Multiplexor

A multiplexor is a device that takes a number of data inputs and selects one of them to pass through as its output.

The interface of a multiplexor provides means to control which data input value is selected.

If there are K data input signals, then at least log K bits are needed to specify which input signal is to be passed through.

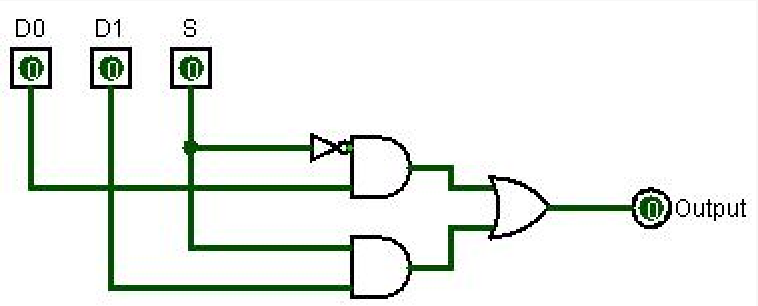
So, in most cases, multiplexors take 2^n data input signals and n control signals.

Designing a Multiplexor

Consider a 2^1 x 1 multiplexor; it takes two data inputs D0 and D1 and a single select bit S:

D0	D1	S	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

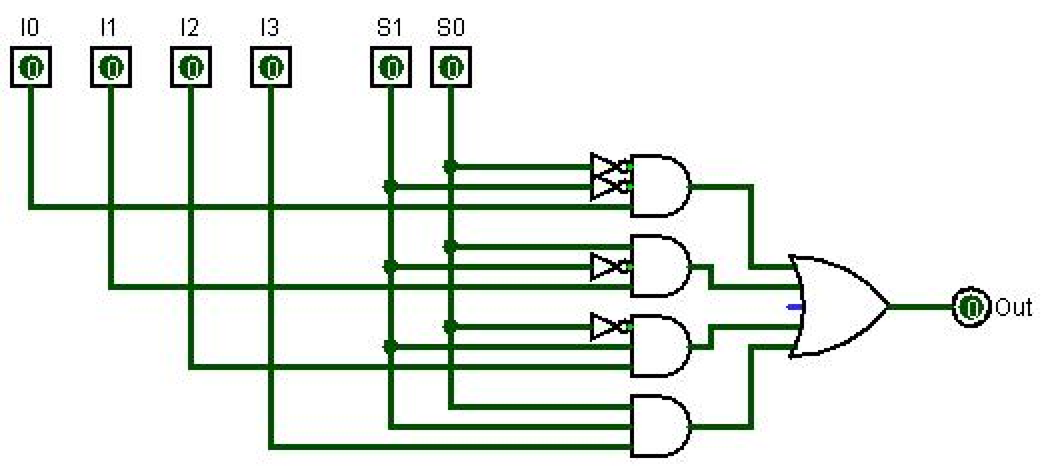
$$\begin{aligned} Output &= \overline{D_0} \cdot D_1 \cdot S + D_0 \cdot \overline{D_1} \cdot \overline{S} + D_0 \cdot D_1 \cdot \overline{S} + D_0 \cdot D_1 \cdot S \\ &= (\overline{D_0} + D_0) \cdot D_1 \cdot S + (\overline{D_1} + D_1) \cdot D_0 \cdot \overline{S} \\ &= D_0 \cdot \overline{S} + D_1 \cdot S \end{aligned}$$



2 x 1 multiplexor

A 4x1 Multiplexor

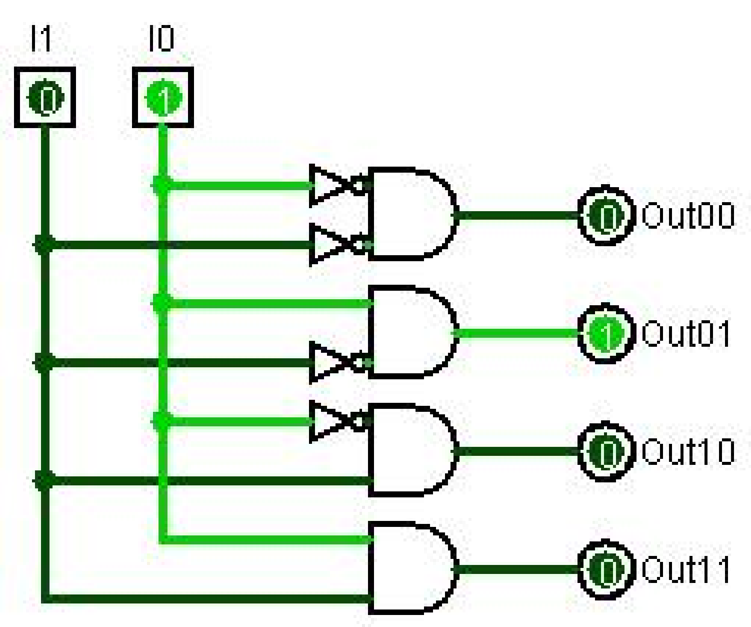
$$Out = I_0 \cdot \overline{S_1} \cdot \overline{S_0} + I_1 \cdot \overline{S_1} \cdot S_0 + I_2 \cdot S_1 \cdot \overline{S_0} + I_3 \cdot S_1 \cdot S_0$$



Decoder

An n x 2^n *decoder* takes n inputs and sets exactly one of its 2^n outputs, based upon the pattern of its inputs.

2 x 4 decoder



$$Out00 = \overline{I_0} \cdot \overline{I_1}$$

$$Out01 = \overline{I_1} I_0$$

$$Out10 = I_1 \cdot \overline{I_0}$$

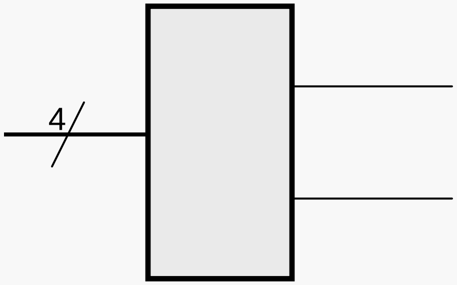
$$Out11 = I_0 \cdot I_1$$

Encoders and Demultiplexors

A 2^n x n *encoder* takes 2^n inputs and sets each of its n outputs, based upon the pattern of its inputs.

Essentially, an encoder is the inverse of a decoder.

4 x 2 encoder



Similarly, a 1 x 2^n demultiplexor:

It takes 1 input and transmits that input on exactly one of its outputs, determined by the pattern of its n selector bits.

1 x 8 demux