1.13 DERIVED LOGICAL BUILDING BLOCKS

Basic logic gates and flops can be combined to form more complex structures that are treated as building blocks when designing larger digital systems. There are various common functions that an engineer does not want to redesign from scratch each time. Some of the common building blocks are multiplexers, demultiplexers, tri-state buffers, registers, and shift registers. Counters represent another building block alluded to in the previous discussion of synchronous logic. A counter is a combination of flops and gates that can count either up or down, depending on the implementation.

Multiplexers, sometimes called *selectors*, are combinatorial elements that function as a multiposition logical switches to select one of many inputs. Figure 1.20 shows a common schematic representation of a multiplexer, often shortened to *mux*. A mux has an arbitrary number of data inputs, often an even power of two, and a smaller number of selector inputs. According to the binary state of the selector inputs, a specific data input is transferred to the output.

Muxes are useful, because logic circuits often need to choose between multiple data values. A counter, for example, may choose between loading a next count value or loading an arbitrary value from external logic. A possible truth table for a 4-to-1 mux is shown in Table 1.15. Each selector input value maps to one, and only one, data input.

TABLE 1.15 Four-to-One Multiplexer Truth Table

S1	S0	Y
0	0	A
0	1	B
1	0	C
1	1	D

A demultiplexer, also called a *demux*, performs the inverse operation of a mux by transferring a single input to the output that is selected by select inputs. A demux is drawn similarly to a mux, as shown in Fig. 1.21.

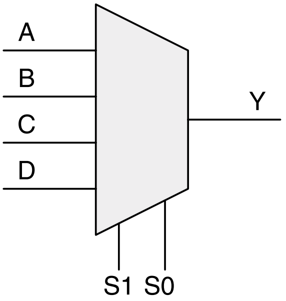

FIGURE 1.20 Four-to-one multiplexer.

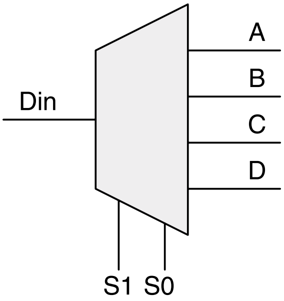

FIGURE 1.21 One-to-four demultiplexer.

A possible truth table for a 1-to-4 demux is shown in Table 1.16. Those outputs that are not selected are held low. The output that is selected assumes the state of the data input.

TABLE 1.16 One-to-Four Demultiplexer Truth Table

S1	S0	A	B	C	D
0	0	Din	0	0	0
0	1	0	Din	0	0
1	0	0	0	Din	0
1	1	0	0	0	Din

A popular use for a demux is as a decoder. The main purpose of a decoder is not so much to transfer an input to one of several outputs but simply to assert one output while not asserting those that are not selected. This function has great utility in microprocessor address decoding, which involves selecting one of multiple devices (e.g., a memory chip) at a time for access. The truth table for a 2-to-4 decoder is shown in Table 1.17. The decoder's outputs are active-low, because most memory and microprocessor peripheral chips use active-low enable signals.

TABLE 1.17 Two-to-Four Decoder Truth Table

S1	S0	A	B	C	D
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

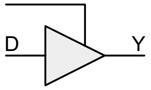

FIGURE 1.22 Tri-state buffer.

Tri-state buffers are combinatorial elements that can drive three output states rather than the standard 0 and 1 states. The third state is off, often referred to as high-impedance, *hi-Z*, or just *Z*. Tri-state buffers enable multiple devices to share a common output wire by cooperatively agreeing to have only one device drive the wire at any one time, during which all other devices remain in hi-Z. A tri-state buffer is drawn as shown in Fig. 1.22.

A tri-state buffer passes its D-input to Y-output when enabled. Otherwise, the output will be turned off as shown in Table 1.18.

Electrically, tri-state behavior allows multiple tri-state buffers to be connected to the same wire without *contention*. Contention normally results when multiple outputs are connected together because some want to drive high and some low. This creates potentially damaging electrical contention (a short circuit). However, if multiple tri-state buffers are connected, and only one at a time is enabled, there is no possibility of contention. The main advantage here is that digital *buses* in comput-