

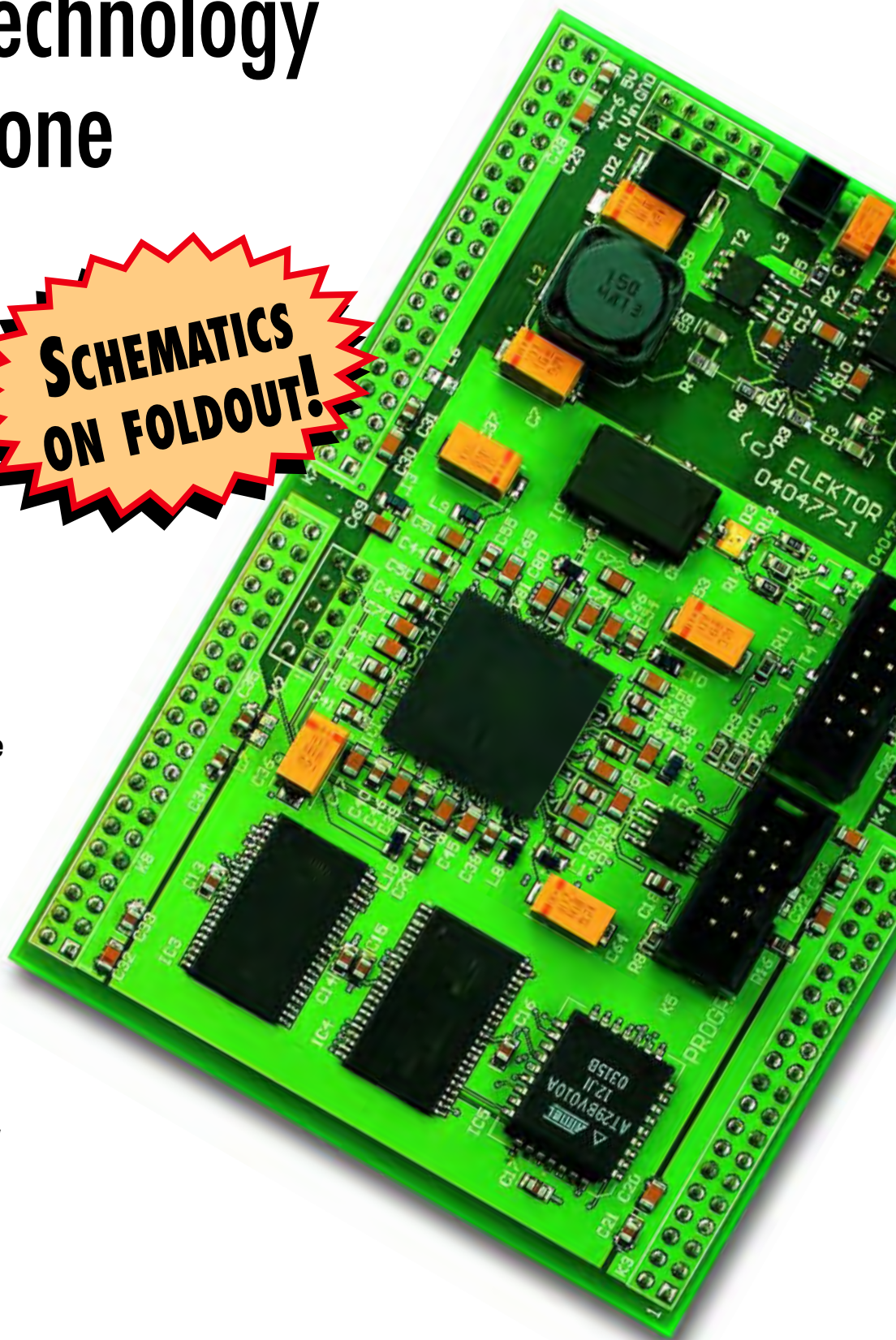
Versatile FPGA M

Modern technology for everyone

Paul Goossens

**SCHEMATICS
ON FOLDOUT!**

FPGAs have established a firm position in the modern electronics designer's toolkit. Until recently, these 'super components' were practically reserved for specialists in high-tech companies. That's all changed now thanks to low component prices and free design software, so it's high time to devote attention to this technology in *Elektor Electronics*. The FPGA module discussed in this article forms the core of various upcoming projects in the magazine.



Module

If you want to use an FPGA in your design, there are several things you can't do without, including a programming interface and configuration memory. We thought it would be a good idea to incorporate the standard items normally associated with an FPGA into a single electronics module, in order to avoid the need for everyone to reinvent the wheel each time. This module can then be used as a 'digital core' for various circuits. A major benefit of this

approach is that it allows designers to concentrate on specific applications without having to worry about the soundness of the FPGA element.

Getting started is the hard part

For those of you who haven't used FPGAs before, this module and the associated prototyping board described elsewhere in this issue form an ideal starting point for learning about FPGAs.

It's impossible to avoid using SMDs in a circuit such as this. To make things even worse, we decided to use an FPGA in a BGA package for this circuit, which means DIY soldering with a normal soldering iron is simply impossible. Soldering the other member of the family (in a PQFP package) is also very difficult. However, using these 'difficult' components allows the overall dimensions of the circuit board to be kept reasonably compact.

Fortunately, you don't have to worry about assembling the board, because the FPGA module is available from *Elektor Electronics* with most of the components pre-assembled. The only components you have to solder by hand are the connectors.

Besides this module, you will need a programming interface that sits between the PC and the FPGA board. Naturally, we've also developed a design for this interface.

Technical features

- **Altera Cyclone FPGA**
- **12,060 logic elements**
- **4 MB configuration memory**
- **8 MB user SRAM**
- **1 MB user flash RAM**
- **on-board 50-MHz clock**
- **JTAG/programming interface**
- **Byteblaster compatible**
- **80 user I/O lines**
- **dedicated clock signals**
- **indicator LED**
- **built-in switch-mode power supply**
- **small multilayer PCB (110 x 77 mm)**
- **supplied ready to use**

It takes more than just an FPGA to make a circuit

As already mentioned, you'll need some peripheral electronics to enable you to use an FPGA in a practical circuit. One of the most important elements beside the FPGA is the configuration memory. In contrast to most FPGAs, the configuration memory retains its data when the supply voltage is switched off.

Each time the unit is switched on, the FPGA must be configured again before it can fulfil its intended function. Fortunately, FPGA manufacturers have developed special memory ICs that can configure FPGA ICs automatically when power is switched on. Our circuit includes one of these memory ICs. Easy programming of the FPGA and the configuration memory during the development phase is also highly desirable. That's why a programming interface (using JTAG) also forms part of the standard peripheral circuitry.

Another quite important part of such a circuit is the power supply. The internal operating voltage of most such ICs is quite low (1.5 V in our case). The power supply must be able to handle short current spikes without problems. In addition, a different voltage is used for the inputs and outputs. Here we decided to use 3.3 V. The supply for this voltage must also be able to deliver fairly heavy currents and remain stable under heavy loads.

Our circuit also includes an oscillator, SRAM and flash memory. They can be used freely by the application.

All the individual elements of the circuit can be easily identified in the schematic drawing.

The design

Due to the size of the schematic drawing of the FPGA module, we decided to print it on a foldout which may be found in the centre of the magazine. Let's start at the beginning with the power supply. It is built around a TPS75003 (**IC2**) and can work with an input voltage in the range of 4.5–6.5 V. This IC is specially designed for use in FPGA circuits. It contains two switch-mode power supply circuits and one series regulator. The latter is not used in our circuit; only the two switch-mode circuits are used.

A step-down regulator that provides a 3.3-V supply voltage is implemented using T1, D1, L1, C1 and associated components. IC2 periodically drives FET T1 into conduction to allow a current to flow from the input voltage via T1, L1 and C1. This causes the current through the inductor to increase, which in turn causes capacitor C1 to be charged. When IC2 switches off the FET, the current flowing in L1 cannot stop immediately, so it continues to flow through C1 and D1. R1, R3 and C3 provide feedback so IC2 can determine whether it has to supply more power or less power.

The values of these components have been chosen such that IC2 tries to maintain the output voltage at 3.3 V. Resistor R2 is a sense resistor. IC2 will limit the current if the voltage across

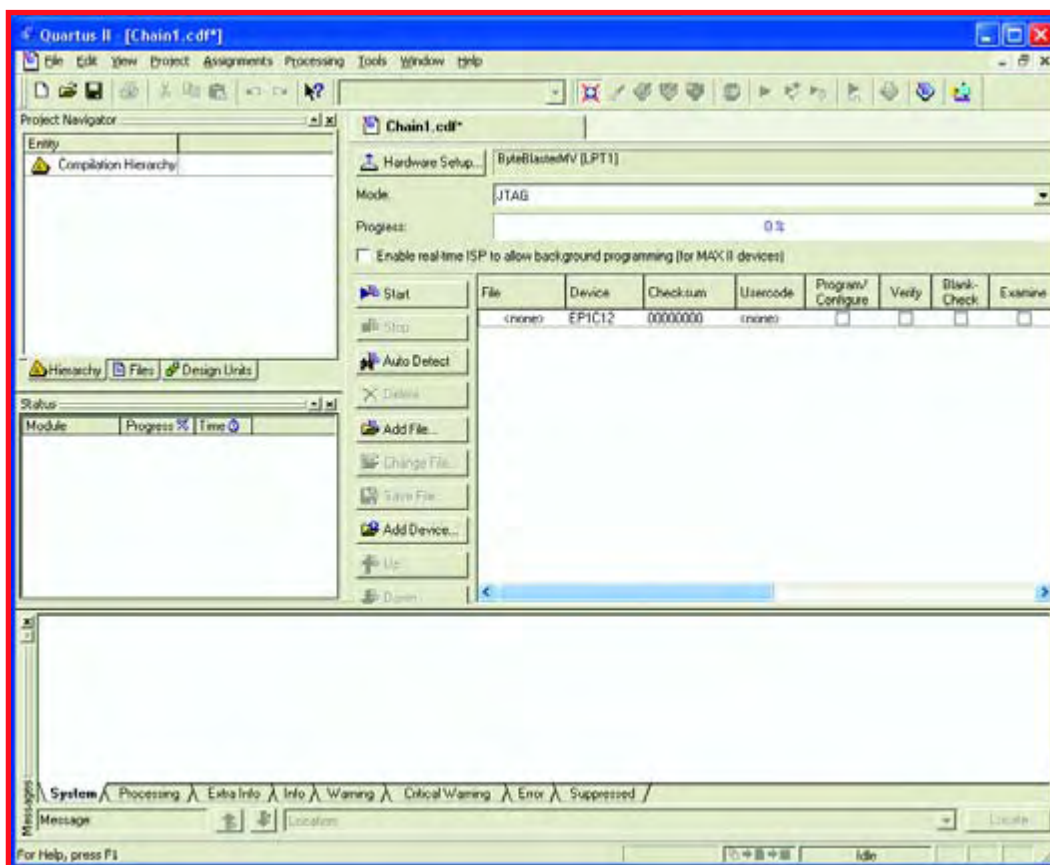


Figure 1. The programming mode of the Quartus program.

this resistor rises above 0.1 V. The value used here (50 mΩ) results in a current limit threshold of 2 A. A similar circuit is built around T2, with only the values of the feedback resistors being different. They are chosen to yield an output voltage of 1.5 V. The electrolytic capacitors at the input buffer the input voltage. They also prevent fast current spikes from leaving the board. That prevents generation of interference that could impair the operation of nearby equipment.

The FPGA is powered from these supply voltages, but not directly. The supply lines for the two sections of the FPGA are first fed through ferrite beads to block high-frequency radiation. The supply voltages are also buffered close to the IC by several additional capacitors. That may appear to be a bit of overkill, but fairly large switching currents at high frequencies can flow through the FPGA, depending on the final circuit configuration. It is always a good idea to keep the paths of such currents as short as possible. Otherwise the circuit can easily generate too much undesirable interference.

And now for the digital part

Now that we've looked after the supply voltages, we can turn to the digital portion of the circuit. The first thing the FPGA needs is a configuration memory. That memory is present in the circuit in the form of IC6, which has been designed to work with the FPGA. When the MSEL signals from the FPGA are at ground level, the IC expects to find a configuration memory (such as the EPCDS4) connected to it. The FPGA will then independently control the memory IC and read data from it in order to configure itself based on the data. That makes the process very easy in actual use.

A programming connector (K5) is provided for programming IC6. The pin assignments of this connector have been chosen to make it compatible with the Altera 'Byteblaster' programming interface.

Connector K6 is used for in-circuit programming of the FPGA from a PC. That makes it unnecessary to first program a design into the configuration memory. This connector can be used to quickly test a design without using the config-

uration memory. Here again, the signals are arranged to be compatible with the Byteblaster interface.

The circuitry around T3–T5 forms an indicator circuit that clearly displays the status of the FPGA using bicolour LED D3. Any error states that may occur are also indicate visibly by D3.

An abundance of I/O

As this circuit is intended to be used for all sorts of applications, it has an abundance of I/O pins. They are accessible via connectors K3, K4, K7, K8 and K2. These connectors fit standard pin headers with 0.1" pin spacing. That avoids

the difficulties of using SMD connectors on the motherboard and thus makes it quite easy to connect the board to your own hardware.

Connectors K3, K4, K7 and K8 present an impressive total of 80 I/O pins of the FPGA to the outside world. The I/O lines are all routed to one side of each connector. The supply voltage (3.3 V) and ground are available on the other side on alternating pins. You can use the supply and ground pins to power your own circuit if it doesn't require too much current. This supply voltage is also filtered on the module by a ferrite bead and two capacitors.

Connector K2 provides a special connection. Besides ground lines, it has several signals that are specifically intended to be used as clock signals. Pin 10 is an input that can be used to supply an external clock signal to the FPGA. The remaining even-numbered pins are all connected to the outputs of the internal PLLs of the FPGA.

Of course, it's possible to provide an external clock signal on any other desired input of the FPGA, but these pins are specifically intended to be used for that purpose.

The extras

Many FPGA-based designs require memory. Although there is memory available in the FPGA, it's not enough for many types of applications. In light of the fact that a 'softcore' processor is often used, a flash memory can be quite handy for storing the firmware of the processor. That makes it possible to use larger amounts of code in application designs without taking up additional space in the FPGA. The flash memory takes the form of **IC5** in the schematic diagram.

A bit of extra RAM can also be desirable in some cases. Besides being useful if one or more softcore processors are used in the FPGA, additional RAM can also be welcome for various types of signal processing.

The RAM memory is provided by **IC3** and **IC4**. These ICs have a capacity of 4 Mbits and are configured as 256K 16-bit words. Signals $\overline{\text{BHE}}$ and $\overline{\text{BLE}}$ can be used as two separate $\overline{\text{CE}}$ lines, with $\overline{\text{BHE}}$ being the chip-enable line for the upper byte (D8–D15) of the data lines and $\overline{\text{BLE}}$ the chip-enable line for the lower byte (D0–D7). That means this memory can also be used as a 512-KB memory with 8-bit data.

IC3 is connected directly to the FPGA, while **IC4** shares its data and address busses with the flash memory (**IC5**). The busses are shared to ensure that enough I/O pins of the FPGA are kept free for the user connectors. However, that means the FPGA cannot read or write the RAM and flash memory at the same time. That doesn't form a major problem in practice, especially when softcore processors are used.

That completes the description of the main parts of the circuit.

We already mentioned that the circuit board for the module is provided nearly fully assembled. The only thing you have to do is to solder the eight provided connectors to the board. Connectors K2, K3, K4, K7 and K8 are fitted on the bottom of the board. The other three connectors are fitted on the top of the board (on the component side). Make sure pin 1 of each connector is in the right position, which is marked by a '1' on the component overlay.

Programming interface

This module by itself is only a starting point. The FPGA and/or configuration memory must be programmed before they can be used. That requires a pro-

gramming interface. The schematic diagram of the programming interface is also shown on the foldout. It connects the FPGA or configuration memory to the parallel port of a PC. The programming interface is compatible with the Altera Byteblaster interface and can be used with free Altera software. The circuit is quite straightforward.

Everything is powered by 3.3 V taken from the FPGA module. The two ICs are used to convert the +3.3-V signals from the FPGA to +5-V signals for the PC and vice versa. The 100- Ω resistors prevent the +5-V signals from raising the supply voltage above the allowable level.

The interface is connected to the parallel port of a PC via connector **K3**. **K2** is a box header that is connected to the FPGA board by a 10-way flat cable with a matching plug at each end.

If you just want to program the FPGA in order to test something, this cable must be connected to **K6** (JTAG) on the FPGA module. If you want to program the module so it can operate on its own, this cable must be connected to **K5** (PROGRAM) on the FPGA module.

Software

A project of this sort naturally includes corresponding software. The manufacturer of the FPGA we use here, Altera, has a very nice package of support software for its FPGAs. The free version of the software is called 'Quartus Web Edition' and can be downloaded from the manufacturer's website (www.altera.com). It's a quite sizeable download (240 MB at the time of writing this article), but you can also request a free CD-ROM on the Altera website.

After installing the software, which by the way goes without a hitch, you will need a licence. You can also obtain this via the Altera website, and it is valid for six months. At the end of the six months you can request a new licence. For the sceptics among our readers, the manufacturer has assured us that this software will remain free in the future.

There's not enough room here to show you everything you can do with this software, and it's equally impossible to present a full course in VHDL here. If you're new to using FPGAs, you should have a look at the accompanying FPGA Prototyping Board described elsewhere in this issue.

Testing

Although this circuit does not have any real input or output capabilities, it's still possible to make a simple test.

After installing the software, it's best to switch off your PC for the next steps. Start by connecting the programming interface to the parallel port of your PC. Next, use the supplied 10-way flat cable to connect the interface to the FPGA module. You should connect the flat cable to connector **K6** (JTAG) on the FPGA module. Now switch on the PC and connect a power source to the FPGA (caution: maximum voltage 6.5 V).

Start up Quartus and then click on the programming icon in the menu bar, or click on the Tools menu and select Programmer (the result will be the same in either case). The programming window will be opened. Click on the Hardware Setup button, and then click on Add Hardware in the new window. Select 'Byteblaster' and specify the printer port you connected to the programming interface (usually LPT1:).

If you then click on the Autodetect button in the main window, the program will automatically detect your FPGA (EP1C12). That way you can be sure the FPGA is working and the supply voltage is present.

(040477-1)

The Versatile FPGA module can be ordered from
Elektor Electronics Readers Services
under order number 040477-91.

Included items

- FPGA module (ready to use)
- programming interface (ready to use)
- programming cable
- interconnect cable (between PC and programming interface)
- 8 pinheaders (supplied separately)

FPGA Prototyping

Paul Goossens

VGA

A genuine VGA output that you can use to display text and imagery on a PC monitor. It's all done with only a few standard components.

See page 24.

Ethernet

Your link to the Internet. The associated IC provides the coupling between the analogue and digital portions.

See page 24.

USB

This USB interface makes communication with your PC fast and easy. And it takes only five components, including the connector!

See page 24.

Analogue I/O

Four analogue inputs and one analogue output. In addition to all the digital artillery on the board, an analogue interface is naturally indispensable. This I/O port has a resolution of 8 bits, which is enough for most applications.

See page 24.

Digital I/O

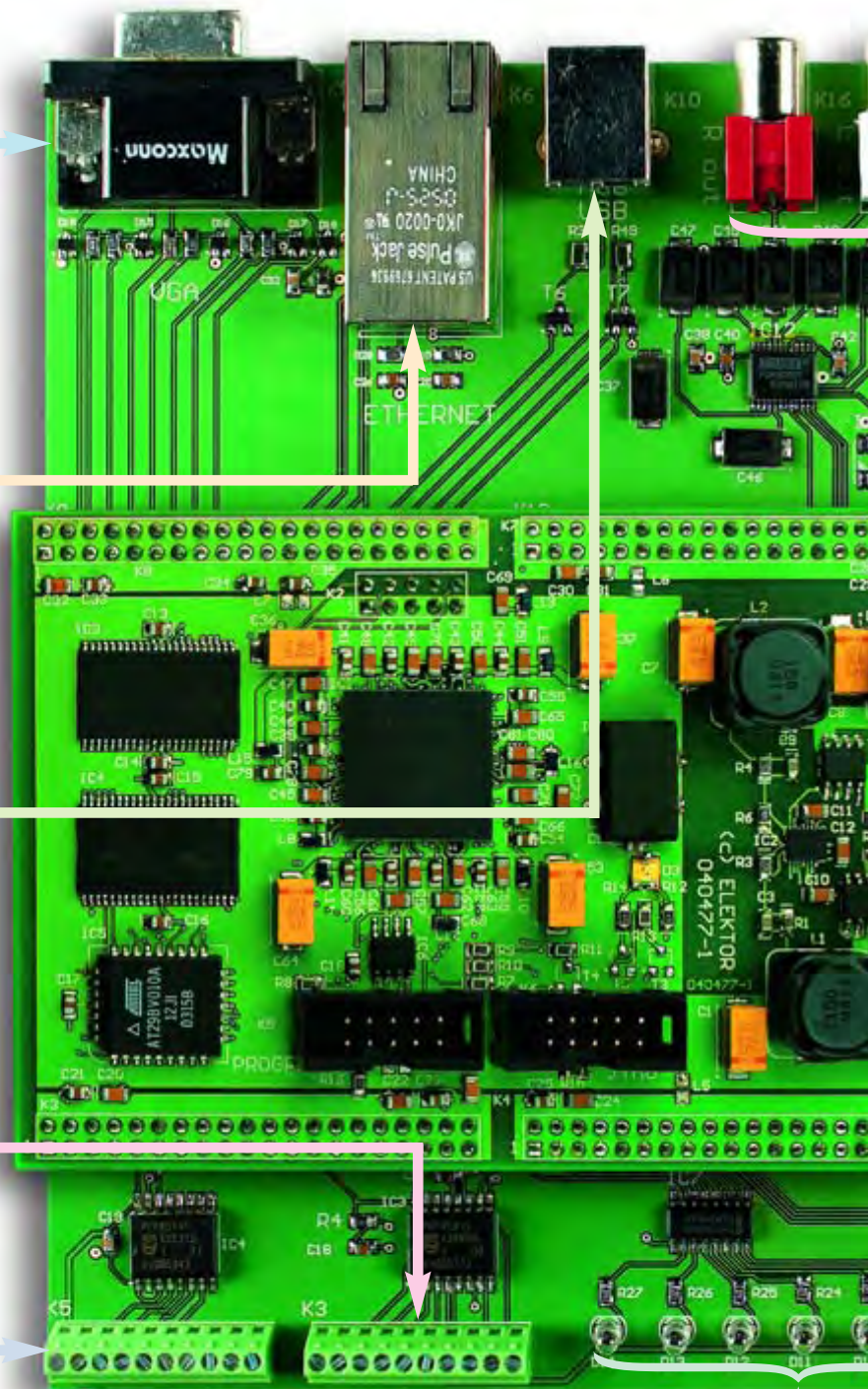
This interface allows you to connect your own circuitry to the prototyping board.

See page 24.

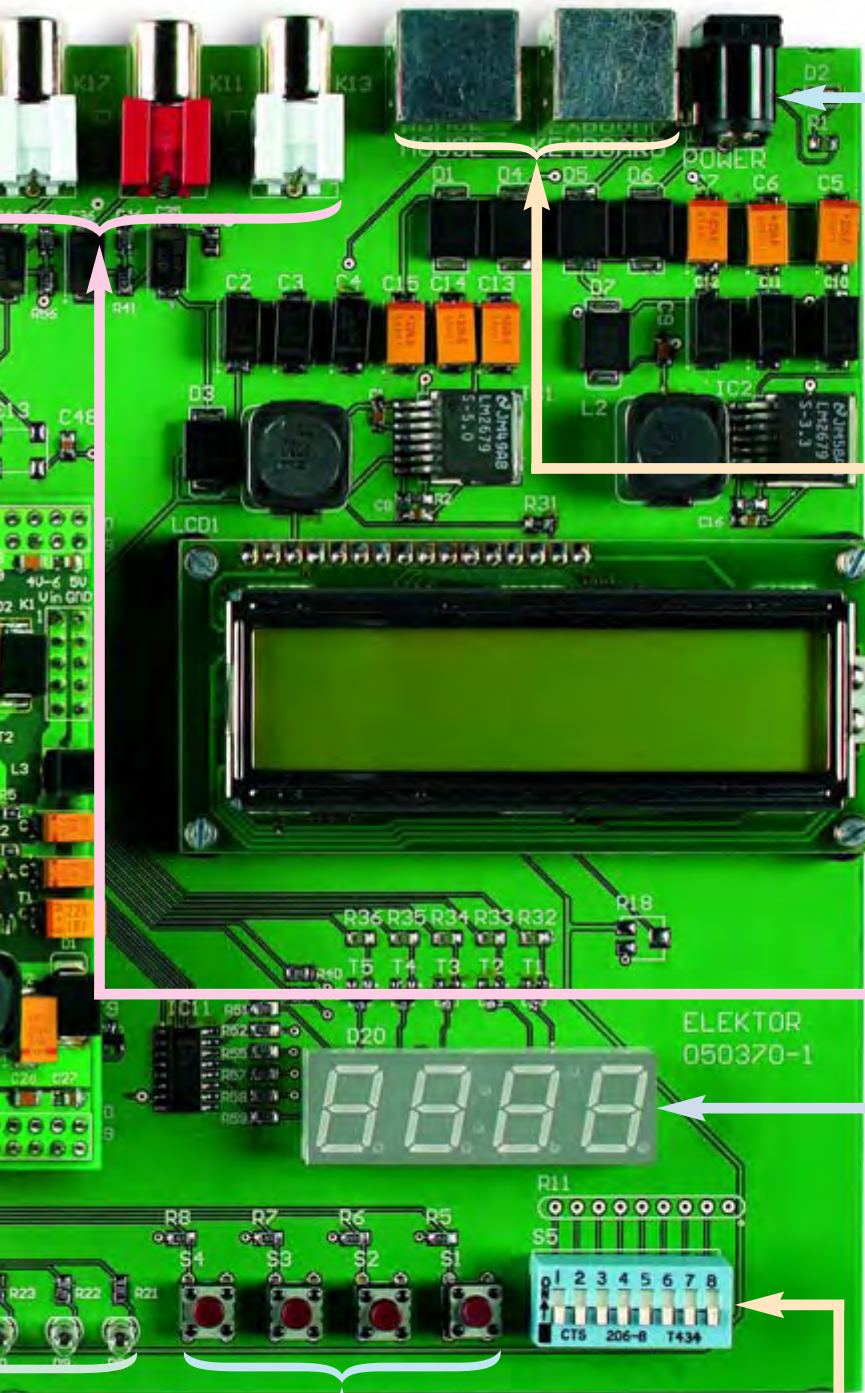
LEDs

You can use the LEDs for a visible indication of the status of various components.

See page 24.



Board FPGA comes to life



Power supply

The power supply is pretty tolerant. As long as the input voltage is somewhere between 6 V and 20 V, the regulator will take care of the rest. You don't even have to worry about getting the polarity right – the power supply circuit doesn't care.

See page 25.

PS/2

A port for a PC keyboard and mouse.

See page 25.

LCD

A two-line LCD module with a maximum of 16 characters per line. An essential part of every proper prototyping board!

See page 24.

Audio I/O

With 16-bit stereo input and output, this prototyping board can also hold its own in the audio world.

See page 25.

Displays

The ideal way to display numbers. Also handy for displaying the date or time.

See page 24.

Pushbuttons

Besides all sorts of sophisticated inputs and outputs, simple operation using pushbuttons is often desirable.

See page 24.

DIP switches

For enabling or disabling options. Naturally, they can also be used as independent switches.

See page 25.

This prototyping board provides an environment for the FPGA module that enables it to do something useful. That gives you an opportunity to discover the capabilities of our FPGA module for yourself. Naturally, you don't have to make this voyage of discovery all on your own – a course in the form of a series of instalments in upcoming issues will guide you on your way.

We've designed a prototyping board that enables the FPGA to be used for a variety of tasks. As you can see from the list of features at the head of the article, it's equipped with quite a few modern interfaces.

For many of our readers, designing a circuit that will ultimately be implemented in an FPGA will be a totally new experience. For that reason, we'll start a series of articles on FPGA programming in next month's issue. It will be based on the prototyping board described here.

Just like the FPGA module, this circuit board is supplied in ready-to-use form.

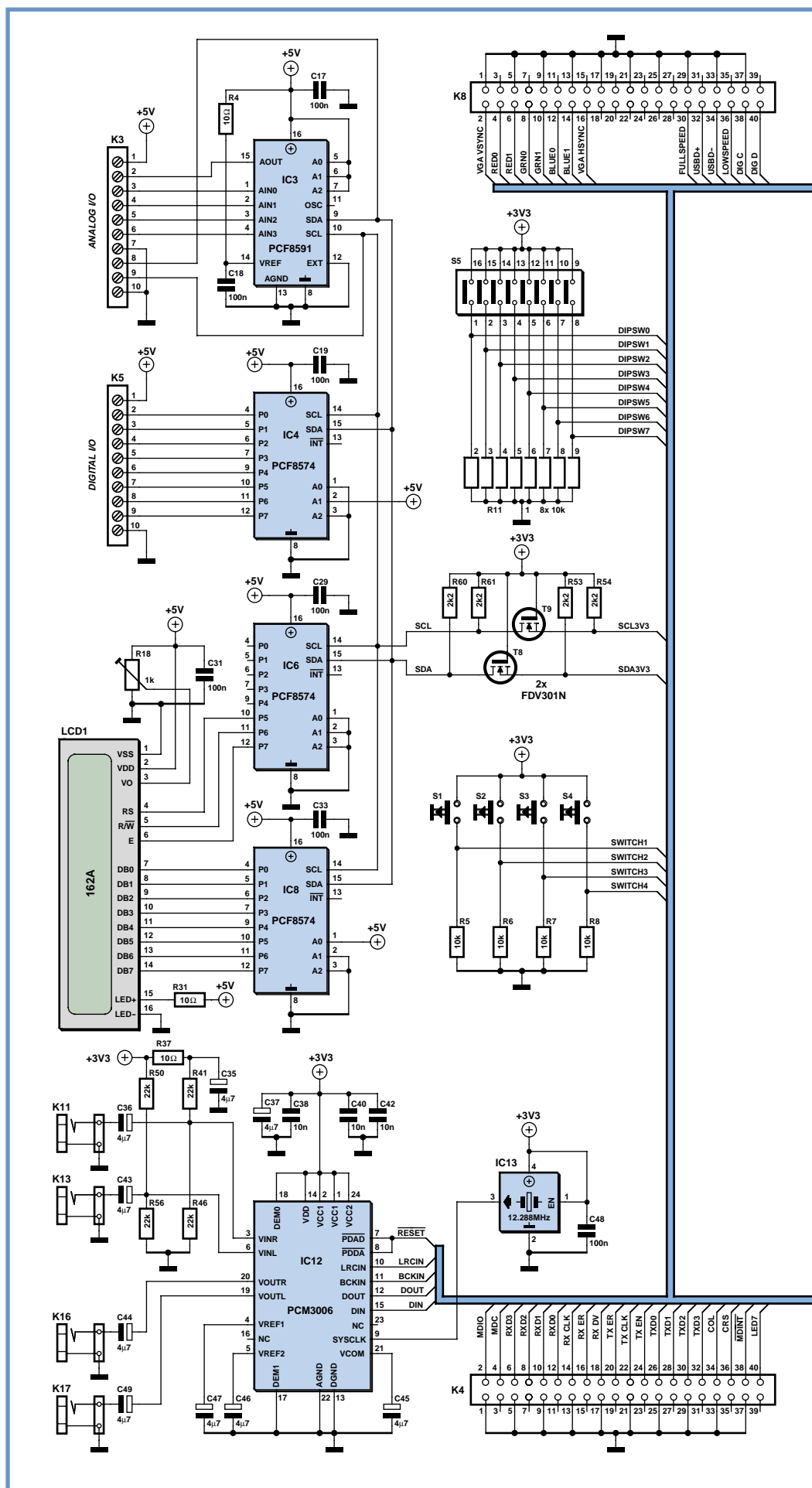
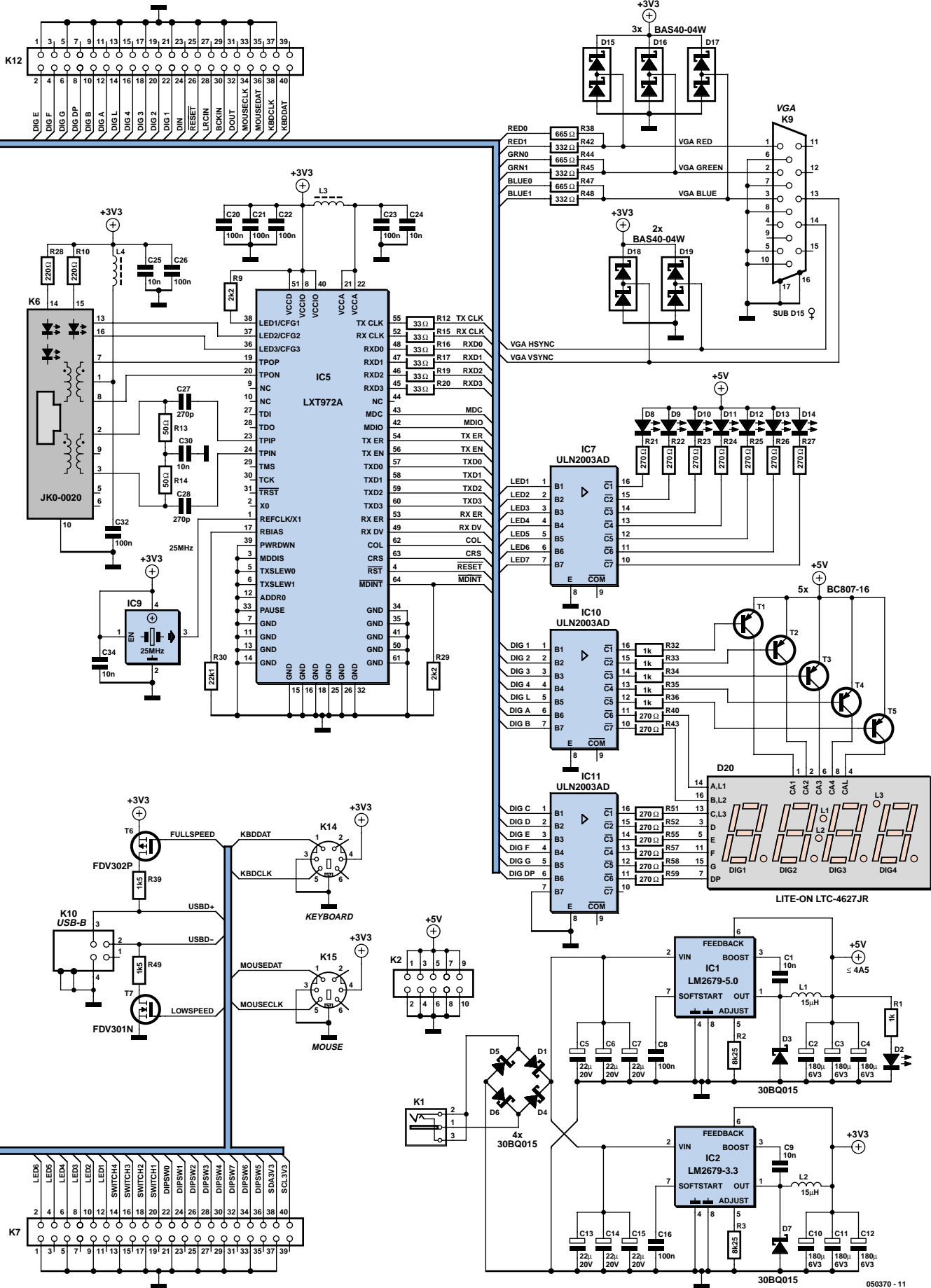


Figure 1. Schematic diagram of the FPGA prototyping board. The board layout is available free of charge in PDF format at www.elektor-electronics.co.uk.



050370 - 11

VGA

Our prototyping board even has its own VGA output. The circuitry for this can be found around K9. Three super-cheap D/A converters are implemented using six resistors. The synchronisation signals can remain in digital form. As signal reflections can be expected here, all of these lines are protected against overshoots by a set of fast Schottky diodes (D15–D19).

Generating a VGA signal is easier than it appears. After reading the **seventh instalment** of the course, you'll know exactly how it's done.

Ethernet

The Ethernet interface is built around IC5, which converts digital signals into analogue signals and the other way around, all according to the Ethernet standard. Driving this interface is fairly complex. In the **ninth instalment**, we'll use this interface to perform some experiments on a network.

7-segment displays

The board has four 7-segment displays with decimal points, which are housed together with three separate LEDs in a single package (D20). This display has several common-anode leads (5 in total). They can be individually connected to the supply voltage to determine which of the five areas of the display should be illuminated. Here four of them are used for the four 7-segment displays and associated decimal points. The fifth lead provides current to the three separate LEDs.

The anode leads can be used to control which elements of these five areas light up.

For the drive signals, we use the same type of IC as for the LED drive signals. Several outputs of IC10 and IC11 are used to switch the anodes to ground under control of the FPGA. Five leads are used to switch transistors T1–T5. In this way, we can use signals DIG1–DIG4 and DIG L to control which part of the display is active.

Driving this display is somewhat difficult because the various parts of the display must be driven in rotation. We'll illustrate how to do this in the **second instalment** of the FPGA course.

Digital I/O

We can easily imagine that the prototyping board will be used as a control unit for something or other. There are eight digital I/O lines to allow user hardware to be connected to the prototyping board.

The IC we selected for this purpose (IC4) has an I²C interface. This interface is also used to communicate with another IC that we'll describe shortly.

The circuit around T8 and T9 has already been used in another *Elektor Electronics* design. It converts the 3.3-V I²C signals into 5-V I²C signals.

The I²C bus uses a serial data transmission protocol. The way this is implemented in the FPGA will be described in the **third instalment** of the course.

USB

A standard USB interface is built round connector K10. It can be optionally configured to present itself as a fast or slow USB device. The PC recognises a fast USB device by the fact that the D+ USB line is pulled up to the positive supply voltage by a resistor. For a slow device, the D- line of the USB is pulled to ground by a resistor.

On the prototyping board, either option can be selected by the FPGA as desired by switching FETs T6 and T7 on or off. Here again, the two data lines are wired directly to the FPGA.

We'll perform a few experiments with the USB bus in the **eighth instalment**.

Analogue I/O

Besides the digital I/O options, the board has four analogue inputs and one analogue output. These 8-bit I/O ports are provided by IC3, which also has an I²C interface. For that reason, this I/O option will also be described in the **third instalment** of the FPGA course.

LEDs

LEDs D8–D14 can be driven directly from the FPGA module by signals LED1–LED7. IC7 buffers the drive signals. Whenever an input is at a logic High level, the corresponding output is pulled to ground. The currents through the LEDs are limited by resistors R12–R27.

This type of drive is very simple. In the **first instalment** of our FPGA course, we will use these LEDs to study various digital functions.

Pushbuttons

One of the input options of the prototyping board takes the form of pushbuttons. This is provided by pushbutton switches S1–S4. Signal lines SWITCH1–SWITCH4 are held low by resistors R5–R8 when the pushbuttons are not pressed. If a pushbutton is pressed, the associated signal line is pulled up to the supply voltage. Signal

lines SWITCH1–SWITCH4 are connected directly to the FPGA module, just like the LED signal lines.

You can use the pushbuttons to control the states of four signal inputs to the FPGA. These four pushbuttons will also be used for simple input functions in the **first instalment** of the FPGA course.

DIP switches

Switch module S5 houses eight small switches, which are wired the same way as pushbuttons S1–S4. They allow an additional eight signal inputs of the FPGA to be controlled by the user.

A good application for switches of this sort is to use them for enabling or disabling various options. We will use these DIP switches in the **second instalment** of the FPGA course.

Audio I/O

The circuitry around IC12 forms a genuine 16-bit stereo input/output port. The sampling frequency is set to 48 kHz by the clock rate of IC13.

This IC communicates with the FPGA via five signal lines, which will be described in more detail in the **fourth instalment** of the course. Naturally, that will also include an example of an application in which the IC is driven by the FPGA.

LCD

The 2-line by 16-character LCD (LCD1 on the schematic diagram) is driven by IC6 and IC8. These ICs operate on the same principle as the ones used for the

digital I/O lines.

We'll put the LCD through its paces in the **fifth instalment**.

PS/2

A PS/2 keyboard and mouse can be connected via connectors K14 and K15. Both signals on this bus are connected directly to the I/O pins of the FPGA module. As with most of the other I/O ports, the FPGA provides 'intelligent' control for these signals. Even the bi-directional data signal can be processed directly by the FPGA without the intervention of any external ICs. This communication uses a special serial transmission protocol, which we'll examine more closely in the **sixth instalment**.

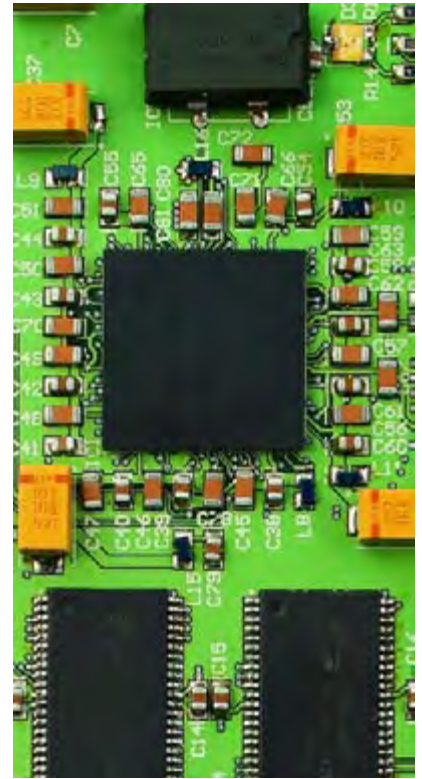
Power supply

All of this is rounded out by the power supply. That's the only part of the circuit that isn't controlled by the FPGA.

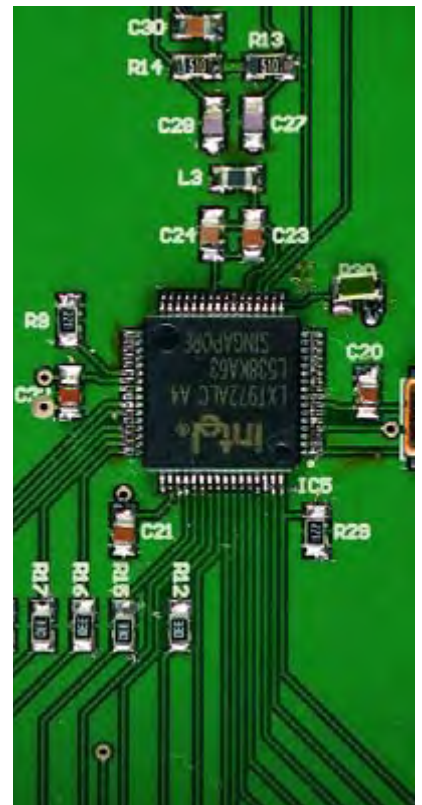
IC1 and IC2 are both step-down controllers. Each of these ICs forms a step-down converter in combination with an inductor, a diode and buffer capacitors. The advantage of these converters is that a high input voltage doesn't cause the supply to dissipate a lot of power in the form of heat, unlike a series-regulated power supply using a 7805 or the like.

The input voltage on K1 can be connected either way round. D1 and D4–D6 sort things out so the polarity of the voltage at the inputs of IC1 and IC2 is always correct. The electrolytic capacitors at the input ensure that fast switching currents stay on the board where they belong. As a result, the power supply generates relatively little interference.

The FPGA is powered via K2.



The "brains" on the Versatile FPGA Unit.



The Ethernet interface built around IC5.

(050370)