Video Accessory

Color Processor/ Noise Filter

ROGER COTA and LLOYD ADDINGTON

Here's an inexpensive add-on device that you can use to create professional fades, reduce noise, and correct color errors in video recordings.

IF YOU OWN A VIDEOCASSETTE RECORDER (VCR) then you are no doubt familiar with the problems of making high-quality recordings. No matter how hard you try, you end up with washed out colors, glaring reds, and other assorted problems. It gets even worse when you record from one tape to another. But those problems can be remedied easily and inexpensively with the video color processor we'll describe here. You'll find that tapes with their color properly restored, their video noise reduced, and with the look of professional editing will be much more enjoyable to view than those you're probably looking at now. And not only that—you can actually save money using the processor by using less tape. That's because you can use slower recording speeds and still obtain acceptable results.

While the color processor can be used for several applications, it's perhaps most useful when recording from one tape to another. You can correct some of the first tape's color distortion and insert fades-to-blacks, where appropriate, to get what looks like a professionally edited tape. That's especially helpful for making home movies.

You can also use the processor when recording off-the-air programs. But the processor does not contain its own RF modulator and it has no audio input. Therefore you'll need an enhancer, a second VCR, an RF modulator with an audio input, or a monitor with a composite-video input so that you can see the results of the processing.

Features

This color processor is designed to let you correct color and contrast errors, create professional quality fade-ins and fade-outs, and eliminate unwanted colors and video noise. You can do those things by using the front-panel TINT, FADER, LEVEL, and BACKGROUND controls—they let you manipulate the picture color and brightness in ways that you cannot with a TV set or monitor alone.

Proper restoration of skin tones and balance of color hues is the function of the TINT control. The FADER control allows adjustment of the picture level from black (0%) to full luminance (100%), and can be used in editing for fading to black or fading from black to full luminance. Color saturation of the picture can be reduced to black-and-white or increased to full color by adjusting the LEVEL control. Unwanted noise in the picture background can be virtually eliminated or background colors can be intensified by adjusting the BACKGROUND control.

Another control, FLASH FILTER, is provided so you can eliminate the problem of single-color dominance that is often present in bright pictures. A PROCESS/BYPASS switch is included so that you can bypass the processor while still leaving it in-line. Finally, a graduated PICTURE LEVEL LED meter lets you monitor the output level.

The processor is powered by 12 volts DC. We used a wall transformer with a 12-

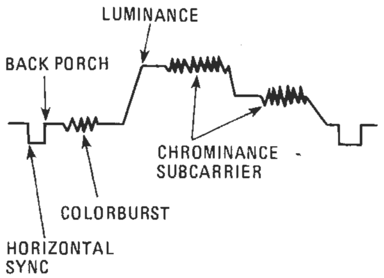


FIG. 1—A COMPOSITE COLOR-VIDEO SIGNAL contains luminance and chrominance information as well as horizontal sync pulses.

volt DC, 300-milliamp output to convert the line voltage to that level. If you want to use the processor for portable applications, such as when using a video camera, you can power it with a 12-volt battery pack.

About the circuit

Before we can talk about the circuit of the color processor, we have to introduce you to a composite video signal, as shown in Fig. 1. As you can see, it is composed of two unrelated signals: *luminance* and *chrominance*. Those signals are processed independently in the color processor and are mixed together again before they are output.

The luminance (or black-and-white) signal contains the picture (or brightness) information. The chrominance (color) signal contains the hue and saturation information. It is used to modulate a subcarrier frequency of 3.58 MHz. That subcarrier is then used to amplitude modulate the video carrier.

Also contained in the chrominance signal is a *colorburst* that is sent to synchronize the receiver's color circuits with the color that is being transmitted. The colorburst is sent during the horizontal blanking interval (when the electron beam snaps back across the screen). It can be thought of as a reference for interpretation of the color information.

We should take a closer look at the information contained in the chrominance signal: the hue and saturation. The hue is commonly called color, while saturation has to do with how pure a given hue is (how much white it contains). The hue that your set displays is determined by the phase relationship between the color subcarrier and the colorburst. The color processor's TINT control lets you adjust that phase and thus the hue. The saturation is determined by the amplitude of the chrominance carrier. That can be adjusted by the LEVEL control.

A block diagram of the color processor is shown in Fig. 2 and its schematic is shown in Fig. 3. Video input enters the chrominance/luminance separator and is broken up into a luminance-and-sync signal and a chroma-and-burst signal. Let's

PARTS LIST

All resistors ¼ watt, 5% unless otherwise specified.
R1, R25, R55, R89—100 ohms
R2, R39—75 ohms B
R3, R9, R32, R71, R75—150 ohms
R4—27,000 ohms
R5, R11, R36, R48, R101—22,000 ohms
R6, R12, R13, R20, R47, R63, R65, R73, R87, R96—1000 ohms
R7, R16, R17, R40, R41, R58, R59, R67, R68, R72, R74, R76, R77, R85, R99, R103, R106—470 ohms
R8, R26, R84, R91—1500 ohms
R10, R19, R54, R79, R80, R97, R98, R104—10,000 ohms
R14, R37, R38, R42, R78—220 ohms
R15, R21, R29, R34, R53, R64, R95, R100—2200 ohms
R18, R94—680 ohms
R22—6800 ohms
R23, R27, R93—4700 ohms
R24, R88—15,000 ohms
R28, R56, R57, R66, R83, R92, R102—3300 ohms
R30, R35, R105—22 ohms
R31, R82, R86—2,000 ohms, potentiometer, linear taper
R33, R49, R90—47,000 ohms
R43, R62, R107—330 ohms
R44—56 ohms
R45—1 megohm
R46—5600 ohms
R50—750 ohms
R51, R52—33,000 ohms
R60—560 ohms
R61—390 ohms
R69—20,000 ohms
R70—12,000 ohms
R81—10,000 ohms, potentiometer, linear taper

Capacitors
C1, C2, C6, C11—22 μF, 10 volts, electrolytic
C3, C40—470 μF, 10 volts, electrolytic
C4—5-55 pF trimmer capacitor
C5, C10, C17—22 pF, 50 volts, ceramic disc
C7, C18, C41, C42—0.1 μF, 50 volts, ceramic disc
C8—68 pF, 50 volts, ceramic disc
C9—100 pF, 50 volts, ceramic disc
C12, C20 through C24, C26, C29, C30, C32, C35 through C38—.05 μF, 50 volts, ceramic disc
C13—680 pF, 50 volts, ceramic disc
C14—.01 μF, 50 volts, mylar
C15, C31—130 pF, 50 volts, ceramic disc
C16—.001 μF, 50 volts, ceramic disc
C19—10 pF, 50 volts, ceramic disc
C25—56 pF, 50 volts, ceramic disc
C27, C28—180 pF, 50 volts, ceramic disc
C33—220 pF, 50 volts, ceramic disc
C34, C43—100 μF, 10 volts, electrolytic
C39—470 μF, 35 volts, electrolytic

Semiconductors
IC1—7808 8-volt positive regulator
Q1 - Q5, Q8, Q9, Q11, Q12, Q16 - Q19, Q21 - Q23, Q25, Q27, Q28, Q32, Q33, Q35 - Q39—2N3904
Q6, Q7, Q10, Q13 - Q15, Q20, Q24, Q26, Q29 - Q31, Q34—2N3906
D1 - D9, D11—1N914 or 1N4148 diode
D10—1N4004
LED1 - LED4—red LED's
J1, J2—RCA-type phono jack
J3—miniature phone jack
L1—33 μH, high-Q inductor
S1, S2, S3—SPDT toggle switch

Miscellaneous:—
117-volts AC to 12-volts DC, 300-mA wall transformer; printed circuit board.

The following are available from Video Control, 3314 H Street, Vancouver, WA 98663 (1-206-693-3834): Complete kit including PC board, wall transformer, case, and all parts for $139.00. Drilled and plated circuit board $28.50. Power adapter (wall transformer) $10.00. Please add $3.50 for postage and handling; WA residents add state and local taxes as applicable. Allow four (4) weeks for delivery.

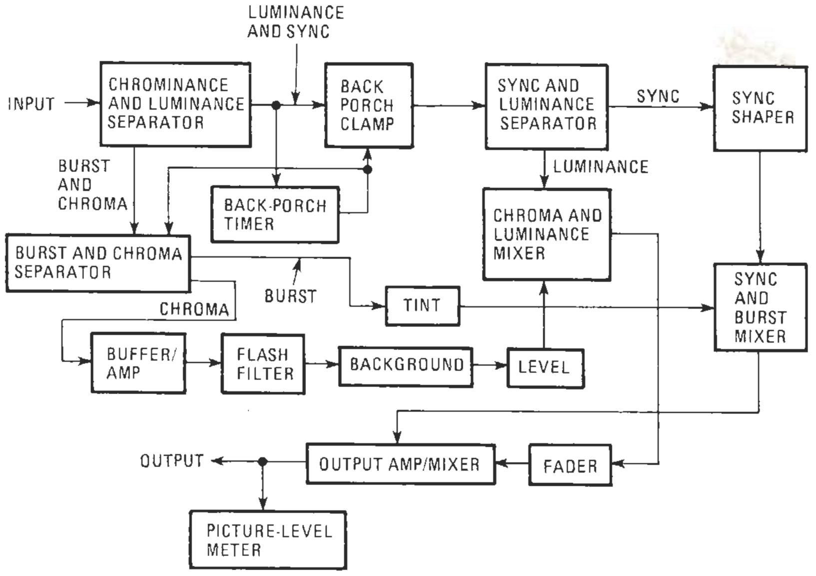


FIG. 2—BLOCK DIAGRAM OF THE COLOR PROCESSOR shows how the luminance and chrominance are processed separately.

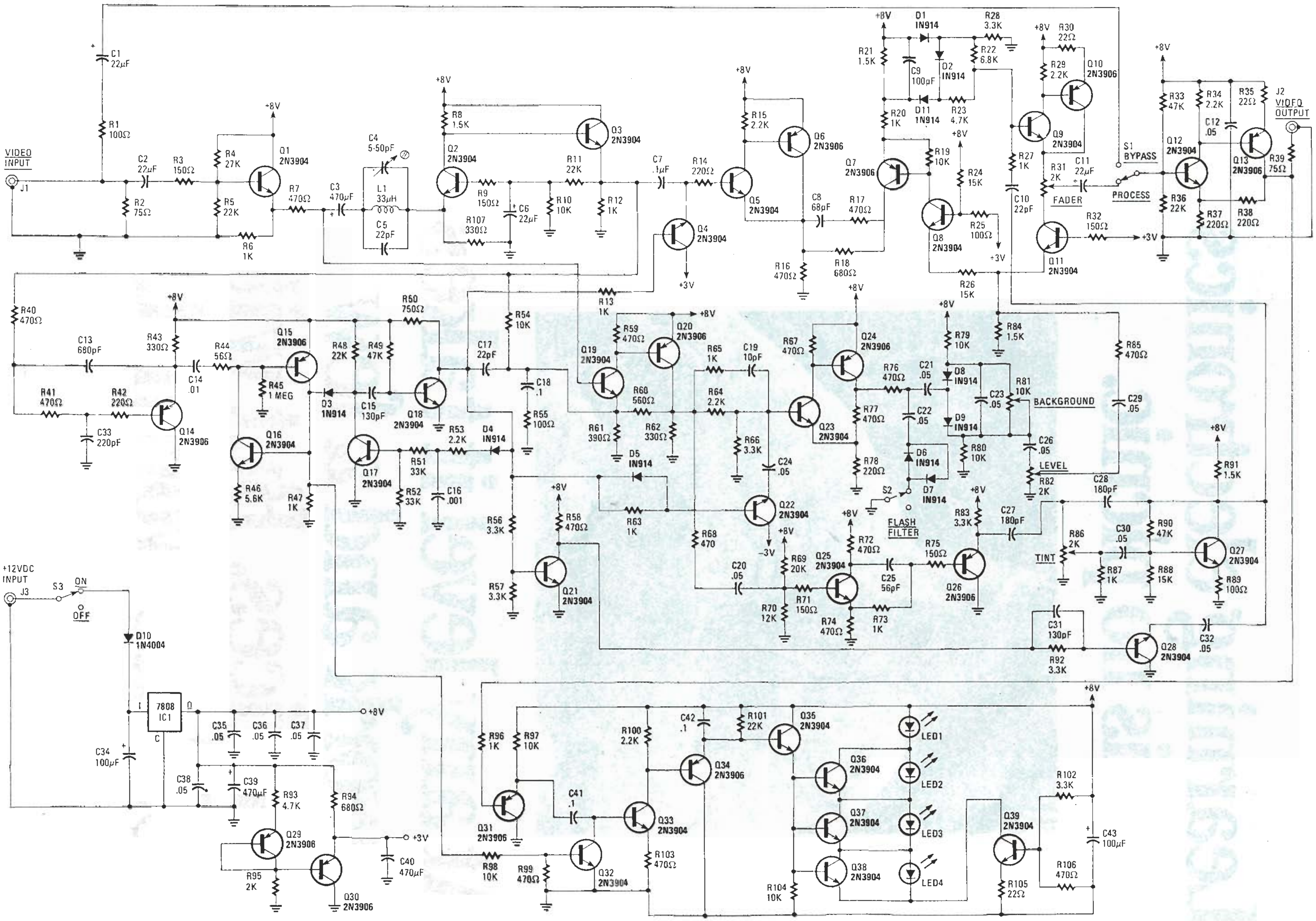


FIG. 3—SCHEMATIC OF THE COLOR PROCESSOR. You can substitute 1N4148 diodes for those marked 1N914.

first see what happens to the luminance-and-sync signal.

The luminance-and-sync signal goes to both the back-porch timer and the back-porch clamp. (The back-porch timer is a circuit that detects the sync pulse. It tells both the back-porch clamp and the burst-and-chroma separator when to begin their action.) From the back-porch clamp, the signal enters the sync-and-luminance separator. The luminance signal is sent to be mixed with the processed chroma signal in the chroma-and-luminance mixer. The sync pulses are properly restored in the sync shaper and are sent to be mixed with the processed burst signal in the sync-and-burst mixer.

Now we'll look at how the burst and chroma signals are processed. After the burst-and-chroma signal is separated from the luminance-and-sync signal, they are sent to the burst-and-chroma separator (Q22 and Q28). That separator acts like a switch that is controlled by the back-porch timer. The chroma subcarrier is sent through the buffer amplifier and then passes through the flash-filter, background-, and level-control sections. The processed chroma signal is then mixed with the luminance.

The burst passes through the tint-control section (phase shifter) and is mixed with the restored sync pulses in the sync-and-burst mixer.

The output signal from the chroma-and-luminance mixer passes through the fader and proceeds into the output amplifier/mixer, where it is mixed with the signal from the sync-and-burst mixer. The LED PICTURE LEVEL meter monitors the processed video output from the output amplifier/mixer.

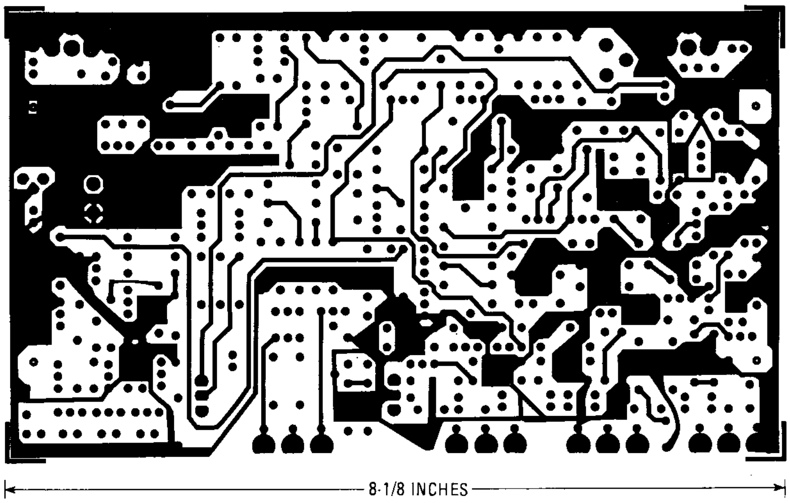


FIG. 4—The component side of the processor board is shown half size.

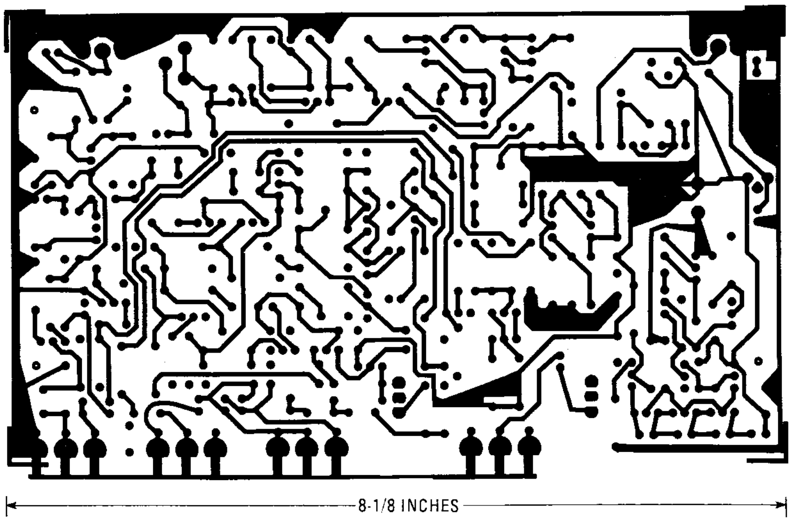


FIG. 5—THE FOIL SIDE of the board is shown here half size.

A look at the circuit

Now that we have an idea of the basic blocks of the color processor, let's take a closer look at the circuit. Its schematic is shown in Fig. 3. The video signal enters at J1 and is buffered by Q1. The color subcarrier, including the colorburst, is separated from the luminance and sync information by a filter made up of L1, C4 and C5. The luminance and sync are amplified through Q2 and Q3, and the DC level is restored by Q4, which clamps the video at the back-porch level. The back-porch timer (made up of Q14, Q15, Q16, Q17, Q18) generates a pulse during the back porch of the signal at Q3. That pulse tells the luminance-and-sync separator and the burst-and-chroma separator when to start their action.

The sync and luminance are separated by Q5, Q6, Q7, and Q8, and sent to the luminance-fader circuit and the sync-restoration circuit. The sync is amplified and shaped by Q7 and diodes D1, D2, and D11, and is then sent to Q9 and Q10, which feed the top of the FADER control. The luminance is amplified by Q11 and is fed to the other side of the FADER control. When the slider of R31 moves toward the collector of Q11, both sync and video are passed. But when the slider is moved away from Q11, the luminance will be attenuated.

The back-porch timer synchronizes the color-processing section by gating (Q22 and Q28) the color-sync burst independently from the color signal. The chrominance subcarrier is amplified by Q19 and Q20. The color burst is phase shifted by Q25, C25, and R73. The TINT control (part of a variable phase-shift network) composed of R86, C27, C28, and Q27 gives additional control over the phase shift. The color burst is then mixed back with the sync pulses through C10 and R27 and is fed to Q9. The remainder of the chrominance subcarrier is amplified by Q23 and Q24 and then limited by D6 and D7, the FLASH FILTER. After limiting, the background-noise gating diodes D8 and D9, whose bias is controlled by R81, set an amplitude that is adjusted to block out low-level noise. The LEVEL control, R82, adjusts the amount of color carrier. The chrominance signal is then mixed back in with the luminance at the emitter of Q11. The output of the FADER control feeds the output amplifier Q12, Q13, and then to the video output J2. From the output of Q13 the total signal is amplified by Q31 and goes to the meter-drive circuit. The signal is clamped at Q33 by Q32 (driven by a timing pulse from the sync separator, Q15). That gives a reference level for amplifier Q33. The signal is amplified and rectified through Q34, which drives the display circuit made up of Q35, Q36, Q37, Q38, and LED1 – LED4.

Construction

A double-sided printed-circuit board is definitely required for this project so that stray capacitance is kept to a minimum. Half-size foil diagrams are shown in Figs. 4 and 5. Figure 6 shows a parts-placement diagram.

A few construction tips are in order. When installing parts, it's important that all parts are inserted in their correct place. (That may seem too obvious to say, but *double check* your work!) Make sure you watch out for polarity, too, especially with transistors, diodes, and electrolytic capacitors. It is also important that all leads be kept as short as possible, because of the

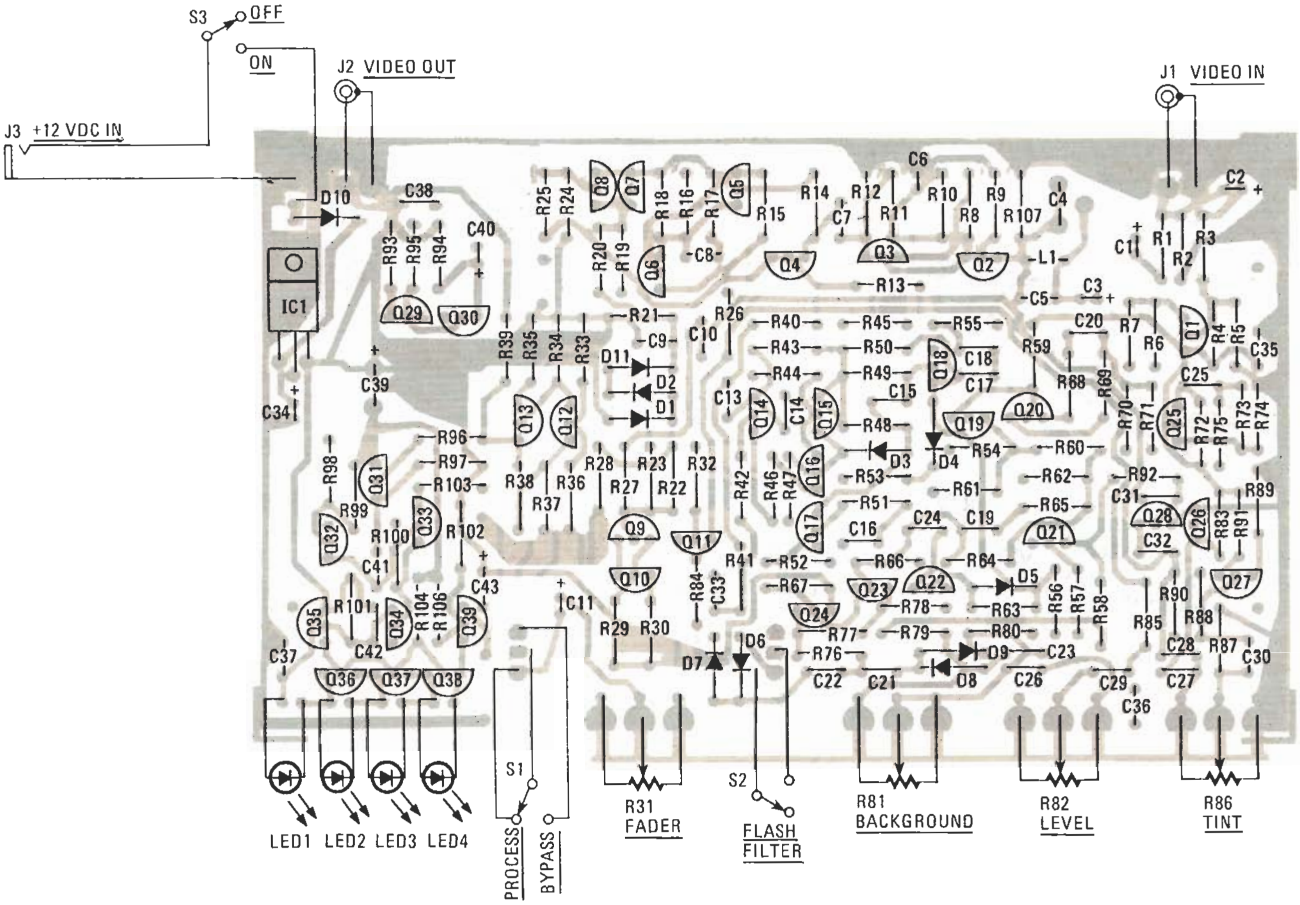


FIG. 6—PARTS-PLACEMENT DIAGRAM. Note that the voltage regulator, IC1, is mounted to the foil with a nut and bolt.

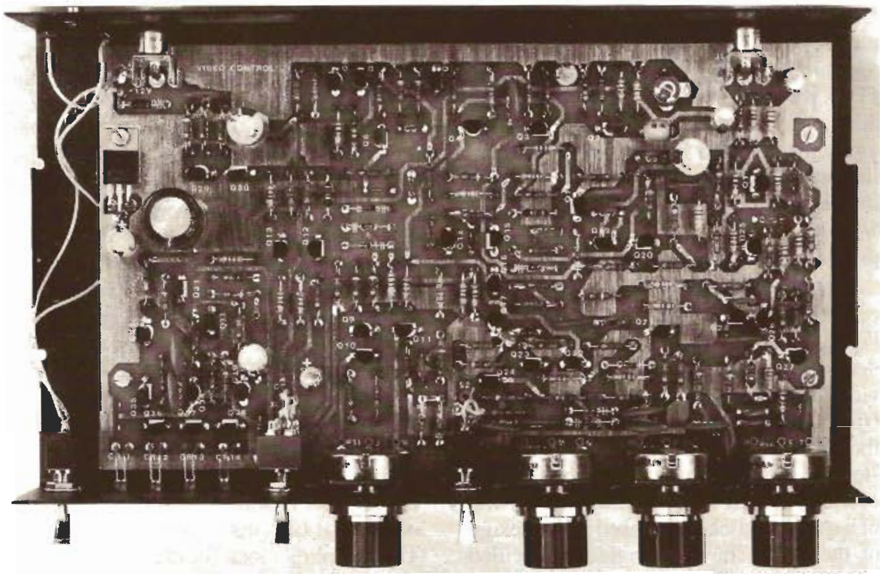


FIG. 7—A METAL CASE IS NECESSARY to ensure shielding. The potentiometers and phono jacks are mounted directly to the board without interconnecting wires.

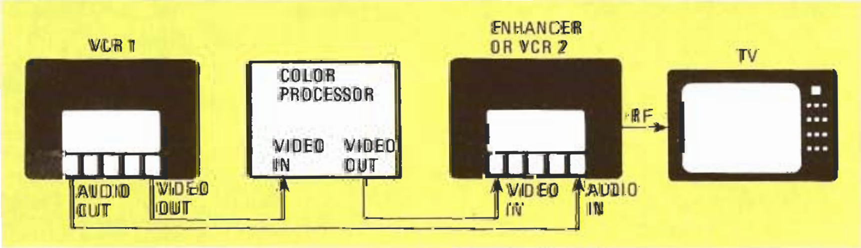
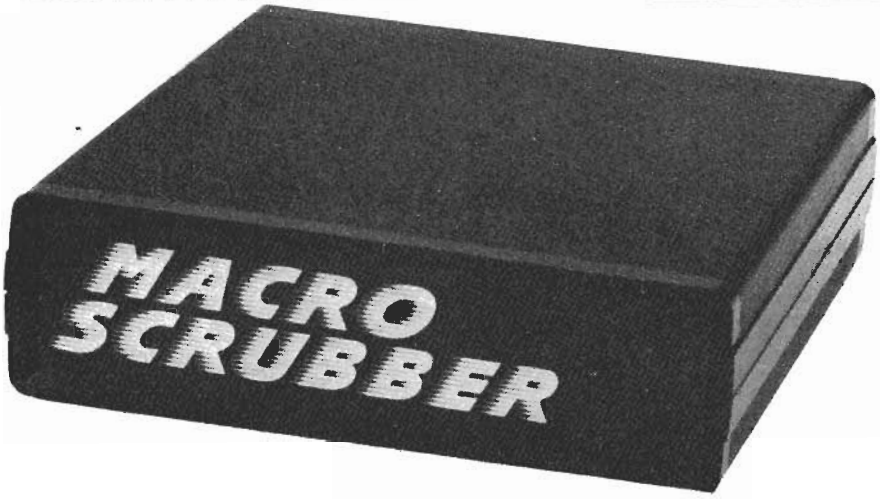


FIG. 8—ONE POSSIBLE HOOKUP for the color processor is shown here.

high frequencies involved. Be careful when you solder. Avoid cold solder joints and remove any flux residue with alcohol after soldering. Then you can inspect the board to make sure that there are no solder bridges between the traces.

The project should be mounted in a metal case to ensure shielding. One possible method is shown in Fig. 7. However, before you place the cover on the chassis, you have to calibrate C4. To do that, you'll have to hook up the unit as shown in Fig. 8. Connect the VIDEO IN jack of the unit to the VIDEO OUT jack of VCR1. Connect the VIDEO OUT jack of the processor to the VIDEO IN jack of VCR2. (If you do not have a second VCR, you can use an enhancer or an RF modulator.) Connect the enhancer's output or the RF output of VCR2 to the TV's antenna input. For calibration, the front-panel controls of the processor should be set as follows: Power switch in ON position, PROCESS/BYPASS switch in PROCESS position, FADER control at 12 o'clock (adjust for slight flickering of 100% PICTURE LEVEL LED), FLASH FILTER in OFF position, BACKGROUND control fully counter-clockwise, LEVEL control fully counter-clockwise, and TINT control at 12 o'clock. After completing the preliminary adjustments, C4 can now be calibrated. Adjust C4 until color disappears and only a black-and- white picture is on the TV screen. Now the processor is properly calibrated. **R-E**



MACROVISION STABILIZER

Are copy-protected video tapes also mucking-up normal viewing on your TV? Then use a Macro-Scrubber to make the picture squeaky-clean.

D. DUPRE

YOU ARE PROBABLY ALREADY AWARE that the movie industry has launched a new front against video tape copying with a new "encoding" scheme called *Macrovision*. Although many new releases from Embassy, CBS/Fox, MGM/UA, HBO/Cannon, MCA, and Disney have been protected with it, its use is generally not advertised on the label. However, you can easily identify a *Macrovision*-processed tape by turning the vertical hold control on your TV (if your set has one) so that the black bar across the top of the picture becomes visible. If the signal contains *Macrovision* encoding, you will see five or six gray or white pulsating "boxes" on the left side of the black bar. According to a top *Macrovision* executive, plans are already in the works to transmit *Macrovision*-encoded signals through cable systems.

The basic idea behind the *Macrovision* process is to render the program material uncopyable to a VCR while allowing the unimpaired viewing of the original tape (a goal not achieved by the original *CopyGuard* system, which has since passed away). Although some proponents of the *Macrovision* process claim that the system meets those goals, numerous consumers who have either purchased or rented a number of *Macrovision*-encoded tapes can attest to the contrary. That is evidenced by the large influx of letters to magazines predominant in the video field, and by continuous complaints to video rental and retail stores.

Both the users and developers of the *Macrovision* process admit that some TV's and VCR's are adversely affected in the PLAY mode, but that that percentage is very small. So, if you are one of the "small percentage" you probably have a significant sum of money invested in the best features that state-of-the-art video has to offer; yet with it you wind up watching a dark, murky picture that may be flashing, rolling or streaking as well.

If you're among the users who have discovered that your VCR or TV equipment simply can't handle the so-called

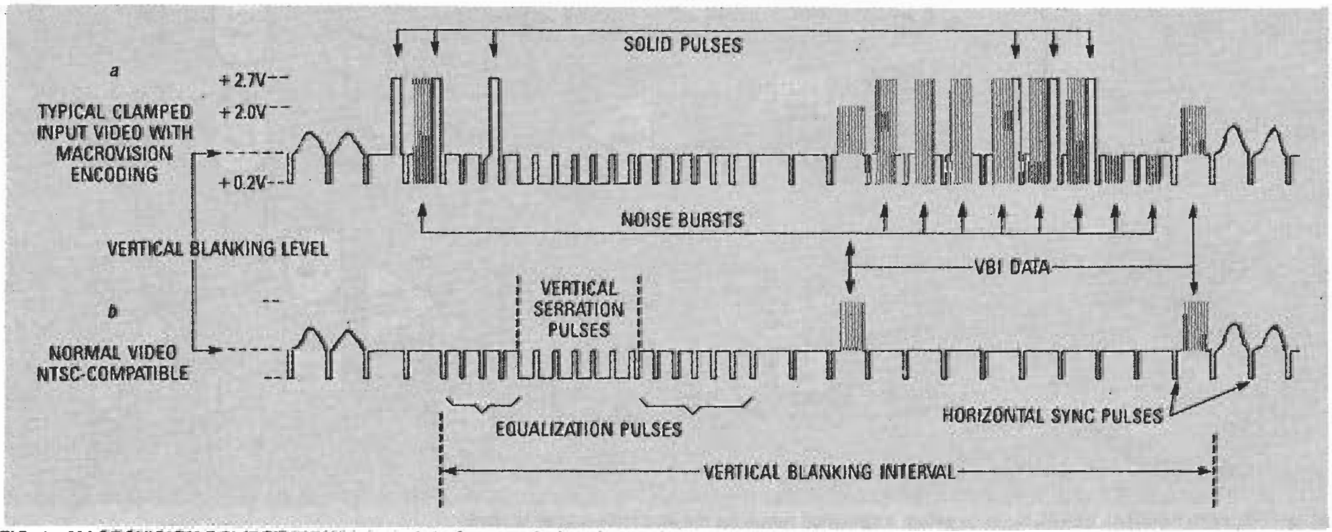


FIG. 1—MACROVISION DELIBERATELY injects interference during the vertical-blanking interval (*a*). The Macro-Scrubber restores the signal to standard NTSC (*b*).

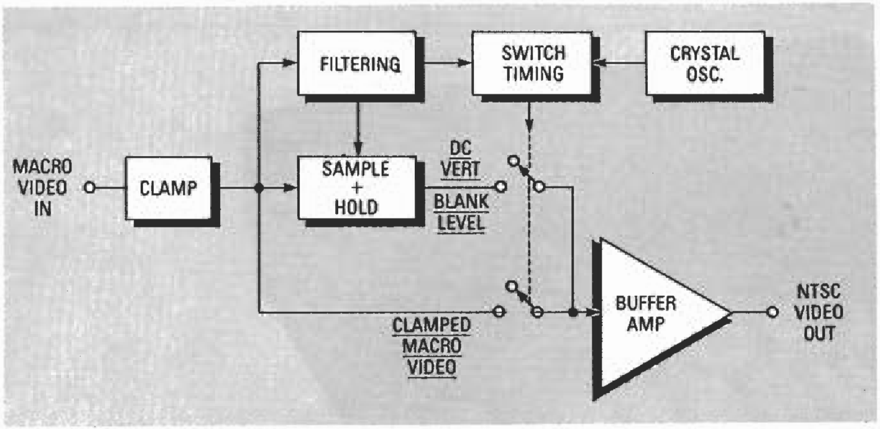


FIG. 2—AN ELECTRONIC DPDT SWITCH is used to restore the video signal to the conventional NTSC format.

WARNING

Duplication of copyright material is prohibited by law. The Macro-Scrubber is recommended for use only between a VCR and its TV or monitor as a solution to viewing problems that are generated within the TV or the monitor by *Macrovision* encoding.

"invisible" *Macrovision* encoding, then you need the Macro-Scrubber, a device that simply eliminates the encoding. Plug the device between your VCR and its TV or monitor and you won't know that *Macrovision* even exists.

Macrovision encoding

Macrovision is not an encoding process at all. If it were, then an appropriate decoder would have to be made available to the consumer just so he could view a *Macrovision*-encoded tape. In fact, the video information remains intact and unmodified in the signal, as does the audio. The encoding is really a disturbance in the TV signal's vertical-blanking interval that is supposed to affect only a VCR attempting to copy the tape. Unfortunately, the encoding also affects some TV's. By simply eliminating the disturbance and returning the offending signal to normal NTSC standards you can view a completely normal picture while playing a *Macrovision*-processed tape.

As shown in Fig. 1, the *Macrovision* process merely injects noise bursts and solid pulses into the signal during selected line times within the vertical blanking interval. One possible form of *Macrovision* encoding is shown in Fig. 1-*a*; the same signal in conventional NTSC form is shown in Fig. 1-*b*. The peak level of the bursts is randomly varied from black to white. Sometimes the bursts are pumped between two or three different levels; at other times the burst level is ramped slowly up and down. The location of the injected noise is also randomly alternated between the available line times; however, the location and level of solid pulses usually remains constant for the duration of a particular title—thus all copies of a particular title have the same encoding.

The *Macrovision* irregularities created during the vertical retrace time are intended to upset a VCR's RECORD-mode AGC circuit so that it records an unviewable picture. Since a VCR is designed to record only the NTSC video signal—which contains no noise transitions during the vertical blanking interval—any fast irregularities in the vertical blanking interval cannot be tracked by the AGC.

The effectiveness of the *Macrovision* anti-copying system varies with the type of VCR used, but in general, synchronization is lost, leaving an unviewable picture on the attempted copy. At best, the resulting dubbed copy will exhibit erratic brightness changes. Sometimes the picture will roll vertically due to a noise burst injected just before vertical sync.

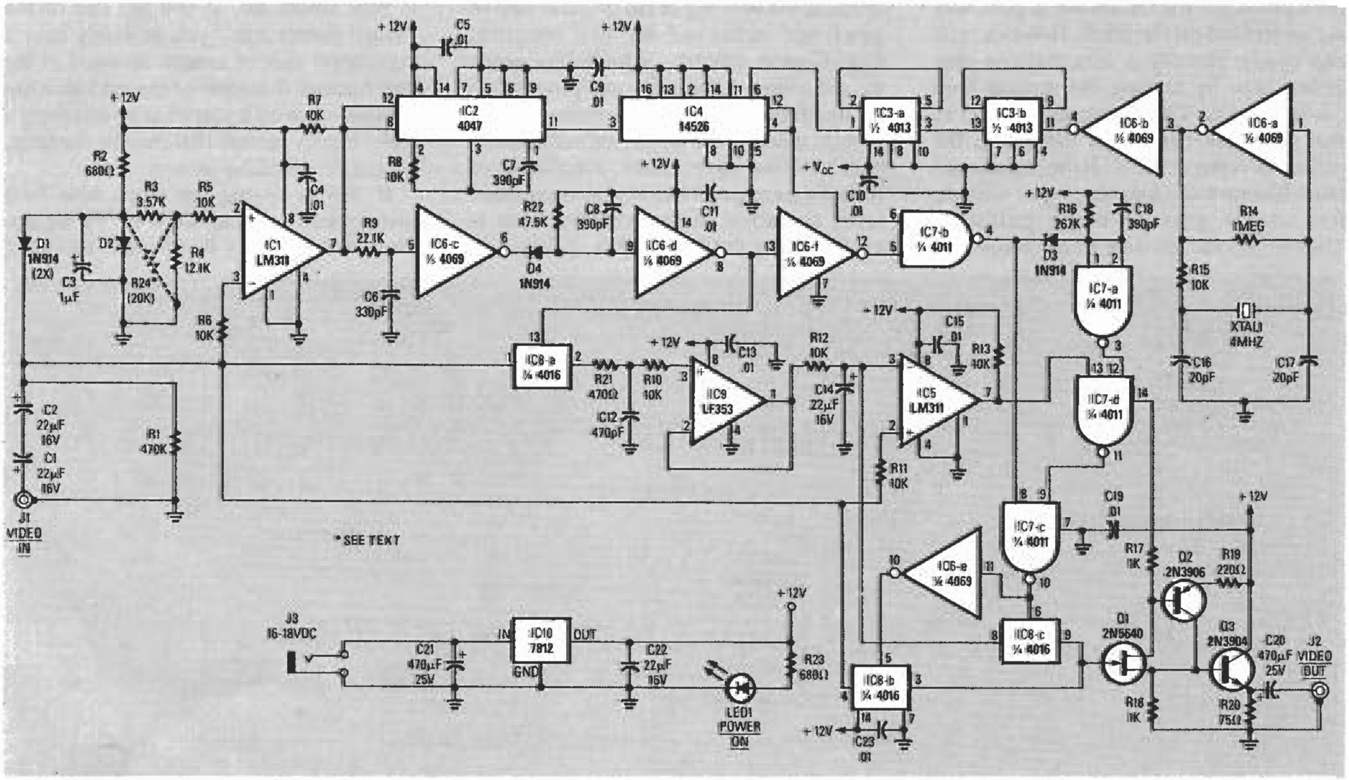


FIG. 3—THE INPUT-SIGNAL LEVEL must be within a specified range for the Macro-Scrubber to work properly. If there is some problem with the input-signal level it is suggested that R24—shown with dashed lines—be substituted for R3 and R4.

COMPLEX IS REALLY BETTER

The majority of devices sold with a purpose similar to that of the Macro-Scrubber either blank or clip the entire vertical-blanking interval, which also removes any other VBI data (color correction signals, closed captions, teletext, etc.). They then attempt to economically reconstruct the entire vertical-blanking interval, including horizontal sync, equalization, and serration pulses. From a technical viewpoint, that is often not very successful because the characteristics of the reconstructed pulses are usually set to some form of "standard" value, and as such do not really match the actual input signal. Since those units do not actually detect and selectively remove the *Macrovision* noise, they also strip the vertical-blanking interval of a normal TV signal if not bypassed or removed from the system.

Many models also require factory-type calibrations of numerous timing potentiometers. Their new blanking level is either preset to some kind of standard value or an external adjustment is provided to compensate for signals from different video tapes or sources.

What makes the Macro-Scrubber unique, when compared to the restoration devices, is that there are there are no precision adjustments; digital filters remove only the *Macrovision* pulses and pass the original vertical-blanking interval data and sync pulses, while sample-and-hold circuits reproduce the correct vertical-blanking level, which is switched into the output signal in place of *Macrovision* pulses. Also, the use of a crystal oscillator eliminates the need for timing adjustments. And since the Macro-Scrubber has no effect on normal video signals, there is no need to switch the Macro-Scrubber out of the system for normal viewing.

Unfortunately, similar symptoms sometimes are experienced when merely playing the original tapes on some VCR/TV combinations.

A logical solution

In comparing Figs. 1-*a* and 1-*b*, you can see that a normal NTSC video waveform is held at the vertical blanking level during times when injected noise exists in the *Macrovision*-encoded waveform. By locating the individual noise bursts and solid pulses, and by connecting the output to a DC voltage that is equivalent to the vertical blanking level at those times, we can essentially recreate the NTSC version of the waveform. When no encoding signals are present, we connect the output to the clamped video input.

Figure 2 shows a block-diagram of how the encoding can be removed. First, the incoming video signal is clamped to hold the negative sync tips at the same level, thereby removing any AC hum or other time-varying offset from the signal; that step is critical for detecting signal transitions against a fixed reference level. The clamped video is then sent to a filtering circuit to accurately locate the noise bursts and the solid pulses from which the switch-timing and the control signal are created. (A crystal oscillator is used so that timing adjustments aren't necessary.)

The sample-and-hold circuit continuously samples the video waveform to generate a DC voltage equivalent to the vertical blanking level of the incoming signal. That assures that the switched-in blanking level is always correct for the actual input signal applied and eliminates the need for any manual adjustment. Finally, an electronic double-pole, single-throw switch that is controlled by the noise-locating signal connects either the clamped input video or the reproduced blanking level to the output buffer amplifier. In so doing, *Macrovision* noise is eliminated and the signal is restored to normal NTSC video.

Circuit description

Figure 3 shows Macro-Scrubber's circuit. The *Macrovision*-encoded video signal is applied to jack J1 and is fed through back-to-back capacitors C1 and C2 to a resistor/diode network that clamps the negative sync tips close to ground potential (approximately 0.2 volt). The clamped video input applied to IC1's inverting input (pin 3) resembles the waveform shown in Fig. 1-*a*. A DC reference voltage derived from diode D2 is fed to IC1's non-inverting input (pin 2). Since the DC reference is slightly higher than the clamped voltage, IC1's output goes positive whenever the input signal is lower than the reference signal. As shown in Fig. 4-*a*, IC1 outputs a waveform that is normally low (0 volts) with high-going pulses con-

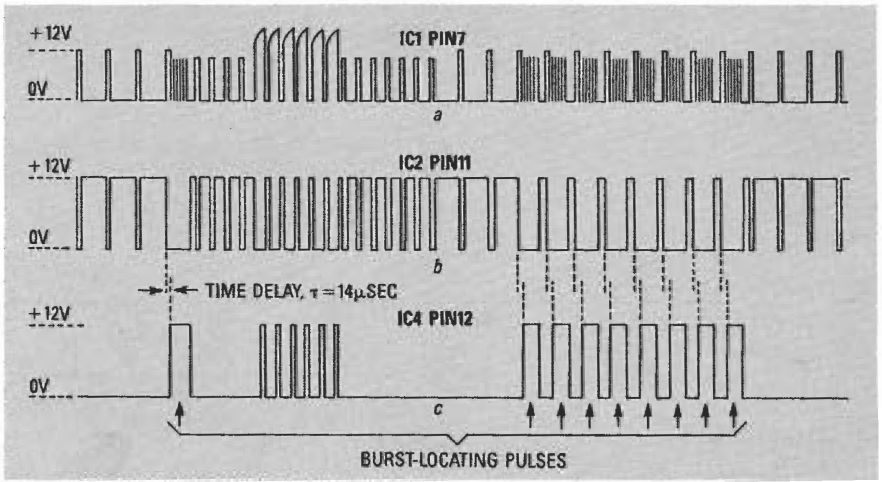


FIG. 4—THE MACRO-SCRUBBER GENERATES noise pulses which are used to locate and suppress the *Macrovision* noise-burst interference.

PARTS LIST

All resistors are ¼-watt, 5% unless otherwise noted.
R1—470,000 ohms
R2, R23—680 ohms
R3—3,570 ohms, 1%
R4—12,100 ohms, 1%
R5–R7, R10–R13, R15—10,000 ohms
R8—10,000 ohms, 1%
R9—22,100 ohms, 1%
R14—1 Megohm
R16—267,000 ohms, 1%
R17, R18—1,000 ohms
R19—220 ohms
R20—75 ohms
R21—470 ohms
R22—47,500 ohms, 1%
R24—20,000 ohms, trimmer potentiometer
Capacitors
C1, C2, C14, C22—22 µF, 16 volts, electrolytic
C3—1 µF, 35 volts, electrolytic
C4, C5, C9–C11, C13, C15, C19, C23—0.01 µF, ceramic disk
C6—330 pF, NPO
C7, C8, C18—390 pF, silver mica
C12—470 pF, polypropylene
C16, C17—20 pF, NPO
C20, C21—470 uF, 25 volts, electrolytic
Semiconductors
IC1, IC5—LM311 comparator
IC2—4047B multivibrator
IC3—4013B dual D flip-flop
IC4—4526B binary counter
IC6—4069B hex inverter
IC7—4011B quad NAND gate
IC8—4016B quad analog switch
IC9—LF353 dual JFET op-amp
IC10—7812, 12-volt regulator
Q1—2N5640, JFET transistor
Q2—2N3906, PNP transistor
Q3—2N3904, NPN transistor
D1–D4—1N914, switching diode
LED1—Light-emitting diode
Other components
XTAL1—4-MHz crystal, AT cut (parallel), HC-18 package
J1, J2—RCA-type phono jacks
Miscellaneous: PC board materials, 16–18-volt, 200-mA AC adapter, etc.
Note: A complete Macro-Scrubber kit, model *MAK-1*, which includes the PC board, cabinet, all components, and an AC adapter, is available for $52.95 plus $3.00 shipping and handling from: The Hobby Helper, P.O. Box 308, Bridgewater, MA. 02324. (617) 339-1026. Massachusetts residents must add appropriate sales tax.

current with negative-going pulses below the vertical-blanking level (i.e. horizontal-sync pulses, equalization pulses, vertical-serration pulses, and *Macrovision* noise bursts).

The clamped video signal is also delivered to the input of a sample-and-hold circuit, which consists primarily of analog switch IC8-a, hold capacitor C12, and op-amp IC9. The switch is driven in such a way that it samples the level of the video signal only during vertical-sync time at the peaks of the vertical-serration pulses. The output of IC9 is a DC voltage equal to the vertical-blanking level, which will be switched into the output waveform in place of the *Macrovision* noise bursts and solid pulses.

Clamped input video is fed to the input of analog switch IC8-b, and the DC output voltage from IC9 is fed to the input of analog switch IC8-c. Notice that the outputs of analog switches IC8-b and IC8-c are connected together, and that because of inverter IC6-e, their control inputs at pins 5 and 6 respectively are driven 180° out of phase. That arrangement creates the electric equivalent of a single-pole, double-throw switch, with either the clamped input video or a DC voltage that is equal to the vertical-blanking level being fed to buffer amplifier Q1 at any one time.

It is through control of the electronic DPDT switch shown in Fig. 2 that the encoded video is restored to a normal NTSC signal. All that is needed is a proper signal to pin 6 of IC8.

The signal from IC1 pin 7 (Fig. 4-*a*), which contains pulses that correspond to the sync and the *Macrovision* noise pulses, is fed to IC2 pins 8 and 12. IC2 is a multivibrator that is configured as a digital low-pass filter. The time constant determined by R8 and C7 causes frequencies greater than twice the horizontal frequency to be filtered out. IC2's output at pin 11 looks like a squared-up and inverted version of its input, except that the high-frequency pulses corresponding to the *Macrovision* noise bursts have been filtered out, leaving a low level for each burst duration. (See Fig. 4-*b*).

The filtered signal is fed to pin 3 (ENABLE) of binary counter IC4. A 4-MHz crystal-oscillator circuit feeds dual flip-flop IC3, which divides the crystal frequency by four, yielding a 1-MHz clock input to IC4 pin 6. When the input at IC4 pin 3 goes high, that counter is asynchronously preset to the binary value determined by preset lines P3, P2, P1, and P0 (pins 2, 14, 11, and 5 respectitvely). With the connections shown in Fig. 3, the preset count is 14 decimal (1110 binary). Whenever the count is not zero, the counter output at pin 12 is low. The counter decrements once for each clock pulse it sees on pin 6 while pin 3 remains low. Thus, 14 μs after the leading, negative-going edge of the input signal on pin 3 goes low, the count reaches 0 and the counter's output switches high. The high output is fed back to pin 4, the INHIBIT line, which prevents any further counting. When the input signal at pin 3 returns high, the counter is again preset to a 14 count and the output returns to its low preset state. low input pulses having a duration less than 14 μsec are ignored because the counter is preset before the count ever reaches zero.

The resulting output signal at IC4 pin 12 is normally low, with high-going pulses that start 14 μs after the beginning of each horizontal-sync pulse that precedes a *Macrovision* noise burst. The 14 μsec delay forces the horizontal sync pulses (and color-bursts) to be switched into the output waveform. Each of the noise-burst locating pulses returns to a low at the end of the corresponding *Macrovision* burst. Those pulses, as shown in Fig. 4-*c*, define the points in time when the bursts occur, with one exception. Concurrent with the vertical-serration pulses, there are a string of pulses that must be removed in order to create a signal that will totally isolate the *Macrovision* noise. In order to remove those pulses we must create a gating signal with a single pulse that lasts only for the duration of the vertical sync pulse in each frame. To do that, sync and noise pulses from IC1 pin 7 (Fig. 4-*a*) are fed to a low-pass filter consisting of R9 and C6.

Narrow, positive-going pulses are attenuated because C6 never gets a chance to charge to a logic-high level unless the pulses are long compared to the time constant determined by R9 and C6. The only pulses wide enough to allow C6 to charge

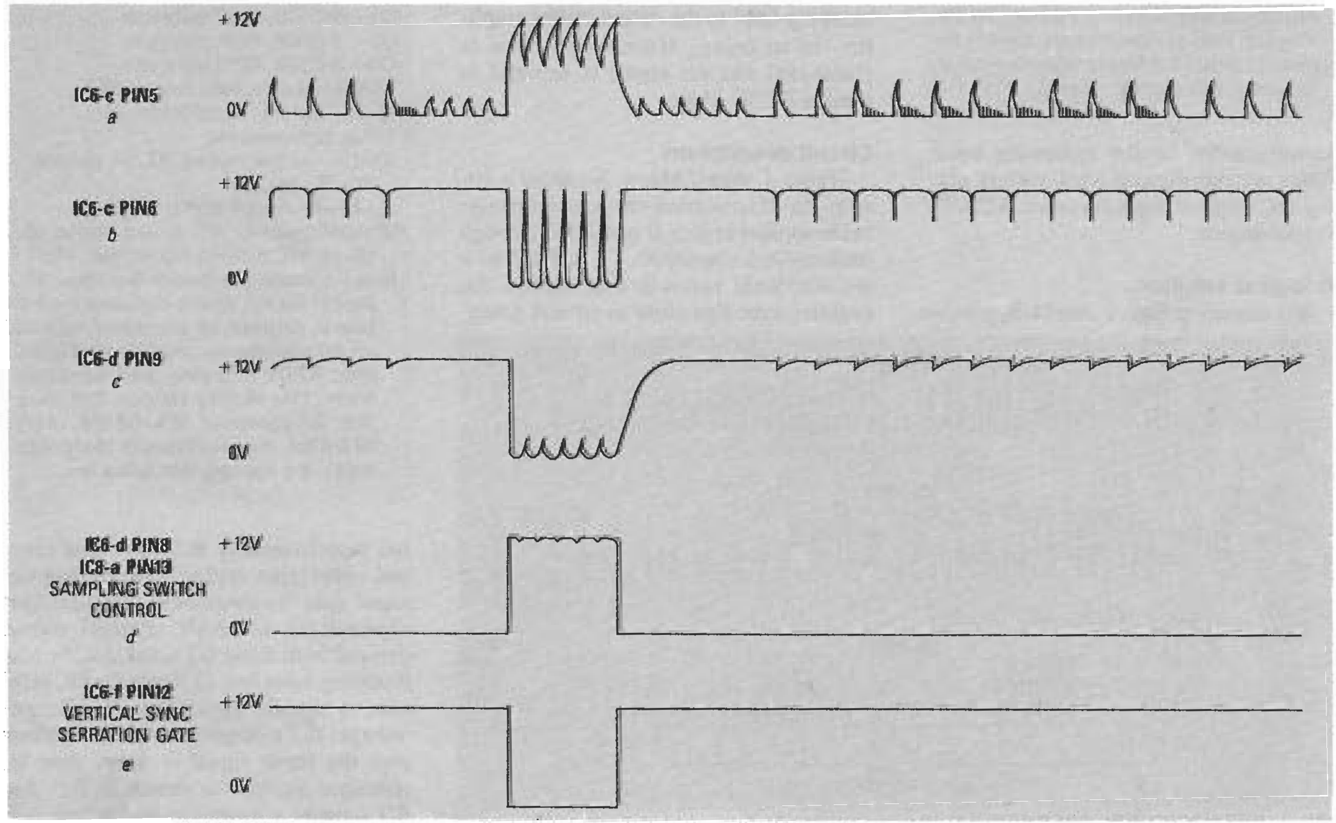


FIG. 5—VERTICAL SYNC AND BLANKING level sampling pulses are derived from the *Macrovision*-induced noise bursts.

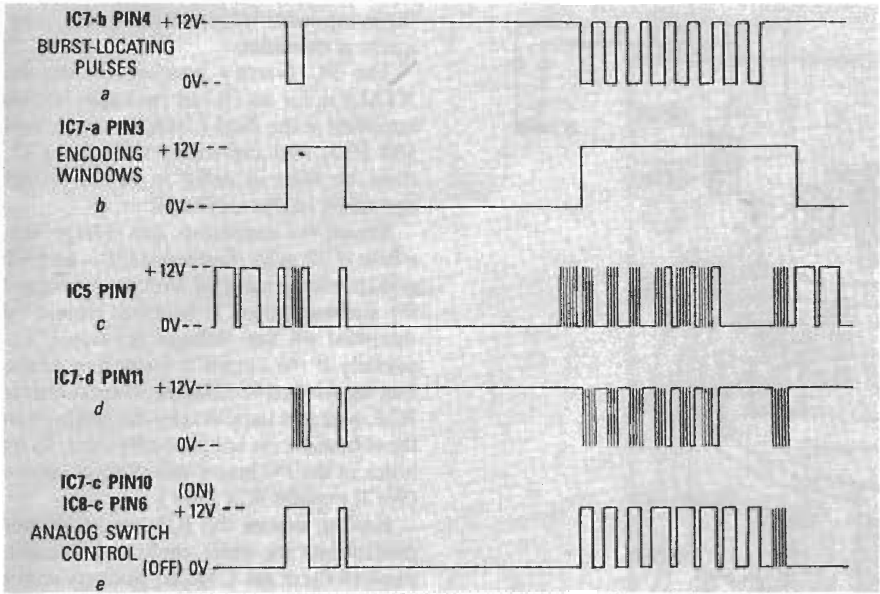


FIG. 6—WINDOWS DERIVED FROM the burst-location pulses provide the switch control signals that eliminate the *Macrovision* interference from the output signal.

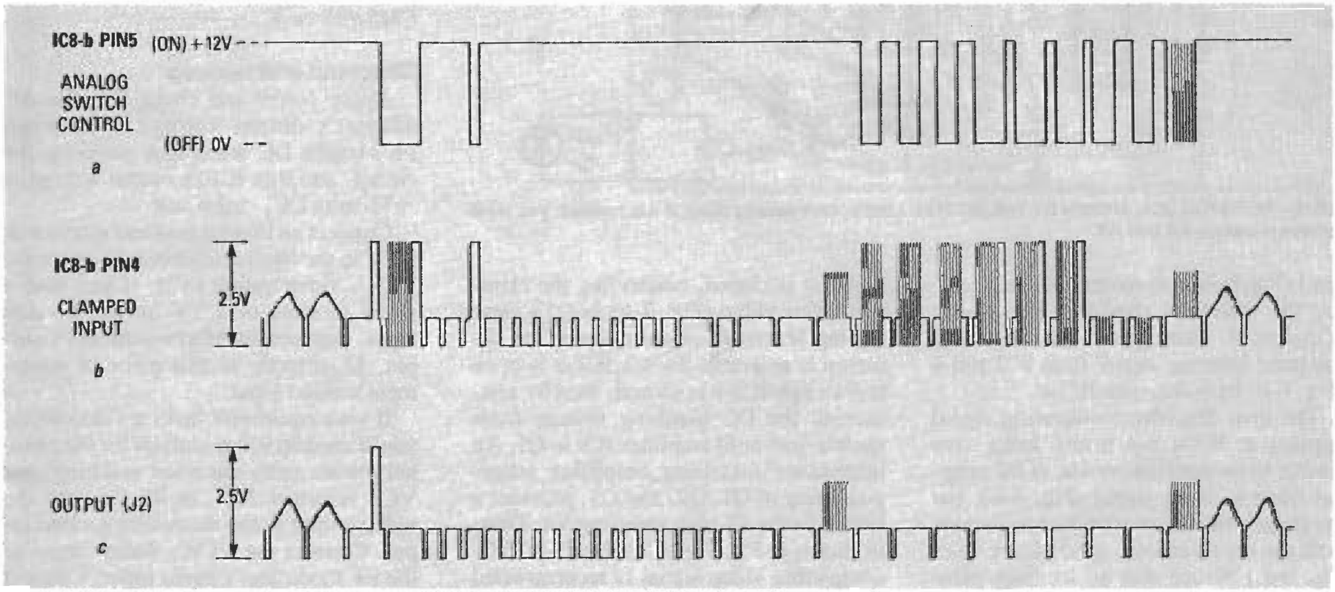


FIG. 7—THE NTSC SIGNAL IS CREATED by selectively switching the encoded input and the sampled vertical-blanking level into the output.

to a logic-high value are those corresponding to the vertical serration pulses.

Therefore, as shown in Fig. 5-*a*, at IC6-c pin 5 we have a signal that has +5-volt pulses corresponding to the horizontal sync pulses, and +12-volt pulses concurrent with the serration pulses during vertical-sync time.

Inverter IC6-c discriminates farther, since input pulses lower than a logic high (approximately +7V for a +12V supply) will not trigger output pulses. Therefore, as shown in Fig. 5-*b*, at inverter IC6-c pin 6 we have a normally high signal with low-going pulses occurring only during the vertical sync period. When that signal goes low, diode D4 becomes forward-biased and inverter IC6-d pin 9 is immediately pulled low as well. When IC6-c pin 6 goes high, D4 is reverse-biased, so that pin 9 charges to a logic-high level at a rate determined by R22 and C8. (See Fig. 5-*c*.) Again, narrow pulses are ignored and, as shown in Fig. 5-*d*, the inverter's output (IC6-d pin 8) is normally low with logic-high sync pulses.

The inverter's output signal turns on sampling-switch IC8-a during the vertical-serration time, and C12 charges to the vertical blanking level. The high input impedance of op-amp IC9, and IC8-a in its OFF state, prevent the charged voltage on C12 from leaking off between samples. Unity-gain amplifier IC9 feeds C12's DC-voltage level to analog switch IC8-c, where it will be switched into the output waveform in place of the *Macrovision* noise.

As shown in Fig. 5-*e*, the vertical-sync signal at IC6-d pin 8 is inverted by IC6-f, and is then fed to NAND gate IC7-b to gate the burst-locating signal from IC4 pin 12 (Fig. 4-*c*). The output signal at IC7-b pin 4 (Fig. 6-*a*), is normally high, with logic-low pulses that last for the duration of the corresponding *Macrovision* noise burst.

Notice, from Fig. 1-*a*, that some *Macrovision* bursts are followed by a solid pulse that doesn't have a negative transition. Since the *Macrovision* noise-locating pulses shown in Fig. 6-*a* were created by detecting high-frequency bursts below the blanking level, they do not locate the solid pulses. In order to remove the solid pulses without affecting any non-*Macrovision* pulses we must: 1) detect positive transitions above the vertical blanking level that occur during the *Macrovision*-encoded areas of vertical blanking, and 2) combine the new signal that locates the *Macrovision* solid pulses with the signal that locates the *Macrovision* noise bursts at IC7-b pin 4.

Macrovision-encoded areas of the signal are defined by feeding the signal at IC7-b pin 4 (Fig. 6-*a*) to a low-pass filter consisting of D3, R16, C18, and IC7-a. The resulting waveform at IC7-a pin 3 contains wide pulses, or *windows*, that define the time periods in which the *Macrovision* encoding is present. That signal is shown in Fig. 6-*b*.

The DC output voltage from the sample-and-hold circuit is fed to the inverting input of comparator IC5, while the clamped video input signal is fed to IC5's non-inverting input. A train of high-going pulses appears at IC5 pin 7 that corresponds to all transitions above the blanking level; including video, vertical-blanking interval data and *Macrovision* pulses. See Fig. 6-*c*.

The signal from IC5 pin 7 (Fig. 6-*c*) is gated by the window pulses from IC7-a

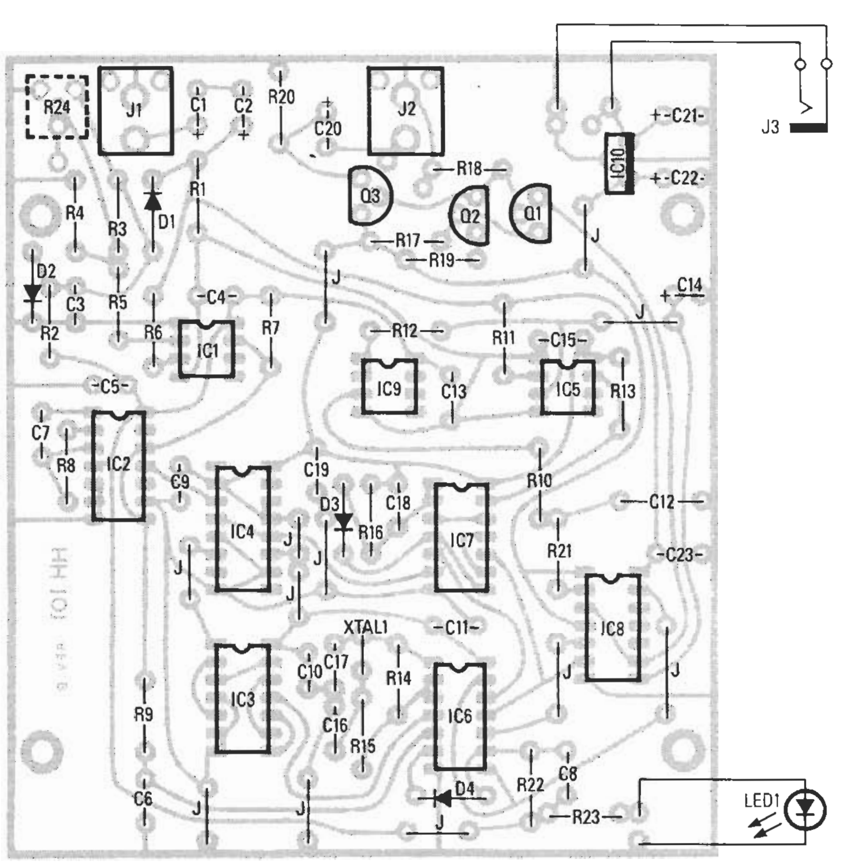


FIG. 8—RESISTOR R24, SHOWN BY THE DOTTED LINES, isn't usually used. If it is needed, you must remove resistors R3 and R4.

pin 3 (Fig. 6-*b*) at NAND gate IC7-d, yielding the waveform shown in Fig. 6-*d* at IC7-d pin 11. That signal is combined with the burst-locating signal from IC7 pin 4 (Fig. 6-*a*) by NAND gate IC7-c.

The final *Macrovision*-locating signal appears at IC8-c pin 6 and looks very similar to an inverted version of the original burst-locating signal (Fig. 6-*a*), but the pulses have been stretched to include both the bursts and the solid pulses. (See Fig. 6-*e*.) Notice that no locating pulse exists to remove the single solid pulse occurring at the start of the vertical blanking shown in the sample input waveform (Fig. 1-*a*). That is because there is no preceding negative noise burst that can be used to locate the solid pulse. As a result, that pulse will remain in the output waveform (Fig. 7-*c*), but it doesn't cause a problem because it is narrow and is not pumped.

The signal shown in Fig. 6-*e* directly feeds the control input, IC8-c pin 6, and an inverted form of the signal (Fig. 7-*a*) feeds analog switch IC8-b via inverter IC6-e.

The removal of the *Macrovision* encoding signal works as follows: During video time, horizontal-sync time, vertical-serration time, and all non-*Macrovision*-encoded vertical-line times, the control input of analog switch IC8-c is low—the switch is open. At the same time, the control input of analog switch IC8-b is high—it is closed, connecting the clamped-video video (Fig. 7-*b*) to Q1's gate. During *Macrovision* noise times, the situation is reversed. Switch IC8-b is open and switch IC8-c is closed, thereby connecting the DC blanking voltage from sample-and-hold amplifier IC9 to Q1. An impedance-matching amplifier stage, consisting of Q1, Q2, and Q3, provides a match for the 75-ohm video output. Thus, as shown in Fig. 7-*c*, a "normal" NTSC-compatible video signal is reconstructed at the output, eliminating only the *Macrovision* noise.

Construction

The circuit is assembled on a printed-circuit board. The foil pattern for that board is provided in PC Service. The parts-placement diagram for that PC board is shown in Fig. 8.

Begin stuffing the printed-circuit board by first installing all resisitors, diodes, and capacitors. Make sure that all of the electrolytic capacitors and the diodes are installed with the proper polarity. Save the clipped component leads for use as jumpers.

Capacitor C12 must be a "polypropylene" type because the extremely low-leakage characteristic of the material prevents the vertical-blanking hold voltage from sagging between samples. As correct R-C time constants are critical to the proper operation of the circuit, use of the component values shown in the schematic is essential.

The PC board's spacing for crystal XTAL1 is for an HC-18 package. Values specified in the Parts List for resistors R14 and R15, and capacitors C16 and C17, must be used in order to insure proper operation of the oscillator.

Mount the transistors and voltage regulator IC10 next. Transistor Q1 is an FET and should be handled with proper regard for static charges. A heatsink should be mounted on the voltage regulator, especially if the circuit is housed in a case that has limited ventilation. Potentiometer R24, which is indicated by dashed lines in the schematic, is not normally used, so its holes in the PC board will remain empty. (We'll explain R24 later.)

Finally, mount the IC's, using proper precautions for static electricity because most of them are CMOS. Sockets aren't necessary, but using them would make any troubleshooting or repair easier. The project will fit nicely into a PAC-TEC *CM5-125* case.

Checkout and hookup

Apply power and check that the AC adapter's output voltage is between 14–24-volts DC when it is powering the circuit, and that IC10's output voltage is +12-volts DC, ±0.6 volt.

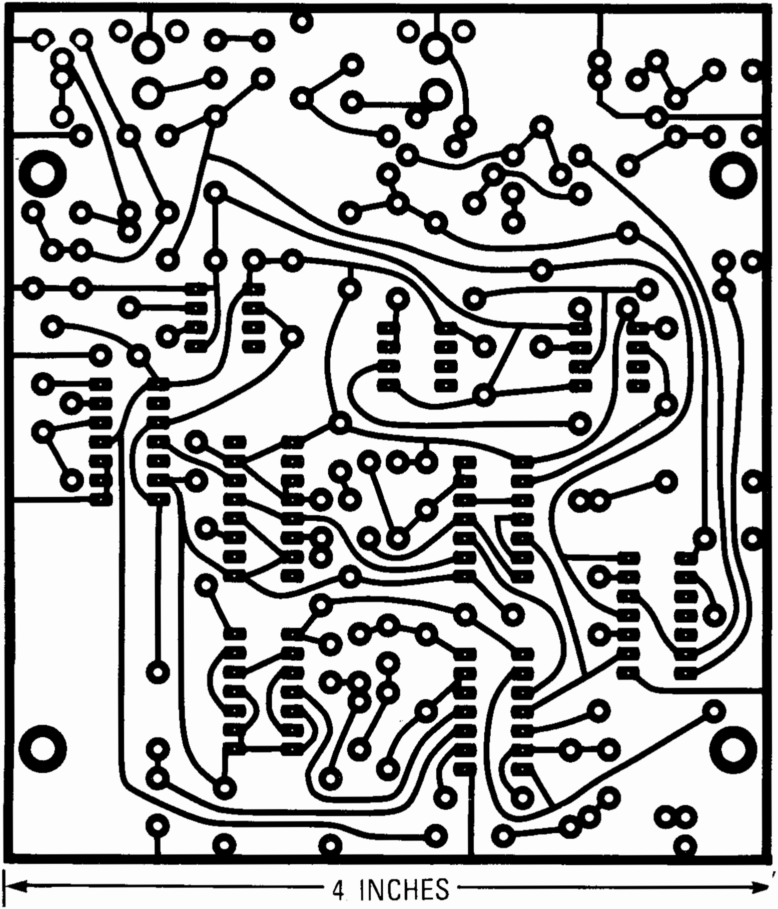
Connect an input signal and monitor or a TV to the Macro-scrubber. Connect the VCR's video output to J1. If you have a video monitor or a TV having a video input, connect the Macro-scrubber's output, J2, directly to that piece of equipment's video input.

If your equipment lacks a video input, you'll need an RF modulator for the channel you normally use when watching your VCR (channel 2, 3, or 4). Connect the video output to the modulator's video input. Connect the VCR's audio output to the RF modulator's audio input. Connect the modulator's RF output to the TV's antenna-input jack or terminals.

Play a video tape that you have already identified as containing *Macrovision* pulses. The picture you see should now be free of interference.

If the *Macrovision*-related viewing problems still exist, or if part of the picture is blanked out, the input-signal level from your VCR may be excessively high or low. In that case, a pattern of holes has been provided in the PC pattern so that fixed resistors R3 and R4 can be replaced with a 20,000-ohm potentiometer. The potentiometer is shown in the schematic by dashed lines, and is identified in the schematic and on the PC placement diagram as R24. (Remember, if you install R24 you must remove R3 and R4.)

To adjust R24, play a *Macrovision* tape, and while observing the TV picture, adjust R24 until the picture appears to be normal—interference-free. **R-E**



BUILD THE MACROSCRUBBER using this board.

VIDEO-EDIT CONTROLLER

Here's a low-cost device for professional-quality video editing.

THOMAS A. NERY

EXCITING HOME VIDEOS USUALLY REquire heavy editing—leaving the deadly-dull stuff "on the cutting-room floor." Unfortunately, the commercial video-edit controllers needed for *pro*-quality editing are usually priced beyond the budget of most video hobbyists, which means that most videos usually end up looking like just another home movie—or worse.

But there *is* a low-cost alternative to commercial video editing. It's our video-edit controller; a relatively simple device that requires the use of only two VCR's, or a VCR and camcorder, to edit video tapes electronically like a professional.

Home videos are usually edited by pausing the recording VCR at the point where recording is to begin, and pausing the source (player) VCR at the point where the new video starts. Once satisfied with the edit points, both pause buttons are released simultaneously to allow the playback and recording to start.

Although the procedure for "pause-editing" is theoretically correct, real life proves that theory and practice are not one and the same, because machines—particularly when dealing with precise timing—don't necessarily function the way we would like them to. The variations in the pause-timing characteristics of VCR's and camcorders usually result in several seconds of lost picture at the edit.

Editing controller

But use our video-edit controller and you will eliminate the lost snatches of picture when editing. That's because our controller allows video editing by frames, rather than by time periods.

To understand the operation of the video-edit controller, it's necessary to understand why several seconds of video are lost when using the simultaneous-pause-release method of videotape editing.

When the source (player) VCR's *pause* is released, the machine starts playing slightly after the point where the tape is positioned. The "slightly after" is a function of the tape getting up to speed before the video is output. The recorder, on the other hand, must synchronize itself to the source. To accomplish that, most newer VCR's—as well as camcorders—use a feature known as *preroll.*

Preroll

Preroll means that the recorder is rewound a predetermined number of frames, put into the play mode, and then shifted into *record* at the point where the recording is actually to start. When editing by dual-pause control, the additive "true start" delays of the source and record machines usually result in several seconds of missed video from the source.

There is also a synchronization problem associated with the dual-pause method of editing. Specifically, the recorder is being asked to synchronize itself to two different sources: the video prior to the source-VCR's getting up to speed, and then the video once speed is attained. That complicates the recorder's operation, and can result in video-breakup at the edit point.

Pro-quality editing

On the other hand, our video-edit controller does not depend on pause controls: It edits in a way similar to some professional editors. First, it rewinds the source-VCR for a fixed amount of time and then switches the VCR to the *play* mode. At the appropriate time, while the source-VCR is playing, the controller starts the recording VCR. The recording VCR uses up its preroll, comes up to speed, and then switches to the *record* mode. If all the timings are correct, the source-VCR is feeding the selected edit frame at the precise instant that the-recording VCR switches to the *record* mode.

Overall editing accuracy is dependent on the ability of the source- and recording-VCR's to consistently repeat their operations in exactly the same time periods. Since the recording- VCR's preroll is designed by the manufacturer to always start the recording after a fixed time interval, it is the source-VCR that's the main synchronizing problem.

Review to time

But we can make the source-VCR's rewind timing more or less consistent if we use the machine's *review* function—rather than the *rewind* function—to back up the tape. That is due to the fact that *review* is a *capstan*-driven function that always operates at a predetermined multiple of the nor-

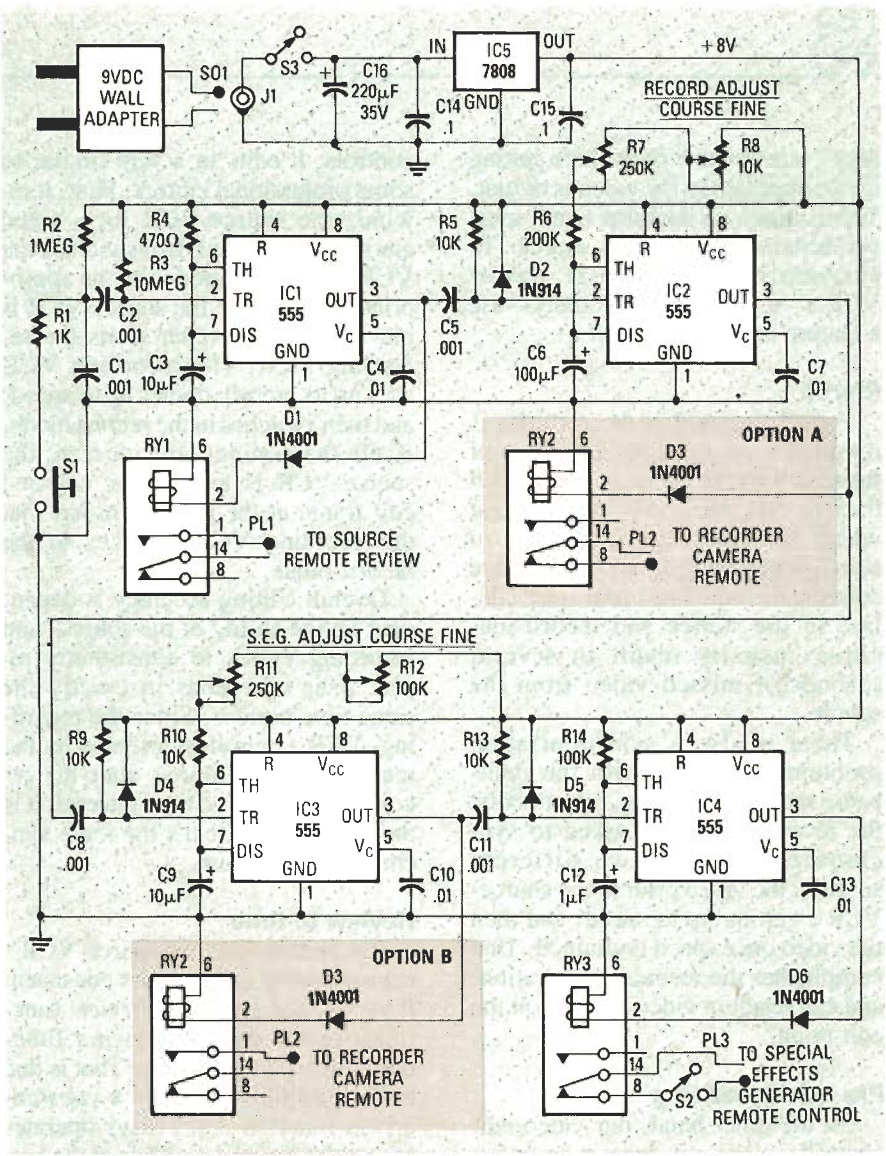


FIG. 1—THE VIDEO-EDIT CONTROLLER basically consists of four similar timer circuits. Both the OPTION A and OPTION B circuits for relay RY2 are built into the PC board. Simply plug the relay into the appropriate socket.

mal play speed. During *review*, the tape is always backed up the same length per period of time; whereas, during *rewind*, the actual amount of tape backed up per unit of time depends on how much tape is left on the supply reel.

In addition to the edit function, the controller also provides a switching circuit for a special-effects generator, such as you might use to cause a fade from, or to, black at the correct time.

How it works

The edit-controller, shown in Fig. 1, consists of four monostable timers. Each timer has the capability to drive a relay, although only three relays are used to interface to the controlled devices. To accommodate different remote-control circuits, relay RY2 can be installed at the locations labeled OPTION A, or OPTION B—more on that later.

The edit operation is started by closing switch S1, which causes a rapid drop to ground of the voltage across capacitor C1. C1's discharge causes a negative-going spike through C2, which triggers timer IC1. The triggering of IC1 causes RY1's contacts to close, and they remain closed during IC1 timing period. The timing period is determined from the equation:

$$\text{time} = 1.1(R4 \times C3)$$

The source-VCR's remote-control *review* jack is connected to RY1's contacts through PL1. The VCR will be held in the *review* mode during IC1's timing period. At the end of the timing period, RY1 is released, its contacts open, and the VCR automatically switches from the *review* to the *play* mode. Also at the end of the timing period, IC1 triggers timer IC2.

Timers IC2–IC4 operate in a similar manner as IC1, the major difference being that IC2 and IC3 have *coarse* and *fine* adjustments for tweaking the time-period. Also, RY2 can be driven either by IC2 or IC3, depending on the requirements of the recording-VCR. If the recorder is started by opening its remote control, RY2 is installed at the OPTION A location. If the recorder is started by closing its remote control, then RY2 is installed at the OPTION B location.

The editor's timing constants are a function of both the type and the speed of the VCR's. While the principles can be applied to any combination of VCR's and speeds, the prototype assumes VHS machines operating at the SP speed. Should a different combination be desired, it will be necessary to adjust the timing components for the selected speed.

Construction

Before building anything, you must make certain that your source-VCR is compatible with the controller. Place a tape in the VCR and start the play. After about 30 seconds, depress the *pause* button. Once the VCR has come to a complete stop—as indicated by a frozen frame on the screen—press and hold the *review* (or dual-function *review/rewind*) button for about five seconds and then release it. The VCR is compatible with the video-edit controller if it rewinds and then automatically enters the *play* state when the *review* button is released. If releasing the button did not cause the VCR to switch automatically into the *play* mode, then it can't be used with the controller.

If the VCR passes the compatibility test, you must make a review-switch modification. Disconnect the VCR from the powerline, open the VCR's case, and locate the *review* switch's contacts. Use a VOM to verify that you have selected the correct contacts. (In some VCR's the *review* switch has DPST contacts that are wired in parallel.) Solder a pair of thin, insulated, stranded wires (i.e. 22 gauge) to the switch's contacts. Then route the wire to an accessible blank portion of the VCR's rear apron. Carefully drill a hole in the apron for a miniature phone jack that will mate with PL1. If the cabinet is metal, use two contacts of a 3-circuit jack and change PL1 to a 3-circuit miniature phone jack. (The plug's *sleeve* connection—which is connected to the VCR's grounded cabinet—should not be used.)

Complete the modification by soldering the wire pair to the phone jack. Then, replace the VCR's cover. At that point, the VCR should be tested for normal operation. Check the modification for a short-circuit if the VCR doesn't operate correctly.

The controller is assembled on a PC board, for which a full-scale template is provided in PC service.

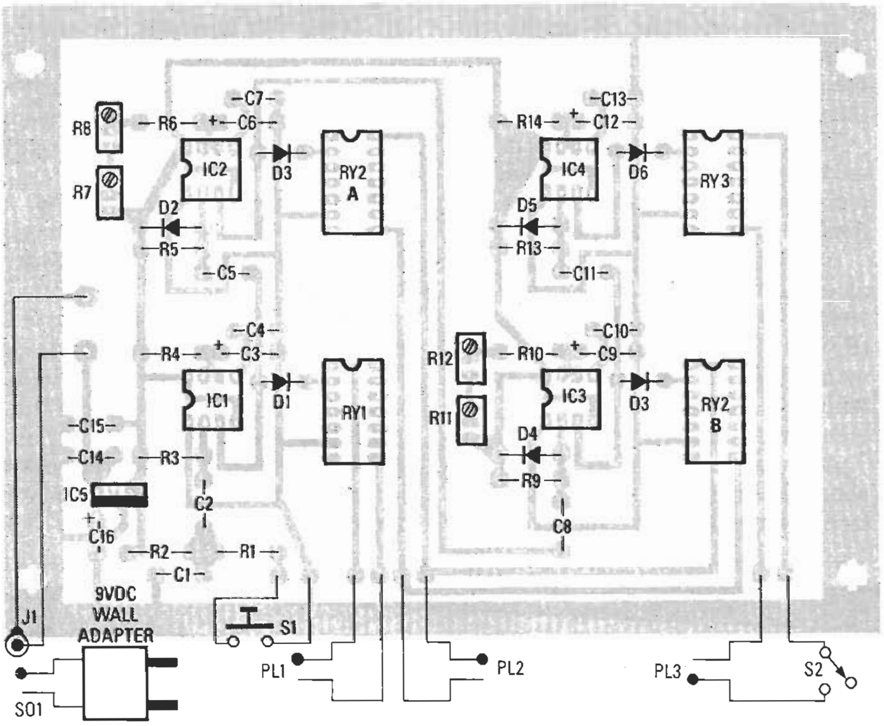


FIG. 2—THE CONTROLLER'S PARTS LAYOUT. Select only one location for RY2; the other remains empty.

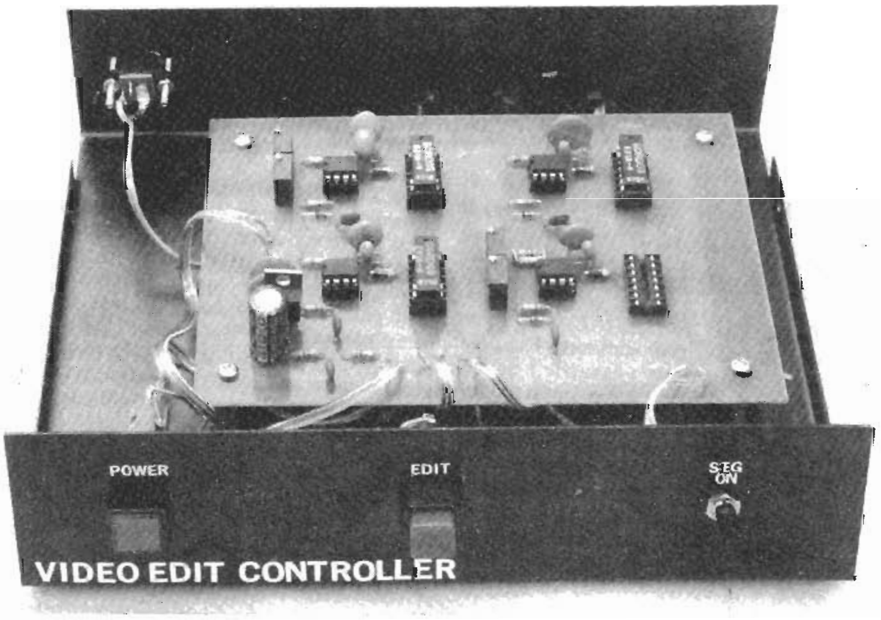


FIG. 3—THE PRINTED-CIRCUIT BOARD is mounted in the cabinet using spacers at each mounting screw. Make certain that there is some kind of wire between the PC board's ground trace and the metal cabinet.

PARTS LIST

All resistors 1/4-watt, 10%, unless specified otherwise.
R1—1000 ohms
R2—1 megohm
R3—10 megohms
R4—470,000 ohms
R5, R9, R13—10,000 ohms
R6—200,000 ohms
R7, R11—250,000 ohms, multiturn potentiometer
R8, R12—10,000 ohm, multiturn potentiometer
R10—47,000 ohms
R14—100,000 ohms
All capacitors rated 10 volts, unless specified otherwise.
C1, C2, C5, C8 ,C11—0.001 μF, disc
C3, C9—10 μF, tantalum
C4, C7, C10, C13—0.01 μF, disc
C6—100 μF, tantalum
C12—1μF tantalum
C14, C15—0.1 μF
C16—1000 μF, 35 volts, electolytic
Semiconductors
IC1-IC4—555, timer
IC5—7808, 8-volt regulator
D1, D3, D6—1N4002, silicon rectifier
D2, D4, D5—1N914, rectifier
Other components
J1—male power-supply mini-jack to match SO1
PL1, PL2, PL3—miniature phone plugs to match VCR equipment
RY1, RY2, RY3—SPDT DIP relay, GORDOS 831A-4
S1—N.O. momentary switch
S2, S3—SPST switch
SO1—power socket, part of 9-volt wall adapter
Miscellaneous
Printed-circuit materials, WA1—9-volt DC wall adapter, DIP sockets, cabinet, wire, solder, etc.

The parts layout is shown in Fig. 2. Notice that there are two locations—labeled A and B—for RY2. If you use DIP sockets for mounting the relay, you will then be able to switch RY2's location easily to conform with the remote-control circuit of the associated VCR.

Figure 3 shows how the prototype's fully assembled PC board looks when it's finished, and also how it is installed in its cabinet.

VCR modification

The controller requires a special, though quite simple, modification to the source-VCR's *review* switch. But be aware that opening the case of the VCR and installing the modification will void the warranty (if it is still in effect).

Remote jack

The recording-VCR or camcorder should have a camera-controlled remote jack. Also, for best results the recorder should also perform a preroll operation prior to initiating the recording action. That feature can often be verified by the recorder's user's manual.

The recording-VCR will run-record when the camera-controlled remote jack is switched by RY2's contacts. The location of RY2 is determined by the requirements of the remote jack. If recording is started by opening a contact, RY2 should be installed in the OPTION A location, which is controlled by IC2. If recording is started by closing a contact, RY2 should be installed in the OPTION B location, which is controlled by IC3.

Calibration

The only items required for calibration are two prerecorded tapes. One is a *source* tape, which contains a clean transition of scenes. The tape can easily be made by making an off-the-air recording of about five minute of program up to a commercial, the commercial, and then five minutes of program. The commercial is only needed so that you can easily recognize a scene transition—from program to commercial and vice versa.

The other tape is the recording tape. It should be pre-recorded with about five minutes of programming.

Connect PL1 to the *review jack* that was added to the source-VCR. Connect PL2 to the recording VCR's camera-controlled remote jack.

Roll the source tape, locate the start of the commercial as closely as possible, and place the source recorder into the *pause* mode.

Then play the second tape in the recording VCR. Locate the end of the recording, set the recorder to *pause*, then activate the record function.

Set the *coarse* adjustment associated with RY2 (R7 or R11) to its smallest value and the *fine* adjustment (R8 or R12) to the center of its adjustment. Press S1. Each of the recorders will do its thing—controlled by the video-edit controller.

After the recording VCR runs for about 30 seconds, stop and rewind its tape to the point where the recording was inserted and press the *pause* button. Then release the *pause* button and time the playing time from the source-tape's entry point until the source-tape's commercial appears.

Using the equation given earlier, calculate the combined resistance value of R7 and R8 (or R11 and R12) that is needed to eliminate the pre-commercial timing. Set the *coarse* adjustment to that value.

Repeat the procedure until the editing controller correctly locates the edit point within about one-half second. At that point, the procedure should be repeated once more, using the *fine* adjustment, until the edit point is "on the nose."

That completes the calibration. A similar method is used to calibrate the switch-in of a special-effects generator via PL3.

Now you're ready to edit some video tapes, and it may take a few tries to become familiar with the system. However, in no time at all, you'll be getting rid of unwanted commercials, splicing together your favorite movie scenes, or removing scenes that you don't want your kids to see. **R-E**

DRAWING BOARD

ROBERT GROSSBLATT

The world of video

BEFORE WE GET INTO CIRCUITS AND stuff, there's a piece of follow-up business I have to take care of. If you're a regular reader, you'll remember that we ran a contest some months ago to see who could come up with a good way to use the space left empty in the EPROM character generator we designed.

There were lots of entries and lots of ideas, but the best one came from Randall Logan of British Columbia, Canada. He used the remaining space in the EPROM to build an automatic hex to decimal converter that still left enough room in the EPROM for the character codes we worked out as our circuit was being developed. The circuit details are on page 72 of the September 1989 issue of **Radio-Electronics**.

Since it's the height of tedium to generate the code for the EPROM, Randall wrote a BASIC program to create the data. He sent me a copy of it and, as promised, I'm sharing it with all of you—it's shown in Listing 1. Although the program is written for an IBM or compatible, it won't take a lot of work to convert it to the BASIC dialect used by other computers. The program is well-commented so you shouldn't have any trouble figuring out how it works and what you'd have to do to change it.

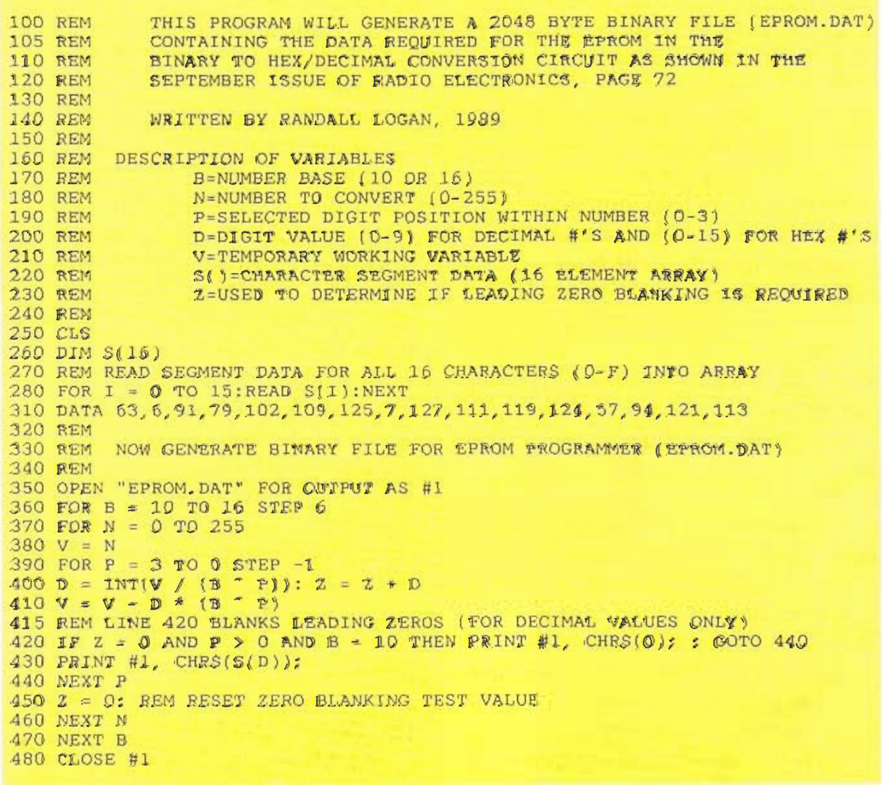
Just about the only reason for changing it would be to match the data outputs of the EPROM to your seven-segment display. The segment assignments (a, b, c, etc.) are pretty much standardized, but sometimes it's easier to lay out a PC board by switching the order of the EPROM data pins. In any event, you've got the circuit and now you've got a program to generate the data for the EPROM.

As a last note on the subject, the program will create a file 2049 bytes long. The extra byte comes about because BASIC (at least IBM BASIC) adds an "end of file mark" (1Ah), whenever it writes a data file. It isn't a big deal since the first 2048 bytes are the ones going into the EPROM, but being aware of the extra byte (as well as knowing why it's there) means there's just one less thing to worry about.

And now for something completely different.

In going through the mail I've received recently, it's evident that video is a really popular subject. Lots of readers are fascinated by the hardware but haven't gotten their hands dirty because they're not familiar with video signals, terminology, and hardware requirements. Over the next few months we'll go through some video basics together. We'll take apart the NTSC standard and examine the individual parts of the waveform. Once we know what we're looking at, we'll see what has to be done to design video circuitry. It's

LISTING 1



```
100 REM      THIS PROGRAM WILL GENERATE A 2048 BYTE BINARY FILE (EPROM.DAT)
105 REM      CONTAINING THE DATA REQUIRED FOR THE EPROM IN THE
110 REM      BINARY TO HEX/DECIMAL CONVERSION CIRCUIT AS SHOWN IN THE
120 REM      SEPTEMBER ISSUE OF RADIO ELECTRONICS, PAGE 72
130 REM
140 REM      WRITTEN BY RANDALL LOGAN, 1989
150 REM
160 REM  DESCRIPTION OF VARIABLES
170 REM          B=NUMBER BASE (10 OR 16)
180 REM          N=NUMBER TO CONVERT (0-255)
190 REM          P=SELECTED DIGIT POSITION WITHIN NUMBER (0-3)
200 REM          D=DIGIT VALUE (0-9) FOR DECIMAL #'S AND (0-15) FOR HEX #'S
210 REM          V=TEMPORARY WORKING VARIABLE
220 REM          S( )=CHARACTER SEGMENT DATA (16 ELEMENT ARRAY)
230 REM          Z=USED TO DETERMINE IF LEADING ZERO BLANKING IS REQUIRED
240 REM
250 CLS
260 DIM S(16)
270 REM READ SEGMENT DATA FOR ALL 16 CHARACTERS (0-F) INTO ARRAY
280 FOR I = 0 TO 15:READ S(I):NEXT
310 DATA 63,6,91,79,102,109,125,7,127,111,119,124,57,94,121,113
320 REM
330 REM  NOW GENERATE BINARY FILE FOR EPROM PROGRAMMER (EPROM.DAT)
340 REM
350 OPEN "EPROM.DAT" FOR OUTPUT AS #1
360 FOR B = 10 TO 16 STEP 6
370 FOR N = 0 TO 255
380 V = N
390 FOR P = 3 TO 0 STEP -1
400 D = INT(V / (B ^ P)): Z = Z + D
410 V = V - D * (B ^ P)
415 REM LINE 420 BLANKS LEADING ZEROS (FOR DECIMAL VALUES ONLY)
420 IF Z = 0 AND P > 0 AND B = 10 THEN PRINT #1, CHR$(0); : GOTO 440
430 PRINT #1, CHR$(S(D));
440 NEXT P
450 Z = 0: REM RESET ZERO BLANKING TEST VALUE
460 NEXT N
470 NEXT B
480 CLOSE #1
```

complicated enough to be interesting and just hairy enough to be fun.

Who could ask for more?

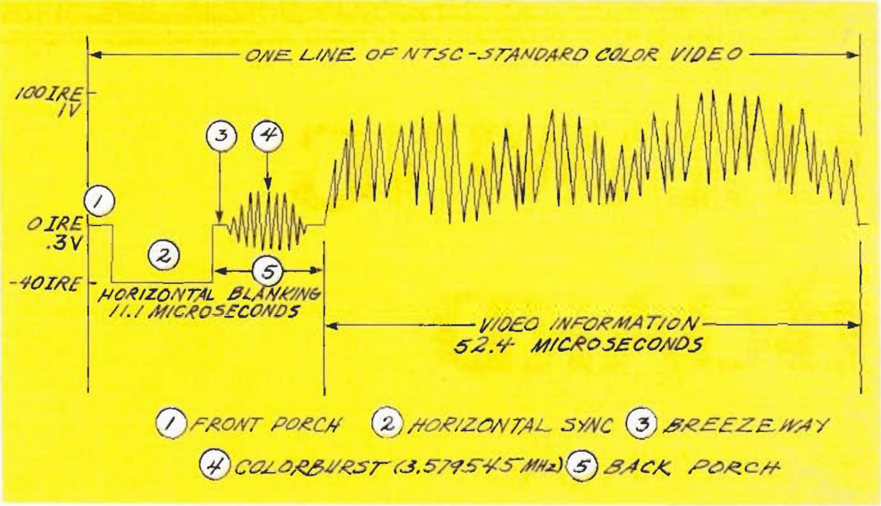


FIG. 1

Some video basics

Back in 1953, a group of relatively unknown people sat down together and made some decisions that affected the lives of every person in this country (and some other countries as well). That sounds like the beginning of a trashy spy novel, but it's not—it's the beginning of the television standard. The people who made the decision were known as the *N*ational *T*elevision *S*ystem Committee, and they established the color-TV standard known as NTSC. There are lots of jokes about what the initials stand for (Never The Same Color, etc.), but whatever you might think of it, NTSC video is (in the United States) the name of the game.

All the video circuitry in a television set is designed with one job in mind—to control the movement of the electron beam that scans across the inside of the picture tube. The video signal that moves the dot on the screen and determines how bright it will be is one of the most complex signals in common use. The key to knowing how video works and making sense out of video hardware is understanding all the component parts of the video waveform.

All video, either broadcast NTSC or high-resolution computer, is made up of the same basic elements. As a simplified explanation, the electron beam in the picture tube scans across the screen from left to right and back again. When it reaches the bottom right of the screen, it moves to the upper left and starts another trip back and forth across the screen. The rate at which it travels left to right, and how often it moves back to the top right of the screen are all defined in the NTSC standard.

The video signal fed to a TV set has two basic jobs. The first is to control the beam's movement, and the second is to control the brightness of the beam. The former tells the TV where to put the picture and the latter tells the TV what picture to put there.

Even though the video circuitry in a television set is constantly moving the beam around the screen from left to right and top to bottom, there's no way to guarantee that the movement is going to be at the precise rates required by the NTSC standard. As we all know, nothing is forever; components age, values drift, and it wouldn't be long before a TV—no matter how well made—would go out of alignment.

Most television sets move the dot around the screen at rates that are close to the NTSC standard, but they have built-in slop factors. That means that there's enough latitude in the TV's control circuitry to enable it to be adjusted to strict NTSC timing. Older television sets had horizontal and vertical hold controls, but most of the newer sets do it automatically.

The horizontal and vertical control of the television image is the job of two components of the NTSC signal known, appropriately enough, as horizontal sync and vertical sync. The former tells the electron beam to move back (or flyback) to the right edge of the screen and the latter causes the beam to move back to the upper right hand corner.

Before we look at the big picture (sorry about that), let's take a look at the most basic unit of the video signal—a horizontal line. The NTSC standard calls for 525 lines of video for each *frame* (the entire picture), and there are two *fields* per frame.

The horizontal line

The whole video signal is really a series of repeated horizontal lines of control and picture information. The NTSC standard allows a total time of about 63.5 microseconds for each line, so a bit of arithmetic tells us that the frequency is about 15.75 kHz. I'm using approximate numbers because the exact value is different for color and black and white. That rate is referred to as the "horizontal scan frequency," and during each cycle the video signal is controlling the placement of the image and carrying the image information, as well.

The illustration in Fig. 1 shows you exactly what's happening in each horizontal line. The left-hand side of the picture corresponds to the trailing edge of the horizontal sync pulse, which is labeled as "2" in the drawing. As you can see, about 20% of the line time is devoted to control signals, and each part of that section has it's own name and allowable duration.

Most television sets "overscan" the image. That means that a bit of the left and right edges of each line take place outside the screen area—think of it as slightly zooming in on the line. That's done for a variety of reasons; most important, it makes sure that the only part of the image you'll see on the tube is picture information—the control data is always off screen.

If you're looking at a color image, the first important piece of information sent on each line is the colorburst signal. That is about 8 to 10 cycles of a 3.579545-MHz clock, and it's used as a standard to

continued on page 91

DRAWING BOARD

continued from page 80

determine both the hue and saturation of the colors displayed on the TV. The two elements, hue and saturation, are usually referred to as "chrominance" or "chroma."

A separate colorburst signal is sent for each line of video and they're all exactly in phase with each other. That's obvious if you realize that inconsistencies would result in color shifts from line to line. (That might make for some interesting pictures, but it's not the kind of stuff sponsors shell out bucks to transmit.)

The picture information immediately follows the back porch and, for the next 52 microseconds or so, the electron beam travels across the screen painting a line of varying color and brightness. The color you see depends on the relative phase difference between the video signal and the colorburst signal in the back porch. The brightness, referred to as the "luminance," is simply a function of the voltage level.

If you look at the voltage scale on Fig. 1, you can see how the voltage varies during a line of video. NTSC video calls for a peak-to-peak signal voltage of 1 volt but, instead of volts, the video signal is usually referred to in IEEE (*I*nstitute of *E*lectrical and *E*lectronic *E*ngineers), or IRE (*I*nstitute of *R*adio *E*ngineers) units. That scale divides the 1-volt range into 140 parts with zero-IRE corresponding to about 0.3 volts. The full scale goes from −40 IRE to +100 IRE units. When you hear a person refer to 50 units of video, they're using the IRE scale.

The reason things are done like that is that there has to be some way to distinguish between control signals and picture information. The NTSC solution was to reserve the area from 0 to 0.3 volts for the control signals and the rest of the space (0.3 to 1 volt) for picture. If the luminance is at 0.3 volts or 0 IRE, the picture will be black—100 IRE will give you white. Everything in the middle will be seen as varying shades of gray or color. The area below 0 IRE is usually referred to as "blacker than black."

Once the end of the picture information is reached, the electron beam is turned off for about 1.5 microseconds (the front porch), and then a horizontal sync pulse is sent. The location and duration of the pulse (4.7 microseconds) is critical to proper placement of the horizontal line on the screen since that causes the electron to move back to the right side of the screen and start a new line.

Messing around with the horizontal sync is a popular activity with the people who dream up ways to scramble TV pictures. If the pulse isn't there at exactly the right time, the television set has no way to identify the start of a new line of video. As a result, the TV will do horizontal retraces at somewhat random intervals and instead of watching a recently released movie, you'll be looking at some images that really belong on a *variable interossiter* (and how many of you know what that is?).

When we get together next time, we'll talk about vertical sync and start to design some video hardware. It should be fun. **R-E**

DRAWING BOARD

ROBERT GROSSBLATT,
CIRCUITS EDITOR

A crystal-controlled video timing generator.

WHEN YOU START FOOLING AROUND with video circuitry, there are three things you always have to keep in mind: timing, timing, and, above all else, timing. The key to designing video circuits is making sure that the right voltages show up at the right time. The values can be slightly off, but if they don't show up exactly when they're supposed to, they might just as well not show up at all.

Since video is fairly complex, everyone has his or her own ideas about the best way to get into it. Theory is important; after all, the video waveform is an agreed-upon standard, and you have to understand its component parts before you can design hardware to produce it. But the best way to learn is to do, and the easiest way to pick up theory is by having it demonstrated; so let's see what's involved in designing basic hardware.

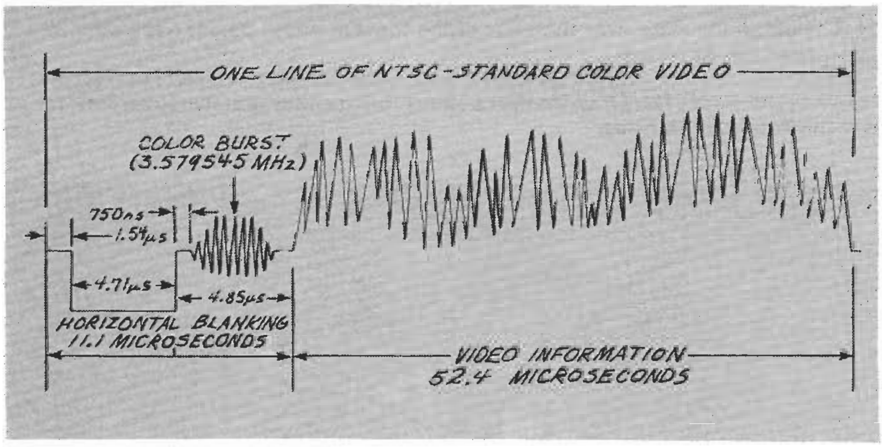


FIG. 1

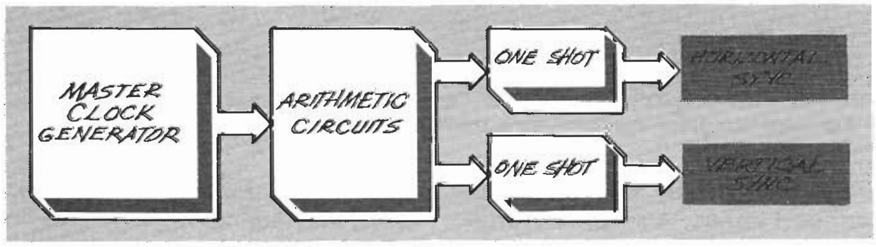


FIG. 2

Hardware basics

The waveform in Fig. 1 is one line of National Television Standards Committee (NTSC) standard horizontal color video. We've already discussed its component parts, but this time I've included all the timing. There are several basic frequencies to be generated, and the pulse width is strictly defined. National Semiconductor, and other manufacturers, make IC's that can take a basic clock frequency in at one end, and put out all needed video frequencies at the other end. They are good chips for designing real-world circuitry, and we'll look at some later; but first let's re-invent the wheel so we know how it works.

Designing a video sync generator is an exercise in pure logic and arithmetic. We need clock generators to control production of the needed frequencies, and one-shots to produce pulses of the right width. Since the most basic signal needed is horizontal sync, that's the place to start.

The block diagram in Fig. 2 gives a good overview of the circuit to be designed. There's a master clock generator, whose frequency is divided down to produce horizontal and vertical sync clocks. They will trigger pulse generators to output the actual sync pulses. By deriving everything from a master clock, the generated sync signals maintain constant timing with regard to one another, an absolute must for video. The sync-generator accuracy will depend completely on that of the master clock, so we have to use a crystal-based circuit. That creates certain problems, but it's the best way to go.

The master clock

Crystal oscillators used to be exotic and expensive, but are now cheap and easy to build. The issue now is what frequency to use, and what chip will produce it. Everyone has his or her own preferences, but one that's often overlooked is the 8284 shown in Fig. 3, in an appropriate oscillator circuit. It was originally designed by Intel as the basic clock for the 808X microprocessors, and any 808X-based computer will have one on the motherboard. Its frequencies and duty cycles are really geared to Intel's microprocessors, but it's a handy general-purpose clock generator, also.

Most of the 8284 control pins like $\overline{\text{AEN1}}$ (pin 3), $\overline{\text{AEN2}}$ (pin 7), RDY1 (pin 4), RDY2 (pin 6), READY (pin 5), and CSYNC (pin 1) are used only when you're using the chip in a computer; we won't use them here. They have to be tied either high or low to make the chip work for our purposes.

All we want the 8284 to do is to act as a stable oscillator. That's easy to set up, and the 8284 has several free extras. The pins used here are crystal inputs X1 (pin 17) and X2 (pin 16), and clock outputs OSC (pin 12), CLK (pin 8), and PCLK (pin 2).

In Fig. 3, the 8284 takes the crystal and provides three different output clocks. The OSC output (pin 12) is a buffered version of the crystal frequency, the CLK output (pin 8) is ⅓ the crystal frequency at a 33% duty cycle, and PCLK (pin 2) is half the CLK frequency with a 50% duty cycle. The CLK output has an unusual duty cycle for use with an Intel microprocessor.

Also, note the F/C (*F*requency or *C*rystal) input on pin 13. Since the chip is a collection of flip-flops and buffers, it needs an input frequency divided down internally to provide the output clocks. The state of F/C determines the origin of the input clock. If F/C is tied low, the 8284 will look at its internal oscillator, the frequency of which depends on the crystal hanging off inputs X1 and X2. If F/C is high, the 8284 will look at the clock being fed to EFI (*E*xternal *F*requency *I*n) input at pin 14.

That means that you can change the output clocks simply by changing the logic level on F/C (pin 13). Most two-speed IBM-XT clones use that feature to switch from "normal" to "turbo" speed. By trapping a scan code from the keyboard and using it to toggle a flip-flop, they change the level of F/C, and switch the master clock speed of the microprocessor, giving a two-speed computer.

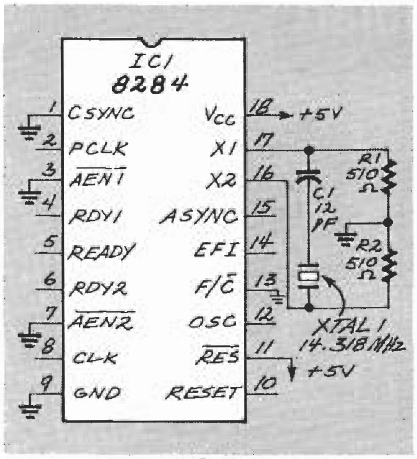


FIG. 3

Now that we have a circuit to produce our clock, we have to decide on what clock to produce. It would be really nice if all the frequencies needed were derivable from one easily available crystal frequency, but they're not. There's just no easy number to provide everything we need. For reasons that'll become apparent when we are farther along in our design, 14.318 MHz is a good choice. One obvious reason is that division-by-four will give 3.579545 MHz, the colorburst frequency. Getting a horizontal frequency of 15734 kHz, however, will take a little more work than that.

The 12-pF capacitor in series with the crystal helps start the 8284's internal oscillator. Since it's similar to an amplifier, the resistance of the crystal network has to be kept as low as possible. If it's too high, the gain drops; and if it's too low, it won't oscillate. The two resistors reduce the effects of stray board capacitance and voltage fluctuations on the frequency. A breadboard is a good choice for construction, but there's considerable capacitance between rows. Since the crystal frequency is pretty high, that stray capacitance can wreak havoc, so the two resistors will keep the oscillator frequency fairly stable.

If your oscillator frequency is outside the 14.3–14.4 MHz range, there's a problem on the breadboard. The easiest way to fix it is to accept the breadboard-induced error as unavoidable, and correct the frequency by varying the capacitor value over 4.7–47 pF. If you're really ambitious, replace it with a small trimmer capacitor. **R-E**

DRAWING BOARD

Let's plunge deeper into video

ROBERT GROSSBLATT,
CIRCUITS EDITOR

VIDEO TIMING RELATIONSHIPS

COLORBURST = MASTER FREQUENCY = 3.579545 MHz

$$\text{HORIZONTAL} = \frac{2 \text{ X } (\text{COLORBURST})}{455} = 15{,}734.26 \text{ Hz}$$

$$\text{VERTICAL} = \frac{2 \text{ X } (\text{HORIZONTAL})}{525} = 59.94 \text{ Hz}$$

FIG. 1

THE MORE YOU START MESSING around with video, the more jokes you hear about the NTSC standard...Never Twice the Same Color, and so on ad nauseum. Now, I'm the first to admit that it's far from being ideal, but let's also remember that the National Television Systems Committee designed the color standard more than thirty years ago and, when they did it, they had to make sure it was compatible with the black-and-white standard that came before it.

If you're old enough to remember the introduction of color TV, you might also remember some of the schemes that were proposed. If you do, you should also realize that the guys who came up with NTSC did a pretty good job. One alternative I remember had something to do with a wheel of colored gels that revolved in front of the camera. It was a mechanical nightmare and, fortunately, I've forgotten the rest of it. If any of you know more about that, or any of the other early color proposals, drop me a note and I'll pass them on to everyone.

In any event, however crazy the NTSC system seems, it's a lot better than any of the others that were proposed at the time, and no matter what else you have to say about it, it works.

The hallmark of the NTSC standard is that all of the individual signals that make up the complete waveform can be derived from one master clock—everything is locked to everything else. Before we get to the arithmetic, however, let's review some video basics and define some terms.

Fields, frames, and interlace

The NTSC standard picture we all watch and love is produced by having the electron beam paint a series of successive lines of video on the face of the TV tube. The electron beam in the tube has to make two passes in order to paint a complete picture, called a "frame," on the screen. The first pass puts out the odd-numbered lines of video and the second pass puts out the even ones. Each of those half images is called a "field." The reason for the system, called "interlace," is to reduce the amount of flicker on the screen. If all the lines were sequentially painted, the top would begin to fade before the screen was completely "painted."

The vertical scan rate is the amount of time it takes the beam to paint each field (half a frame) of video. In the black-and-white days, the vertical rate was set to match the 60-Hz power-line frequency to minimize interference on the screen. Since each field took 1/60th of a second, a full frame took twice that—or 1/30th of a second. Knowing that, and remembering that each frame of video is made up of 525 horizontal lines, we can begin to understand where the TV-signal frequencies come from.

Since it takes 1/30th of a second to paint 525 lines, the horizontal scanning frequency is 525 + x30Hz = 15.75 kHz and each line is painted in 1/15.75 kHz = 63.5 + mu sec. Those numbers were slightly modified when the NTSC

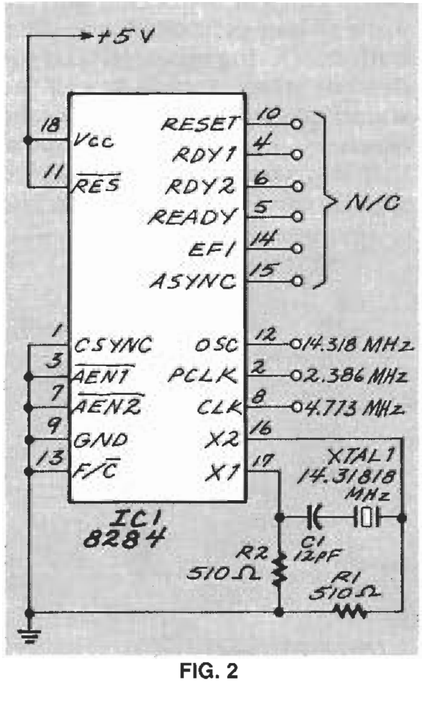


FIG. 2

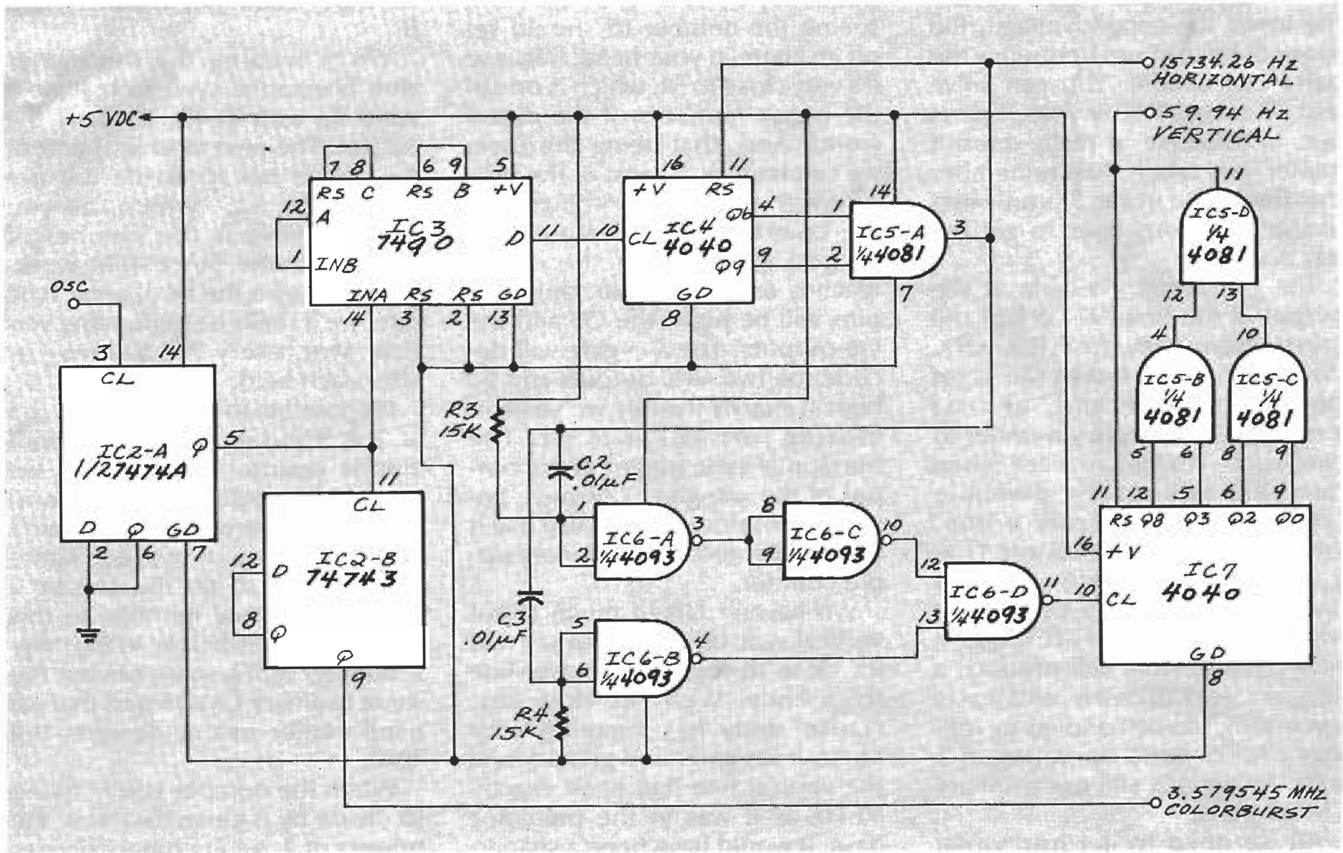


FIG. 3

color standard was developed, but the simple exercise we just went through should help you understand where the numbers came from in the first place.

The master clock for the NTSC color standard was set to be 3.579545 MHz, and all the other signals are derived from it. While the older standard used the power-line frequency and number of scan lines as the basis for the TV picture, the 1953 color standard is built around the burst frequency and a strict arithmatical relationship defined all the other frequencies.

The horizontal scan rate was defined to be the colorburst frequency multiplied by 2/455 (not exactly obvious), and when you do the arithmetic you'll wind up with 15,734.26 Hz—not as neat as the old 15,750-Hz standard, but close enough to lock the horizontal circuitry in old black-and-white TV sets.

The same sort of "close enough" results show up when you calculate the vertical frequency. Given a horizontal scan rate of 15,734.26 Hz, it follows that each line takes 63.56 microseconds, or just slightly longer than the older standard of 63.50 microseconds. Since the new color standard still contained 525 lines in each frame, the vertical frequency works out to 59.94 Hz. That's close enough to the original 60-Hz to avoid interference and still accommodate black-and-white TV sets.

A lot of you might be wondering what I have in mind when I talk about "older black-and-white TV sets." If you pay a visit to your local TV store, you'll find that it's almost impossible to buy a black-and-white TV set anymore, just as, back in 1953, when the NTSC defined the color standard, it was almost impossible to buy a color TV. Since the majority of existing TV sets were black-and-white, compatibility was the overriding issue. It's not such a big deal now but, back then, when the NBC peacock opened its tail, a lot of viewers were happy to hear the magic words "compatible color."

Generating sync triggers

Now that we've put the historical matters behind us (the important numbers and the relationships between them are summarized in Fig. 1), our job is to develop some hardware. The only circuitry we have working so far is the 8284 clock circuit shown in Fig. 2, and I've indicated the frequencies that it produces.

Getting colorburst from the circuit is easy since the osc output is a buffered version of the crystal. A bit of simple arithmetic tells us that 14.318 MHz divided by four equals the burst frequency of 3.579 MHz. You can see from Fig. 1 that getting horizontal sync from the burst is going to take some multiplication as well as division. Since that's a real pain in the neck, we'll start our search for horizontal sync with one half osc, rather than the burst. That way we can stick with frequency division and make things somewhat easier.

We all have our own preferences but, if you're a regular reader, you know I like to use CMOS whenever I can. In this case, however, we're going to use some TTL to do the first part of the division since the circuit is operating at 5 volts (because of the 8284). CMOS parts can certainly operate at 5 volts, but the lower the supply voltage, the lower the maximum frequency the parts can handle. You can solve that by using 74HC or 74HCT parts but, in this case, it really doesn't matter very much. Just remember that the circuit in Fig. 3 is only one of the numerous ways to get the job done.

The frequency available at the output of the first 7474 is half the crystal frequency, or 7.159 MHz. We have to divide that by 455 to get horizontal sync and, at first glance, 455 is a screwy number to deal with—it's much nicer when numbers are readily divisible rather than apparently prime. Well, 455 can be factored into 13 × 5 × 7, so the first thing we'll do is divide the frequency by 7 using what should be an old TTL friend, a 7490 counter. That will produce a frequency of 1.022 MHz, and that is low enough to be handled by regular CMOS parts operating at 5 volts. So we can still use ordinary CMOS parts.

All we need to get horizontal sync is a way to divide the resulting frequency by 13 × 5, or 65. Now, seeing the number 65 should set off an alarm in your head, because it's very close to 64, which is one of the magic numbers of the digital world. And, that being the case, we can handle the rest of the division with a 4040 CMOS binary ripple counter, and a simple two-legged gate. When the count reaches 65 in the 4040, only two pins will be high; the Q0 and the Q6 outputs. The AND gate will decode the two 4040 outputs and go high at exactly the rate we've been looking for; 15,734.26 Hz. The horizontal sync trigger at the output of the AND gate is positive-going so therefore we can also use it to reset the 4040 CMOS binary ripple counter.

We haven't talked much about vertical sync other than to say that it's close to the 60-Hz power-line frequency. Well, as they say, "close" only has meaning with horseshoes and hand grenades. If the vertical rate had been exactly 60 Hz as it was in the pre-color days, it would have been a snap to generate. Just feed a 5369AA with the burst frequency and you'll get 60 Hz out at the other end. As we saw earlier in the column, however, things aren't that simple since vertical sync has to be derived from the signals we've generated so far and the number of lines on the screen.

The NTSC standard calls for a 525-line image, but don't forget that each image is made of two interlaced fields. That means that we are going to need a vertical-sync signal after each field of video—or after every 262.5 lines of video. That's the relationship shown in Fig. 1, and that's also the design job that we have laid out in front of us.

We don't have any handy frequency available to avoid multiplication this time, so we'll have come up with a way to do it electronically, and that's exactly what we're doing with the NAND gate in Fig. 3. The first two gates, A and B, are set up as simple edge detectors. Gate A responds to the negative-going edge and gate B to the positive-going edge. If you work out the truth table on a sheet of paper, you'll see that the frequency at the output of Gate D is twice the input frequency at Gates A and B.

We're feeding the NAND gates with horizontal sync so that we'll wind up with 31,468.52 Hz at the output. The next (and last) part of the circuit has to divide the frequency by 525, which, as you should know, is the number of lines in a frame. Since we're working with twice the horizontal sync rate, we'll really be generating vertical sync every 262.5 lines—or after each field.

It's possible to factor 525 as 5 × 5 × 7 × 3 and do the division with simple counters but, just as we saw with producing horizontal sync, you have to keep your mind open when you're trying to figure an easy way to do division by a larger than usual number. In this case, 525 is pretty close to the magic number of 512, so we can use the same ordinary CMOS part that we used earlier in our design—the 4040.

When the number you're trying to divide by is close to one of the powers of 2, a long binary divider becomes a good choice. In our case for example, we're only thirteen away from 512, and that means we only have to decode four of the 4040 outputs (525 decimal is 1 0000 1101 in binary). The job becomes even simpler since we still have three spare AND gates in the circuit. As with horizontal sync, the vertical-sync trigger is positive-going, so we're also using it to reset the 4040 binary ripple counter IC.

When you put the circuit together and power it up, you should see colorburst (3.579545 MHz), a horizontal-sync trigger at 15,734.26 Hz, and a vertical-sync trigger at 59.94 Hz. Keep in mind that the last two aren't the final signals. They're only the frequencies we need to trigger the one shots that will output pulses of the proper width to work as horizontal and vertical sync for the video that we'll be generating very shortly.

When we get together next time, we'll design the one-shots and put together a circuit that will actually produce video images. We won't be seeing an image of the Mona Lisa on the screen, but what we see will be recognizable. I promise. **R-E** scope.

So where were we?

Believe it or not we're almost finished with our video circuitry. I'm the first one to admit that it's grown to occupy lots of real estate on the breadboard. If you find that upsetting, remember that in the bad old days before IC's, sync generators like the one we're building took up a lot more room, and cost a whole lot more than a handful of IC's.

There are IC's available that can replace most of the hardware we've been assembling. However, as with most special-purpose IC's, the price you pay for using one dedicated IC in place of some MSI stuff is a loss of circuit flexibility.

Since we're designing the timing generators and one-shots, we can set the pulse widths, delays, and scan frequencies to be anything we want. Admittedly, we're after NTSC, but it wouldn't take a lot of modification to generate PAL, EGA, VGA, or any other type of video we want. Also, building a sync generator can show you a lot more than just how video works—it can also show you how to make video not work.

to put the finishing touches on the hardware.

Making video

Everything we've done so far has been aimed at generating the two sync pulses that are being produced at the outputs of the 4528. Both the horizontal and vertical pulses are needed to control the deflection circuitry in the TV, but they have to be combined into a composite signal in order to be used to make NTSC-compatible video. And we have to make provisions in the circuitry to be able to add some picture information to the signal, as well.

Even though video is usually thought of as an analog signal, the sync component is essentially digital. After all, it's really nothing more than either high or low. So there are several ways we can combine the separate sync signals, such as resistors and diodes in a home-made Mickey Mouse gate arrangement, standard gates, and others. The choice is really yours.

Even though both of our sync signals are being derived from the master clock, they're being generated by separate circuitry using the two halves of a 4528 (**Radio-Electronics**, May 1990). And since we're producing a vertical sync pulse that's three horizontal lines long, the horizontal sync generator is going to keep producing pulses even during the time that the vertical sync pulse is being generated.

In order to avoid potential problems, we can prevent that from happening by putting a low signal on the clear inputs (pins 3 and 13) of the 4528. That prevents the inverted outputs (pins 7 and 9) from going low. (Remember that the sync pulses are active low.)

The simple way to make sure that only one type of sync pulse is generated at any one time is to modify the connections made to the 4528 as shown in Fig. 1. By gating the vertical sync generator with horizontal sync and the horizontal sync generator with vertical sync, there's no possibility of signal conflict. During the period that vertical sync is being produced, the horizontal sync generator is disabled. There's really no reason for us to do the same thing to the vertical sync but it can't hurt anything, so we might as well. If you have a scope, you can try it both ways and see how it works.

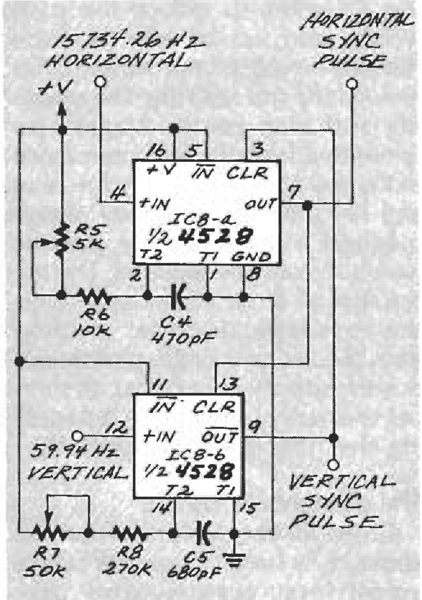


FIG. 1

Since we've eliminated the possibility of having two different sync signals show up at the same time, we can safely produce a *composite* sync signal. For reasons you'll see in just a second, I like to use gates. The requirements aren't very strict since we've made sure that both signals can't be low at the same time.

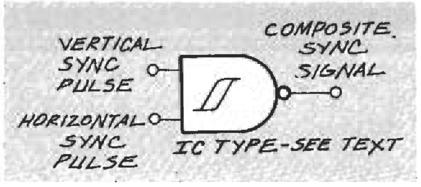


FIG. 2

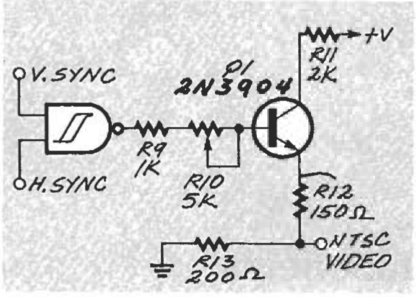


FIG. 3

We want a low to be produced only when either of the sync signals go low—the rest of the time we want a high. You would think we can use a simple AND gate but, as it turns out, it's easier to first produce an inverted version of sync. That's because the inactive sync level (5-volts) has to be at 0 IRE, or about 0.3-volts DC, and the easiest way to translate levels is with a bunch of resistors and a transistor. And, the transistor will invert the signals applied to the base (since it operates as a switch), so we're better off feeding it with an inverted version of sync.

By the way, there's no reason why you can't use the non-inverting outputs of the 4528 and feed those into an AND gate to combine them. I used the inverting outputs because I prefer to have NAND gates on the board. You never know what you'll be adding to the circuit, and inverting gates are more useful.

Even though Fig. 2 uses a 4093 to combine the sync signals, you can use a 4011, or any other plain NAND gate. The 4093, however, is a Schmitt-trigger part and will produce nice, crisp pulses, even if there's a bit of noise at the inputs. Since noise is always a potential problem on solderless breadboards, it's better to be safe than sorry.

All that's left for us to do is design a circuit to translate the digital signals to NTSC standard. Remember that right now our circuit is making a 5-volt swing, and that is slightly beyond the NTSC-standard 1-volt range...to put it mildly.

The circuit shown in Fig. 3 will take the composite sync at the output of the 4093 and cut it down to NTSC levels. You can use the trimmer to fine tune the voltage level at the output. Just remember that the high (inactive) part of the signal should be at 0.3-volts DC to meet the NTSC specs.

Now that we're producing a signal that can be fed into any video input, we can start to play around with it. Try putting video on the screen and seeing what can be done to scramble it.

When we finish this off next month, you'll have a really good idea of how to look at broadcast video. In the meantime, try to get your hands on a scope (if you don't have one already), and take a look at what's fed into the back of your TV set. By the way, most scrambling methods aren't really that complicated, and as soon as you see what's been done, you can figure out what you have to do to fix it. Now that really sounds terrific. **R-E**