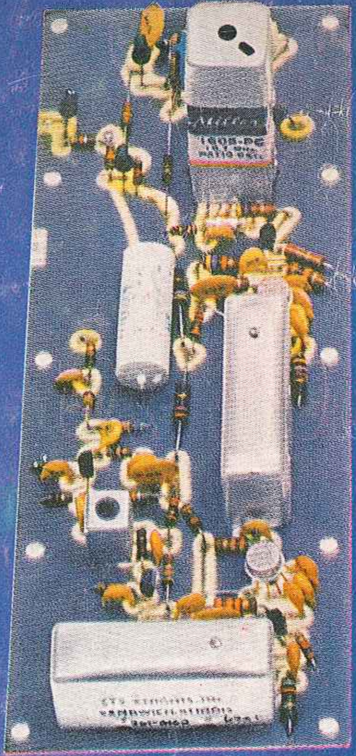
60c ■ MAY 1970

Radio-Electronics®

BUILD "The Ultimate"

R-E's NEW STEREO FM TUNER



EXPERIMENT WITH

Computer logic circuits

10 Emitter-follower circuits

S FEATURES

x™ transistor agc

raphy 1970

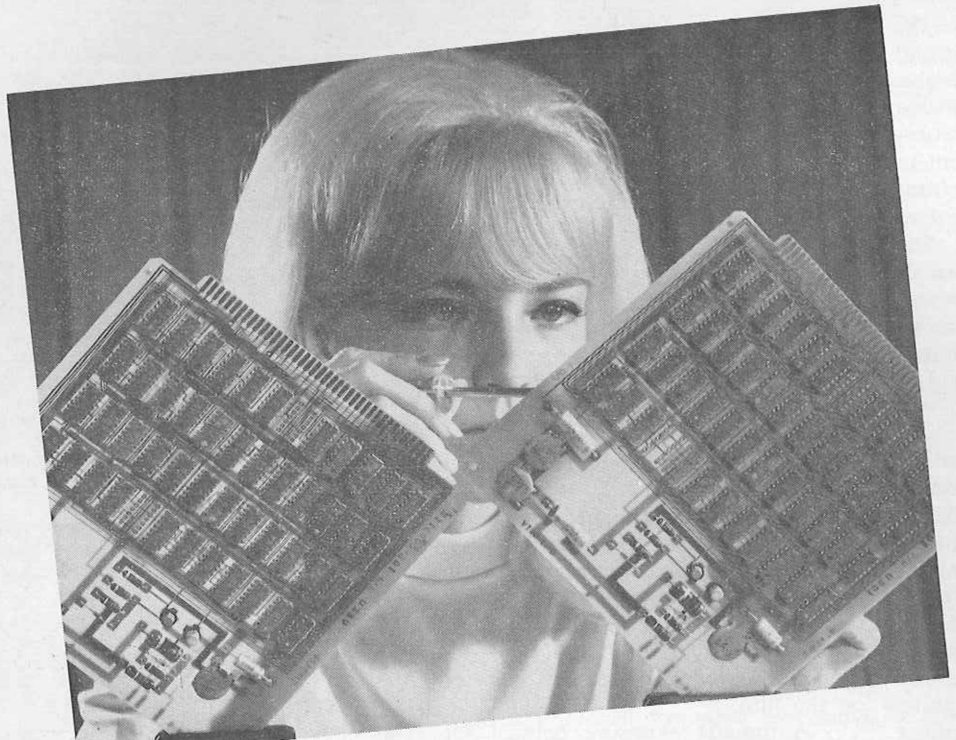
BUILD ONE OF THESE

Electronic flash tester

Pulsed darkroom timer

Horizontal efficiency tester

A GERNSBACK PUBLICATION



WHILE MODERN DIGITAL COMPUTERS use thousands of transistors, diodes and other components, the number of circuit *types* are few. Logic functions are performed by repeatedly using certain circuits, in series or parallel combinations, to perform the required tasks of counting, sorting and routing signals, and of modifying binary representations as needed.

Since logic circuits are not complex, they are easily assembled. Thus, if you build your own you not only learn the over-all circuit function, but get better acquainted with the function of individual components and the logic which applies to their use. With chips and modules it is almost impossible to acquire a true understanding of complete circuit makeup and its various functions.

Switches and gates

Most computer logic circuitry has switching functions. Signals are gated and routed to conform to preset conditions. Since a switch is either closed or open the two states are related to 0 and 1 (the binary system). Steady-state or pulse-type signals can be used to represent logic 0 or logic 1. If a positive voltage is selected as representing logic 1, then the absence of a voltage, or a negative voltage, represents logic 0. Similarly, if a negative voltage represents 1, logic 0 is either a positive voltage or no voltage. In the initial design of a computer, either a positive or negative signal is selected to represent logic 1.

Diodes have the advantage of forming simple and inexpensive logic circuitry, though transistors provide signal gain where needed and can also alter logic by phase inversion, as shown later. If you construct the circuit shown in Fig. 1 you can observe the function of two basic, and important, logic gates. Diodes other than 1N34 types can be used as long as they have a good front-to-back ratio. Use a vtvm or a 20,000-ohms-per-volt meter for output signal observations.

While pulse-type signals would normally be used for this circuit, its function can be readily observed with steady-state signals. The type logic obtained with this circuit depends on whether we select a negative or a positive signal to represent 1. Initially, let's assume that a positive signal is logic 1 and a negative signal (ground) is 0. Set both spdt switches so inputs A and B to the diodes are at ground-voltage level. This now represents 0 input at A and B.

Now throw the A input switch to the positive-signal input. Note that a reading is obtained at the output, representing a 1 output (positive voltage). Place the A switch at ground again and repeat the procedure for the B switch. Note that an output is obtained again. Note also that an output is obtained when *both* switches engage a positive input.

You have now observed the function of what is known as an OR gate (or switch). The logic is: If either A *or* B *or* both inputs are applied, an output is obtained. This OR circuit has a special symbol which is also shown in Fig. 1.

Easy-to-Build Computer Logic Circuits

by **MATTHEW MANDL**
CONTRIBUTING EDITOR

All photos on these pages are of the RCA Spectra 70 computer system. Courtesy of RCA.

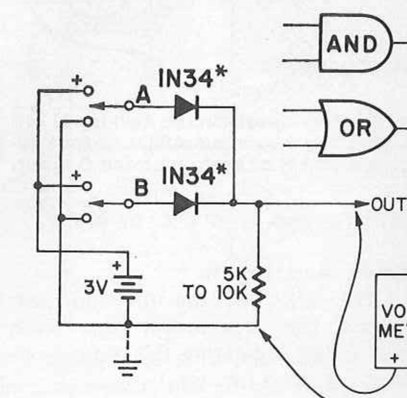
Logic circuits are the basic elements of a computer. Discover how they work by making your own—it's easy!

If we invert the logic 1 representation, we form an entirely different logic gate. Assume *negative* voltages represent logic 1 and place each spdt switch to the plus position for an initial logic 0 input at both A and B. The meter at the output reads a steady-state positive voltage, now representative of logic 0. Since a positive voltage appears at the input to each diode, both conduct.

If *either* the A or B input is changed for a logic 1 input (switching from positive- to negative-ground input), no appreciable voltage change is felt at the output, because one diode will still conduct. If, however, both A *and* B inputs have a negative voltage applied (logic 1's) both diodes stop conducting and the output voltage drops to the ground (negative) level, representing a logic 1 output of the circuit.

Thus, for this gate to produce an output when negative voltages represent logic 1, both the A *and* B inputs must be applied. Hence, this is called an AND gate or switch. Its special symbol is also shown in Fig. 1. This is sometimes also called a *coincidence* gate, because simultaneous inputs must be applied to obtain an output signal.

For more thorough understanding of the OR and AND functions reverse the diodes and the battery potential of Fig. 1. Now if you repeat the same test procedures you will find that an OR gate is formed when *negative* signals are used for logic 1's and an AND gate function is obtained for conditions where *positive* signals are logic 1's.



* OR EQUIVALENT

Fig. 1—OR function is shown when A or B or both inputs (switch to +) provide an output. AND function, or coincidence gate, requires both A and B switches in negative position for an output signal (negative voltages for a logic 1).

In-depth analysis of factors relating to switching-circuit logic is done by using a special math known as Boolean algebra. In this system the OR and AND functions are indicated by using a plus sign (+) for the OR logic and a multiplication sign (•)

for the AND logic. These signs are called *logical connectives* and if the circuit shown in Fig. 1 is used as an OR gate, the Boolean expression is: A+B, where the plus sign is read as OR (A *or* B). For the AND function, we can show it as A·B, or use the algebraic method for displaying the multiplication, by placing the variables close together as AB.

More than two inputs can be used in the OR and AND gates of Fig. 1. If three inputs are used, for instance, the OR expression would be A+B+C (assuming our third input is designated as a C). Other designations can, of course, be used. If we have a three-input AND circuit, with X, Y, Z inputs, the expression that would be used is X·Y·Z, or XYZ.

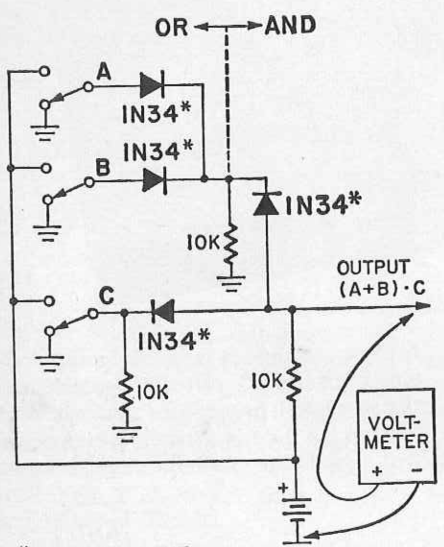


Fig. 2—Two- level circuit: two-input OR followed by two-input AND. Output requires A and B or both, plus the C input.

Tables and levels

The logic of the OR and AND switches can be shown in "truth table" form, indicating the output obtained for given inputs:

OR gate		AND gate	
A+B+C	Output	A·B·C	Output
0 0 0	0	0 0 0	0
0 0 1	1	0 0 1	0
0 1 0	1	0 1 0	0
0 1 1	1	0 1 1	0
1 0 0	1	1 0 0	0
1 0 1	1	1 0 1	0
1 1 0	1	1 1 0	0
1 1 1	1	1 1 1	1

These tables show that an output is obtained for any one or more inputs to the OR switch, but an output is obtained from the AND gate only when all inputs are present.

One logic gating system can follow another to obtain specific functions, as shown in Fig. 2. This circuitry is a *two-level* type and could also consist of an AND circuit feeding an OR circuit. For the one shown we have a two-input OR gate followed by a two-input AND switch. Thus, since coincidence must occur at the AND gate inputs, we must have the C input *and* at least one or both of A and B. In this system we are using positive voltages for the logic 1's. Again, 1N34 diodes (or equivalent) are used.

Two-level logic in symbolic form is shown in Fig. 3. The OR-AND combination which produces an output of (A+B)C applies to Fig. 2, and a C input must be present with either A or B (or both) for an output. The single AND gate feeding the OR circuit with an output of (AB)+C is obtained if we reverse the diodes of Fig. 2 and also the battery potential, while still using positive voltages to represent logic 1. Now, C alone will produce an output, or A and B in coincidence, as well as all three inputs at the same time.

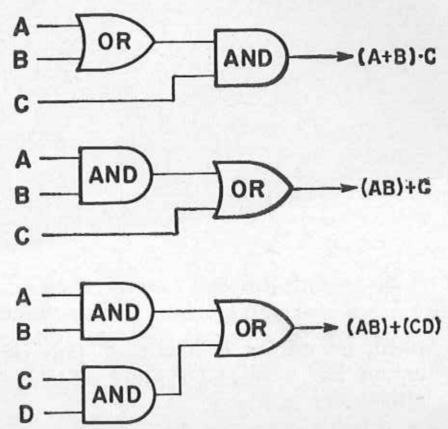


Fig. 3—OR-AND two-level logic symbol (top), AND-OR (middle) and the AND-AND-OR (bottom). Text shows (A+B)C truth table.

Fig. 4-*a*—(right) Emitter-follower circuit keeps input signal phase and has no voltage gain. *b*—NOT circuit inverts phase.

For Fig. 2 the C input could be replaced by another OR circuit to produce the logic (A+B)·(C+D), where at least one input from each OR circuit must be used to produce an output. With reversed logic we have the two AND gates feeding an OR circuit, as also shown in Fig. 3. Now the logic is (AB)+(CD). Now either A and B or C and D will give an output.

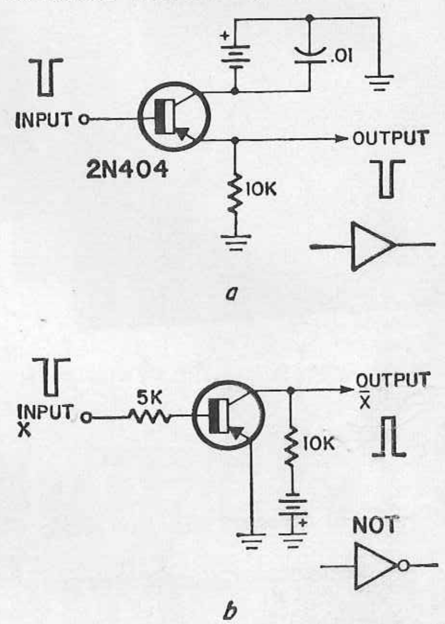
Truth tables can also be constructed for these circuits, showing the logic which applies. The following example applies to the (A+B)C system:

(A+B) C	Output
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

This table shows that only three input combinations produce an output: A and C, B and C, or all three inputs.

Transistor logic

Transistors, while more costly than diodes, provide amplification as well as additional logic functions not obtainable with diodes, For the circuit in Fig. 4-a, the output is obtained from the emitter resistor, hence the circuit is called an *emitter follower* (and compares to the tube-type cathode follower). If a negative signal (pulse or dc voltage) is applied to the base input line, the necessary forward bias is applied to permit the transistor to conduct.



Now, electron flow is from the negative terminal of the battery to the collector and emitter and down through the output resistor. Thus, a negative signal is developed at the output. While such a circuit has no voltage gain, there is signal current amplification. Note that the phase of the output signal "follows" that of the input—hence the term "emitter follower." The triangular symbol for amplifier is used as shown.

For the circuit in Fig. 4-b the output is taken from across the collector resistor. Electron flow is up through this resistor, producing a positive-signal output as shown. When we have such a phase reversal of the signal between input and output, the logic which applies relates to the output signal *not* being in phase with the input signal. Hence this is known as a NOT circuit.

A line over a letter indicates the NOT function, as $\bar{A}$ for "NOT-A" and $\overline{X+Y}$ for NOT-X OR NOT-Y. A triangular symbol is used as with the amplifier, but a small circle denotes the NOT function. This logic function is useful when it is necessary to change a logic 1 to logic 0 or vice versa in computer circuitry design.

Diodes can be combined with transistors as shown in Fig. 5. Two diodes provide an OR gate input to an emitter follower in Fig. 5-a. Additional diodes could be included to provide more input lines. Such circuitry is known as diode–transistor logic and sometimes referred to as a DTL circuit. For the circuit in Fig. 5-b the output is from the collector and hence a NOT function is also obtained. Now we have the logic condition of NOT-OR, is commonly termed a NOR circuit. Note the OR-circuit symbol now has the small circle denoting the logic inversion function.

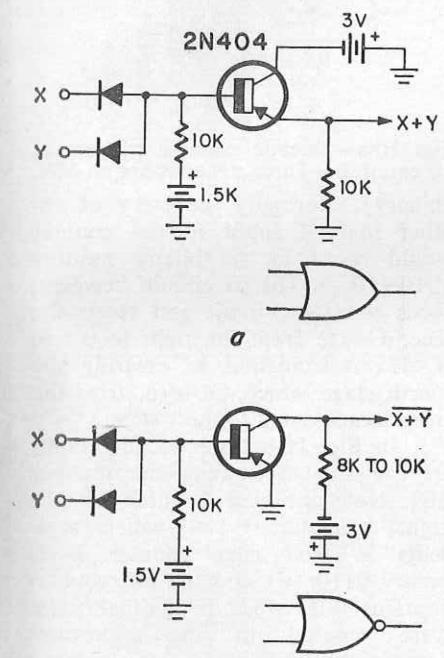


Fig. 5-*a*—DTL (diode-transistor logic) OR circuit. *b*—NOT-OR (NOR) circuit.

For the circuits in Fig. 5, a reverse bias is applied to the base circuit, thus cutting off transistor conduction. A negative signal above −1.5V applied at either (or both) the X and Y inputs overcomes the reverse bias and causes conduction. Electron flow up through the collector resistor (Fig. 5-b) develops a voltage drop across it, producing a positive-signal output. The diodes could be replaced with resistors for resistor–transistor logic (RTL) though diodes work better.

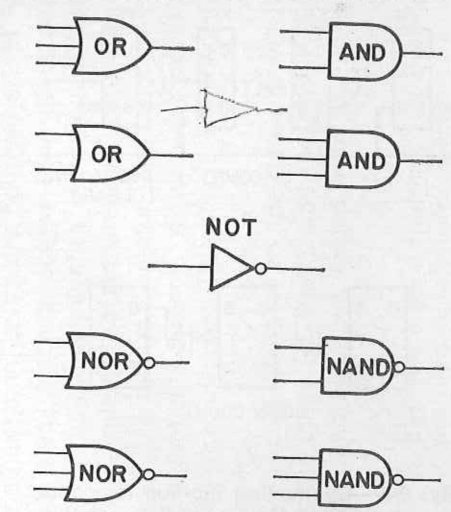


Fig. 6—Symbols for two- and three-input OR, NOR, AND, NAND gates.

Fig. 7—Below is a bistable flip-flop, which produces output only with input.

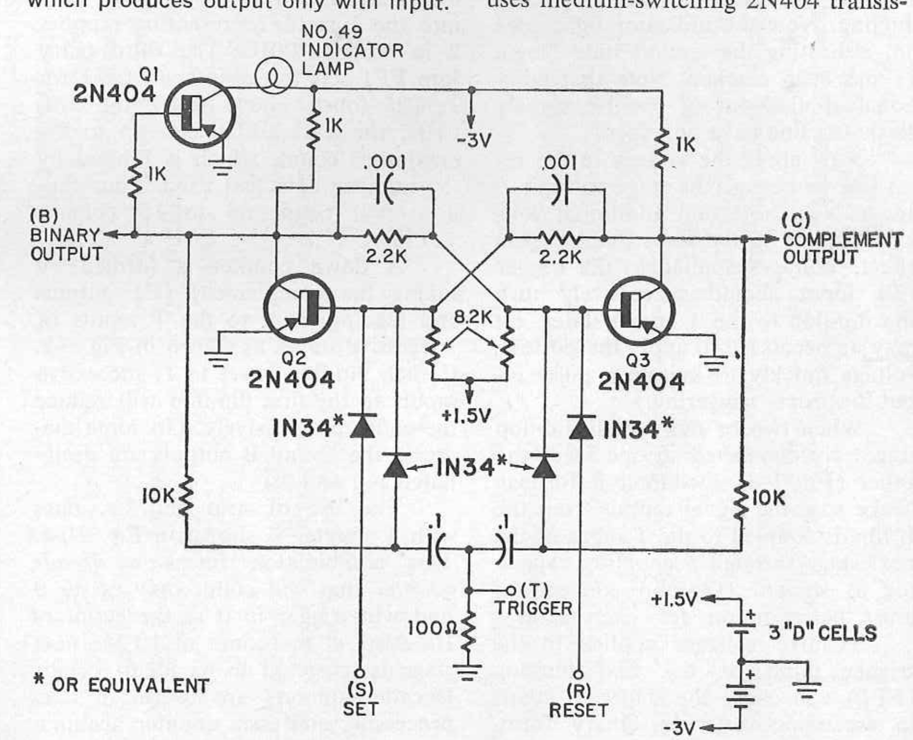
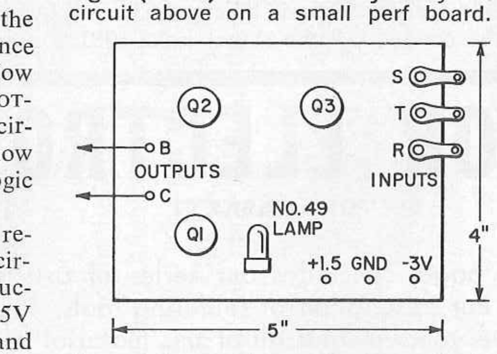


Fig. 8 (below)—Practical way to lay out circuit above on a small perf board.



As with the diode logic gates discussed earlier, the AND function can also be obtained from these circuits. For Fig. 5-b the result is a NOT-AND logic function, known as a NAND circuit. The symbol for the NAND circuit is shown in Fig. 6, where others are also shown for comparison.

The flip-flop

The flip-flop circuit performs many functions in a digital computer. A string of flip-flop circuits forms counters, temporary storage registers, accumulators (accumulating counts as basic addition), scalers (a form of binary division), and switches (providing "on" and "off" conditions as needed).

The flip-flop is a *bistable* device with one state representing logic 0 and the other logic 1. It is not a free-running oscillator and produces an output signal change only when an input signal is applied.

The flip-flop shown in Fig. 7 uses medium-switching 2N404 transistors and operates with only three D-cells as shown. It has an extra transistor to drive the indicator lamp to show the logic 1 state of the flip-flop. The circuit can be built on a 4 x 5-inch panel as shown in Fig. 8. The output lines should be at the left so when several flip-flops are connected in series they will show binary magnitudes from right to left to conform to the arithmetic place system.

Note the polarity of the base diodes. These restrict the signals to positive voltages for representing logic 1. When the flip-flop is in the logic 0 state, transistor Q2 conducts while Q3 does not. At this time the binary (B) output line is less negative than the complement (C) output line, hence it can be considered positive with respect to the other.

When power is first applied to a flip-flop it may assume either the 0 or 1 state, depending on which transistor conducts initially. Hence all computers reset all flip-flop stages to 0 immediately after power is applied. If your flip-flop indicator light goes on after you connect batteries you will have to reset it.

Use a jumper connected to the positive terminal of the battery and apply a voltage momentarily to the reset (R) input line. If the circuit has been assembled properly, the indicator light will go out. The positive voltage applies a reverse bias to the base of Q3 and causes it to stop conducting. Now the high negative potential at the *collector* of Q3 is felt at the *base* of Q2, permitting the latter to conduct.

Apply the positive voltage to the set (S) line and Q2 will stop conducting. Now the indicator light goes on, indicating the second state (logic 1) has been reached. Note that additional applications of positive signals to the set line have no effect.

Now apply the voltage to the reset line to change the stage to logic 0 again. Now note that additional voltages applied to the reset line have no effect. Voltages applied to the trigger (T) input should successively turn the flip-flop to the 1 and 0 states. (It may be necessary to apply the positive voltage quickly to imitate a pulse input for proper triggering.)

When two or more such flip-flop stages are connected so one feeds the other (Fig. 9-a) a counter is formed. Make sure the signal output from the B line is coupled to the T input of the next stage through a coupling capacitor as shown. (Ground connections must be common for each stage.)

Positive voltages applied to the trigger input of the first flip-flop (FF1) will cause the stages to count in ascending order in binary form. Thus, one signal entry into the first flip-flop will set the first stage to 1, with the indicator light on, representing 0001. Another pulse entry at FF1 shuts off the indicator light of the first state and the second stage goes into the 1 state, representing number 2 in binary: 0010. The third entry into FF1 sets the counter to 0011 for 3. The fourth shows 0100, the fifth 0101, the sixth 0110, etc. up to the maximum count, which is limited by the number of stages used. Four flip-flops will count up to 15 (binary 1111).

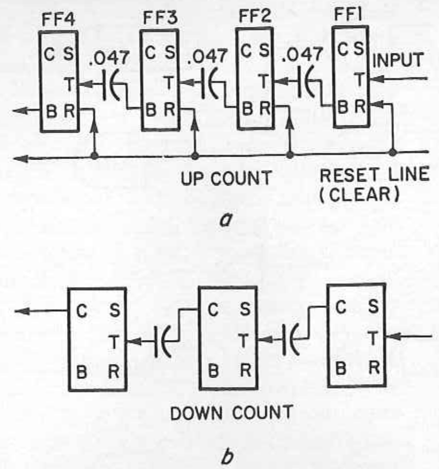


Fig. 9-a—**Connecting flip-flop** for count up. b—Count-down flip-flop circuit.

A down counter is formed by taking the complement (C) outputs and feeding them to the T inputs of successive stages as shown in Fig. 9-b. If each flip-flop is set to 1, successive inputs to the first flip-flop will reduce the count progressively. (In some diagrams the C and B outputs are designated as 1 and 0.)

The use of AND and OR gates with a counter is shown in Fig. 10-a. This combination forms a *decade counter* that will count only up to 9 and will trigger to 0 at the count of 10. Also, at the count of 10 the next stage is triggered to its logic 1 state. Decade counters are useful in data processing and each counter accumulates counts in base-10 fashion but still operates in basic binary mode. At the count of 9 the stages hold 1001 (binary). Normally the entry of another logic 1 input to the counter would result in the binary number 1010 (10_{10}). The OR circuit, however, feeds the AND circuit and resets the second stage from the right to 0 and is also instrumental in clearing the fourth stage which, in turn, trips the next decade count to the 1 state.

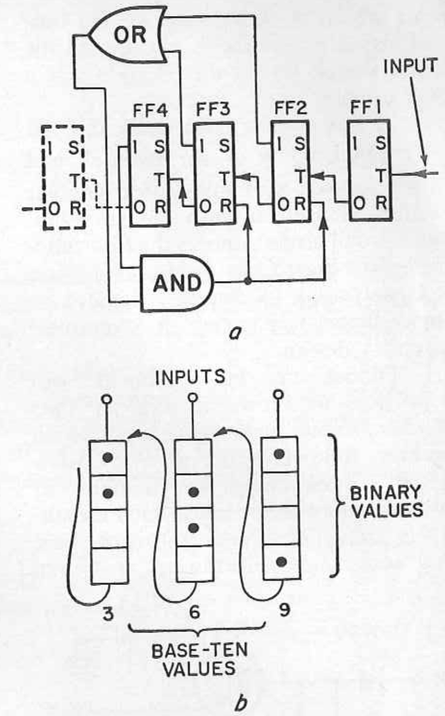


Fig. 10-a—**Decade counter** triggers at 10 count. b—Three decades set on 369.

In Fig. 10-b three decade counters are shown, holding the number 369. Note the first counter (at the right) holds binary 1001 which represents 9. The next counter holds binary 0110 (6) and the left counter contains 0011 (3). If another pulse were entered in the right-most counter, it would trip to 0 and send another pulse to the center counter, changing the latter to 7. Now the count would be 370 and the system would have added an additional 1 which was entered to the existing number. R-E

TOOLS FOR ELECTRONICS

by TOM HASKETT

This issue, starting on the facing page, concludes our series of articles on soldering tools for electronics. It is the third part of our description of soldering tools. This month we will also start a series on nutdrivers. We believe you will find all of this material a handy, practical addition to your R-E Reference Manual.

If you wish, you can purchase a special hardcover binder to keep your Reference Manual pages together. It has a dark blue fabric cover and is gold stamped Radio-Electronics Reference Manual. The cost is $1.00, postpaid. Order from N. Estrada, 17 Slate Lane, Central Islip, L.I., N.Y. 11722.

Iron and pencil accessories

You cannot simply lay an iron or pencil down on the work surface, for it can burn something or injure someone. Most people use a **holder** (also called a **stand** or **cradle**), as shown in Fig. 12. Some have a solder cup at the bottom, so the tip stays tinned.

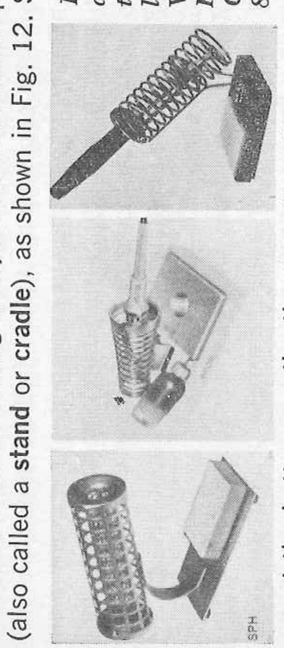


Fig. 12—Three soldering-iron holders that help make life a little easier. Left, Wall SPH. Center, Hexacon 890. Right General Electric 1197-879.

Keeping the tip of a pencil or iron clean is another continuing task. You can use a rag, but if you do a lot of soldering it's convenient to use a **tip cleaner**, shown in Fig. 13. It's merely a wet sponge in a water well, and requires only one hand to clean the tip.

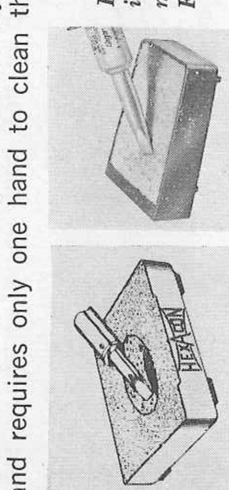


Fig. 13—Two ways to keep soldering iron tips clean. Left is Hexacon model 812. Right is Ungar 400, Kleen-Tip.

Other techniques

One very unusual soldering tool—a gun which contains a coil of solder which you can feed out into the joint—is very useful for one-hand jobs.

If you have to do some soldering in the field, away from an ac power line, you have a problem. The answer is a field iron, clamped across an auto battery. Models are available (Fig. 14) to work on 12 or 24 volts, and they come with battery clamps.


Fig. 14—When you have to solder away from any ac outlets try a battery powered iron. The Weller TCP-12 is shown.

current to the probes. Sometimes a single probe is used together with a ground return connection.

Other specialized soldering methods are used in production work: high-frequency induction soldering, dip soldering, oil-bath soldering, ultrasonic soldering and oven soldering. These highly specialized techniques are seldom used in the electronics fields covered here. R-E

NUTDRIVERS

The tool used for turning a hexagonal or square bolthead or nut is called a **wrench**. In electronics, many machine screws have hexagonal heads, and turning them with a right-angle wrench is tedious. Most people use a **nutdriver** (sometimes called a **socket wrench**) to turn hex-head screws and nuts up to about 3/4 inch in diameter. You can work a nutdriver much faster than a right-angle wrench. (Drivers for square nuts are made, but seldom used in electronics. This article will concentrate, then, on hex-socket nutdrivers.)

The common nutdriver is almost identical to the screwdriver; it consists of the **tip** (or **socket**), the **blade** (or **shaft**), and the **handle**, as you can see in Fig. 1. The socket and blade are forged from tool steel, and the socket is made to a precise size. The nutdriver operates, like the screwdriver, in the same plane as the bolt it's turning. The only difference is the tip, which fits over the bolthead or nut being worked. The tip, or socket, must fit very closely with the nut, or you can't turn it. Thus you must use a separate nutdriver for each size of nut (with one exception, which you'll learn about later).

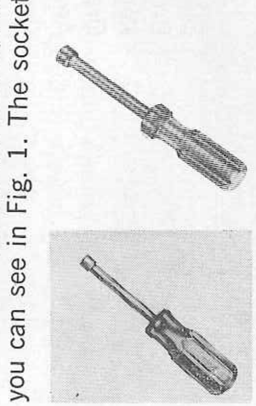


Fig. 1—Common nutdriver is adequately described as a screwdriver wrench. Channellock unit (left) and Crescent driver right are typical examples of available nutdriver types.

Socket sizes and the color code

Hex nutdrivers are available with sockets from 3/32 inch to 7/8 inch in steps of 1/32 inch, measured between parallel faces. Each