominio di collisione.
Come vedremo, un caso tipico di trasmissione broadcast da parte di un host avviene quando è noto l'indirizzo IP (di livello 3) ma non si conosce l'indirizzo MAC dell'host destinatario (ARP request).
È quindi necessario che un frame venga spedito in tutti i domini di collisione e venga quindi ritrasmesso dai dispositivi del livello 2, hub oppure switch.
A tal fine viene definito un particolare frame, chiamato proprio frame di broadcast, che ha la caratteristica di avere come indirizzo di destinazione la sequenza FF-FF-FF-FF-FF-FF: tutte le NIC riconoscono questo indirizzo e rispondono al messaggio.

Flusso dei dati

Prima di proseguire con il livello 3 riassumiamo il comportamento dei dispositivi sinora incontrati:

- livello 1: i dispositivi di livello 1 lasciano sempre passare i frame su tutti i segmenti di rete, senza fare filtraggio. Rigenerano e risincronizzano i segnali portandoli alla stessa qualità di trasmissione originale;
- livello 2: i dispositivi di livello 2 filtrano i dati in base all'indirizzo MAC di destinazione e lasciano passare solo i frame destinati a host che non siano sullo stesso segmento della sorgente; quindi spezzano i domini di collisione senza però interrompere il dominio di broadcast.
  Al frame vengono aggiunti gli indirizzi MAC di sorgente e di destinazione;
- livello 3: come vedremo i dati a livello 3 vengono incapsulati aggiungendo gli indirizzi IP di sorgente e di destinazione, mentre un frame è inviato solo se l'indirizzo di rete di destinazione è diverso da quello della sorgente: i dispositivi di livello 3 filtrano quindi i pacchetti in base all'indirizzo IP di destinazione, spezzando sia i domini di collisione sia quelli di broadcast.

Verifichiamo le conoscenze

1 **Con la topologia 10BaseT la topografia di rete è:**

- ○ a bus
- ○ a stella
- ○ ad anello
- ○ dipende dalla configurazione

2 **La distanza massima tra un nodo e un hub è:**

- ○ 10 m
- ○ 100 m
- ○ 1000 m
- ○ dipende dal tipo di doppino

3 **Il meccanismo di funzionamento dei bridge prende anche il nome di:**

- ○ store and forward
- ○ link and store
- ○ link and forward
- ○ store and repeat

4 **Quale affermazione in merito ai bridge è errata?**

- ○ operano un filtro sui pacchetti
- ○ isolano i domini di collisione
- ○ aumentano il throughput
- ○ non introducono limitazioni sul numero massimo delle stazioni
- ○ richiedono di modificare gli adattatori dei computer

5 **Nella "filtering table" per ogni host sono presenti i seguenti dati (indica quello non presente):**

- ○ indirizzo NIC
- ○ numero di porta
- ○ timestamp
- ○ numero di segmento

6 **La durata dell'ageing time è circa di:**

- ○ 5 secondi
- ○ 30 secondi
- ○ 5 minuti
- ○ 30 minuti

7 **Quale tra i seguenti è un indirizzo broadcast?**

- ○ FF-FF-FF-FF-FF
- ○ 00-00-00-00-00
- ○ FF-FF-FF-FF-FF-FF
- ○ 00-00-00-00-00-00

8 **In una LAN basata su switch e link 100BaseT viene perso un pacchetto: quale delle seguenti affermazioni è certamente falsa?**

- ○ si è perso perché potrebbe essersi verificata una collisione
- ○ si è perso perché uno switch può non aver avuto memoria libera
- ○ si è perso perché potrebbe essersi verificato un errore di trasmissione
- ○ si è perso perché potrebbe essersi rotto un doppino

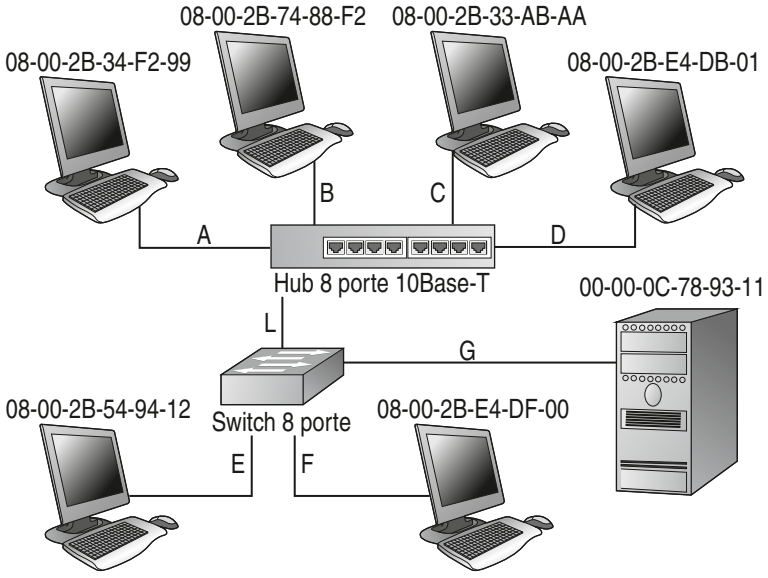
Test vero/falso

1	La trasmissione in full duplex raddoppia la banda.	V F
2	Nel metodo cut through lo switching è detto sincrono.	V F
3	Il metodo store and forward è detto switching asincrono.	V F
4	Nella tecnologia Gigabit Ethernet gli hub prendono il nome di "buffered distributors".	V F
5	Una rete LAN nella quale sono presenti gli hub prende anche il nome di HLAN.	V F
6	I bridge operano da filtro sui pacchetti.	V F
7	I bridge ritrasmettono i frame destinati allo stesso segmento.	V F
8	I bridge non possono essere utilizzati in presenza di differenti tecnologie sulla stessa rete.	V F
9	Una rete LAN nella quale sono presenti i bridge prende anche il nome di BLAN.	V F
10	La "filtering table" è la tabella che il bridge si crea.	V F
11	Nella "filtering table" dopo 300 secondi di inattività una riga viene rimossa.	V F
12	Una trama broadcast viene inoltrata su tutte le porte tranne quella da cui si è ricevuta.	V F
13	Una trama broadcast viene inoltrata su tutte le porte.	V F
14	Viene applicato un algoritmo di istradamento del tipo backward learning.	V F

Verifichiamo le competenze

Esprimi la tua creatività

Considera la seguente rete locale:



1 All'accensione dello switch sono stati trasmessi solo i seguenti pacchetti:

- ○ mittente: 08-00-2B-74-88-F2, destinatario: broadcast
- ○ mittente: 00-00-0C-78-93-11, destinatario: 08-00-2B-74-88-F2
- ○ mittente: 08-00-2B-54-94-12, destinatario: 00-00-0C-78-93-11
- ○ mittente: 08-00-2B-74-88-F2, destinatario: 08-00-2B-E4-DB-01

A quale o quali interfacce di rete verrà inoltrato il pacchetto n. 4? Perché?

2 All'accensione dello switch sono stati trasmessi solo i seguenti pacchetti:

- ○ mittente: 08-00-2B-74-88-F2, destinatario: broadcast
- ○ mittente: 00-00-0C-78-93-11, destinatario: 08-00-2B-74-88-F2
- ○ mittente: 08-00-2B-54-94-12, destinatario: 00-00-0C-78-93-11
- ○ mittente: 08-00-2B-74-88-F2, destinatario: 00-00-0C-78-93-11

A quale o quali interfacce di rete verrà inoltrato il pacchetto n. 4? Perché?

3 All'accensione dello switch sono stati trasmessi solo i seguenti pacchetti:

- ○ mittente: 08-00-2B-74-88-F2, destinatario: broadcast
- ○ mittente: 00-00-0C-78-93-11, destinatario: 08-00-2B-74-88-F2
- ○ mittente: 08-00-2B-54-94-12, destinatario: 00-00-0C-78-93-11
- ○ mittente: 08-00-2B-74-88-F2, destinatario: 00-00-0C-78-93-11

A quale o quali interfacce di rete verrà inoltrato il pacchetto n. 2? Perché?

4 All'accensione dello switch sono stati trasmessi solo i seguenti pacchetti:

- ○ mittente 08-00-2B-54-94-12, destinatario 08-00-2B-E4-DF-00
- ○ mittente 08-00-2B-34-F2-99, destinatario 08-00-2B-74-88-F2
- ○ mittente 08-00-2B-34-F2-99, destinatario 08-00-2B-54-94-12
- ○ mittente 00-00-0C-78-93-11, destinatario FF-FF-FF-FF-FF-FF
- ○ mittente 08-00-2B-E4-DF-00, destinatario 08-00-2B-34-F2-99
- ○ mittente 08-00-2B-74-88-F2, destinatario 08-00-2B-34-F2-99

a) Quali di questi frame saranno inoltrati dallo switch sull'interfaccia F? Perché?
b) A quali porte sarà inoltrato il frame n. 5? Perché?

5 All'accensione dello switch sono stati trasmessi solo i seguenti pacchetti:

- ○ mittente: 08-00-2B-34-F2-99, destinatario: 08-00-2B-E4-DB-01
- ○ mittente: 08-00-2B-E4-DB-01, destinatario: 08-00-2B-74-88-F2
- ○ mittente: 08-00-2B-54-94-12, destinatario: 00-00-0C-78-93-11
- ○ mittente: 08-00-2B-E4-DF-00, destinatario: 08-00-2B-34-F2-99

A quale o quali interfacce di rete verrà inoltrato il pacchetto n. 2? e il pacchetto n. 4? Perché?

ESERCITAZIONI DI LABORATORIO 1

UTILIZZO DI WIRESHARK

Premessa

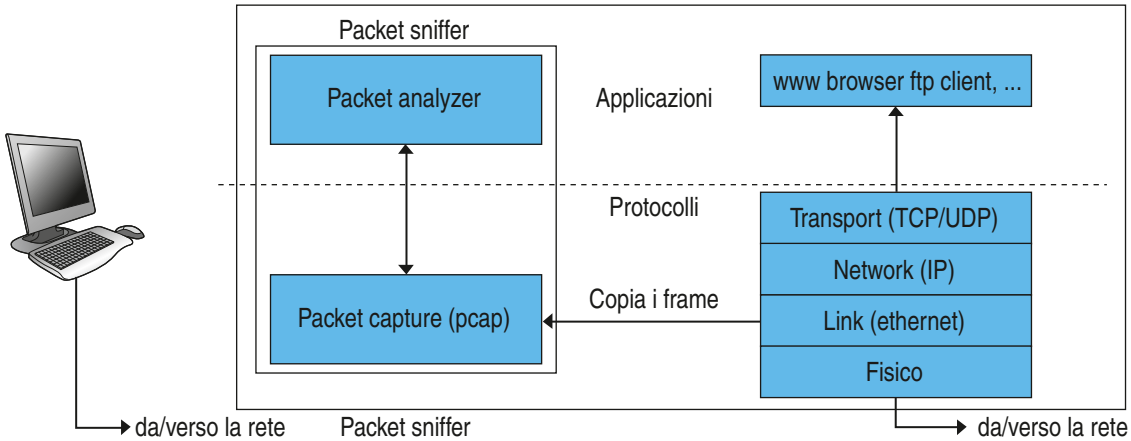
Lo studio dei protocolli di rete può essere effettuato con la prova diretta, osservando la sequenza di messaggi scambiati tra due entità di protocollo, approfondendo i dettagli delle operazioni e forzando i protocolli a effettuare determinate azioni per verificarne le conseguenze.

Per poter osservare i messaggi scambiati tra i diversi dispositivi è necessario un apposito programma, chiamato packet sniffer.

Il nome stesso dà un'indicazione sulla sua modalità di funzionamento: copia passivamente (ossia "sniffa", "annusa") i messaggi che vengono inviati e ricevuti dal computer sul quale è installato e li visualizza isolando i diversi campi e messaggi del protocollo utilizzato.

> Un packet sniffer è un'entità passiva: non manda mai dei pacchetti sulla rete ma si limita a osservare il "traffico" inviato e ricevuto dalle applicazioni e dai protocolli in esecuzione sul computer.

La figura seguente mostra la struttura di un packet sniffer: sulla destra ci sono i protocolli e le applicazioni (come i browser web o i client FTP) che normalmente girano su un computer, mentre sulla sinistra sono separate le componenti dell'architettura dello sniffer.



Il packet sniffer è organizzato in due parti:

- la pcap, o "libreria di cattura" dei pacchetti, riceve una copia di ogni frame che a livello di collegamento viene inviato o ricevuto dal computer: questo consente di ottenere tutti i messaggi ricevuti o inviati da tutti i protocolli e le applicazioni in esecuzione sullo stesso;
- l'analizzatore di pacchetti visualizza il contenuto di tutti i campi all'interno dei messaggi.

Un potente programma packet sniffer è Wireshark [http://www.wireshark.org], che consente di visualizzare i contenuti di tutti i messaggi inviati/ricevuti dai protocolli a differenti livelli della pila protocollare.

Wireshark è un software libero che gira su Windows, Linux/Unix e Mac OS.

È un prodotto collaudato e stabile e online è disponibile una buona documentazione del software:

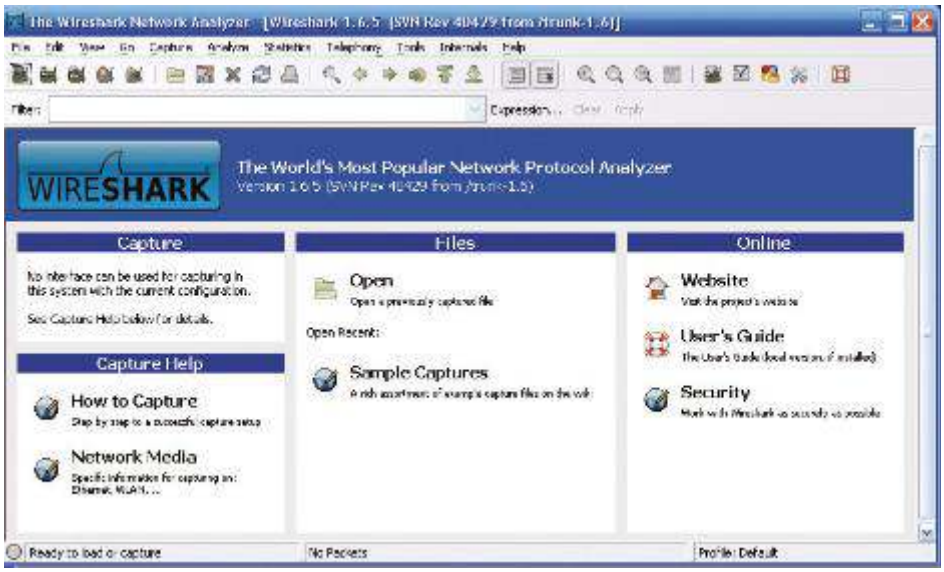
- una guida per l'utente [http://www.wireshark.org/docs/wsug_html_chunked/];
- un manuale operativo [http://www.wireshark.org/docs/man-pages/];
- una FAQ dettagliata [http://www.wireshark.org/faq.htm].

Scaricare, installare ed eseguire Wireshark

Per il sistema operativo Linux, Wireshark è incluso in pressoché qualunque distribuzione, per cui è sufficiente installarlo come si fa per tutto l'altro software.
Per gli altri sistemi operativi, è possibile scaricare il codice eseguibile di Wireshark dalla pagina http://www.wireshark.org/download.html (oppure nel materiale online nella sezione riservata a questo volume sul sito web http://www.hoepliscuola.it).

È opportuno anche scaricare la guida utente e consultare le FAQ che contengono un buon numero di consigli e informazioni varie, particolarmente utili se si hanno problemi nell'installazione o nell'esecuzione di Wireshark.

Mandando in esecuzione Wireshark viene visualizzata l'interfaccia utente grafica, simile a quella della figura seguente:



Le componenti principali dell'interfaccia di Wireshark sono descritte di seguito.

- Il menu dei comandi è un tipico menu a discesa collocato in cima alla finestra. Di principale interesse per noi sono i menu File e Capture. Il menu File consente di salvare i dati catturati, aprire un file contenente dati precedentemente catturati e uscire da Wireshark.

- Nella parte alta della finestra di Wireshark, sotto la barra dei comandi, è presente il campo Filter per specificare un filtro sui pacchetti da visualizzare, all'interno del quale è possibile digitare un nome di protocollo o altre informazioni per *filtrare* i pacchetti da visualizzare nell'elenco dei pacchetti catturati (e quindi anche nelle finestre del dettaglio e dei contenuti).

Nell'esempio che segue utilizzeremo il filtro per istruire Wireshark a nascondere tutti i pacchetti a eccezione di quelli che corrispondono a messaggi HTTP.

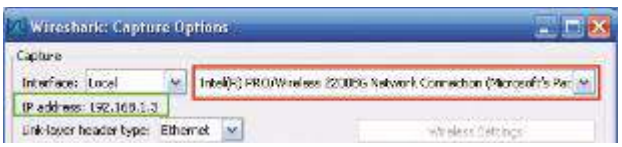
La parte centrale è suddivisa in tre sezioni, dove sono presenti:

- **A** le opzioni per effettuare la "cattura" dei pacchetti;
- **B** le opzioni per salvare/ripristinare o consultare pacchetti salvati in precedenza;
- **C** le opzioni per gli aiuti online.

> Wireshark utilizza per la cattura delle librerie già presenti nel PC: se non ci fossero, è necessario scaricarle dal sito www.winpcap.org.

Il menu Capture consente di iniziare la cattura dei pacchetti: seleziona Capture dal menu a discesa e quindi Options; verrà visualizzata la seguente schermata:

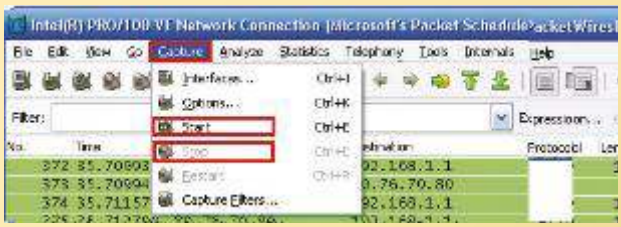
Lascia inalterati i valori di default presenti in questa finestra tranne Hide capture info dialog sotto Display Options, che va deselezionato.
Nel list box in alto sono elencate le interfacce di rete che sono all'interno del computer: è necessario selezionare l'interfaccia attiva, dalla quale si desidera "sniffare" i pacchetti.
Dopo qualche secondo verrà visualizzato l'indirizzo IP di rete dell'host: è un primo segnale che il programma sta funzionando correttamente.



Ora puoi iniziare a catturare i frame facendo clic sul pulsante Start.

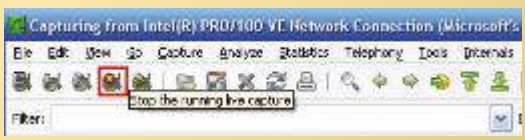
Le interfacce presenti nel PC verranno successivamente visualizzate anche nella sezione Capture della schermata inziale: è possibile iniziare a catturare i pacchetti anche facendo clic sulla scheda desiderata da questo elenco.

È inoltre possibile avviare la cattura dei pacchetti tramite l'opzione Start del menu Capture dove è presente anche il comando di Stop che interrompe la cattura.

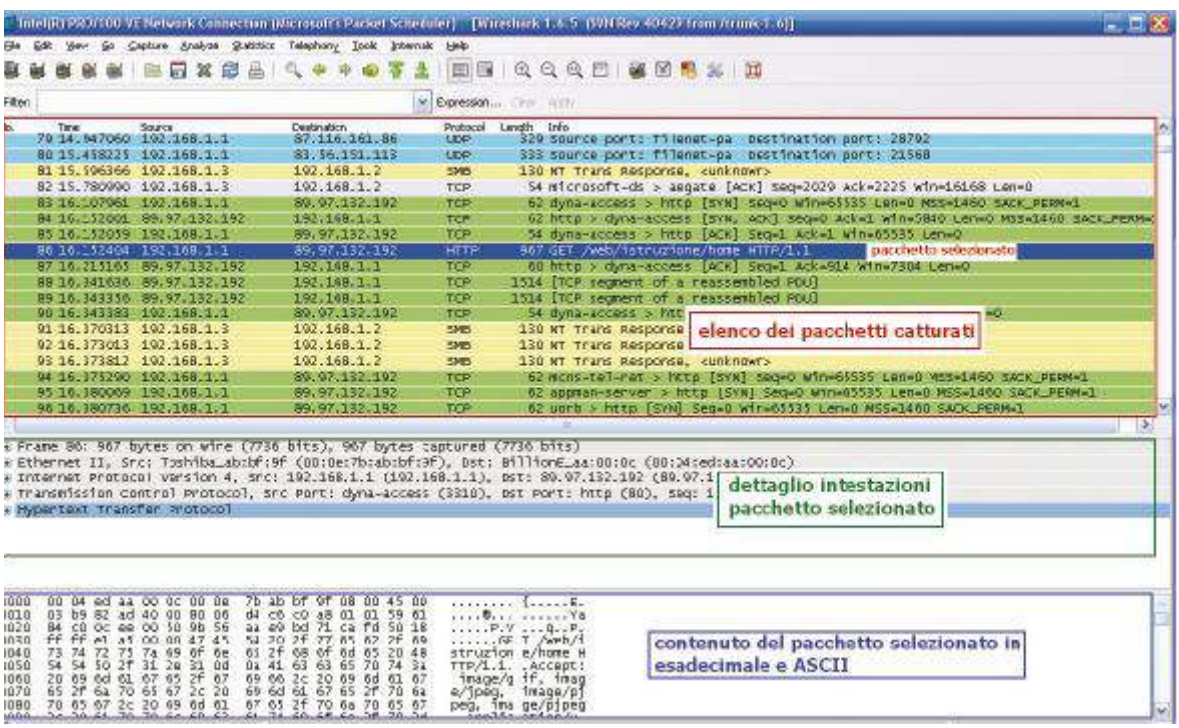


Un'ultima possibilità per avviare la cattura dei pacchetti è quella di utilizzare l'icona presente sulla barra dei comandi

dove si trova anche l'icona che interrompe la cattura.



Il programma inizia a catturare i pacchetti: la schermata iniziale si trasforma in una nuova videata composta da tre sezioni:



Nella parte superiore viene visualizzato l'elenco dei pacchetti catturati: una riga per ogni pacchetto che inizia con un numero progressivo assegnato da Wireshark, sulla quale vengono visualizzati:

- il tempo nel quale il pacchetto è stato catturato;
- gli indirizzi sorgente e destinazione;
- il tipo di protocollo;
- la lunghezza;
- altre informazioni specifiche del protocollo contenuto nel pacchetto.

Facendo clic sul nome della colonna è possibile cambiarne l'ordinamento di visualizzazione selezionando quello desiderato, in senso crescente o decrescente.

> Il campo con il tipo di protocollo mostra il protocollo di livello più alto contenuto nel pacchetto, ovvero il protocollo che utilizza la sorgente o il destinatario finale per il pacchetto.

La parte centrale è la finestra del dettaglio che mostra appunto, tra i pacchetti catturati e presenti nell'elenco, i dettagli sul pacchetto selezionato. Questi ultimi possono riguardare informazioni sul frame Ethernet e sul datagramma IP contenuti nel pacchetto.
L'ultima riga riporta anche i dettagli sul protocollo di livello superiore che ha inviato o ricevuto il pacchetto.

Nella finestra inferiore viene visualizzato l'intero contenuto del pacchetto, sia in forma esadecimale che ASCII.

Effettua ora una prova digitando all'interno di un browser un indirizzo a piacere (per esempio www.istruzione.it).

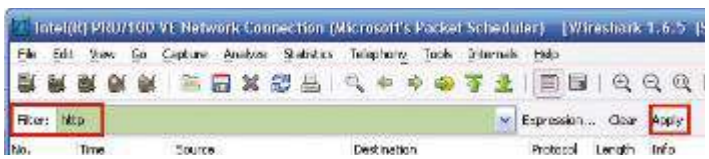
Il browser contatterà il server HTTP del ministero e scambierà con esso dei messaggi HTTP allo scopo di scaricare la pagina mentre Wireshark catturerà tutti i frame Ethernet contenenti i messaggi HTTP.

Appena viene visualizzata la home page del ministero, interrompi la cattura selezionando Stop dal menu Capture (oppure dalla barra dei comandi).

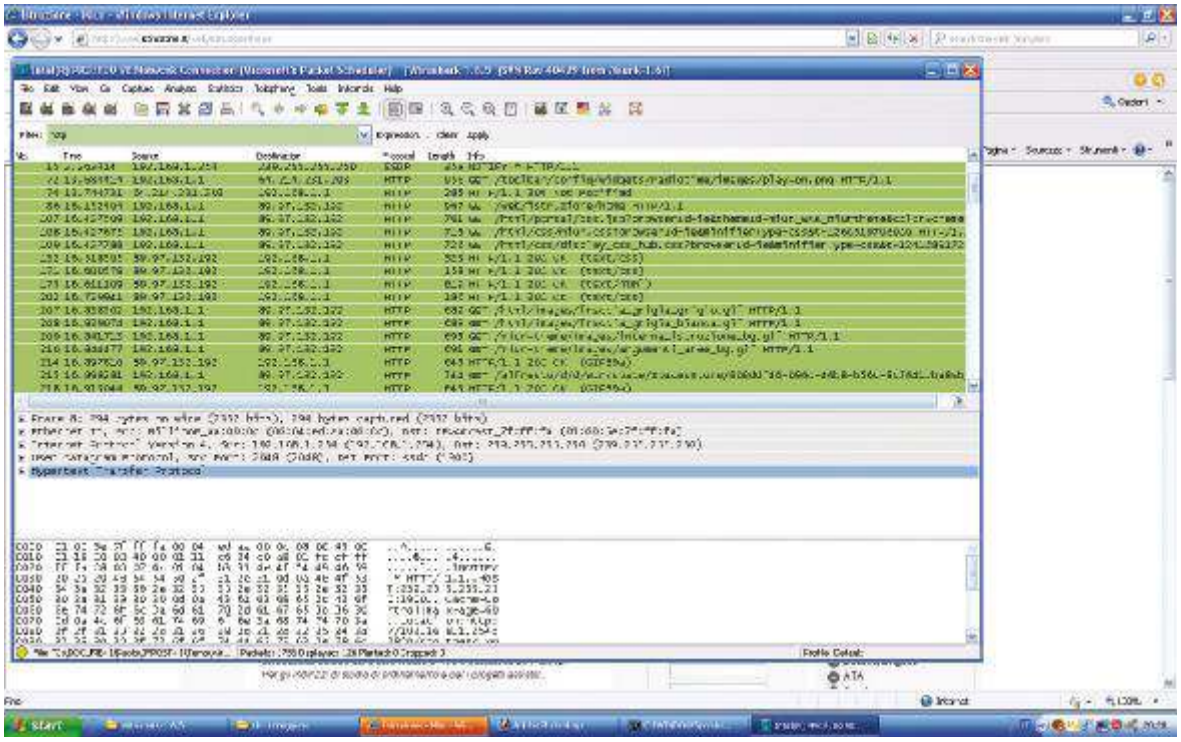
Nella parte superiore vengono visualizzati i pacchetti catturati "dal vivo" che contengono tutti i messaggi di tutti i protocolli che sono stati scambiati tra il nostro computer e le altre entità di rete, tra cui i messaggi HTTP con il server web del ministero www.istruzione.it.

> Nell'elenco sono presenti anche altri tipi di pacchetti: alcuni utilizzano protocolli "ignoti", com'è possibile vedere nella colonna Protocol anche se abbiamo effettuato una sola azione; altri pacchetti circolano sulla nostra NIC e vengono "sniffati" da Wireshark.

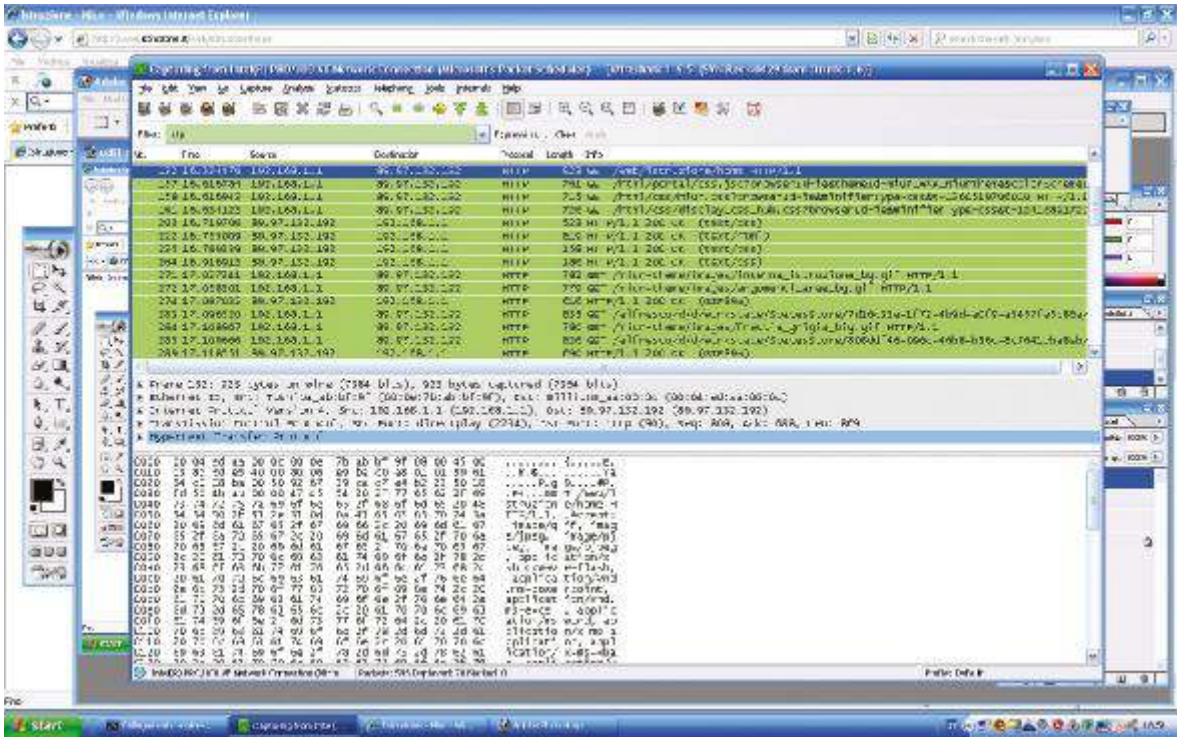
Per "fare un poco di pulizia" modifica il filtro nella parte alta della schermata digitando http in minuscolo nel campo Filter (dato che in Wireshark tutto deve essere scritto minuscolo) e confermando con il pulsante Apply



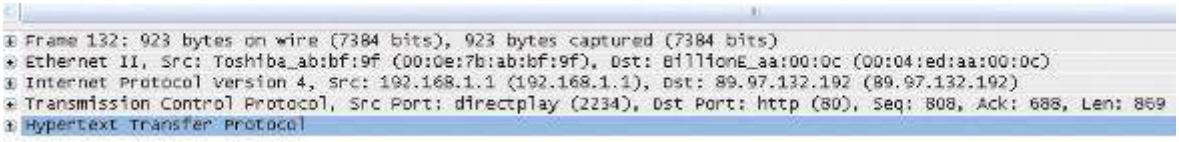
verranno visualizzati solo i pacchetti che utilizzano il protocollo HTTP (figura seguente).



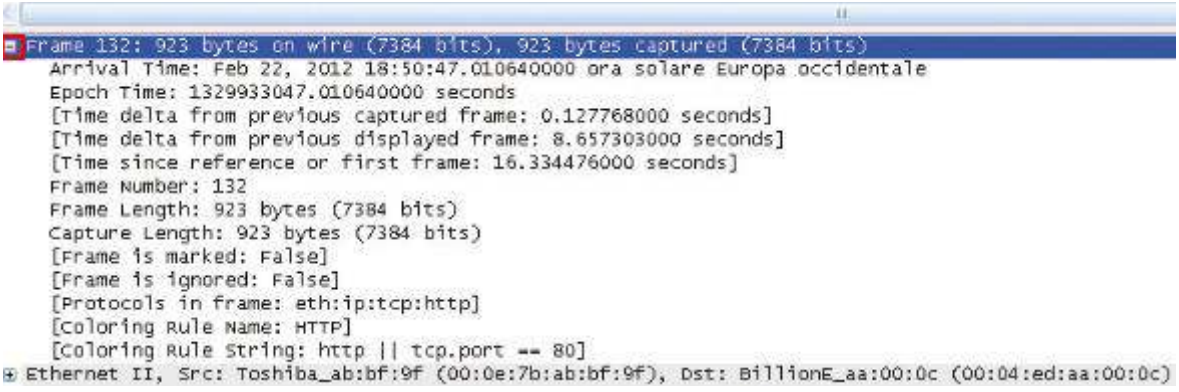
Seleziona tra i primi pacchetti presenti nell'elenco quello che contiene il messaggio GET mandato dal nostro computer a www.istruzione.it.



Quando selezioni il messaggio GET verranno visualizzate nella finestra centrale dei dettagli le informazioni sull'intestazione del frame Ethernet, del datagramma IP, del segmento TCP e del messaggio HTTP.



Seleziona ora il [+] vicino alla prima riga per visualizzare i dettagli del frame.



Ripeti la stessa operazione su ogni riga e osserva che cosa viene visualizzato, così potrai renderti conto della "potenza" di questa applicazione.

Prova adesso!

Dopo aver provato le altre opzioni presenti nel menu di Wireshark, riavvia la cattura dei pacchetti e visita un sito a tuo piacere, quindi rispondi alle seguenti domande.

1. Indica dieci differenti protocolli che appaiono nella colonna dei protocolli nell'elenco dei pacchetti catturati.
2. Quanto tempo è passato tra il momento in cui il messaggio HTTP GET è stato inviato e quello in cui il messaggio HTTP di risposta è stato ricevuto completamente?
   (Di default, il valore della colonna Time nell'elenco dei pacchetti è l'ammontare di tempo, in secondi, da quando Wireshark ha iniziato la cattura: per visualizzare, nel campo Time, anche le ore del giorno devi modificare le impostazioni mediante il menu a discesa View, modificando in Display Formats il Time-of-day.)
3. Qual è l'indirizzo internet del sito che hai visitato? Qual è l'indirizzo internet del tuo computer?
4. Stampa due dei messaggi HTTP visualizzati selezionando Print dal menu a discesa File, successivamente seleziona Selected Packet Only e Print as displayed e fai clic sul pulsante OK.

ESERCITAZIONI DI LABORATORIO 2

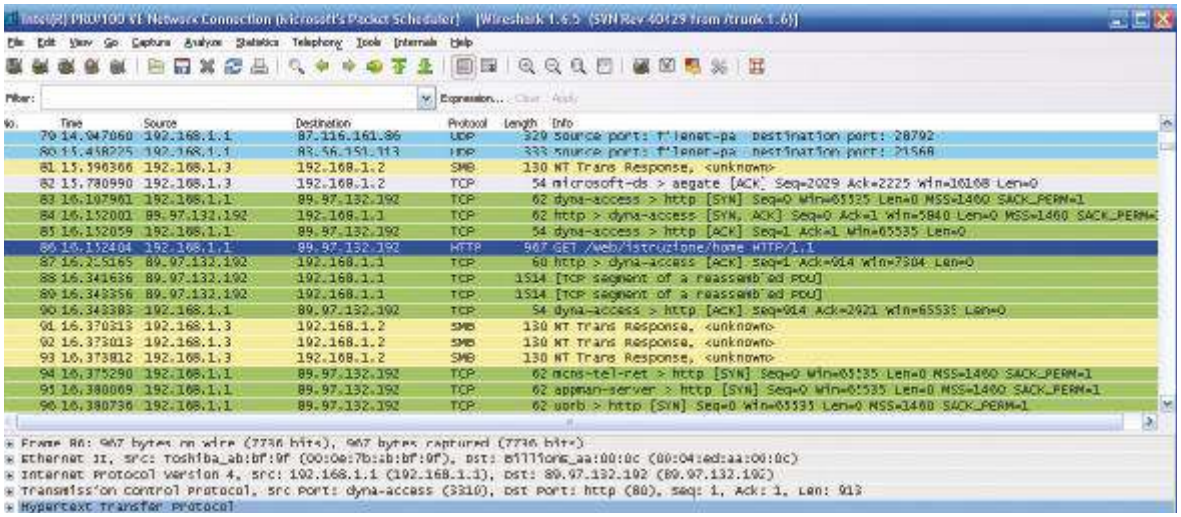
IL PROTOCOLLO ETHERNET

In questa esercitazione utilizzeremo Wireshark per studiare il protocollo Ethernet.
I riferimenti teorici per affrontare questa esercitazione sono presenti nelle Unità didattiche 1 e 2 del Modulo 5.

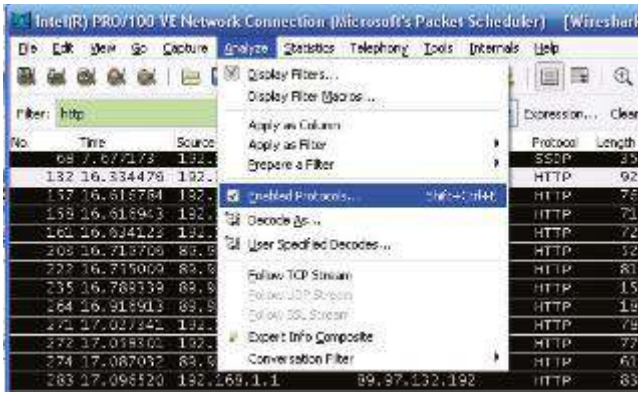
Catturare e analizzare i frame Ethernet

Prima di iniziare a catturare nuovi pacchetti è necessario che la cache del browser sia vuota:
- **A** con Firefox seleziona Strumenti → Cancella cronologia recente → Cancella adesso;
- **B** con IE seleziona Strumenti → Generale → Cronologia esplorazioni → Elimina.

1. Iniziamo l'esercitazione catturando una serie di frame Ethernet che andremo poi ad analizzare.
2. Avvia Wireshark e la cattura dei pacchetti.
3. Carica la seguente pagina web: http://www.istruzione.it.
4. Interrompi la cattura dei pacchetti.
5. Determina il numero di pacchetto (prima colonna a sinistra) del messaggio HTTP GET che è stato inviato dal tuo computer e l'inizio del messaggio di risposta HTTP inviato al tuo computer dal server del ministero.
6. Individua nella schermata la riga simile a quella evidenziata.



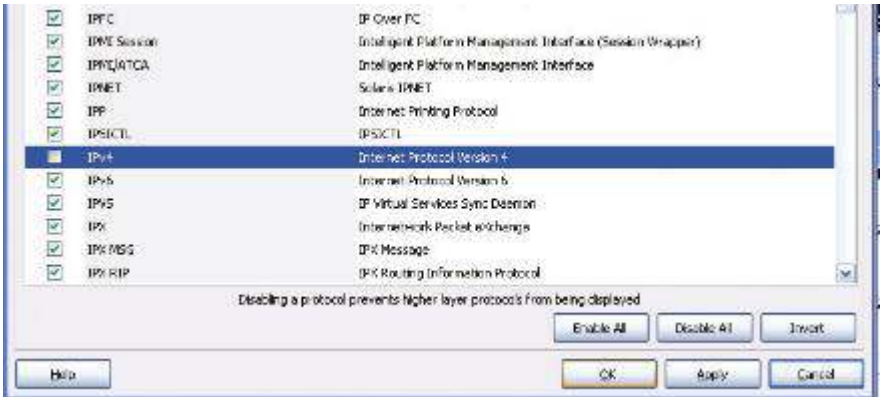
Lo scopo di questa esercitazione è quello di analizzare il protocollo Ethernet, quindi dobbiamo escludere tutto ciò che è "a livello superiore", a partire dal protocollo IP.



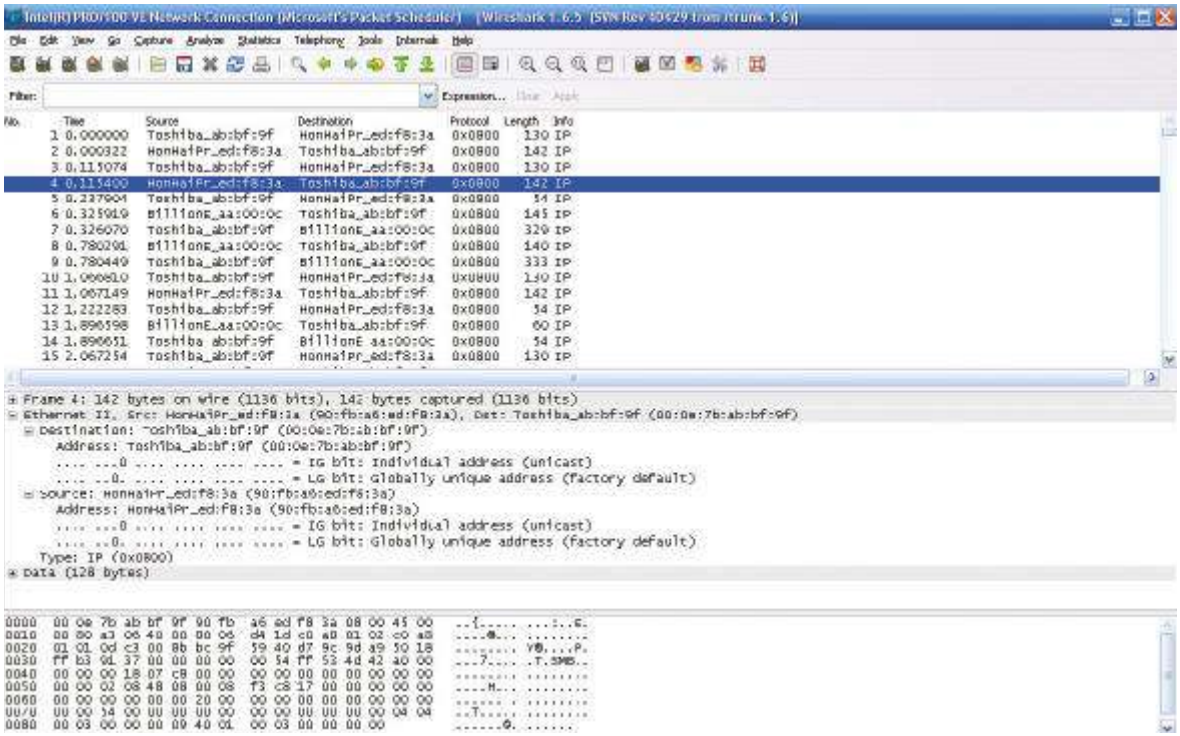
Configura Wireshark in modo da visualizzare solo le informazioni sui protocolli di livello inferiore a IP.

Seleziona quindi Analyze → Enabled Protocols ▶

… e nella schermata successiva disattiva IPv4 e conferma facendo clic sul pulsante OK.



Ora sullo schermo dovresti avere una videata simile alla seguente:



Per rispondere alle domande che seguono devi controllare le finestre con il dettaglio e con il contenuto dei pacchetti (sono le due finestre in basso nella finestra principale di Wireshark).

Seleziona il frame Ethernet contenente il messaggio HTTP GET: esso è trasportato all'interno del segmento TCP, che si trova a sua volta dentro un datagramma IP, che è infine trasportato dentro un frame Ethernet.

Espandi le informazioni Ethernet II nella finestra con il dettaglio dei pacchetti come indicato nella figura precedente.

Nella finestra inferiore è possibile visualizzare tutto il contenuto del frame Ethernet (intestazione e payload).

Prova adesso!

Stampa il pacchetto selezionato da File → Print, attiva Selected packet only e Packet summary line e scegli la minima quantità di dettagli necessaria per rispondere alla domanda.

1. Quanti byte compongono il pacchetto?
2. Individua il preambolo analizzando i valori esadecimali: quale osservazione puoi fare?
3. Qual è l'indirizzo Ethernet del tuo computer (6 byte)?
4. Qual è l'indirizzo di destinazione nel frame Ethernet (6 byte)?
5. Individua il campo Type nel frame Ethernet. Quali bit hanno valore 1?
6. Qual è la lunghezza del PAD?
7. Qual è il valore esadecimale del campo CRC in questo frame Ethernet?

Individua ora il primo frame di risposta (quello con OK nel campo info) e rispondi alle seguenti domande.

1. Quanti byte compongono il pacchetto?
2. Qual è il valore del campo indirizzo sorgente Ethernet?
3. Qual è l'indirizzo di destinazione del frame Ethernet?
4. Individua il campo Type nel frame Ethernet. Quali bit hanno valore 1?
5. Qual è il valore esadecimale del campo CRC in questo frame?

ESERCITAZIONI DI LABORATORIO 3

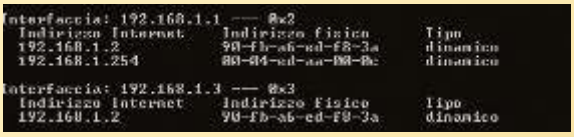
PROTOCOLLO ARP

In questa esercitazione utilizzeremo Wireshark per studiare il protocollo ARP.
I riferimenti teorici per affrontare questa esercitazione sono presenti nell'Unità didattica 3 del Modulo 5 e nella specifica del protocollo ARP che puoi scaricare da http://tools.ietf.org/html/rfc826.

Il protocollo ARP mantiene una cache della tabella di traduzione da indirizzi IP a indirizzi Ethernet: si utilizza il comando arp per vedere e manipolare il contenuto di questa cache.
Digita il comando

arp-a

al prompt dei comandi e visualizza il contenuto della cache ARP presente nel tuo computer come nel seguente esempio:



Che cosa puoi osservare e quali indicazioni trai da questa schermata?

Prima di iniziare l'esercitazione è necessario azzerare il contenuto della memoria cache in modo da poter osservare il computer che spedisce e riceve messaggi ARP.

Infatti, se l'associazione tra indirizzo IP e indirizzo Ethernet è già presente nella cache non verrà inviato alcun messaggio ARP.

Per cancellare la cache basta digitare il comando

arp -d *

dove
- il flag -d indica un'operazione di cancellazione;
- il carattere jolly indica di cancellare tutte le righe della tabella.

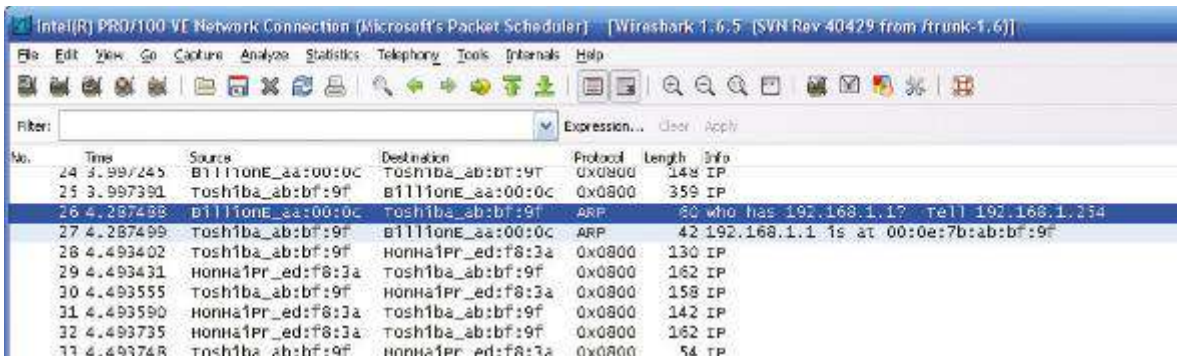
Su Linux/Unix devi avere i privilegi di root (amministratore di sistema) per poter cancellare la cache ARP.

Dopo aver cancellato la cache ARP svuota anche la cache del browser e manda in esecuzione Wireshark in modo che inizi a catture i pacchetti, digitando nel browser www.istruzione.it.

Non appena viene visualizzata la home page del sito della Pubblica Istruzione, interrompi la cattura dei pacchetti.

Dato che non siamo interessati ai protocolli di livello superiore deseleziona dalla tendina Analyse il procotollo IPv4 in modo da selezionare solo le informazioni sui protocolli che stanno sotto il livello IP.

Otterrai una finestra simile alla seguente:



In questo esempio i frame 26 e 27 contengono messaggi ARP.

Prova adesso!

Espandi il dettaglio della descrizione dei singoli componenti del frame, come riportato nella figura seguente:

```
⊞ Frame 26: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
⊟ Ethernet II, Src: BillioneE_aa:00:0c (00:04:ed:aa:00:0c), Dst: Toshiba_ab:bf:9f (00:0e:7b:ab:bf:9f)
  ⊟ Destination: Toshiba_ab:bf:9f (00:0e:7b:ab:bf:9f)
      Address: Toshiba_ab:bf:9f (00:0e:7b:ab:bf:9f)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
  ⊟ Source: BillioneE_aa:00:0c (00:04:ed:aa:00:0c)
      Address: BillioneE_aa:00:0c (00:04:ed:aa:00:0c)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Type: ARP (0x0806)
    Trailer: 000000002ab58680000000002ab588240000
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    [Is gratuitous: False]
    Sender MAC address: BillioneE_aa:00:0c (00:04:ed:aa:00:0c)
    Sender IP address: 192.168.1.254 (192.168.1.254)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.1 (192.168.1.1)
```

Stampa i due pacchetti selezionati da File → Print, attiva Selected packet only e Packet summary line scegliendo la minima quantità di dettagli che è necessaria per rispondere alle seguenti domande.

1. Quanti byte compongono il pacchetto?
2. Qual è l'indirizzo sorgente del frame contenente il messaggio di richiesta ARP (6 byte)?
3. Qual è l'indirizzo di destinazione del frame contenente il messaggio di richiesta ARP (6 byte)?
4. Individua il campo Type nel frame Ethernet. Quali bit hanno valore 1?
5. Dall'inizio del frame Ethernet, dopo quanti byte comincia il campo Opcode ARP?
6. Qual è il valore del campo Opcode all'interno della request?
7. Il messaggio ARP contiene l'indirizzo IP del mittente?
8. Dove si individua la richiesta ARP dell'indirizzo IP della macchina desiderata?

Individua ora il messaggio di ARP reply inviato come risposta.

1. Quanti byte compongono il pacchetto?
2. Da quanti byte è formata la risposta (reply)?
3. In quale posizione è il campo Opcode ARP?
4. Qual è il valore del campo Opcode?

MODULO 6

LO STATO DI RETE E IL PROTOCOLLO TCP/IP

OBIETTIVI

- Sviluppo di Internet e del protocollo TCP/IP
- Il confronto tra i livelli ISO/OSI e TCP/IP
- I 4 strati del modello TCP/IP e le loro funzioni
- La struttura degli indirizzi IP
- Le classi degli indirizzi IP
- Differenze tra indirizzamento pubblico e privato
- Assegnazione statica e dinamica degli indirizzi
- La messaggistica ICMP
- Il protocollo ARP/RARP
- Il funzionamento del protocollo DHCP

ATTIVITÀ

- Funzioni degli indirizzi IP riservati
- Scomporre una rete in sottoreti
- Definire reti con maschere di lunghezza variabile
- Aggregare più reti in una supernetting
- Assegnare staticamente gli indirizzi IP
- Utilizzo di ARP per ottenere gli Indirizzi MAC
- Configurare manualmente un PC
- Configurare automaticamente un PC con il DHCP
- Visualizzare lo stato di un PC

UNITÀ DIDATTICA 1

IL TCP/IP E GLI INDIRIZZI IP

IN QUESTA UNITÀ IMPAREREMO...

- il confronto tra i livelli ISO/OSI e TCP/IP
- il formato dei dati nel TCP/IP
- la struttura degli indirizzi IP
- le classi degli indirizzi IP

Cenni storici

Nella prima metà degli anni '70 il Ministero della Difesa statunitense DARPA (Defence Advanced Research Project Agency), nell'ambito del progetto di una rete di computer che potesse sopravvivere anche in caso di attacco bellico, definì il modello sul quale si sarebbe poi basato lo sviluppo della rete Internet.

Questo modello, cui venne dato il nome di Internet Protocol Suite, presenta due protocolli principali:

- IP (*Internet Protocol*);
- TCP (*Transmission Control Protocol*).

Proprio da tali protocolli deriva il nome TCP/IP che viene usato per identificare questa architettura di rete.

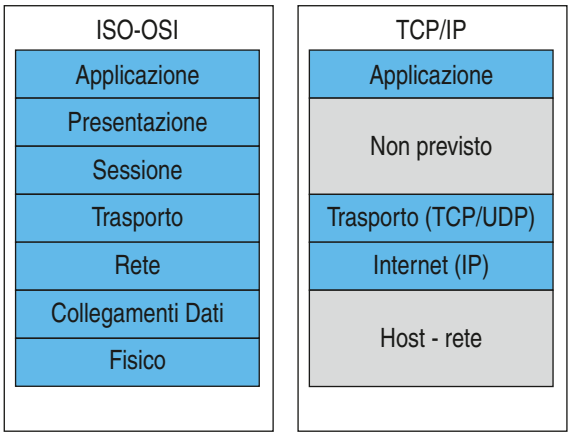
Nel 1974 la rete ARPAnet collegava la costa dell'Atlantico con quella del Pacifico grazie a un sistema a commutazione di pacchetto in grado di connettere elaboratori eterogenei.

ARPAnet continuò a svilupparsi in ambito universitario e governativo nel corso di tutti gli anni '70, ma fu nel 1974, con la prima standardizzazione del protocollo TCP/IP, che il progetto della rete venne denominato Internet.

◀ RCF Gli RCF sono pubblicati dalla IAB (*Internet Architecture Board*), ossia dal gruppo tecnico incaricato di stabilire gli standard per Internet: sul sito RCF Editor (http://www.rfc-editor.org/) si possono trovare tutti i documenti pubblicati e le informazioni necessarie per produrre un nuovo documento. ▶

Da allora la suite TCP/IP si è evoluta e al suo interno sono stati aggiunti molti altri protocolli, ciascuno dotato di un proprio compito, che sono specificati con documenti detti ◀ RFC ▶ (*Request For Comment*): famoso è rimasto il documento RFC 791 Internet Protocol, datato 1981, che specifica il protocollo IP come lo conosciamo e lo utilizziamo noi ora.

L'architettura del modello TCP/IP si basa solo su quattro livelli, a differenza del modello ISO/OSI che ne prevede sette.
Dal confronto dei due modelli si possono trarre alcune osservazioni preliminari:



- **A** due livelli sono in corrispondenza:
  - il livello 3 ISO/OSI Rete con il livello Internet del TCP/IP;
  - il livello 4 ISO/OSI Trasporto con il livello Trasporto del TCP/IP: hanno lo stesso nome, ma svolgono funzioni diverse;
- **B** il livello Internet si "appoggia" sull'equivalente del livello 2 ISO/OSI, cioè i due livelli più bassi della pila OSI corrispondono a quello che in TCP viene chiamato livello host-rete;
- **C** in TCP/IP non sono previsti i livelli di presentazione e sessione: le loro caratteristiche utili devono quindi essere implementate dall'applicazione.

I livelli del TCP/IP

Descriviamo ora succintamente i quattro livelli del TCP/IP.

Applicazione

Il livello di applicazione (o Application Layer) comprende tutti i protocolli di alto livello e di dialogo con l'utente, tra cui quelli specifici per il trasferimento di file, le e-mail, il login remoto.
Di seguito sono riportati alcuni dei protocolli presenti a questo livello, che tuttavia saranno oggetto di studio nei corsi successivi:

- FTP (File Transfer Protocol): è un servizio TCP per il trasferimento di file;
- DNS (Domain Name System): è un sistema usato in Internet per tradurre i nomi di dominio (un indirizzo simbolico, per esempio un indirizzo di posta elettronica) in indirizzi della rete (cioè in un indirizzo IP);
- NFS (Network File System): è un protocollo che permette l'accesso a file memorizzati su un dispositivo remoto, per esempio su un hard disk in rete;
- SMTP (Simple Mail Transfer Protocol): amministra la trasmissione di e-mail in rete;
- Telnet (Terminal Emulation, "terminale virtuale"): offre la possibilità di accedere da remoto a un altro computer.

Grazie all'evoluzione cui l'architettura TCP/IP è stata oggetto negli anni, oltre ai servizi appena descritti se ne sono sviluppati altri: uno di essi è l'HTTP, un protocollo usato per caricare le pagine del World Wide Web.

Trasporto

◀ Acknowledge In un protocollo il termine inglese acknowledge, tradotto letteralmente con l'espressione "accusare ricevuta", indica la modalità per cui durante la trasmissione il mittente attende una conferma dal destinatario per proseguire le sue operazioni o ritrasmettere il messaggio andato perduto. ▶

Il livello di trasporto (o Transport Layer) crea una connessione logica tra sorgente e destinazione indipendentemente dalla rete utilizzata, assemblando e segmentando i dati che riceve dal livello di applicazione e inviando al destinatario un segmento per volta.
A ogni segmento viene assegnato un numero che permette di verificarne la consegna al destinatario, in modo da garantire l'affidabilità della trasmissione (◀ acknowledge ▶).

A livello di trasporto si distinguono due tipi di servizi che, come già specificato per il livello di applicazione, saranno oggetto di studio nei corsi successivi:

- servizi affidabili orientati alla connessione, detti di tipo stream, offerti dal TCP (*Transmission Control Protocol*);
- servizi senza connessione, detti di tipo datagram, offerti dall'UDP (*User Datagram Protocol*).

Internet

Il livello di Internet (Internet Layer, detto anche semplicemente livello IP) ha lo scopo di selezionare il miglior percorso attraverso la rete per recapitare il messaggio al destinatario.

L'obiettivo di interconnettere più reti in modo da ottenere un alto grado di robustezza, intesa come la capacità di funzionamento anche in caso di guasto di nodi intermedi, ha portato alla scelta di un *livello di rete* a *commutazione di pacchetto* di tipo *connectionless*.

Zoom su...

CONNECTIONLESS

La comunicazione di tipo *connectionless* si basa sul fatto che tra due punti finali della rete in cui un messaggio può viaggiare non esiste una connessione diretta e il mittente trasmette i dati senza accertarsi che il destinatario sia disponibile e in attesa della comunicazione. In questo tipo di comunicazione può essere necessario ritrasmettere più volte le parti di messaggio che vengono perse nella rete.

A ogni computer di una rete TCP/IP viene assegnato un indirizzo IP, detto indirizzo logico, che rappresenta un identificativo univoco e opera a livello 3 della pila ISO/OSI, a differenza dell'indirizzo MAC, che individua fisicamente un dispositivo a livello 2.
Il messaggio da trasmettere ad alto livello viene suddiviso in pacchetti (segmentazione) di dimensioni inferiori, non più di 1526 byte, che è la dimensione massima possibile su una rete Ethernet.
L'Internet Layer effettua la spedizione dei pacchetti che costituiscono il messaggio in ogni tipo di rete e in un qualsiasi nodo di destinazione attraverso il protocollo IP (Internet Protocol), che li inoltra all'indirizzo IP finale utilizzando meccanismi di istradamento (routing) per mezzo di dispositivi appositi (router).
Generalmente i pacchetti nei quali il messaggio è stato segmentato raggiungono la destinazione seguendo percorsi differenti e quindi anche l'ordine di arrivo potrebbe essere diverso dall'ordine di partenza: è compito del livello superiore (Transport) ricomporre correttamente l'intero messaggio.

La comunicazione in Internet avviene quindi nel modo seguente: il livello di trasporto gestisce le informazioni in forma di data stream, che vengono frammentati in datagrammi a livello IP e inviati attraverso percorsi diversi sulla rete all'indirizzo IP del destinatario.

Zoom su...

ROUTER

Il router (o gateway IP) è un dispositivo che collega due o più reti locali: quando esso riceve un frame, innanzitutto ne estrae il pacchetto ed esamina l'indirizzo IP. A questo punto, due sono le situazioni che possono determinarsi:

- il router sa risolvere l'indirizzo IP: traduce quindi l'indirizzo logico in indirizzo fisico dell'host che si trova su una delle altre reti a cui è collegato, crea un frame ed effettua l'inoltro su quella rete;
- il router non sa eseguire la risoluzione: in tal caso, invia il frame a un altro router che si trova in una delle reti alle quali è connesso, "sperando" che esso sia in grado di risolvere l'indirizzo IP.

RISOLVERE L'INDIRIZZO IP

Con l'espressione "risolvere l'indirizzo IP" si intende l'operazione che permette di individuare l'indirizzo fisico oppure il nome di una macchina in una rete o in un suo segmento.

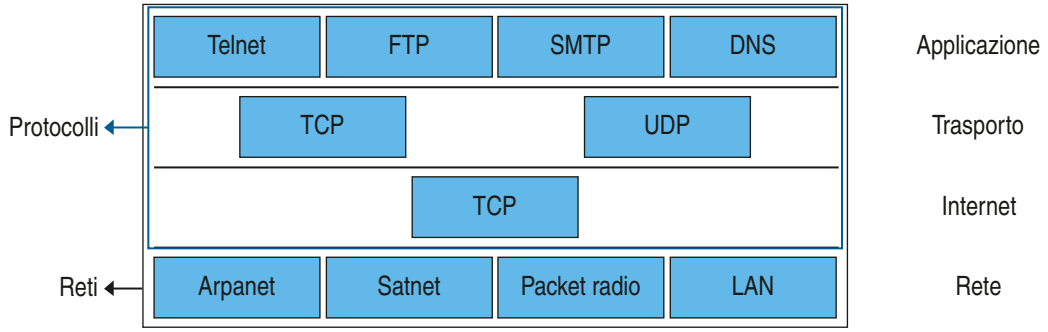
Per risolvere l'indirizzo IP sono disponibili a questo livello due strumenti (protocolli), che verranno descritti nel seguito della trattazione:

- ARP (*Address Resolution Protocol*): fornisce l'indirizzo MAC a partire dall'indirizzo IP;
- RARP (*Reverse Address Resolution Protocol*): determina l'indirizzo IP a partire dall'indirizzo MAC.

Rete

Il livello sottostante l'Internet Layer, detto anche host-to-network layer, non è in realtà specificato rigorosamente dal modello di riferimento TCP/IP, in quanto il protocollo che si utilizza varia da un host all'altro e da rete a rete, includendo quindi tutte le tecnologie LAN e WAN e tutti i dettagli contenuti nei livelli 1 e 2 del modello OSI.

Possiamo riassumere nel seguente schema, livello per livello, i diversi protocolli della pila TCP/IP:



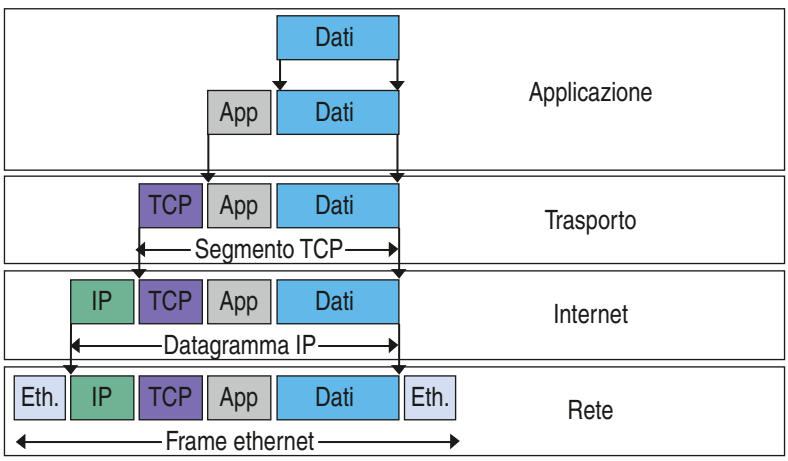
Formato dei dati nel TCP/IP

Il protocollo IP fornisce l'instradamento dei pacchetti in modalità detta best-effort delivery ("miglior sforzo per spedire a destinazione"), ma non è affidabile e non effettua la correzione di errore. Vediamo, di seguito, come viene costruito un pacchetto IP.

Quando un'applicazione invia dei dati utilizzando l'architettura TCP/IP, questi seguono un percorso "dall'alto verso il basso" attraverso tutti i livelli della pila fino a essere trasmessi dal livello fisico; ogni livello aggiunge una serie di informazioni di controllo ai primi dati che riceve, gli header (intestazione), fino a giungere al livello di rete che, come abbiamo visto, oltre all'intestazione aggiunge anche alcuni dati in coda (il cosiddetto trailer).

Quindi, ricapitolando:

- lo strato di applicazione aggiunge un'intestazione (header-app) ai dati utente prima di passarli allo strato di trasporto;
- il protocollo TCP (oppure UDP) dello stato di trasporto aggiunge anch'esso un'intestazione: l'unità di dati prende ora il nome di segmento e viene passata allo strato di rete;
- lo strato di rete accluse a sua volta un'intestazione comprendente l'indirizzo IP: a questo punto il dato assume la denominazione di datagramma IP;
- questa unità di informazione viene infine passata ai livelli inferiori, dove lo strato di collegamento aggiunge la propria intestazione (header) e una coda (trailer): siamo finalmente arrivati alla trama (frame ethernet).



IMBUSTAMENTO

Con il termine imbustamento (o incapsulamento) si intende l'insieme delle operazioni che ogni livello in spedizione esegue aggiungendo i dati caratteristici del livello stesso (busta) per poi spedirli al livello sottostante.

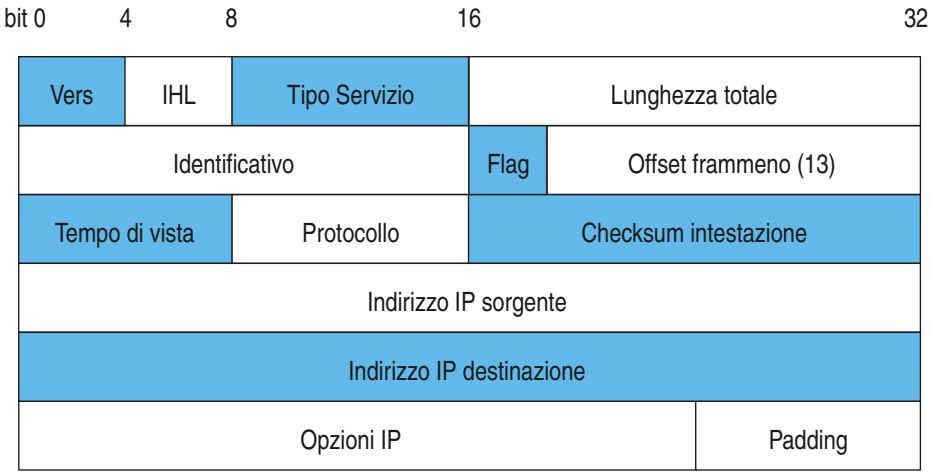
In ricezione ogni livello acquisisce quindi una "busta" contenente dati, grazie alla quale è in grado di trattare i dati presenti al suo interno: ogni livello tratta come "dati grezzi" le informazioni ricevute dal livello superiore non distinguendo tra i dati originari forniti dall'applicativo e gli header (intestazioni) aggiunti dai livelli che essi hanno già attraversato.

Nel protocollo TCP/IP questa operazione prende anche il nome di imbustamento multiplo, dato che viene effettuata a ogni livello.

L'intestazione IP

Vediamo in dettaglio che cosa viene aggiunto in forma di intestazione dal protocollo IP, come descritto dall'RFC 791.

L'intestazione IP è costituita da alcuni gruppi di 4 byte, come indicato nel seguente schema:



- Vers: è il campo che identifica la versione del protocollo IP del pacchetto; oggi quella più utilizzata è l'IPv4, sebbene ci si stia indirizzando sempre più verso l'uso della versione IPv6 (denominata anche IPng, *IP next generation*).
- IHL (*Internet Header Length*): contiene 4 bit impostati a un valore che indica la lunghezza dell'intestazione dei datagrammi, espressa in parole di 32 bit: la lunghezza minima è 5.
- Tipo di servizio (TOS): specifica il tipo di servizio richiesto, oltre a venire usato come indicatore di priorità.
- Lunghezza totale: specifica la lunghezza totale del datagramma misurata in byte e comprende la lunghezza dell'intestazione e dei dati: la lunghezza massima possibile per un datagramma è di 65.535 byte.

Il protocollo IP utilizza tre campi nell'intestazione per controllare la frammentazione e il riassemblaggio dei datagrammi: si tratta dei campi Identificativo, Flag e Offset (scostamento) frammento.

- Identificativo: valore intero che serve all'host ricevente per identificare univocamente un datagramma;
- Flag: bit 0 reserved (R) sempre a 0
  bit 1 don't fragment (DF)
  DF = 0 si può frammentare
  DF = 1 non si può frammentare
  bit 2 more fragments (MF)
  MF = 0 ultimo frammento
  MF = 1 frammento intermedio
- Offset frammento: indica qual è la posizione di questo frammento nel datagramma; il valore si misura in unità di 8 byte;
- Tempo di vita (TTL Time To Live): corrisponde al tempo massimo di permanenza del datagramma nella rete, decrementato da ogni nodo (router) che esso attraversa; il datagramma verrà distrutto dal nodo che riduce questo valore a 0. In altre parole, indica quindi il numero massimo di tratti di rete che un segmento può attraversare;

- Protocollo: indica a quale protocollo di livello superiore appartengono i dati contenuti nel datagramma;
- Checksum dell'intestazione: rappresenta il controllo di errore della sola intestazione, ricalcolato da ogni nodo che elabora il datagramma;
- Indirizzo IP sorgente e destinazione: corrisponde agli indirizzi dell'host mittente e del destinatario;
- Opzioni IP: ha lunghezza variabile e contiene opzioni relative al trasferimento del datagramma (registrazione del percorso, meccanismi di sicurezza);
- Padding: consiste in bit privi di significato aggiunti in funzione della lunghezza del campo precedente per far sì che l'intestazione sia multipla di 32 bit.

Struttura degli indirizzi IP

Nelle reti TCP/IP a ogni host deve essere assegnato l'indirizzo IP; quest'ultimo, composto da 32 bit (4 byte), viene espresso scrivendo i valori decimali di ciascun byte da 0 a 255, separati dal carattere punto (rappresentazione dotted-decimal).
Un esempio di indirizzo IP è il seguente:

137.204.213.2 -> 10001001.11001100.11010101.00000010

Utilizzando 32 bit il numero massimo teorico di indirizzi è dato da

$$2^{32} = 4.294.967.296$$

Possiamo osservare come questo numero sia relativamente basso, in quanto è impensabile che al mondo siano connessi "solo" poco più di 4 milioni di calcolatori.
Inoltre, come vedremo, non tutti gli indirizzi possono essere utilizzati, e quindi il numero risulta ancora più esiguo.
Ricordiamo infine che non sono i nodi ad avere un indirizzo IP, bensì le interfacce: quindi se un nodo (per esempio un router) ha tre interfacce, esso è dotato di tre indirizzi IP.

Per poter connettere il maggior numero di calcolatori utilizzando "solo" 32 bit sono state introdotte alcune tecniche particolari, che analizziamo di seguito.

Per prima cosa si è "spezzato in due parti" l'indirizzo IP:
- indirizzo di rete (Net-ID o Network-ID);
- indirizzo dell'host (Host-ID).

Gli indirizzi IP sono univoci a livello mondiale e assegnati da un'unica autorità, la ICANN (Internet Corporation for Assigned Names and Numbers). In realtà non viene assegnato l'indirizzo del singolo host ma quello della rete: sarà poi il gestore della rete a decidere quali indirizzi attribuire alle proprie macchine.

Fino al 1998 questa funzione era svolta dall'ente denominato IANA (*Internet Assigned Numbers Authority*), a ciò preposto con mandato governativo degli Stati Uniti.

Oggi per ottenere un indirizzo IP si hanno due possibilità:
- richiederlo all'autorità centrale di registro ICANN;
- acquistarlo da un rivenditore (tipicamente, un Internet provider).

L'indirizzo IP non identifica l'host fisico, come l'indirizzo MAC, ma la connessione di un host alla relativa rete: quindi se il PC viene connesso a un'altra rete il suo indirizzo IP cambia, ed è il caso tipico delle connessioni wireless dei dispositivi mobili, sia PC che smartphone.

Classi di indirizzi IP

A seconda del valore dei bit più significativi gli indirizzi IP sono suddivisi in classi, che si differenziano in base alle dimensioni della rete.

Classe	Bit più significativi
A	0
B	1 0
C	1 1 0
D	1 1 1 0
E	1 1 1 1 1

Le WAN hanno generalmente **IP-address** di **classe A**, mentre le LAN hanno **IP-address** di **classe B** o **C**.

Oltre che per il prefisso, le varie classi di indirizzi si distinguono per la diversa ripartizione dei bit tra l'identificativo della rete locale (Net-ID) e quello della scheda di rete (Host-ID).

Classe A	Net-ID	Host-ID		
Ottetto	1	2	3	4

Classe B	Net-ID		Host-ID	
Ottetto	1	2	3	4

Classe C	Net-ID			Host-ID
Ottetto	1	2	3	4

Classe D	Host-ID			
Ottetto	1	2	3	4

ESEMPIO

Nella classe C abbiamo a disposizione solo un byte per indirizzare gli host in quanto i primi 24 bit individuano la rete (3 fissi + 21 di rete): questo ci impone una limitazione sul numero di interfacce che possono essere presenti nella rete, cioè 256.

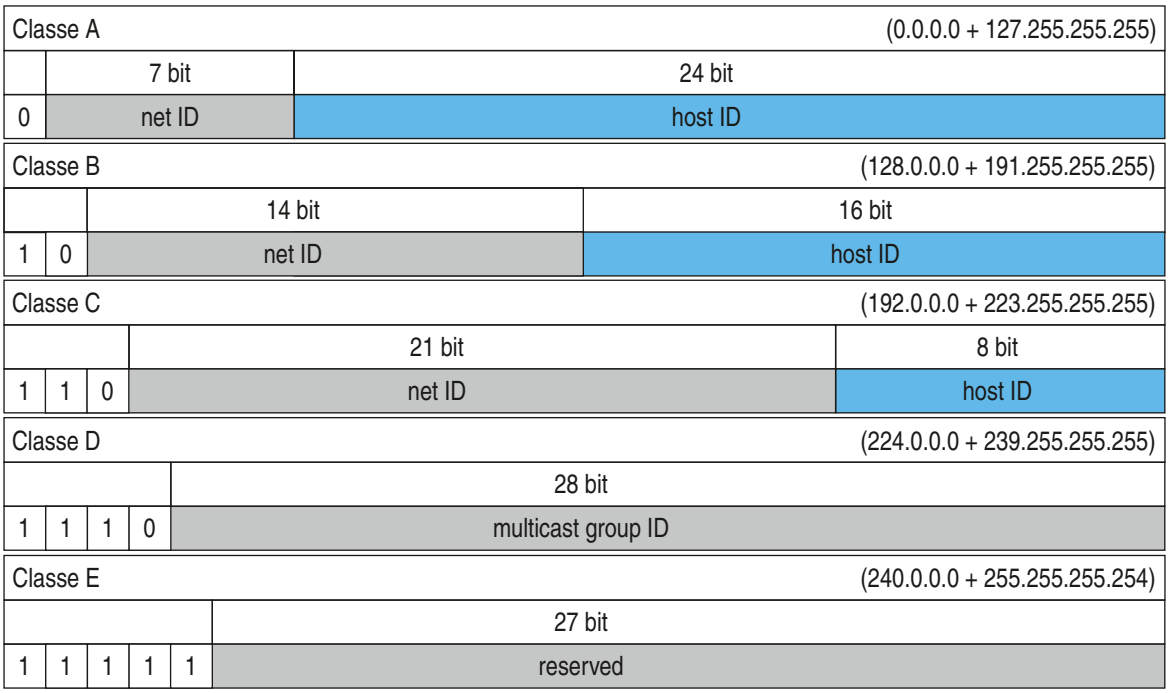
È quindi necessario scegliere l'indirizzo di rete in funzione del numero di macchine che devono essere connesse:

- classe A: sono indirizzi utilizzati in reti che hanno un numero cospicuo di host, in quanto il campo dell'Host-ID è di 24 bit e può pertanto identificare circa 16 milioni di host. I 7 bit dedicati al Net-ID permettono di definire solo 128 reti di classe A in tutto il mondo;
- classe B: sono adatti per reti di dimensioni intermedie, dato che agli host sono dedicati circa 64.000 indirizzi e che sono disponibili circa 16.000 reti di questa dimensione (14 bit di indirizzo di rete);
- classe C: sono i più utilizzati e permettono di definire reti di non più di 256 host; nonostante questa limitazione, il numero delle reti è elevato dato che vengono usati 21 bit nel Net-ID;
- classe D: sono dedicati al multicasting (RFC 1112);
- classe E: sono riservati per usi futuri.

Traduciamo in dotted-decimal il range di indirizzi di ogni classe e riportiamo il risultato in una tabella:

Classe A	da 0.0.0.0	a 127.255.255.255
Classe B	da 128.0.0.0	a 191.255.255.255
Classe C	da 192.0.0.0	a 223.255.255.255
Classe D	da 224.0.0.0	a 239.255.255.255
Classe E	da 240.0.0.0	a 255.255.255.255

Riassumiamo in uno schema riepilogativo la lunghezza degli indirizzi evidenziando i valori dei bit più significativi che ci permettono di individuare la classe di appartenenza degli indirizzi IP e il range degli indirizzi di ogni classe.



ESEMPIO

1 Individuare la classe di appartenenze del seguente indirizzo IP indicando l'indirizzo di rete e di host:

30.50.18.45

Traduciamo in binario il primo ottetto, ottenendo

$$\mathbf{30_{10}} = 0\mathbf{001\ 1110_2}$$

Dato che il primo bit è uguale a 0, siamo in presenza di un indirizzo di classe A.
L'indirizzo di rete è quindi composto da 7 bit: $001\ 1110_2$.
L'indirizzo dell'host è composto da 24 bit: $(50.18.45)_{10}$ = 00110010.0010010.00101101.

2 Individuare la classe di appartenenza del seguente indirizzo IP indicando l'indirizzo di rete e di host:

200.50.18.2

Traduciamo in binario il primo ottetto, ottenendo

$$\mathbf{200_{10}} = 11\mathbf{00\ 1000_2}$$

Dato che i primi due bit sono uguali a 1 e il terzo a 0, siamo in presenza di un indirizzo di classe C.
L'indirizzo di rete è quindi composto da 21 bit: 01000.00110010.0010010.
L'indirizzo dell'host è composto da 8 bit: 2_{10} =00000010_2.

Quando viene assegnato a una rete il Net-ID, spetta all'amministratore della rete attribuire alle macchine i rispettivi valori di Host-ID (piano di indirizzamento).
Tutte le macchine che appartengono alla stessa rete **devono avere** lo stesso indirizzo di rete Net-ID, vale a dire che nel caso di classe C devono avere i primi 24 bit uguali.
Nelle prossime unità didattiche vedremo come effettuare la scelta degli indirizzi sia di rete che di host.

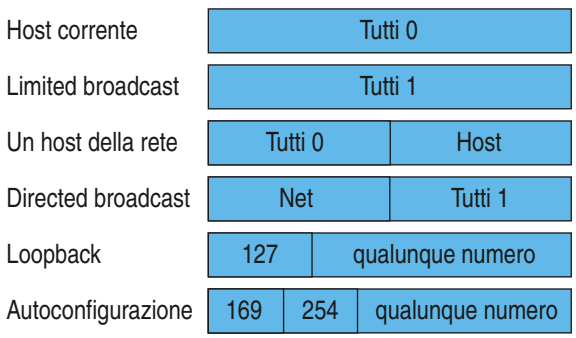
Se si vuole trasmettere un messaggio a tutti i PC di sottorete basta porre uguali a 1 tutti i bit dell'indirizzo di host: ogni macchina riconosce questo messaggio come messaggio di broadcast e ne legge il contenuto. Per estrarre l'indirizzo di rete da un indirizzo IP è sufficiente porre uguale a 0 l'indirizzo dell'host.

ESEMPIO

Dall'indirizzo precedente individuiamo il Net-ID e l'indirizzo di broadcasting:
- indirizzo di un host 200.50.18.2
- indirizzo di rete 200.50.18.0
- indirizzo di broadcast 200.50.18.255

Avendo utilizzato due indirizzi di host per scopi particolari, di fatto si è ridotto lo spazio di indirizzamento della parte di Host-ID di 2 unità: quindi in classe C si hanno solo 254 indirizzi disponibili.

Oltre ai due che abbiamo già considerato (0.x, che indica un certo Host-ID sulla rete corrente senza specificare il Net-ID, e x.1, che indica tutti gli host sulla rete corrente: directed broadcast), i documenti RFC 1700 e RCF 3927 segnalano anche altri indirizzi come riservati:
- 0.0.0.0 indica l'host corrente senza specificarne l'indirizzo;
- 255.255.255.255 è l'indirizzo di limited broadcast;
- 127.x.y.z è il loopback, che ridirige i datagrammi agli strati superiori dello stesso host;
- 169.254.x.y è riservato per l'autoconfigurazione degli host.



Reti IP private (RFC 1918)

La IANA ha riservato tre blocchi di indirizzi a reti IP private, ovvero reti IP che non sono connesse a Internet e di conseguenza neppure alle altre reti pubbliche: quando raggiunge un router presente su queste reti, il datagramma non viene instradato "fuori dalla rete" e perciò neppure inoltrato ad altri router.
Quindi gli host che hanno un indirizzo privato non sono raggiungibili da altri host oltre a quelli che appartengono alla stessa rete privata.

Si distinguono tre blocchi di indirizzi, ciascuno per ogni classe:
- il primo (10.0.0.0) rappresenta un'intera classe A;
- il secondo (172.16.0.0) è costituito dall'insieme di 16 reti di classe B contigue;
- il terzo (192.168.0.0) rappresenta 255 reti di classe C contigue.

Classe A	10.0.0.0	a 10.255.255.255
Classe B	172.16.0.0	a 172.31.255.255
Classe C	192.168.0.0	a 10.255.255.255

Questi indirizzi possono essere utilizzati ciascuno all'interno di una rete IP senza limitazioni, ma non sono direttamente raggiungibili da Internet né, viceversa, possono accedere direttamente a Internet.

Per connettere a Internet una rete con indirizzi privati occorre effettuare una traslazione di indirizzi da privati a pubblici chiamata NAT (*Network Address Translation*), che verrà analizzata in seguito.

Possiamo affermare che nel mondo esisteranno sicuramente migliaia (o milioni) di host con lo stesso indirizzo IP: la condizione è che si tratti di un indirizzo privato, disponibile solo nella rete di appartenenza e quindi non visibile da nessun'altra rete, né pubblica né privata.

I motivi per cui si ricorre all'indirizzamento IP privato sono essenzialmente due:

1. maggiore sicurezza: le macchine con indirizzo privato non sono direttamente raggiungibili da Internet e non possono quindi essere utilizzate da intrusi come bersaglio;
2. abbondanza di spazio di indirizzamento: come vedremo, gli indirizzi privati consentono di scegliere liberamente la tecnica di attribuzione degli stessi (o statica o dinamica), al contrario degli indirizzi registrati.

La scelta di utilizzare indirizzi privati offre il vantaggio di poter assegnare sia un indirizzo IP statico a ciascuna macchina sia, data l'abbondanza di indirizzi a disposizione, interfacce IP multiple allo stesso host.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Il protocollo TCP offre un servizio:
- ○ Connection-oriented, non affidabile
- ○ Connection-oriented, affidabile
- ○ Connectionless, non affidabile
- ○ Connectionless, non affidabile

2 Gli indirizzi IP sono costituiti da:
- ○ 32 byte
- ○ 8 byte
- ○ 4 byte
- ○ 2 byte

3 La classe di indirizzi IP che consente di associare a una data rete il massimo numero di host è:
- ○ la classe A
- ○ la classe B
- ○ la classe C
- ○ la classe D

4 Gli indirizzi di rete
- ○ non possono essere assegnati agli host
- ○ in caso di necessità possono essere assegnati a un host
- ○ in caso di necessità possono essere assegnati a più host
- ○ in caso di necessità possono essere assegnati a un router

5 Quale autorità si occupa dell'assegnazione degli IP?
- ○ *Internet Assigned Numbers Authority* (IANA)
- ○ *World Wide Web Consortium* (W3C)
- ○ *University of California, Los Angeles* (UCLA)
- ○ *Internet Corporation for Assigned Names and Numbers* (ICANN)

6 Com'è strutturato, in binario, un indirizzo di classe B?
- ○ 10 network (14) host (16)
- ○ 01 network (14) host (16)
- ○ 110 network (13) host (16)
- ○ 110 network (14) host (15)

7 Qual è il range, in decimale e in binario, del primo ottetto di tutti i possibili indirizzi di rete di classe B?
- ○ 128-190 10000000-10111110
- ○ 128-191 10000000-10111111
- ○ 129-190 10000001-10111110
- ○ 129-191 10000001-10111111

8 Quale tra i seguenti indirizzi è di una rete privata di classe B?
- ○ 72.16.11.23
- ○ 127.16.11.23
- ○ 169.16.11.23
- ○ 172.16.11.23
- ○ 192.168.1.23

Test vero/falso

1	Un servizio connectionless non è in grado di garantire il controllo del flusso.	V F
2	In un servizio connection-oriented non viene rispettato l'ordine dei dati.	V F
3	Il livello 3 del TCP/IP offre un servizio di tipo connectionless.	V F
4	In un servizio connection-oriented l'indirizzo di destinazione è stabilito al setup della connessione.	V F
5	In un servizio affidabile non c'è riscontro dell'avvenuta ricezione dei pacchetti.	V F
6	Un servizio datagram e connection-oriented non è affidabile.	V F
7	Gli indirizzi IP vengono assegnati a tutte le interfacce degli host e dei router.	V F
8	A ogni sottorete viene assegnato un indirizzo IP.	V F
9	Il funzionamento base di IP prevede che a ogni rete fisica corrisponda una rete o sottorete.	V F
10	Senza la suddivisione degli indirizzi in classi non sarebbe possibile indirizzare tutti gli host.	V F
11	Il protocollo TCP non garantisce la ricezione dei messaggi nello stesso ordine con cui vengono trasmessi.	V F
12	Un servizio datagram non è orientato alla connessione.	V F

Verifichiamo le competenze

Esprimi la tua creatività

1 Indica a quale livello della pila ISO/OSI si collocano i seguenti protocolli.

DNS .....

IEEE 802.3 .....

TCP .....

IP .....

UDP .....

ICMP .....

2 Identifica la classe a cui appartengono i seguenti indirizzi IP, dopo averli convertiti in notazione binaria.

229.94.110.51 .....

101.123.5.45 .....

231.201.5.45 .....

128.23.45.4 .....

192.168.20.3 .....

193.242.100.255 .....

3 Indica se le seguenti coppie di indirizzi IP appartengono o no alla stessa rete.

130.186.1.2 e 130.186.2.3 .....

150.110.1.2 e 158.120.2.3 .....

197.110.1.2 e 197.120.1.2 .....

10.10.1.2 e 10.10.2.3 .....

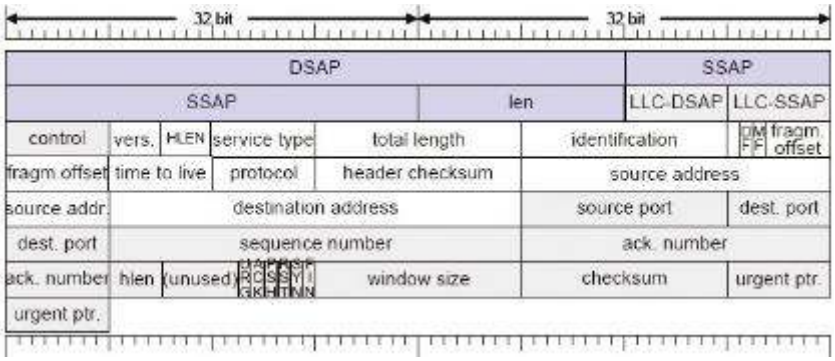
150.110.1.2 e 158.110.2.3 .....

197.110.1.2 e 197.110.2.3 .....

4 Uno "sniffer" ha catturato il seguente pacchetto:

```
08 00 2B 00 9A 7D 08 00 5A 1A 6C BE 00 2F 06 06 03 45 00 00 2C 58 E3 00 00 3C 06 18 53 82 C0 04
12 82 C0 04 04 04 14 00 17 33 A3 26 2D 56 67 24 01 60 12 40 00 72 2D 00 00 02 04 05 AE
```

Aiutandoti con lo schema di imbustamento multiplo rappresentato nella figura, rispondi alle seguenti domande.



a) Qual è l'indirizzo MAC del mittente? .....

b) Qual è l'indirizzo MAC del destinatario? .....

c) Qual è l'indirizzo IP del mittente? .....

d) Qual è l'indirizzo IP del destinatario? .....

UNITÀ DIDATTICA 2

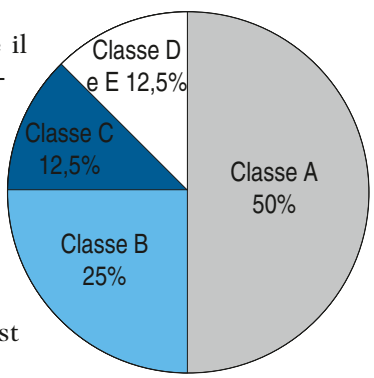
INTRODUZIONE AL SUBNETTING

IN QUESTA UNITÀ IMPAREREMO...

- il concetto di sottorete
- a definire le Subnet-mask
- ad assegnare gli indirizzi agli host

IPv4 e IPv6

In sede di progetto del TCP/IP non si poteva certo prevedere che il protocollo sarebbe diventato lo standard mondiale per l'interconnessione dei personal computer; ecco perché oggi ci troviamo in una situazione di carenza di indirizzi, anche a causa dell'aumento delle utenze dei paesi dell'Est.
Le classi A e B sono poche, circa 17.000, ma detengono il 75% dello spazio di indirizzamento possibile di IPv4, l'indirizzamento composto da 4 byte, cioè 32 bit.
I possibili indirizzi di classe C sono solo il 12,5% circa degli indirizzi utili e sono i più richiesti per realtà con un numero di host inferiore a 24. ▶



◀ L'IETF è una grande comunità internazionale aperta composta da progettisti di reti, operatori, produttori e ricercatori interessati all'evoluzione dell'architettura Internet e al suo mantenimento organico e uniforme: vi possono aderire tutti coloro che sono interessati, liberamente. ▶

Già dal 1992, anno in cui è iniziata la diffusione "massiccia" di Internet, l'◀ IETF ▶ (*Internet Engineering Task Force*), a fronte dell'imminente esaurimento degli indirizzi IP disponibili per l'assegnazione, ha introdotto alcune tecniche per migliorare l'efficienza di IPv4: sono state sviluppate le Subnet-mask e il CIDR (*Classless InterDomain Routing*), che inizieremo ad analizzare in questa unità didattica.
Il futuro, però, è rappresentato da un nuovo protocollo IP, che già inizia a diffondersi con i nuovi sistemi operativi: si tratta di una versione più estesa e scalabile di IP, detta IP versione 6 (IPv6), che usa 128 bit anziché i 32 di IPv4 e mantiene comunque la compatibilità con la versione 4.

Subnetting: generalità

Le tre classi nelle quali sono suddivise le reti vincolano a usare dimensioni prefissate in termini di indirizzi disponibili e non possono certo soddisfare le esigenze di tutte le possibili realtà di LAN: in molti casi una rete di classe A o B è troppo grande (molti indirizzi inutilizzati) e una di classe C troppo piccola.

Classe A: 2^{24} = 6.777.216
Classe B: 2^{16} = 65.536
Classe C: 2^{8} = 256

Inoltre, le grandi organizzazioni hanno l'esigenza di distribuire gli indirizzi su molte reti fisiche distinte e/o distanti tra loro.

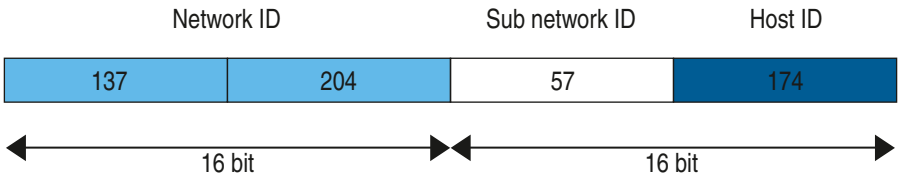
L'RFC 950 ha definito una procedura che permette di scomporre le tre classi standard in sottoreti, cioè in reti di dimensioni minori di quelle previste dalle classi A, B, C: questo processo prende il nome di subnetting.

SUBNETTING

Il subnetting è l'operazione che viene effettuata sul campo host suddividendolo in due parti per organizzare la rete in parti più piccole: le **sottoreti**.

Con il subnetting abbiamo a disposizione un altro metodo per gestire gli indirizzi IP dividendo gli indirizzi di rete "in pezzi più piccoli": all'esterno delle LAN la suddivisione non è visibile.

Un esempio di indirizzo di classe B con subnetting è il seguente:



L'indirizzo di network rimane invariato mentre i 16 bit dell'indirizzo di host in questo caso sono stati "spezzati" in due parti di 8 bit:

- la prima parte, cioè i primi 8 bit, identifica una porzione della rete in questione (57);
- la seconda identifica il singolo host della sottorete (174).

Subnet-mask

Fare delle subnet in una rete consiste nell'usare una Subnet-mask per dividere una rete grande in segmenti più piccoli, più efficienti e più maneggevoli.

SUBNET-MASK

La Subnet-mask è una sequenza di bit posti al valore 1 in corrispondenza dei bit dedicati al Network-ID e al Subnetwork-ID, mentre i bit che corrispondono all'Host-ID sono posti a 0.

Per l'esempio precedente la Subnet-mask è la seguente:

- Binaria :11111111.11111111.11111111.00000000
- Dotted decimal :255.255.255.0
- ◀ CIDR ▶ : /24 (è il numero di bit a 1 della Net-mask)

◀ Il CIDR (*Classless Inter-Domain Routing*) è il metodo che ha soppiantato l'utilizzo delle classi ed è oggi impiegato per indicare gli indirizzi IP: lo analizzeremo più approfonditamente nella prossima unità didattica, ma fin da ora iniziamo ad adottarne la comoda notazione che indica la lunghezza degli "1 nella maschera". ▶

Le Subnet-mask di default delle tre classi di indirizzi sono le seguenti:
- classe A: 255.0.0.0 /8;
- classe B: 255.255.0.0/16;
- classe C: 255.255.255.0/24.

Applicando le maschere di default si ottiene un prefisso equivalente al Network-ID di classe; mediante il subnetting è possibile definire Subnet-mask con qualunque numero di 1, come per esempio:
- 255.128.0.0 /9 è una sottorete derivata da una classe A;
- 255.255.224.0/19 è una sottorete derivata da una classe B;
- 255.255.255.192/26 è una sottorete derivata da una classe C.

La maschera che estrae l'indirizzo di rete viene chiamata Subnet-mask quando si effettua un subnetting e Net-mask in caso di supernetting, come vedremo nella prossima unità didattica.

Zoom su...

A CHE COSA SERVE LA SUBNET-MASK

La Subnet-mask permette di ricavare immediatamente l'indirizzo di rete da un indirizzo IP mediante un'operazione di AND bit a bit tra l'indirizzo IP e la maschera stessa.
Consideriamo per esempio l'indirizzo IP di un host **10.40.80.3** con Subnet-mask **255.0.0.0**.
Scriviamo i due valori in binario:

10.40.60.3 = 00001010.00101000.01010000.00000011
255. 0.0.0 = 11111111. 00000000. 00000000.00000000

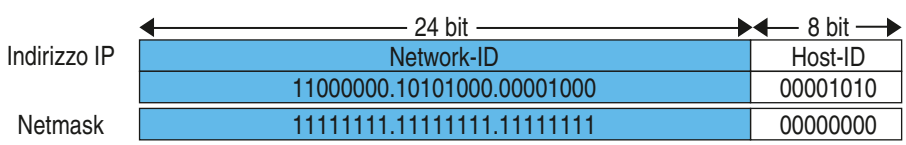
ed eseguiamo l'operazione di AND bit a bit:

00001010.00101000.01010000.00000011
11111111. 00000000. 00000000.00000000
==========
00001010. **00000000. 00000000.00000000**

Otteniamo 10.**0.0.0**, che è l'indirizzo della rete di classe A sul quale è collegato l'host.

ESEMPIO 1

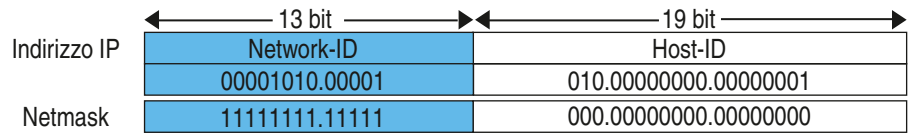
192.168.8.10/24 (255.255.255.0)



Network-ID = 192.168.8.0

ESEMPIO 2

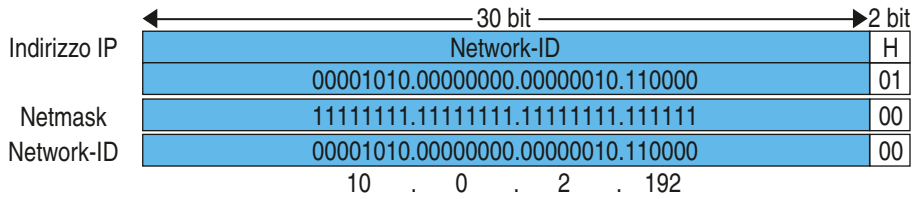
10.10.0.1/13 (255.248.0.0)



Network-ID = 10.0.0.0

ESEMPIO 3

10.0.2.200/30 (255.255.255.252)



Network-ID = 10.0.2.192/30

Formato della Subnet-mask

Per definizione una Subnet-mask ha sempre tutti i bit a 1 a sinistra e tutti quelli a 0 a destra: non esistono Subnet-mask con degli 0 tra gli 1. La seguente Subnet-mask

11111011.11011111.11100000.00010000

non è valida!

Questa imposizione pone un limite al numero delle sottoreti in quanto i singoli byte di una Subnet-mask non possono assumere tutti i 256 valori possibili, ma solo 9.

–00000000 = 0
–10000000 = 128
–11000000 = 192
–11100000 = 224
–11110000 = 240
–11111000 = 248
–11111100 = 252
–11111110 = 254
–11111111 = 255

Numero di host

Il numero massimo di host che una sottorete può contenere è $2^h - 2$, dove h è il numero di bit di Host-ID: due indirizzi vengono esclusi in quanto sono riservati all'indirizzo che identifica la sottorete stessa (tutti i bit a 0, "all 0") e all'indirizzo broadcast della sottorete (tutti i bit a 1, "all 1"). Inoltre nelle sottoreti non isolate, almeno un indirizzo deve essere assegnato al Default Gateway, che è il dispositivo (router) verso cui instradare tutto il traffico diretto al di fuori della sottorete; generalmente ad esso si assegna come indirizzo quello precedente all'indirizzo di broadcast.

ESEMPIO 4

Avendo come Subnet-mask 255.255.255.0 il numero di bit di Host-ID è uguale a 8, $2^8 - 2 = 254$ host:
- 137.200.69.0/24 identifica la sottorete 69 di una rete di classe B;
- 137.200.69.255/24 è l'indirizzo broadcast di tale sottorete;
- 137.200.69.254 /24 è l'indirizzo del suo default gateway;
- il range di indirizzi validi per gli host va da 137.200.69.1/24 a 137.200.69.253/24.

Numero di sottoreti

Calcoliamo il numero delle sottoreti possibili: in passato erano stati dichiarati riservati gli indirizzi composti da tutti 1 e tutti 0, per gli host come per i Subnet-ID, ma successivamente questo vincolo è stato tolto e ora viene consentito l'utilizzo di tutti i possibili Subnet-ID (RFC 1878); ne consegue che il numero delle sottoreti è direttamente calcolato da 2^s, dove s è il numero di bit di Subnet-ID.

Riprendiamo l'indirizzo dell'esempio precedente, dove viene utilizzato il terzo byte come indirizzo di sottorete, quindi 8 bit: sarà possibile avere $2^8 = 255$ diverse sottoreti in classe B con range di indirizzi.
- 137.200.0.0/24 identifica la prima sottorete;
- 137.200.1.0/24 identifica la seconda sottorete;
- …
- 137.200.255.0/24 identifica l'ultima sottorete.

ESEMPIO 5

Dati i seguenti indirizzi IP, individuare i rispettivi numeri di sottoreti possibili per ogni rete.
a) Host-ID = 192.168.10.200 con Subnet-mask **255.255.255.192**
Iniziamo col tradurre in binario la Subnet-mask:

11111111.11111111.11111111.11000000

Contando il numero degli uno possiamo anche scrivere l'indirizzo in questo modo

192.160.10.200/26

e individuare l'indirizzo di rete:

Network-ID + Subnet-ID = 192.168.10.192/26

Poiché l'indirizzo è di classe C, su 26 bit i primi 24 sono riservati alla rete e 2 alla sottorete; perciò sono possibili quattro sottoreti a 26 bit, con i seguenti indirizzi:
- I indirizzo di sottorete : x.y.z.0000 0000 = 192.168.10.0/26
- II indirizzo di sottorete : x.y.z.0100 0000 = 192.168.10.64/26
- III indirizzo di sottorete : x.y.z.1000 0000 = 192.168.10.128/26
- IV indirizzo di sottorete : x.y.z.1100 0000 = 192.168.10.192/26

b) Host-ID = 10.143.0.0 con Subnet-mask **255.128.0.0**
Iniziamo col tradurre in binario la Subnet-mask:

11111111.10000000.00000000.00000000

contando il numero degli uno possiamo anche scrivere l'indirizzo nel modo seguente

10.143.0.0/9

e individuare l'indirizzo di rete:

Network-ID + Subnet-ID = 10.128.0.0/9

Poiché l'indirizzo è di classe A, su 9 bit i primi 8 sono riservati alla rete e 1 alle sottoreti; perciò sono possibili due sottoreti a 9 bit, con i seguenti indirizzi:

- I indirizzo di sottorete : x.0000 0000.0000000.00000000 = 10.0.0.0/9
- II indirizzo di sottorete : x.1000 0000.0000000.00000000 = 10.128.0.0/9

Dagli esempi precedenti possiamo osservare come a seconda del numero di bit riservati al Subnet-ID la rete originaria venga suddivisa per multipli di 2:

- 1 bit: si divide la rete a metà, quindi in due sottoreti ciascuna con il 50% degli indirizzi per gli host;
- 2 bit: si divide la rete in 4 parti, ciascuna con il 25% degli indirizzi per gli host;
- 3 bit: si divide la rete in 8 parti, ciascuna con il 12,5% degli indirizzi per gli host;
- ecc.

Assegnazione degli indirizzi

Con il subnetting abbiamo quindi scomposto un indirizzo IP in tre parti:

- un campo network;
- un campo subnet;
- un campo host.

I campi subnet e host vengono creati dal campo host originario dell'intera rete.

L'assegnazione degli indirizzi prende il nome di piano di indirizzamento e l'amministratore trasferisce dei bit dal campo host originale al campo subnet; il numero minimo di bit che si possono spostare è 1 e bisogna lasciare almeno 2 bit nella parte host.
Gli host che appartengono alla stessa sottorete comunicano direttamente (direct delivery, cioè consegna diretta) mentre per comunicare con altre sottoreti oppure LAN è necessario utilizzare un gateway (indirect delivery, cioè consegna indiretta) al quale viene assegnato l'indirizzo immediatamente precedente rispetto a quello di broadcast (ricordiamo che l'indirizzo broadcast, costituito da tutti i bit a 1, serve per raggiungere contemporaneamente tutti gli host di una rete o sottorete IP).

Di seguito iniziamo a pianificare gli indirizzi partendo dal Network-ID.

1. Richiediamo all'*autorità di registro* oppure a un service provider un Network-ID della classe che più si avvicina per dimensione al numero di computer che compongono la nostra rete (per esempio, se abbiamo 30 PC richiederemo un Network-ID di classe C come 196.69.160.x).
2. Attribuiamo l'indirizzo di gateway (nell'esempio 196.69.16.254).
3. Attribuiamo progressivamente alle varie macchine collegate alla rete gli Host-ID disponibili (nell'esempio, da 196.69.16.1 a 196.69.16.253).

In questo modo si sprecano molti indirizzi, in quanto viene attribuito un prefisso di classe anche a reti con pochi host; è quindi doveroso porsi due domande:

1. è necessario che ogni host abbia un indirizzo IP pubblico assegnato in modo statico o è sufficiente che lo abbiano solo gli host attivi?
2. tutti gli host della rete IP devono essere visibili su Internet oppure l'esigenza principale è che possano comunicare tra loro?

Entrambe le domande trovano una soluzione passando a un piano di indirizzamento che utilizzi un indirizzo di rete privato, e non pubblico, e in un secondo tempo sia dotato di dispositivi che permettano di attribuire dinamicamente gli indirizzi agli host che devono collegarsi a Internet.

Utilizziamo quindi come indirizzo di classe C 198.162.1.0/24 e attribuiamo progressivamente alle varie macchine collegate alla rete gli Host-ID disponibili: non abbiamo problemi di sprechi in quanto stiamo utilizzando un indirizzo privato e abbiamo "gratuitamente" tutti gli indirizzi che ci servono.

Configuriamo quindi ogni host connesso alla rete IP con tre parametri essenziali:

1. l'indirizzo IP completo dell'host, scelto tra quelli disponibili da 198.162.1.0 a 198.162.1.255, senza considerare i due estremi che sono riservati;
2. la Subnet-mask, che permette l'estrazione del prefisso di sottorete (255.255.255.0);
3. il gateway predefinito, al quale saranno inviati i pacchetti IP i cui destinatari non si trovano sulla stessa subnet del mittente (198.162.1.254).

Se il gateway è assente nella rete la comunicazione è limitata alla sola subnet in questione.

Partizionare una rete

Il partizionamento di una rete in sottoreti deve essere effettuato utilizzando maschere della stessa lunghezza; di conseguenza, occorre stabilire la lunghezza in base alla sottorete che deve indirizzare il numero maggiore di host.

ESEMPIO 6

Un'azienda possiede tre sedi **S1**, **S2**, **S3** (due filiali e un magazzino) in un'area urbana; ciascuna sede è dotata di una LAN interna e di un router di uscita verso il mondo esterno.
Sapendo che gli host sono 50 nella sede S1, 40 nella S2 e solo 20 nella S3 e che le tre sedi devono essere interconnesse tra loro attraverso una rete MAN a maglia completa M, come quella riportata nella figura, si chiede di progettare una rete di classe C a cui viene assegnato l'indirizzo 200.69.96.0.

Lo schema generale dell'architettura del sistema è quello della figura a lato. ▶
La sottorete con **numero maggiore** di host è la sede S1, per la quale sono necessari 6 bit di indirizzi (2^6 = 64): utilizziamo quindi come terzo byte della maschera di sottorete **11**000000, ottenendo la seguente Subnet-mask

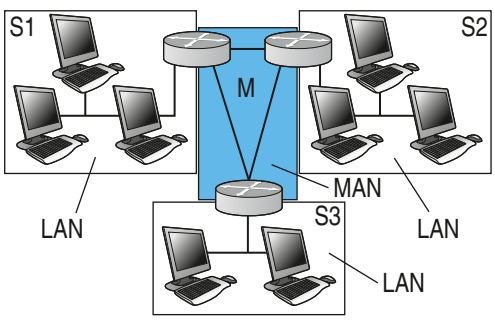
Subnet-mask: 255.255.255.192

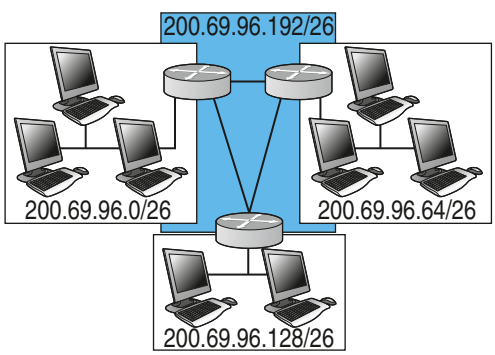
che ci permette di definire 4 sottoreti, delle quali tre saranno utilizzate per le LAN e la quarta per interconnetterle in una MAN: ▶

S1: 200.69.96.0/26
S2: 200.69.96.64/26
S3: 200.69.96.128/26
M: 200.69.96.192/26

Il primo passo consiste nell'assegnare gli indirizzi di broadcast:

Broadcast S1: 200.69.96.63/26
Broadcast S2: 200.69.96.127/26
Broadcast S3: 200.69.96.191/26
Broadcast M: 200.69.96.255/26





e quindi gli indirizzi dei router verso le LAN:

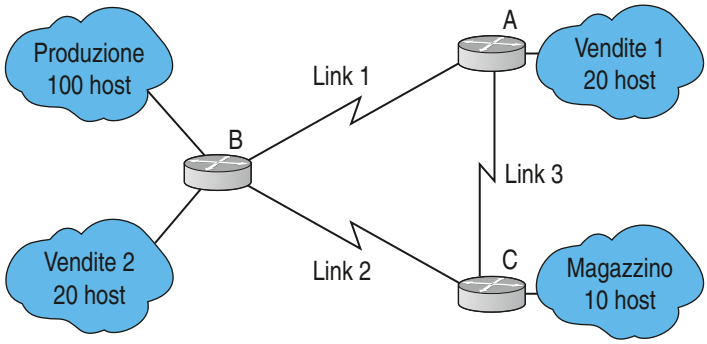
Router S1: 200.69.96.62/26
Router S2: 200.69.96.126/26
Router S3: 200.69.96.190/26

Gli indirizzi dei router della MAN e gli indirizzi degli host possono essere scelti nell'ambito dei seguenti range:

host S1: tra 200.69.96.1 e .61
host S2: tra 200.69.96.65 e .125
host S3: tra 200.69.96.129 e .189
router M: tra 200.69.96.193 e .254

ESEMPIO 7

Supponiamo di dover collegare tra loro quattro sedi di un'azienda: le richieste di host sono disomogenee e le macchine da connettere, che si trovano per la maggior parte nella sede di produzione, sono circa 100.



Da questo valore dipende la lunghezza della maschera di sottorete che, dovendo avere bit sufficienti per assegnare gli indirizzi agli host, sarà quindi 10000000.
Tuttavia, per avere quattro reti di 128 host non basta un indirizzo di classe C, che ne offre solamente due: si deve salire alla classe B.
In classe B si ha un numero di indirizzi di 2^{16} (64 K), che sono decisamente sprecati per il nostro fabbisogno appena superiore a 150 macchine.
È necessario trovare una soluzione alternativa.

■ Appendice: tabelle per i subnetting (RCF1878)

Creazione di subnet partendo da un indirizzo e da una maschera di default di classe A.

Bit aggiuntivi necessari	Sottoreti massime	Numero massimo di host per sottorete	Maschera
(n)	(2^n)	($2^{(24-n)} - 2$)	
0	0	16.777.214	255.0.0.0
1	2	8.388.604	255.128.0.0
2	4	4.194.302	255.192.0.0
3	8	2.097.150	255.224.0.0

Bit aggiuntivi necessari	Sottoreti massime	Numero massimo di host per sottorete	Maschera
4	16	1.048.574	255.240.0.0
5	32	524.286	255.248.0.0
6	64	262.142	255.252.0.0
7	128	131.070	255.254.0.0
8	256	65.534	255.255.0.0

Creazione di subnet partendo da un indirizzo e da una maschera di default di classe B.

Bit aggiuntivi necessari	Sottoreti massime	Numero massimo di host per sottorete	Maschera
(n)	(2^n)	$(2^{(16-n)} - 2)$	
0	0	65.534	255.255.0.0
1	2	32.764	255.255.128.0
2	4	16.382	255.255.192.0
3	8	8190	255.255.224.0
4	16	4094	255.255.240.0
5	32	2046	255.255.248.0
6	64	1022	255.255.252.0
7	128	510	255.255.254.0
8	256	254	255.255.255.0

Creazione di subnet partendo da un indirizzo e da una maschera di default di classe C.

Bit aggiuntivi necessari	Sottoreti massime	Numero massimo di host per sottorete	Maschera
(n)	(2^n)	$(26^{(8-n)} - 2)$	
0	0	254	255.255.255.0
1	2	128	255.255.255.128
2	4	62	255.255.255.192
3	8	30	255.255.255.224
4	16	14	255.255.255.240
5	32	6	255.255.255.248
6	64	2	255.255.255.252
7	non valido	non valido	255.255.255.254
8	non valido	non valido	255.255.255.255

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **La Subnet-mask serve**

- ○ per nascondere l'indirizzo di rete di un host
- ○ per definire l'ampiezza dei campi Net-ID e Host-ID in un indirizzo IP
- ○ per modificare l'indirizzo di una rete
- ○ per nascondere l'indirizzo di una rete

2 **Con quale classe o quali classi di indirizzi può essere utilizzata la Net-mask 255.255.192.0?**

- ○ Classe A
- ○ Classe B
- ○ Classe A e classe B
- ○ Classe C
- ○ Classe B e classe C

3 **Quale di queste Subnet-mask non è valida?**

- ○ 255.255.255.224
- ○ 255.255.255.192
- ○ 255.255.255.216
- ○ 255.255.255.252

4 **Devi suddividere la rete di classe B in 30 sottoreti: qual è la Subnet-mask?**

- ○ 255.255.192.0
- ○ 255.255.224.0
- ○ 255.255.248.0
- ○ 255.255.252.0

5 **Devi suddividere la rete di classe C in sottoreti con al massimo 30 PC: qual è la Subnet-mask?**

- ○ 255.255.255.224
- ○ 255.255.255.192
- ○ 255.255.255.216
- ○ 255.255.255.252

6 **Qual è l'indirizzo di broadcast della rete 198.16.211.0 con Subnet-mask 255.155.155.192?**

- ○ 198.16.211.63
- ○ 198.16.211.64
- ○ 198.16.211.254
- ○ 198.16.211.255

7 **Devi suddividere la rete 128.0.0.0 in 62 sottoreti: qual è la Subnet-mask?**

- ○ 255.255.255.192
- ○ 255.255.255.216
- ○ 255.255.255.252
- ○ Non è possibile

8 **Nella rete 198.111.24.0 la Subnet-mask è 255.255.255.192. Quali coppie di indirizzi non appartengono alla stessa subnet?**

- ○ 198.111.24.6 e 198.111.24.9
- ○ 198.111.24.26 e 198.111.24.29
- ○ 198.111.24.126 e 98.111.24.129
- ○ 198.111.24.226 e 198.111.24.229

Verifichiamo le competenze

Esprimi la tua creatività

1 Partendo dalla maschera di sottorete di un indirizzo di classe C 255.255.255.0 e operando su questa con subnetting avente maschera fissa, quante sottoreti si possono ottenere?

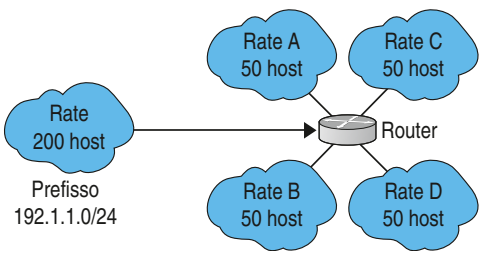
Subnet-mask (dot notation – binario)	N. reti	N. host

2 Supponi di sapere che nella stessa rete 158.110.0.0 la Subnet-mask è 255.155.192.0.
Dopo aver definito tutti gli indirizzi delle diverse sottoreti, indica se 158.110.191.3 e 158.110.192.3 appartengono alla stessa sottorete.

Subnet	Indirizzo della subnet	Range di indirizzi validi	Indirizzo di broadcast
1			
2			
3			
4			

3 Scomponi una rete di 200 PC in quattro sottoreti di 50 PC. ▶

Indirizzo di rete di classe C (Net-ID) 192.1.1.x

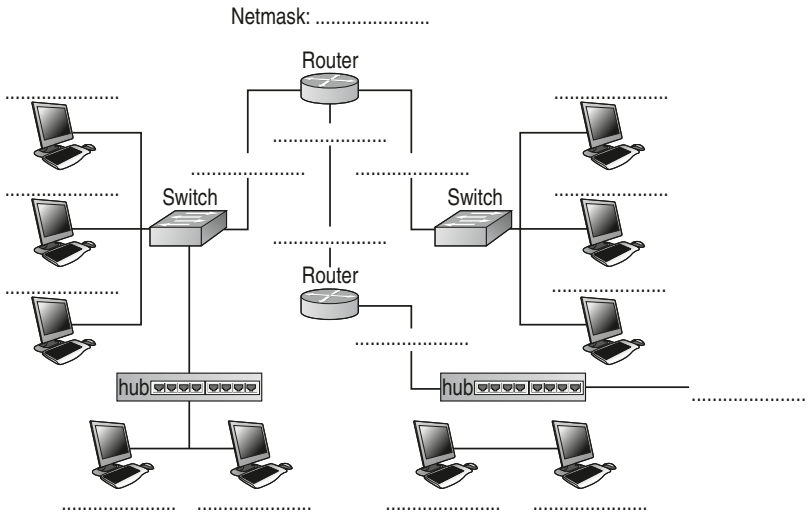


4 Definisci il piano di indirizzamento della rete riportata nella figura a lato sapendo che: ▶

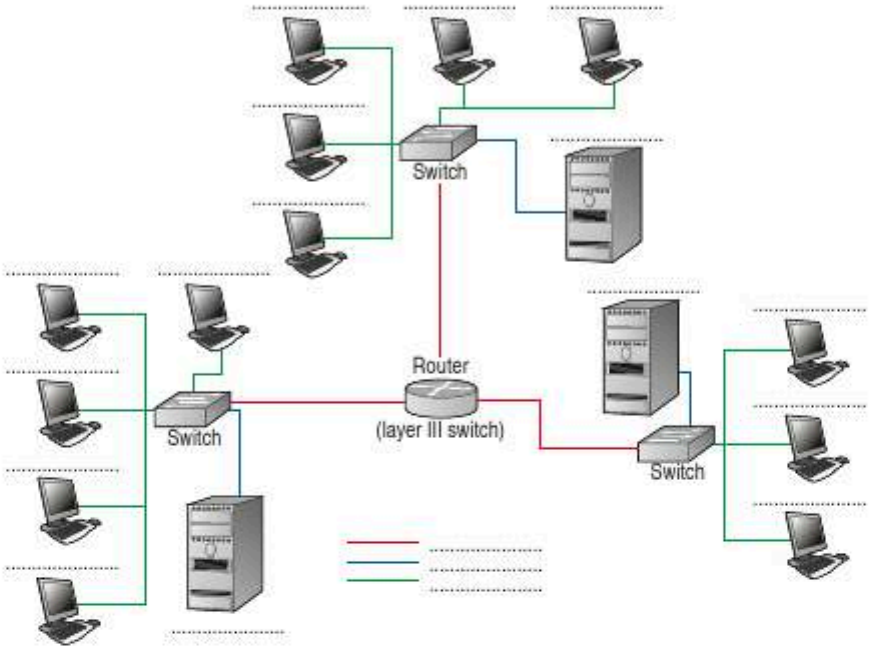
– è a disposizione un indirizzo di classe C 200.200.200.0;

– in nessuna sottorete ci sono al massimo 14 elaboratori;

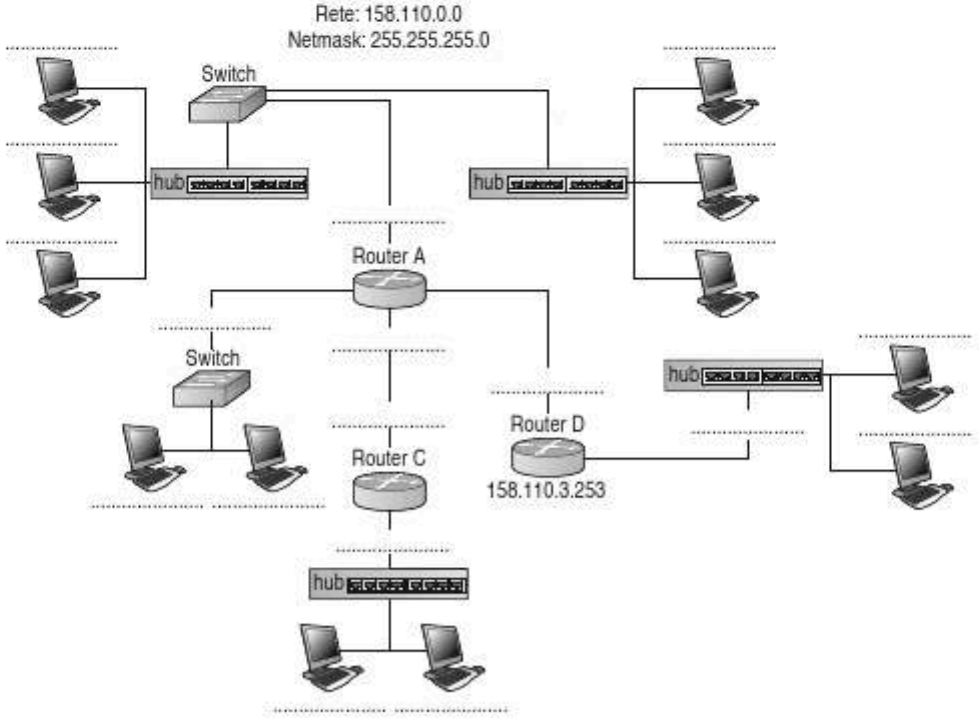
– la rete potrà crescere fino a un massimo di 10 sottoreti connesse con 2 soli router.



5 Una rete suddivisa in 3 reti locali ha 15 elaboratori client e 3 server: evidenzia i tipi di apparecchiature utilizzate e i mezzi trasmissivi, quindi assegna tutti gli indirizzi IP necessari partendo da una rete 130.186.0.0.



6 Considerando la rete aziendale rappresentata nello schema seguente:



Indica un possibile piano di indirizzamento completo.

UNITÀ DIDATTICA 3

SUBNETTING: VLSM E CIDR

IN QUESTA UNITÀ IMPAREREMO...

- il concetto di Subnet-mask variabile
- la tecnica CIDR e le super-reti

VLSM

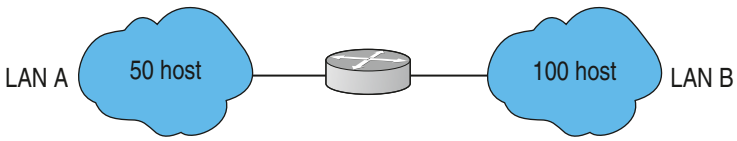
Abbiamo visto che dover utilizzare una Subnet-mask di lunghezza fissa per ogni indirizzo di rete pone un grosso limite al subnetting tradizionale: una volta scelta la Subnet-mask, si è vincolati a un numero fisso di sottoreti aventi tutte le stesse dimensioni, generalmente con un enorme spreco di indirizzi oltre a un'estrema rigidità in caso di evoluzioni della rete stessa, che potrebbe comportare il rifacimento completo del piano di indirizzamento se anche solo una subnet crescesse al di sopra del numero massimo di host previsti.

Nel 1987 l'RFC 1009 ha specificato come utilizzare più di una Subnet-mask all'interno della stessa rete: se in una rete viene assegnata più di una Subnet-mask, questa viene considerata una rete con maschere di lunghezza variabile, cioè una VLSM (*Variable Length Subnet-Mask*).

Vediamo un primo esempio dove abbiamo solo due sottoreti.

ESEMPIO 8

La richiesta è quella di suddividere la rete in due sottoreti, LAN A e LAN B, rispettivamente di 50 e 100 host ciascuna, a partire da un rete privata con indirizzo 192.168.1.0/24, quindi di classe C.



Poiché la LAN A necessita di 50 indirizzi per gli host, bastano 64 indirizzi host (che divengono 61 togliendo all-1, all-0 e l'indirizzo di gateway): quindi 6 bit.
L'indirizzo di sottorete valido per la LAN A è perciò x/26.

La LAN B necessita di 100 indirizzi per gli host, quindi ne servono 125, ottenibili con 7 bit. L'indirizzo di sottorete valido per la LAN B è perciò y/25.

> Se in una rete sono presenti sottoreti di dimensioni differenti tra loro esse comunicano attraverso un router (o gateway).

Dato che nelle due sottoreti non possono esserci host con il medesimo indirizzo IP, l'amministratore deve accuratamente assegnare gli indirizzi di sottorete, a partire da quelli con il prefisso più corto.

Per la LAN A abbiamo a disposizione due possibili indirizzi:
- I indirizzo di sottorete : x.y.z.0000 0000 = 192.168.1.0/27
- II indirizzo di sottorete : x.y.z.1000 0000 = 192.168.1.128/27

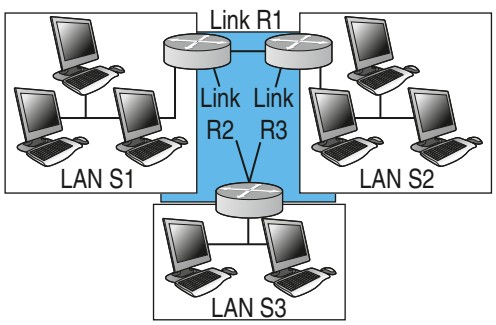
Se assegniamo il primo indirizzo, quello con il bit più significativo **uguale a 0**, al momento di scegliere gli indirizzi di sottorete per la LAN B dovremo escludere gli indirizzi che hanno tale bit uguale a 0, altrimenti avremmo nella rete qualche host con lo stesso indirizzo IP; scartiamo i primi due indirizzi:
- ~~I indirizzo di sottorete : x.y.z.0000 0000 = 192.168.1.0/26~~
- ~~II indirizzo di sottorete : x.y.z.0100 0000 = 192.168.1.64/26~~
- III indirizzo di sottorete : x.y.z.1000 0000 = 192.168.10.128/26
- IV indirizzo di sottorete : x.y.z.1100 0000 = 192.168.10.192/26

e facciamo la nostra scelta tra i due che presentano quel bit uguale a 1, per esempio l'ultimo.

Quindi i due indirizzi di rete sono:
LAN A = 192.168.1.0/27
LAN B = 192.168.1.192/26

ESEMPIO 9

Nell'esempio 6 dell'unità precedente abbiamo utilizzato sottoreti con indirizzo di lunghezza uguale, mentre è possibile "risparmiare" indirizzi con sottoreti aventi lunghezze diverse.
Modifichiamo quindi l'architettura definendo tre sottoreti per la connessione delle tre sedi al posto di un'unica MAN, come indicato nello schema a lato. ▶

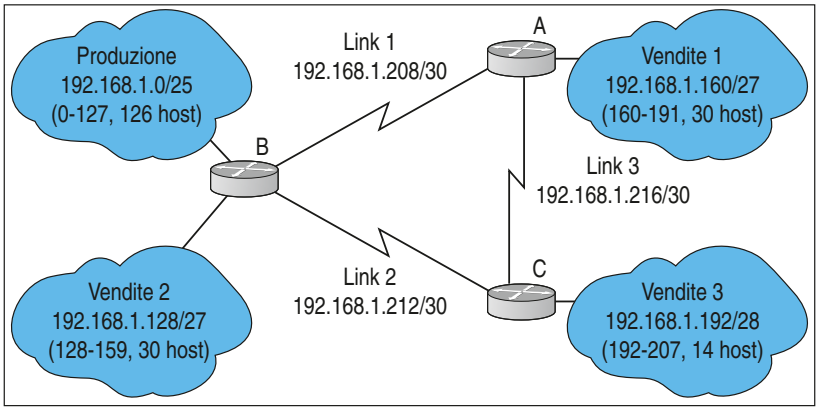


Poiché le tre reti R1, R2 e R3 necessitano solo di 2 indirizzi (più 2 di servizio), per esse si può utilizzare una sottorete con soltanto 2 bit di indirizzamento (11111100) e quindi con Subnet-mask 255.255.255.192 (per esempio, indirizzi possibili sono 200.69.96.240/30, 200.69.96.244/30, 200.69.96.248/30).
La rete S1 ha 40 host indirizzati con 6 bit (terzo byte a 1100000): 200.69.96.0/26.
La rete S2 ha 50 host indirizzati con 6 bit (terzo byte a 1100000): 200.69.96.64/26.
La rete S3 ha soli 20 host e quindi bastano 5 bit per indirizzarli (terzo byte a 11100000): 200.69.96.128/27.

Ricapitolando, abbiamo assegnato:
LAN S1 : 200.69.96.0/26
LAN S2 : 200.69.96.64/26
LAN S3 : 200.69.96.128/27
LINK R1: 200.69.96.240/30
LINK R2: 200.69.96.244/30
LINK R3: 200.69.96.248/30

ESEMPIO 10

Come ultimo esempio dell'uso VLSM proponiamo la soluzione dell'esempio 7 presentato nella precedente unità didattica, dove la rete è stata realizzata utilizzando un solo indirizzo di classe C, il **192.168.1.0**: nella figura seguente si propone un piano di indirizzamento:



192.168.1.0

(Lo svolgimento completo viene lasciato come esercizio all'allievo.)

Si noti che usando il semplice subnetting era necessario un indirizzo di classe B (oppure di due indirizzi di classe C), mentre ora la rete è stata coperta con un solo indirizzo di classe C. Il VLSM si presenta, quindi, come un meccanismo molto più versatile.

Forwarding diretto e indiretto

Una classificazione degli host può anche essere fatta in base al meccanismo di consegna di un pacchetto introducendo i concetti di destinazione diretta e indiretta, legati alla logica di routing.

HOST

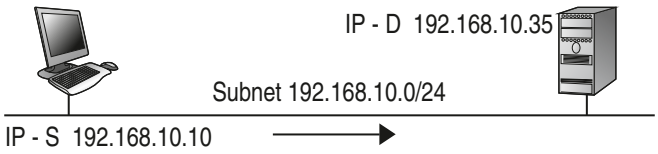
Un host diretto è una stazione collegata direttamente al segmento di rete dell'host d'origine.
Un host indiretto è un host di destinazione situato su una rete diversa da quella dell'host di origine.

Si parla quindi di *forwarding diretto* quando la trasmissione di un IP datagram avviene tra due host connessi su una rete con lo stesso Net-ID e quindi non vengono coinvolti i router; invece nel *forwarding indiretto* i datagram passano da un router all'altro finché non ne raggiungono uno che può trasmetterli direttamente al destinatario.

I router realizzano l'interconnessione tra le diverse reti: il termine ◀ hop ▶ indica il passaggio di un pacchetto di dati tra due nodi di una rete effettuato attraverso un router.

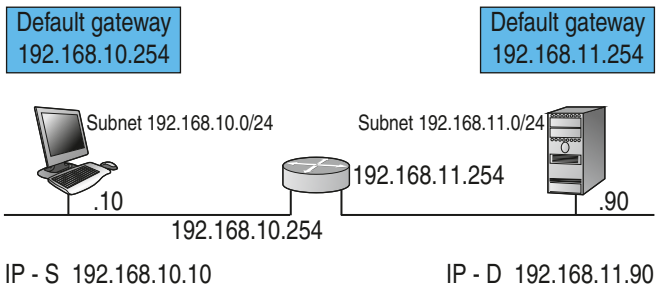
◀ Hop ("salto") è il termine usato nel routing per indicare un router o altro dispositivo di rete "attraversato" dai pacchetti trasmessi lungo la rotta verso un host di destinazione: "distanza 10 hop" significa che tra mittente e destinatario 10 dispositivi hanno ritrasmesso il pacchetto. ▶

Nella figura seguente viene presentata una situazione di forwarding diretto.



La macchina sorgente deve inviare un pacchetto all'indirizzo IP 192.168.10.35/24 e analizzando la propria interfaccia si accorge di essere connessa sulla rete con indirizzo 192.168.10.0/24 dove è connessa anche la macchina di destinazione, che quindi appartiene alla sua sottorete.

La figura seguente illustra invece un esempio di forwarding indiretto.



In questo caso host sorgente e host destinatario sono su due subnet diverse e quindi il mittente invia il pacchetto al router, del quale conosce l'indirizzo in quanto è quello predefinito di gateway. Nel pacchetto dovrà essere presente non solo l'indirizzo IP del destinatario, ma anche l'indirizzo del router (di cui viene utilizzato l'indirizzo MAC) al quale deve essere consegnato il pacchetto per la ritrasmissione: questo indirizzo viene sostituito con quello del destinatario quando un router lo riconosce connesso al proprio segmento, altrimenti viene sostituito con quello di un successivo router che inoltra il pacchetto a un'altra rete, alla ricerca del destinatario.
L'indirizzo IP serve quindi per effettuare la connessione logica, mentre quello MAC per risolvere la connessione fisica.

> Gli indirizzi di livello 3 non cambiano mai durante il tragitto di un datagram mentre quelli di livello 2 individuano gli apparati interessati alla trasmissione e ricezione del pacchetto all'interno di una particolare sottorete.

Subnetting: ripartizione logica e fisica

Il meccanismo del subnetting introduce una nuova visione delle macchine, differenziando la cosiddetta connessione logica (rete logica) da quella che invece chiamiamo connessione fisica (rete fisica).

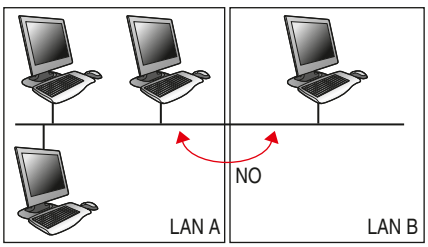
RETE LOGICA E FISICA

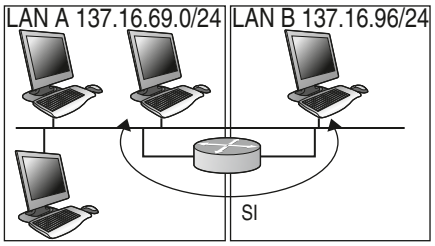
Si parla di rete fisica per indicare le macchine, o meglio le interfacce, che sono connesse su una stessa sottorete nella quale una particolare tecnologia di trasporto assicura la connessione.
Una rete logica è l'insieme delle interfacce a cui è stato assegnato lo stesso indirizzo di subnet e in cui il routing è implicito: due macchine all'interno della stessa rete logica possono comunicare senza dover passare attraverso un router.

Alla nascita del protocollo IP c'era una corrispondenza biunivoca tra reti fisiche e logiche, mentre oggi è possibile avere più reti logiche nella stessa rete fisica; in questo caso il routing tra reti logiche diverse è esplicito e gestito dai router.

Poiché l'operazione di subnetting di una rete IP è solamente logica e non fisica, si possono verificare le situazioni illustrate di seguito.

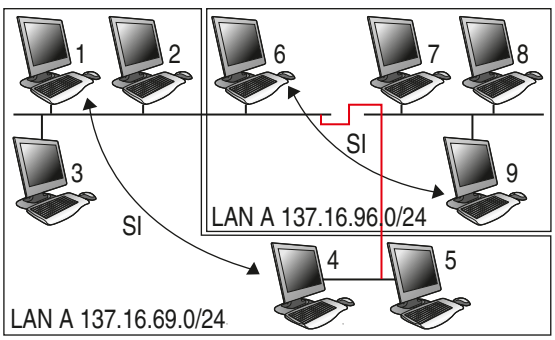
1 Host di diverse subnet possono essere sulla medesima rete fisica, ma non possono dialogare direttamente e hanno perciò bisogno di un gateway.





2 Host della medesima subnet possono essere su reti fisiche diverse e in questo caso possono dialogare direttamente perché appartengono a un unico dominio broadcast e quindi non necessitano di un gateway ma semplicemente di un dispositivo di livello 2 (bridge/switch). ▶

Gli host 1, 2, 3, 4, 5 appartengono alla stessa sottorete logica 137.16.69.0/24, mentre gli host 6, 7, 8, 9 appartengono alla stessa sottorete logica 137.16.96.0/24.



LIS

Con la denominazione Logical IP Subnet si definisce un'entità amministrativa separata, composta da un gruppo di nodi IP collegati alla stessa rete e appartenenti alla stessa IP subnet. Si tratta in definitiva di una rete fisica in cui gli host possono colloquiare direttamente tra loro senza passare attraverso router, ma utilizzando sistemi posti a livelli inferiori della pila OSI, quali switch, bridge e hub.

CIDR

Per rendere più flessibile la gestione degli indirizzi IP si è introdotto un nuovo sistema che supera i limiti della suddivisione in classi: il CIDR (*Classless InterDomain Routing*).

Definito nel 1992 dagli RFC 1518 e RFC 1519, il CIDR permette di svincolarsi dal rigido meccanismo della divisione in classi accorpando le reti secondo le diverse esigenze, in modo da formare un'unica super-rete (supernetting). Per poter realizzare le supernetting è però necessario avere a disposizione indirizzi contigui e definire una maschera che selezioni i bit che le due reti hanno in comune: questa maschera, che viene quindi utilizzata per identificare il Network-ID, prende il nome di maschera di supernetting.

Vediamo un esempio.

ESEMPIO 11

Vogliamo costruire una super-rete che comprenda tutti gli indirizzi di classe C compresi tra i due seguenti:

204.16.168.0	11001100	00010000	10101000	00000000
204.16.175.255	11001100	00010000	10101111	11111111

Analizzando il terzo byte osserviamo che le reti da aggregare sono 8, che vanno dall'indirizzo 168 fino all'indirizzo 175, comprendendo tutti gli indirizzi seguenti:

LAN 1				
204.16.168.0	11001100	00010000	10101000	00000000
204.16.168.255	11001100	00010000	10101000	11111111
LAN 2				
204.16.169.0	11001100	00010000	10101001	00000000
204.16.169.255	11001100	00010000	10101001	11111111
LAN 3				
204.16.170.0	11001100	00010000	10101010	00000000
204.16.170.255	11001100	00010000	10101010	11111111
LAN 4				
204.16.171.0	11001100	00010000	10101011	00000000
204.16.171.255	11001100	00010000	10101011	11111111
LAN 5				
204.16.172.0	11001100	00010000	10101100	00000000
204.16.172.255	11001100	00010000	10101100	11111111
LAN 6				
204.16.173.0	11001100	00010000	10101101	00000000
204.16.173.255	11001100	00010000	10101101	11111111
LAN 7				
204.16.174.0	11001100	00010000	10101110	00000000
204.16.174.255	11001100	00010000	10101110	11111111
LAN 8				
204.16.175.0	11001100	00010000	10101111	00000000
204.16.175.255	11001100	00010000	10101111	11111111

Tutti gli host hanno in comune i primi 21 bit, quindi la maschera di supernetting è la seguente:

255.255.248.0	11111111	11111111	11111000	00000000

Praticamente si effettua il procedimento inverso rispetto a quello di subnetting visto in precedenza: invece di scomporre la parte di Host-ID in due parti, si raggruppano gli indirizzi a formare una rete di dimensione maggiore.

Quindi le reti possono essere definite in termini di due componenti, cioè solo Net-ID/Net-mask

204.16.168.0/21

e la Net-mask viene semplicemente indicata con il numero dei bit che hanno valore 1.

Si parla perciò di Net-mask e non più di Subnet-mask in quanto il concetto di sottorete è superato.

Il CIDR ci permette non solo di utilizzare in modo più efficiente lo spazio degli indirizzi, ma anche, come vedremo, di semplificare le tabelle di routing in quanto le righe delle reti contigue vengono rappresentate nelle tabelle di routing da un'unica riga.

Vediamo ora un altro esempio.

ESEMPIO 12

Dobbiamo individuare la rete più opportuna per un'azienda pubblica che necessita di 2000 indirizzi IP.

Se utilizziamo il meccanismo classico di indirizzamento abbiamo bisogno di 11 bit per poter indirizzare tutti gli host ($2^{11} = 2048$); si rende quindi necessario acquistare un indirizzo di classe B, che però sarebbe sottoutilizzato in quanto sprecheremmo oltre 60.000 indirizzi (utilizzo del 3%).

Adottando la tecnica di aggregazione ci bastano 8 indirizzi di classe C, dato che ogni indirizzo ci permette di assegnare 256 host, quindi $8 \times 256 = 2048$ indirizzi.

Per esempio, prendiamo un gruppo di indirizzi che va da 222.10.0.0 a 222.10.7.0 e formiamo una supernetting basandoci sulla seguente tabella:

Identificato di rete	222.10.0.0/21	
Supernet-mask	255.255.248.0	
Indirizzo di broadcast	222.10.7.254	
Range di indirizzi	222.10.0.1	iniziale
	222.10.7.253	finale

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 L'acronimo VLSM significa:
- ○ *Validate Lan Subnet-Mask*
- ○ *Validate Length Subnet-Mask*
- ○ *Variable Length Subnet-Mask*
- ○ *Variable Lan Subnet-Mask*

2 Per aggregare gli indirizzi IP di un insieme di reti
- ○ si usa una notazione chiamata FLSM
- ○ si usa una notazione chiamata CIDR
- ○ si ricorre alla definzione di un range di indirizzi
- ○ si utilizza un router Cisco IPv6

3 La netmask serve
- ○ per nascondere l'indirizzo di rete di un host
- ○ per definire l'ampiezza dei campi Net-ID e Host-ID in un indirizzo IP
- ○ per modificare l'indirizzo di una rete
- ○ per nascondere l'indirizzo di una rete

4 Una rete logica è
- ○ l'insieme delle reti che condividono una stessa tecnologia
- ○ l'insieme degli host che condividono una stessa rete
- ○ l'insieme delle interfacce in cui una particolare tecnologia assicura la trasmissione
- ○ l'insieme delle interfacce di rete a cui è stato assegnato uno stesso indirizzo IP di Subnet e in cui il routing è diretto

5 Due host della stessa rete logica possono comunicare senza passare attraverso un router
- ○ sempre
- ○ mai
- ○ se sono sulla stessa rete fisica
- ○ se hanno la stessa Subnet-mask

6 Quale delle seguenti affermazioni è errata per l'indirizzo di gateway:
- ○ è un indirizzo composto da 4 byte
- ○ è necessario per il forwarding diretto
- ○ è necessario per comunicare se sono presenti sottoreti di dimensioni differenti
- ○ indica l'indirizzo di una porta di un router

7 L'acronimo CIDR significa:
- ○ *Classfree Intra Domain Routing*
- ○ *Classless Intra Domain Routing*
- ○ *Classfree Inter Domain Routing*
- ○ *Classless Inter Domain Routing*

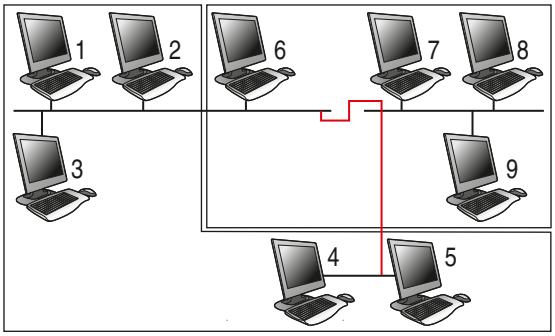
Test vero/falso

1	Con VLSM si intende il meccanismo che in una rete permette di avere maschere di lunghezza variabile.	V F
2	Un host diretto è collegato al segmento di rete dell'host d'origine mediante un router.	V F
3	Un host indiretto è situato su una rete diversa da quella dell'host di origine.	V F
4	Nel forwarding diretto la trasmissione di un IP datagram non coinvolge i bridge.	V F
5	Nel forwarding indiretto i datagram passano almeno da un router.	V F
6	Una rete logica è l'insieme delle interfacce a cui è stato assegnato lo stesso indirizzo di subnet.	V F
7	Due macchine all'interno della stessa rete logica possono comunicare senza dover passare attraverso un router solo se sono sulla stessa rete fisica.	V F
8	Host di diverse subnet possono essere sulla medesima rete fisica.	V F
9	Host della medesima subnet possono essere su reti fisiche diverse.	V F

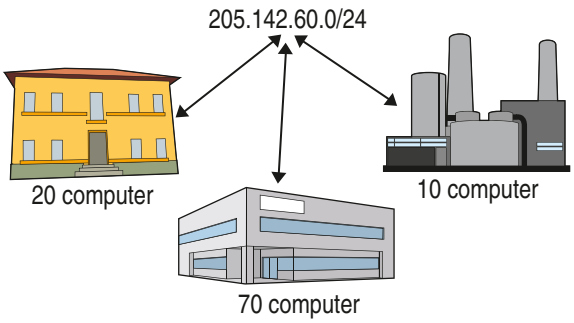
Verifichiamo le competenze

Esprimi la tua creatività

1 Assegna un possibile piano di indirizzamento per la rete seguente, costituita da due sottoreti logiche, a partire da un indirizzo 137.16.0.0.



2 Dato l'indirizzo di classe C 205.142.60.0/ 24 e la situazione rappresentata nella figura



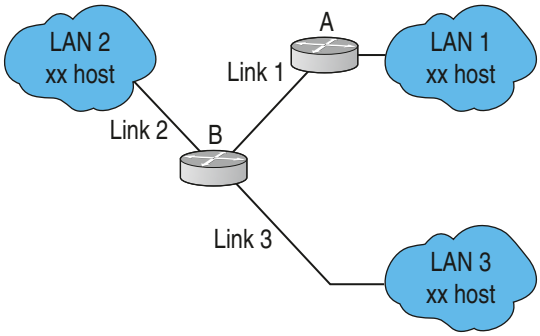
a) progetta 3 sottoreti, con i relativi dispositivi HW (router, switch), che sfruttino quest'unico indirizzo disponibile con subnet a lunghezza variabile;

b) indica la percentuale di utilizzo degli indirizzi disponibili.

3 Nel seguente elenco raggruppa gli indirizzi IP appartenenti alla medesima subnet.

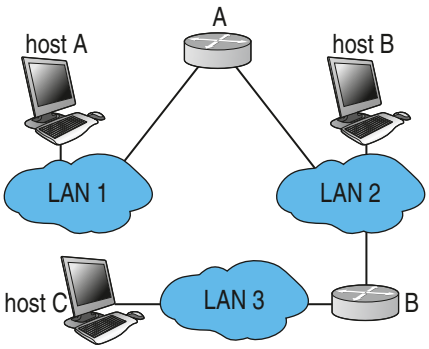
200.150.150.161, netmask 255.255.255.224
150.150.150.150, netmask 255.255.240.0
200.150.150.190, netmask 255.255.255.224
150.150.134.145, netmask 255.255.240.0
200.150.150.180, netmask 255.255.255.224
150.150.150.151, netmask 255.255.252.0
200.150.150.151, netmask 255.255.255.224
200.150.150.192, netmask 255.255.255.224
200.150.150.150, netmask 255.255.252.0
200.150.150.182, netmask 255.255.255.224
150.150.134.154, netmask 255.255.240.0
200.150.151.150, netmask 255.255.252.0
200.150.150.162, netmask 255.255.255.224
200.150.151.151, netmask 255.255.252.0
200.150.150.152, netmask 255.255.255.224
150.150.134.158, netmask 255.255.240.0

4 A un'azienda organizzata come risulta dallo schema di cui alla figura sottostante viene assegnato il blocco di CIDR di indirizzi 16.5.1.0/24.
Assegna gli indirizzi di rete alle tre LAN (LAN1, LAN2, LAN3) partizionando il blocco in modo che LAN1 e LAN2 possano contenere almeno 32 stazioni mentre LAN3 ne possa contenere almeno 64 e assegnando il primo numero disponibile di ogni LAN all'interfaccia del router ad esso collegato (indica gli indirizzi in forma sia binaria che decimale).



5 Con riferimento alla situazione precedentemente descritta, per ogni LAN specifica l'indirizzo di broadcast spiegando come si ottiene e calcola l'efficienza di utilizzo in percentuale (indica gli indirizzi in forma sia binaria che decimale).

6 A un'azienda organizzata come mostrato nella figura che segue viene assegnato il blocco di CIDR di indirizzi 16.5.1.0/24.
Assegna gli indirizzi di rete alle tre LAN (LAN1, LAN2, LAN3) partizionando il blocco in modo che ciascuna LAN possa contenere lo stesso numero di stazioni, massimizzando tale numero, e assegnando il primo numero disponibile di ogni LAN all'interfaccia del router ad essa collegato (indica gli indirizzi in forma sia binaria che decimale).



UNITÀ DIDATTICA 4

CONFIGURARE UN PC: IP STATICO E DINAMICO

IN QUESTA UNITÀ IMPAREREMO...

- a configurare manualmente un PC
- a configurare automaticamente un PC con il DHCP
- a visualizzare lo stato di un PC
- il protocollo ARP/RARP

Configurazione di un PC in una LAN

Affinché un host possa appartenere a una LAN e quindi comunicare con gli altri PC deve essere appositamente configurato. Ogni host connesso alla rete IP deve avere al suo interno tre parametri essenziali:

1. l'indirizzo IP completo dell'host;
2. la Subnet-mask che permette l'estrazione del prefisso di rete (o sottorete);
3. il gateway predefinito a cui saranno inviati i pacchetti IP i cui destinatari non si trovano sulla stessa subnet del mittente.

Nelle reti TCP/IP questa operazione può essere effettuata in due modalità:

- manuale;
- automatica.

Nella configurazione manuale l'amministratore di rete deve provvedere a effettuare il piano di indirizzamento, seguendo per esempio le seguenti fasi operative in caso di subnetting:

- determinazione del numero di Subnet-ID necessari oggi e per le future esigenze di crescita;
- determinazione del numero massimo di indirizzi IP su ogni subnet, considerando la crescita futura;
- determinazione delle Subnet-mask;
- calcolo dei Subnet-ID da utilizzare;
- calcolo degli Host-ID e assegnazione degli indirizzi IP agli host.

Una volta definiti tutti i parametri, questi devono essere "introdotti" manualmente in ogni host.

Con l'assegnazione manuale ogni macchina ha sempre lo stesso indirizzo IP.

Nella configurazione automatica, invece, l'assegnazione degli indirizzi viene effettuata a ogni accensione dell'host da un particolare programma, il DHCP.

Con l'assegnazione automatica ogni macchina può avere sempre un indirizzo IP diverso.

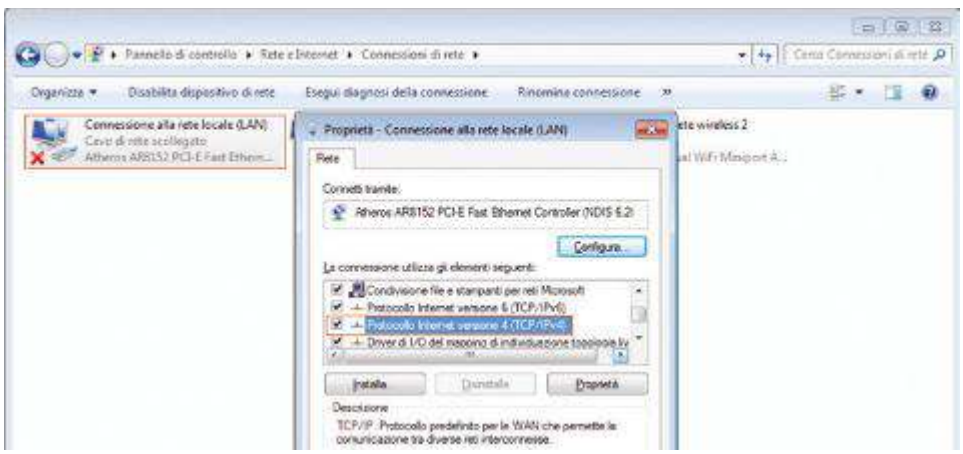
Assegnazione manuale

In questo caso tutti i parametri devono dapprima essere definiti dall'amministratore di rete e successivamente introdotti manualmente in ogni host.

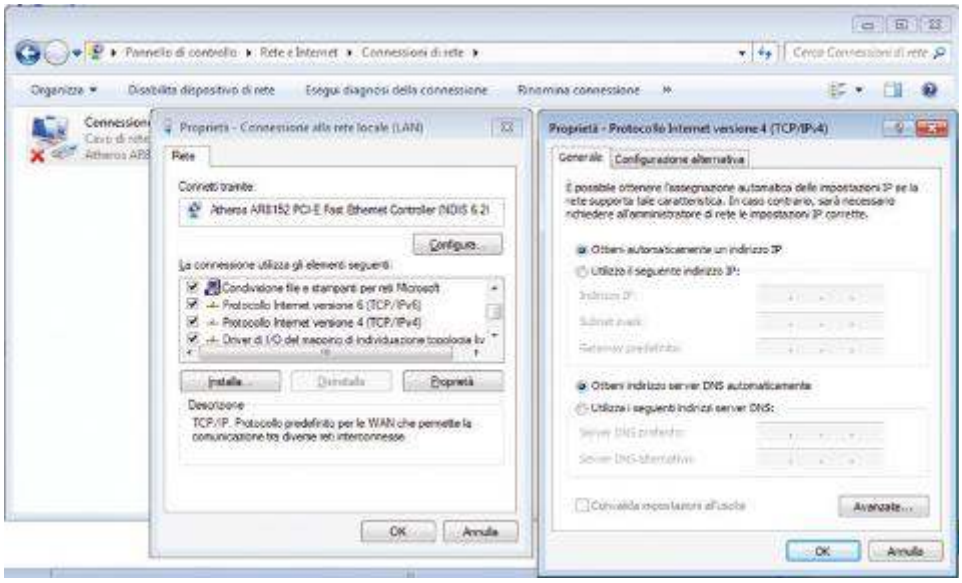
In Windows 7 questi parametri vengono inseriti dal pannello di controllo, dopo aver selezionato l'opzione Rete e Internet:

Si seleziona Visualizza stato della rete e attività:

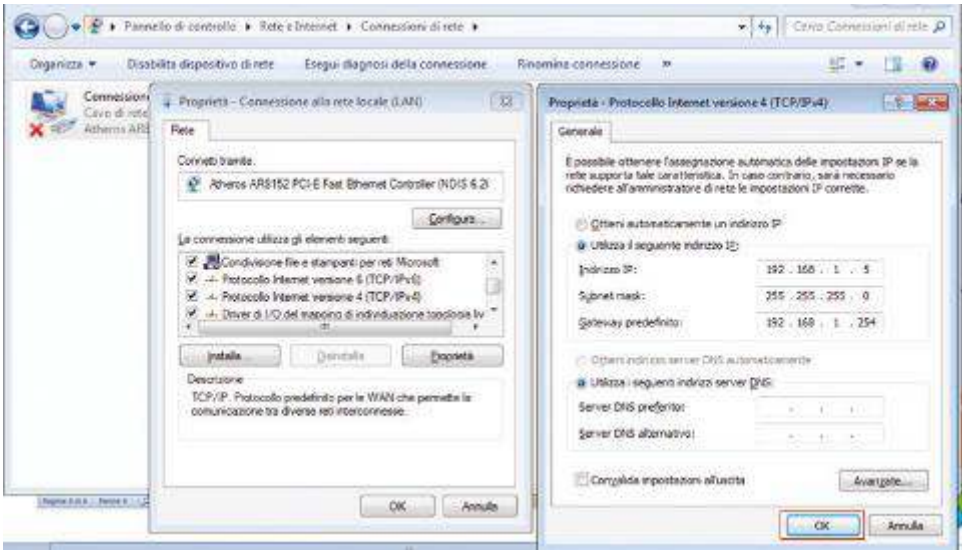
e nella finestra che appare si fa clic su Modifica impostazione scheda:



Dopo aver fatto clic su Connessione alla rete locale si seleziona il Protocollo Internet versione 4.



Nella finestra che segue è possibile inserire i parametri di configurazione, dopo aver selezionato Utilizza il seguente indirizzo IP:



Se si fa clic sul pulsante OK i parametri vengono salvati nella memoria della macchina.

In Linux distribuzione Ubuntu questi parametri vengono inseriti dopo aver cliccato sull'icona con i due PC nella barra di notifica:

e alla comparsa della successiva finestra

si seleziona il pulsante Edit… per visualizzare la seguente schermata

nella quale, dopo aver selezionato la tab IPv4 Settings, è possibile settare i valori per l'indirizzo IP, la Net-mask e l'indirizzo di gateway predefinito.

Assegnazione mediante DHCP

Se non è indispensabile assegnare agli host indirizzi permanenti si può usare il metodo dinamico di attribuzione.
In questo caso viene utilizzato il DHCP (*Dynamic Host Configuration Protocol*) RFC 2131,2132, che è un programma di attribuzione dinamica degli indirizzi IP.

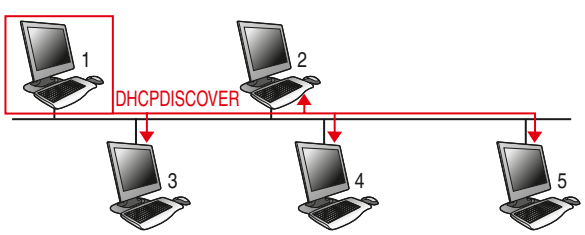
Una volta definito sulla rete un server DHCP, gli viene comunicato l'elenco degli indirizzi che deve assegnare e viene creato un database DHCP: al momento dell'accensione di un host, questo entra in comunicazione con il server DHCP richiedendo i parametri di rete per effettuare l'autoconfigurazione.

> La generazione dell'indirizzo IP viene fatta dal server in modo casuale, nel range dei suoi indirizzi disponibili, e appena un nuovo indirizzo viene creato si rende necessaria una verifica di unicità: a tale scopo si utilizza il protocollo ARP, descritto nel prossimo paragrafo.
> Nel caso in cui si fossero generati 2 indirizzi uguali, l'ultimo verrebbe scartato.

Ci sono due possibili modalità di funzionamento del DHCP:
- allocazione automatica: il DHCP assegna permanentemente un indirizzo IP;
- allocazione dinamica: il DHCP assegna un indirizzo IP per un intervallo di tempo limitato.

Entrambi utilizzano lo stesso meccanismo per l'assegnazione degli indirizzi, che di seguito analizziamo nel dettaglio.

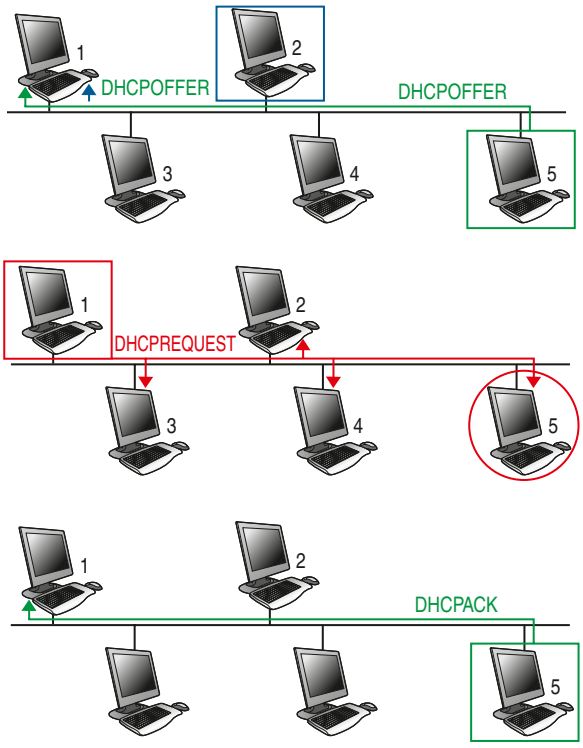
1. Supponiamo che un host, il PC1, venga acceso: la sua interfaccia di rete invia in modalità broadcast un messaggio DHCPDISCOVER in cerca di un server DHCP.



2 Tutti gli eventuali server DHCP presenti sulla rete rispondono all'host con un messaggio DHCPOFFER con cui ciascuno propone un indirizzo IP: nel nostro esempio sono presenti due server DHCP; uno su PC2 e l'altro su PC5. ▶

3 L'host accetta una sola delle due offerte che gli giungono, quella di PC5, e manda sempre in broadcast un messaggio DHCPREQUEST richiedendo a PC5 di trasmettergli la configurazione completa. ▶

4 Il server DHCP di PC5 risponde all'host con un messaggio DHCPACK specificando i parametri di configurazione, che sono:
- l'indirizzo IP;
- la Subnet-mask;
- l'indirizzo di broadcast;
- il gateway predefinito;
- il server DNS (le cui funzioni saranno descritte in seguito). ▶



Il DHCP, il cui principale vantaggio è un utilizzo efficiente degli indirizzi IP che vengono assegnati solo alle macchine in funzione e connesse, è indispensabile nel caso si abbiano a disposizione pochi indirizzi IP validi su Internet.

Questo meccanismo viene utilizzato dai provider Internet per attribuire indirizzi temporanei ai computer che si collegano via modem.

Inoltre in questo modo si eliminano gli errori di configurazione degli host in quanto niente viene fatto manualmente.

Zoom su...

DHCP LEASING

Quando un host viene spento o disconnesso deve rilasciare l'indirizzo e renderlo disponibile per una nuova allocazione: il rilascio può essere esplicito (DHCPRELEASE) o avvenire allo scadere del periodo di lease (parametro fornito dal server nel DHCPOFFER).

Comando ipconfig

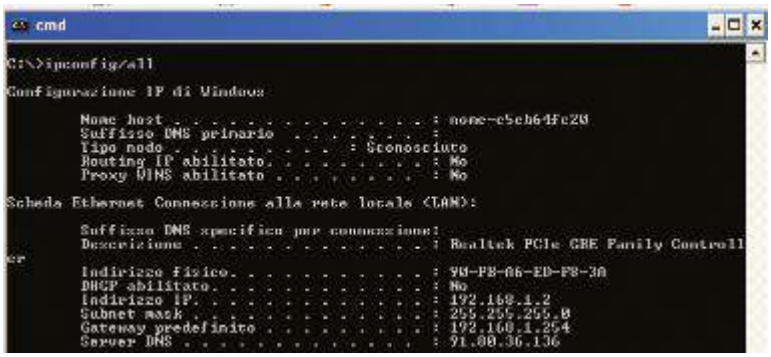
Il comando ipconfig ci permette di visualizzare le principali configurazioni IP del nostro PC.
Con l'opzione /all

ipconfig/all

visualizza la configurazione IP corrente di ciascuna interfaccia di rete presente nella macchina:

- indirizzo MAC;
- indirizzo IP;
- Subnet-mask;
- default gateway;
- server DNS;

oltre ad altre informazioni, come possiamo vedere nella schermata a lato. ▶



ARP: Address Resolution Protocol

Nel *forwarding diretto* la trasmissione di un IP datagram tra due host connessi su una rete con lo stesso Net-ID non coinvolge i router mentre nel *forwarding indiretto* i datagram passano da un router all'altro finché non ne raggiungono uno in grado di trasmetterli direttamente al destinatario (ogni passaggio prende anche il nome di hop).

Se un pacchetto può essere consegnato direttamente o indirettamente viene individuato dallo stesso trasmettitore analizzando gli indirizzi IP del mittente e del destinatario ma, come sappiamo, la trasmissione fisica a livello 2 utilizza gli indirizzi MAC per trasmettere i dati tra due NIC.
Per effettuare la consegna del pacchetto (o del datagramma) è quindi necessario passare da indirizzo IP a indirizzo MAC e viceversa, utilizzando il protocollo ARP.

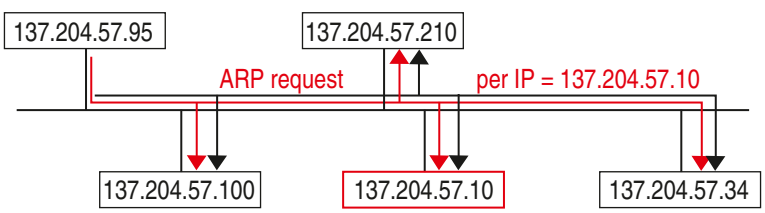
ARP (RFC 826), acronimo di *Address Resolution Protocol*, gestisce la traduzione degli indirizzi IP in indirizzi fisici e nasconde questi indirizzi fisici agli strati superiori.
ARP funziona con tabelle di mappatura chiamate cache ARP, che forniscono la corrispondenza tra un indirizzo IP e un indirizzo fisico, e viene usato tutte le volte che un host vuole inviare un pacchetto a un altro host sulla sottorete, di cui conosce solo l'indirizzo IP.
La cache ARP viene compilata dal software di rete man mano che viene a conoscenza degli indirizzi MAC degli host ai quali ha inviato un pacchetto mediante l'indirizzo IP.

Vediamo, di seguito, una schematizzazione del funzionamento di questo protocollo.

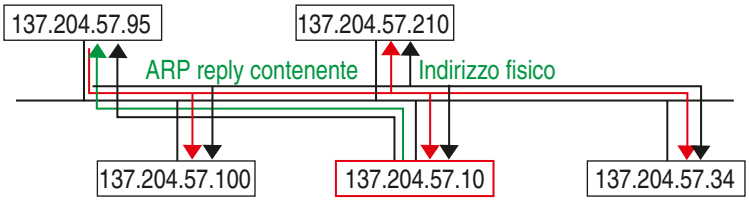
All'atto dalla trasmissione per prima cosa il mittente stabilisce, dall'analisi dell'indirizzo IP, se il destinatario è sulla sua stessa LIS (rete logica) e attiva il software ARP che esegue la seguente procedura:

Ⓐ viene controllata la cache ARP del mittente per verificare se contiene la corrispondenza tra indirizzo fisico e indirizzo IP;

Ⓑ se non viene trovata una corrispondenza, cioè se si tratta del primo pacchetto che viene inviato a quel destinatario, il mittente invia una richiesta (ARP request) a tutte le macchine della rete locale mediante una trasmissione broadcast. L'interrogazione è del tipo: "se riconosci come tuo questo indirizzo IP fammi sapere qual è il tuo indirizzo hardware";



C ogni host sulla rete locale controlla se l'indirizzo IP richiesto corrisponde al proprio e risponde (ARP replay) solo in caso affermativo.



L'ARP replay viene inviata direttamente al mittente ed è costituita da una coppia di valori, del tipo

20-4C-4F-4F-50-20 (137.204.57.10)

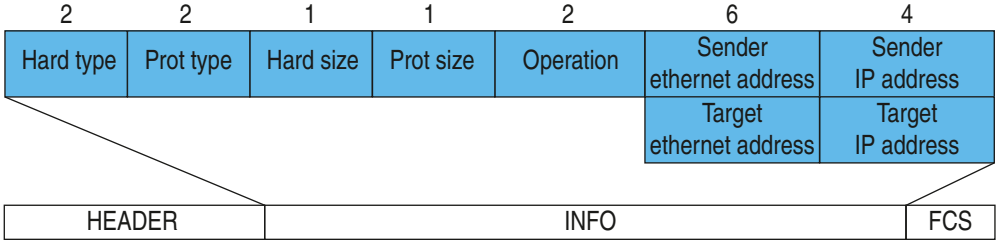
che può essere così tradotta: "Io sono l'host con indirizzo IP 137.204.57.10 e il mio indirizzo Ethernet è 20-4C-4F-4F-50-20";

D a seguito di questo scambio di informazioni mittente e destinatario aggiornano le proprie cache ARP con le corrispondenze tra indirizzo IP e indirizzi Ethernet per le trasmissioni successive.

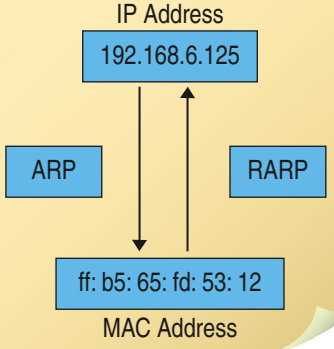
Il protocollo ARP si appoggia direttamente sul livello data link e non su IP e il pacchetto ARP è incapsulato nella PDU del livello data link, che potrebbe essere per esempio una trama Ethernet.

Oltre ai campi per gli indirizzi di livello 2 e 3 di sorgente e destinazione, il pacchetto ARP contiene:

- hard type: specifica il tipo di indirizzo di livello 2; per indicare che l'indirizzo è di tipo MAC si usa il valore 1;
- protocol type: specifica il tipo di indirizzo di livello 3; per indicare indirizzi IP si usa 0x0800;
- hard size: indica la lunghezza dell'indirizzo di livello 2;
- protocol size: indica la lunghezza dell'indirizzo di livello 3;
- operation: indica il tipo di comando ARP, e cioè 1 per ARP-request, 2 per ARP-reply.



Oltre al protocollo ARP è presente un secondo protocollo, utilizzato sempre per associare gli indirizzi tra due livelli diversi: il protocollo RARP (*Reverse Address Resolution Protocol*). Questo effettua il procedimento inverso, cioè a partire dall'indirizzo MAC individua l'indirizzo IP: viene utilizzato dagli host che non hanno memoria di massa al momento del bootstrap per reperire il proprio indirizzo *IP*.

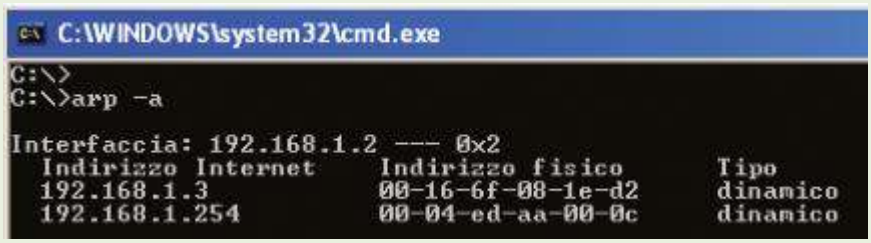


Zoom su...

COMANDO ARP

È possibile visualizzare il contenuto della cache ARP con le diverse corrispondenze tra indirizzi IP e MAC digitando direttamente il comando

arp–a



Questo comando ha anche altre opzioni, il cui elenco è direttamente consultabile digitando semplicemente arp senza alcuna opzione.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Quali sono i parametri minimi che devono essere configurati in un PC connesso a una LAN?

- ○ Indirizzo MAC
- ○ Indirizzo IP
- ○ Subnet-mask
- ○ Gateway predefinito
- ○ Indirizzo di loopback
- ○ Server DNS

2 Nelle reti TCP/IP ci sono due modalità per effettuare questa operazione: quali sono?

- ○ Manuale
- ○ Diretta
- ○ Automatica
- ○ Indiretta

3 Che cosa significa DHCP?

- ○ *Direct Host Configuration Protocol*
- ○ *Direct Host Configuration Program*
- ○ *Dynamic Host Configuration Protocol*
- ○ *Dynamic Host Configuration Program*

4 Quali meccanismi di funzionamento del DHCP esistono?

- ○ Allocazione manuale
- ○ Allocazione automatica
- ○ Allocazione dinamica
- ○ Allocazione statica

5 Indica la sequenza delle operazioni necessarie per l'assegnazione di un indirizzo IP con DHCP.

Invio di DHCPRELEASE .....
Invio di DHCPACK .....
Invio di DHCPOFFER .....
Invio di DHCPDISCOVER .....
Invio di DHCPREQUEST .....

6 Che cosa significa ARP?

- ○ *Address Resolution Program*
- ○ *Address Resolving Protocol*
- ○ *Address Resolving Program*
- ○ *Address Resolution Protocol*

7 A chi il mittente di un pacchetto invia una richiesta ARP?

- ○ A tutti i PC in broadcast
- ○ Al destinatario
- ○ Al router più vicino
- ○ Al gateway predefinito

8 Quali sono i campi contenuti nel pacchetto ARP? (indica quello errato)

- ○ Hard type
- ○ Protocol type
- ○ Hard size
- ○ Protocol size
- ○ Operation type

Test vero/falso

1	Con l'assegnazione manuale ogni macchina ha sempre lo stesso indirizzo IP.	V F
2	Con l'assegnazione automatica ogni macchina ha sempre lo stesso indirizzo IP.	V F
3	L'allocazione automatica assegna permanentemente un indirizzo IP.	V F
4	L'allocazione dinamica assegna un indirizzo IP per un intervallo di tempo limitato.	V F
5	Il comando DHCPRELEASE effettua un rilascio esplicito dell'indirizzo IP.	V F
6	Il comando "ipconfig" visualizza le principali configurazioni IP del PC.	V F
7	ARP gestisce la traduzione degli indirizzi IP in indirizzi fisici.	V F
8	ARP funziona con tabelle di mappatura chiamate cache ARP.	V F
9	RARP gestisce la traduzione di indirizzi fisici in indirizzi IP.	V F
10	L'ARP request viene inviata dal mittente del pacchetto IP.	V F
11	L'ARP replay viene inviata direttamente al mittente del pacchetto.	V F

Esercizi di completamento

1 Descrivi le fasi di una comunicazione DHCP.

2 Descrivi le fasi di una comunicazione ARP.

3 Che cosa si intende per DHCP leasing?

4 Che cosa viene visualizzato con il comando ipconfig/all?

5 Che cosa viene visualizzato con il comando arp-a?

6 Mediante un analizzatore di rete vengono visualizzati i primi 12 byte di alcuni pacchetti: tra questi, uno non è un ARP. Quale? Perché?

- ○ 11 11 11 11 11 11 08 00 2B CF AA 28
- ○ FF FF FF FF FF FF 80 00 2B 09 00 12
- ○ 08 00 2B AA 2B 11 11 11 11 11 11 11
- ○ 80 00 2B 09 11 12 FF FF FF FF FF FF

7 Un host ha indirizzo 192.168.0.5, Net-mask 255.255.255.0 e indirizzo Ethernet 00-01-4A-01-02-03 ed è collegato a una LAN il cui router ha due interfacce: un'interfaccia Ethernet, collegata alla LAN dell'host, con indirizzo IP 192.168.0.254, Net-mask 255.255.255.0 e indirizzo MAC 00-00-0C-12-34-56, e un'interfaccia ADSL con indirizzo IP 190.190.190.190, Net-mask 255.255.255.0 e indirizzo MAC 00-00-0C-98-76-54.L'host deve trasmettere un pacchetto all'indirizzo 168.168.1.4. Descrivi la sequenza completa dei pacchetti ARP e dei dati che transiteranno sulla LAN, indicando per ciascuno gli indirizzi IP e MAC di mittenti e destinatari.

UNITÀ DIDATTICA 5

INOLTRO DI PACCHETTI SULLA RETE: NAT, PAT E ICMP

IN QUESTA UNITÀ IMPAREREMO...

- a inoltrare i pacchetti sulla rete
- ad assegnare dinamicamente gli indirizzi
- il protocollo ICMP
- a seguire i pacchetti sulla rete

Premessa

Nel TCP/IP le applicazioni operano a livello 4 e, come per ogni altro livello, quando si vuole trasferire una o più TPDU (*Transport Protocol Data Unit*) da una sorgente a una destinazione, occorre specificare mittente e destinatario.

Come vedremo nel seguito del corso, il protocollo di livello 4 utilizza un indirizzo detto TSAP address (*Transport Service Access Point address*) diverso da quello di livello 3.

Infatti, poiché sullo stesso host possono girare diverse applicazioni, ogni host può offrire più servizi e, contemporaneamente, più host possono offrire gli stessi servizi sulla rete; è quindi necessario aggiungere all'indirizzo IP un'ulteriore informazione supplementare, il cosiddetto NSAP address (*Network Service Access Point address*), che permetta di individuare in modo univoco uno specifico host e, in particolare, un servizio offerto da quell'host.

Per questo motivo i protocolli di livello 4 tipicamente definiscono come indirizzo una coppia, formata da un indirizzo di livello network che identifica l'host (indirizzo IP) e da un'altra informazione che identifica un punto di accesso in quell'host: il numero di porta.

Il numero di porta non è un indirizzo fisico, in quanto gli host generalmente hanno solo una porta fisica (interfaccia NIC) di connessione alla LAN, bensì una "porta logica", ossia un valore numerico univoco memorizzato in un campo a 16 bit, che quindi può assumere un valore tra 0 e 65535.

Per esempio, in TCP/IP un TSAP address (ossia un indirizzo TCP o UDP) ha la seguente forma:

IP address : port number

I valori da 1 a 1024 sono riservati ai servizi più diffusi, alcuni dei quali sono riportati nella tabella riportata a lato. ▶
Per esempio, la coppia (127.104.69.16 : 23) indica la porta 23 dell'host xyx di indirizzo IP 127.104.69.16, cioè il punto di accesso all'applicazione che permette a un utente di collegarsi mediante il servizio telnet (porta 23) all'host xyz.

Servizio	Porta	Servizio	Porta
Ping	7	DNS	53
ftp	21	Web	80
Ssh	22	Pop3	110
telnet	23	Imap	143
smtp	25	UUCP	540

Lo studio del livello 4 e di tutti i servizi sopra elencati verrà affrontato dettagliatamente nel seguito del corso.

Network Address Translation

L'utilizzo di indirizzi privati permette di trasferire i dati solo all'interno dell'intranet in quanto i router li riconoscono e non li inoltrano sulla rete Internet: i NAT (*Network Address Translator*) hanno la funzione di offrire a ogni host della rete privata la possibilità di accedere a ogni indirizzo Internet utilizzando un numero limitato di indirizzi pubblici: comportano quindi un beneficio economico riducendo i costi di connessione.

Il NAT offre anche la soluzione all'esaurimento degli indirizzi IP e "all'esplosione" delle tabelle di instradamento nei router; infine, il suo principale vantaggio consiste nel fatto che la sua installazione non richiede la riconfigurazione di router e host.

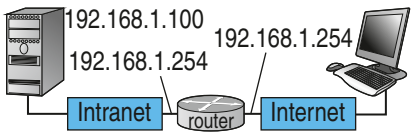
STUB DOMAIN
La rete interna, che come tale usa i soli indirizzi IP interni, viene definita anche stub domain o stub network (letteralmente, "dominio tronco" o "troncone di rete").

Il NAT viene posto "al confine" di uno stub domain, in uscita da una rete locale verso Internet: dato che ogni macchina della rete interna ha un indirizzo IP privato non annunciabile all'esterno e unico solo all'interno di quella sottorete, affidiamo al NAT il compito di convertire tale indirizzo in un indirizzo pubblico unico, che comparirà come indirizzo sorgente all'interno dei pacchetti verso la destinazione stabilita su Internet.

La traduzione degli indirizzi effettuata dal NAT server può essere di due tipi:
- statica (static NAT), che avviene mediante mappatura statica uno-a-uno tra indirizzi interni ed esterni;
- dinamica (IP masquerading): in tal caso la corrispondenza tra indirizzo interno e indirizzo esterno è definita solo all'occorrenza.

Mappatura statica

Si opta per la mappatura statica, per esempio, quando si vuole rendere raggiungibile un server della rete interna; è necessario introdurre una entry statica nella tabella del router (static NAT) con il rischio di abbassare il livello di sicurezza della rete interna. ▶



Static NAT table			
Ext address	Ext port	Int address	Int port
158.109.1.253	80/TCP	192.168.1.100	80/TCP

Con la traduzione dinamica, che non risente della traduzione di indirizzo e di porta effettuati dal router, un client della rete interna può connettersi a qualunque server della rete Internet; la tabella NAT viene creata solo in caso di richiesta di connessione.

Mappatura dinamica o IP masquerading

IP MASQUERADING

Viene detto IP masquerading (o NAT dinamico) il caso particolare NAT in cui le connessioni generate da un insieme di computer vengono "presentate" all'esterno con un solo indirizzo IP.

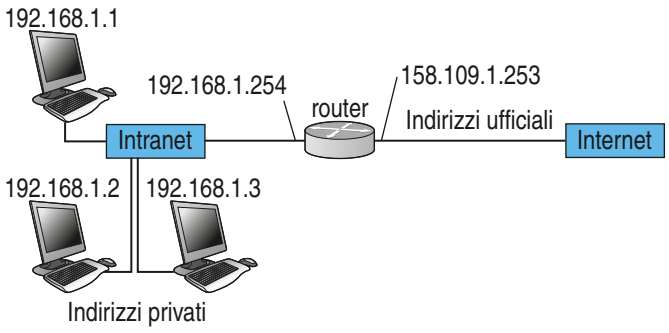
La tecnica utilizza anche il meccanismo PAT (*Port Address Translation*) e prende il nome di IP Overloading o NAPT (*Network Address and Port Translation*), in quanto vengono modificati non solo gli indirizzi IP ma anche le porte TCP e UDP delle connessioni in transito per individuare il reale mittente/destinatario del pacchetto.

Il router (o il server) tiene traccia della corrispondenza tra gli indirizzi reali e gli indirizzi "traslati" in una tabella, la Dynamic NAT table, che viene appunto creata dinamicamente, aggiungendo una riga alla prima richiesta effettuata da ogni client e inserendovi i suoi dati, indirizzo IP e numero di porta.

Il caso tipico di utilizzo del IP masquerading è quello delle reti aziendali; in quest'ambito infatti permette a più calcolatori presenti nell'intranet di condividere una singola connessione a Internet (un solo indirizzo IP pubblico) e viene utilizzato nel caso degli accessi xDSL e ISDN per piccole LAN.

Alcuni sistemi operativi offrono tale funzionamento in modo nativo (Windows Xp e Seven, Linux).

La Dynamic NAT table del router, che all'accensione del dispositivo risulta vuota, acquisisce una nuova entry a ogni flusso TCP o UDP uscito dalla rete intranet.



Le porte TCP e UDP esterne usate dal router per la *Port Address Traslation* (PAT) vanno da 1024 a 65535, come riportato nella RCF 2663.

Vediamo ora un esempio di funzionamento, dove indichiamo con SPort la porta sorgente usata dal mittente dal protocollo TCP e con DPort la porta del destinatario.

1. Supponiamo che l'host 192.168.1 debba contattare un server web su Internet: il client invia al default router un pacchetto IP avente:

mittente	**SIP** 192.168.1.1	**SPort** 30231/TCP
destinatario	**DIP** 62.41.244.2	**DPort** 80/TCP

2. Il router, prima di inoltrarlo, sostituisce ai dati del mittente i propri, cioè cambia:

indirizzo IP	: **SIP** 192.168.1.1	in 158.109.1.253
numero porta	: **SPort** 30231/TCP	in 1024/TCP

e per conservare memoria di questa sostituzione aggiorna la Dynamic NAT table inserendo un record in modo da tenere traccia del flusso di dati uscente e associando in modo univoco mittente e numero di porta:

Dynamic NAT table				
Int address	Int port	Ext address	Ext port	Age
192.168.1.1	30231/TCP	158.109.1.253	1024/TCP	

Il router inoltra quindi il pacchetto modificato dapprima al next hop router e poi al server web.

NEXT HOP ROUTER

Il next hop router è un contatore presente all'interno del pacchetto che viene incrementato da ciascun router ogni volta che il pacchetto stesso viene trasmesso, in modo da ricordare quanti nodi questo ha attraversato sulla rete.

3. Una volta che il pacchetto arriva a destinazione, il server risponde al client con un pacchetto avente:

mittente	**SIP** 62.41.244.2	**SPort** 80/TCP
destinatario	**DIP** 158.109.1.253	**DPort** 1024/TCP

dove il destinatario non è l'host sorgente, ma il router "di confine" dell'intranet.

4. Quando il pacchetto arriva al router, quest'ultimo consulta la Dynamic NAT table alla ricerca di eventuali regole di traduzione da applicare a ritroso, individua il record analizzando i dati della porta e infine si prepara all'inoltro, sostituendo IP e porta di destinazione:

indirizzo IP	: 158.109.1.253	in **SIP** 192.168.1.1
numero porta	: 1024/TCP	in **SPort** 30231/TCP

5. Il client riceve il pacchetto correttamente.

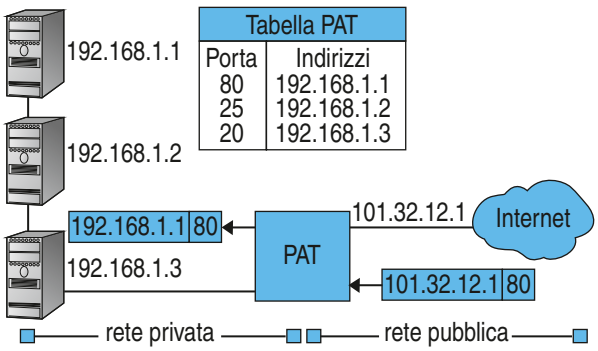
PAT

Il servizio PAT (*Port Address Translation*) traduce non solo gli indirizzi IP, ma anche i numeri delle porte TCP e UDP e permette quindi di avere molti host "dietro" a un singolo indirizzo IP; effettua inoltre un servizio complementare al PAT per rendere accessibili, anche da Internet, i servizi realizzati su calcolatori collocati nella rete privata.

ESEMPIO 13

I server web rispondono sulla porta 80 e quando noi indichiamo l'indirizzo di un server, per esempio www.itisXYZ.it, sottintendiamo che vogliamo accedere alla porta 80 del server www.itisXYZ.it.

Un server web su rete Internet con un IP fisso è in grado di rispondere a tutte le interrogazioni che arrivano su queste porte (vedremo in seguito come opera): il problema nasce quando il server si trova in una rete privata e possiede quindi un indirizzo nascosto, non accessibile da quella pubblica. È necessario perciò disporre di un meccanismo come il PAT, in grado di intercettare tutte le richieste che giungono su una specifica porta e reinstradarle verso un server della rete nascosta. In questo caso il dispositivo che esegue il PAT deve essere accessibile dalla rete pubblica e quindi avere un indirizzo pubblico.
Vediamo, di seguito, un esempio nel quale il PAT viene eseguito da un host con indirizzo pubblico 101.32.12.1.



Nella rete privata sono presenti tre server che offrono tre diversi servizi rispettivamente sulle loro porte 80, 25 e 20: la tabella di reinstradamento del PAT è una tabella statica che contiene i numeri di porta e i corrispondenti indirizzi IP dei tre server.

Il PAT, dopo aver acquisito un pacchetto di dati proveniente dalla rete pubblica e destinato all'indirizzo IP della rete privata, esamina la tabella "entrando" con la porta del servizio a cui era destinato per ricavarne una corrispondenza tra il numero di porta e l'indirizzo IP del server sulla rete privata; effettua poi il cambio di indirizzo, sostituendo l'IP privato al proprio indirizzo e inoltrando il pacchetto sulla rete privata.

Il percorso inverso, dal server sulla rete privata alle macchine su Internet, come abbiamo visto viene compiuto in collaborazione con il NAT: infatti, NAT e PAT sono sempre contemporaneamente presenti.

Utilizzando il meccanismo NAT/PAT per accedere all'esterno da una rete privata con un solo indirizzo vengono utilizzate sempre porte non privilegiate, con valori maggiori o uguali a 1024.

ICMP: Internet Control Message Protocol

Il protocollo IP, offrendo un servizio di tipo "best effort", non garantisce la corretta consegna dei datagrammi: se necessario si appoggia a protocolli affidabili di livello superiore (TCP).
Per la segnalazione degli errori sopravvenuti nel corso dell'elaborazione di un datagramma e per la generazione di messaggi amministrativi e di stato utilizza un modulo denominato *Internet Control Message Protocol* (ICMP) che raggruppa un insieme di messaggi di controllo.
ICMP effettua la segnalazione di problemi legati all'instradamento dei datagrammi, per altre funzioni di amministrazione e di report di stato della rete.

ICMP non rende affidabile IP, ma si limita a segnalare errori e malfunzionamenti, senza effettuare le necessarie correzioni.

Il protocollo ICMP, descritto in RFC 792, svolge funzioni di controllo per IP, quindi è incluso in tutte le implementazioni IP come un protocollo a basso livello che si appoggia direttamente su IP: risiede su ogni elaboratore, host o router, come protocollo abbinato all'IP e i pacchetti ICMP sono incapsulati all'interno di datagrammi IP.
Quando si verifica un errore o un malfunzionamento che impedisce la consegna del datagramma, viene generato un messaggio che comunica all'host che il suo destinatario è irraggiungibile.
I messaggi viaggiano nel campo dati del datagram IP e vengono manipolati dal software IP, non da applicativi utente. ▶
Quando un messaggio ICMP viene imbustato in IP viene posto al valore 1 il campo protocol.

La struttura del messaggio ICMP è la seguente:

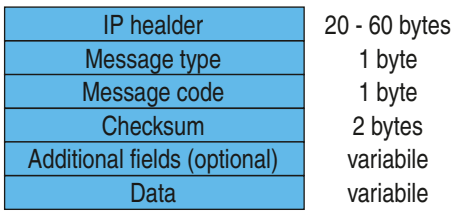
- tipo: definisce il tipo di messaggio ICMP, che può essere:
  – messaggio di errore;
  – messaggio di richiesta di informazioni.
  I valori sono riportati nella tabella che segue.

Tipo	Descrizione	Tipo	Descrizione
0	Echo Reply	13	Timestamp Request
3	Destination Unreachable	14	Timestamp Reply
4	Source Quench	15	Information Request
5	Redirect (change a route)	16	Information Reply
8	Echo Request	17	Address Mask Request
11	Time Exceeded for a Datagram	18	Address Mask Reply
12	Parameter Problem for a Datagram		

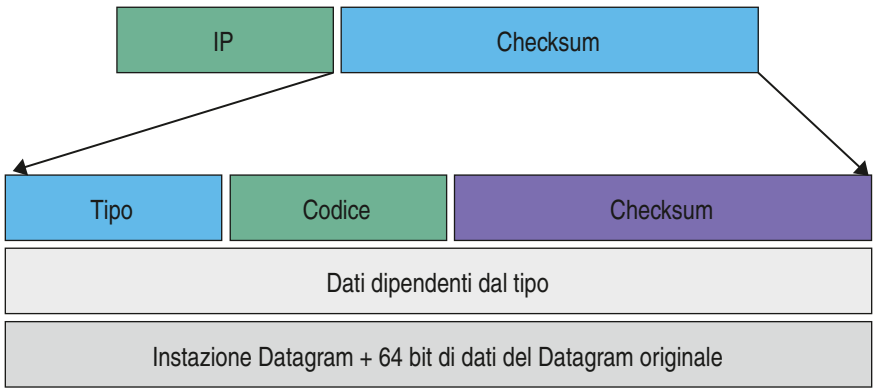
- codice: viene usato in alcuni messaggi ICMP per descrivere il tipo di errore e ulteriori dettagli; nella tabella seguente sono riportate le possibili situazioni del tipo = 3, cioè del caso di destinazione irraggiungibile (Destination Unreachable).

Tipo	Codice	Descrizione	RFC
3		DESTINATION UNREACHABLE	792
	0	Net Unreachable	792
	1	Host Unreachable	792
	2	Protocol Unreachable	792
	3	Port Unreachable	792
	4	Fragmentation Needed and Don't Fragment was Set	792
	5	Source Route Failed	1812
	6	Destination Network Unknown	1812
	7	Destination Host Unknown	1812
	8	Source Host Isolated	1812
	9	Communication with Destination Network is Administratively	1812
	10	Communication with Destination Host is Administratively	1812
	11	Destination Network Unreachable for Type of Service	1812
	12	Destination Host Unreachable for Type of Service	1812
	13	Communication Administratively Prohibited	1812

- checksum: per il controllo di errore; viene calcolato su tutto il pacchetto ICMP;
- add. field: dipendono dal tipo di messaggio ICMP;
- data: la parte rimanente viene usata per trasmettere dati legati al particolare messaggio ICMP o parte del datagramma che ha generato l'errore (figura sottostante). ▼



La figura successiva mostra un pacchetto di ICMP, del tipo error message, in cui nella parte di dati sono inclusi l'intestazione IP e altri 64 bit del pacchetto che ha generato l'errore:



Zoom su...

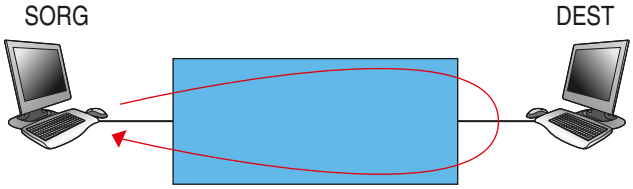
TIPI DI MESSAGGI PREVISTI DA ICMP

I messaggi ICMP sono classificati nei tipi elencati di seguito.

0-8	**Richiesta di eco e relativa risposta** (Echo Request/Reply): serve a determinare lo stato di una rete; viene inviato a qualsiasi indirizzo IP che deve restituire una risposta al trasmittente.
3	**Destinazione irraggiungibile** (Destination Unreachable): viene utilizzato da un router quando incontra problemi nel raggiungere la rete di destinazione specificata nell'indirizzo di destinazione IP.
4	**Rallentamento della sorgente** (Source Quence): viene utilizzato da un router se non possiede memoria sufficiente per accodare i datagrammi in arrivo.
5	**Reindirizzamento** (Redirect): viene inviato da un router all'host che ha spedito il datagramma indicandogli di scegliere un percorso migliore, cioè un altro router al quale indirizzare il messaggio: ricevendo il messaggio l'host può aggiornare la propria tabella di routing aggiungendo la riga che il router indica come parametro del messaggio (router alternativo).
11	**Superamento del tempo massimo di durata del datagramma** (Time Exceeded for a Datagram): viene inviato da un router quando scarta un datagramma giunto alla scadenza del tempo massimo di durata.
12	**Parametro inintelligibile** (Parameter Problem for a Datagram): questo è il messaggio di errore che viene inviato dall'host o dal router che incontra problemi nell'elaborare parte di un'intestazione IP.
13-14	**Registrazione del tempo e relativa risposta** (Timestamp Request/Reply): vengono utilizzate dai router e dagli host per determinare il ritardo con cui sono stati consegnati i dati.
15-16	**Richiesta di informazioni e relativa risposta** (Information Request/Reply): consentono a un host di identificare la rete alla quale è collegato.
17-18	**Richiesta di maschera di indirizzi e relativa risposta** (Address-Mask Request/Reply): vengono utilizzati da un host per ottenere la maschera di sottorete a cui appartiene.

Ping

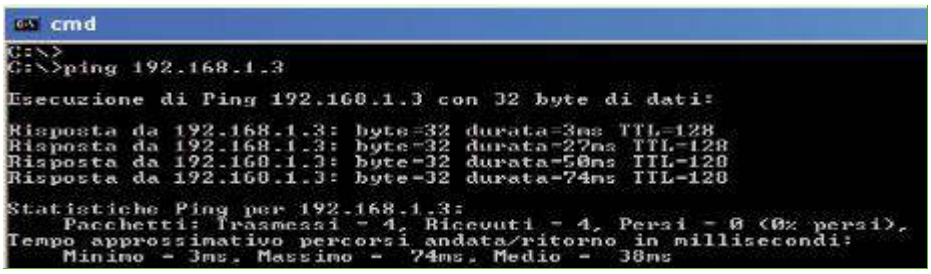
La più semplice applicazione del protocollo TCP/IP viene eseguita con il comando ping, che permette di controllare se l'host di destinazione DEST è raggiungibile o meno dal mittente SORG. ▶



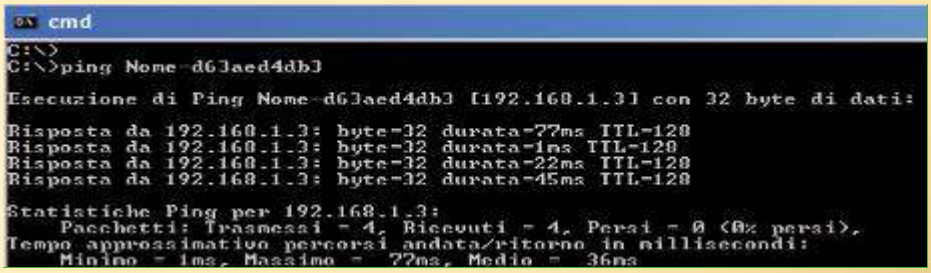
Con il comando

ping DEST

l'host mittente invia uno o più pacchetti di tipo 0 "echo request" verso una determinata destinazione e, se l'host DEST è raggiungibile da SORG, DEST risponde restituendo un pacchetto ICMP di tipo 8 "echo reply": viene effettuata contestualmente la misurazione del *Round Trip Time* (RTT), cioè del tempo necessario affinché un pacchetto compia il percorso di andata e ritorno.



Il destinatario del messaggio può essere indicato mediante il suo indirizzo IP oppure anche con il suo nome simbolico: la traduzione tra **nome simbolico** e **indirizzo IP** viene effettuata da un altro programma di servizio che esamineremo in seguito.

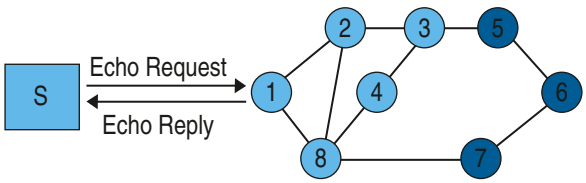


Non sempre è possibile servirsi del ping per verificare la connessione tra due host: per esempio, se la destinazione è protetta da un sistema che filtra gli ingressi in base alle porte interessate e agli applicativi coinvolti oppure se siamo in una situazione di IP masquerading, il comando ping non è utilizzabile. Inoltre la mancanza di una risposta da parte del destinatario non permette di individuare né la natura né la posizione nella quale è presente un problema; pertanto, di fatto, ping non è un buon mezzo diagnostico.

ESEMPIO 14

Nella situazione illustrata nella figura a lato i nodi 5, 6, 7 non rispondono al ping: ▶
Le cause possono essere diverse:

- potrebbero essere tutti e tre guasti;
- potrebbe essere guasto solo uno dei tre che si trova in una situazione intermedia sul percorso di andata o di ritorno dei pacchetti verso l'host "pingato";
- potrebbe non essere definito un percorso su una tabella di routing;
- ecc.



Zoom su...

OPZIONI COMANDO PING

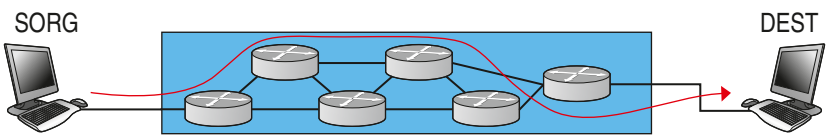
Il comando ping ha diverse opzioni, riportate nella tabella seguente.

Opzione	Descrizione
-n N	permette di specificare quanti pacchetti inviare (un pacchetto al secondo)
-l M	specifica la dimensione in byte di ciascun pacchetto
-t	esegue ping finché non viene interrotto premendo la combinazione di tasti Ctrl+C
-a	traduce l'indirizzo IP in nome DNS
-f	setta il bit *don't fragment* a 1
-i T	setta *time-to-live* = T
-w Tout	specifica un timeout in millisecondi

Maggiori informazioni si ottengono consultando direttamente l'help con il comando: ping /?

Traceroute

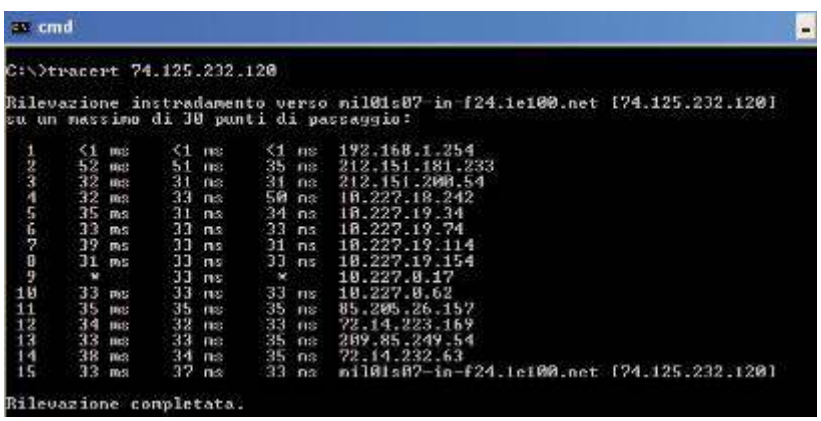
Con ping si può valutare solo la raggiungibilità di una certa destinazione, mentre l'applicativo TRACEROUTE permette di conoscere il percorso seguito dai pacchetti inviati da un mittente SORG e diretti verso una destinazione DEST.



Con il comando

tracert DEST

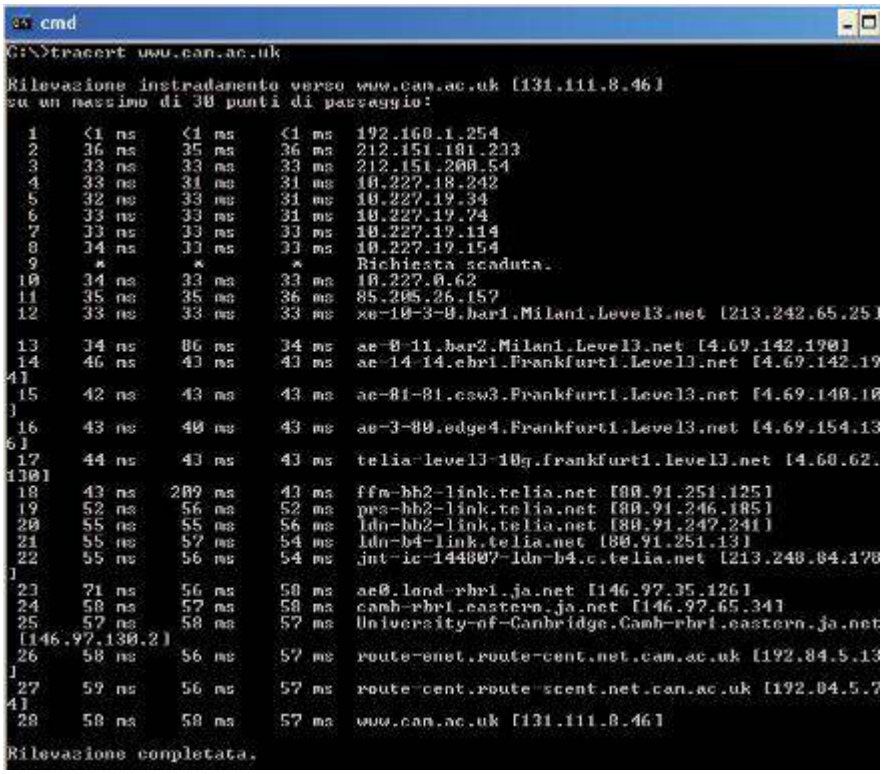
l'host SORG invia a DEST una serie di pacchetti ICMP di tipo "echo request" con un *Time-To-Live* (TTL) che inizia da un valore massimo (255 per default) per poi venire decrementato fino a 0. Ciascun nodo intermedio all'atto della rispedizione del pacchetto decrementa TTL e se un nodo rileva che tale valore è divenuto TTL = 0 invia a SORG un pacchetto ICMP di tipo "time exceeded" che indica il fallimento della consegna a destinazione del pacchetto nel suo "tempo di vita utile". Tutti i nodi attraversati da SORG a DEST vengono inseriti in una lista insieme ai tempi di percorrenza dei vari segmenti: ▶



L'output mostra il TTL, il nome ◀ DNS ▶ (ove possibile) e l'indirizzo IP dei nodi intermedi e il Round Trip Time (RTT).
(L'indirizzo "pingato" è quello del motore di ricerca Google all'indirizzo www.google.it.)

◀ DNS Il DNS (*Domain Name System*) è un sistema utilizzato per la risoluzione di nomi degli host in indirizzi IP e viceversa: consiste in un insieme di database composto da tabelle di corrispondenza tra indirizzo IP e indirizzo simbolico. ▶

Vediamo per esempio (nella schermata sottostante) il percorso da compiere per arrivare all'università di Cambridge (www.cam.ac.uk).



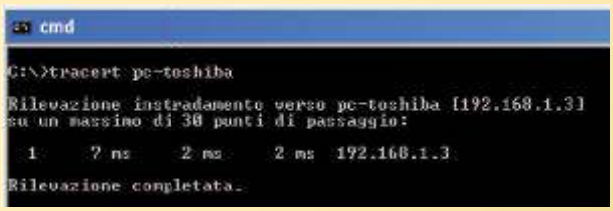
Prova adesso!

• Il comando tracert

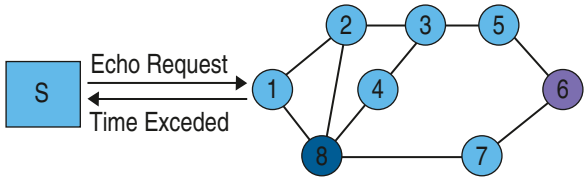
VAI ALLA LINEA DI COMANDO DEL SISTEMA OPERATIVO

Prova a visualizzare il percorso per raggiungere i seguenti server in due giorni diversi, confrontando i percorsi:
www.istruzione.it
www.berkeley.edu
www.columbia.edu
www.microsoft.com
69.171.234.64

Naturalmente è possibile "pingare" anche due host appartenenti allo stesso segmento, ma in questo caso verrà visualizzata solo una riga nella tabella, dato che viene effettuato un unico hop.



Anche il traceroute, come il ping, non ha una buona capacità diagnostica: cerchiamo di raggiungere il nodo 6 nella rete rappresentata nella figura seguente:



Ipotizziamo un guasto al nodo 8: se i percorsi di andata e ritorno sono diversi per motivazioni di traffico sulla rete, per esempio

- andata 1-2-3-5-6
- ritorno 6-5-3-4-8-1,

un guasto sul nodo 8 impedisce l'identificazione del percorso di andata oltre il nodo 2 in quanto gli unici messaggi di ritorno vengono inviati dai nodi 1 e 2; il messaggio di ritorno dal nodo 3 viene instradato verso il nodo 8 guasto, anche se in realtà il percorso di andata è perfettamente funzionante.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Quale tra i seguenti è un indirizzo di livello 4 corretto?

- ○ 23: 127.104.69.16
- ○ 127.104.69.16 : 23
- ○ 227.0.0.1 : 23
- ○ 127.104.69.255 : 23

2 Quali sono i principali vantaggi dei NAT? (*indica l'affermazione errata*)

- ○ Rimedia all'esaurimento degli indirizzi IP
- ○ Riduce la tabella di instradamento nei router
- ○ Assegna dinamicamente l'indirizzo IP all'host
- ○ Non necessita della riconfigurazione del router

3 Che cosa viene assegnato nell'IP masquerading?

- ○ Indirizzi MAC
- ○ Indirizzi IP
- ○ Porte TCP/UDP
- ○ Indirizzo di broadcast

4 Qual è la funzione del servizio PAT? (*indica l'affermazione errata*)

- ○ Traduce gli indirizzi IP
- ○ Traduce i numeri delle porte
- ○ Permette di avere molti host "dietro" a un singolo indirizzo IP
- ○ È utilizzato con il PAT statico

5 ICMP è un protocollo:

- ○ che si appoggia sul protocollo di trasporto TCP
- ○ che si appoggia sul protocollo di trasporto UDP
- ○ paritetico a IP che si appoggia direttamente sul protocollo di sottorete fisica
- ○ che si appoggia direttamente su IP al pari dei protocolli di trasporto

6 Il comando ping (*indica l'affermazione errata*):

- ○ serve per connettere due host in una LAN
- ○ utilizza il protocollo ICMP
- ○ invia pacchetti di tipo 0 "echo request"
- ○ non può essere utilizzato in caso di IP masquerading

7 Traceroute prevede l'impiego di:

- ○ particolari pacchetti UDP, detti di "discovery"
- ○ pacchetti ICMP di "discovery"
- ○ pacchetti ICMP di "echo request", "echo replay" e "time exceded"
- ○ meccanismi di tracciamento della connessione insiti

8 Quali campi non sono presenti nel record ICMP?

- ○ tipo
- ○ codice
- ○ priorità
- ○ checksum
- ○ add. field
- ○ contatore
- ○ data
- ○ protocol

Test vero/falso

1	Il servizio WEB utilizza generalmente la porta 80.	V F
2	Un TSAP address ha la forma (IP address: port number).	V F
3	Il valore di porta logica può essere compreso tra 0 e 65535.	V F
4	Il NAT offre a ogni host della rete privata la possibilità di accedere a un indirizzo pubblico.	V F
5	Con "stub domain" si intende la rete interna che usa quindi solo indirizzi IP interni.	V F
6	Il caso tipico di utilizzo dell'IP masquerading è quello delle reti aziendali con un solo IP pubblico.	V F
7	La static NAT contiene una mappatura statica uno-a-uno tra indirizzi interni ed esterni.	V F
8	Il protocollo IP offre un servizio di tipo "best effort".	V F
9	ICMP rende affidabile il protocollo IP.	V F
10	Il comando PING permette di controllare se l'host di destinazione DEST è raggiungibile.	V F
11	Il traceroute a differenza del ping ha una buona capacità diagnostica.	V F

Verifichiamo le competenze

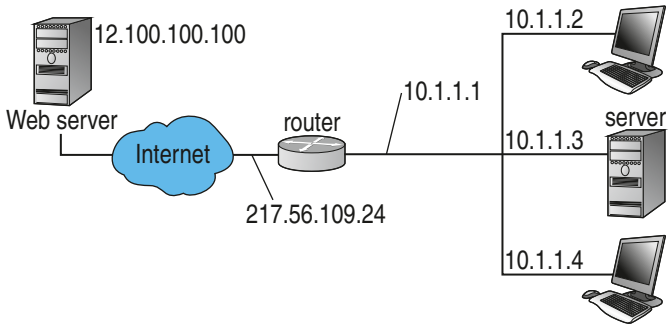
Esprimi la tua creatività

Esercizi di completamento

1 **Indica il significato di ognuno degli acronimi elencati di seguito.**

TPDU .....
TSAP address .....
NSAP address .....
NAT .....
PAT .....
NAPT .....
ICMP .....
TTL .....
RTT .....
DNS .....

2 Data la rete di cui alla figura sottostante, scrivi una possibile NAT table per il router nel caso che ogni host abbia effettuato un accesso a Internet (per esempio, la numerazione delle porte del router inizia da 55000).



NAT table				
Mittente			Destinatario	
ID	IP address	Int port	Ext address	Est port
1				
2				
3				
4				

ESERCITAZIONI DI LABORATORIO 1

PROTOCOLLO ICMP

In questa esercitazione utilizzeremo Wireshark per studiare messaggi ICMP generati dal programma Ping.

I riferimenti teorici per affrontare questa esercitazione sono presenti nell'Unità didattica 5 del Modulo 6 e nella specifica del protocollo ICMP che puoi scaricare da http://tools.ietf.org/html/rfc792 con le successive integrazioni.

Il programma Ping consente di verificare il collegamento (o la presenza attiva) di un host sulla rete, conoscendo il suo indirizzo IP (oppure il suo nome, ma come questo avviene lo vedremo nel corso del prossimo anno).
Il calcolatore sorgente invia un pacchetto (in gergo si dice che "pinga") al destinatario che se è presente e attivo sulla rete gli risponde con un messaggio di conferma.
Entrambi i messaggi appartengono al protocollo ICMP.

Prima di iniziare a catturare i dati con Wireshark è necessario conoscere l'indirizzo di un host da contattare: puoi individuare un PC sulla tua rete locale privata oppure anche un host di Internet (per esempio 89.97.132.192, che è l'indirizzo IP del sito della Pubblica Istruzione www.istruzione.it).

Manda in esecuzione Wireshark e al prompt dei comandi digita

ping -10 89.97132.192

dove con l'opzione -10 viene indicato di spedire 10 pacchetti: visualizzerai una situazione come quella riportata nella schermata sottostante.

```
C:\>ping -n 10 89.97.132.192

Esecuzione di Ping 89.97.132.192 con 32 byte di dati:

Risposta da 89.97.132.192: byte=32 durata=44ms TTL=48
Risposta da 89.97.132.192: byte=32 durata=44ms TTL=48
Risposta da 89.97.132.192: byte=32 durata=43ms TTL=48
Risposta da 89.97.132.192: byte=32 durata=44ms TTL=48
Risposta da 89.97.132.192: byte=32 durata=43ms TTL=48
Risposta da 89.97.132.192: byte=32 durata=44ms TTL=48
Risposta da 89.97.132.192: byte=32 durata=43ms TTL=48
Risposta da 89.97.132.192: byte=32 durata=44ms TTL=48
Risposta da 89.97.132.192: byte=32 durata=43ms TTL=48
Risposta da 89.97.132.192: byte=32 durata=44ms TTL=48

Statistiche Ping per 89.97.132.192:
    Pacchetti: Trasmessi = 10, Ricevuti = 10, Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 43ms, Massimo =  44ms, Medio =  43ms
```

L'esito di questo comando è che tutti i 10 pacchetti inviati sono andati a buon fine con un tempo medio (round trip time RTT) di 43 ms.

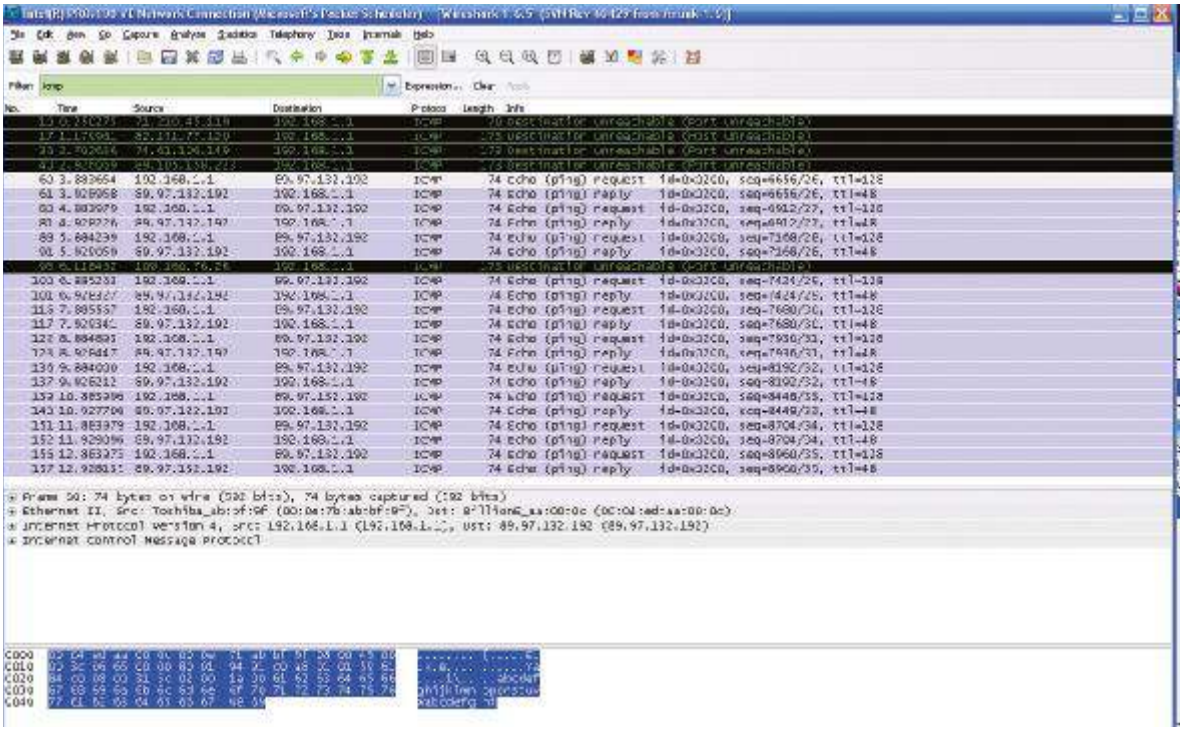
Per prova, digita ora

ping -5 202.9.94.53

```
C:\>ping -n 5 202.9.94.53

Esecuzione di Ping 202.9.94.53 con 32 byte di dati:

Risposta da 202.9.94.53: byte=32 durata=350ms TTL=32
Risposta da 202.9.94.53: byte=32 durata=351ms TTL=32
Risposta da 202.9.94.53: byte=32 durata=350ms TTL=32
Risposta da 202.9.94.53: byte=32 durata=349ms TTL=32
Risposta da 202.9.94.53: byte=32 durata=350ms TTL=32

Statistiche Ping per 202.9.94.53:
    Pacchetti: Trasmessi = 5, Ricevuti = 5, Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 349ms, Massimo =  351ms, Medio =  350ms
```

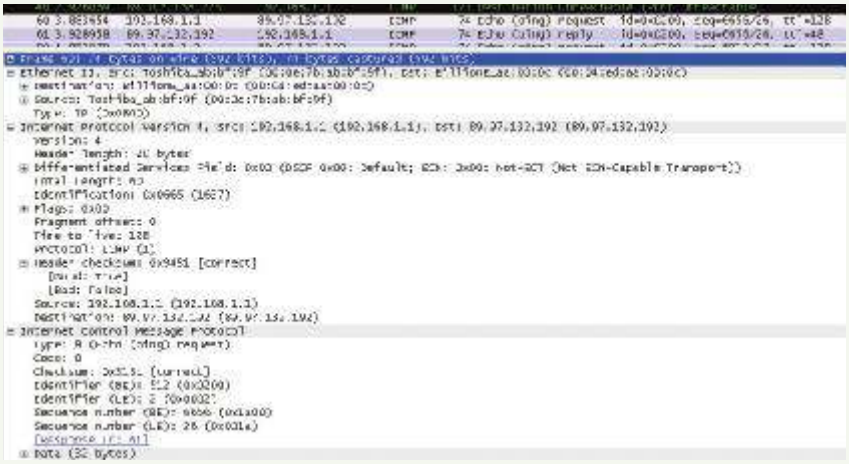
… hai scambiato 5 pacchetti con Sydney, in Australia, con un TTL di 349 ms.

Ferma ora la cattura dei pacchetti da parte Wireshark e per visualizzare solo i pacchetti del protocollo ICMP digita il comando all'interno del filtro, ottenendo una situazione simile a quella della figura riportata di seguito:



Prova adesso!

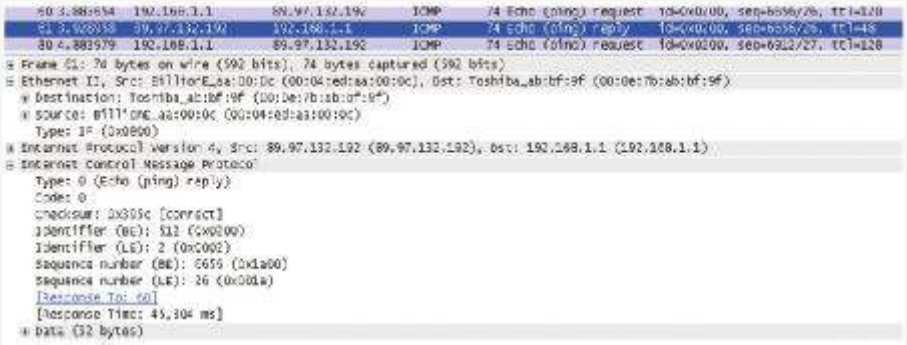
Seleziona il primo pacchetto inviato (nel nostro esempio è il frame numero 60) e visualizza i dettagli della descrizione dei singoli componenti del frame, come riportato nella figura seguente.



Stampa il pacchetto selezionato da File → Print, attiva Selected packet only e Packet summary line scegliendo la minima quantità di dettagli che è necessaria per rispondere alle seguenti domande.

1. Quanti byte compongono il pacchetto?
2. Qual è l'indirizzo sorgente, cioè il vostro host?
3. Qual è l'indirizzo di destinazione del frame?
4. Individua il campo Type nel frame Ethernet. Che posizione ha all'interno del frame? Quali bit hanno valore 1?
5. Individua il campo Code nel frame Ethernet. Che posizione ha all'interno del frame? Che valore ha?
6. Da quanti byte è formato il campo di checksum?
7. Da quanti byte è formato il campo dell'identificatore di sequenza?
8. Da quanti byte è formato il campo del numero di sequenza?

Successivamente individua il messaggio di risposta corrispondente (figura sottostante).



1. Come si fa a individuarlo?
2. Quanti byte compongono il pacchetto?
3. Quali valori ha il campo Type?
4. Quali valori ha il campo Code?
5. Da quanti byte è formato il campo di checksum?
6. Da quanti byte è formato il campo dell'identificatore di sequenza?
7. Da quanti byte è formato il campo del numero di sequenza?
8. Quali sono le differenze rispetto al frame di richiesta?

Zoom su...

UTILIZZO DI NOMI SIMBOLICI

È possibile eseguire il comando ping seguito dal nome simbolico del sito: la sostituzione del nome simbolico con l'indirizzo IP viene fatta dal DNS (*Domain Name System*), che verrà studiato nel corso del prossimo anno (consiste in un insieme di database composto da tabelle di corrispondenza tra indirizzo IP e indirizzo simbolico).
Per esempio, se digiti ping www.berkeley.edu otterrai la schermata sottostante.

```
C:\>ping www.berkeley.edu

Esecuzione di Ping www.w3.berkeley.edu [169.229.131.81] con 32 byte di dati:

Risposta da 169.229.131.81: byte=32 durata=191ms TTL=35
Risposta da 169.229.131.81: byte=32 durata=191ms TTL=35
Risposta da 169.229.131.81: byte=32 durata=192ms TTL=35
Risposta da 169.229.131.81: byte=32 durata=192ms TTL=35

Statistiche Ping per 169.229.131.81:
    Pacchetti: Trasmessi = 4, Ricevuti = 4, Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 191ms, Massimo =  192ms, Medio =  191ms
```

Analogamente, se digiti ping www.istruzione.it otterrai la schermata sottostante.

```
C:\>ping www.istruzione.it

Esecuzione di Ping iostudio.pubblica.istruzione.it [89.97.132.192] con 32 byte d
i dati:

Risposta da 89.97.132.192: byte=32 durata=907ms TTL=51
Risposta da 89.97.132.192: byte=32 durata=456ms TTL=51
Risposta da 89.97.132.192: byte=32 durata=466ms TTL=51
Risposta da 89.97.132.192: byte=32 durata=406ms TTL=51

Statistiche Ping per 89.97.132.192:
    Pacchetti: Trasmessi = 4, Ricevuti = 4, Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 406ms, Massimo =  907ms, Medio =  558ms
```

ESERCITAZIONI DI LABORATORIO 2

PROTOCOLLO IP E FRAMMENTAZIONE

In questa esercitazione utilizziamo Wireshark per studiare il protocollo IP analizzando i pacchetti trasmessi dal programma Traceroute o PingPlotter, a seconda del sistema operativo:

- con Windows si utilizza il pacchetto PingPlotter;
- con Linux/Unix si possono utilizzare sia Tracerout che PingPlotter.

I riferimenti teorici per affrontare questa esercitazione sono presenti nelle Unità didattiche 1 e 6 del Modulo 6 e nella specifica del protocollo IP all'indirizzo ftp://ftp.rfc-editor.org/in-notes/rfc791.txt.

Procederemo inviando datagrammi di dimensioni diverse allo stesso indirizzo IP di destinazione e individueremo quando i datagrammi verranno frammentati analizzandoli dettagliatamente.

Zoom su...

TRACEROUTE

Ricordiamo brevemente il funzionamento di Traceroute:

- inizia a trasmettere più volte lo stesso diagramma incrementando a ogni trasmissione il campo *time-to-live* (TTL) nell'intestazione IP a partire dal valore 1 fino a un valore massimo in modo che volta per volta possa "aumentare" la distanza percorsa sulla rete (numero di hop);
- i router analizzano questo campo e lo decrementano prima di ritrasmetterlo al prossimo router;
- se il valore del campo TTL raggiunge il valore 0, il router non inoltra il datagramma ma manda un segnale di errore ICMP di risposta al mittente (tipo 11 – TTL-exceeded).

Il router potrebbe diminuire il valore TTL anche di un valore superiore all'unità: RFC 791 segnala che il router deve decrementare il campo TTL di almeno uno, ma non indica il valore massimo.

In questo modo ogni router apprende la strada di andata e ritorno dal mittente al destinatario "seguendo" gli indirizzi dei datagrammi di errore che gli vengono inviati.

Spedire pacchetti di dimensione variabile

A Se si utilizza il sistema operativo Linux/Unix è possibile indicare sulla linea di comando la dimensione del datagramma come parametro: per esempio, volendo inviare un datagramma di dimensione di 3000 byte al sito della Pubblica Istruzione il comando è il seguente:

```
traceroute  www.istruzione.it  3000
```

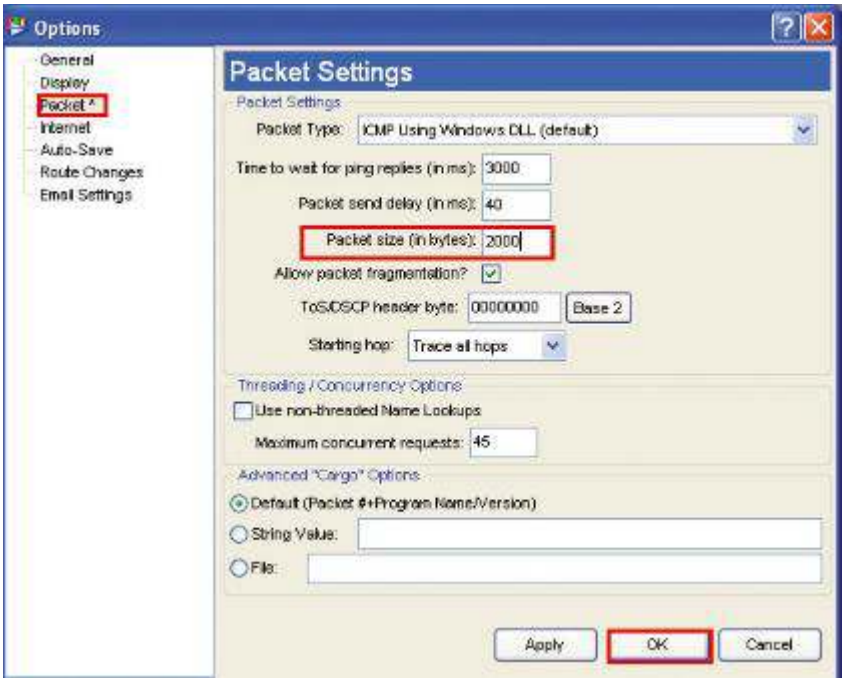
B Se si utilizza il sistema operativo Windows è necessario scaricare un nuovo programma shareware dal sito http://www.pingplotter.com in quanto Traceroute non ci permette di definire datagrammi di dimensione variabile e quindi non ci permette di effettuare tutte le operazioni previste per questa esercitazione.

L'interfaccia di questo nuovo programma è quella che appare nella figura seguente.

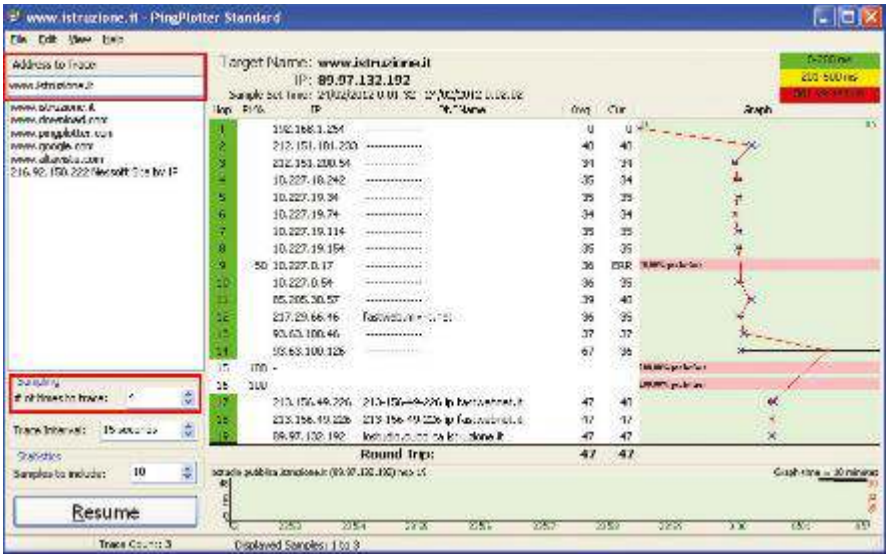
Per utilizzare PingPlotter basta selezionare l'opzione Options dal menu Edit.

Quindi seleziona Packet* (figura che segue) tra le possibili opzioni e modifica la dimensione nel campo Packet Size che di default è di 56 byte.

Il programma PingPlotter, a differenza del comando traceroute, offre molteplici possibilità di personalizzazione: si consiglia di "dedicare un po'di tempo" a studiarlo e scoprirne tutte le potenzialità dato che in futuro questo programma verrà sempre più utilizzato grazie alla sua potenzialità e flessibilità.



1. Manda in esecuzione Wireshark e inizia a catturare i pacchetti.
2. Avvia la trasmissione dei pacchetti con PingPlotter inserendo:
   – l'indirizzo di destinazione www.istruzione.it nel campo Address to Trace;
   – il valore 4 nel campo # of times to Trace per limitare il tempo di attesa.
3. Clicca sul pulsante di [start] per ottenere una videata come la seguente:

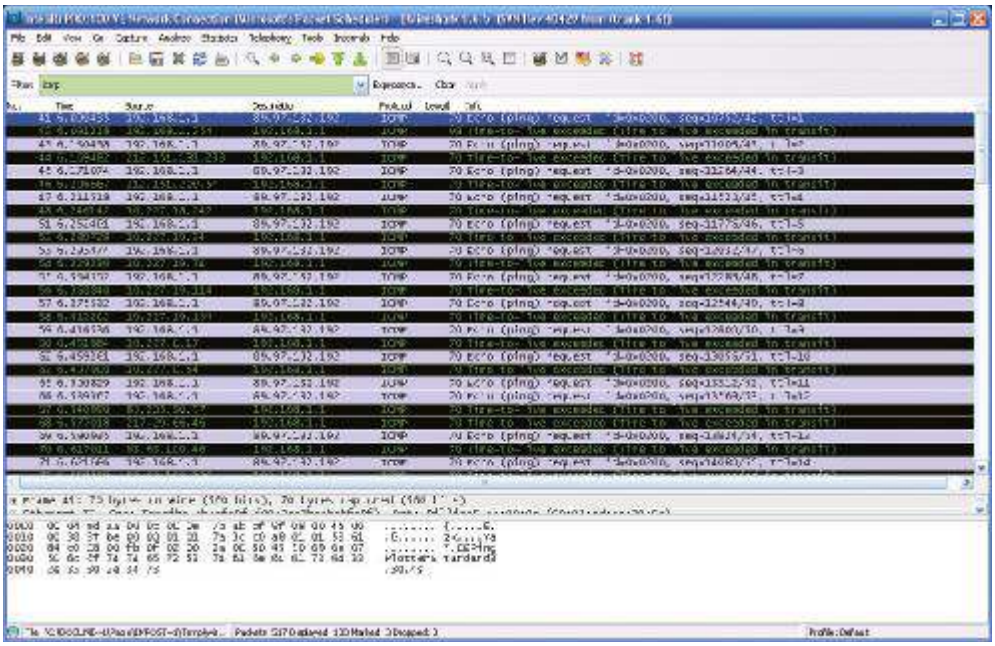


4. Ripeti questa operazione modificando la dimensione del datagramma digitando rispettivamente nel campo packet size 56, 2000, 3000 dell'opzione Packet sopra descritta confermando con [Apply] e [OK] e riavviando la trasmissione mediante il pulsante [Resume].

Con il sistema operativo Linux/Unix basta modificare il parametro sulla linea di comando e ripeterlo tre volte:

- traceroute www.istruzione.it 56
- traceroute www.istruzione.it 2000
- traceroute www.istruzione.it 3000

Ferma ora la cattura dei pacchetti da parte di Wireshark: per visualizzare solo i pacchetti del protocollo ICMP digita (in minuscolo) icmp all'interno del campo filtro, ottenendo una situazione simile a quella riportata di seguito:



Prova adesso!

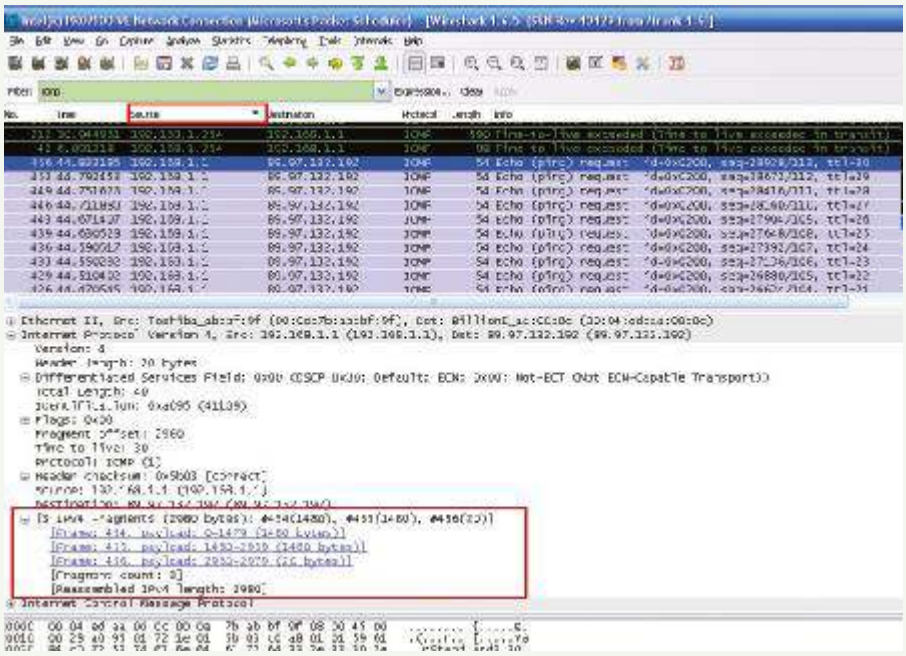
Seleziona il primo pacchetto inviato dal tuo host ed espandi il dettaglio della sezione relativa a Internet Protocol nella finestra centrale, ottenendo una videata simile alla seguente:

```
Frame 41: 70 bytes on wire (190 bits), 70 bytes captured (190 bits)
Ethernet II, Src: Toshiba_ab:bf:9f (00:08:7b:ab:bf:9f), Dst: 3ComEuro_aa:00:0c (00:04:ed:aa:00:0c)
Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 80.97.132.192 (80.97.132.192)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 56
    Identification: 0x9fbe (40894)
  ⊞ Flags: 0x00
    Fragment offset: 0
  ⊞ Time to live: 1
    Protocol: ICMP (1)
  ⊞ Header checksum: 0x7a3c [correct]
    Source: 192.168.1.1 (192.168.1.1)
    Destination: 80.97.132.192 (80.97.132.192)
Internet Control Message Protocol
```

1. Qual è l'indirizzo IP del tuo computer?
2. Qual è la dimensione dell'header IP?
3. In quale posizione è il campo con la dimensione del pacchetto?

4 All'interno dell'intestazione IP, qual è il valore del protocollo di livello superiore?
5 Quanti byte ci sono nell'intestazione IP?
6 Quanti byte sono utilizzati per il checkum?
7 Questo datagramma IP è stato frammentato? Se si, in quante parti?

Ordina ora i pacchetti in base all'indirizzo IP: fai clic due volte sull'intestazione della colonna Source in modo da avere i pacchetti ordinati per indirizzi e in ordine decrescente, come nella figura seguente; quindi posizionati sul primo pacchetto trasmesso.



Scorri in verticale i datagrammi e osserva la composizione del dettaglio evidenziato.

1 Quali campi nei datagrammi IP cambiano *sempre* da un datagramma al successivo ?
2 Quali campi nei datagrammi IP rimangono costanti?
3 Individua i datagrammi che hanno il medesimo TTL: quali differenze presentano? Perché?
4 Quali sono i valori del campo Identification e del campo TTL?
5 Individua i datagrammi trasmessi al 5 hop: quanti te ne aspetti di trovare?
6 Ordinali cronologicamente e confrontali tra loro.
7 Cosa rimane invariato nei pacchetti ICMP TTL-exceeded inviati al tuo computer dal router di primo hop?
8 Ordina di nuovo i pacchetti, questa volta in base al tempo, facendo clic sulla colonna Time e individua il primo segmento di dimensione 2000 byte che è stato inviato: che osservazione puoi fare?
9 Stampa il primo frammento del datagramma IP frammentato: da quale dato capisci che fa parte di un messaggio frammentato?
10 Da dove capisci che è il primo frammento della sequenza oppure uno successivo?
11 Quanto è lungo questo datagramma IP?
12 Stampa il secondo frammento del datagramma IP frammentato: da quale dato capisci che non è il primo frammento del messaggio?
13 Quali campi cambiano nell'intestazione IP tra il primo e il secondo frammento?
14 Individua il primo messaggio di 3500 byte: in quanti frammenti è stato diviso? Perché?

ESERCITAZIONI DI LABORATORIO 3

PROTOCOLLO ICMP E TRACEROUTE

In questa esercitazione utilizzeremo Wireshark per studiare messaggi ICMP generati dal programma Traceroute.

Traceroute può essere usato per scoprire il percorso che un pacchetto segue dalla sorgente alla destinazione ed è stato descritto nella Unità didattica 5 del Modulo 6 e nella precedente esercitazione di laboratorio.

1. Manda in esecuzione Wireshark
2. Al prompt dei comandi digita

   tracert www.paris.fr

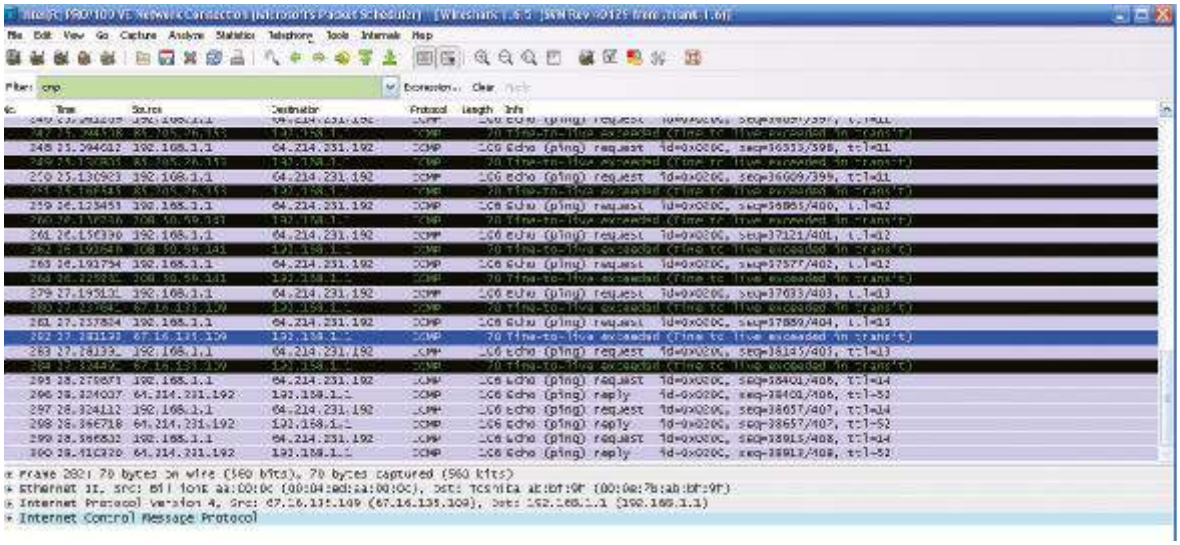
   oppure

   tracert 64.214.231.192

   (Sappiamo che il nome simbolico verrà tradotto in indirizzo IP dal DNS).
3. La schermata di Traceroute è la seguente:

```
C:\>tracert www.paris.fr

Rilevazione instradamento verso a1193.b.akamai.net [64.214.231.192]
su un massimo di 30 punti di passaggio:

  1    <1 ms    <1 ms    <1 ms  192.168.1.254
  2    37 ms    37 ms    36 ms  212.151.181.233
  3    32 ms    33 ms    33 ms  212.151.200.54
  4    32 ms    33 ms    31 ms  10.227.10.242
  5    33 ms    33 ms    33 ms  10.227.19.34
  6    33 ms    33 ms    33 ms  10.227.19.74
  7    33 ms    34 ms    34 ms  10.227.19.114
  8    33 ms    33 ms    33 ms  10.227.19.154
  9     *       32 ms     *     10.227.0.17
 10    34 ms    33 ms    33 ms  10.227.0.2
 11    39 ms    35 ms    35 ms  85.205.26.153
 12    33 ms    32 ms    33 ms  Te7-1.csr1.LIN1.gblx.net [208.50.59.141]
 13    43 ms    47 ms    43 ms  po5.ar3.FRA3.gblx.net [67.16.135.109]
 14    43 ms    42 ms    43 ms  64.214.231.192

Rilevazione completata.
```

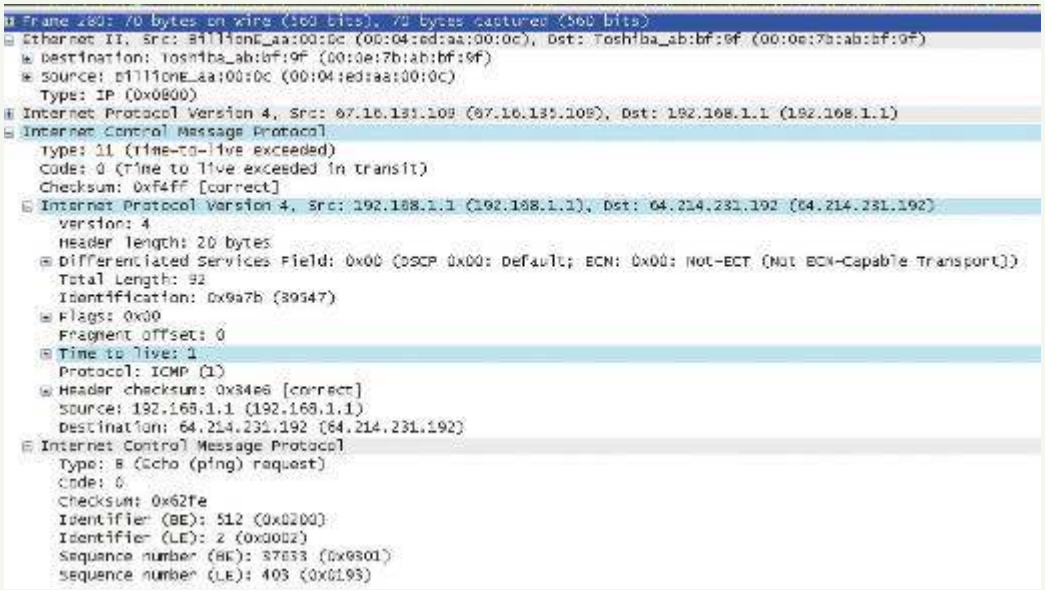
4. Osserva che dopo 14 tentativi il pacchetto è giunto a destinazione a Parigi.
5. Ferma ora la cattura dei pacchetti da parte di Wireshark.
6. Per visualizzare solo i pacchetti del protocollo ICMP lo devi digitare in minuscolo all'interno del filtro, ottenendo una situazione simile a quella riportata nella figura che segue.



Si può osservare che il calcolatore sorgente invia non un solo pacchetto ping (come il programma PingPlotter che abbiamo utilizzato nella precedente esercitazione), ma tre pacchetti per ogni valore di TTL: tutti quelli che non giungono a destinazione in quanto arrivano a un router con valore TTL decrementato a 0 vengono rispediti al mittente con messaggio di errore ICMP TTL-exceeded.

Prova adesso!

Seleziona un pacchetto ICMP di errore restituito da un router (nel nostro esempio è il frame numero 280) e visualizza i dettagli della descrizione dei singoli componenti del frame, come riportato nella figura seguente.



Stampa il pacchetto selezionato da File → Print, attiva Selected packet only e Packet summary line scegliendo la minima quantità di dettagli necessaria per rispondere alle seguenti domande.

1. Quanti byte compongono il pacchetto?
2. Confronta la dimensione del frame rispetto a quello di un messaggio ping: cosa si può constatare?
3. Qual è l'indirizzo IP sorgente, cioè quello del tuo host?
4. Qual è l'indirizzo IP di destinazione del frame?
5. Individua il campo TTL: che valore ti aspetti di trovare?
6. In che cosa si differenziamo i pacchetti spediti con il medesimo numero di TTL?
7. Da quanti byte è formato il campo di checksum?
8. Che valore ha il campo Code?
9. Che valore ha il campo Type?
10. Confronta i pacchetti di errore ricevuti: in che cosa si differenziano?
11. Individua i frame che sono giunti a destinazione e le loro corrispondenti risposte: come vengono accoppiate?
12. Qual è il valore TTL di questo messaggio? Come viene definito?
13. Qual è la dimensione di un messaggio replay da tracert rispetto a un ping?

Manda ora in esecuzione Wireshark e al prompt dei comandi digita un indirizzo IP a caso nella rete privata, dopo esserti assicurato della presenza dell'host di destinazione, utilizzando il comando arp –a; per esempio digita

tracert 192.168.1.1

e dai una spiegazione a quello che osservi sia nella finestra di tracert sia in quella di Wireshark, analizzando ogni frame che vedi sullo schermo (figura sottostante).

ESERCITAZIONI DI LABORATORIO 4

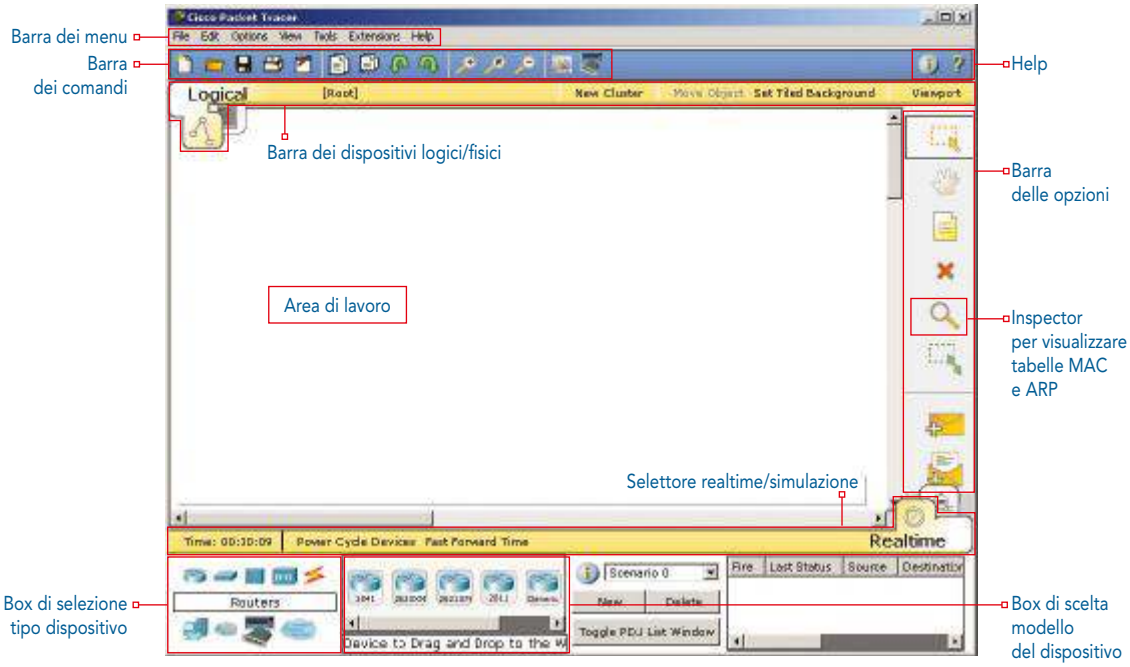
L'EMULATORE CISCO PACKET TRACER

Cisco Packet Tracer è un software didattico per l'emulazione di apparati di rete, distribuito liberamente a studenti e istruttori del Programma Cisco Networking Academy, che permette di creare topologie di rete composte da apparati generici e/o proprietari di Cisco.

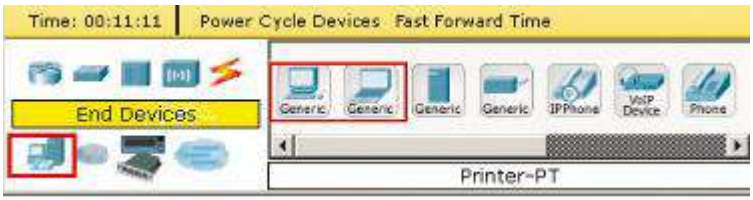
Nato da un team di programmatori, istruttori ed ex studenti del programma sotto la guida di Dennis Frezzo di Cisco Systems, è un ottimo strumento per la simulazione di rete che facilita l'apprendimento del networking.

Tramite una semplice interfaccia, GUI permette di configurare gli apparati di rete e verificarne il funzionamento mediante la creazione di scenari di traffico: osservando il corrispondente comportamento della rete, consente di ispezionare dinamicamente in ogni momento lo stato di ciascun dispositivo e il formato di ciascun pacchetto inviato sulla topologia di rete.

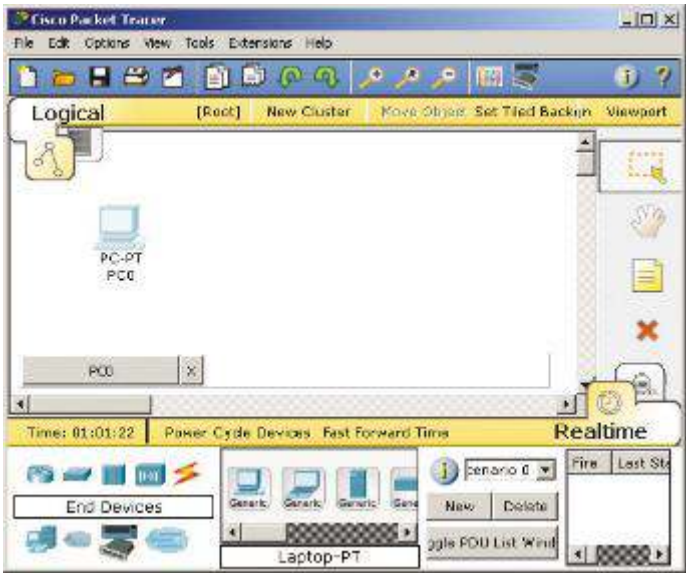
Dopo aver installato il programma, alla sua esecuzione viene presentata la seguente schermata:



I componenti della rete (computer, switch e cavi di collegamento) possono essere prelevati dal menù in basso a sinistra, dove è possibile selezionare il tipo di dispositivo desiderato tra computer, connettori, dispositivi di commutazione ecc. ▶
Facendo clic su [End Devices] (ultima icona in basso a sinistra) nella parte bassa centrale vengono elencati i vari dispositivi terminali: Generic PC, Generic laptop, Generic server ecc. ▼



Selezioniamo un generico PC e trasciniamolo nell'area di lavoro: packet tracer gli assegna un nome composto da "PC" e un numero progressivo (che può essere modificato a piacere). ▶
A titolo di esempio, realizziamo una rete con 5 PC alla quale assegneremo un indirizzo di classe C a partire da 192.168.0.0: disponiamo quindi cinque dispositivi nell'area di lavoro. ▼



Il PC attualmente in uso è quello che viene visualizzato in chiaro.

Per connettere i PC utilizziamo uno switch: lo possiamo scegliere tra quelli disponibili selezionando nella finestra in basso a sinistra la seconda icona, che fa apparire nella finestra dei componenti l'elenco dei dispositivi disponibili.

Scegliamo il primo, il 2950-24, uno switch a 24 porte, fin troppo "grande" per il nostro progetto.

Trasciniamolo nel nostro progetto, al centro della rete: otterremo una situazione come quella rappresentata nella figura a lato. ▶
Per collegare i 5 PC allo switch dobbiamo selezionare un mezzo trasmissivo facendo innanzitutto clic sull'icona delle connessioni (il lampo giallo) nella finestra dei dispositivi: ora nella finestra centrale appaiono i vari tipi di cavi di connessione: connessione automatica, cavo console, cavo diritto (per collegare il PC allo switch), cavo cross, collegamento wireless ecc. ▼

Utilizziamo il cavo diritto per i collegamenti: facciamo clic sul pulsante contrassegnato dalla linea nera continua e successivamente su PC0, scegliendo di connetterlo alla porta FastEthernet (figura A); a questo punto trasciniamo il cavo (figura B) fino allo switch dove scegliamo di collegarlo alla porta Ethernet 0/1 (figura C).

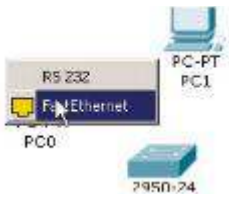


figura A

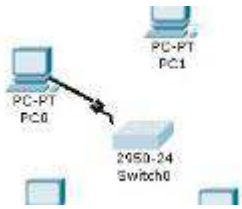


figura B

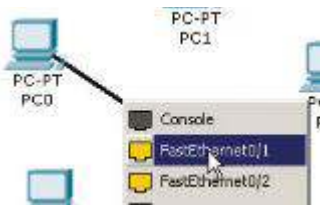
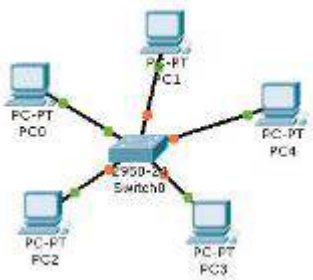


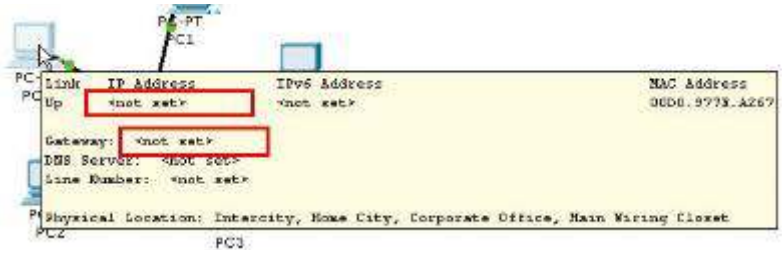
figura C

Ripetiamo la stessa procedura anche per gli altri PC fino a completare la rete, come si evince dalla figura che segue.



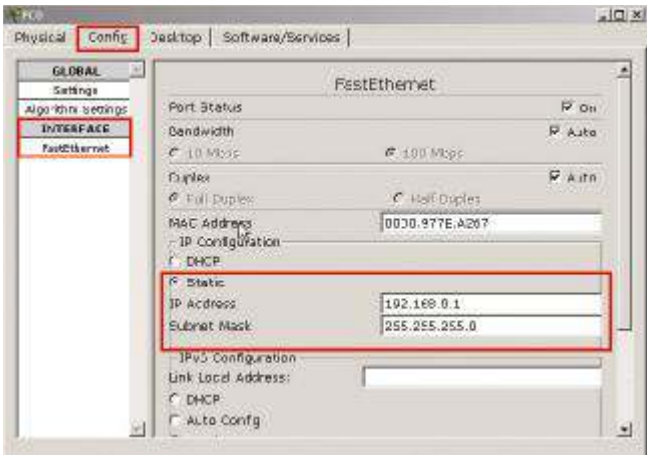
Ai capi di ogni link sono rappresentati dei "led" che indicano lo stato dell'interfaccia relativa, e possono essere di tre colori:
verde: indica che l'interfaccia è UP (è lampeggiante quando c'è attività sul link);
rosso: indica che l'interfaccia è DOWN;
ambra: l'interfaccia è "BLOCCATA" in attesa che termini il processo di loop-breaking (questo stato può manifestarsi solo sulle interfacce degli switch).

Passando il puntatore del mouse su un qualsiasi PC (figura che segue) si apre una finestra che mostra le principali impostazioni di quel computer: possiamo osservare che i PC non sono configurati.



Facciamo clic su PC0 e otteniamo la finestra riprodotta a lato, che consente di configurare tale computer. ▶
Selezioniamo la scheda "Config" e premiamo il pulsante "INTERFACE FastEthernet": nella schermata di destra, nei campi "Static IP Address" e "Subnet Mask", possiamo ora inserire i valori 192.168.0.1 e 255.255.255.0 come mostrato nella figura precedente.

In questa scheda è possibile osservare che sono già selezionate la velocità della banda passante, impostata a 100 Mbps, e la modalità di comunicazione bidirezionale, cioè Full Duplex.

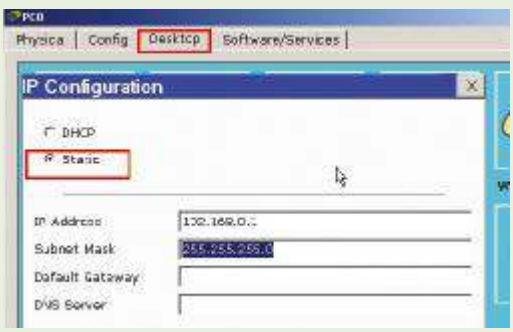


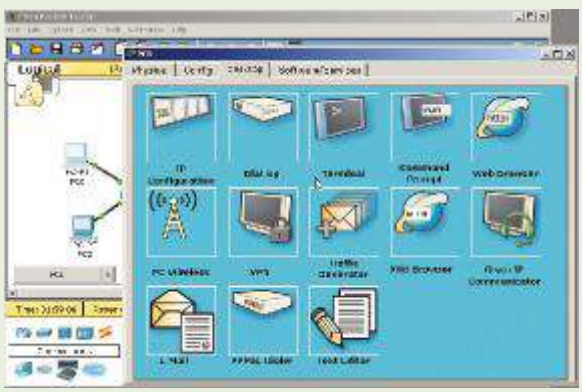
Viene anche assegnato un indirizzo fisico MAC alla scheda FastEthernet: nel nostro caso abbiamo 000D.977E.A267 (12 digit in formato esadecimale, quindi indirizzo MAC a 48 bit).

Zoom su...

CONFIGURAZIONE HOST

È anche possibile inserire la configurazione selezionando la scheda Desktop e successivamente l'icona IP configuration. ▶
Otterremo in questo modo la schermata della figura che segue. ▼





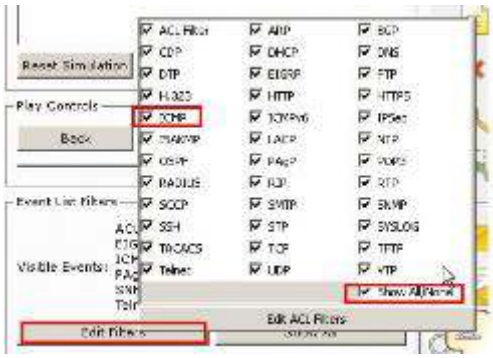
Configuriamo tutti gli indirizzi IP come indicato nella tabella a lato. ▶

Host	Indirizzo IP	Subnet-mask
PC0	192.168.0.1	255.255.255.0
PC1	192.168.0.2	255.255.255.0
PC2	192.168.0.3	255.255.255.0
PC3	192.168.0.4	255.255.255.0
PC4	192.168.0.5	255.255.255.0

La configurazione IP è statica: vedremo in seguito come effettuare anche l'assegnazione dinamica mediante la modalità DHCP (*Dinamic Host Controlled Protocol*) nel caso sia presente un router che possa offrire questo servizio.

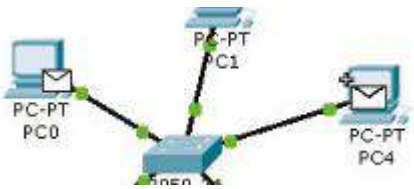
A questo punto possiamo simulare il funzionamento della rete: bisogna passare dalla stato di Realtime (quello che permette l'editing) a quello di Simulation (simulazione) facendo clic sull'icona in basso a destra della figura a lato. ▶
Ora abbiamo una videata con una finestra nella parte destra dello schermo denominata Event List: conviene visualizzare solamente il protocollo ICMP (Internet Control Message Protocol) per seguire il trasferimento del pacchetto dal mittente al destinatario e quello di risposta che va dal destinatario al mittente, a conferma dell'avvenuta ricezione.

Per far ciò si preme il pulsante Edit Filter (figura a lato), si deselezionano tutti i filtri spuntando la casella Show All/None e, dopo aver controllato che tutti i filtri risultino disabilitati, si abilita soltanto ICMP. ▶

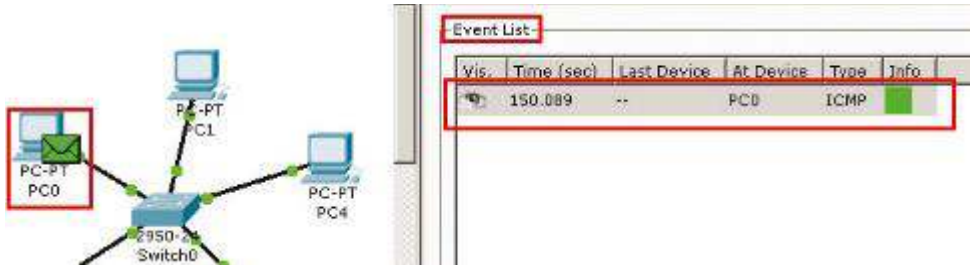


Nella figura che segue facciamo clic sul pulsante contrassegnato dal simbolo della busta chiusa, posto a destra in basso e portiamolo, trascinandolo col mouse, prima su PC0 ▼

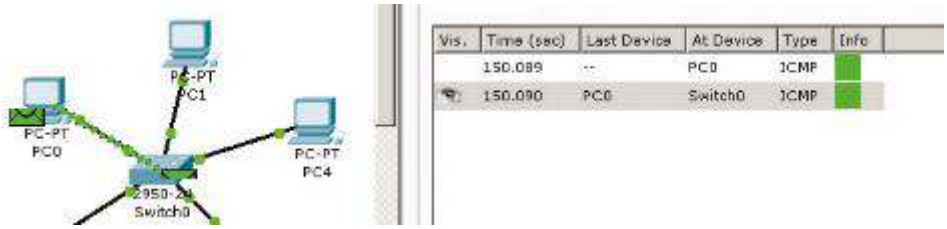
e quindi su PC4:



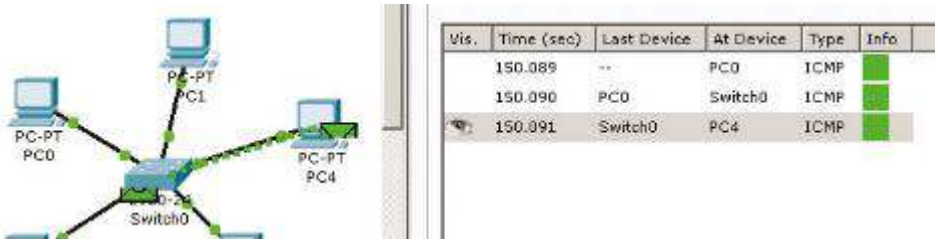
Il pacchetto, ormai pronto a partire, viene visualizzato nella finestra Event List.



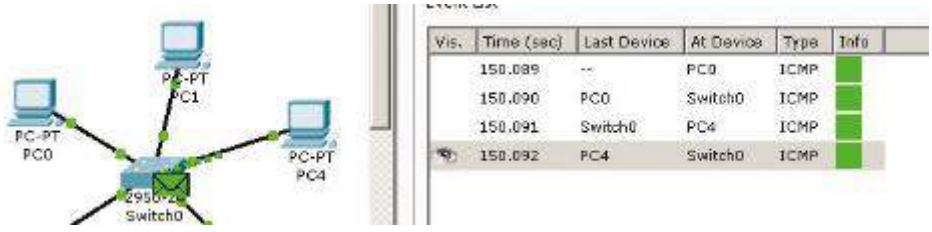
Iniziamo la spedizione del messaggio facendo clic sul pulsante [Capture/Forward]: al secondo clic il pacchetto parte da PC0 e viaggia verso Switch0:



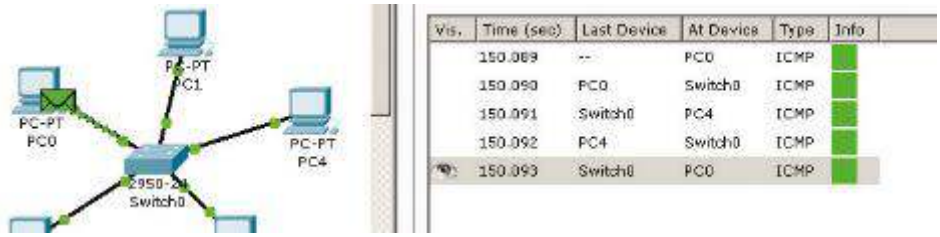
al terzo clic il messaggio da Switch0 raggiunge PC4:



al quarto clic da PC4 il messaggio raggiunge Switch0:



al quinto clic il pacchetto raggiunge PC0, chiudendo il percorso di andata e ritorno.



I cinque passaggi sarebbero stati effettuati in modo automatico se avessimo premuto il pulsante [Auto Capture/Play]. Effettuando questa operazione vedremo il pacchetto spostarsi da PC0 a Switch0 e poi da questo a PC4: quest'ultimo risponderà con un pacchetto che, dopo aver raggiunto Switch0, ritornerà infine a PC0.

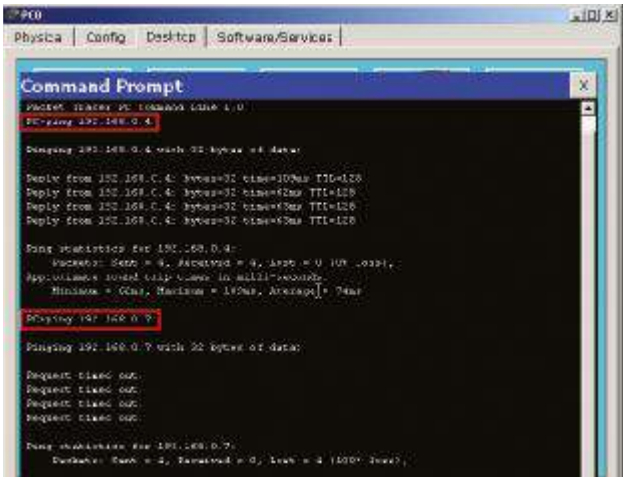
È possibile simulare il ping nel modo seguente: si fa clic su un PC a piacere, per esempio su PC1, si seleziona la scheda Desktop e si fa di nuovo clic sul pulsante [Command Prompt].

Appare una sessione DOS da cui è possibile lanciare il comando ping a un altro computer. ▶

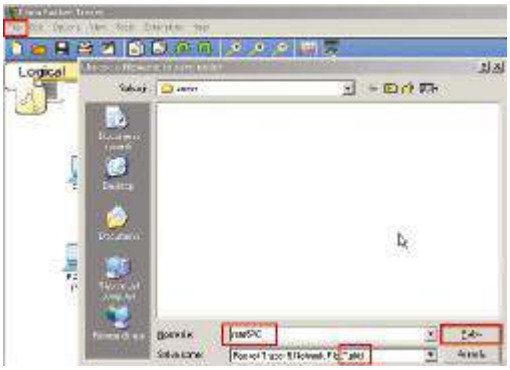
Proviamo ora a "pingare" il PC con indirizzo 192.168.0.4: ▼

Non succede nulla, in quanto siamo in modalità Simulation mentre per eseguire il comando ping dobbiamo portarci in modalità Realtime: proviamo a "pingare" due indirizzi, il 192.168.0.5 che è presente nella nostra rete e il 192.168.0.10, che invece non esiste.

I risultati sono rappresentati nella schermata riprodotta a lato. ▶



Per salvare il nostro progetto possiamo selezionare dal menù File l'opzione [Save] e quindi assegnare il nome, che avrà come suffisso.pkt (figura seguente).



Verifichiamo le competenze

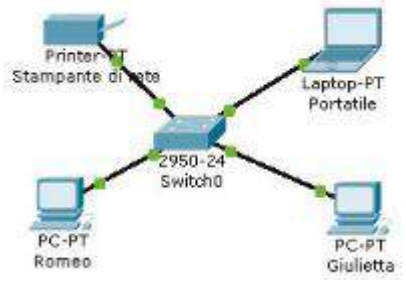
Esprimi la tua creatività

Le soluzioni sono disponibili online all'indirizzo www.hoepliscuola.it nella sezione riservata a questo volume.

Esercizio 1

Crea la rete mostrata nella figura assegnando i seguenti indirizzi IP: ▶

PC Romeo	192.168.0.1
PC Giulietta	192.168.0.2
Laptop	192.168.0.3
Stampante	192.168.0.10



1. Effettua il ping tra il PC Romeo e il PC Giulietta precedente, anche in modalità *Simulation*, dopo aver effettuato il reset della rete.
2. Analizza lo scambio di pacchetti che avviene tra i vari nodi: sulla rete transitano solo pacchetti ICMP?
3. Con lo strumento Inspect (lente di ingrandimento) durante la simulazione analizza la tabella ARP del PC Romeo e del Laptop.
4. Con lo strumento Inspect esegui il monitoraggio della tabella MAC dello Switch0.

Esercizio 2

Crea la rete mostrata nella figura a lato assegnando i seguenti indirizzi IP: ▶

Server	10.0.0.1
Zio Paperino	10.0.0.100
Qui	10.0.0.101
Quo	10.0.0.102
Qua	10.0.0.103

1. Dopo aver effettuato il reset della rete dal PC Quo invia un messaggio al server.
2. Analizza lo scambio di pacchetti che avviene tra i vari nodi: sulla rete transitano solo pacchetti ICMP?
3. Con lo strumento Inspect analizza la tabella ARP di Quo e di Server0 durante la simulazione.
4. Con lo strumento Inspect esegui il monitoraggio della tabella MAC di Switch0.
5. Sempre con lo strumento Inspect analizza i pacchetti che transitano in rete: in modo particolare osservane gli indirizzi mittente e destinazione.

Verifichiamo le competenze

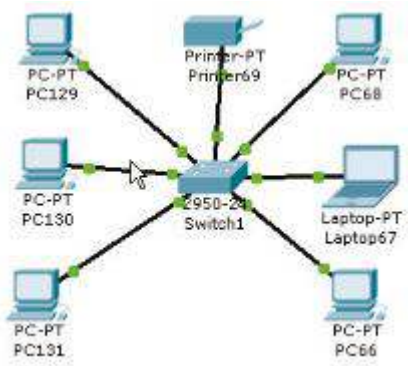
Esprimi la tua creatività

Esercizio 3

Crea la rete, composta da due sottoreti, di cui alla figura a lato, assegnando i seguenti indirizzi IP: ▶

sottorete LAN A
IP 192.168.0.129----131
sub 255.255.255.192
Gateway 192.168.0.65

sottorete LAN B
IP 192.168.0.66----69
sub 255.255.255.192
Gateway 192.168.0.65



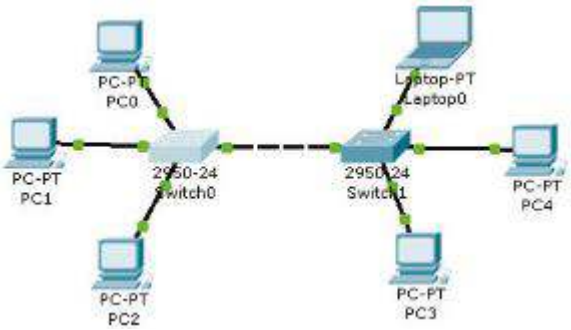
1. Dopo aver effettuato il reset della rete dal PC66 invia un messaggio a PC129.
2. Analizza lo scambio di pacchetti che avviene tra i vari nodi: quali tipi di pacchetti transitano?
3. Con lo strumento Inspect analizza la tabella MAC di Quo e di Switch1 durante la simulazione.
4. Sempre con lo strumento Inspect analizza i pacchetti che transitano in rete: in modo particolare osservane gli indirizzi mittente e destinazione.

Esercizio 4

Crea la rete di cui alla figura a lato, composta da due sottoreti connesse tra loro mediante un collegamento tra gli switch, e assegna i seguenti indirizzi IP: ▶

sottorete A
PC0 PC1 PC2
IP 192.168.0.129----131
sub 255.255.255.192
Gateway 192.168.0.65

sottorete B
PC3 PC4 LAPTOP
IP 192.168.0.66----69
sub 255.255.255.192
Gateway 192.168.0.65



1. Dopo aver effettuato il reset della rete da PC1 invia un messaggio a PC4.
2. Analizza lo scambio di pacchetti che avviene tra i vari nodi.
3. Con lo strumento Inspect analizza la tabella ARP di PC4 e di Switch0 durante la simulazione.
4. Successivamente da PC1 invia un messaggio a PC2.
5. Con lo strumento Inspect esegui il monitoraggio della tabella MAC di Switch0 e di Switch1.
6. Al termine della simulazione con il comando prompt del PC4 visualizza la tabella di ARP.

Verifichiamo le competenze

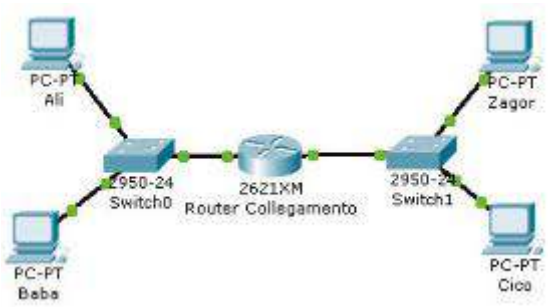
Esprimi la tua creatività

Esercizio 5

Considera la topologia di rete IP (figura a lato) costituita da 1 router, 2 switch e 4 host che sono tra loro organizzati come due reti, una di classe C e l'altra di classe B. ▶

Rete LAN A Classe C
PC 192.168.0.X con X = 1....253
Gateway 192.168.0.254

Rete LAN B Classe B
PC 172.16.0.X con X = 1....65533
Gateway 172.16.255.254



1. Assicurati che le interfacce del router siano ON osservando i "led" ai capi dei link che le collegano ai relativi switch.
2. Imposta su tutti i PC e sui server l'indirizzo corretto del default gateway.
3. Invia un messaggio dal PC Ali della LAN A al PC Cico della LAN B.
4. Analizza lo scambio di pacchetti: che cosa cambia rispetto al comando Ping?
5. Al termine della simulazione con il comando prompt del PC Ali e del router visualizza la tabella di ARP.
6. Accedi al router in remoto dal PC Baba e visualizza mediante il comando show le informazioni principali.

Esercizio 6

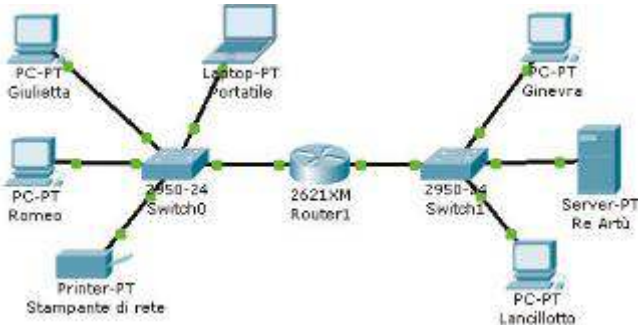
Realizza la topologia di rete mostrata nella figura, assegnando alla seconda rete i seguenti indirizzi:

LAN A

PC Romeo	192.168.0.1
PC Giulietta	192.168.0.2
Laptop	192.168.0.3
Stampante	192.168.0.10

LAN B

Re Artù	11.0.0.1
Lancillotto	11.0.0.100
Ginevra	11.0.0.101



Interconnetti le due reti con un router Cisco 2621XM e assegna loro i seguenti indirizzi IP:
10.0.0.254
11.0.0.254

1. Assicurati che le interfacce del router siano ON osservando i "led" ai capi dei link che le collegano ai relativi switch.
2. Imposta su tutti i PC e sui server l'indirizzo corretto del default gateway.
3. Effettua un Ping dal PC Romeo al server Re Artu in modalità simulazione.
4. Analizza lo scambio di pacchetti: che cosa cambia rispetto all'attività dell'esercizio 4?
5. Al termine della simulazione con il comando prompt del PC Romeo visualizza la tabella di ARP.

e-ISBN 978-88-203-5340-7

www.hoepli.it

Ulrico Hoepli Editore S.p.A.
via Hoepli, 5 - 20121 Milano
e-mail hoepli@hoepli.it