

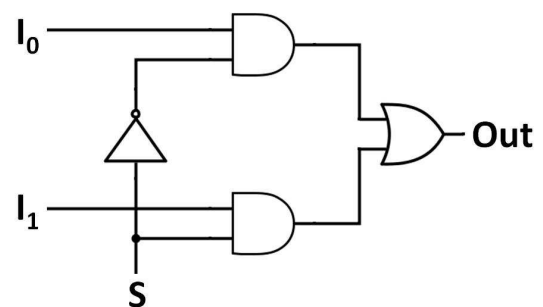
# MULTIPLEXER

The basic idea of multiplexing is to transmit two or more analog messages or digital signals concurrently over a single communication channel, thus sharing what might be an expensive resource. As an example in the telephone industry, a number of phone calls can be carried on a single wire. Another example is a home stereo system remote control that allows one to select between a CD player, a DVD player, or cable TV. Sound systems having digital output can carry several channels over a single fiber optic cable. In electronics, a multiplexer, or MUX for short, is a device that performs this multiplexing function— forwarding the selected input into a single channel. In this lesson we will study digital multiplexing in which the number of inputs is a power of two (2, 4, 8, 16, ...) and there is one output. We will see how to select one of the inputs and send it down the single output channel.

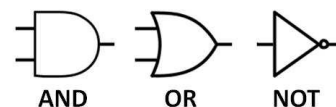
We start with the simplest of digital multiplexers—a 2-to-1 MUX. In this MUX one selection line (S) is used to select one of  $2^1=2$  input lines,  $I_0$  and  $I_1$ , whose data is to be sent to the output (Out). We could set up a truth table to show how this MUX works:

S	$I_0$	$I_1$	Out
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

We notice that when S is 0, then  $I_0$  is the value of the output. On the other hand, when S is 1, then  $I_1$  is the value of the output. In a future lesson, we will learn how to use what are known as Karnaugh maps to determine the best circuit. But this circuit is intuitive enough to present it directly and then discuss how it works:



We see that the circuit consists of two AND gates, one OR gate, and one NOT gate. The gate symbols are as follows:



If S is 0:

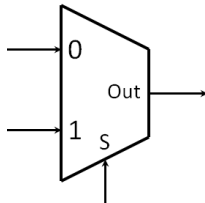
- The output of the lower AND gate is 0, regardless of the value of  $I_1$ .
- The inverted value of S (which is 1) enters the upper AND gate, and the output of the upper AND gate is then whatever  $I_0$  is.

Similar reasoning tells us that if S is 1:

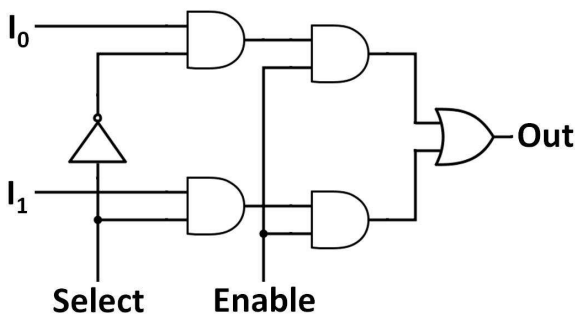
- The output of the lower AND gate is whatever the value of  $I_1$  is.
- The inverted value of  $S$  (which is now 0) enters the upper AND gate, and the output of the upper AND gate is then 0 regardless of the value of  $I_0$ .

The OR gate then allows as output whichever signal was able to get through its AND gate.

A 2-to-1 MUX is commonly diagrammed as follows, in a way that hides its inner-workings, giving us a new building block from which more complex multiplexers can be built. When  $S$  is 0 the top input line is transmitted. When  $S$  is 1, the bottom input line is transmitted.



A somewhat more complex 2-to-1 MUX has one additional input often called *Enable*. When the Enable input is 0, neither  $I_0$  nor  $I_1$  is allowed through, and the output is always 0. When the Enable input is 1, then the selected input (either  $I_0$  or  $I_1$ ) is allowed through. The circuit diagram for the MUX with the Enable is:



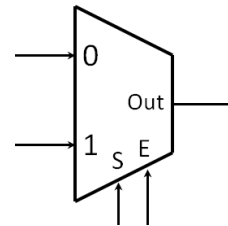
The outputs from the existing AND gates now enter two additional AND gates. If the Enable input is 0, neither  $I_0$  nor  $I_1$  get through. If the Enable input is 1, then the selected input gets through to the OR gate.

The top two input gates, taken together, effectively constitute a 3-input AND gate, where the output of the AND gate on the left is one of

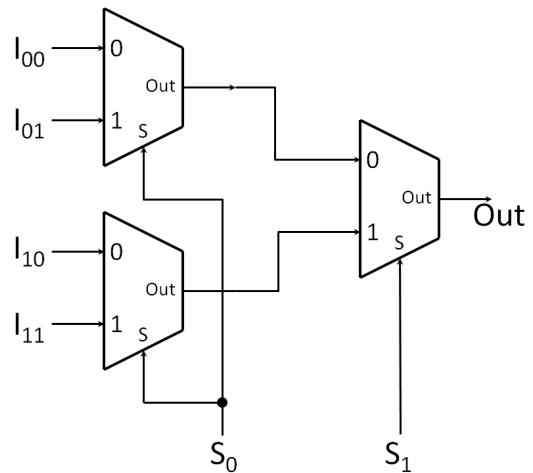
the inputs of the AND gate on the right.

Similarly, the bottom two AND gates, taken together, also constitute a 3-input AND gate. All three inputs of a three-input AND gate must be 1 in order for the output to be 1.

A 2-to-1 MUX with Enable can be diagrammed in a way that hides its inner-workings as shown below. When  $E$  is 0, neither of the input lines is transmitted. When  $E$  is 1 and  $S$  is 0 the top input line is transmitted. When  $E$  is 1 and  $S$  is 1, the bottom input line is transmitted.

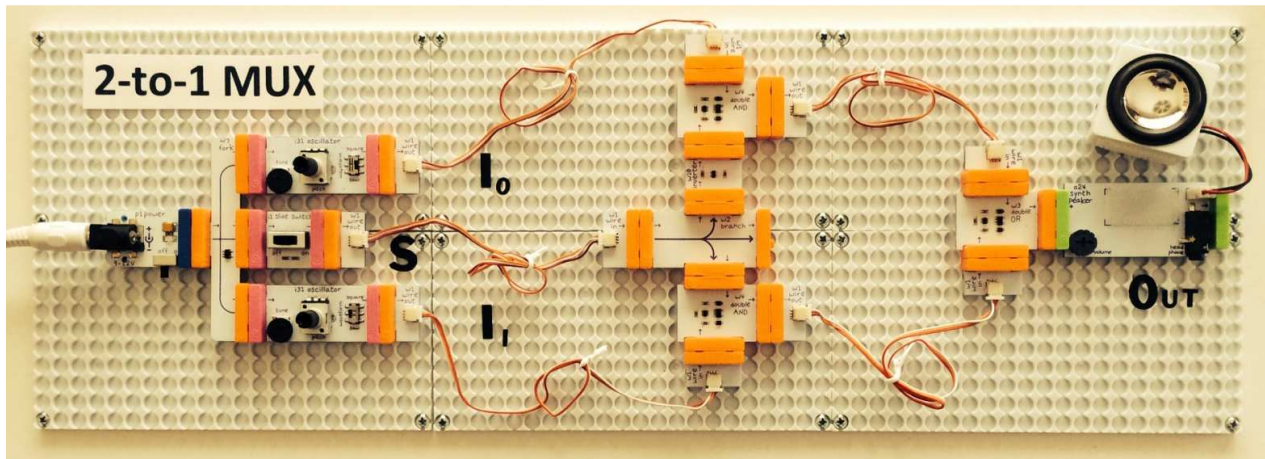


Once we have a 2-to-1 MUX, we can construct a 4-to-1 MUX by using three 2-to-1 MUXs as shown below. For simplicity, we use MUXs without enable inputs.



Some thought about how MUXs work, reveals the following truth table. This table shows which line is output for a given combination of enable inputs.

$S_1$	$S_0$	Out
0	0	$I_{00}$
0	1	$I_{01}$
1	0	$I_{10}$
1	1	$I_{11}$



### ACTIVITY 1: Construct a 2-to-1 MUX without Enable

Let's begin by constructing a 2-to-1 MUX without an enable input, as shown in the above photo. The inputs  $I_0$  and  $I_1$  are oscillator modules, each set to a different frequency, so they can be easily distinguished by their sound. The output *Out* is a synth speaker module. The select input *S* is a slide switch module. This MUX is similar to a MUX on a remote control that allows you to switch between, say, a DVD player and a CD player. When the slide switch is in the OFF (0) position, then oscillator  $I_0$  is the output. When the slide switch is in the ON (1) position, then oscillator  $I_1$  is the output.

#### YOU NEED:

- 1 power
- 1 fork
- 1 branch
- 5 wires
- 2 ANDs
- 1 OR
- 1 inverter (NOT)
- 2 oscillators
- 1 synth speaker
- 1 slide switch

### ACTIVITY 2: Convert the 2-to-1 MUX without Enable to a Time Division Multiplexer (TDM)

Replace the slide switch select input *S* with a pulse module. You will then have what is commonly known as Time Division Multiplexing or TDM for short. In TDM the input lines are alternately allowed to transmit for a period of time in a cyclic fashion. By adjusting the speed of the pulse module, you can change the rate at which the two lines alternate transmission. (You can also produce some very interesting sound effects by adjusting the two oscillator frequencies and even more interesting sounds by adding a delay module just before the synth speaker.)

#### ADDITIONALLY YOU NEED:

- 1 pulse
- 1 synth speaker

### ACTIVITY 3: Add an Enable Input to the MUX

Beginning with your multiplexer from activity 2, add the logic and other components required for an Enable input. When you have completed this, you should notice that setting the enable switch to OFF (0) will stop all transmission (except some echoing from a delay, if present).

#### ADDITIONALLY YOU NEED:

- 1 fork
- 1 toggle switch (or a slide switch or button)
- 2 ANDs
- 1 split
- 3 wires

#### ACTIVITY 4: Construct a 4-to-1 MUX

If you are in a classroom setting, and each lab group of students has constructed a 2-to-1 MUX, you might find it interesting, challenging, and fun to connect three lab group 2-to-1 MUXs together into a 4-to-1 MUX. To keep the number of components required to a minimum, it is suggested that you use 2-to-1 MUXs **without** enable lines. Also, use slide or toggle switches for the two enable inputs on your 4-to-1 MUX, and use oscillators for the four inputs  $I_{00}$ ,  $I_{01}$ ,  $I_{10}$ , and  $I_{11}$ .

Set each oscillator at a different frequency, so you can more easily make sure that the circuit is working correctly when you test it with different combinations of enable inputs  $S_0$  and  $S_1$ . You will probably need a few more wires and a split to complete your class's 4-to-1 MUX. Also, note that there is nothing wrong with having more than one power source on a given project—so you can leave the power modules as they are on your two 2-to-1 MUXs with the four input lines if you wish.