

The truth table we've been using as an example describes a very useful combinational device called a 2-to-1 multiplexer.

A multiplexer, or MUX for short, selects one of its two input values as the output value.

When the select input, marked with an S in the diagram, is 0, the value on data input D0 becomes the value of the Y output.

When S is 1, the value of data input D1 is selected as the Y output value.

MUXes come in many sizes, depending on the number of select inputs.

A MUX with K select inputs will choose between the values of 2^K data inputs.

For example, here's a 4-to-1 multiplexer with 4 data inputs and 2 select inputs.

Larger MUXes can be built from a tree of 2-to-1 MUXes, as shown here.

Why are MUXes interesting?

One answer is that they provide a very elegant and general way of implementing a logic function.

Consider the 8-to-1 MUX shown on the right.

The 3 inputs — A, B, and CIN — are used as the three select signals for the MUX.

Think of the three inputs as forming a 3-bit binary number.

For example, when they're all 0, the MUX will select data input 0, and when they're all 1, the MUX will select data input 7, and so on.

How does make it easy to implement the logic function shown in the truth table?

Well, we'll wire up the data inputs of the MUX to the constant values shown in the output column in the truth table.

The values on the A, B and CIN inputs will cause the MUX to select the appropriate constant on the data inputs as the value for the COUT output.

If later on we change the truth table, we don't have to redesign some complicated sum-of-products circuit, we simply have to change the constants on the data inputs.

Think of the MUX as a table-lookup device that can be reprogrammed to implement, in this case, any three-input

equation.

This sort of circuit can be used to create various forms of programmable logic, where the functionality of the integrated circuit isn't determined at the time of manufacture, but is set during a programming step performed by the user at some later time.

Modern programmable logic circuits can be programmed to replace millions of logic gates.

Very handy for prototyping digital systems before committing to the expense of a custom integrated circuit implementation.

So MUXes with N select lines are effectively stand-ins for N -input logic circuits.

Such a MUX would have 2^N data inputs.

They're useful for N up to 5 or 6, but for functions with more inputs, the exponential growth in circuit size makes them impractical.

Not surprisingly, MUXes are universal as shown by these MUX-based implementations for the sum-of-products building blocks.

There is some speculation that in molecular-scale logic technologies, MUXes may be the natural gate, so it's good to know they can be used to implement any logic function.

Even XOR is simple to implement using only 2-to-1 MUXes!