Experimenter's Corner

By Forrest M. Mims

THREE-STATE LOGIC

IF YOU want to stay abreast of the latest developments in digital logic and microprocessor technology, you need to know something about three-state logic. This month, we're going to experiment with circuits that will teach you the basics of three-state logic in an hour.

Suppose you need to connect the outputs from two or more gates to a common terminal, perhaps the input to another gate. This is OK in the unlikely event *all* the outputs are consistently low or high; but what happens if the outputs are at different logic states? Obviously, it's not possible to place logic 0's and 1's on a common terminal without creating mass confusion—and possibly damaging one or more gates.

Three-state logic provides an efficient solution to this design problem. The output of a conventional logic gate is *always* low or high as long as power is applied. A three-state gate, however, employs a clever circuit that effectively isolates the gate from the output terminal. This requires that a special control terminal called the *enable input* be added to the gate.

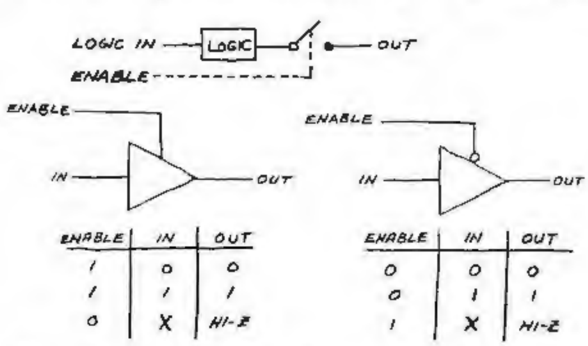
Figure 1 shows two buffers with three-state outputs. When their enable inputs are activated, these buffers pass the logic state of their inputs to their outputs. When the buffers are not enabled, the outputs enter a high-impedance state. This high-impedance output state means the outputs of a dozen or more buffers (or any other three-state logic gate) can be connected to a common terminal if only one is enabled at any one time.

Many digital circuits, particularly microprocessors and memories, use common terminals called *buses* to transmit binary bits or words (a group of bits). Thanks to three-state logic it's possible to connect many different circuits to a common bus so long as one simple rule is followed: The output of only *one* circuit connected to a bus can be enabled at any one time. If more than one output is enabled, logic 0's and 1's will be placed on the bus at the same time, and we're back to the problem that first caused us to employ three-state logic.

We'll look at three-state buses again later. First, let's get some hands-on experience with a three-state buffer.

Three-State Buffer Demonstrator. Figure 2 shows a simple circuit you can quickly build on a solderless breadboard to demonstrate how three-state logic works. It uses one of the gates in a 74125 quad three-state buffer. The two LED's indicate the logic state applied to the input of the buffer when the enable input is at logic 0. When *LED1* is on, the input is low. When *LED2* is on, the input is high.

When the enable input is high, the output of the buffer enters and remains in the high-impedance state irrespective of the logic state at the buffer's input. Both LED's will then glow at about half their normal brightness, conducting a limited amount of current along the path between 5 volts and ground through the series resistors and the LED's.



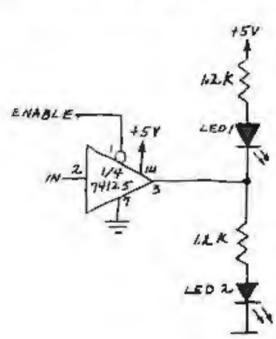


Fig. 1. Two three-state buffer configurations (left).
Fig. 2. Three-state buffer demonstrater (right).

Here's a truth table that sums up the operation of the demonstrator circuit:

Enable	Input	Output LED1	Output LED2
0	0	ON	OFF
0	1	OFF	ON
1	0	*	*
1	1	*	*

*Both LED's at half brightness.

Three-State Multiplexer. A multiplexer is a data selector. Apply an appropriate input select signal and one of several inputs will be applied to a single output. Figure 3 shows how you can make a 4-to-1-line multiplexer from a quad, three-state buffer like the 74125. The enable inputs of the buffers are used as the data select inputs. Remember, only one buffer can be enabled at any one time. With that in mind, here's the truth table for the multiplexer:

Data Inputs A	B	C	D	Data Select A	B	C	D	Output
0	X	X	X	0	1	1	1	0
1	X	X	X	0	1	1	1	1
X	0	X	X	1	0	1	1	0
X	1	X	X	1	0	1	1	1
X	X	0	X	1	1	0	1	0
X	X	1	X	1	1	0	1	1
X	X	X	0	1	1	1	0	0
X	X	X	1	1	1	1	0	1

Note: The X means "don't care"; the input can be either a 0 or 1.

If you build the circuit in Fig. 3, you can apply the data select inputs with a 4-position rotary switch (rotating contact connected to ground) or a 1-of-4 decoder like half of a 74139. The decoder will condense the data select inputs to four 2-bit addresses.

Three-State Bus Demonstrator. Figure 4 shows a simple circuit that will teach you how a three-state bus works. The circuit uses a 74173 4-bit data register with a built-in, three-state output buffer. This means you can connect both the inputs and outputs of the register to the same bus (!) *and* control the transfer of data into and out of the register by applying appropriate signals to the register's read and write inputs.

For best results, build this circuit on a solderless breadboard. Use four rows of adjacent terminal receptacles for the bus and an 8-position DIP switch for the data input and control switch. To write a data word into the register, place the word on the bus by loading it into the first

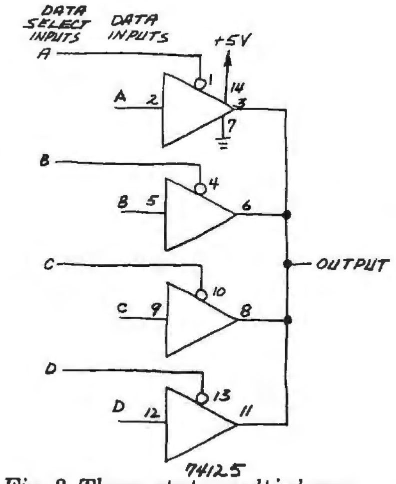


Fig. 3. Three-state multiplexer.

four poles of the DIP switch (let on = 1 and off = 0) and turning switch 8 on. The LED's will display the word you've switched into the input (LED one = logic 1 and LED off = logic 0).

The register will accept a data word from the bus when the WRITE input is low and the positive edge of a clock pulse arrives at pin 7. Prepare to load the data word into the register by turning switches 6 and 7 on. Then apply a clock pulse by turning switch 7 off. This disconnects the CLOCK input of the 74173 from ground, which is the equivalent of applying a positive pulse (unconnected TTL inputs go high). Don't worry about extra clock pulses from the bouncing that occurs when you throw the switch. The data word is copied on the first rising bounce, and any subsequent bounces simply recopy the same word.

After the data word is written into the 74173, turn switch 8 off to remove the input data from the bus. Then turn switch 6 off. To see the word stored in the register, just turn switch 5 on. This will activate the READ input of the 74173 and connect the register's output to the bus. This will display the stored word.

Going Further. You can expand the three-state demonstrator by adding a second 74173 to the data bus. You can connect the CLOCK input of the new register to the CLOCK input of the original 74173, but you'll need a couple of switches on a second DIP switch for the additional READ and WRITE inputs.

Can you think of a practical use for the three-state bus demonstrator? A bus system like this can send data between registers in *either* direction. Therefore, it's often called a *bidirectional data bus.* If that rings a bell, it's because the bidirectional data bus is used in most microprocessors. In fact the simple three-state bus demonstrator we've been experimenting with is functionally equivalent to part of a microprocessor.

In a real microprocessor, of course, the signals that activate the control inputs of the various registers and circuits are automatically supplied by a circuit called a *controller*. The signals from the controller are binary bit patterns called *microinstructions*. ◇

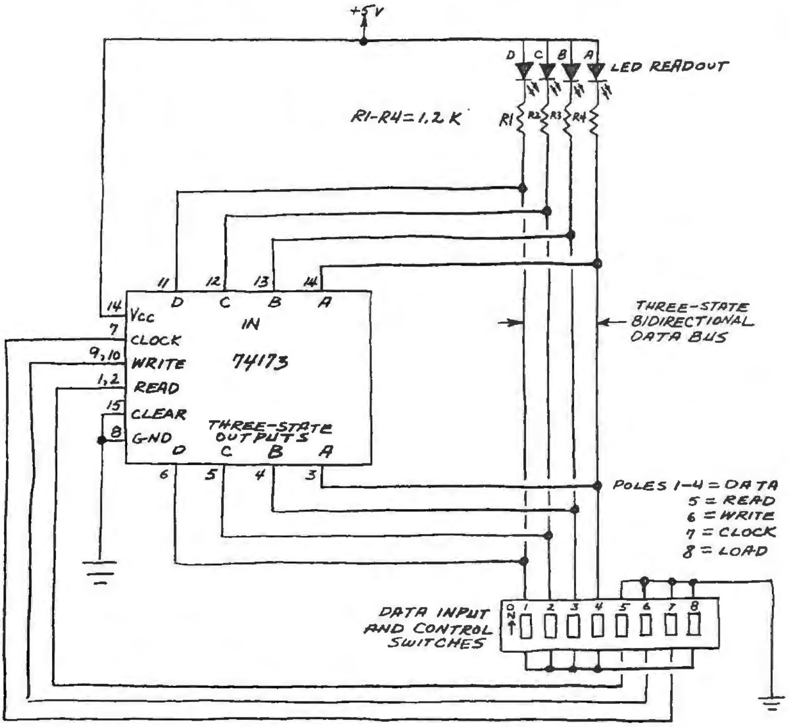


Fig. 4. Three-state bus demonstrater.