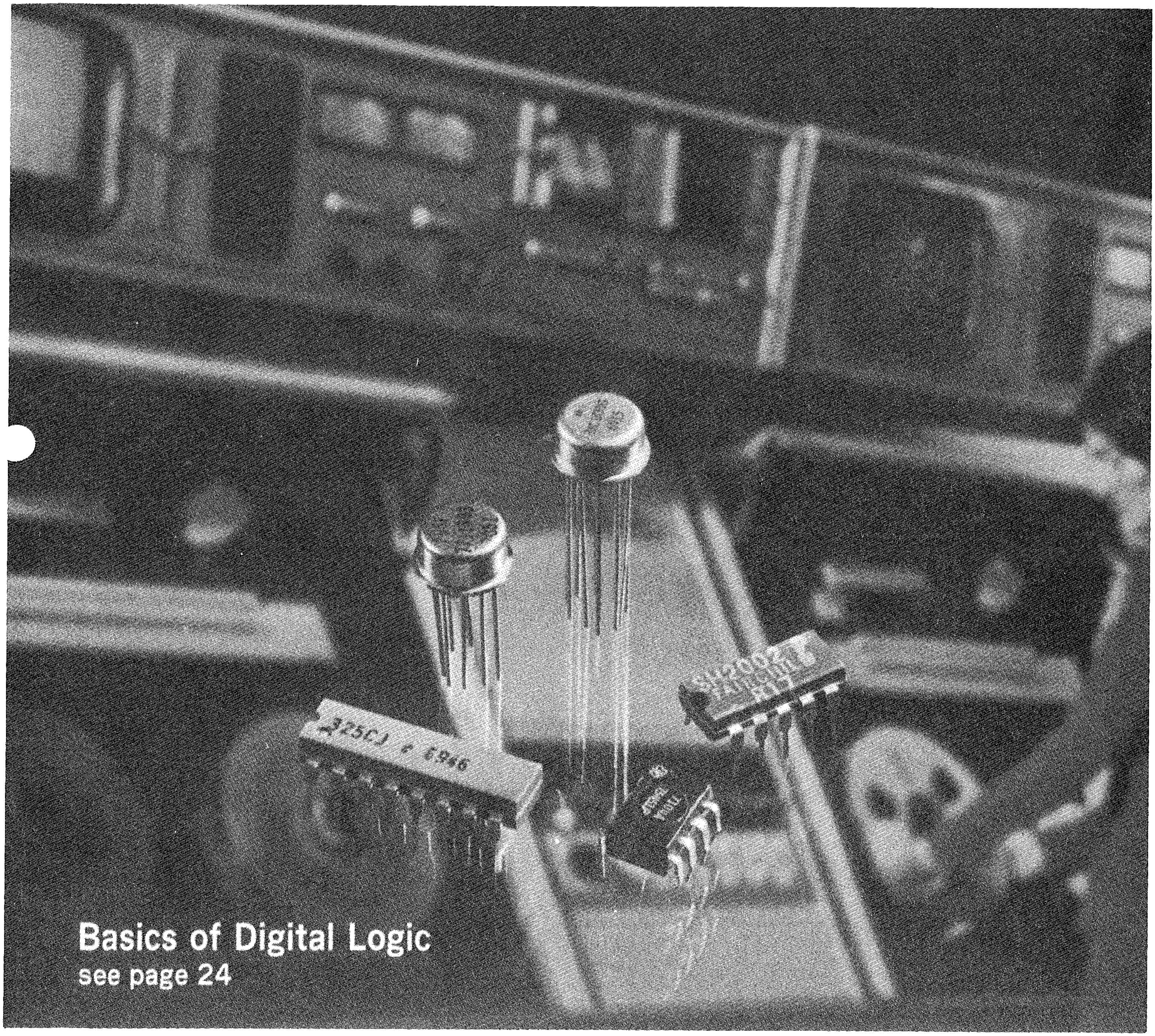
September, 1971/75 cents

Broadcast Engineering®

the technical journal of the broadcast-communications industry

A HOWARD W. SAMS PUBLICATION



FM Overmodulation

Automatic Enhancer

Cart Quality Control

Digital Logic Basics

Part 1 of a four-part series

By E. S. Busby, Jr.*

Competition and automated manufacture have drastically reduced the cost of integrated circuits. With this inducement the makers of broadcast equipment embraced the IC with the same fervor and suddenness as they adopted the transistor a few years ago. The especially low cost of digital IC's coupled with the spread of computer technology is reflected in a drift toward digital solutions to circuit design problems rather than analog ones.

IC's are simply ready-made, conveniently packaged circuits made up of transistors, resistors, very few capacitors and no inductors. "Linear" IC's are amplifiers whose output voltage is proportional to the input voltage. "Digital" IC's are circuits in which the transistors either conduct fully or not at all. The term "digital" is actually a misnomer; "digital" implies "ten" since we humans have ten "digits" or fingers. The name stems from the first computers which were structured around base ten arithmetic. A more proper term would be "binary", since the terminals of digital IC's, like relays and switches have only two defined states.

Except for the following mention, this and the succeeding articles of this series deal only with digital devices and techniques.

The Linear

Generally, any IC that isn't a

*Engineer, Ampex Corp., Redwood City, Calif.

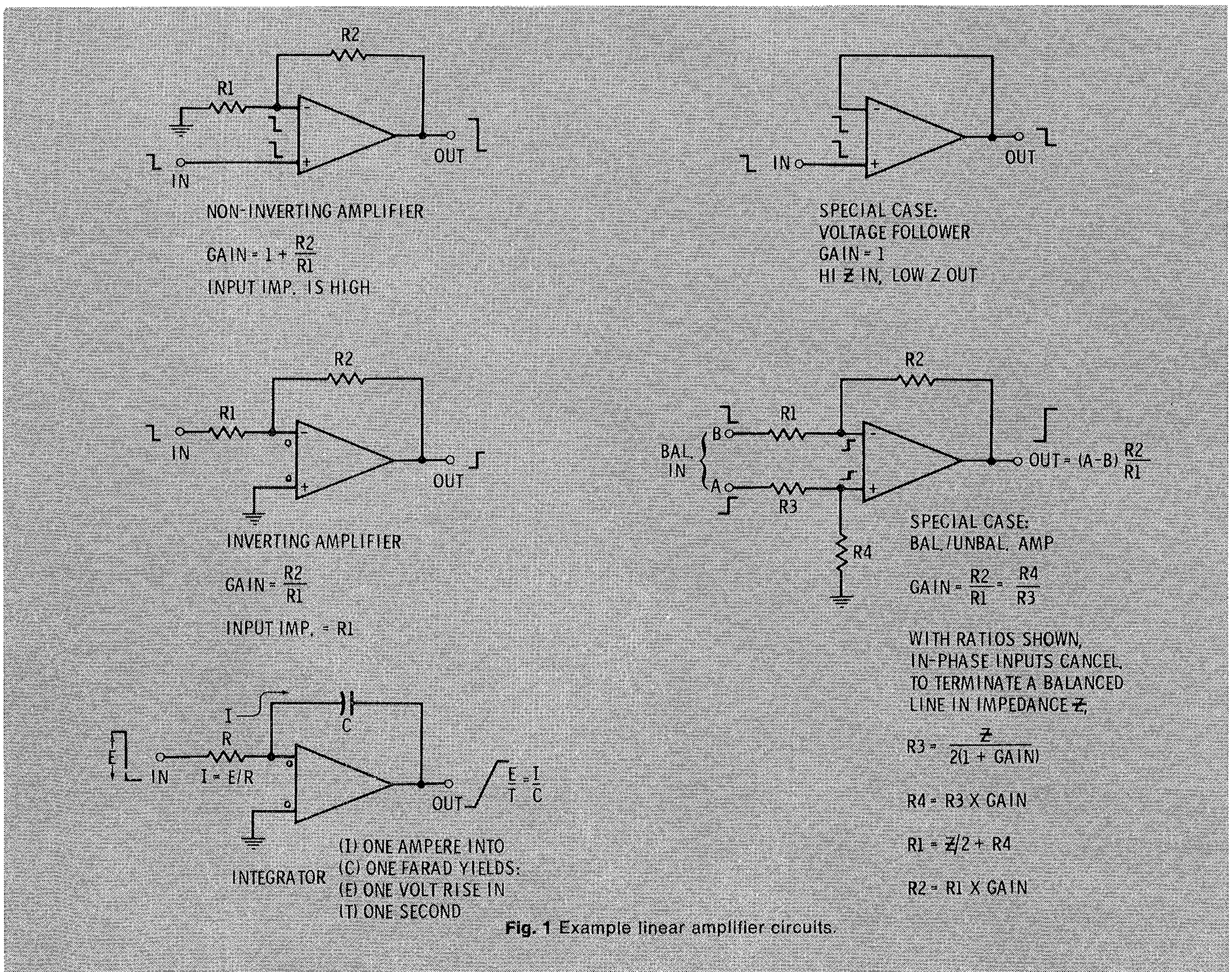


Fig. 1 Example linear amplifier circuits.

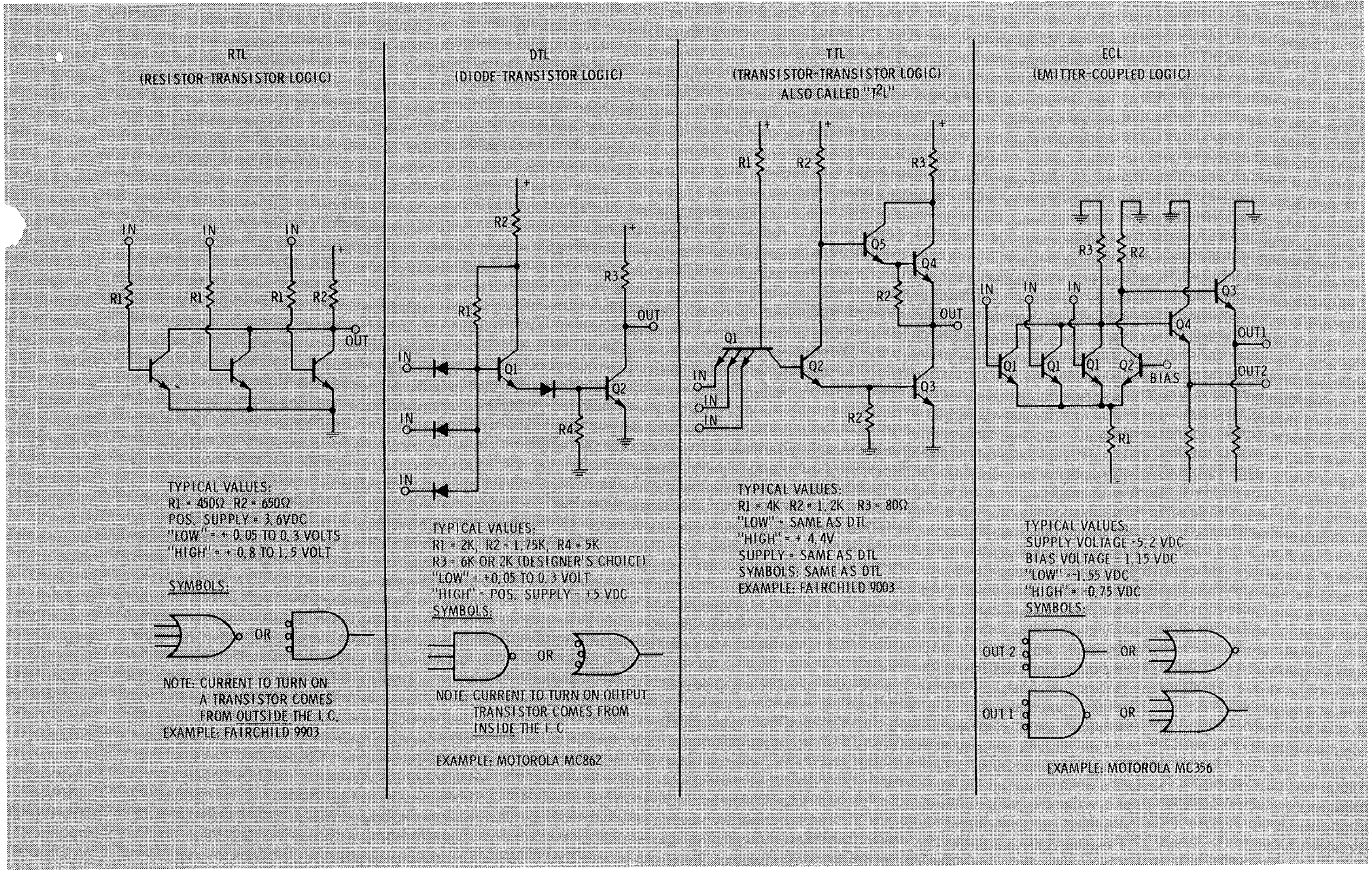


Fig. 2 Four kinds of three-input gates. Note: **For RTL:** taking any input positive furnishes base current to its transistor and turns it on, shorting R2 to ground. If no input is positive, R2 is not shorted and other inputs connected to this output will be taken positive via R2. **For DTL:** taking any input to ground lowers emitter voltage of Q1, turning off Q2. The output voltage rises to + supply. If all inputs are high, all diodes are back-biased and Q1's emitter furnishes base current to Q2, turning it on and shorting R3 to ground. The output then serves to ground any inputs or other outputs tied to it. Q1's emitter current is limited by feedback from collector to base via R1. It doesn't saturate.

Note For TTL: Same logic and supply voltage as DTL. Grounding any emitter of Q1 pulls down base of Q2, turning off Q3 and Q2. Base of Q5 rises, pulling Q4 emitter and the output with it. Short circuit current is limited by R3. Output has low source impedance in both states because of "active pull-up" action of Q4 & Q5. Good for driving capacitive loads, but prevents outputs being connected together.

Note For ECL: R1 furnishes current to emitters of Q1 & Q2. A bias voltage is supplied to base of Q2. If any input is made more positive than the bias, current is steered through its Q1 into R3, and Q2 is cut off. If all inputs are more negative than bias, all current goes through Q2 and R2, and all Q1's are cut off. Q3 and Q4 are emitter-follower outputs. Since outputs are opposing, circuit is good for driving twisted pair lines. Output impedance is low. Bias voltage is supplied by special IC which controls bias to compensate for temperature.

strictly logic device is lumped under the title "linear". Many, such as FM limiters, though anything but linear, are so classed. The linear amplifier found in broadcast equipment is typically a high gain (80-120 dB) DC amplifier, having balanced inputs (called "inverting" and "non-inverting") and a single ended output. Useful bandwidths extend to a megahertz or so, though some units are useful to 10-20 MHz, but exhibit less gain. The full gain is rarely used as inverse feedback from the output to the inverting input is almost always applied.

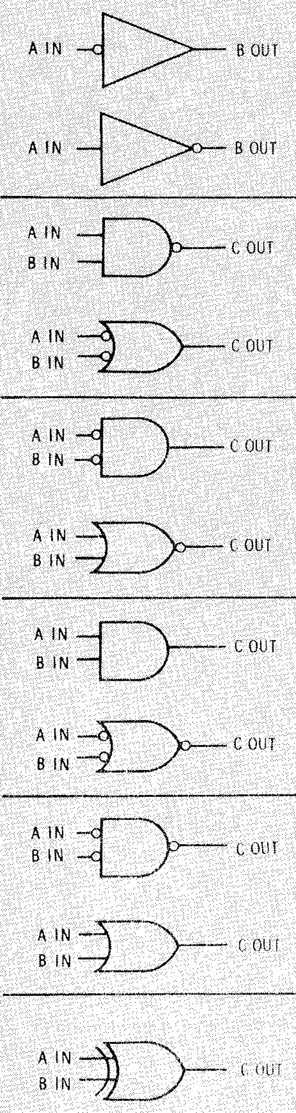
Most IC amplifiers require from one to three "stabilizing" components externally attached. Their values depend on the nature of the feedback path. They roll off the response to prevent oscillation.

For analysis and diagnosis of trouble, it is adequate to presume that a linear amplifier with negative feedback applied (also called an "operational amplifier") will always act so as to maintain the voltage **between** its input terminals at zero. From this and Ohm's law, all else follows. See Figure 1 for some typical examples.

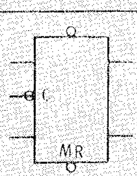
By The Numbers

Broadcast equipment makes heavy use of digital circuits in the performance of simple two-state control logic, pulse shaping and frequency division (as in counting to 525 in a sync generator). In these uses the parameters are fixed, i.e. it's **always** 525 lines and **always** six equalizing pulses. The ability to economically expand the use of digital circuits has resulted in their application to variables.

Frequencies and time periods can be easily converted to a digital form by simply counting cycles of the



SYMBOL	NAME	MEANING	EXAMPLE	BOOLEAN EXPRESSION	REMARKS
A IN — B OUT	INVERTER	IF A IS LOW, THEN B IS HIGH	MOT. MC836	$B = \bar{A}$	= MEANS "IF"
A IN — B OUT	"	IF A IS HIGH, THEN B IS LOW	"	$\bar{B} = A$	$\bar{B}$ MEANS B NOT HIGH B " " IS "
A IN, B IN — C OUT	TWO-INPUT "NAND" GATE	IF A IS HIGH AND B IS HIGH, THEN C IS LOW.	MOT. MC846	$\bar{C} = A \times B$ ALSO $\bar{C} = AB$	X Means "AND" AB MEANS A AND B
A IN, B IN — C OUT	"	IF A IS LOW OR B IS LOW, THEN C IS HIGH	"	$C = \bar{A} + \bar{B}$	+ MEANS "OR"
A IN, B IN — C OUT	TWO-INPUT "NOR" GATE	IF A IS LOW AND B IS LOW, THEN C IS HIGH	MOT. MC1810	$C = \bar{A} \times \bar{B}$	"N" IN "NOR" AND "NAND" INDICATES INVERSION
A IN, B IN — C OUT	"	IF A IS LOW OR B IS HIGH, THEN C IS LOW.	"	$\bar{C} = A + B$	(AND shape) IS "AND" SHAPE
A IN, B IN — C OUT	TWO-INPUT "AND" GATE	IF A IS HIGH AND B IS HIGH, THEN C IS HIGH.	MOT. MC1806	$C = A \times B$	(OR shape) IS "OR" SHAPE
A IN, B IN — C OUT	"	IF A IS LOW OR IF B IS LOW, THEN C IS LOW.		$\bar{C} = \bar{A} + \bar{B}$	BUBBLE INDICATES "LOWNESS"
A IN, B IN — C OUT	TWO-INPUT "OR" GATE	IF A IS LOW AND B IS LOW, THEN C IS LOW	MOT. MC1808	$\bar{C} = \bar{A} \times \bar{B}$	NO BUBBLE INDICATES "HIGHNESS"
A IN, B IN — C OUT	"	IF A IS HIGH OR IF B IS HIGH, THEN C IS HIGH	"	$C = A + B$	
A IN, B IN — C OUT	EXCLUSIVE "OR" GATE	IF A IS HIGH OR IF B IS HIGH, BUT NOT BOTH, THEN C IS HIGH. IN OTHER WORDS, IF A AND B ARE DIFFERENT, C IS HIGH	MOT. MC1812	$C = A \oplus B$	$\oplus$ IS SPECIAL SYMBOL FOR EXCLUSIVE OR



VARIOUS OTHER DEVICES

IF "MR" MEANS "MASTER RESET", BUBBLE INDICATES THAT A "LOW" CAUSES RESET.

IF C MEANS "COUNT" (EDGE-TRIGGERED) BUBBLE MEANS COUNT OCCURS ON A LOW-TO-HIGH TRANSITION. LACK OF BUBBLE INDICATES HIGH-TO-LOW CAUSES THE ACTION.

FOR FULL INFORMATION ON COMPLEX DEVICES, COUNTERS, ETC. CONSULT MFG'S DATA SHEET OR CATALOG.

Figure 3

input frequency for a known period of time, or counting a known frequency over an input time period. Note that in each case the variable is converted into a **number** (of counts). Today's VTR servo systems and time-base correction accessories employ these techniques.[1],[2]

A variable amplitude can also be converted to numeric form, or "digitized". In this form, the variable can have the same operations performed upon it as before: it can be added to, subtracted from, multiplied, divided, compared with other variables, encoded and decoded, transmitted and received, maxima, minima and zero crossings detected . . . with one important difference: **the exact result of any operation can be predicted.** All the binary elements which constitute and define the number are clearly either on or off. No calibrating, no balancing, no drift . . . the only way to distort a digitized quantity is for one or more of the digits to be totally wrong. To process a variable through a digital computer, it **must first** be digitized.

Except for satellite transmissions,[3] it is not yet customary to digitize program video and audio signals. When the advantage of doing so justifies the cost, it will be done.

A kind of digitized video is widely used, however. Circuits using "read-only" digital memories (a "look-up" table) are used in character generators to produce synthetic video that displays numerals, punctuation marks, and letters of the alphabet on a standard TV monitor.

In some studio cameras, control voltages are digitized, then "serialized" (sent one binary element at a time), so that many controls may time-share one wire in the cable.[4] In the camera, these signals are re-converted to DC voltages and routed to the proper pick-up tube electrodes or circuits. One video switcher uses a similar technique in that each button, when pressed, emits a serial "code" which is recognized at the switch point and initiates the desired action.[6] The savings in weight, copper and installation time are appreciable.

Complex digital circuits are becoming more so, are being used more often, and employ an ever-widening spectrum of prepackaged circuit functions. From the maintenance point of view this is a mixed blessing. On one hand, the behavior of a digital circuit can be predicted **exactly,** usually mentally, sometimes with pencil and paper. To do this quickly, however, requires knowledge of digital techniques. Digital circuits rarely have any adjustments and failure to work cannot be corrected by "tweaking". Rapid diagnosis and isolation requires a new and more complicated reasoning process . . . and when the culprit is found, it is more difficult to replace than a transistor. The first and most important tool you will need is knowledge.

The "Families"

Let's first divide all digital IC's into two classes: those which employ ordinary transistors, and those which use field effect transistors (usually MOSFET types). The FET types are called "MOS devices" and little more will be said about them beyond this:

1. Like ordinary transistors, they turn on and off.
2. They use less power and therefore more circuit functions can be fitted into a given package.
3. They operate more slowly.

Devices using ordinary transistors (mostly NPN) are called "bipolar". In this category are four families likely to be found in broadcast equipment. Please refer to Figure 2 and study it carefully before continuing.

Of the four circuits you have just examined, one (ECL) is unique in that none of its transistors ever saturate. (The collector voltage does not closely approach the emitter voltage.) Since it takes extra time to drive a transistor out of saturation when turning it off, ECL circuits are particularly fast, and are used when high frequencies (15 MHz to 100 MHz) are to be handled. All families offer a "buffer" unit, which is simply a logic element with a high-current output stage capable of driving three or four times as many inputs as an ordinary gate.

DTL and TTL devices are compatible and can often be directly interchanged. Special units are available which act as an interface between otherwise incompatible families.

King-Sized or Regular? Filter or Non-Filter? Flip-Top or Soft Pack?

The digital designer is faced with a bewildering choice.

A. He will choose a "family", based on economics and the system's requirements. But within each family there is a choice of:

B. Operating temperature ranges. -55°C to +125°C—full military range 0°C to 75°C—industrial range.
Some plastic packages are limited to 15°C to 50°C, but are otherwise the same as their "industrial" counterpart. Just as 5% resistors are of no better quality than 10% ones, the temperature range does not necessarily reflect on quality, but rather on testing, marking and, of course, price. Besides the temperature range he will choose a:

C. Package. There are round, hermetically sealed metal cans (about the size of a TO-5 transistor), having 8, 10 or 12 leads, or a round molded plastic equivalent. The "Dual in-line package", comes in 8, 14, 16, 24 and 36 pin versions, often available in either plastic or ceramic. For high circuit board density, there is a "flat pack", usually ceramic, whose leads solder to the top of a circuit board rather than protrude through holes. There are still other packages", too numerous to mention. Besides the package, the designer may elect to choose:

D. Low power units. These units operate at a somewhat slower speed, but consume about 1/8 the power of normal devices. Using the same supply voltage, they may be intermixed with normal power units if certain rules are followed. Circumstances may dictate that the designer choose:

E. High level logic. These units use a higher supply voltage (typically +12V), and offer a greater immunity to extraneous noise signals. They may

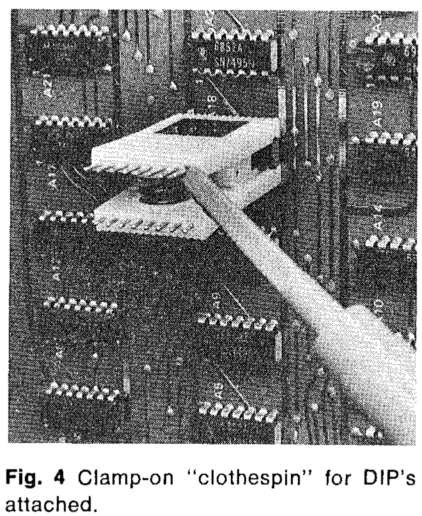


Fig. 4 Clamp-on "clothespin" for DIP's attached.

be used when an existing design must be modified, and only +12V is available. Some industrial equipments use it to provide greater immunity from the transients generated by motors, welders, etc.

Not all combinations of these choices are available. The most popular devices used in new broadcast equipment designs are of the DTL and TTL family, in the dual in-line 14 or 16 pin package, normal power level and supply voltage, and temperature-rated for 0°C to 75°C. Some earlier designs made wide use of RTL; some newer ones employ ECL.

Most IC's Fail On Weekends

It is reassuring to have on hand an exact replacement, bearing all the same numbers. Often, however, one must improvise. Here are some suggestions:

A. Manufacturers license others to make and sell their designs, but not all makers use the original numbering system. Sometimes it is identical, sometimes vaguely similar, sometimes completely different. The NAME of the device and its pin configuration are much more standardized than its number.

B. Except for DTL and TTL, stay within the family. Many TTL and DTL units are pin-compatible, and may be substituted. If replacing a DTL with a TTL device, be **sure** to check that no output is connected to another output. This is not normally permitted with TTL circuits. DTL, being slower than TTL, might not work in high-speed circuits.

C. Devices with a wide temperature range may be substituted for ones with a narrower range. The opposite will usually work as it is seldom that any broadcast equipment exceeds 75°C. The worst that can happen is failure to operate properly when it gets hot, in which case, open the door and plug in a fan.

New Hieroglyphics

The old schematic diagram with its maze of Q's and R's, and to a degree the block diagram, have been supplanted by the logic diagram, which, equipped with package numbers and pin numbers, serves as a schematic. Over the years a number of logic symbol sets have been devised and used, but one survives: ANSI Y32.14.[7] In this scheme, the shape of the symbol tells what purpose a device serves in the circuit, and the presence or absence of a small circle or "bubble" at the input and output terminals defines the voltage level at that terminal when the device is doing what the shape suggests. Study Figure 3 before going further.

Beware: there are old diagrams around which use obsolete symbols, and ones which employ the right shapes, but wrongly use the bubbles. Unless you like the idea of indulging in symbolic translation 10 minutes before sign-on, you would do well to examine and correct your schematics.

Or In Other Words

The terms "positive logic" and "negative logic" refer only to explanatory notes and do not affect the schematic otherwise. A statement (such as RECORD BUS) in "positive logic" means the bus is **high** when **active;** in "negative logic" it would be low when active. A negated statement (such as $\overline{\text{RECORD BUS}}$, in positive logic, means the line, when active (recording), is NOT high (therefore is low).

Many instruction books fail to state which notation is being used and are inconsistent as well. Negative notation is often used with RTL designs and positive notation for others. "Purify" the notations on your schematics. It is good practice and a worthwhile exercise.

ECL devices typically use a negative power supply with collectors returned to ground. An ECL "high" is a less negative voltage than a "low", therefore for the families mentioned, a "high" is further up on the scope than a "low".

Tricks and Tools Of The Trade

Plan to use a scope. Use low-capacity probes. For VTR's with

Fig. 5 Pneumatic device for removing solder.

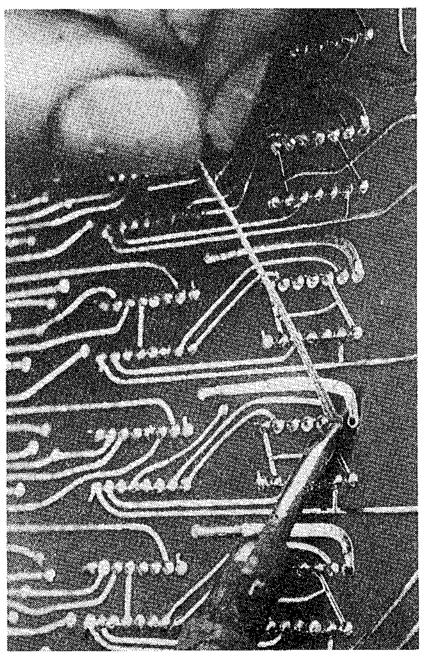


Fig. 6 Flux-impregnated braid can be used to "wick up" solder by capillary action.

editing equipment, cassette tape transports, or anything with a slow cycle time, a memory scope is needed. Have access to a dual-trace preamp, and for memory scopes, four traces can be useful. The preamp sensitivity should be such that 0.1 V is easily discernable. A clamp-on "clothespin" connector for DIP's is available.[8] See Figure 4. Get as many as you have preamp inputs plus one for the trigger probe. When not using the clothespin, use a sharp probe to avoid slipping. Many an IC has become silicon vapor from an accidental short from the positive supply to an output.

Ultimately reason, logic, observation and (as a last resort) consulting the instruction book lead to an illogical IC. If the IC isn't in a socket it pays to search for external causes of malfunction before performing surgery.

A. If an output is stuck at the power supply voltage, touch a 3.3K resistor between output and ground. If the output voltage fails to lower at all, chances are it is shorted to the supply somewhere.

B. If the output is exactly at ground, touch the resistor from output to supply voltage. Failure to change indicates a short to ground.

C. If the output remains logically "low", but exhibits the typical 50 to 300 milivolts above ground of a saturated collector, it is possibly shorted to another output which is "low". Outputs are sometimes deliberately connected together in what is called a "wired-or" connection. First try an ohmmeter check to other outputs on the circuit board. Then isolate the output from all loads by clipping the pin or cutting a circuit board trace. (These are quicker to repair than a full replacement.)

D. If a DTL or TTL input never gets more positive than about 1.5 to 2 volts, check for an open circuit on the board. Plated through holes on double sided boards are notorious offenders.

Unless it is a very expensive IC, dont try to save it intact. It isn't worth the time. The author has found it most convenient with DIP's to nibble the thing off at the ankles with a pair of end-cutting offset nippers so that each lead may be worried out of its hole individually. A spring-operated pneumatic piston device[9] is available for getting out solder. (Figure 5). Another popular method is flux-impregnated tinned braid.[10] Pressed next to a solder filled hole with a hot iron, it picks up the solder by capillary action. (Figure 6). When most of the solder around a lead has been extracted, grip the lead with a pair of long-nosed pliers and vigorously wiggle it laterally until it is free in the hole.

Before removing an IC, carefully note its position. There are nine wrong ways to install a round 10-pin IC. Mark the position of the tab or other reference mark on the package. Use only a wax-type colored pencil . . . never a graphite pencil. When free, remove it gently. Forcing or prying an IC often results in peeling a trace from the top of the board.

When installing a round IC, clip off a tiny bit of each lead except one, such that all are different lengths. This way you must guide only one lead at a time into its hole. Solder first, then clip off excess length.

With DIP's it is considered prudent to solder pins on top the board as well as on the bottom wherever a top trace joins a pin. This uses the IC pin as a guaranteed connection between top and bottom. . . just in case the plated through hole was damaged during removal.

Finally, let the IC cool a few tens of seconds before applying power.

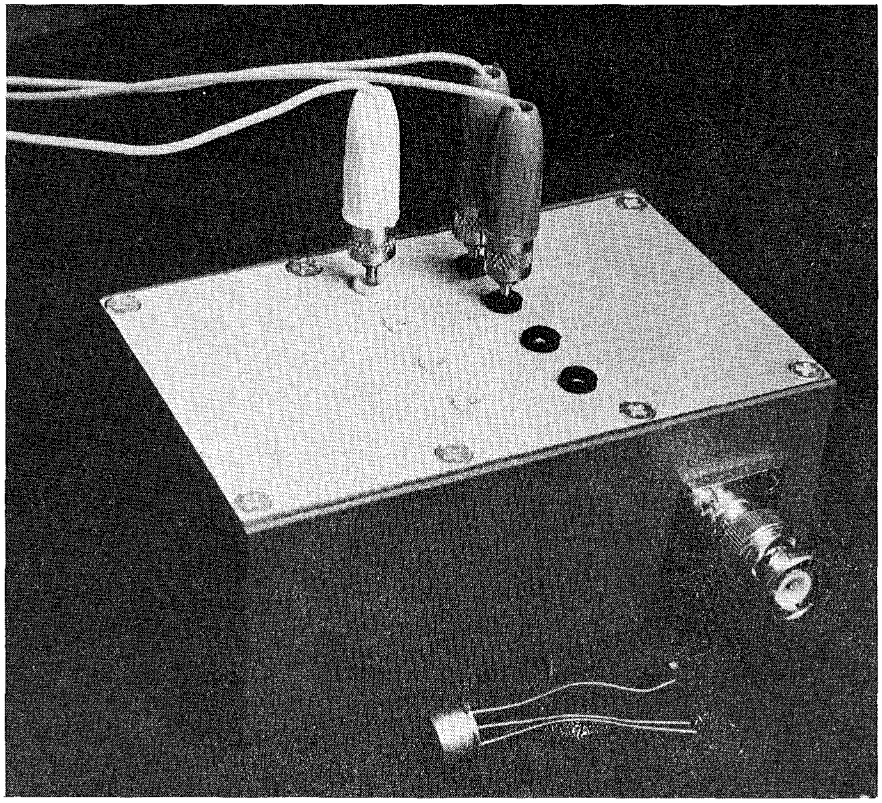


Fig. 7 "And-Gate" trigger with multiple inputs for triggering a scope one count before something is supposed to happen.

Handyman's Special

In many dynamic circuits; i.e. counters, data modulators, etc.,

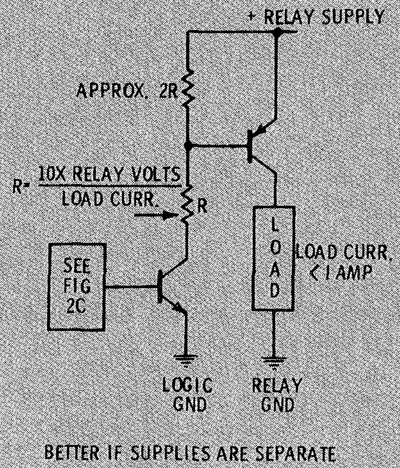


Fig. 2b

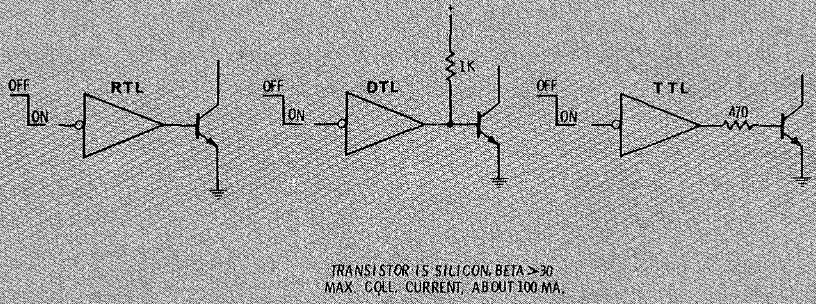


Fig. 2c

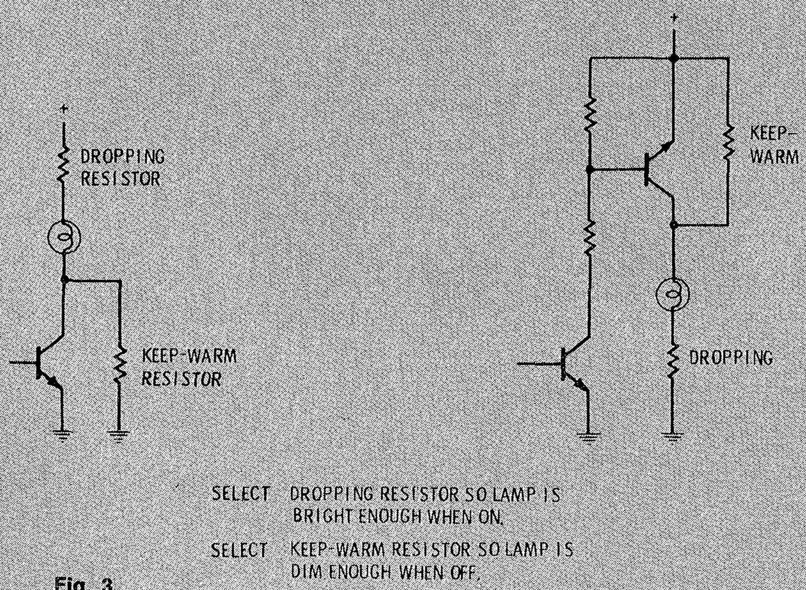


Fig. 3

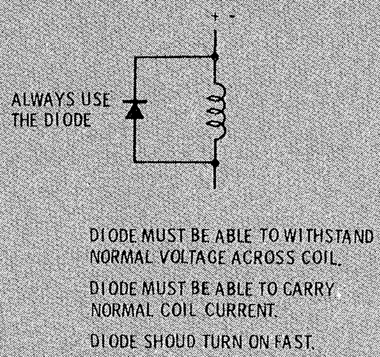


Fig. 4

The idea is . . . timing, control, and memory are something else again.

Also obvious from these examples is that inputs are likely to originate from buttons, switches and other non-IC items, and result in the operation of lamps, relays, solenoids, motors and the like. The IC logic is in between.

This installment will dwell on the problems of getting signals into and out of logic IC's (interfacing), some of the non-combinational uses of IC gates, and edge-triggered devices.

Remoting Panel Controls

A popular equipment modification is to extend panel controls to a comfortable place where there is a chair. If the panel control connects directly to IC's, be wary. A relay takes from 4 to 40 milliseconds and 50 to 500 milliwatts to energize. An IC equivalent can do it in 4 to 40 **NANO** seconds on 1 to 5 milliwatts. The wire to the remote control can easily pick up this much energy from an adjacent wire, especially if that wire is "unsanitary".

Since it is difficult to press a button for less than 10 milliseconds, some form of R-C filtering at the IC input can be used to de-sensitize it to short noise spikes (the usual problem). A simple shunt capacitor across the input will work, but renders its control wire unsanitary because of the large current surge through it when the capacitor discharges through the wire and the switch contact. Figure 1 shows a couple of workable answers. The idea is to furnish a moderate current, over an extended time, and also limit the rate of voltage change in the control wire (out of respect to its neighbors).

The wires leading to IC inputs should be bundled separately from those leading to higher powered devices such as lamps and relays. If a ground is necessary at the remote point, assign one or more wires in the cable as ground returns. **Don't** use a shield for this purpose. A shield must be **able** to conduct, but shouldn't be used so that it **must.**

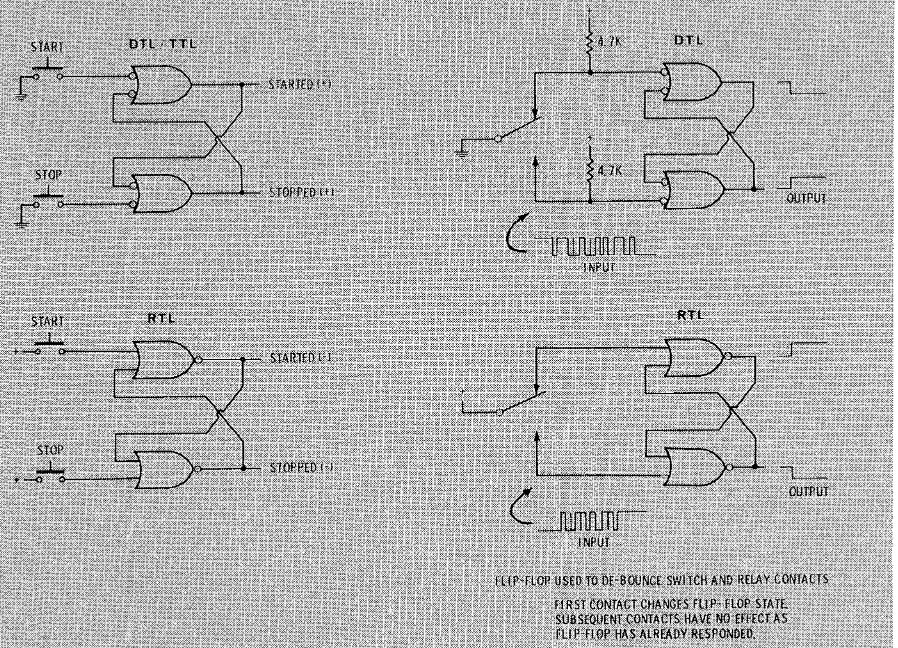


Fig. 5

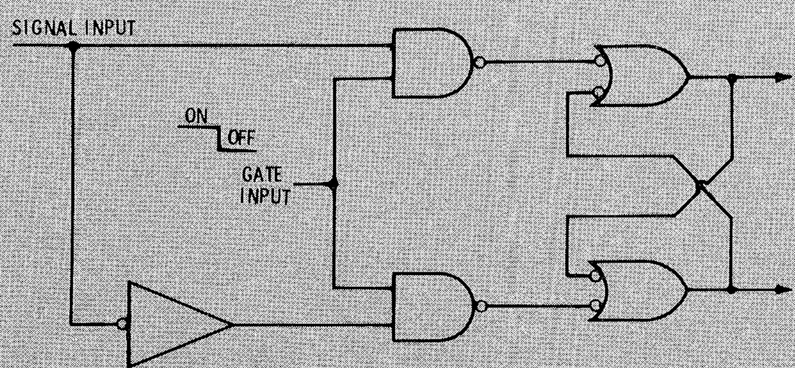


Fig. 6

GATED LATCH

Stealing Logic Signals From Existing Equipment

As thief, it is your responsibility to avoid fouling up a logic circuit in the process of extracting a signal from it. If the wire you attach is going any distance, isolate it by interposing an inverter or gate between the signal source and your wire. Unused spare gates and inverters are frequently found on circuit boards. Steal them.

Always isolate when taking a signal from a flip-flop. Antennas on flip-flops are indiscreet. Make sure your input doesn't cause the drive capability of the output you are using to be exceeded.

Altering Logic In Existing Equipment

Probably the cleanest approach is to design your new circuit so that its installation requires minimum surgery to the circuit board. Mount and prewire your new circuit on a piece of pre-punched board. Mount this outrigger board on or nearby the one to be modified. The idea is to make it easy to restore the equipment to original condition. Its second-hand market value is better that way.

Driving With IC's

Most of the common items that need turning on, like lamps, relays and solenoids, use a higher voltage than the IC's use, and are powered by a supply that is not only separate, but may often be part of another system. Figure 2 shows a method of turning on something that is part of another system without tying the grounds tightly together. It is tolerant of minor differences in ground potential.

Incandescent lamps present a turn-on problem because the cold filament resistance is so low. Figure 3 illustrates two approaches which reduce the turn-on current surge. Either one or both together help, and extend the life of the lamp as well.

The problem with relays and solenoids is turning them off. Being inductive, they resent it, and kick up a vicious voltage "spike" when their current supply is interrupted. Properly used, this effect operates automobile spark plugs, but is death to transistors. A diode connected across the coil will "catch" this spike and prevent it from exceeding the supply voltage. See Figure 4. The presence of the diode causes the magnetic field to decay more slowly than it normally would. If this added turn-off delay is a problem, you probably shouldn't be using such a device anyway.

It is prudent to include a similar diode in lamp circuits, since sooner or later some less knowledgeable person will attach a relay to the lamp circuit without benefit of diode.

Put It In Writing

Whatever you do to existing equipment, document what you do. Put a note on the old schematic referring to the new page you will add for the new circuit. If it's complicated, write an explanation of how it works and put that in the book, too. The kind of job security that comes with "keeping it all in your head" can get a little **too** secure.

The Most Common "Memory"

Figure 5 shows how two ordinary gates can be connected to each other to form a bistable flip-flop.

In the configuration shown, the outputs of the flip-flop will always reveal whether its corresponding input is **active,** or was the **last to be active.** Ideal for remembering that a momentary push-button was pressed, this circuit is also called a "latch", and an "R-S" flip-flop. (The inputs are called the "set" input and the "reset" input.)

Metallic relay and switch contacts are unreliable for the first few milliseconds after they are whacked together. Figure 5 also shows how a simple flip-flop can be used to derive a single edge from a switch instead of the usual shower of sparks.

When a button, switch, or relay contact is feeding an IC circuit which produces alternate action (ON-OFF-ON-OFF) or which counts the number of button pushes, it is mandatory to "debounce" the button.

The Gated Latch

The circuit of Figure 6 illustrates how a gate can be connected between an input and the latch. As shown, when the gate is on, the output of the latch will "follow" the input. When the gate is shut off, the latch is disconnected from the input, and stores the input's state at the instant the gate shut off.

The fact that the output follows the input while the gate is on can sometimes be a problem, in which case simply turn on the gate with a short pulse at the time you want to "examine" the input.

Making Pulses

To make a short pulse out of an "edge" (a transition from high to low or vice versa) simply AC couple the transition to a gate or inverter's input as in Figure 7A. Figure 7C shows what to do to get long pulses from short inputs.

The internal component values of IC's can vary about ±25 percent between samples. Therefore don't use these circuits where tight control of pulse widths is required. Pulse widths will vary with temperature as well.

Pre-packaged "one-shots" (monostable multivibrators) are available, one or two to a package. You connect a resistor and capacitor to each to determine the one-shot's active period. They may be connected to be positive-edge or negative-edge triggered, and may be made "retriggerable". This means that if a new triggering edge arrives during the active period started by a previous trigger, a **new** active period is defined. Thus a string of input triggers whose repetition period is shorter than the active period will cause the output to stay continuously "active". This feature makes the unit useful as a "missing pulse" detector.

Edge-Triggering

No matter how short a pulse is, there comes a point in the design of things when something must happen at the beginning **or** the end of a pulse (upon an edge) instead of **during** the pulse. There is a vast

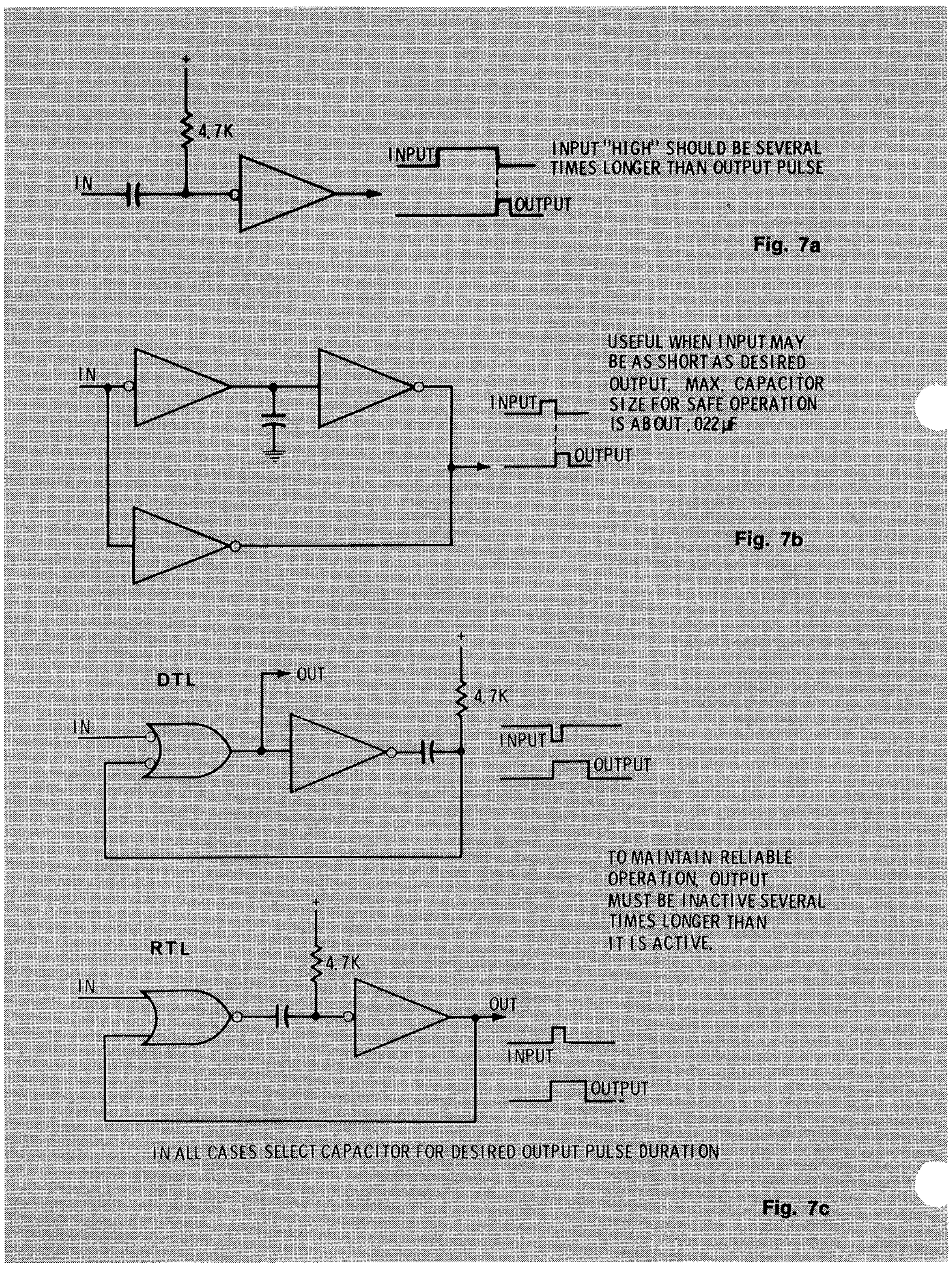


Fig. 7a

Fig. 7b

Fig. 7c

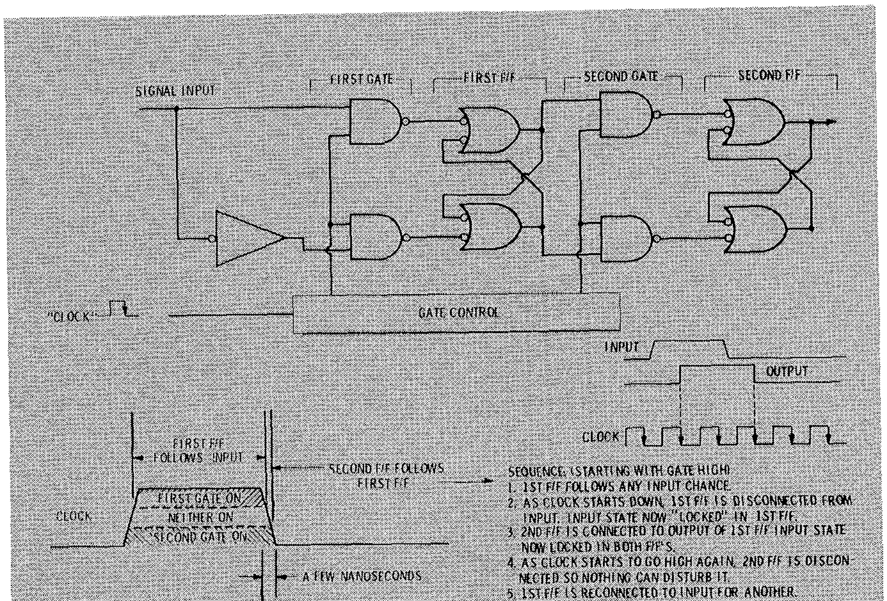
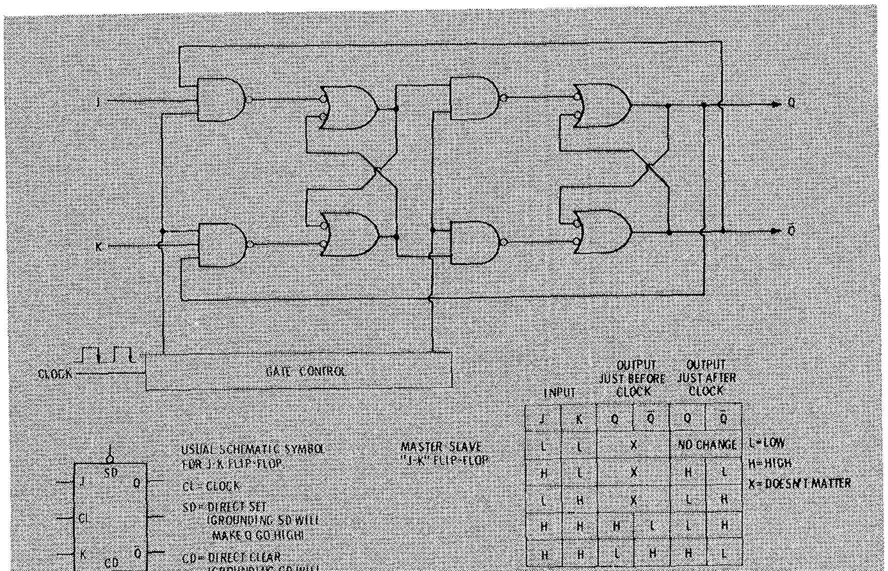


Fig. 8



INPUT		OUTPUT JUST BEFORE CLOCK		OUTPUT JUST AFTER CLOCK	
J	K	Q	$\bar{Q}$	Q	$\bar{Q}$
L	L	X		NO CHANGE	
H	L	X		H	L
L	H	X		L	H
H	H	H	L	L	H
H	H	L	H	H	L

J-K FLIP-FLOP TRUTH TABLE

Fig. 9

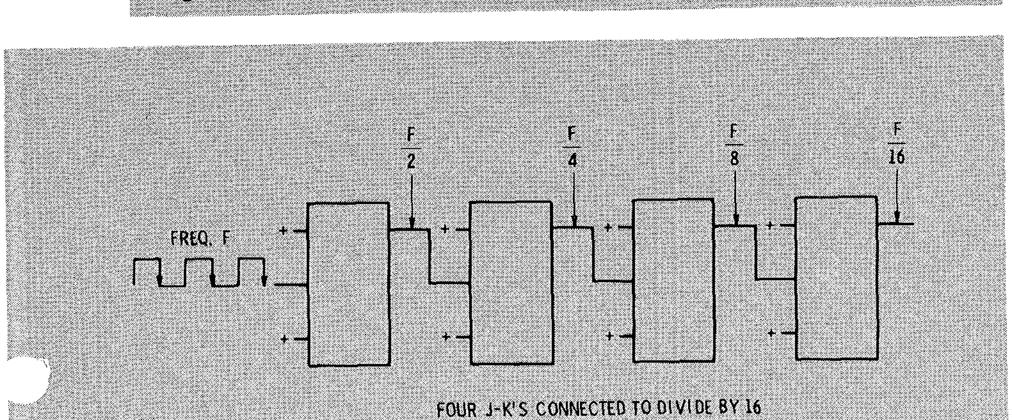


Fig. 10

array of IC devices which do their thing only upon a **transition** at the control input.

Figure 8 shows how two gated latches may be connected in series to form a master-slave type flip-flop. In the form shown, it is called a "D" flip-flop (I don't know why). The output will "follow" the input, but only after a high-to-low transition at the control input. The control input to edge-triggered devices is usually called the "clock" input.

Figure 9 is the same circuit with feedback applied from outputs to inputs. The resulting action has been found to be very useful. The circuit shown is one way of forming what is known as a J-K flip-flop. (There **are** other ways of doing it.)

The significant rules of such a flip-flop are as follows:

1. Nothing at all can happen except upon a specified edge. (Either a positive-going or negative-going one, depending on the device.)

2. If both inputs are inactive, the output will never change.

3. If one output is active, and the other is not, the output will assume the status of the inputs a few nanoseconds after the clock edge. If the outputs were **already** like the inputs, they don't need to, and won't change.

4. If both inputs are active, the output will change to the opposite of what it was before.

The important thing is this: every cycle of an input waveform has two transitions—one negative, one positive. Only **one** of these can produce a change in the flip-flop. If operated as in rule 4 above, every **two** input transitions produce **one** output transition. The result is a divide-by-two, a binary divider.

J-K flip-flops come one or two to a package. Besides the two inputs (called J&K for reasons I know not), two outputs are provided (usually called Q and $\bar{Q}$) as well as one or two **DIRECT** set and clear inputs, which affect the output directly whenever they are energized. This is done so the flip-flop can be used as an R-S flip-flop if desired. Also, since flip-flops can "wake up" either way when power is first applied, they are useful to "initialize" the device, to make sure it gets out of the right side of bed.

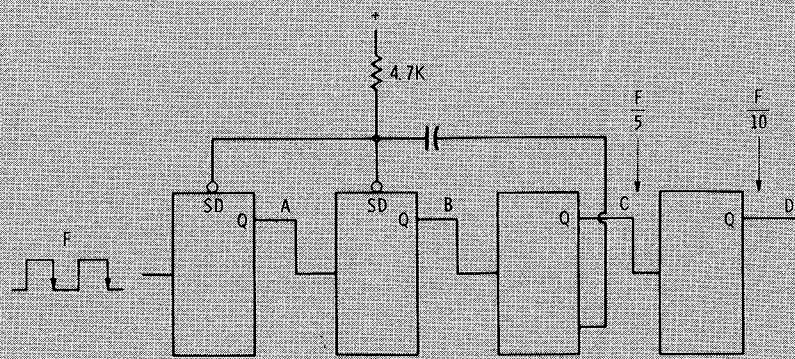
Operating a J-K as a binary divider, and connecting its output to the "clock" input of another produces two divide-by-two's or a divide by four. Three stages divide by 2^3 or 8. Four stages divide by 2^4 or 16, and so on. Applying judicious feedback from later stages to earlier ones can "force" some "counts" that really didn't happen, causing the output edge from the last stage to occur after some number of inputs **less** than normal. A divide by 16 can thus be connected to divide by 16 or any number **less** than 16. Figure 10 shows a divide by 16, and Figure 11, two ways of dividing by 10.

Figure 12 shows a number of J-K's connected in series with all clock inputs driven from a common source. At each clock edge, each J-K's output assumes the condition of its inputs, and the input signal advances one "J-K" per clock edge. Since the output can only change **after** a clock edge, the progression of the signal from left to right can **never** jump **two** J-K's in one go.

The connection shown in Figure 12 is called a "shift register". The input condition, whatever it is, is "shifted" one step to the right for each clock edge. Shift registers with feedback applied from output to input can form a kind of frequency divider.[2] The maximum division is 2xN where N is the number of J-K's in the circuit. It's useful for dividing by small numbers.

Shift registers are also useful in delaying an input logic signal. The delay time is NxCLOCK PERIOD. Prepackaged shift registers are available with up to several hundred stages. Ones having 4 or 8 stages have pins connecting to each stage's output. Larger ones provide input, output, and clock connections and sometimes a few "taps".

Sometimes whole logic systems, or large parts of logic systems, operate on a "clocked" basis, that

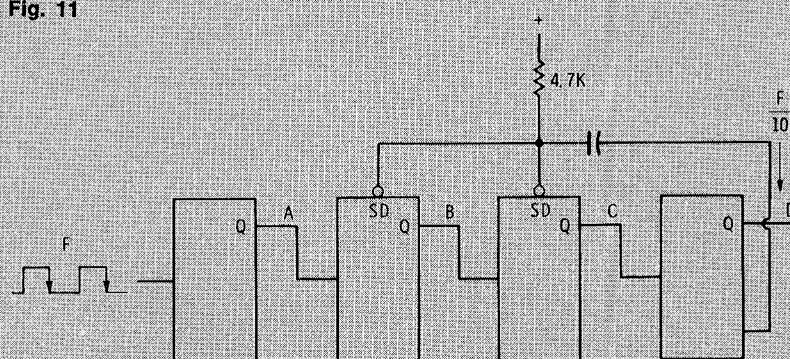


SEQUENCE

		A	B	C	D
	0	L	L	L	L
	1	H	L	L	L
	2	L	H	L	L
	3	H	H	L	L
	4	L	L	H	L
(FEEDBACK) →	4	H	H	H	L
	5	L	L	L	H
	6	H	L	L	H
	7	L	H	L	H
	8	H	H	L	H
	9	L	L	H	H
(FEEDBACK) →	9	H	H	H	H
	0	L	L	L	L

OUTPUT IS A SYMMETRICAL SQUARE WAVE.

Fig. 11



SEQUENCE

		A	B	C	D
	0	L	L	L	L
	1	H	L	L	L
	2	L	H	L	L
	3	H	H	L	L
	4	L	L	H	L
	5	H	L	H	L
	6	L	H	H	L
	7	H	H	H	L
	8	L	L	L	H
(FEEDBACK) →	8	L	H	H	H
	9	H	H	H	H
	0	L	L	L	L

OUTPUT IS NOT SYMMETRICAL

TWO WAYS TO DIVIDE BY TEN

is, anything that **can** change will do so **only** at a clock edge. In such a system you can at least predict when something **won't** happen. You can also predict that nasty switching transients will always occur **after** a clock edge when all the devices are insensitive. Such a system is called "synchronous".

Figure 13 illustrates a method of "clocking in" asynchronous inputs (like from a push-button) into a synchronous system. It produces a single pulse per input edge, which begins precisely at the clock edge following the input, and ends precisely at the next clock edge after that. It guarantees an output pulse of fixed duration and known timing, and is often more useful than a one-shot in making short pulses out of input transitions.

Something was left unsaid about frequency dividers. A divide-by-16 makes one complete cycle of events per 16 input cycles. Four J-K's connected thusly have 2^4 or 16 **different** combinations of which flip-flops are "on" and which are "off". Each input edge produces a different combination. By properly examining the circuit one can tell **HOW MANY** input edges have arrived. A frequency divider can be and usually is a **COUNTER.** ▲

Footnotes

1. Karnaugh, M. "The map method for synthesis of combinational logic circuits". Trans. AIEE, Vol. 72, pt. 1, pp. 593-599; Nov. 1953.
2. Irwin, Jack. "Using the J-K flip-flop in small modulo counters": Fairchild application note APP-120.

Other Reading

1. J. Anderson, T. Gray, R. Walker. "A second generation one-shot", Fairchild Application bulletin APP-173.
2. **Application Memos;** Signetics Corp. 811 East Arques Ave., Sunnyvale, CA. 94086. 1968.
3. Robert G. Middleton, **Pulse Circuit Technology,** pub. by Howard W. Sams.

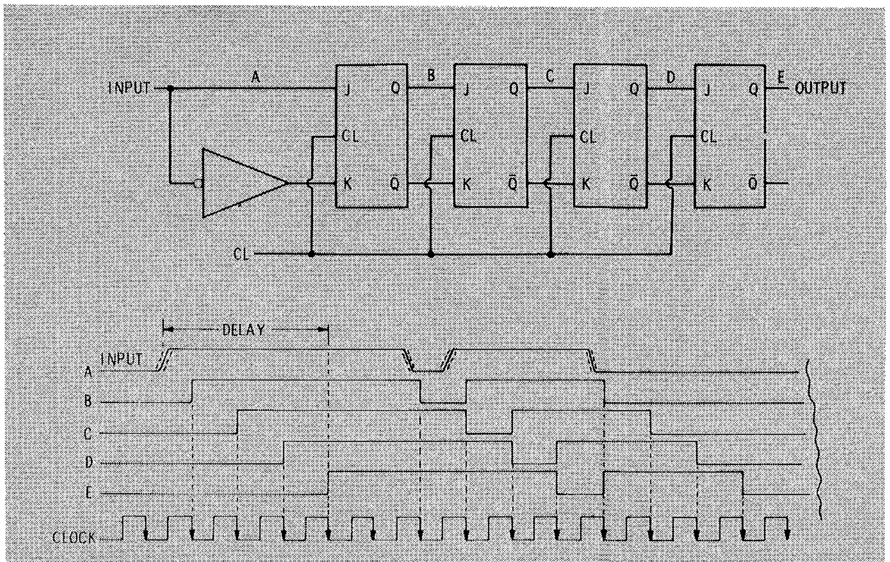


Fig. 12

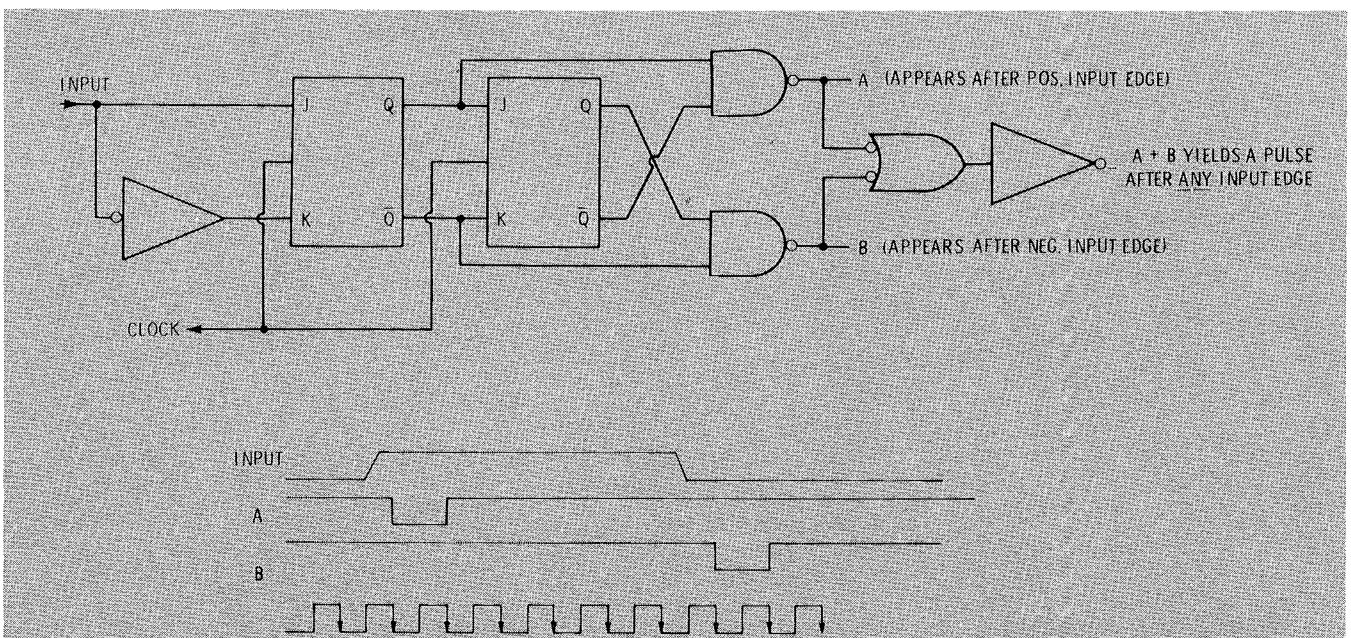


Fig. 13

Digital logic basics

part 3 of a four-part series

By E. S. Busby, Jr.*

Ordinary two-state logic concerns itself with one wire at a time and is fine for yielding the answers to yes/no, true/false, on/off, whether or not, etc. An array of wires can be used to define "how many" and "how much". Two wires, each of which has two possible states, have 2 X 2, or 4 possible arrangements. Add one more wire and there are twice again as many possibilities. If there are N wires there are 2^N different arrangements of ons and offs among them.

If the wires and their states are named in a way everyone understands then the states of a number of wires can define "how many".

Imagine a wire that can have ten different states (ten voltages, perhaps). Each state has a name, called a numeral, running from zero through nine. Each wire has a name, called its "weight", there are units (10^0), tens (10^1), hundreds (10^2), thousands (10^3), and so on. Thus, as we are all taught, when 8092 is written we know that there are:

2 many units=2 X 1
9 many tens=9 X 10
no hundreds=0 X 100
and 8 many thousands=8 X 1000

This scheme (the decimal system) for counting things is used by people with 10 fingers, gears with 10 teeth, and stepping relays with 10 contacts. They all work slowly. Wires with 10 states are more trouble than they are worth. Wires with **two** states we have. So, to make use of two state wires, there came to be a "binary" numbering system. At the risk of creating some confusion, two well-known numerals were borrowed to name the two states. They are zero (0) and one (1). To add to the confusion the names of the wires (the weights) were **also** borrowed from the decimal system. These names are units, twos, fours, eights, sixteens, etc., the mutiples of two.

One And One Are 10

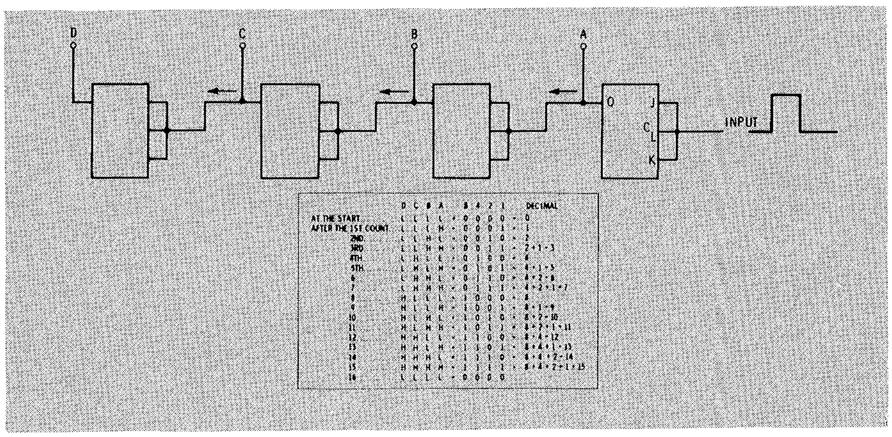
The binary system (as well as some others) are taught to the kids in school these days. To them 111101 means:

A unit	=	1
no two	=	0
A four	=	4
An eight	=	8
A sixteen	=	16
A thirty-two	=	32
TOTAL	=	61

You **think** in decimals. Much of today's equipment **operates** in binary. To do the translation between systems in your head, it is advisable to commit to memory the first 12 multiples of two:

$1 = 2^0$	$128 = 2^7$
$2 = 2^1$	$256 = 2^8$
$4 = 2^2$	$512 = 2^9$
$8 = 2^3$	$1024 = 2^{10}$
$16 = 2^4$	$2048 = 2^{11}$
$32 = 2^5$	$4096 = 2^{12}$
$64 = 2^6$	

To convert from binary to decimal, for example 100111, first examine the right-most "place" or "bit". If it is a 1, then there is a ONE involved, so add it in. If the next to the left is a 1, then a TWO



	D	C	B	A		8	4	2	1		DECIMAL
AT THE START	L	L	L	L	=	0	0	0	0	=	0
AFTER THE 1ST COUNT	L	L	L	H	=	0	0	0	1	=	1
2ND	L	L	H	L	=	0	0	1	0	=	2
3RD	L	L	H	H	=	0	0	1	1	=	2+1=3
4TH	L	H	L	L	=	0	1	0	0	=	4
5TH	L	H	L	H	=	0	1	0	1	=	4+1=5
6	L	H	H	L	=	0	1	1	0	=	4+2=6
7	L	H	H	H	=	0	1	1	1	=	4+2+1=7
8	H	L	L	L	=	1	0	0	0	=	8
9	H	L	L	H	=	1	0	0	1	=	8+1=9
10	H	L	H	L	=	1	0	1	0	=	8+2=10
11	H	L	H	H	=	1	0	1	1	=	8+2+1=11
12	H	H	L	L	=	1	1	0	0	=	8+4=12
13	H	H	L	H	=	1	1	0	1	=	8+4+1=13
14	H	H	H	L	=	1	1	1	0	=	8+4+2=14
15	H	H	H	H	=	1	1	1	1	=	8+4+2+1=15
16	L	L	L	L	=	0	0	0	0		

Fig. 1 Four-bit binary counter.

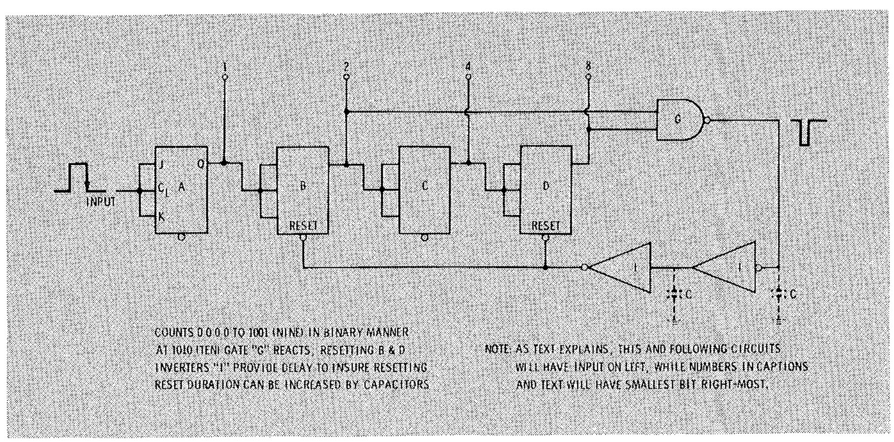


Fig. 2 Example of partial decoding, including reset.

*Engineer, Ampex corporation,
Redwood City, Calif.

is involved, so add it in. If the next to the left is a 1, add in a FOUR, etc. In the example shown there is a ONE, a TWO, a FOUR, and a THIRTY-TWO represented, for a total of 39.

To convert from decimal to binary, for example 525, ask yourself: "Which is the largest multiple of two which **isn't** larger than 525?" 512 fits, so write a "1" in the 512 column. 525 minus 512 leaves 13 unaccounted for. The largest multiple not larger than 13 is 8, so a "1" goes in the 8 column. This leaves 5 unaccounted for, so a "1" goes into the 4 column, leaving a "1" in the 1 column. Write zeroes in all other columns. Result:

512	256	128	64	32	16	8	4	2	1	
1	0	0	0	0	0	1	1	0	1	= 525

Figure 1 shows how the outputs of cascaded flip-flops directly reveal the (binary) number of input edges received. In the example shown, the sixteenth input edge returned the counter to its starting position. The counter exceeded its capacity or "overflowed" at sixteen. Counters which overflow at an exact multiple of two are known as "pure binary". Counters can be arranged by feedback from later stages to earlier ones so that they count in a binary fashion, but return to zero prematurely.

One popular compromise between the decimal world with its 10-fingered people and the binary devices with their two states is the binary counter which returns to zero on the 10th count. Six of the possible 16 states are not used. A 4-bit binary "word" which never exceeds the range 0000 through 1001 (0 through 9) is called BCD, or "Binary Coded Decimal".

This scheme for representing decimal numerals is used a great deal where people enter numbers into a system (with a keyboard for example) or where decimal numerals must be displayed or printed. It is not so much used in computation. There are BCD counters, BCD decoders and BCD/binary and binary/BCD converters. Most decimal numeric displays use a BCD input. Besides the popular 8-4-2-1 binary weighting there is a large number of other weighting schemes using 4 bits to describe the set of 10 decimal numerals. (More about these next month.)

Look at Figure 1 again. The schematic is odd in that the input is at the right instead of the left, as is usual. This was done to avoid a popular source of confusion. In counters, the units output is the one nearest the input, and in left-to-right circuit diagrams the units or "least significant" output is the left-most. When writing numbers, it is customary to place the least significant bit **right**-most. You should learn to visualize binary numbers arranged either right-to-left or left-to-right. To force you to do this, in this and the next article, circuit diagrams will be arranged with inputs on the left, and numbers will be arranged with the units column right-most.

Pick A Number

To know when a counter contains some particular number, it is necessary to "examine" all the stages of the counter to see if that number's unique arrangement of highs and lows exists. Convert the desired number into binary form. Keeping in mind that the unit column is the one nearest the counting input, figure which stages should have high (1) outputs and which should have low (0) outputs. Connect an input of an "AND" gate to the regular output of each stage which should be "ONE" and

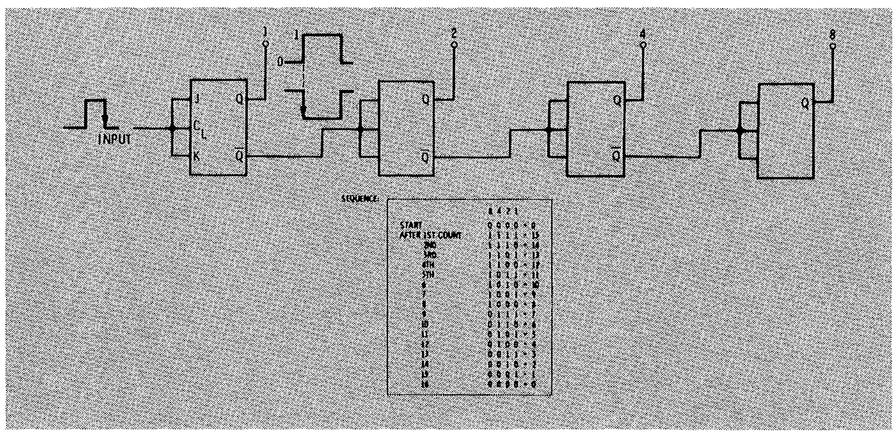


Fig. 3 Down-counter (binary).

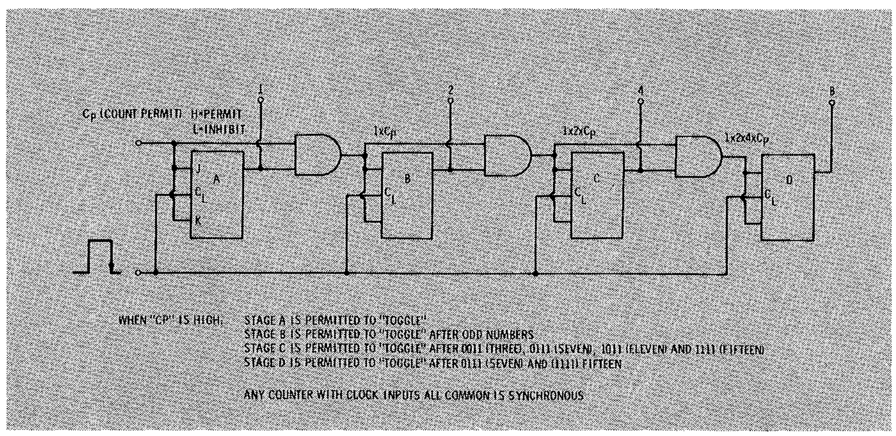


Fig. 4 Synchronous binary counter.

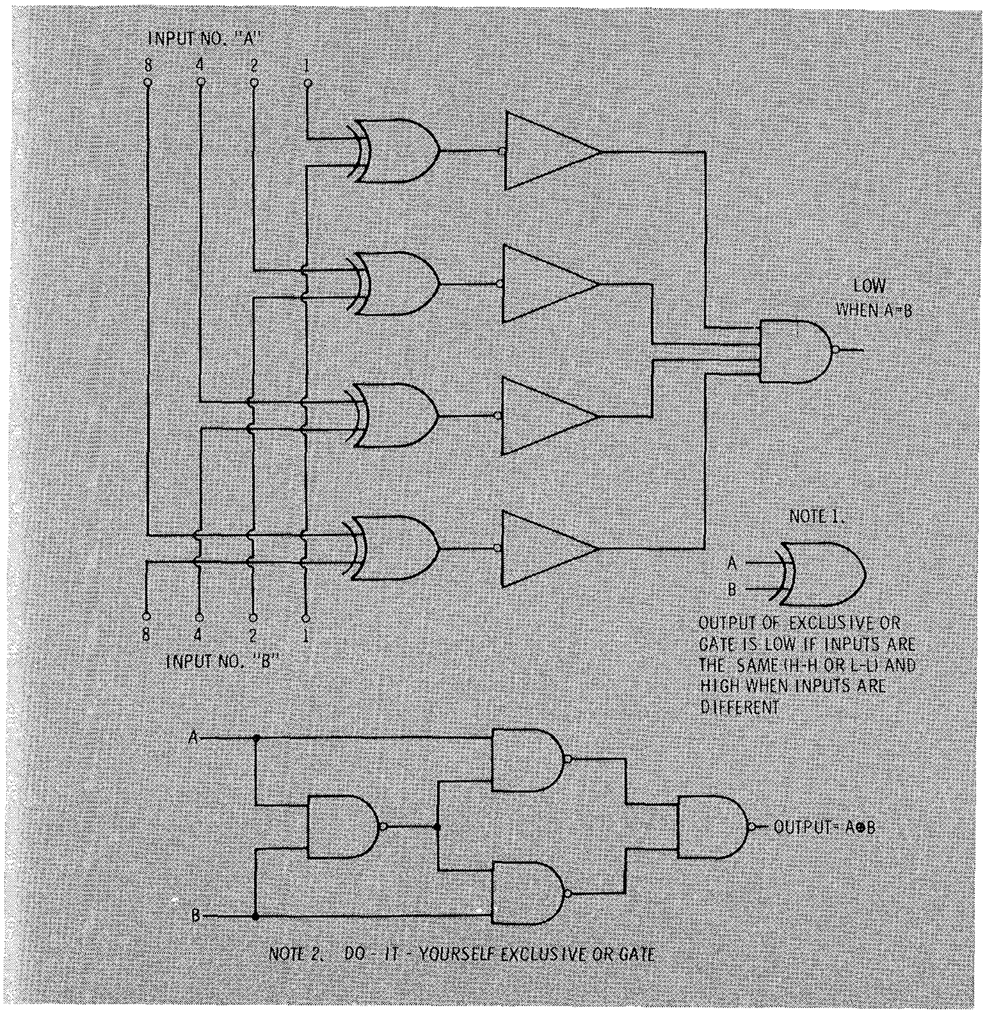


Fig. 5 Simple four-bit comparator.

to the inverted output of each stage which should be "ZERO." This way all the inputs to the AND gate will be high when the desired number has been counted. **No other count** can produce this result. This process is called "decoding" a count. As many decoding gates may be attached as there are different numbers to be decoded.

Partial Decoding

Often a counter is reset to zero as soon as some desired count is reached. In this case it is permissable to examine only the stages which should be active (1) at the desired count. To decode 525, for example, requires only the four-input gate attached to the 512, 8, 4, and 1 stages. If the counter were permitted to count beyond 525, false outputs would appear at 541, 557, 573, etc., but if resetting occurs right after 525 there is no problem.

Sometimes It Pays To Wait

Using the output of a decoding gate directly to reset a counter can cause what is called a "race" condition. This effect is often encountered in logic circuits where the detection of something is used to undo the very thing that was detected. In a counter the danger is that one stage will reset faster than another and thereby negate the resetting signal before **all** stages get reset.

Adding delay between the gate and the resetting terminals, as in Figure 2, will usually assure resetting. Adding capacitors will further lengthen the reset period, maybe even to the point that it's long enough to see on a scope.

Figure 2 happens to be a BCD counter. It counts from 0000 (zero) through 1001 (nine) in a binary manner. On the tenth count (1010), the partial decoding gate reacts and quickly resets the second and fourth stages to zero.

Cape Kennedy Connection

Most counters "increment" (get one bigger) for each input count. Figure 3 shows how a counter can be arranged to "count down" or decrement one for each input edge. The difference is that a counting edge is passed to the next stage when the output changes from 0 to 1 instead of the other way around.

In the counters discussed so far, the stages were connected in cascade. Such counters are called "ripple counters" and suffer from two limitations. The fastest speed at which recognizable counting can occur is limited by the **total** delay time through **all** the stages. Try to go any faster, and the first stage is reacting to an input before the previous count has rippled all the way to the output.

Even when counting slowly, ripple counters are prone to producing "decoding spikes" at decoding gate outputs. You might think that the reception of the eighth count might look like this: FROM: 0111
TO. 1000

It doesn't. It looks like this: 0111
THEN: 0110
THEN: 0100
THEN: 0000
THEN: 1000

with the "rippling" going from right to left in the example. A decoding gate set for zero, four or six would produce a short, but troublesome output during the transition.

Figure 4 illustrates a synchronous counter. The counting inputs of all flip-flops are fed from a common source. The steering inputs, which determine whether or not the flip-flop will flip at clock edge time, are connected, using gates, to the previous stages. Outputs can only change at clock edge time, and when several stages are changing at once, they all do it together.

Speed is limited by the delay of **one** flip-flop and the gate delays. Decoding spikes are drastically reduced if not eliminated.

When not counting at high frequencies and when using partial decoding and resetting, a ripple counter is as good as any. Otherwise, synchronous counters are preferable.

Following, stated somewhat tersely, are some approaches using counters to answer questions and solve problems:

1. HOW MANY?

Start the counter at zero. Make edges out of the events to be counted. When you get curious, examine the counter outputs. Convert these output states from binary (or perhaps BCD) to decimal and there's the answer.

2. HOW FAST?

Start the counter at zero and count input events. Allow the counter to proceed for exactly some known length of time, like one second. (This may take yet another counter.) The number in the counter is then events per second.

3. HOW LONG?

Start the counter at zero. Make edges at some known frequency, like 1 megacycle. Start the counter when one input event is received, and stop it when the next arrives. The number of counts is the **period** of or "time between" the events.

4. IS "A" EQUAL TO "B"?

See Figure 5, which shows the simplest of comparators. It is essentially an adjustable decoding gate.

Figure 6 shows a more complex comparator. It has three outputs, one of which is always active. It reveals whether the counter number is less than, equal to, or greater than the comparator's other input.

5. IS "A" FASTER THAN "B"?

If an "A" arrives, count **up** one. If a "B" arrives, count **down** one. If both arrive at once, do nothing. If the counter reaches its maximum value, stop counting "A"s. If it reaches its minimum value (zero) stop counting "B"s.

If the counter hangs up at the maximum value, A is faster than B. If it hangs up at zero, B is faster than A.

This up/down counter approach is often used as a frequency/phase comparator. When used to phase-lock a local oscillator to some reference input, the filtered output of the counter's most significant stage is used to control the local oscillator's frequency. Phase-lock is indicated by the most significant stage turning on about half the time. An exact 50 percent duty cycle indicates that the two inputs are exactly 90° apart.

Words Of Warning

When measuring time and frequency you cannot usually predict whether you started and stopped the counter just before a count or just after one, leaving an inherent uncertainty of ± 1 count.

When measuring that which lies **between** events, like the length of tape between N frame pulses, or the length of wire between N telephone poles, don't forget that the first

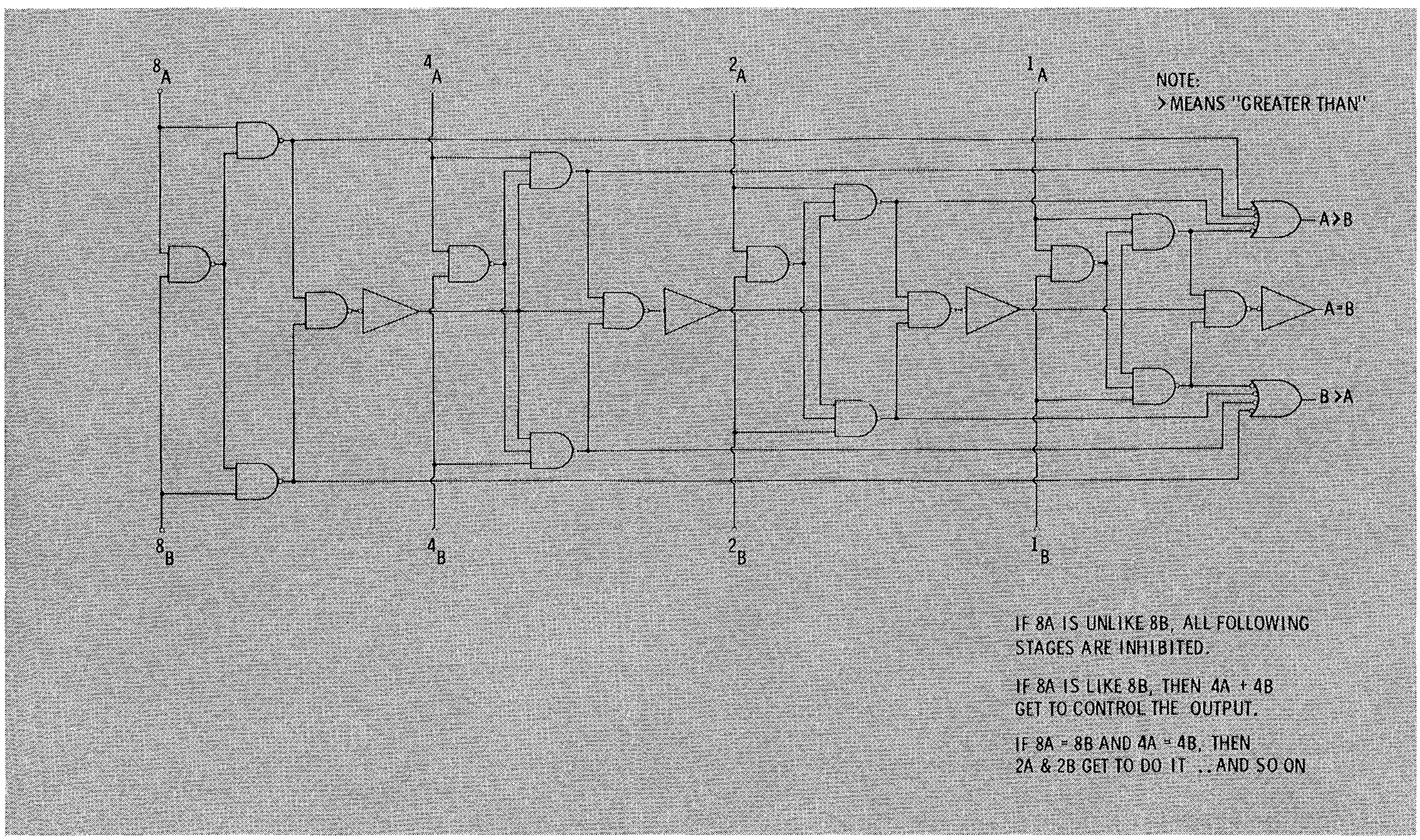


Fig. 6 Full comparator.

phone pole counts one before any wire goes by. For these and similar problems, **remember to subtract one** from the total count.

Put it this way: If you walk four blocks, you touch five corners. The counter counts corners.

When starting and stopping counters, it is best to avoid gating the input signal on and off. Instead, manipulate the first stage's steering inputs. If the input happened to be active when you shut off the input gate, an extra count would result at shut-off. If you must gate the input, it is best to make narrow pulses out of the events to be counted.

Counters and the circuits which are often hung onto them were among the first to undergo medium-scale integration (MSI). MSI devices generally cost about the same, maybe a little less, use a little less power and occupy much less real estate than the equivalent made out of separate flip-flops and gates. There are MSI synchronous and ripple counters in BCD, decimal, and 12 count. Some count up and down. Almost all have four stages.

MSI decoders typically have one output for each possible input pattern, with only one output active at a time. Some can withstand the ignition voltage of gas-discharge numeric display tubes. Others can energize more than one output at a time and can switch the currents used by the incandescent lamps of seven-segment numeric displays. Several 5-bit comparators are available. These have three outputs available: A greater than B, B greater than A, and A equals B. Four-bit binary adders and subtractors are available (more about this next month). Most MSI devices are TTL.

New MSI devices appear on the market at a frightening pace. Read the ads and don't rely on out-of-date catalogs.

How Much?

Some of the variables of life are already "quantitized". Your pay check, when and if it changes, must change by at least one cent, the smallest increment, or some whole number times one cent. Money is expressed as a number anyway, so HOW **MUCH** MONEY really means HOW **MANY** CENTS.

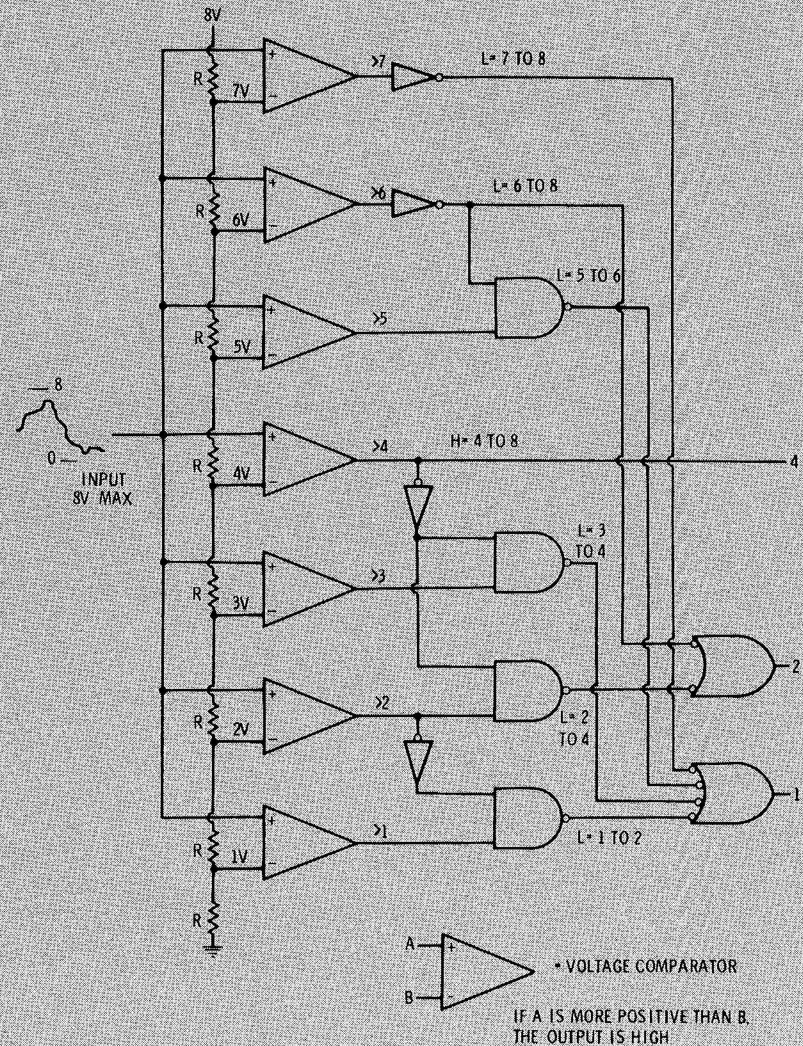


Fig. 7a A three-bit (8 level) A/D converter. No counting is involved.

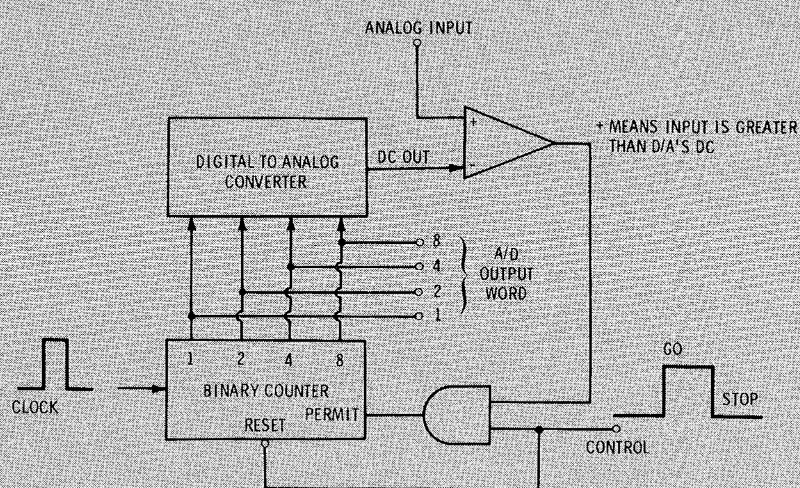


Fig. 7b Counter-type A/D conversion.

The words in the dictionary differ by at least one whole letter. There is no such thing as half-way between "card" and "care". Most goods have some minimum quantity, a division of which does not exist or is not important. So there is a stick of gum, a jigger of whiskey, a page, a word, a grain of sand, and "one each" soldier. The quantity of these usually consists in finding how **MANY.**

Other variables, such as temperature, pressure, brightness, weight, etc. may be measured in terms of some standardized unit like seconds, degrees, grams, centimeters, etc. HOW MUCH TIME then becomes HOW **MANY** SECONDS (or milli-seconds, or microseconds.) It is up to you to decide what is the smallest increment of interest. Even if you are fussy, there is no point in being interested in an increment smaller than the accuracy of the measuring device, nor in an increment so small it is obscured by noise.

Whereas sticks of gum are counted, temperature and pressure are measured. It is often handy to convert one of these continuously variable quantities into a voltage. For this there are numerous transducers like microphones, photoelectric cells, strain gauges, etc. It is sometimes even handier to convert or quantitize the variable voltage into an electrical number. In this form, the variable may be displayed (as a number instead of a meter deflection) or massaged by a computer.

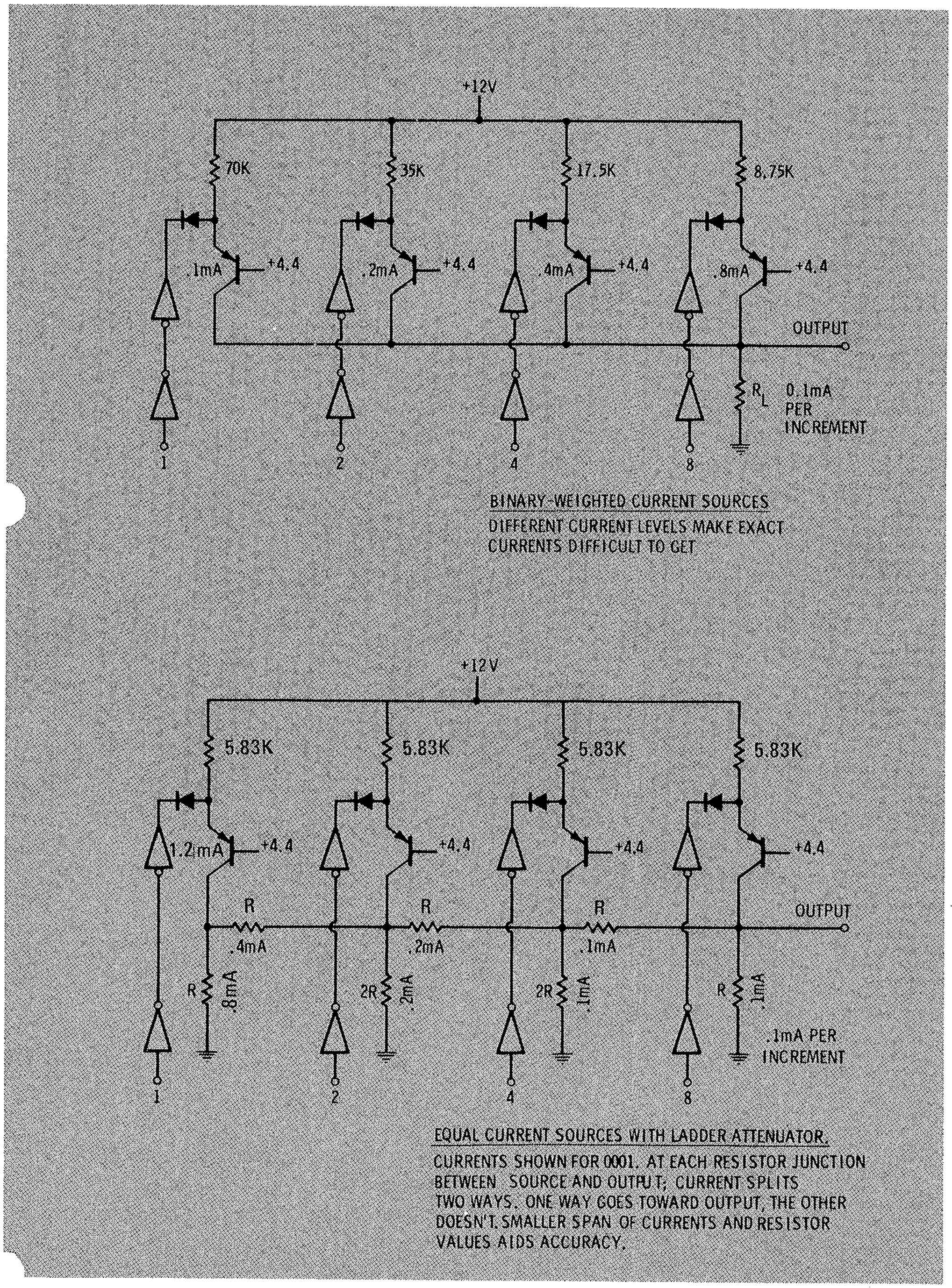


Fig. 8 D/A converters. In the examples shown, current is taken from the emitter via the diode when the bit is inactive (low).

Converting a variable voltage into a numeric quantity is the job of an Analog-to-Digital (A/D) converter. The size of the number that must be used depends on the smallest increment of interest and the maximum range of the variable. The unit "bit" must represent a quantity not larger than the smallest increment, and the number of "bits" must be enough to allow for the maximum variable value.

Thus, to quantitize the outdoor temperature from –40°C to +60°C to the nearest tenth degree, you must accommodate 1000 different temperatures. 10 binary "bits" have 1024 combinations, so a 10-bit binary number is adequate.

There are more methods of A/D conversion than there is room to describe, but generally they take two forms:

1. Using a massive resistive volttage divider "ladder", with a voltage comparator at each rung of the ladder, find out which rung's voltage is equalled or just exceeded by the input variable voltage. Convert this "rank" into a binary number. This method has the advantage of speed in that the output number is determined in one go. It requires lots of hardware, however.

2. The outputs of a counter are connected to a D/A (Digital-to-Analog) converter, whose output is then compared (in a single comparator) to the input variable voltage. Any time an A/D conversion is desired, start the counter from zero and let it count until the D/A output just exceeds the input variable, then stop. The number in the counter is the desired conversion.

It takes time to do the count ing, and this is a disadvantage of this approach. It tends to use less hardware, however, and besides the D/A part can often be used to advantage elsewhere in the system.

Figure 7 shows, in very simple form, two D/A approaches. Figure 8 shows two popular D/A approches. Remember that a D/A output is not truly an analog output. It cannot vary continuously, but only in steps. It is therefore a "dedigitized" voltage.

Both D/A and A/D converters should have an accuracy of at least ½ a basic increment. Thus when a digital number makes that one-step change from 011111111 to 100000000, the output voltage really does go **up** by at least ½ a notch, and no more than 1½ notches.

Once a quantity is made into an electrical number, whether by counting, keyboard entry, thumbwheel switches, or A/D conversion, there **are** some digital techniques that make the whole process worth the effort. That's the subject for next month's article. ▲

Recommended Reading

1. DIGITAL ELECTRONICS WITH ENGINEERING APPLICATIONS, Thomas P. Sifferlen and Vartan Vartanian. Prentis-Hall, 1970. Chapters 4, 6, 7.

2. DESIGNING WITH MSI, Vol. I, "Counters and Shift Registers," by Les Brock. Pub. by Signetics Corp.

3. HOW TO USE SHAFT ENCODERS, Pub. by DATEX Div. of Conrac Corp.

4. HOW TO USE INTEGRATED CIRCUIT LOGIC ELEMENTS, Jack Streater. Pub. by H. W. Sams (Cat #20755).

5. DIGITAL COMPUTERS—STORAGE AND LOGIC CIRCUITRY, Chapt. 6. Pub. by H. W. Sams (Cat. #20131)

6. ANALOG AND DIGITAL COMPUTERS — ORGANIZATION, PROGRAMMING AND MAINTENANCE. Chapter 6. Pub. by H. W. Sams (Cat. #20132)

7. SOLID STATE COMPUTER CIRCUITRY. Chapt. 8. Pub. by H. W. Sams (Cat. #20133)

8. Fairchild Application Notes:
   APP-119, "Ways to increase speed in Large-Count Binary Counters", Jack Irwin.

   APP-120, "Using the J-K Flip-Flop in Small Module Counters", Jack Irwin.

   APP-160, "Applications of the CCSL 9301 Decoder", R. Clive Ghest.

   APP-164 "Applications of the μA722 10-bit Current Source", M. Rudin, G. Erdi, R. Walker and R. Ricks.

   APP-169 "CCSL 9306 Up-Down BCD Counter", R. Clive Ghest.

   APP-170 "Applications of the 9311 One-out-of-Sixteen Decoder".

9. Signetics Application Memos:
   "Binary to BCD Converter", J. Eccher.
   "Self-Correcting Ring Counters", Tom McCarthy
   "321 Dual J-K Binary Applications", Robert Juan.
   "Up-Down Binary Counters", Joe Eccher.
   "TV Sync Generators", Joe Eccher.
   "Television Pattern Generator", R. Seymour.
   "High Voltage Decoders with Conventional Elements", Dale Mrazek.

Digital Logic Basics

By E. Stanley Busby, Jr.*

Part 4 of a 4-part series

One of the advantages of a digital logic system is its ability to manipulate variable quantities (once they are in digital form) without error. A simple comparator (discussed in last month's article) which tells that two quantities (now numbers) are the same, or which one is larger is often inadequate. Sometimes it must be known *how much* bigger one is than the other.

What's The Difference?

Where broadcast equipment uses electrical arithmetic, it is likely to involve subtraction rather than addition. Servo systems, for example, generate a corrective influence which is proportional to the *difference* between what is and what ought to be. Those who service video recorders are familiar with "error voltages." Used in this sense, the word "error" is not the same as "mistake," but means a departure from the ideal... a *difference*.

It is not the purpose of this article to attempt explanation of how a computer finds roots and logarithms, but the simplest of any arithmetic, addition and subtraction, is now in such wide use in commercial equipment that you need to understand it.

One and One Are 10

Since there are only two numerals (1 and 0) in the binary system, its arithmetic is at least five times easier than that of the decimal system. The multiplication table is absurdly simple:

1. 0 times anything = 0
2. 1 times 1 = 1

To make things even easier, we never bother to add a long column of figures a la third grade. If you want to know the sum of A+B+C +D, you add A and B, then add C to their sum, then add D to *that* sum, etc. Lots of wives do their check stubs this way.

Binary addition works exactly like any other scheme... start at the top of each column of numerals, beginning with the least significant column. Add the column. If the sum gets bigger than one "digit," write down the least significant

*Engineer, Ampex Corp., Redwood City, Calif.

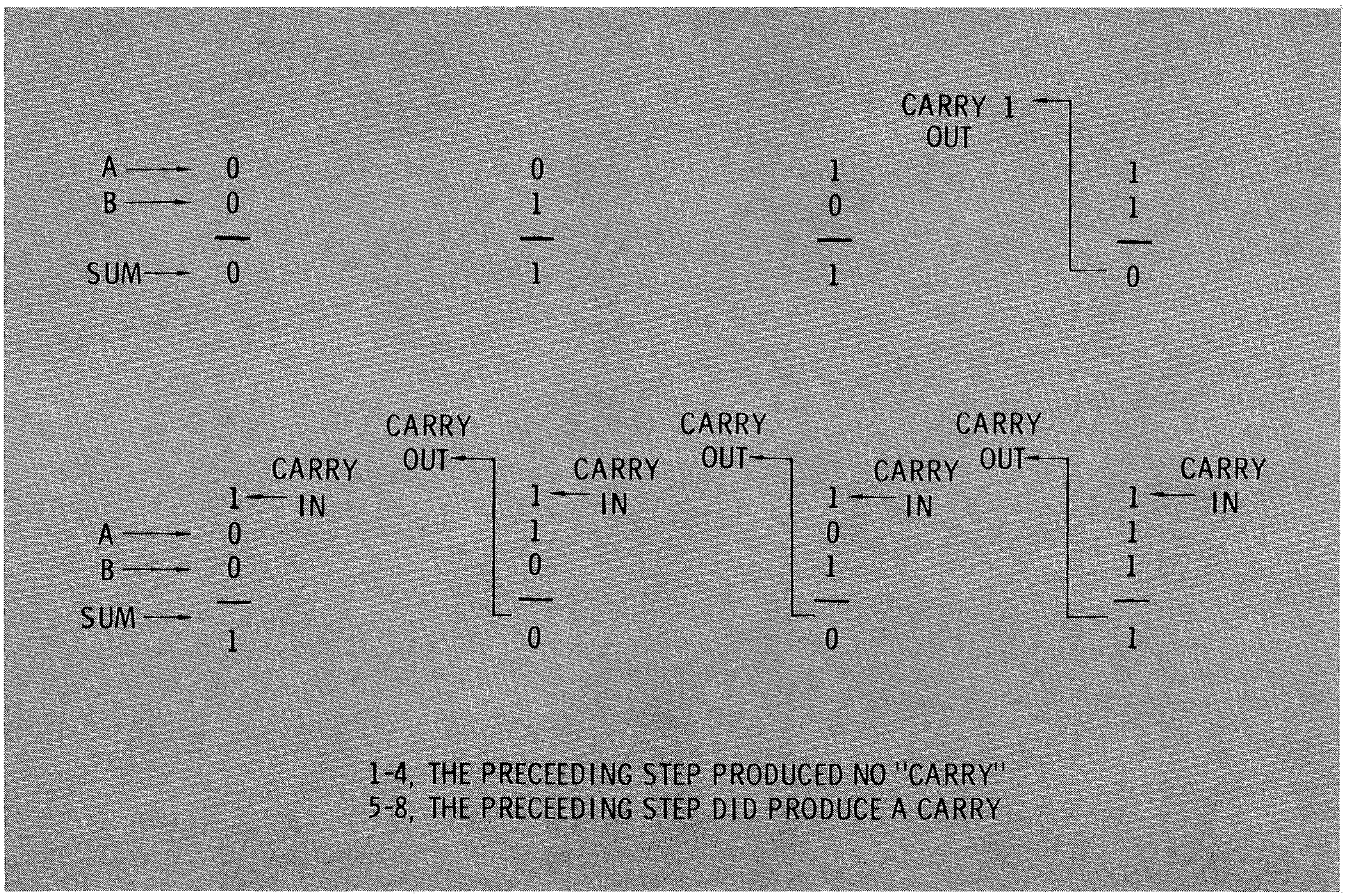


Fig. 1 The eight possibilities of binary addition.

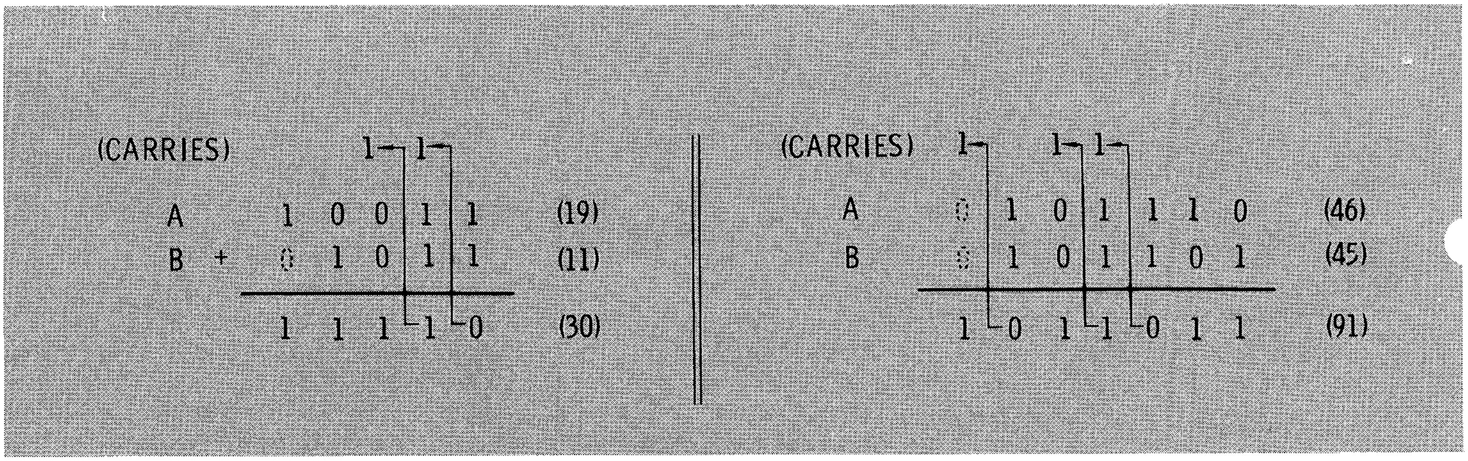


Fig. 2 Two examples of binary addition.

digit of the sum (which will be a "1" or a "0") and "carry" the rest (also a "1" or a "0") to the top of the next column and repeat.

Since we are concerned with only two numbers, no column can be more than three numerals deep . . . number A, number B, and maybe a carry. Study Figure 1 before going further.

The table in Figure 1 can be reduced to four easily remembered sentences:

1. If a column has no "1's" in it, the sum is 0.
2. If a column has one "1" in it, the sum is 1.
3. If a column has two "1's" in it, the sum is 0 and carry a 1.
4. If a column has three "1's" in it, the sum is 1 and carry a 1.

Figure 2 shows two examples of binary addition.

VALUE	8	4	2	1	
0	0	0	1	1	"EXCESS THREE CODE" "XS3"
1	0	1	0	0	
2	0	1	0	1	
3	0	1	1	0	
4	0	1	1	1	
5	1	0	0	0	TO CONVERT ORDINARY BCD TO XS3, ADD 0011 (THREE)
6	1	0	0	1	
7	1	0	1	0	
8	1	0	1	1	
9	1	1	0	0	

VALUE	4	2	2	1	
0	0	0	0	0	
1	0	0	0	1	ONE FORM OF A 4-2-2-1 CODE
2	0	0	1	0	
3	0	1	0	1	
4	0	1	1	0	
5	1	0	0	1	THERE ARE OTHERS
6	1	0	1	0	
7	1	1	0	1	
8	1	1	1	0	
9	1	1	1	1	

Fig. 3 Two other forms of BCD.

100 Take-Away 1 Equals 11

An old rule of grade-school arithmetic says: to subtract, change the sign; then add. Changing the sign of a binary number is simple. Change all the 1's to 0's and all the 0's to 1's. This is called "complementing" the number. For example:

```
      1100 (twelve)                          1100
                      IS THE SAME AS
minus 0101 (five)                       plus 1010
                                  ┌ carry ← 0110
                                  └─────────→  1
                                             0111 (seven)
```

Note that if the addition in the most significant column creates a "carry" it is added to the least significant column. This is called an "end-around" carry.

Sometimes the number being subtracted is larger than the one being subtracted from. Whenever this happens, there will be no carry from the last column. The answer, which is obviously negative, will have its ones and zeros reversed and must be re-complemented before it can represent a true magnitude. For example:

```
      0101 (five)                            0101
                      IS THE SAME AS
minus 1100 (twelve)                     plus 0011
                          NO CARRY ←——— 1000
                   1000 complemented is 0111 (seven)
```

The rule for subtraction can be stated in three easily remembered sentences:

1. Invert (complement) the number to be subtracted, then add it.
2. If there is an "overflow" (a carry from the last stage) the answer is positive, and one (the carry) must be added to the sum.
3. If there is *no* carry, the answer is negative and must be complemented.

When numbers are present in Binary-Coded-Decimal form (BCD) some tricky circuitry must be employed if binary adders are to be used. Alternatively, the BCD numbers can be converted to pure binary, added, (using readily available binary adders), then reconverted to BCD. MSI circuits are available which convert BCD to binary and vice versa.

There's BCD And There's BCD

In previous articles, only one form of BCD was mentioned. In it, each "decimal place" was "worth" twice as much as its right-hand neighbor. It is sometimes called "8-4-2-1" BCD. There *are* other forms. This is possible since there are only ten decimal numerals and sixteen combinations to choose from. Two popular forms are tabulated in Figure 3.

Each of the two forms shown in Figure 3 have two characteristics in common:

If you were to serially transmit all ten of the decimal numerals, you would have transmitted twenty "ones" and twenty "zeroes." This equality is an advantage in some transmission systems.

Each pair of numerals that add up to nine are "mirror images" of each other. This is very useful when performing BCD arithmetic.

There are many other special-purpose BCD "codes."

Show And Tell

The BCD format (usually the 8-4-2-1 kind) is widely used in conjunction with numeric display devices (read-outs). Two types of number display devices have already found wide application in broadcast equipment, and another fairly new type is coming into use. They are:

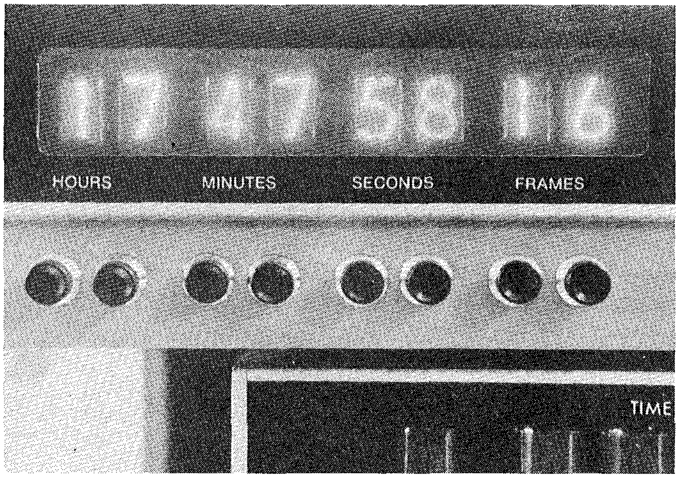


Fig. 4 The nixie tube read-out display.

1. The "Nixie" tube. (See Figure 4.) This is a gas discharge tube (remember the VR-105?). It has one anode, returned to a 200 Volt (approximately) positive supply through a current limiting resistor. There are ten cathodes, each one made of thin wire and shaped like a numeral. One cathode at a time is grounded and the others left open circuit. The ionized gas immediately surrounding the wire of the grounded cathode glows an orange color and forms a highly visible numeral.

MSI devices are available which accept BCD input and have ten outputs... one for each cathode. They are designed to withstand the voltage to which the open cathodes rise (about 70 Volts).

2. The "seven segment" display. (See Figure 5.) This display can also be made of seven gas discharge elements, but usually uses incandescent filaments. MSI devices are available for these, also—BCD in, seven lamp driving outputs, and logic to light up the right ones.

3. The light-emitting diode array. The diodes in the array operate from the typical +5 Volt supply used with DTL and TTL logic devices. Similar ones are available having a full 35-diode array (7 tall by 5 wide), with which they can portray not only numbers, but the letters of the alphabet and punctuation marks as well. The version shown in Figure 6 is a partial array and can form only the numerals, a minus sign and a decimal point, but inside the dual-in-line package along with the diodes is a gated four-bit memory and the necessary logic to turn on the right diodes. The input is BCD, of course. Like the seven segment display, all parts of the display lie in the same plane. Like the Nixie tube, the numerals approach the shape of normal printed numerals.

Memories

Electrical arithmetic is complex enough that it is economical to *store* numbers somewhere, fetch them when needed, perform the arithmetic, and *store* away the answer until it is needed. When we wish to see a number displayed, our eye must see it long enough to recognize it. The display device is usually fed by some sort of memory which contains the result of arithmetic performed some time ago. Meanwhile, the arithmetic circuits are busy with another problem.

The important characteristics of a memory device are these:

1. Volatility. It is volatile if it loses its mind when the power is turned off.

It is non-volatile if it doesn't.

Examples: flip-flops are volatile.

Delay lines are volatile.

Magnetic cores are non-volatile.

Punched paper tape is non-volatile.

2. Read-out—destructive or non-destructive. Read-out is destructive if in the course of finding out what is in the memory you must erase it.

Examples: Destructive—most magnetic core memories.

Non-destructive—almost everything else.

3. Access—random or serial. Random access is like that of a pigeon-hole desk—one "reach" fetches any one item. Serial memory is like a lazy-Susan serving

tray. You *may* luck out and find what you want in front of you. Then again, you may have to turn it all the way around. Random access is faster. Serial access is cheaper.

Examples: Magnetic cores are random access.

Most flip-flop arrays are random access.

Various kinds of tapes and punched cards are serial access.

A track on a magnetic drum or disc is serial access.

A shift register, like a delay line, is used as a memory element by connecting the output to the input and pumping the contents around and around. To enter a new bit, the output-input connection is broken when the old bit appears at the output, the new bit allowed to enter the input, then the connection re-established. To keep track of where a bit is, a counter may be used to count shift pulses.

Shift register memories are of two types—dynamic and static. The static type essentially consists of a number of J-K flip-flops connected in tandem. Each clock pulse shifts everything one step along the chain. Information can stay in place as long as power is applied. The dynamic kind stores information in the form of capacitive charges which are shifted along by clock pulses and "re-charged" in the process. If the clock stops, the information will die away. Like an airplane, it must operate above some minimum speed.

The "Read-Only" Memory

A read-only "memory" is a memory only in the sense that it remembers how it was arranged at the factory. Imagine a tic-tac-toe grid having 32 x 32 lines. There will be 1024 cross-points. Initially each cross-point is conductive. The customer specifies which of the 1024 bits shall be ones, and which zeros. The manufacturer then selects the specified bits and carefully zaps them open. The "memory" then contains a permanent bit pattern. Read-only devices are capable of great density, currently as many as 8196 bits in one dual-in-line package. They are very useful in situations where the logic required is unchanging, as in code conversion, character generation, desk calculator instruction sequences, look-up tables, etc.

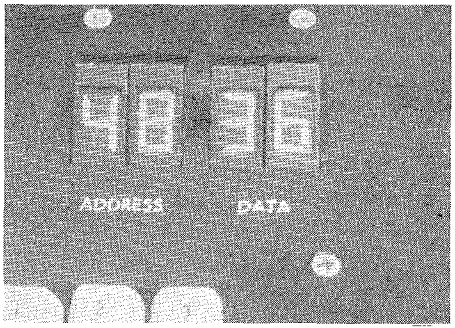


Fig. 5 The "seven segment" display. All numbers look square.

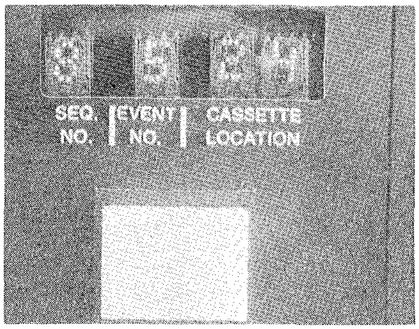


Fig. 6 The light-emitting diode array.

Transmission

Digital information is seldom sent (for more than a few feet) using a wire for each bit. For distances up to a few thousand feet, a well-terminated twisted pair might be used, serially transmitting one bit at a time. For longer distances, for transmission over telephone circuits or for recording on a single track of a tape recorder, data is "encoded," or modulated on a carrier. Amplitude modulation of a tone has been used for years. Pure FM, sometimes called FSK (frequency shift keying), is almost as old. A number of phase-modulation methods have come into use and three popular ones are outlined in Figure 7.

One common characteristic of serial transmission schemes is the use of a unique progression of bits, which never occurs and *cannot* occur during the transmission of data. This progression, or "pattern," marks the boundary between one "word" and the next and is used to synchronize receiving equipment.

The Society of Motion Picture and Television Engineers has recently proposed a "code" for the digital recording of time (in hours, minutes, seconds and TV frames), as well as 32 bits of extraneous information.[1] The "synchronizing interval" is this progression: 0011111111111101.

The data is arranged so that twelve ones never happen *except* in the sync interval. Receipt of twelve ones followed by 01 indicates forward tape travel. Twelve ones followed by 00 indicates reverse motion.

The time is encoded in eight 8-4-2-1 BCD digits, utilizing 32 bits. The 32 "spare" bits may be used in any manner the user desires. All told, there are 32 "time code" bits, 32 "spare" bits, and 16 bits in the "sync interval," which totals 80 bits. The modulation format used is bi-phase mark. The highest *frequency* developed in the process of encoding is 80 x 30 = 2400 Hz (2000 Hz in 50 Hz countries), easily accommodated on the cue track of VTR's or other recorders. One complete "word" is encoded and recorded each TV frame.

A unique feature of this time code is that it may be used to encode the proper time of day even though the frame rate is locked to a color sync generator, whose frame rate is *not* 30 per second, but is 0.1 percent slow. To correct the time count, the time counter (clock) may be caused to "skip over" two frame counts each minute, except each tenth minute. The Europeans are lucky...their color frame rate turned out to be *exactly* 25 frames per second.

"Frame numbers" recorded on a tape permit it to be controlled by today's complex tape editing equipment.

The transmission of data from machine to machine, often over telephone lines, has created the need for standardization of transmission format. The American Standard Code for Information Interchange (ASCII) is a standard transmission format.[2] It consists of eight data bits. Seven of these (permitting 128 combinations) are used to identify all the numerals, upper and lower case letters of the alphabet, the common punctuation marks, and some non-printing instructions or commands, like "who are you?" and "rub out." The eighth bit may, if desired, be used to cause the total number of "ones" to be consistently either an odd or even number, so that the received "word" may be checked for accuracy. Checking for accuracy of transmission in this manner is called a "parity check."[3] If *one* bit is altered by noise or other disturbance a parity check will reveal it and warn of error. *Two* bits in error in the same word (or any even number) would not be revealed. It is very unlikely that two bits in the same word will be so affected.

Logic devices are no longer the exclusive domain of the computer people. They are now finding application in broadcast equipment, manufacturing machine control, desk calculators, and are even used in some toys. The two-way capabilities of Cable TV opens the possibility of remote meter-reading and home-owner access to large central computers. The TV receiver can image letters and numerals just as well as it can image a bar of soap. You can be sure that when these products of the "data age" intrude into everyday life, the man who is *already* familiar with the techniques used can command a premium.

Much has gone on and much is still going on. That which has gone on for a while can be found in books. That which is still going on is reported in magazines. Read both.

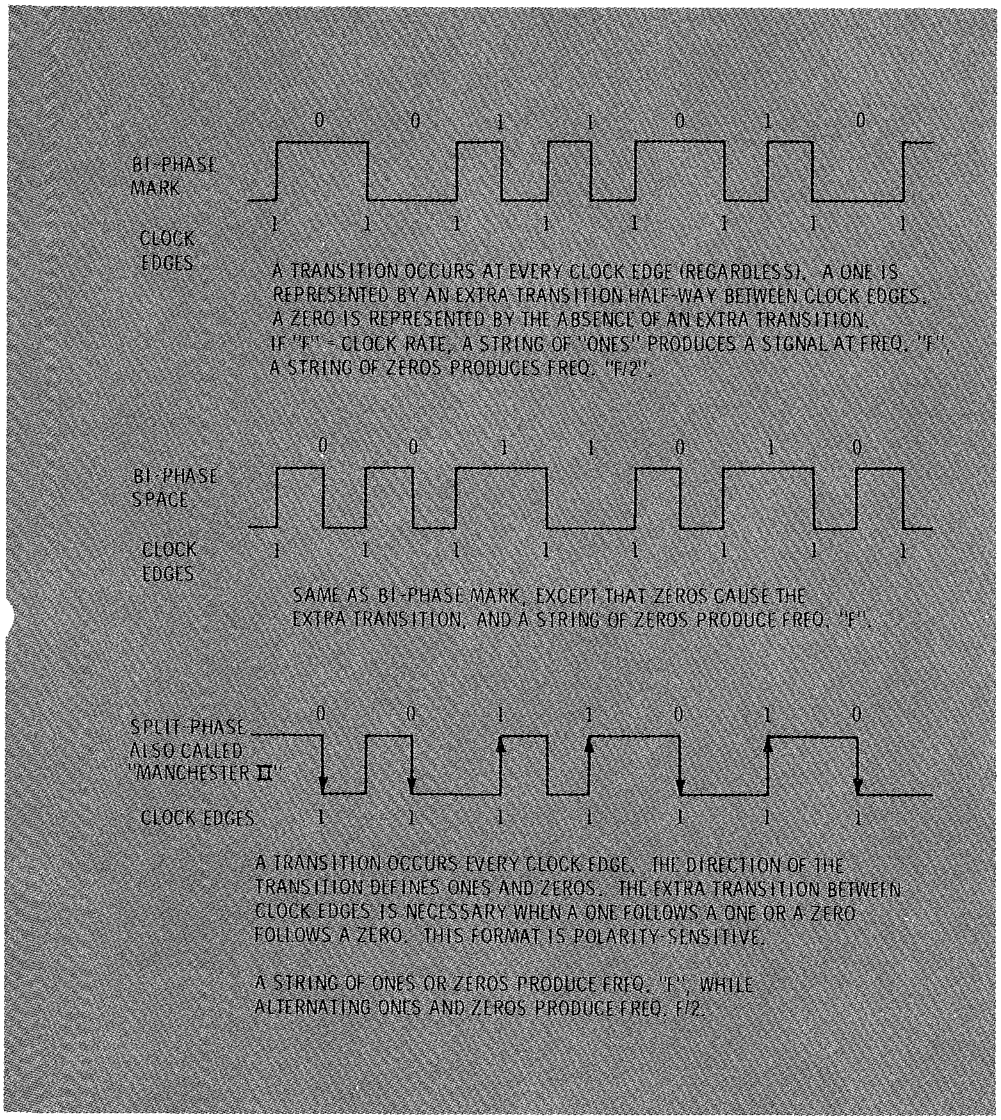


Fig. 7 Three phase modulation formats for data transmission. These are examples of "self-clocking" formats, because a transition always occurs at "clocktime". This is an advantage when demodulating a signal of variable rate, such as playback of a tape during high speed search.

Footnotes

1. "Standardization for time and control code for video tape and audio recorders," Ellis K. Dahlin; *SMPTE JOURNAL* Vol. 79, No. 12, Dec. 1970, p. 1086-1088.
2. American Standard Code for Information Interchange, ANSI X3.4—1968 (AMERICAN NATIONAL STANDARDS INSTITUTE).
   Also see "Data Communication," A. J. Boyle, *THE ELECTRONIC ENGINEER*, Aug. 1971, p. DC-9.
3. a. "Generalized parity checking," Harvey L. Garner, *IRE TRANSACTIONS ON ELECTRONIC COMPUTERS;* Vol. EC-7, No. 3, September 1958, p. 207-213.
   b. "Error Detecting Codes," R. W. Hamming, *BELL SYSTEM TECHNICAL JOURNAL*, Vol. 26, April 1950, p. 147-160.
   c. "Error-correcting Codes," W. W. Peterson—Wiley, 1961.

Other Reading

"Ultrasonic delay lines used to store digital data," A. H. Meitzler, Bell Labs Record, Vol. 42, No. 9, Oct. 1964, p. 315-319.

"A magnetorestrictive delay line shift register," L. E. Hargrave, *IRE TRANSACTIONS ON ELECTRONIC COMPUTERS*, Vol. EC-10, No. 4, Dec. 1961, p. 702-708.

"Digital Computers—Storage and Logic Circuitry," H. W. Sams, #20131.

"Fundamentals of Digital Magnetic-Tape Units," H. W. Sams, #20580.

"Servicing Digital Devices," by Jim Kyle, H. W. Sams, #20618.

AMPEX

Ampex Corporation, Audio-Video Systems Division
401 Broadway
Redwood City, California 94063

LITHO IN U.S.A.—V410—4-72