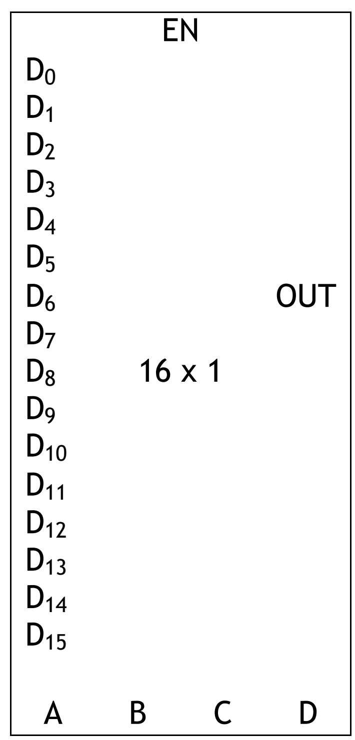
MUXes

- A multiplexer is like a switchboard. You specify which input to relay to the output.
- The *n* selection lines determine which of 2^n inputs to relay to the output.
- An enable input allows the entire MUX to be turned on or off.
- A MUX can be used to implement any Boolean function.
- In a type 0 implementation, for n variables we need n selection lines and 2^n inputs. We simply apply all the variables to the selection lines and tie each input to 0 or 1 according to the desired output value. Type 0 implementations are inefficient and are rarely used.
- In a type 1 implementation, for n variables we need n-1 selection lines and 2^{n-1} inputs. For example, to implement a function of 3 variables, we need a 4 x 1 MUX (it has 4 inputs, 2 selection lines, and of course one output). We apply n-1 variables to the selection lines. Then, each input is 0, 1, x, or $\bar{x}$ where x is the last variable.
- In a type 2 implementation, for n variables we need n-2 selection lines and 2^{n-1} inputs. For example, to implement a function of 4 variables, we need a 4 x 1 MUX. We apply n-2 variables to the selection lines. Then, each input is some simple function of the two remaining variables. External gates other than an inverter may be required for a type 2 implementation.

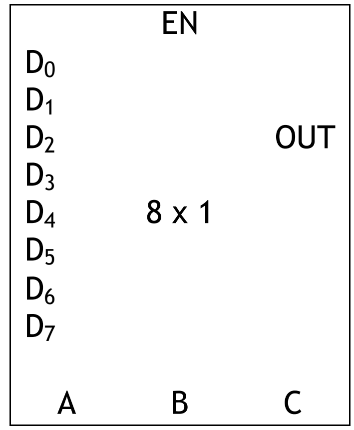
MUX Example: Worksheet

$F = \bar{B}\bar{D} + BC + ABD$

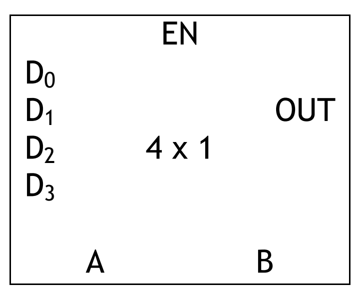
A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



Sequential Logic

General

- Sequential circuits have memory. The output state depends not only on the current input states, but also on the state of the circuit itself (the current output state).
- The truth table thus contains the current output state Q as in input. The output is Q_{n+1}, representing the value of the next state of Q.
- Synchronous sequential circuits are controlled by a clock, whose pulses advance the states of each element in synchrony.

Flip-flops

- The types of flip-flops that we will consider are: RS, D, JK, and T.
- The RS flip-flop is the basic type of flip-flop. When both inputs R and S are 0, the flip-flop retains its current state. When S is 1, it sets the flip-flop to 1. When R is 1, it resets the flip-flop to 0. When both are 1, the output is indeterminate.
- In a D flip-flop, the output is simply whatever the input is (at the time of the last clock pulse).
- A JK flip-flop is just like an RS flip-flop, except that when both J and K are 1, the flip-flop changes (toggles) between one state and the other.
- A T flip-flop toggles the output if the input is 1.
- Know the characteristic table and excitation table for each flip-flop.

Characteristic Tables

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	ind.

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Q_n'

D	Q_{n+1}
0	0
1	1

T	Q_{n+1}
0	Q_n
1	Q_n'

Excitation Tables

Q_n	Q_{n+1}	S	R
0	0	0	d
0	1	1	0
1	0	0	1
1	1	d	0

Q_n	Q_{n+1}	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0