

BUILD R-E'S LOGIC DEMONSTRATOR

Simple demonstrator shows how the six most common logic functions operate

by DON LANCASTER

HERE'S A SIMPLE PROJECT THAT SHOWS you the basics of OR, NOR, AND, NAND, EXCLUSIVE OR and EXCLUSIVE NOR logic functions. You can use this as a personal study aide, as a teaching demonstrator, as a science fair project, or as a school lab project. It's also a dandy first TTL (Transistor-Transistor Logic) project, showing you the basics of mounting, supplying, decoupling, and visually indicating states with TTL. Kits for this project are commercially available.

About logic

Logic is the fundamental language of the digital world of computers, calculators, digital instruments, and digital electronics in general. It is one way machines have of talking with each other or with humans, and it is the way they are taught to provide the correct responses for a given set of input situations.

The most common logic in use today has two possible conditions or *states*. These states are ON or OFF, YES or NO, or, in digital language, a 1 or a 0. Logic is simply a set of rules of what a circuit, called a *logic block*, will do. A logic block will provide either a 1 or a 0 in predetermined response to a specified collection of 1's and 0's at its inputs. Enough logic blocks taken together provide the memory for a computer, the answer for a calculator, or the counting for a digital voltmeter.

The *Logic Demonstrator* uses a TTL integrated circuit. When using TTL, it is common to call a 0 a voltage very near ground or 0 volts and a 1 any voltage between +2.4 and +5 volts. This is called a *positive logic convention* and is often, but not always, used with TTL.

A TTL logic block *accepts* 1's and 0's at its inputs and then *provides* new 1's and 0's at its outputs in response to a predetermined set of logic rules it has been trained to respond to.

One-input logic blocks

The simplest useful logic block would have one input and one output. The output could be taught to do any of *four* possible things: It could *ignore* the input and always put a 1 out; it could *ignore* the input and always put a 0 out; it could *follow* the input and put a 1 out



when the input is 1 and a 0 out when the input is 0; or it could *complement* the input, and put a 1 out when the input is 0 and a 0 out when the input is a 1.

The first two possibilities are patently worthless, while the third would be useful only if we were increasing the drive capability or something else rather special. Thus only the fourth possibility is genuinely useful. We call a one input, one output logic block that complements its input an *inverter*.

Logic people have a way of building up a little chart that lists what a logic block will do. This chart is called a *Truth Table*, and it lists the output response you will get for every possible combination of every possible input. With an inverter, there is only one input, and it can only be a 1 or a 0. The truth table looks like this:

INVERTER		
A		X
0		1
1		0



Inverters are used whenever we want to generate the complement of a logic signal. They are sometimes also used to increase the drive capability of a logic system.

Two-input logic blocks

Logic blocks with two inputs and one output are far more versatile than single inverters, for they may be used in

combination to build up *any* logic function, however complex, from the simplest NAND gate to the biggest computer memory. With the two-input logic block, there are four possible input conditions—00, 01, 10, and 11. The output can be anything we teach the block to do, with a 1 or a 0 cropping up in any of the four output slots.

Some thought will tell you there are *sixteen* different ways we can teach or program the logic block. Of these, six ignore at least one input and thus are essentially worthless. Four others are rather specialized and thus see little use. The remaining six logic block programs, called the OR, NOR, AND, NAND, EXCLUSIVE OR, and EXCLUSIVE NOR blocks are the workhorses of digital logic, and, together with the inverter, give you a stock of seven basic logic blocks with which any digital machine can be built up.

OR function

The OR function gives you a 1 out for a 1 in on *either or both* inputs:

OR		
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1



It is used any time we want to verify the presence of a 1 on any input.

NOR function

The NOR gives you a 0 out for a 1 in on either or both inputs. One way to build it is to follow an OR gate with an inverter. Its truth table looks like this:

NOR		
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

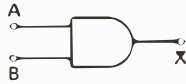


It's usually used anytime we want to stop something from happening if a 1 shows up on any input.

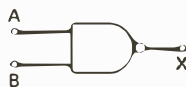
AND and NAND

The AND function needs a 1 on both inputs to get you a 1 out, while a NAND is a AND plus inverter that gives you a zero out for 1's coincident on the input. The truth tables look like this:

AND		
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1



NAND		
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



The AND is used to detect a coincidence of 1's at its inputs; NAND is used to stop something from happening at the coincidence of input ones. Two NAND gates back-to-back form a set-reset flip-flop, or the most basic digital storage element.

EXCLUSIVE OR and NOR

The EXCLUSIVE OR gives you a 1 out if one but not both inputs have a 1 on them, while the EXCLUSIVE NOR gives you a 0 for one but not both inputs being a 1. Looking at it another way, the EXCLUSIVE NOR gives you a 1 out if the inputs are identical and a 0 if they are different:

EXCLUSIVE OR		
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0



EXCLUSIVE NOR		
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1



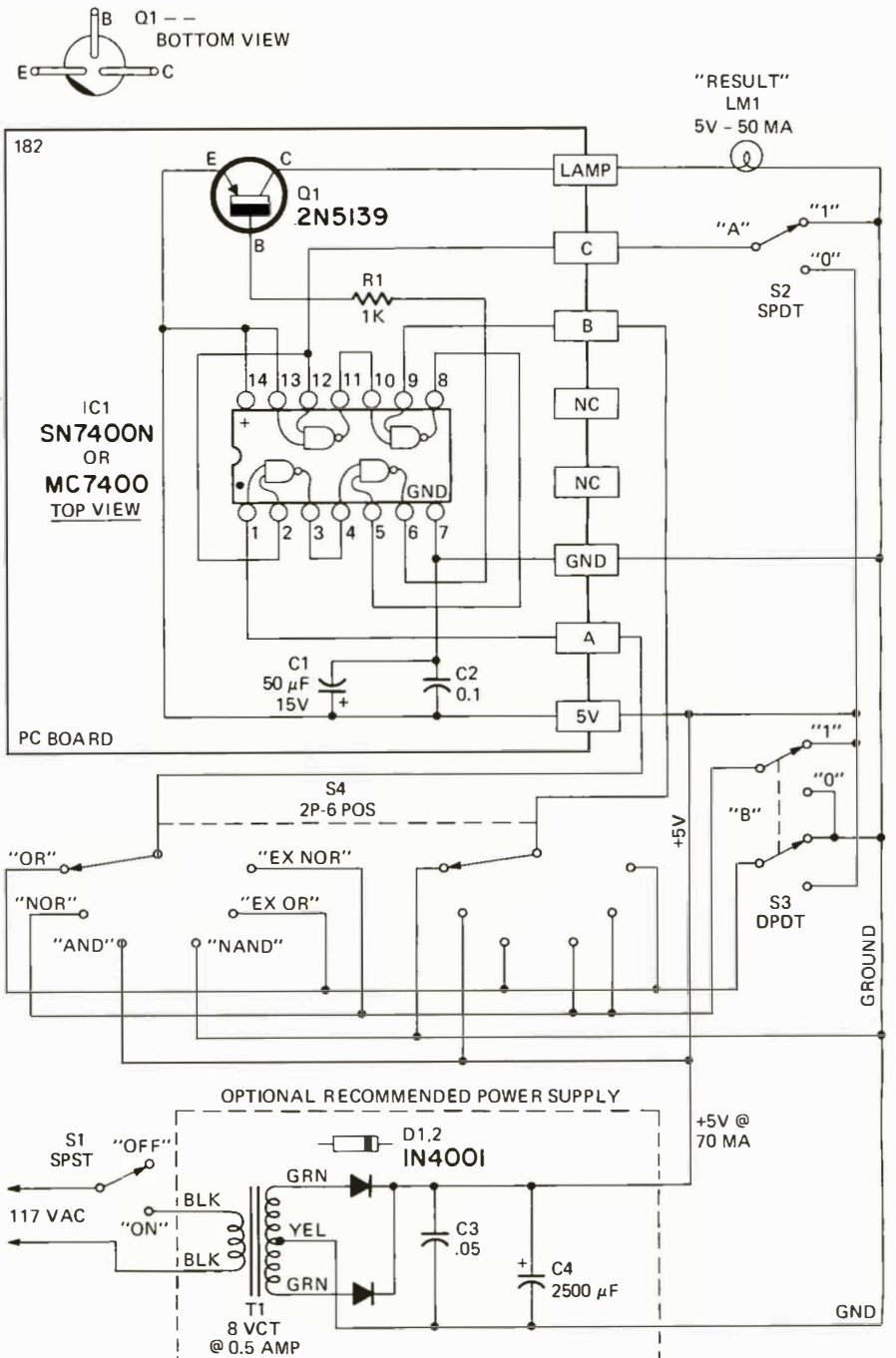
The EXCLUSIVE OR circuit is used to perform binary arithmetic. When doing this, it goes by the name of a HALF ADDER. Two half adders and an inverter can perform binary addition, and thus EXCLUSIVE OR circuits are the cornerstone to digital computation. The EXCLUSIVE NOR is also called a COM-

PARTS LIST

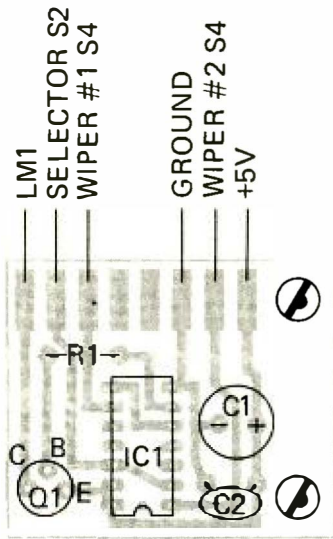
- C1—50 μ F, 15V, electrolytic
- C2—0.1 μ F, 10V, disc ceramic
- *C3—.05 μ F, mylar or disc ceramic
- *C4—2500 μ F, 10V, electrolytic
- *D1, D2—1 amp, 50 PIV, power diode: 1N4001 or equal
- IC1—SN7400 or MC7400 TTL Quad Two-Input Gate
- LM1—5V, 50 mA, pilot lamp assembly
- Q1—2N5139
- R1—1000 ohms, 1/4 watt
- S1—spst slide
- S2—spdt slide
- S3—dpdt slide
- S4—2-pole 6-position non-shorting selector switch (Mallory 3226J or equal)
- *T1—primary, 117V_{ac}; secondary, 8V_{ct}, 0.5A

transformer
MISC—1 1/2" x 1 1/2" PC Board (see text); PC Mounting brackets; switch hardware; 3/4" knob; walnut and gold colored case, bottomplate and endplate assembly, prepunched and prefinished; mounting feet (4); wire; solder; line cord and strain relief*; terminal strip*; power supply hardware*; wire nut*; etc. . . .

NOTE: The following parts are available from Southwest Technical Products, 219 W. Rapsody, San Antonio, Texas, 78216: Etched and drilled PC Board No. 182, \$3.00 Complete kit of all parts less power supply No. 182K, \$8.25 Power supply components (* above) No. 182PS \$4.25 postpaid in US.



COMPLETE SCHEMATIC OF THE LOGIC DEMONSTRATOR. You will note that the unit is built around a single IC and its functions are switch selected.



COMPONENT SIDE

PARATOR, for it lets you *test* to see if inputs are identical or different. It is used in coincidence and counter circuits.

Build a demonstrator

The schematic is shown in Fig. 1. A small printed circuit board is recommended to support the TTL logic gate

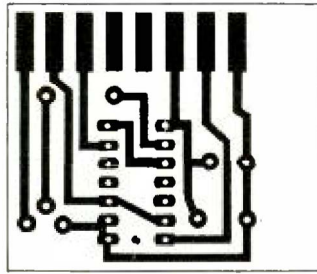


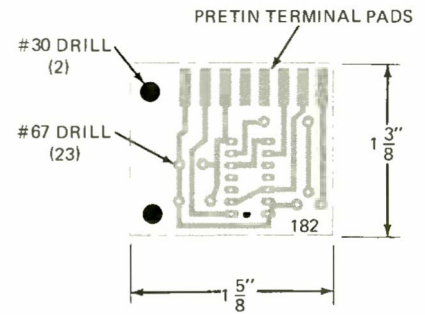
FIG. 2—FULL-SIZE FOIL PATTERN above is the circuit board you need for your own logic demonstrator. Use this pattern as is.

FIG. 3—DRILLING GUIDE (top right) shows drill bit sizes to make proper size holes in the circuit board.

FIG. 4—PARTS LAYOUT (left) on the circuit board. All connections are shown actual size.

and lamp driver transistor. You can buy this item commercially, or you can build one following the layout of Fig. 2 and the drilling guide of Fig. 3. Components are located per Fig. 4. Watch the polarity on C1 and the IC, and be sure to use a small soldering iron and fine solder for assembly.

The photos and Fig. 1 should serve as an assembly guide. The PC board



ONE REQ'D -- MAKE FROM 1 1/16" G-10 PC MAT'L
FOIL SIDE

mounts on two brackets above S2, and interconnections are soldered directly to the terminal pads on the PC board.

It is best to check the circuit out function by function as you progressively wire S4 to prevent any possible switch wiring errors. Use the truth tables to verify operation.

The circuit may be powered by a 4½ volt battery, a 5 volt, 100 mA bench supply, or the recommended power supply shown in Fig. 1. The recommended supply may be assembled inside the bottom of the case for trouble free line operation. **R-E**

makeshift ac wattmeter

Just the thing for ac power measurements when accuracy and convenience do not warrant a lab wattmeter.

by George Lennie

In the March 1971 issue, Jack Darr presented valuable information on using a wattmeter when servicing home appliances. I have a makeshift wattmeter that was originally made up to check the current consumption of an automobile block heater. Its basic operation is illustrated in Fig. 1.

A low-resistance power resistor (R) is inserted in series with the line and the load. By measuring the voltage drop across R, we calculate the load current from $I = V1/R$ and load wattage from $W = I \times V2$. The wattage calculation is approximate—neither impedance nor power factor have been considered—but is sufficient to indicate whether power consumption is much out of line.

A 1-ohm, 25-watt resistor was mounted in a surface-type outlet box along with a parallel-ground duplex receptacle. Leads from the ends of the resistor are brought out to a terminal strip as in Fig. 2. (Both meter terminals are "hot" to ground and present a shock hazard. Insulated binding posts or jacks in the side of the outlet box will be safer.—Editor) A 3-wire cable and plug complete the wattmeter adapter. A Heathkit audio voltmeter is used since it can indicate very low ac voltages. The line voltage drop due to the resistor is small and can be ignored in most instances. **R-E**

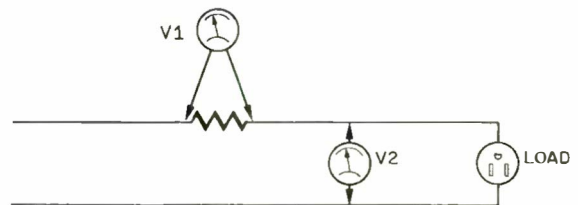


FIG. 1

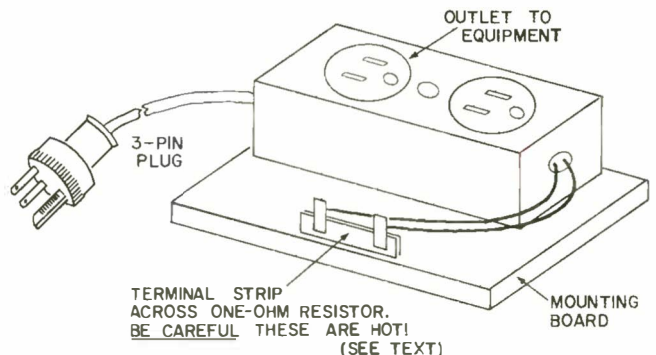


FIG. 2