BACK-TO-SCHOOL SECTION – DIGITAL TROUBLESHOOTING

$1.25 ■ NOV. 1978

Radio-Electronics®

THE MAGAZINE FOR NEW IDEAS IN ELECTRONICS

DIGITAL WIPER DELAY

Build it and add it to your car. Provides intermittent windshield wiper operation and is programmed automatically. Turn to page 60.

AUDIO BREAKTHROUGH

New metal-particle tape for recording promises ultra-performance in the near future. For the complete story, turn to page 49.

VIDEO TANK GAME

Use your cannon to destroy your opponent first but watch out for the barriers. Construction starts on page 52.

CAPACITANCE METER

Part 2: 4-digit display reads out between 1 pF and 9999 μF. Turn to page 67.

COVER STORY

...ISUAL
...ONE

...ndset
...plete
...one and
...er. To see
...s done,
...o page 37.

PL...

★ D...ts
★ C...er
★ H...
★ S...

A GERNSBACK PUBLICATION

0 70989 34140 11

Figure 5 is the schematic of the speaker amplifier. Coming from the volume control, the telephone line signal is first amplified in a differential pair with an active load. A diode string (D9–D11) and resistor R21 are biased through D8/Q107 at a current of 500 μA. These four elements match the internal base-emitter diodes of Q18/Q20 and Q108/R22, so that the idling current of the output stage Q is also 500 μA. The signal is fed into the diode string through Q17 and is amplified in the output stage. The closed-loop gain is set at approximately 5 by R19 and the input impedance.

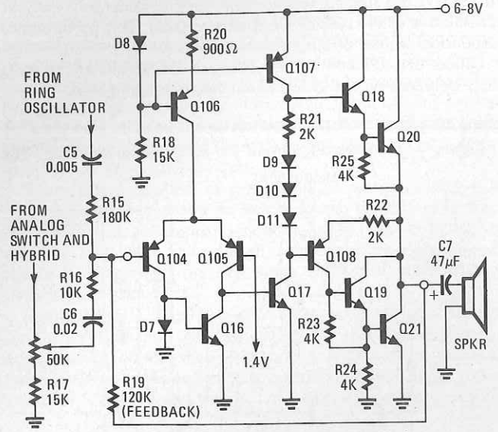


FIG. 5—SPEAKER AMPLIFIER circuit generates 250 mW of power.

Ring detectors and oscillators

The final circuit causes a ringing sound in the loudspeaker that eliminates the need for the bell. As shown in Fig. 6, the ring signal is picked up directly at the telephone line through a combination of Zener diodes, capacitors and resistors. In direct contrast to the voice signal, the ring signal has a very large amplitude that varies between 60 and 150 volts peak-to-peak at 20 Hz.

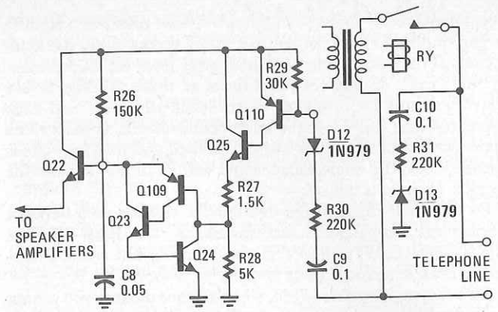


FIG. 6—RING DETECTOR AND OSCILLATOR generates a 1-kHz tone that varies between 60 and 150 volts P-P at a 20-Hz rate.

When the negative peak of the ring signal exceeds the Zener diode breakdown, Q110 and Q25 turn on. This powers up the relaxation oscillator made up of Q23/Q24 and Q109 through voltage divider R27/R28. The frequency of the relaxation oscillator is set by R26 and C8 at 1 kHz. The l-kHz sound signal, which is modulated by 20 Hz, is fed into the speaker amplifier.

One large custom IC contains three of the four circuits discussed plus part of the control circuitry. As a result of using this type of IC, the Tridar speakerphone is smaller than the smallest telephone now offered and and weighs less than the handset of a desk telephone. The custom integrated circuit is making its presence felt in consumer electronics. **R-E**

Oscilloscopes gaining worldwide acceptance in computer field

As the mechanical design, layout and circuitry of oscilloscopes have improved so has their worldwide marketability. Hans Toorens, oscilloscope product manager for Philips Test & Measuring Instruments, Inc., believes that the broader range of applications and functions for an oscilloscope in computer servicing, as well as in other areas, has caused the market to expand both in the U.S. and abroad. He also believes oscilloscopes are destined to become the "workhorse" of the computer-service industry.

Purchasers of test equipment such as oscilloscopes have become increasingly sophisticated in their requirements—they use more precise parameters in judging an oscilloscope, looking for electronic measurement capability, cost reliability and its effectiveness both on the bench and in the field.

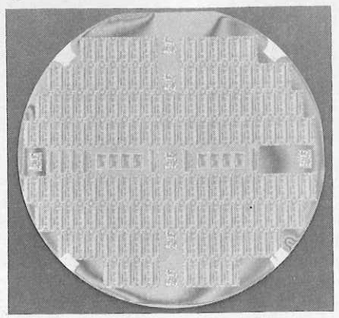
Connecticut judge dismisses radar-detector case

The city of Waterbury, CT, has probably experienced its last arrest for possession of a radar detector. This past June, Judge Norman Buzaid dismissed charges against Dr. Howard Rofsky who had been arrested for possession of a detector. Judge Buzaid termed the 1962 state ruling against radar detectors "ridiculous."

Most Connecticut courts have been reluctant to prosecute such cases, and Dr. Rofsky's case was only one of several lesser-known incidents. However, the Connecticut decision came only days after the Supreme Court of the State of Virginia struck down a portion of that state's anti-radar law, declaring that a presumption of guilt in the statute was unconstitutional. The law will still be on the books, but police will now have a hard time proving the detector in the car was being used illegally. Virginia is the only state in the country to have a law against radar detectors.

The charge-coupled device, an idea come of age

In 1969 Bell Labs scientists Willard S. Boyle and George E. Smith devised the charge-coupled device (CCD), an invention that since then has inspired a great many technical papers but has now finally come of age.



CCD WAFER has a 3-inch diameter and contains 169 CCD devices.

Scientists Boyle and Smith arrived at the idea for the CCD almost accidentally. They were looking for a semiconductor device that could control and handle data similar to magnetic bubble devices. (Bubbles store data in a very small space, making possible high-density data storage.) While at that time bubbles were fairly new devices, the semiconductor was well-known; the resulting invention was the CCD—a result of applying the principles of bubble devices to a semiconductor.

The CCD has three layers—one layer made of metallic electrodes, another of silicon crystal, and an insulating layer of silicon dioxide. Standard MOS processing techniques can be used to manufacture a CCD, which can then be used to perform many electronic functions now performed by larger, more complicated IC's.

Negatively charged electrons in the CCD can be moved about. The information (in the form of electrical charge packets) is stored in small areas or "wells" on the surface. Applying the desired voltage results in data transfer from one well to another. Since the amount of charge in a well can be varied continuously, this makes the CCD an analog device that can be used in analog communications transmissions. When the charge packets are digitized (i.e., the wells are either empty or full), the CCD can be used as a digital memory. If the charge packets are introduced *optically*, instead of electronically, with an image focused on the silicon surface, an imaging device results. It can then be used as an image sensor in an all solid-state television camera. **R-E**

How To Troubleshoot Digital Circuits

Isolating a fault in a digital circuit can be both difficult and time consuming if you don't know how to go about it. Here's an in-depth look at the faults, the right troubleshooting procedures and the test equipment to make it easier.

L. STEVEN CHEAIRS

MOST PROTOTYPE INSTRUMENTS HAVE A few design or fabrication bugs, and any system can malfunction due to a defective component. Unfortunately, these problems do not generally announce themselves in a way that makes them easy to identify. This is because a defect in the system can affect many other components and produce a number of secondary faults. This article will try to present a method that the average hobbyist can use to debug his own prototype or to repair malfunctioning logic circuits.

A list of digital troubleshooting equipment will be included. For each unit, we will try to outline when and where it should be used, noting its limitations. The reader will be led step-by-step through a few typical problems and we'll show various methods of obtaining the data required to pinpoint the fault. Since most of you only come into contact with TTL, CMOS and MOS devices, only these technologies will be covered.

Test equipment

Traditionally, when the individual components of a circuit were accessible, you could perform relatively simple tests to verify proper operation of each component by using a signal generator, voltmeter, ohmmeter, diode tester, transistor tester, tube tester, or an oscilloscope. All this equipment is defined as traditional troubleshooting equipment.

Now that a circuit element must be viewed as a little "black box," where the components are not accessible, a new troubleshooting philosophy must be adopted. Simply the IC must be tested for a complete circuit operation rather than just for a few characteristics such as capacitance, inductance, resistance and turn-on voltages. Note also that now we must observe complex digital signals, rather than simple circuit characteristics, to determine if the IC is functioning correctly. Complicating the situation further, you must consider a large number of inputs and outputs and compare the relationship of incoming signals to outgoing signals. This requires an intricate knowledge of many complex circuits. However, I am not trying to scare you off but just trying to show why a new variety of test equipment was developed.

Multimeter and oscilloscope

The only two pieces of traditional test equipment that you will find some use for in digital troubleshooting are your *multimeter* and *oscilloscope*. With the multimeter you can check the power-supply voltage of a malfunctioning circuit. An oscilloscope can also find much use, especially if it is a dual-trace unit with triggered sweep. However, the older scopes will be next to useless because both time and frequency measurements cannot be made, and the amplitude, the Y-axis, is not generally calibrated, therefore preventing threshold measurements.

When viewing logic levels, absolute amplitudes are unimportant. A digital signal has three states. Only two of these states are used to convey information; they are logic 1 and logic 0 (the third state is designated as undefined). These states are defined by the threshold levels of the logic family; for TTL (*T*ransistor-*T*ransistor *L*ogic) the low threshold is 0.4 volt and the high threshold is 2.4 volts. When the amplitude of the signal is less than 0.4 volt, it is considered to be at a logic 0 level; when it is above 2.4 volts it is at a logic 1 level; and when the amplitude is between 0.4 volt and 2.4 volts, the output is undefined. *Therefore, when you use an oscilloscope, all signals must be checked in order to determine if the data is at a valid logic level!* I have seen technicians spend days troubleshooting a circuit only to discover a problem that had been staring at them all along because they were only looking at waveforms and not logic levels.

Logic clip and probe

Two logic test instruments that almost anyone who works with logic has become familiar with are the *logic clip* and *logic probe*—both overcome the oscilloscope problem. When observing static or low-frequency data, the logic clip and logic probe function quite well; many units have a pulse stretcher and/or memory to enable short pulses to be observed. The clip is simply piggybacked onto the IC; it will find its own power. If pin-to-pin contact is good, then the LED's will indicate the logic levels (where LED on = logic 1 and LED off = logic 0).

To use the logic probe, connect the two wires that exit from the end opposite the tip to the power supply (the red wire is positive and the black is negative). Some probes only have one indicator light; thus, a lit indicator is a logic 1 and an unlit indicator is a logic 0. Other probes may have two or more indicator lights; for instance, a light for the logic 1 state, another for the logic 0 state, and yet another light connected to a pulse stretch circuit or memory. To use the probe, simply touch a PC card run or the pin of one of the IC's and observe the indicators. The logic probe can detect pulses of short duration that have low repetition rates; these pulses would be difficult for an oscilloscope to detect. The probe should be used in one of two ways: First, you can let the circuit operate at its normal clock rate and monitor only key signal lines. Second, you can slow the system clock down using a low-frequency clock source so the logic changes are slow enough for the probe (or clip) to be useful.

Unfortunately, when you slow the clock down, the circuit will probably react differently—due to propagation delay. Many design problems that occur are due to the delaying of parallel signals; the glitches produced by propagation delays in two or more logic paths; or because of PC track inductance and capacitance (both of these are frequency sensitive).

The solution to these problems was provided by two instruments, the *logic comparator* and the *logic analyzer*. The logic comparator is a small hand-held unit having a cord protruding from one end, a 16-LED display and a cradle for a reference IC. The end of the cord contains an IC clip. The instrument has two modes of operation: First, connecting the IC clip to an IC and observing the LED provides you with a simple logic clip (when no reference IC is used and the mode switch is set to the clip position); second, insert-

ing the known good reference IC and setting the mode switch to the comparator position gives you a tester that will detect operational differences in the two IC's.

Since the reference IC is known to be good, it is assumed that any fault must be due to a bad IC; however, this is not always the case. Problems can occur when say a flip-flop or counter is being tested and the two IC's are not synchronized. The solution here is to use a logic pulser to reset the devices before testing. When testing memories or shift registers, attach the comparator for a long enough period for both IC's to acquire the same data.

The comparator does have some serious problems, however; it will not function properly with IC's that have a TTL input and a non-TTL output. Sometimes it will detect a problem that may not be caused by the test IC but by another IC, a bad clip connection, or a faulty circuit board. The worst problem the comparator has, and it shares this problem with all the other logic tools (with the exception of the current tracer), is that it cannot test any wired-OR/wired-AND connections. Thus, the whole set of open-collector IC's are untestable.

The logic analyzer is probably one of the most useful digital test instruments. Very simply, it monitors one or more signal lines until it observes a preprogrammed qualifier that is user-programmed. After detecting the qualifier, it loads its internal memory at a sample rate that is also user-determined; the rate may be synchronous or asynchronous to the data lines. It is also possible to wait a predetermined number of clock pulses (or amount of time) before or after the qualifier. The data is then displayed on a CRT screen as waveforms or truth tables. The logic analyzer also has difficulties testing wired-OR/wired-AND configurations.

The best solution to the wired-OR/wired-AND problem (as well as shorts and opens in the PC card) is to use a current tracer. This probe can be obtained in two forms. The first resembles a logic probe; one end has power leads and the other has a tip. The tip is not used to make electrical contact as with the logic probe; instead it contains a magnetic sensor used to monitor the field produced by current flow. Also, the probe can only detect low-impedance faults.

To detect a wired-OR/wired-AND fault, simply place the probe near the pull-up resistor and adjust the gain control until the indicator lamp lights up (see Fig. 1). Next, place the probe to each output pin of the open-collector gates—only the defective gate will cause the indicator to light up. Also, when a low-impedance fault exists between two gates, the current tracer and a logic pulser can be used as quick diagnostic tools (the logic pulser should be used on all faults responsible for abnormal current flow if the current flow does not provide enough stimulation to use the current tracer). If a signal line between IC's is shorted to ground, place the logic pulser on the foil trace midway between the IC's. Next, use the current

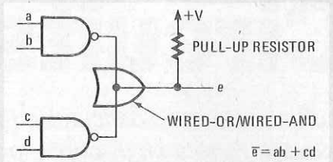


FIG. 1—WIRED-OR/WIRED-AND configuration is formed by connecting the outputs from open-collector gates and connecting a pull-up resistor.

tracer to follow the current to the defective component. A solder bridge may also be detected using the current tracer. Start with the gate that is sourcing the current and follow the current flow until you find the short. Another useful application is when a buffer is driving numerous inputs and one input is shorted to ground. Adjust the current tracer at the output of the buffer and then touch the probe to each input; the one that causes the indicator to light is the defective gate.

Logic pulser

A *logic pulser* is simply a pulse generator that provides either a single-pulse or a continuous-pulse stream. To use a logic pulser, place the tip of the pulser to the input of the suspected logic element. Next, press the trigger button and the logic element will be forced into its opposite logic state. With most conventional signal generators, either circuit components have to be unsoldered and/or the foil trace has to be cut. Most pulsers will provide around 700 mA of current, which is more than sufficient to force the IC to change states. The pulser must generate a very narrow pulse to prevent any IC damage that might be caused by applying too much current. The pulser, as should all other test equipment, should have a high-impedance input to prevent adversely affecting normal circuit operation while using it.

Beyond this collection of equipment all you need is a set of IC test clips and jumper wires. It should also be emphasized that all the equipment described above will be required in any single application. The scope is a general-purpose instrument that can be used in almost any application, whereas the voltmeter may find use only in specific cases—such as checking the power supply. The logic clip should only be used for checking static or low-frequency signals. The logic probe also works best at low-frequency levels, but can be used on higher-frequency circuits if it is used just to monitor key control lines. The IC comparator is useful only for testing a restricted number of the available IC types (because of the need for a reference IC). Both the pulser and current tracer are very valuable troubleshooting tools. The logic analyzer is unquestionably a valuable tool—especially when used with systems that have parallel information, such as synchronous counter strings, address buses, data buses, etc.

IC technologies

To troubleshoot effectively, a knowledge of the individual IC's and the logic

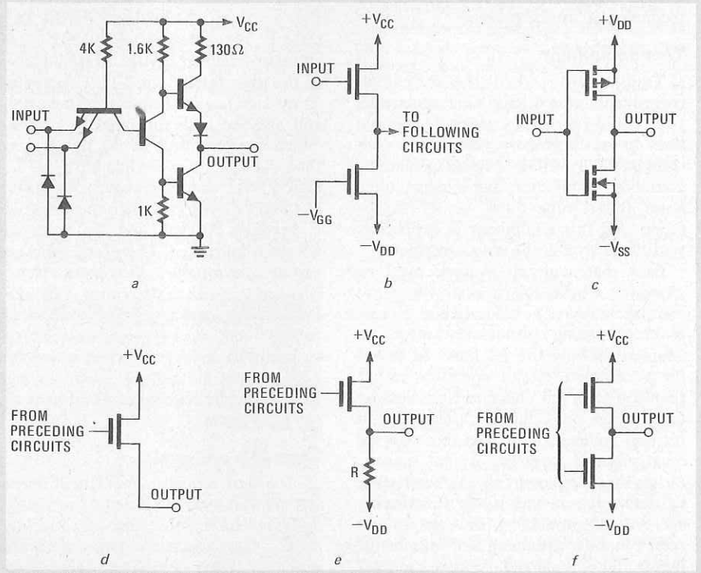


FIG. 2—INPUT AND OUTPUT CIRCUITRY FOR TTL, PMOS, CMOS and MOS logic families. Input circuitry for a 2-input TTL NAND gate is shown in *a*. Input of PMOS gate is shown in *b*. Circuit of a CMOS inverter is shown in *c*. Open-drain MOS output circuit is shown in *d*. Resistor pull-down MOS output circuit is shown in *e*. Push-pull MOS output circuit is shown in *f*.

family associated with them is required. Thus, the troubleshooter must have a library containing the pinouts and functional truth tables for all the common IC's, plus some not so common ones. Also, a firm understanding of the IC's input and output characteristics for the logic family is required. Figure 2 shows the input and output circuitry associated with TTL, CMOS and MOS logic families.

This article will not discuss such design characteristics as propagation delays, set-up times, power supplies, filtering, bypassing, fan-in/fan-out, etc. When discussing specific logic families, only threshold levels and failure modes will be examined. Furthermore, emphasis will be placed upon poor PC layout, with less emphasis on defective components. This is because more than 90% of the problems are a result of design and fabrication failures and less than 10% a result of faulty components. Of course, the exception to this rule is the digital circuit that experiences a logic failure after proving itself in operation for an extended period. In this case, the odds are in favor of a defective component.

I will assume that you have access to IC pinout information and description of all the IC functions. If not, such information is available from many sources. The most common IC's used by hobbyists are of the MOS, CMOS, or TTL technologies. For the MOS types—NMOS and PMOS—you should consult the individual components data sheet for switching-threshold levels. For most MOS/CMOS IC's that are TTL-compatible, the maximum low-level output, V_{OL}, with no noise at the input and assuming only capacitive loading, is given as $V_{OL} = V_{SS} + 0.01$ volts. The minimum high-level output, V_{OH}, for the same conditions, is given as $V_{OH} = V_{DD} - 0.01$ volts. For optimum performance, the high-level input should equal V_{DD} and the low-level should equal V_{SS}. Due to the high DC noise immunity of MOS/CMOS devices, an acceptable logic 1 is $V_{DD} - 0.3(V_{DD})$ and a logic 0 is $V_{SS} + 0.3(V_{DD})$. Therefore, as a worst case, V_H may equal 3.5 volts, and V_L may equal 1.5 volts, using a 5-volt power supply. Also, variations in the power-supply potential, V_{DD}, will directly affect the threshold point.

Transistor-Transistor Logic (TTL or T^2L) can be subdivided into five families—regular, low-power, low-power Schottky, Schottky and high-speed. Table 1 shows threshold-level comparisons of each type. In this table, both the absolute and typical levels are shown; when testing use only the absolute levels. Thus, for standard TTL the low-level threshold is 0.4 volt and the high-level threshold is 2.4 volts (the area between 0.4 volt and 2.4 volts is undefined).

The failure modes for these logic families are similar, although some logic families may be more prone to one particular type of failure. Basically, there are only five kinds of failure: (1) failure of the internal logic; (2) the input/output is shorted to ground; (3) the input/output is shorted to V_{CC}; (4) the input/output is open; and (5) there is a short between two or more of the IC pins.

The first failure, that of internal logic, results in erratic circuit operation. Therefore, the output of the IC will not be predictable. This failure will block normal signal flow and most assuredly has a catastrophic effect upon circuit operation.

The second and third failure modes, the input/output being shorted to ground or to V_{CC}, do not qualify as catastrophic failures since in many cases normal circuit operation continues with only occasional circuit malfunctions. In most cases, however, some expected system operations will not exist and thus can manifest themselves as catastrophic failures. The actual effect of this type of fault is that an input/output will be held high or low, depending upon the failure, and thus prevent normal circuit operation.

The fourth failure, open input/output, generally will be due to a bad internal wire bond between the IC pin and the silicon wafer. When the problem is due to an open input bond, then no signal will be delivered to the silicon wafer. Thus, the input will float at some undefined level. Note also that there will be no effect upon the signal source driving this input. An open bond on an output will cause the IC inputs being driven by that output to float (high in the case of TTL). For both TTL and DTL, a floating input drifts to about 1.4 volts to 1.5 volts. Therefore, an open bond on an output (for TTL or DTL) forces all corresponding inputs to a logic 1 level (even though this is below the valid logic 1 level).

The final failure, a short between two pins, is one of the harder cases to analyze. When two pins are shorted, two outputs that drive those lines are placed in opposition; i.e., they both will attempt to pull the other high or low. These outputs may be on the bad IC (when two outputs are shorted) or external to the IC (when two inputs are shorted), or there may be only one output on the IC (when an output and input are shorted together). In the case where one output attempts to go high and the other tries to go low, excessive current will be drawn through both outputs. The end result is that the short will pull both outputs to a low state. Therefore, when both outputs are driven low or high simultaneously, the circuit will perform normally, but when they are driven into alternate states, the output will go low. Note that since the outputs must dissipate more power under these conditions, at some point a catastrophic failure may occur due to excessive heat buildup.

TABLE 1—THRESHOLD LEVELS for TTL logic families. Values are given in volts.

	Logic Level	7400	74L00	74LS00	74S00	74H00
V_L	Typical	0.2	0.2	0.35	0.35	0.2
	Maximum	0.4	0.4	0.5	0.5	0.4
V_H	Typical	3.4	3.2	3.4	3.4	3.4
	Minimum	2.4	2.4	2.7	2.7	2.4

A troubleshooting example

Whenever you troubleshoot, it is mandatory to narrow the problem down as much as possible by observing the symptoms. By using the controls and displays provided on the front panel of the malfunctioning equipment, isolate the problem to as few circuits as possible. Obtain all the written information available, such as service manuals, operator's manuals and schematics. *Look and think before touching!* Remember you have enough problems now, do not create any more by hasty actions. A well-written service manual can provide information about key signals that may aid in isolating the failure to a single circuit.

When troubleshooting a complex system, such as a computer that contains hundreds of IC's, the first step is to isolate the malfunction to as few IC's as possible. Most of your time will be spent searching for the problem, not fixing it. Next, by using key signals and the proper test equipment, zero in on the faulty IC.

Let's look at a typical circuit containing a few problems. To enhance your skills for both general-purpose repair work and prototype testing, two separate troubleshooting paths will be followed: The first is to assume that the circuit has only one failure at a time and it will be due to a faulty component (this is basically what we see when a proven circuit fails); second, in order to simulate design, layout and fabrication problems, several concurrent problems are shown to exist.

The circuit is shown in Fig. 3 This circuit is one-half of the logic used in building a limited graphic terminal that uses subjective color. Subjective color, unlike standard color, can be displayed on a black-and-white monitor screen—if you stay within its limitations. The circuit board contains the video-output amplifier, sync generation circuits, composite blanking circuit, character generator, subjective color circuits, and the computer I/O circuits.

The schematic (see Fig. 3) shows six circuit failures. The first is an open circuit at the output of the video multiplexer. The symptom observed on the monitor is: No picture, with a white raster being displayed. This immediately

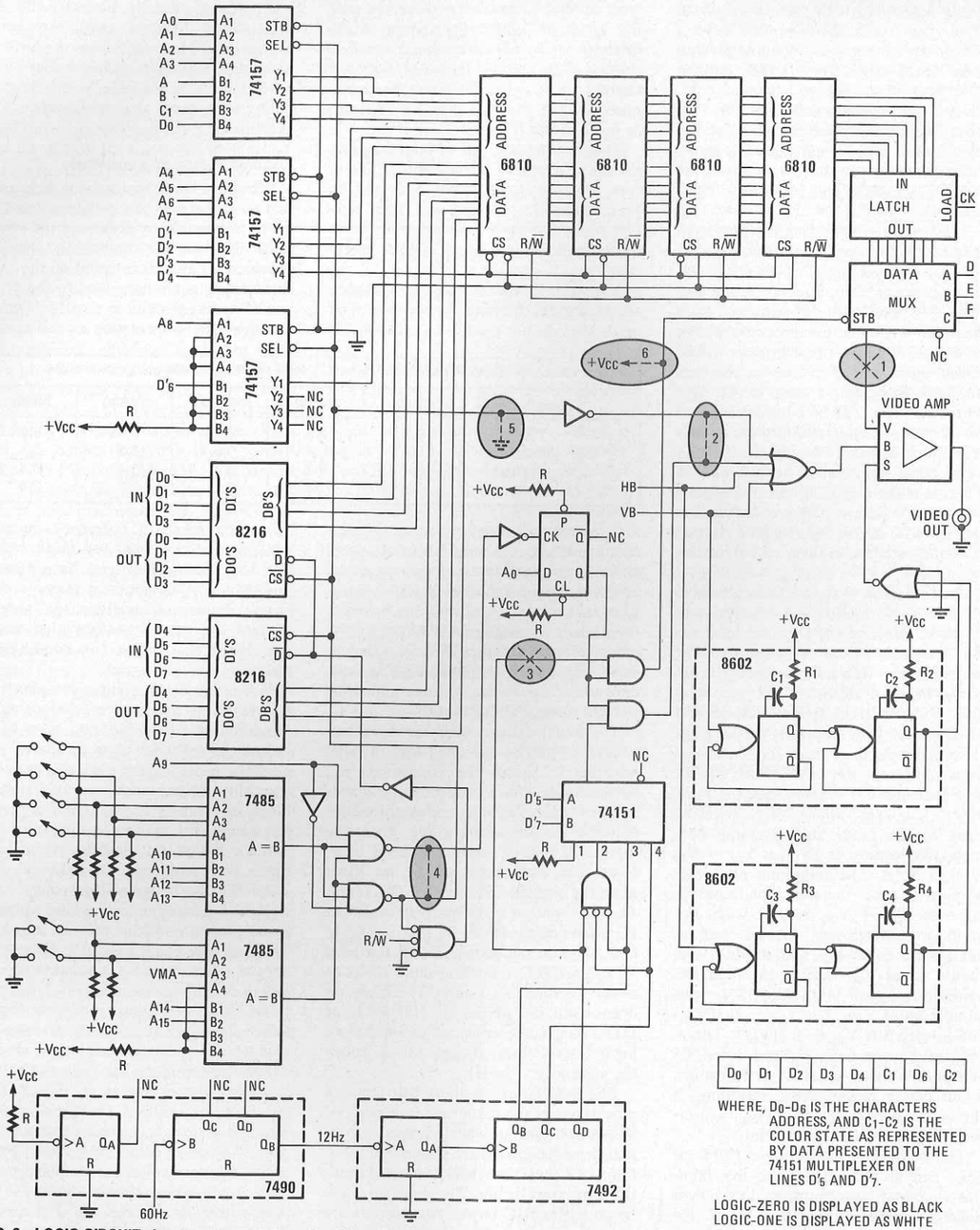


FIG. 3—LOGIC CIRCUIT showing six possible malfunctions.

suggests you should check the signal path between the video amplifier and the multiplexer, because both blanking and sync seem to be working. Since this is a high-frequency signal with only one line, use the oscilloscope. If there is no signal on that line, check the strobe input using a logic probe (this input should be a logic 0 or be checked at a low frequency). Since this input checks OK, next use the logic analyzer or oscilloscope to verify that the 8-bit data lines and 3-bit control lines are active—they should be. After verifying these inputs, the only possible problem could be an open bond inside the multiplexer on the output.

The second fault is between two input pins of the composite summing circuit; this fault causes the color-blanking and computer-access blanking outputs to be tied together. The symptoms of this failure are (1) the color cannot be observed, and (2) erratic images are displayed during computer access. During normal non-color operation when the computer is not loading a new character pattern into the

continued on page 87

CUT ALONG DOTTED LINE

Digital Reference Charts

NUMBER SYSTEM

Our present number system (decimal) uses base 10 to perform all necessary arithmetic operations. Example: $312 = 3 \times 10^2 + 1 \times 10^1 + 2 \times 10^0$. Binary number system uses combinations of 0's and 1's to represent and perform all arithmetic operations that are possible in the decimal system. Binary number system uses base 2:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Table progresses to the left indefinitely as 2^n, where n = 0, 1, 2, 3, 4, . . . To change any binary number to base 10, write binary number underneath table and add decimal equivalent where 1's appear. Example, change 11000100 to decimal:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	1	0	0	0	1	0	0

$2^7 + 2^6 + 2^2 = 128 + 64 + 4 = 196$. The number then is 196.

To change decimal to binary: Divide number by 2 and keep remainder that will always give 1 or 0. Example: 196.

	Remainder
196 ÷ 2 = 98	0
98 ÷ 2 = 49	0
49 ÷ 2 = 24	1
24 ÷ 2 = 12	0
12 ÷ 2 = 6	0
6 ÷ 2 = 3	0
3 ÷ 2 = 1	1
1 ÷ 2 = 0	1

To write binary number from decimal begin at bottom row of remainder table and write the digits from left to right. Example: 196 = 11000100

BINARY WORDS

A bit is a digit in binary language. A digit can be either a 1 or 0. Binary numbers are also known as binary *words*. Consequently a *5 bit* binary number is referred as a *5 bit* word. The longer the word the greater the decimal number that can be represented. Example: A 5 bit word can represent 32 different combinations. A 2 bit word can represent 4 different combinations. By the equation: combinations = 2^n where n is the number of bits in the digital word. Example: Represent the maximum number of combinations with a 2 bit word. Combinations = $2^n = 4$. See table.

Decimal	Binary
0	00
1	01
2	10
3	11

The highest decimal number that can be expressed for a given word is the total number of combinations minus one. Equation: Decimal Number = $2^n - 1$. Example: What is the highest decimal number that can be expressed by a 2 bit word. The total decimal number is $2^n - 1 = 2^2 - 1 = 3$.

ADDITION

Binary addition is manipulated the same as decimal addition but keep in mind that 1 + 1 = 0 and 1 carries over.

Carry 1
Carry 1
Carry 1

```
  1 1 1
  1 1 1
1 0 1 0
```

Addition table	
	Carry
0 + 0 = 0	0
0 + 1 = 1	0
1 + 0 = 1	0
1 + 1 = 0	1

Examples:

```
Add 101011      Add 10000
  + 111001        + 00100
  1100100         10100
```

SUBTRACTION

Reversed procedure from addition remembering to borrow one from the left column.

```
  1001
 - 100
  0101
```

Here borrow one from the left which makes the next number 1 – 1 = 0

Subtraction table	
	Borrow
0 – 0 = 0	0
1 – 0 = 1	0
1 – 1 = 0	0
0 – 1 = 1	1

MULTIPLICATION

Proceed as arithmetic multiplication.

Example:

```
  10011
X   101
  10011
  00000
 10011
1011111
```

TABLE
0 X 0 = 0
1 X 0 = 0
0 X 1 = 0
1 X 1 = 1

DIVISION

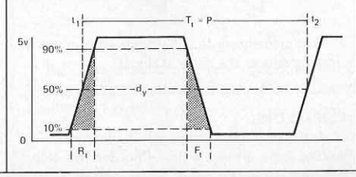
Binary division: proceed as arithmetic division.

```
   1100
10 | 11000
     10
     010
     10
     0000
```

Division Table
0 ÷ 1 = 0
1 ÷ 1 = 0

DIGITAL PULSES

In digital logic, pulses are defined as voltage transitions that occur during a determined lapse of time. Timing pulses are usually generated by a circuit called a clock. A set of digital pulses is known as a train of pulses. The duration and length of the pulses can be changed to fit the needed application. A pulse is an abrupt change in a voltage level.



For the pulse shown in the figure: the maximum amplitude is 5 volts; the reference level is 0 or ground. The total time duration is from t_2 to $t_1 = t_{total}$. If the pulse occurs 1 time in one second, the frequency is one pulse per second. The period = 1/f. The duty cycle = d_y/P.

P = 1/frequency

The left-hand side of the pulse is called the leading edge. The right-hand side of the pulse is called the trailing edge. The width of the dotted area under the leading edge is known as the pulse risetime. The width of the dotted area under the trailing or decay edge is known as the falltime. Digital pulses are fast pulses with very sharp leading edges and very sharp trailing edges. *Positive logic* equates the maximum amplitude of the pulse as the digital number 1 and the ground or zero level as digital logic zero. *Negative logic* equates the maximum amplitude of the pulse as the digital number 0 and the ground or zero level as digital 1.

OCTAL NUMBER SYSTEM

Number system widely used in computer language. The octal system is a simpler way to store and recall values stored in computer memory banks. The binary numbers to be stored are converted to octal saving considerable time and possible errors. The octal number system has a base of 8. The symbol to represent the numbers are: 0, 1, 2, 3, 4, 5, 6 & 7. Notice the absence of the number 8. The conversion from decimal to octal follows the same procedure as the decimal to binary, but the base is 8.

8^4	8^3	8^2	8^1	8^0
4096	512	64	8	1

The table progresses indefinitely to the left as 8^n; where n = 0, 1, 2, 3, 4 . . . .

Convert the octal number 421_8 to decimal.

8^4	8^3	8^2	8^1	8^0
4096	512	64	8	1
		$4X8^2$	$2X8^1$	$1X8^0$
		256	16	1

Add the powers of 8 to get the decimal number. Example: 421 octal = $4 \times 8^2 + 2 \times 8^1 + 1 \times 8^1 = 273_{10}$ (decimal). To convert decimal number to octal, divide decimal number by 8 and keep remainder that will give octal number.

	Remainder
273 ÷ 8 = 34	1
34 ÷ 8 = 2	2
4 ÷ 8 = 0	4

To write octal number begin at the bottom row or last digit of the remainder and write octal number from left to right. Example: 421_8 (octal) = 273_{10} (decimal). The conversion of a binary digit to octal is accomplished by grouping the binary number into groups of 3 digits. Then proceed to evaluate the octal number in each group of 3 digits. Example: Change binary number 111100111110 to octal. Separate from right to left in groups of 3. Example: 111-100-111-110 = 7476_8. Change each group of 3 digits to octal converting binary to decimal. $111 = 7_8$; $100 = 4_8$; $110 = 6_8$.

CUT ALONG DOTTED LINE

BOOLEAN ALGEBRA RULES

▲ symbol indicates a logic variable. This can be either a logic 0 or a logic 1 level.

1	$0 + ▲ = ▲$
2	$1 + ▲ = 1$
3	$▲ + ▲ = ▲$
4	$▲ + \overline{▲} = 1$
5	$0 \cdot ▲ = 0$
6	$1 \cdot ▲ = ▲$
7	$▲ \cdot ▲ = ▲$
8	$▲ \cdot \overline{▲} = 0$
9	$\overline{(\overline{▲})} = ▲$
10	$▲ + Y = Y + ▲$
11	$▲ \cdot Y = Y \cdot ▲$
12	$▲ + (Y + Z) = (▲ + Y) + Z$
13	$▲ (YZ) = (▲Y) Z$
14	$▲ (Y + Z) = ▲Y + ▲Z$
15	$▲ + ▲Z = ▲$
16	$▲ (▲ + Y) = ▲$
17	$(▲ + Y) (▲ + Z) = X + YZ$
18	$▲ + \overline{▲}Y = ▲ + Y$

BOOLEAN SIMPLIFICATION

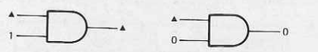
Simplify $P + X + \overline{X}$. Using Rule 4 from above table, $P + X + \overline{X}$ reduced to $P + 1$. Using Rule 2 reduced $P + 1$ to 1. Therefore, $P + \overline{X} + X$ is always equal to a logic 1.

Simplify $PQ\overline{Q}L$. By Rule 8 of above table, $Q \cdot \overline{Q} = 0$. Therefore, $PQ\ \overline{Q}L = P \cdot Q \cdot L$. By Rule 5, $P \cdot O \cdot L = 0$. Therefore, $PQ\overline{Q}L = 0$.

Simplify: $\overline{M} + CS\overline{C} + F$. Use Rule 8: $CS\overline{C}$; $C\overline{C} = 0$. Use Rule 5: $0 \cdot S = 0$. Expression becomes $\overline{M} + 0 + F$, which reduces to $\overline{M} + F$. Note we should not apply Rule 4 of boolean table $\overline{M} + F$ because variables are not identical. The answer is $\overline{M} + F$ or an OR gate with inputs $\overline{M}$ and F.

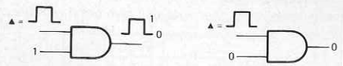
LAW OF PRODUCTS

The law of *Products* is also called the law of *Intersection*. This law explains the behavior of an AND gate. It follows Rules 5 and 6 from the above table.

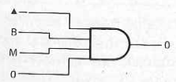


According to Rule 6 ($1 \cdot ▲ = ▲$), if we apply a logic 1 and the variable ▲ to the input of an AND gate, the output will be equal to the variable ▲. According to Rule 5 ($0 \cdot ▲ = 0$), if we set the variable ▲ equal to a binary 1, the output of an AND gate is still 0.

For a pulsed input:

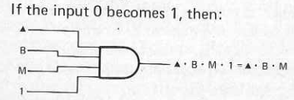


Consequently for a four input AND gate, applying Rule 5 of the boolean angebra table:



$▲ \cdot B \cdot M \cdot 0 = 0$. It is obvious that in the preceding AND gate expression if any of the variables is a logic 1 level, but either of the inputs is zero, the output will be 0.

If the input 0 becomes 1, then:



THE LAW OF UNIONS

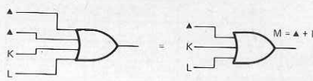
This law pertains to the OR gate and is related to Rule 1 and 2 of the boolean table.

According to Rule 1 ($0 + ▲ = ▲$), if one of the inputs of an OR gate is 0 and we apply the variable ▲ to the other input, the output will be the variable ▲. According to Rule 2 ($1 + ▲ = 1$),

if we apply a 1 and the variable ▲ to the inputs of an OR gate, the output will be 1.

LAW OF TAUTOLOGY

The known *Law of Tautology* applies to Rules 3 and 7 of the boolean table. Rules 3 and 7 apply to AND gates and OR gates. Using this law, simplification of long algebraic expressions becomes simple. The rules merely state that equal variables in an equation should be omitted. Example: Simplify the equation ($M = ▲ + ▲ + K + L$). It is obvious that the variable ▲ repeats twice. By Rule 3, the equation simplifies to $M = ▲ + K + L$.



THE LAW OF COMPLEMENTS

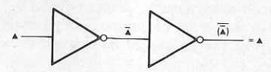
If a logic signal and the complement of this logic signal is applied to a logic gate the resulting output is 1 or 0 depending on the logic gate being used. The law of the complement is stated in Rules 4 and 8 of the boolean table. Let's apply this rule to an OR gate.

According to Rule 4 ($▲ + \overline{▲} = 1$), if one of the inputs of the OR gate is logic ($1 = ▲$) and the other input is ($0 = \overline{▲}$) the output will be 1. Example: Pulsed

According to Rule 8 ($▲ + ▲ = 0$), if one input to an AND gate is variable ▲ and the other input is 0, the output of the gate will be 0.

THE LAW OF DOUBLE NEGATION

The Law of Double Negation is expressed by Rule 9 of the boolean algebra table. This law states that feeding the negation of a variable through an inverter produces the original variable.



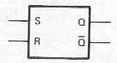
Complementing a signal an even number of times produces the original signal.

FLIP-FLOPS

A digital logic circuit able to memorize by storing logic levels. A flip-flop has two stable states: It will remain in either set or reset state until its state is changed by external signals. The data stored in a flip-flop can be quickly checked by using an oscilloscope or meter to detect the state of its output. There are three basic types of flip-flops.

1 The RS
2 The D type
3 The JK

The logic symbol for an RS flip-flop is

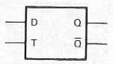


The inputs are S and R. The outputs are Q and $\overline{Q}$. Application of a logic 1 level on the S input will make the Q output go to a logic 1 level and the $\overline{Q}$ output go to a logic 0 level. If the logic 1 level is applied to the R input, the output levels are reversed. The unused input must be held at a logic 0 level.

RS FLIP-FLOP LOGIC TABLE			
INPUTS		OUTPUTS	
R	S	Q	$\overline{Q}$
High	Low	Low	High
Low	High	High	Low
Low	Low	Unchanged	
High	High	Not Permitted	

When the S input is 1 and the R is 0 the flip-flop is reset. When the S is 0 and R is 1, it is set. All the other input combinations produce ambiguous or race states.

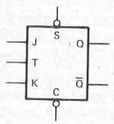
The D-type flip-flop logic symbol.



The D-type flip-flop generally behaves like the RS flip-flop but the main difference is that a low-to-high transition must be applied to the T input for the D flip-flop to toggle and store information.

D-TYPE FLIP-FLOP LOGIC TABLE			
INPUT		OUTPUT	
D	T	Q	$\overline{Q}$
Low	Low	Previous state	
Low	High	Low	High
High	Low	Previous state	
High	High	High	Low

The JK flip-flop symbol

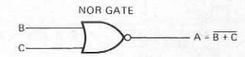


The S and C inputs presets the JK flip-flop to a desired state before another operation is begun. The S and C inputs are referred to as asynchronous inputs because they don't require a transition on the T input. The J and K inputs only affect the Q and $\overline{Q}$ outputs when a transition occurs on the T or clock input. If the J input is 1 and the K input is 1, the flip-flop will reset from the previous state in the presence of a low-to-high transition on the T input. To set the JK flip-flop, apply a 1 to the J input and a 0 to the K input, then apply a low-to-high transition (clock pulse) to the T input. This operation is referred as synchronous with the clock operation.

CUT ALONG DOTTED LINE

THE NOR GATE

A logic circuit with two or more input capable of resolving the equation $A = \overline{C + B}$. The NOR gate is a combination of an OR logic gate followed by an inverter.

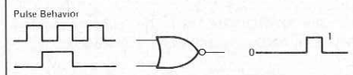


The NOR gate could be constructed using an OR gate followed by an inverter.

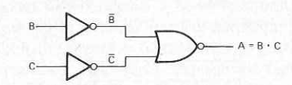
The operation of a *NOR* gate is represented in the following truth table.

Input		Output
B	C	$A = \overline{B + C}$
0	0	1
0	1	0
1	0	0
1	1	0

Notice that the NOR table is the exact opposite or complement of the OR truth table. Summary: The 2 input NOR gate produces an output when both of the inputs are logic 0 level. If either of the inputs is 1, the output is always a logic 0 level.



A NOR gate using inverters in the input will act as an AND gate.



Truth Table for inverted input NOR gate shown above.

B	C	$\overline{B}$	$\overline{C}$	A
1	0	0	1	0
0	1	1	0	0
0	0	1	1	0
1	1	0	0	1

DUALITY OF LOGIC GATES

Gates can provide different functions depending on the assumed reference logic level applied to the input. There are two widely used types of combinational logic levels used in present logic circuits. These are known as *positive logic* and *negative logic*.

POSITIVE LOGIC LEVELS

Input	Output
Logic 1 = +5 volts	+5 volts
Logic 0 = 0 volts to +0.2 volts	0 volts to +0.2

The logic 0 is relatively close to the 0 or ground reference level but in practical gate design the 0 reference is usually a few tenths of a volt above ground level.

NEGATIVE LOGIC LEVELS

Input	Output
Logic 1 = 0 volts to +0.2 volts	0 volts to +0.2 volts
Logic 0 = +5 volts	+5 volts

Truth Table for Positive Logic 2-input AND gate

Voltage Table

Input		Output
B	C	A
0 V	0 V	0 V
0 V	+5 V	0 V
+5 V	0 V	0 V
+5 V	+5 V	+5 V

Truth Table

Input		Output
B	C	A
0	0	0
0	1	0
1	0	0
1	1	1

Compare above Truth Table with Truth Table below for a negative logic 2-input AND gate.

Negative Logic AND gate.

Voltage Table

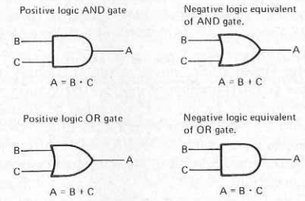
Input		Output
B	C	A
0V	0V	0V
+5V	0V	0V
0V	+5V	0V
+5V	+5V	+5V

Truth Table

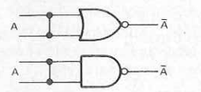
Input		Output
B	C	A
1	1	1
0	1	1
1	0	1
0	0	0

Notice that the Truth Table for the negative logic AND gate is exactly opposite to the Truth Table for the positive logic AND gate. The negative logic AND gate acts as a positive logic OR gate.

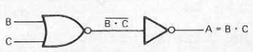
Consequently an AND gate can provide the OR function and an OR gate can provide the AND function by selecting positive or negative logic level assignments.



Flexibility in implementation of the basic gate functions. NOR gates and NAND gates can be used to implement any of three basic logic functions. Example: By connecting all the inputs of a NOR or NAND gate together we can implement an inverter.

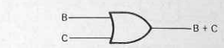


By connecting an inverter in the output of a NOR gate, we can implement an OR gate. (Same applies for a NAND gate.)

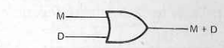


BOOLEAN ALGEBRA

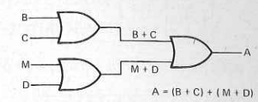
Is the mathematical method of analyzing logic circuits. Boolean equations describe the operation and provides the mathematical tool for the manipulation of logic circuits. For example, draw the logic circuit that solves the boolean equation $A = (B+C) + (M+D)$: The expression indicates that there are two OR gates being OR'ed by another single OR gate: Analysis: Draw the sumbol for the first member of the equation. (B+C):



Then draw the symbol for the second member of the equation (M+D).



Use a single OR gate to combine the two OR gate outputs as required by the indicated + symbol.

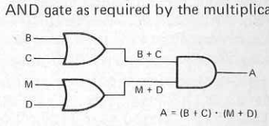


Draw symbol for the second term of the equation (M+D).

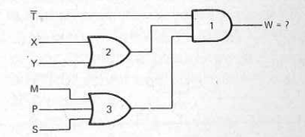
Draw the logic circuit to solve the boolean equation $A = (B+C) \cdot (M+D)$.

The first term (B+C) of the equation indicates an OR gate with inputs B and C.

Combine the two OR gate outputs using a single AND gate as required by the multiplication.



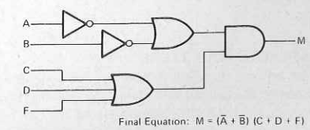
Write the boolean algebraic expression from the given logic circuit.



First write the expression describing the output of gate 2. This is an OR gate, therefore, the expression is X+Y. Secondly write the output equation for gate 3. This is another OR gate, so the expression is M + P + S. Notice that the algebraic expressions are being AND'ed by gate 1. Consequently the output expression so far is $(X+Y) \cdot (M+P+S)$. Input T could have been included anywhere in the equation because it is being AND'ed by gate 1 with the other two equations. The complete output equation is:

$W = (X + Y)\ (M + P + S)\ \overline{T}$.

Write the boolean equation for the logic diagram.



AND GATE

Logic circuits with two or more inputs and a single output capable of resolving an output with combinations of input variables. The two-input AND gate resolves the equation A = BC. The output (A) is expressed in terms of the two variables (B) and (C). The expression A = BC does not imply multiplication but rather (A) is the result of quantity (B) *AND* quantity (C) presented at the input. AND gate symbol

The operation of the AND gate is better represented by the use of a Truth Table that indicates the output for the various input combinations.

Truth Table for 2-input AND gate:

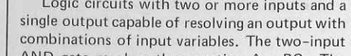
IN		OUT
B	C	A
0	0	0
1	0	0
0	1	0
1	1	1

The AND gate performs binary multiplication.

MULTIPLICATION TABLE
0 X 0 = 0
1 X 0 = 0
0 X 1 = 0
1 X 1 = 1

The total number of possible input combinations for a gate with an even number of inputs is given as $(\text{inputs})^2$ = outputs. For a two input AND gate; $2^2 = 4$. The truth table then contains 4 possible input combinations. Summary: A two-input AND gate gives an output only when both inputs are logic 1.

For a pulse input.

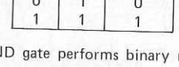


Notice that the AND gate produces 2 output pulses out of 4 input pulses arriving at the B input because the duration of pulse C is exactly twice the total duration of input pulses B.

Algebraic equations. Example: For A = W · TX · MPS. Using AND gate

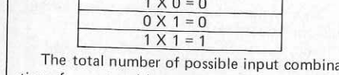
THE INVERTER

A digital gate that inverts the input signal. It is also known as a complementary gate because the output is inverted in relation to the input. The inverted output is written with a bar over the inverted variable.



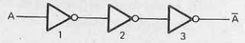
The inverter has only one input connection and one output connection.

For a pulse input:

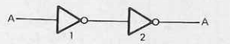


CASCADED INVERTERS

If an odd number of inverters are connected in series, the output will always be the negation of complement or the input variable.

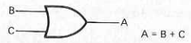


If an even number of inverters are connected in series, the output will be the same as the input with no inversion.



THE OR GATE

Logic circuit with two or more inputs that provides an output when any input is a logic (1). The two input OR gate resolves the equation A = B + C. The output is expressed in terms of either variable B or C acting on the input. The expression A = B + C does not imply addition but rather A is the result of either B or C acting on the input.



The operation of the OR gate is better represented by the use of a Truth Table that indicates the output when the inputs are modified by (1) or (0):

2 INPUT OR GATE

INPUT		OUTPUT
B	C	A
0	0	0
0	1	1
1	0	1
1	1	1

Below is a 3-input OR gate Truth Table.

INPUT			OUTPUT
A	B	C	D
0	0	0	0
1	0	0	1
1	1	0	1
1	1	1	1
0	0	1	1
0	1	1	1
0	1	0	1
1	0	1	1

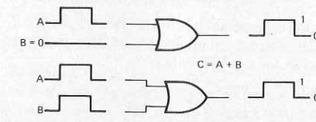
The number of inputs determines the number of combinations in the same manner as for the AND gate: input = combinations – 1. Example: $3^2 - 1 = 8$.

Note: The equation; $\text{inputs}^2 = \text{combinations} - 1$; is for an ODD number of inputs. Use $\text{inputs}^2 = \text{combinations}$ for an EVEN number of inputs. Example: An OR gate with 2 inputs has $2^2 = 4$ possible input combinations. An OR gate with 5 inputs has $5^2 = 25 - 1 = 24$ possible input combinations.

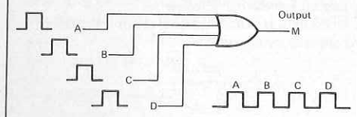
The OR gate equation satisfies the rule of binary addition.

LOGICAL ADDITION TABLE
0 + 0 = 0
1 + 0 = 1
0 + 1 = 1
1 + 1 = 1

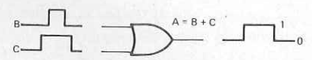
Note: Binary addition and logical addition are not EQUAL. Summary: The OR gate gives an output when any input is a logic one. Output is zero when all inputs are zero.



The OR GATE preserves the individual characteristics of pulses arriving at the input.

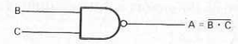


Note: Each output pulse has the same time interval at the output as it had in the input. If there is time coincidence at the input, the output pulse will be equivalent to the longest pulse at the input. Example:

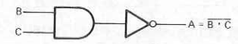


THE NAND GATE

The NAND (NOT-AND) gate is the combination of an AND gate and inverter. The operation of a NAND gate is represented by the equation $A = \overline{B \cdot C}$ and is read A is the result of B and C operating at the input of the NAND gate but inverted at the output. The solid bar over B · C means inversion.



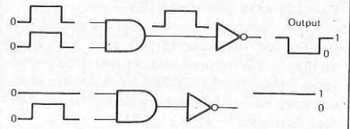
The NAND gate could be constructed by using an AND gate followed by an inverter.



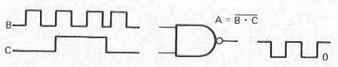
The operation of a 2-input NAND gate is easily represented by a truth table form.

INPUT		OUTPUT
C	B	A = $\overline{B \cdot C}$
0	0	1
1	0	1
0	1	1
1	1	0

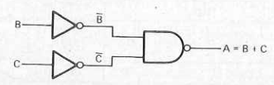
Note: The NAND gate truth table is the complement of the AND gate truth table. Both inputs must be at a logic 1 level to produce a logic 0 output. Pulse behavior:



Non-coincident input pulses have no effect on the output of a NAND gate. The output always stays at logic 1 level. Coincidence at input of logic 1 level pulses produces a negative going pulse at the output.



A NAND gate with inverters connected to the inputs will act as an OR gate.



New Breakthrough In Audio Tape

New metal particle tape for recording provides increased output level, reduced distortion, added high-frequency response and improved S/N ratio.

LEN FELDMAN
CONTRIBUTING HI-FI EDITOR

THE HISTORY OF CASSETTE TAPE RECORDing has been marked by many minor and a few major technological breakthroughs. When the cassette format was originally introduced in 1964 by the Philips Company of the Netherlands, it was not much more than a tape recording medium suitable for low-fi voice recording. Over the years, there have been improvements both in tape hardware (i.e., better electronics, more efficient tape heads, smaller tape-head gaps and such noise-reduction techniques as Dolby and dbx) and, equally significantly, in tape software.

Initially all tapes used ferric-oxide particles suspended in a binder and coated onto a plastic base material. Later on, chromium dioxide (CrO_2) and other particles (such as cobalt-doped ferric oxide) appeared. These formulations primarily improved high-frequency performance, but also the signal-to-noise capability of the cassette recording medium, since they could produce higher maximum-output levels at high frequencies before reaching magnetic saturation.

To understand how these improved tapes worked, it is necessary to examine a typical hysteresis loop associated with magnetic recording. Figure 1 shows such a hysteresis loop. Coercivity, plotted along the horizontal axis, demonstrates how difficult it is to magnetize (or demagnetize) a given tape formulation. Remanent magnetization (also known as remanence) indicates how strong such a magnetization can be. The area shown in the upper-left quadrant of Fig. 1 helps us in judging any particular tape formulation.

If the area in the upper-left quadrant is redrawn with what magnetic tape engineers call "frequency load lines," two things become apparent (see Figs. 2 and 3). In Fig. 2 typical load lines were drawn for 20 kHz and 200 Hz. The lower partial hysteresis curve might be considered typical of ferric-oxide-particle tape; the upper curve could typify the characteristics of chrome tape or one of the treated ferric compounds. Note that increasing the remanent magnetization has a profound effect upon high-frequency performance but little effect upon performance at lower frequencies.

Conversely, Fig. 3 shows that if the coercivity of a given formulation is improved compared with a reference tape, the tape's low-frequency performance is improved significantly while its high-frequency performance remains virtually the same. Clearly, then, an ideal tape is

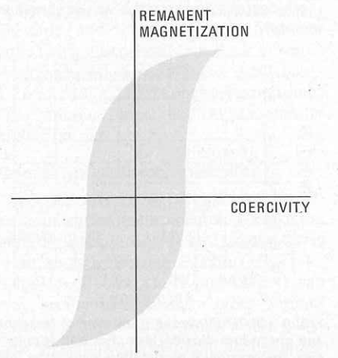


FIG. 1—HYSTERESIS LOOP is obtained by plotting remanence versus coercivity.

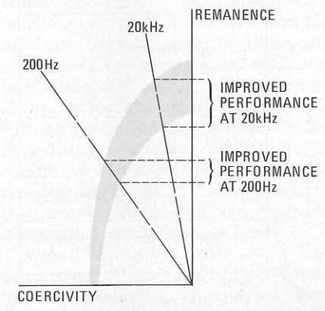


FIG. 2—INCREASING REMANENCE improves high-frequency performance.

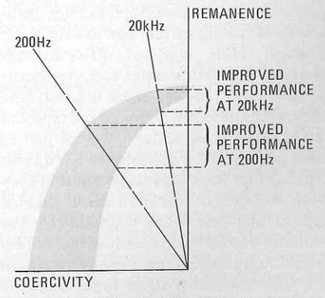


FIG. 3—INCREASING COERCIVITY improves low-frequency performance.