DRAWING BOARD

Here's how to make sure your SSAVI descrambler uses the correct sync.

ROBERT GROSSBLATT

No matter how carefully you plan things and how explicitly you set your goals, projects always take on a life of their own. It's very hard to decide officially that a job is finished. The more time you put in, the more difficult it is to stop putting time in. Project goals just seem to grow as the project develops.

There are only a few things to take care of in finishing the design of a basic SSAVI descrambler. We need a detector to let us know when lines 24 and 257 show up because they mark the points where we use our manufactured sync and where we switch to the transmitted sync, respectively. As we discussed last month, this subcircuit is no big deal because it just involves a counter that will keep track of the horizontal line numbers and some decoding circuitry that will signal the arrival of the two lines we're interested in.

You can use any counter you want, and the 4040 we used earlier is as good as any one of them. The tricky part of the design is that the decimal 257 is 100000001 in binary code. That means we need a counter/decoder that can handle nine lines. The 4040 can output the correct count, but the decoder must be able to "watch" nine lines. That's not a problem if you're using discrete logic gates to do the decoding because you can have as many input legs as you want. If you're using an EPROM, it's obvious that the extra available address lines for the input have to be tied to either power or ground.

Somewhere out there in designland, there's a combination of gates that will decode the two numbers (24 and 257), but finding it is pretty tedious and, to make matters worse, there's not a lot to be learned from doing it. I haven't given it a lot of thought, but it can probably be done with a handful of gates. I'll leave the rest of this as an exercise for some of you out there. If someone works it out, drop me a note and I'll pass it along.

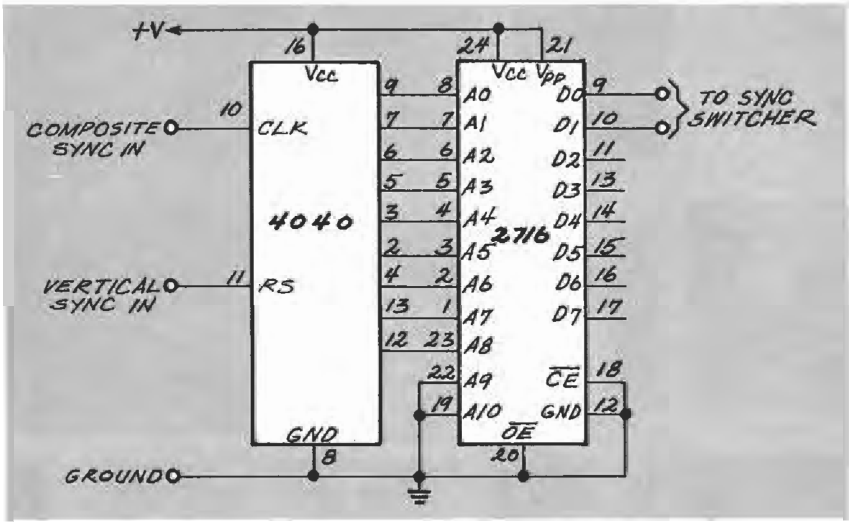

FIG. 1—THE EPROM WILL DECODE two input addresses (24 and 257) with only one or two data lines.

TABLE 1—EPROM CHARACTER GENERATOR CHART

	Input	Programmed Output Data								
Line Number	EPROM Address	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	Hex Data Byte
1	000	0	0	0	0	0	0	0	1	1
2	001	0	0	0	0	0	0	0	1	1
3	002	0	0	0	0	0	0	0	1	1
4	003	0	0	0	0	0	0	0	1	1
5	004	0	0	0	0	0	0	0	1	1
6	005	0	0	0	0	0	0	0	1	1
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
24	017	0	0	0	0	0	0	1	0	2
25	018	0	0	0	0	0	0	1	0	2
26	019	0	0	0	0	0	0	1	0	2
27	01A	0	0	0	0	0	0	1	0	2
28	01B	0	0	0	0	0	0	1	0	2
29	01C	0	0	0	0	0	0	1	0	2
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
257	080	0	0	0	0	0	0	0	1	1
258	081	0	0	0	0	0	0	0	1	1
259	082	0	0	0	0	0	0	0	1	1
260	083	0	0	0	0	0	0	0	1	1
261	084	0	0	0	0	0	0	0	1	1
262	085	0	0	0	0	0	0	0	1	1

This EPROM can also be used to decode line 20 (see text) by programming one of the data lines high when line 20 (address OFh in the EPROM) is reached.

Continued on page 84

DRAWING BOARD

continued from page 18

Don't get me wrong—decoding like this is sometimes necessary, and if you're new to design, it's a good learning experience. But a smarter way to go about this is to use an EPROM, a one-chip solution to the problem.

You can use any EPROM you happen to have around, because we're interested in only two input addresses (24 and 257), and need only one or two data lines (depending on how we design the circuit that enables the manufactured or transmitted sync). The circuit is shown in Fig. 1, and the EPROM's truth table is shown in Table 1. I'm using two data lines to switch between sync sources, but a design could easily be worked out that uses only a single data line.

This is easier to understand when you look at Fig. 2, a block diagram that shows all of the circuitry we've just been talking about. There are two possible sources of horizontal sync pulses: the ones from the original video signal, and the ones being generated by the phase-locked loop circuit. We want to use the transmitted sync during the vertical interval (it's sent in the clear during the vertical interval), and the generated sync for the rest of the time. We have 28 lines of signal with transmit-

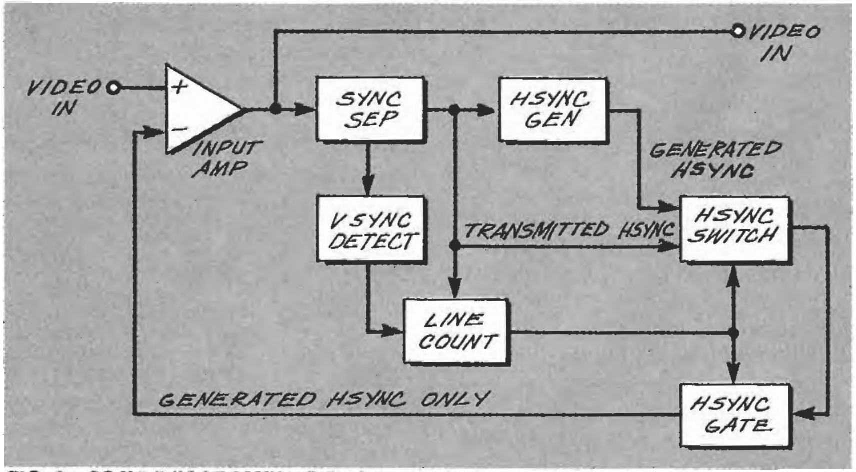

FIG. 2—SSAVI DESCRAMBLER BLOCK DIAGRAM. We use the transmitted sync during the vertical interval and the generated sync for the rest of the time.

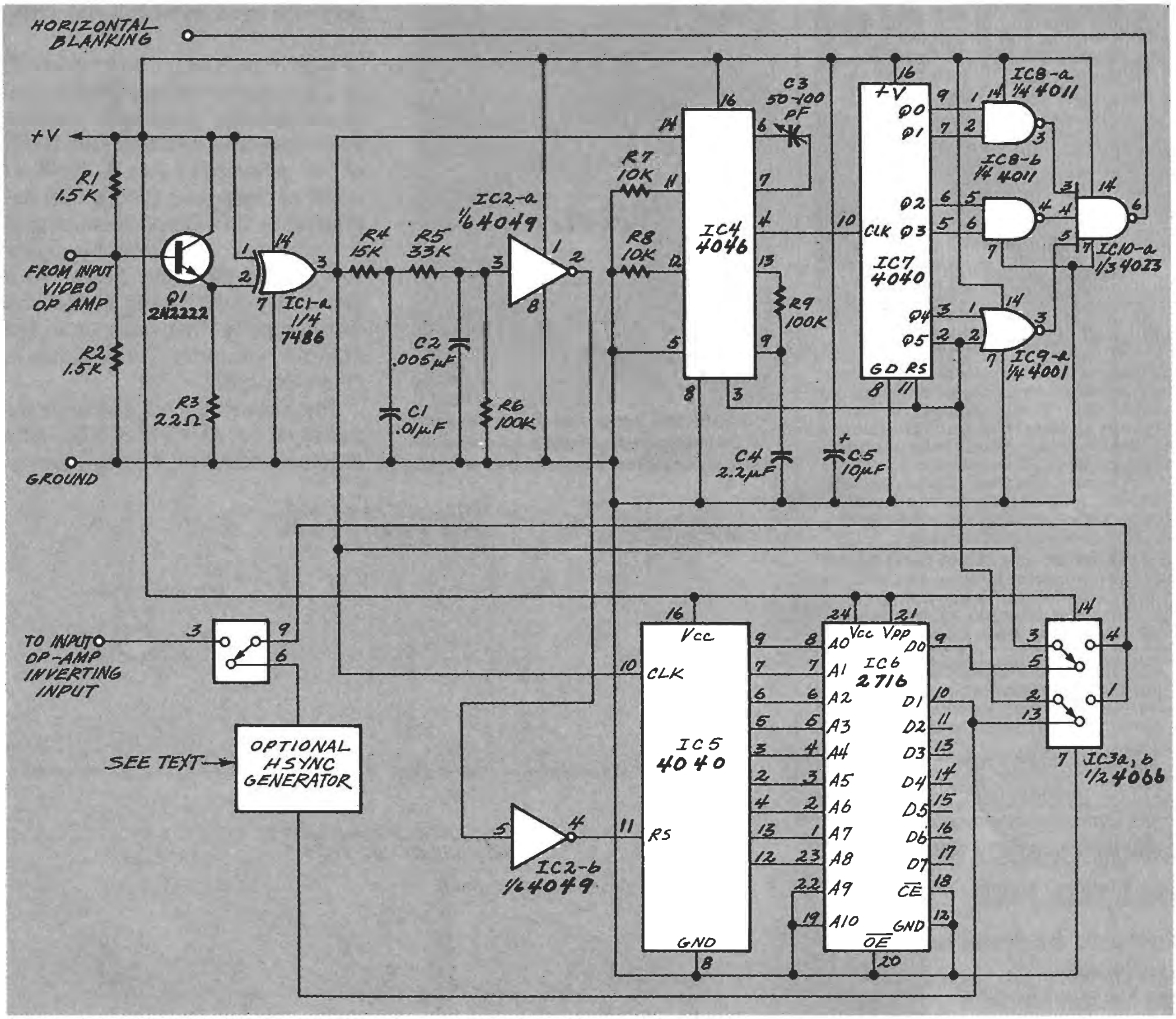

FIG. 3—THE SYNC SWITCHER is really just an electronic single-pole, double-throw switch. Here we've added the EPROM and the 4066 switch. The EPROM's data lines directly control the two 4066 control lines.

ted horizontal sync; they are lines 261 and 262 at the end of one frame, and lines 1–26 at the start of the next frame.

The sync switcher we need is really just an electronic version of a single-pole, double-throw switch, and the easiest way to put one of them together is to use a 4066 just as we did some months ago. Figure 3 shows our circuit so far, with the addition of the EPROM and the 4066 switch. Notice that the EPROM's data lines directly handle the two 4066 control lines. That can be done because the EPROM outputs change state only when the 4040 counter signals the arrival of either line 24 or line 257. If you use gates to decode the counter output, you'll have to find a way to do the same thing. One approach would be to use some of the gates left over last month to build a set/reset flip-

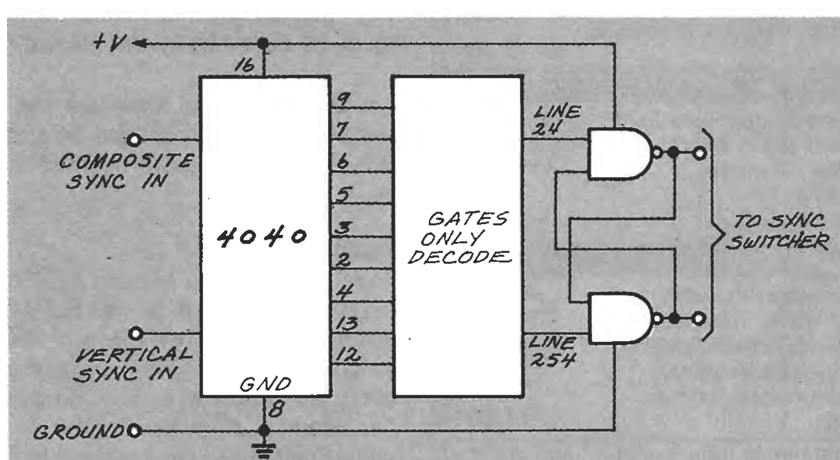

FIG. 4—INSTEAD OF AN EPROM, you can use some of the gates left over from last month's project to build a set/reset flip-flop.

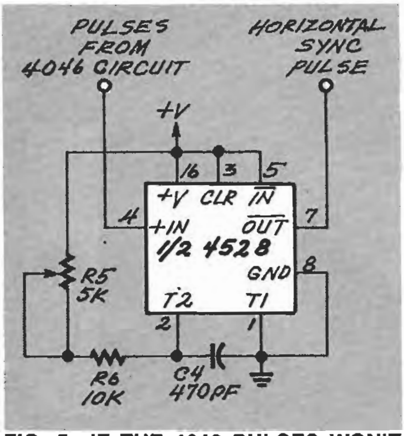

FIG. 5—IF THE 4046 PULSES WON'T TRIGGER the horizontal flyback in your TV, this circuit will generate 4.7-microsecond pulses when triggered by the 4046.

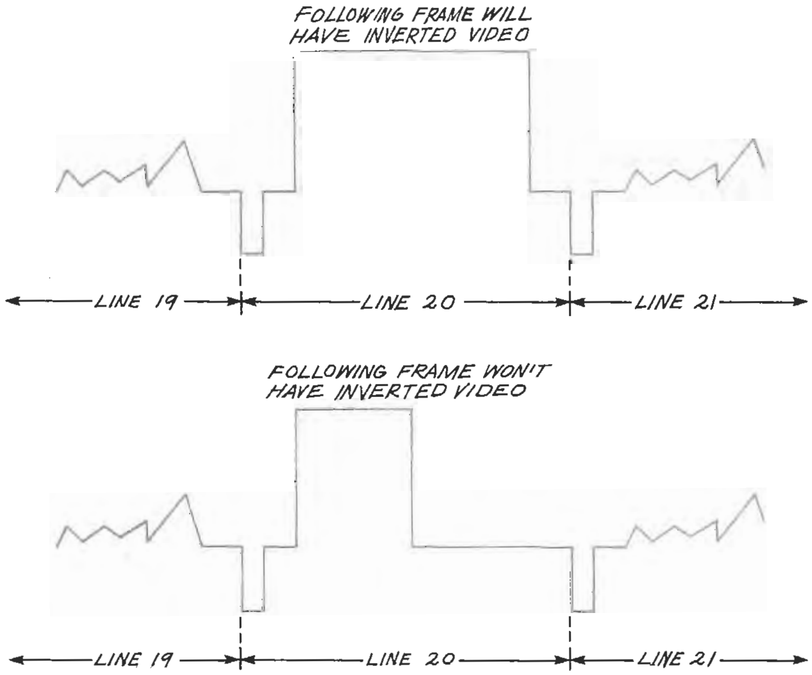

FIG. 6—TO CORRECT INVERTED VIDEO, we need the information buried in line 20.

flop whose control lines are triggered by the arrival of lines 24 and 257. The basic idea is outlined in Fig. 4.

Back to theory

Let's go over the general SSAVI theory for a minute. In the SSAVI system, there are two parts of the video signal that get messed up: the first is the horizontal sync pulse and the second is the polarity of the picture portion of each individual line of video. All the circuitry we've been developing so far has been aimed at taking care of horizontal sync. The circuitry has become a bit complicated, but we now have a way to generate sync even if it has been left out.

The circuitry we've built so far will do a good job of restoring horizontal sync. Just about the only problem you might have relates to the width of the generated pulse. The official width of horizontal sync pulse, according to NTSC specifications, is 4.7 microseconds—and the closer you get to that, the better your chances are of having everything work properly. That leads us to the age-old question, "How close is close enough?"

The answer to that question depends on your TV's horizontal sync detector; some of them will recognize pulses that are as much as 50 percent off, while others will turn up their noses at any deviation beyond 25 percent. If you find that the 4046 pulses aren't triggering the horizontal flyback in your TV, you can use those generated pulses to trigger additional circuitry that will produce a pulse of exactly 4.7 microseconds. Then feed that to the input video op-amp instead of using the 4046 pulses directly to do the job. We designed a circuit to do exactly that, some months ago, when we built a video-sync generator. The circuit is shown in Fig. 5.

Now let's address the problem of inverted video. When we first started this project, we talked about how the SSAVI system encodes information about the polarity of the next frame of video. The original SSAVI system buried this information on line 20, as shown in Fig. 6. Now that we have circuitry to count the lines of video, it's a piece of cake to signal the arrival of line 20 and examine it.

Restoring inverted video is a completely separate deal, and we'll have to wait until next month to get into the nitty gritty of it. Ω

I have to strap batteries to him—my building doesn't allow pets!