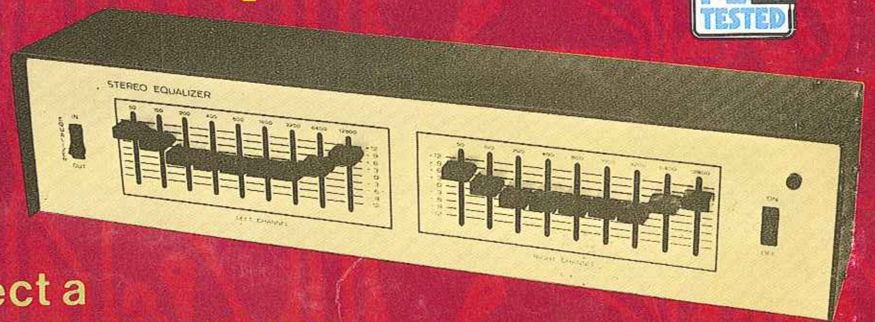
Popular Electronics

MAY 1974/FIFTY CENTS

Build a
9-BAND STEREO EQUALIZER
Overcomes room acoustic problems

PE TESTED



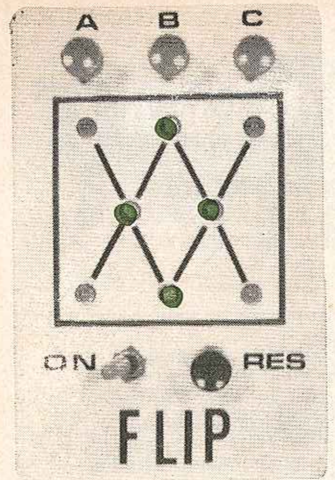
How to Select a
SHORTWAVE RECEIVER
plus English-Language Broadcast Directory

HOW TO BUILD:
- Home Freezer Temperature Alarm
- FLIP- A New CMOS Game
- Micropower Audible Continuity Tester

TEST REPORTS:
- Concord CR-400 4-Channel Receiver
- Koss HV-1 Stereo Headphones
- Pace CB-1023 SSB/AM CB Transceiver
- RCA WR-525 TV Signal Generator/Marker
- Helpmate CD-5 Automatic Garage Door Opener

18101

MAY 1974



BUILD

FLIP

A CMOS GAME COMPUTER

Do you dare challenge a handful of CMOS chips to a game of logic?

BY JOSEPH A. WEISBECKER

HERE IS a fascinating new electronic game based on digital logic. Called "Flip," it will introduce you to some basic computer concepts, pose a number of interesting mathematical questions, and provide a set of challenging puzzles. The puzzles are easily solved, however, when the proper logic sequence is understood.* Using low-cost CMOS logic and LED readouts, construction of Flip is simplified.

Circuit Operation. There are 8 flip-flops (*A* through *H*) connected as shown in Fig. 1. Eight LED indicators on the front panel show the state of each flip-flop (Fig. 2). A trigger pulse applied to a flip-flop reverses its state. Momentary-contact switches *S1*, *S2*, and *S3* provide trigger pulses for flip-flops *A*, *B*, and *C*. For example, pressing switch *S1* will trigger flip-flop *A* so that, if *LED1* was on, it will go off, and vice versa. The transition from off to on also supplies a pulse to trigger flip-flop *D*. The reversal of *D* then supplies a trigger pulse to *F* or *G*.

The circuits in Fig. 1 actually form a number of 2- and 3-bit interacting counters. For example, flip-flops *C*, *E*, and *G* form a 3-bit binary counter that is triggered each time *S3* is pressed. Figure 3 shows how this counter works. Pressing the reset switch, *S4*, sets the *C*, *E*, and *G* lights as shown in the top row. Now, repeated pressing of *S3* causes the lights to go on and off in the 3-bit binary sequence shown in Fig. 3. The combinations of flip-flops *BEH*, *BDF*, *ADF*, etc. also form 3-bit binary counters.

The circuit in Fig. 1 also contains 8 "memory" cells which remember an 8-bit pattern. This pattern (or state) can be modified by the input switches and is displayed by the LED's. A wired-in "program" controls the change-in-state of the device as a function of the previous state and an input switch. Pressing a single input switch 8 times always returns the device to its initial state, thereby demonstrating its ability to count input switch depressions.

In Fig. 1, *IC5* and *IC6* are quad 2-input NAND gates connected to form three set/reset flip-flops for debounce of the switches. Eight D-type flip-flops are provided by *IC1* through *IC4*, which are triggered by a positive-going edge. Flip-flops *A*, *B*, and *C* are triggered directly by the three debounce flip-flops. Flip-flops *D*, *E*, *F*, *G*, and *H* are each triggered by transitions of other flip-flops. The capacitance-resistance combinations differentiate the outputs of these flip-flops to form positive pulses. For example, *C1-R1* and *C7-R7* differentiate the positive-going not-Q outputs of *A* and *D* to feed an OR gate formed by *D1* and *D7* and trigger flip-flop *F*. Trigger pulses for *D*, *E*, *G*, and *H* are derived in a similar manner.

Integrated circuits *IC7* and *IC8* are hex-inverting buffers used to drive the displays. Resistors *R22* through *R29* were chosen to limit the LED current to about 7 mA. Any LED that provides reasonable brightness for this current can be substituted—possibly reducing the cost. Resistors *R22-R29* can also be reduced in value to increase the brightness of the LED's; but this loads

**Flip is an electronic version of a plastic computer game called "Think-A-Dot" made by Edu-Cards Corp. An article entitled "Mathematical Theory of Think-A-Dot" in the Sept.-Oct. 1967, issue of Mathematics Magazine provided a detailed analysis of the game. The original Think-A-Dot instruction book also provided an extensive discussion of the device, with methods for demonstrating counting, adding, and subtracting of 8-bit binary numbers.*

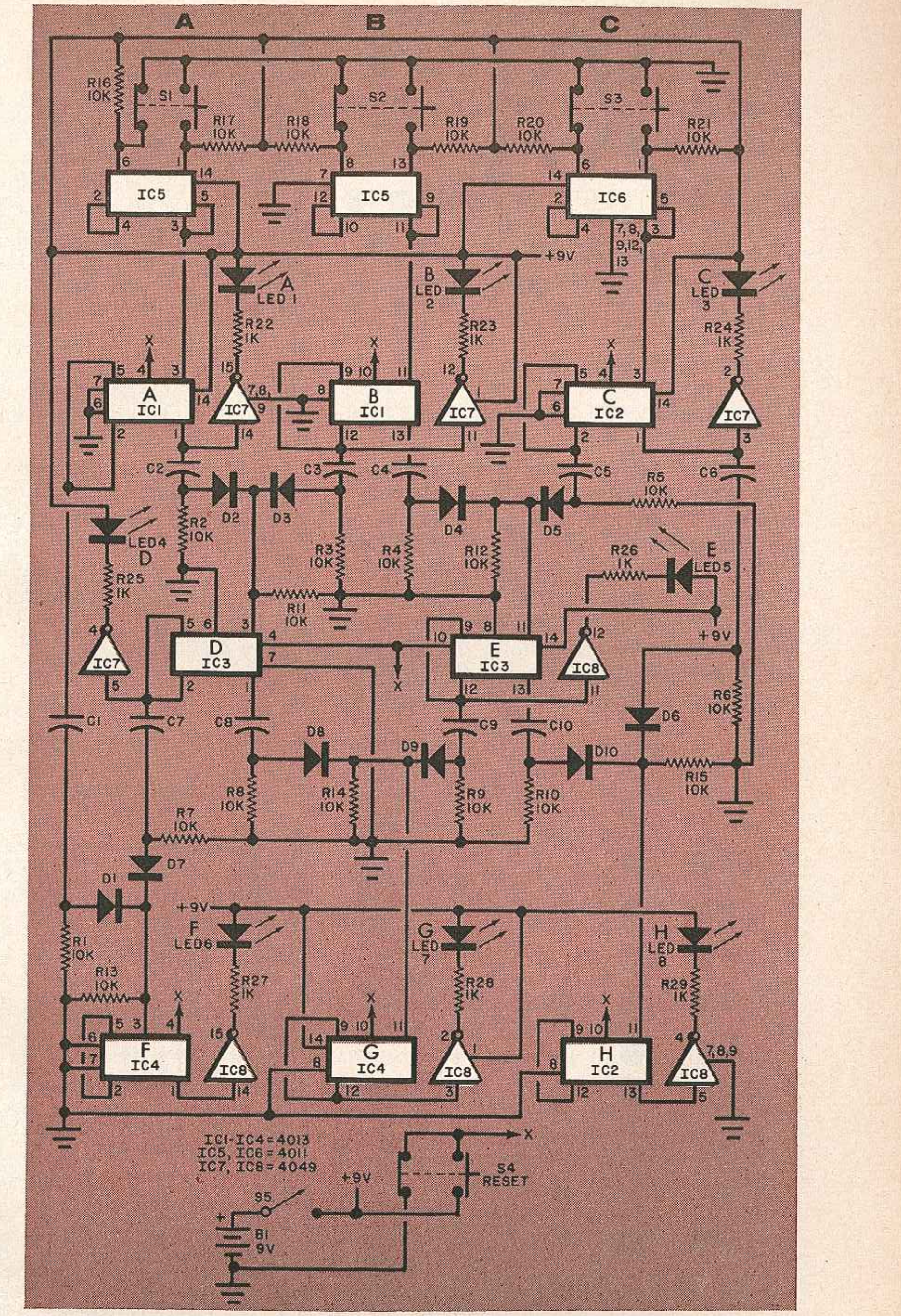


Fig. 1

PARTS LIST

B1—9-volt alkaline/mercury battery
C1-C10—0.033-μF disc capacitor (low voltage)
D1-D10—Silicon diode (1N914 or similar)
IC1-IC4—CD4013 integrated circuit
IC5,IC6—CD4011 integrated circuit
IC7,IC8—CD4049 integrated circuit
LED1-LED8—Any light-emitting diode
R1-R21—10,000-ohm, ¼-watt resistor
R22-R29—1000-ohm, ¼-watt resistor
S1-S4—Spdt switch, momentary closed (Alco MSP-105F or similar)
S5—Spst switch
Misc.—Battery connector, suitable cabinet, "dry-transfer" type, adhesive tape, etc.

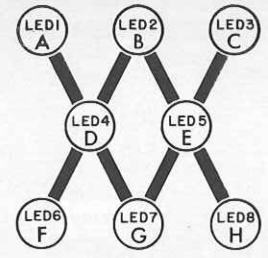


Fig. 2. Arrangement of LED's on the front panel of Flip.

IC7 and *IC8* above rated values and will also decrease battery life.

Construction. The Flip circuit uses CMOS logic circuits since they require low power, have good noise immunity and can be operated with unregulated voltage between 3 and 15 V. However, in using CMOS, some precautions must be kept in mind. All unused gates must have their inputs tied to the plus or minus supply voltage to prevent potential chip burn-out. Care must also be taken in installing the devices. Avoid any possibility of static charges on the inputs. Keep them in the insulation in which they are shipped until ready to solder and use a grounded soldering iron. Low-temperature solder and a low-power iron should be used.

Diodes *D1* through *D10* are not critical; low-current switching types (silicon) were used in the prototype.

The circuit can be assembled on a perf board or on a pc board as shown in Fig. 4. To avoid complexity on the pc board, some short cuts have been taken. Note that *C1* through *C10*, *D1* through *D10*, and *R1* through *R10* are attached together as shown in the insert in Fig. 4 before inserting the loose ends in the pc board. Note that the capacitor end is called out as A, the diode end as B and the resistor end as C on the overall component layout.

There are 19 jumpers that must be made of thin insulated wire and connected between similarly numbered points in Fig. 4 (point 1 to point 1, etc. up to point 16 to point 16). The last three jumpers are from point 17 on *IC1*, *IC2* and *IC3* to point X, the reset circuit.

The eight LED's and the three switches are mounted on the front panel as shown in Fig. 2 and the photo. Also mount the reset and on/off switches on the front panel. The lines connecting the lights on the front panel can be added in any way desired.

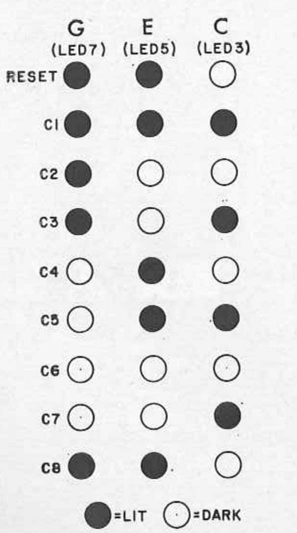
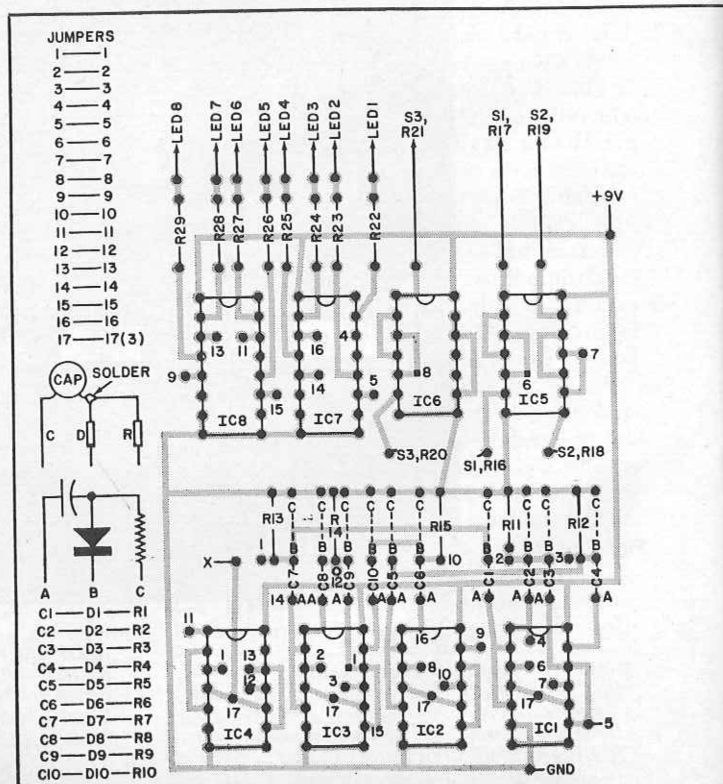


Fig. 3. It takes eight operations of a pushbutton to make the cycle. This shows which LED's come on in sequence.



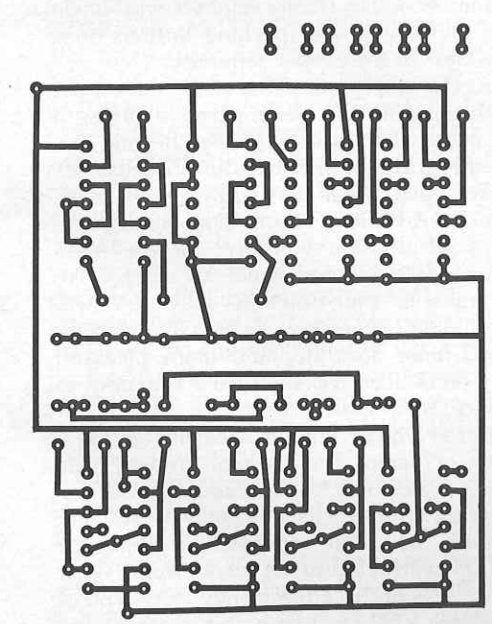


Fig. 4. Either a perf board or printed circuit board can be used for the circuit. Note how the C-D-R assembly is made. Be sure to install the jumpers.

Testing. Turning on the power switch should cause a random pattern to appear on the LED display. Pressing the reset switch should result in the P1 pattern of Fig. 5. If it doesn't, check the reset wiring and voltage connections. After obtaining the P1 pattern, press switches *A*, *B*, and *C* one at a time to verify that all flip-flops are being triggered properly as indicated in Fig. 3. Check signals and wiring for any that fail to operate properly. If the signals to a flip-flop are correct but it still fails to trigger, replace the chip.

Use. Figure 5 shows how Flip is used to solve puzzles. Pressing reset switch *S4* provides the pattern of lights shown at P1. As a sample problem, try to get from pattern P1 to pattern P2 by pressing one or the other of the input switches just 7 times. The other patterns in Fig. 5 can be obtained with the indicated number of switch operations.

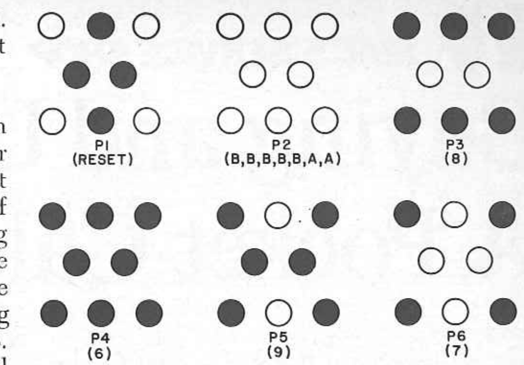


Fig. 5. Pattern after reset is P1. To get P2, press switches as shown. Other patterns take indicated number of switch operations.

An interesting game that can be played is to try to generate specific patterns, with players taking turns pressing just one switch at a time. Starting with the reset switch operated to set the original pattern, the goal is to obtain a pattern consisting of a triangle of lights (either *ACDEG* or *BDEFH*). It doesn't matter if additional lights are on as long as one of the two winning triangles appears. Of course, other patterns, easier or harder, can be chosen as the winning pattern. Since it is possible to predict what pattern is going to appear next, considerable skill can be developed.

Flip provides some insight into why bugs occur in large computers after months or even years of use. These machines have thousands of possible states, many of which remain untested until someone happens to write a program that causes one of these states to occur. Flip, with only 8 flip-flops, has relatively few possible states, but it is still nontrivial in a mathematical sense. For example, how many of the potential 256 states (or patterns) can be obtained starting from the reset state? Can you develop an algorithm (set of rules) for finding the shortest sequence of switch depressions to transform one pattern to another?

Here is another interesting property of Flip. If the sum of the lights that are on in the top and bottom rows is even, then pressing *A*, *B*, and *C* any number of times will leave this sum even. In other words, the parity of these 6 bits (lights) can't be changed by the input switches. This concept of parity is used for error checking in computers. For example, a switch input can only change the parity of the 6 bits of the top and bottom rows if a circuit malfunction occurs. This condition could easily be detected and used to turn on an error light. ◈