# DRAWING BOARD

## Let's get back to our SSAVI descrambler.

ROBERT GROSSBLATT

Before we get into the nitty gritty of the SSAVI system, I have to take a few lines to make a few things clear. I've touched on them before, but it's important to repeat them. All we've gone into so far is gated sync suppression, which can be considered pretty unsophisticated these days.

The SSAVI system isn't the last word in scrambling but it's reasonably recent and it, or some variation of it, will be the scrambling system of choice for some time to come. That is true even if you consider only the economic side of changing scrambling methods.

I have no doubt that many cable companies use this scrambling technique, or at least something very similar. If that's true in your area, it would be a good idea to do some experimenting with the actual signal—hands-on experience is always the best. What you'll probably find out in the real world is that the system used in your area isn't an exact match for the one we'll be taking apart. The general principles might be the same, but the particular details will undoubtedly vary.

When the SSAVI system first started, there were some constants in the video signal that could be used to descramble it. Remember that the picture can be messed up in any one of three ways (see October's column for the details), and the instructions for the descrambler are transmitted somewhere in the vertical interval. The word "somewhere" is a late addition to the SSAVI system. When it first started, the descrambling information was always on the same line. That's where we'll start.

Once upon a time, the sanctity of the vertical interval was closely guarded by the FCC, but as alternatives to standard broadcast TV became more popular (cable, satellite, etc), more and more junk started to show up there.

When the SSAVI system started, lines 0 to 9 were left alone by a request of the FCC, but lines 10 to 13 were where the cable companies transmitted individual subscriber codes. Don't forget that there are unique ID numbers stored in an EPROM (or some other kind of memory) in the cable box. There's also logic circuits there to count the video lines, read the transmitted code, and match it up against the one stored in the box. This is a big thing for the cable companies because it prevents a New York box from being used in California. The scrambling is the same, but the codes are completely different.

The decoder circuitry is also controlled by this coding process because a match between the transmitted bytes and the ones stored in the box will enable or disable the decoder. That is true for both the premium cable services and the pay-per-view events.

That kind of coding might be important to the cable companies, but it doesn't mean anything to us. We can build an experimental descrambler without paying any attention to them.

Since the video can be transmitted with either normal or inverted picture information, one of the tasks that has to be done by the descrambler is to tell the rest of the circuit what has been done to the picture. The place to find that information was originally in line 20, but it has been moved around since the system became popular. As you can see in Fig. 1, the last half of the line will tell you whether the picture is normal or inverted. Remember that

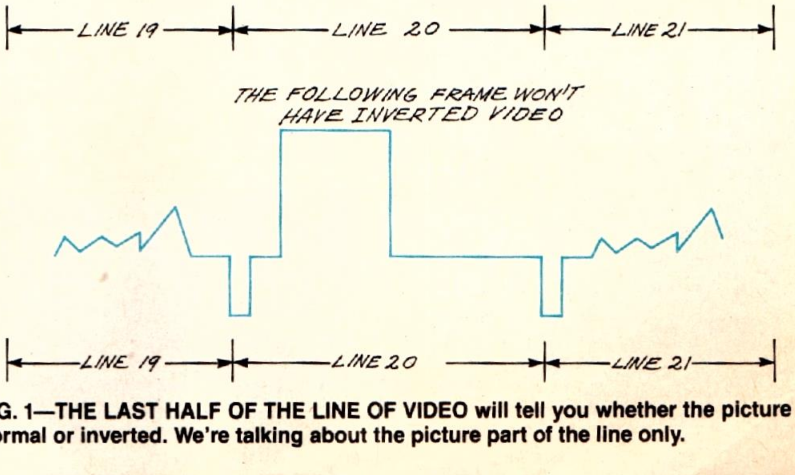

THE FOLLOWING FRAME WILL HAVE INVERTED VIDEO
LINE 19
LINE 20
LINE 21
THE FOLLOWING FRAME WON'T HAVE INVERTED VIDEO
LINE 19
LINE 20
LINE 21

**FIG. 1—THE LAST HALF OF THE LINE OF VIDEO will tell you whether the picture is normal or inverted. We're talking about the picture part of the line only.**

re talking about the picture part of the line only, and not the control section.

If the following field is normal, the last half of the line will be black, and if the picture is normal, the whole line is white. One of the things a decoder needs, therefore, is some way to detect the line and store the data it contains. The stored data is then used as a switch by the circuit to route the video through an inverter if the picture is being transmitted upside down.

This is pretty straightforward stuff. Since we're looking at only one piece of information, all we need is a place to store one bit of information. Your basic piece of cake. The circuitry needed to detect the data, however, is a bit more complex. We need a reference in the signal. So establish a zero point for a line counter, and some counting circuitry to keep track of which line is being received.

You might be wondering what we can count if the signal is being scrambled. But remember, that in the vertical interval (the first 26 lines of video), the signal is being sent in the clear.

Now that we have an approach to handling the possibility of an inverted picture, the last problem to tackle is the one of varying horizontal sync pulses. Sometimes they're there, sometimes they're absent, and sometimes they're not at the proper level. Anything that unstable is a pretty poor choice for a reference signal. So, to avoid a mammoth circuit design problem, the best way to deal with it is to scrap the transmitted horizontal sync (even when it's there), and come up with a way to generate the signal ourselves.

That can also seem to be an insurmountable problem but, just as in the case of the inverted picture, the answer is going to be found in the vertical interval. Once again, remember that the first 26 lines of video are sent in the clear and, even during the rest of the video frame, (no matter what's going on with the picture), the horizontal sync pulse is never inverted. It might be weak or missing entirely, but it's never upside down. That's important to keep in mind because if we generate our own horizontal sync, we don't want an upside down, positive-going sync signal present. If that was the case, the two sync signals would add together and cancel, which is not a good thing.

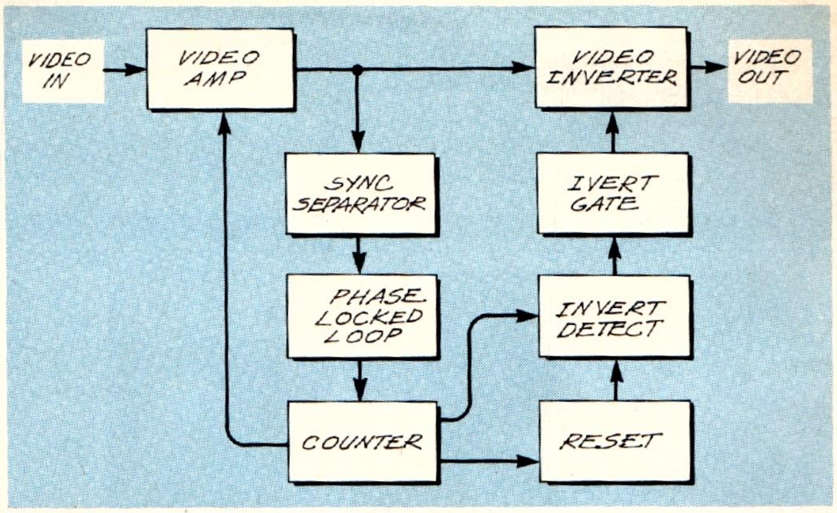


**FIG. 2—THIS CIRCUIT WILL MAKE SURE the picture is always present at the output in a non-inverted state, and that it has horizontal sync pulses present at the right level and the right position.**

We've talked about how to regenerate sync when the signal being received is unreliable. If you don't remember it or haven't read it, go back to October's column and review it. Basically, the approach is to take the horizontal pulses sent in the clear during the vertical interval and use them as the reference for a phase-locked loop that will supply the missing pulses during the rest of the video frame. If you've got twenty or so reliable pulses per frame, you can accurately generate the missing two hundred and forty or so for the rest of the frame.

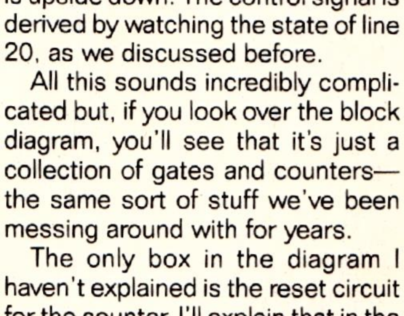
The block diagram of the circuit we need is shown in Fig. 2. In a nutshell, the job of the circuit is to make sure the picture is always present at the output in a non-inverted state, and that it has horizontal sync pulses present at the right level and the right position.

The scrambled video is fed to an op-amp and the output is sent to a sync separator—the same basic circuit that's found in every TV set in the universe. The sync pulses drive a phase-locked loop whose output is decoded to provide the missing sync pulses for the video lines outside the vertical interval (where most of the interesting stuff is found). These generated sync pulses are mixed with the incoming video and then sent, through a gated inverter, to the back of your TV set.

The gated inverter is controlled by a signal that tells it whether or not the picture portion of the video is upside down. The control signal is derived by watching the state of line 20, as we discussed before.

All this sounds incredibly complicated but, if you look over the block diagram, you'll see that it's just a collection of gates and counters—the same sort of stuff we've been messing around with for years.

The only box in the diagram I haven't explained is the reset circuit for the counter. I'll explain that in the next column but you should be able to figure out for yourself exactly what it is. If you get it right, you've got a good handle on the subject of video in general and scrambling in particular. If you can't figure it out, spend the next month getting ready by boning up on the essentials of basic video theory. **R-E**



# DRAWING BOARD

**Our descrambler starts to take shape.**

ROBERT GROSSBLATT

For reasons that I've never understood, even people who manage to build circuits of mind boggling complexity turn green at the gills when it comes to designing circuits that can manipulate baseband video. Maybe it's due to the complexity of the signal or the speed of some of its components (in the single-digit microsecond range), but lots of very talented circuit designers tend to shy away from anything but the most basic video circuitry.

Building a circuit that can make sense out of SSAVI-encoded signals isn't simple, but it's not impossible, either. Best of all, it can teach you a tremendous amount about basic video, too. We've reached the point in this subject where we start turning to hardware. If you look over the block diagram in Fig. 1, you'll see that the circuit we'll need is no more complex than any of the others we've put together over the years.

The final thing we talked about last month was a reset pulse that's needed to initialize the various line counters that will be part of the SSAVI descrambler. We need to find something in the scrambled signal that's stable enough to use as a reset for our digital circuitry.

Remember that everything in the vertical interval is sent "in the clear." One of the components there is vertical sync—an ideal candidate for generating a reset pulse. When you look at scrambled video on a scope or waveform monitor, you may wonder how anything can be picked off the signal. (Incidentally, you stand a much better chance of successfully viewing the scrambled signal if you have a dual-channel scope. Just feed standard video into one channel, use that for the trigger, and view the scrambled stuff on the other channel of the scope.)

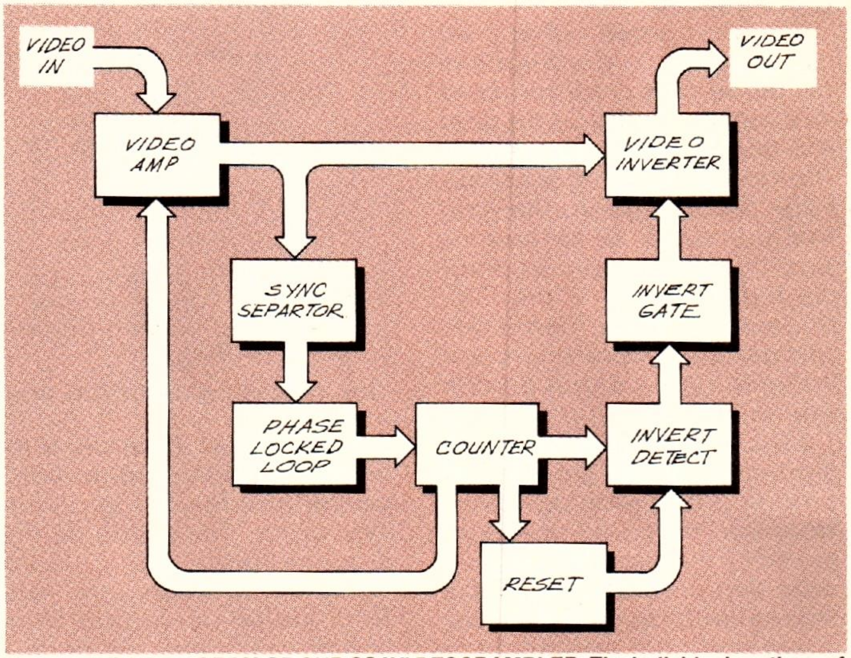


FIG. 1—BLOCK DIAGRAM OF OUR SSAVI DESCRAMBLER. The individual sections of the circuit are no more complex than other circuits we've put together over the years.

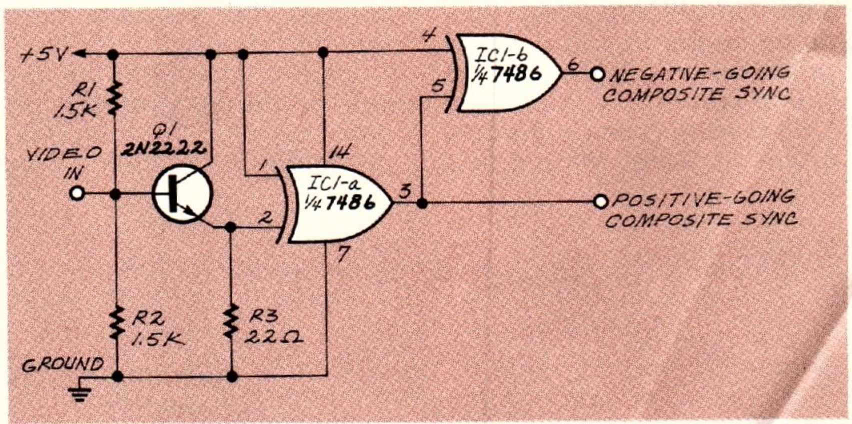


FIG. 2—THIS CIRCUIT WILL TAKE VIDEO in at one end and give you positive- and negative-going composite sync signals at the other end.

Scrambled video may look like a mess, but even broadcast video that's sent in the clear is incredibly jittery. It's a tribute to TV designers and the video standard in general, that the TV set can lock onto anything that comes in over the airwaves.

If you tune your TV to a scrambled signal, you'll note that although the picture is messed up, the screen always shows a full frame. That's because, even though horizontal sync has been altered by the cable company, vertical sync can still be recognized by the circuitry in your TV.

The first piece of hardware we built some months ago was a simple demonstration circuit that enabled you to mess up the horizontal sync signal. I hope you haven't thrown that away just yet because we can use part of it now. The first thing we have to do to the video signal to descramble it is separate the sync from the picture.

The circuit shown in Fig. 2 will take video in at one end and give you two versions of the composite sync part of the signal out the other end: positive- and negative-going. The transistor is working as a simple buffer and, by adjusting the video level at its output, we can have the incoming negative sync fall below the high threshold of the TTL EXCLUSIVE-OR (XOR) gate. The first gate produces the composite sync and the second gate works as a simple inverter.

If this is the first time you've seen this circuit, you can get a full description of it by going back to the November 1992 column where it appeared for the first time. There are other ways to separate sync, but this one has the advantage of giving you an output that swings close to the supply rails, has a very low noise component, and is at TTL logic levels, which makes it much more reliable for feeding the digital circuits we'll be designing for the rest of the descrambler.

While we're looking at the composite sync signal, this is a good time to work out the details of the reset circuit since it has to isolate vertical sync from the composite sync signal. The way to do that should be obvious when you look at Fig. 3, the composite sync waveform. Just as it's supposed to be, vertical sync is the most negative part of composite sync. To isolate the vertical sync, we need a simple low-pass filter; a suitable one is shown in Fig. 4.

The two gates after the filter clean up the sloppy waveform produced by the R-C circuit. You'll notice that CMOS 4049 inverters square up the shoulders of the waveform. The low-pass filter (or vertical integrator, as it's sometimes called) is being fed with a positive-going version of composite sync and, since it's going through two inverters, it's producing a positive-going vertical sync pulse at the output of the circuit.

That's necessary because we a positive-going vertical sync for the rest of the circuit. As with most things electronic, there are several ways to do the same job, but bear with me until we've gone through the whole design before changing

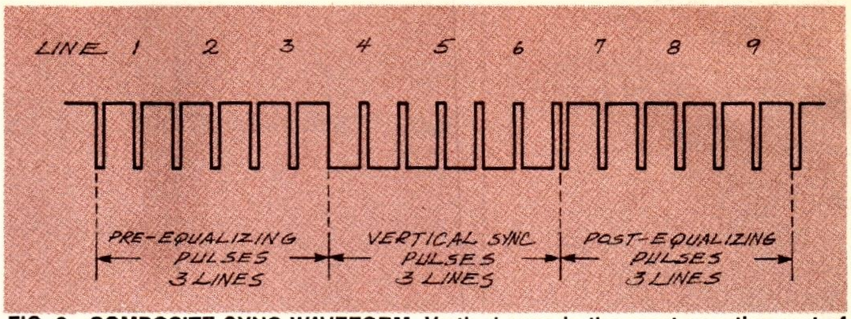

LINE 1 2 3 4 5 6 7 8 9
PRE-EQUALIZING PULSES 3 LINES
VERTICAL SYNC PULSES 3 LINES
POST-EQUALIZING PULSES 3 LINES

**FIG. 3—COMPOSITE SYNC WAVEFORM. Vertical sync is the most negative part of composite sync.**

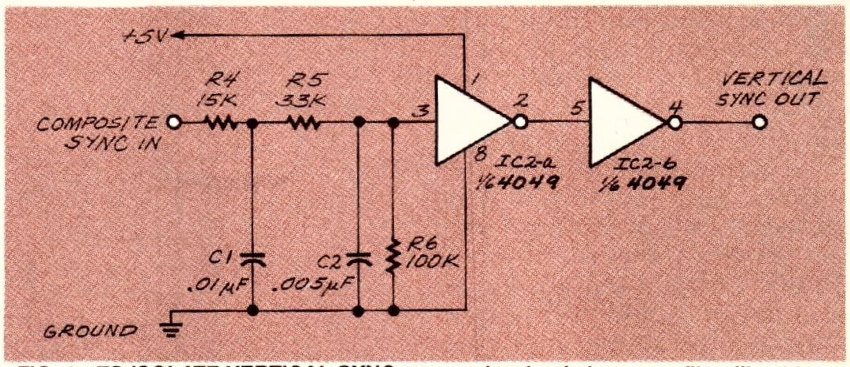


**FIG. 4—TO ISOLATE VERTICAL SYNC, we need a simple low-pass filter like this.**

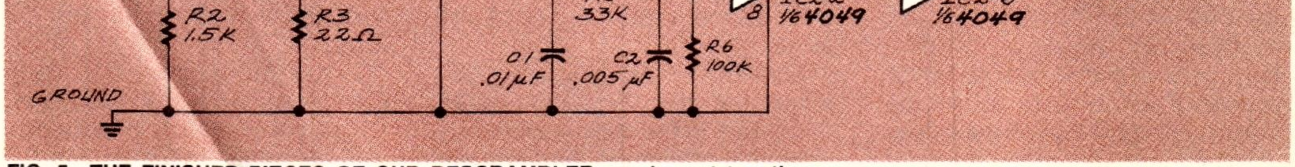


**FIG. 5—THE FINISHED PIECES OF OUR DESCRAMBLER can be put together as shown here. Video goes in at one end and we're able to isolate the sync pulses at the other end.**

things around. Once you understand the circuit in its entirety, you can start modifying it to your heart's content.

Even though we haven't completed the design of the descrambler yet, the pieces we've finished can be put together as shown in Fig. 5 to produce some interesting and extremely informative waveforms. Video goes in at one end and we're able to isolate the sync pulses at the other end. I leave it to you to imagine what a bit of creative gating can do—especially if you use these signals to control the switches in a 4066 as we did some months ago in the demonstration circuit.

Now that we have vertical sync isolated, the next job to do (and the most critical for the descrambler) is to come up with a way of producing horizontal sync. That is obviously more difficult because we know that it won't be present all the time in the received video signal. As a matter of fact, it's a lot better if we operate under the assumption that it's never there at all.

Generating horizontal sync pulses is, in and of itself, a fairly simple business. We know the pulse width and frequency we need for it, but we're missing the starting point. There has to be some reference somewhere in the signal that we can identify so we know how to establish the correct time relationship between our artificially generated horizontal sync pulses and the received video signal.

The answer to this is the same one we've found all along—again don't forget that video is sent in the clear during the vertical interval. If you look there you'll find 26 lines of normal video with normal horizontal sync pulses. That gives us 26 lines of reference signals. Our job is to design a circuit that will continue to produce horizontal sync at the same rate and at the same interval for the rest of the frame.

Because there are 260 or so lines in each frame of video, we'll have a reference available for about ten percent of the time. That's more than enough of a reference for a well-designed phase-locked loop circuit to maintain the correct repetition rate for horizontal sync throughout the entire frame. As we pointed out some months ago, the color-burst signal is present for only 2.5 microseconds, and it serves as a reference for 53 microseconds of picture on each video line. That means the color reference is around for less than four percent of the time. Since we'll have a reference for more than twice that time for horizontal sync, we shouldn't have any problems.

The key to getting good video from a SSAVI-encoded signal is the design of the phase-locked loop that will generate the missing horizontal sync pulses and put them on each video line at exactly the right time. Next month we'll see what has to be done to design that part of the circuit.

There's some math involved and we'll be using a 4046 CMOS phase-locked loop as the heart of the circuit. Get yourself a data sheet on the chip because using it is a bit more involved than the standard gates and counters in the rest of the circuit. **R-E**

[illegible]
to match the color of the knobs.

Scribe a vertical line on the piece of clear plastic to be used as a window over the rotary dial and behind the slot cut in the front panel. With a sharp black felt-tip pen carefully trace over the scribed line so that it will appear as a distinct hair line to serve as the cursor.

Place the painted bezel on the outside of the front panel and the scribed window on the inside (with the cutout window in the front panel between them) and assemble them with screws and nuts. Wind on the nylon drive cord as shown in Fig. 13 and fasten it to the spring.

**Power supply**

The prototype SWX6 receiver was powered by a regulated 12-volt power supply. The receiver draws about 75 milliamperes, so any source of 12 volts, including batteries, at that current level can be used. Provision must be made, however to obtain the 5 volts DC to power the variable frequency oscillator.

**Enclosure finishing**

Assemble the three principal circuit boards to the base plate of the enclosure with No.4-40 machine screws and two No. 4-40 nuts used as spacers in the positions shown in the photograph Fig.5.

Complete all wiring. Make the connection from the antenna jack J1 to the wiper of switch S1-a with RG-174/U coaxial cable, and make all connections from the six terminals of switch S1-a to the six filters (detailed in Table 2) with enamel-coated magnet wire. Then make all connections from the output side of the filters to the contacts of S1-b with RG-174/U coaxial cable. Also, connect the S1-b wiper contacts to capacitor C15 with RG-174/U coaxial cable.

Make the following connections with insulated wire:
- Five volts to the wiper of S1-c
- Six terminals of S1-c to each of the oscillators shown in Fig. 4
- The six output connections from the oscillators S1-d

Make the following connections with RG-174/U coaxial cable:
- Wiper of switch S1-d to C14
- Electrolytic capacitor C41 to speaker/headphone jack J2

**Enclosure assembly**

Mount the front and back panels on the base panel with angles, nuts and bolts. The side panels should be assembled with angles set so they are concealed behind the left and right edges of the front and back panels, under the left and right edges of the top panel, and over the side edges of the baseplate.

In the prototype, dry-transfer labels were applied to identify the manual controls. A separate circular plastic band switch plate was used to identify the band positions on the multi-deck switch. (This plate can be easily changed if you want to change the band positions or frequencies.) Clear lacquer was applied over the dry labels to protect them. Install four rubber feet on the bottom of the base plate with sheet metal screws to prevent the exposed screw heads from scratching the table on which the receiver is located. R-E