Combinational Logic Circuits (II)

- A Universal Logic Solution
- Encoders
- Encoder Applications

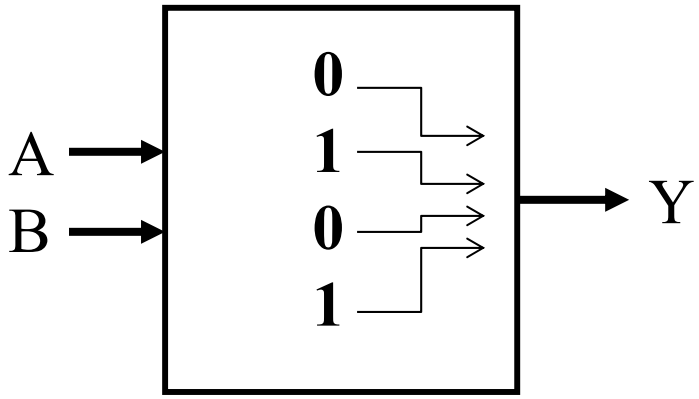
Logic Expressions from a Truth Table

When we obtain a logic expression from a truth table, it is in a standard form, usually a Sum of Products.

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	1

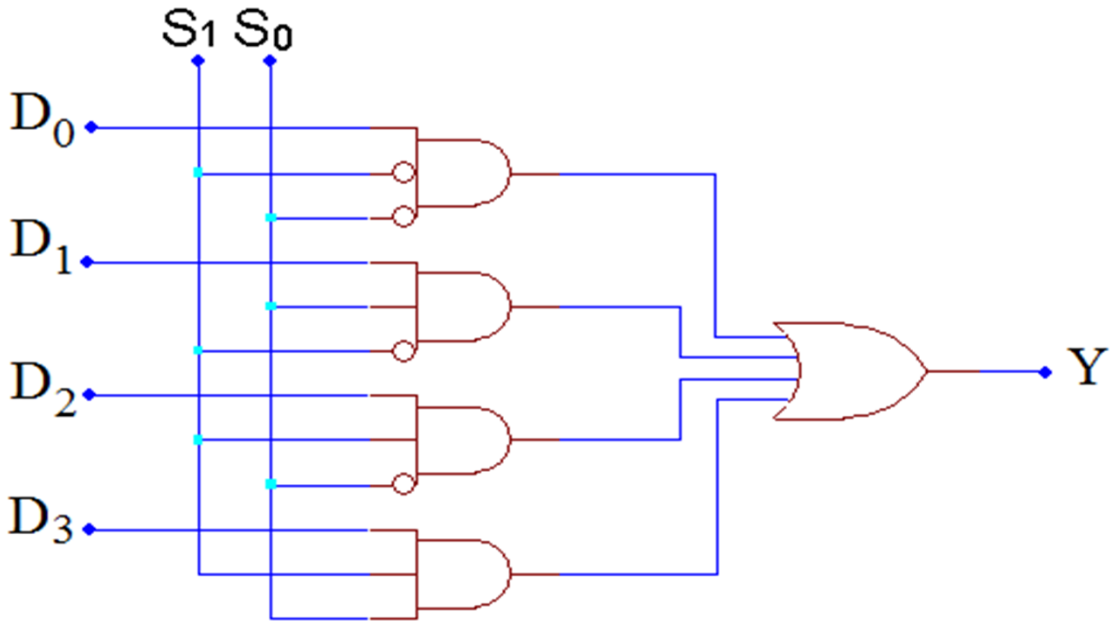
$$Y = \overline{A}.B + A.B$$

It is possible to use the current input combination to **select** the correct output value for F from a list.



A Universal Logic Solution

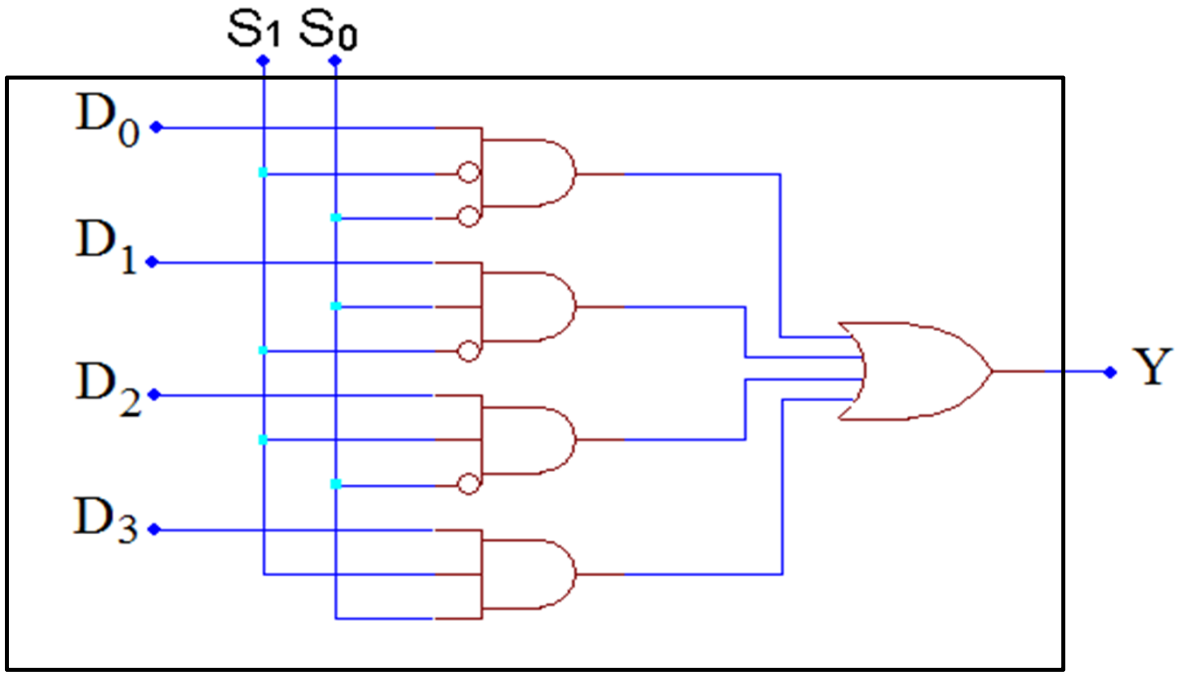
Consider the logic circuit for a 4-to-1 mux:



Every possible combination of the control lines is fed to an AND gate, generating all of the minterms for a two variable function. Individual minterms can be selected by the data inputs.

A Universal Logic Solution

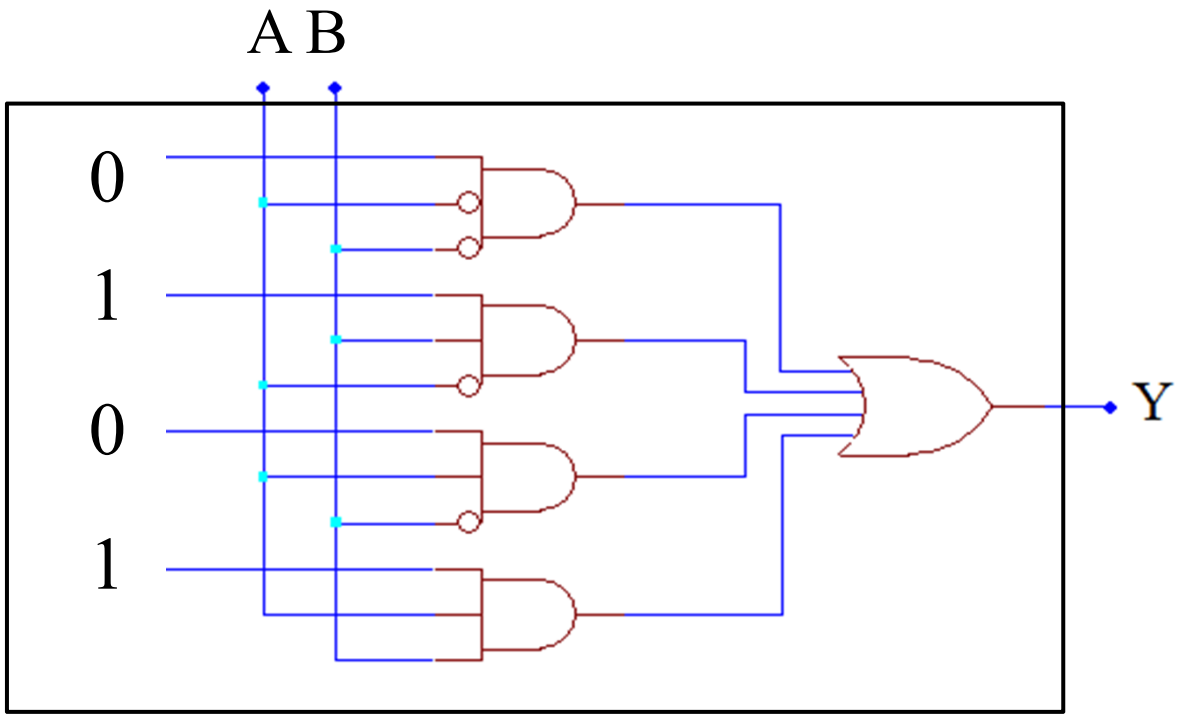
Consider the logic circuit for a 4-to-1 mux:



Every possible combination of the control lines is fed to an AND gate, generating all of the minterms for a two variable function. Individual minterms can be selected by the data inputs.

A Universal Logic Solution

Consider the logic circuit for a 4-to-1 mux:



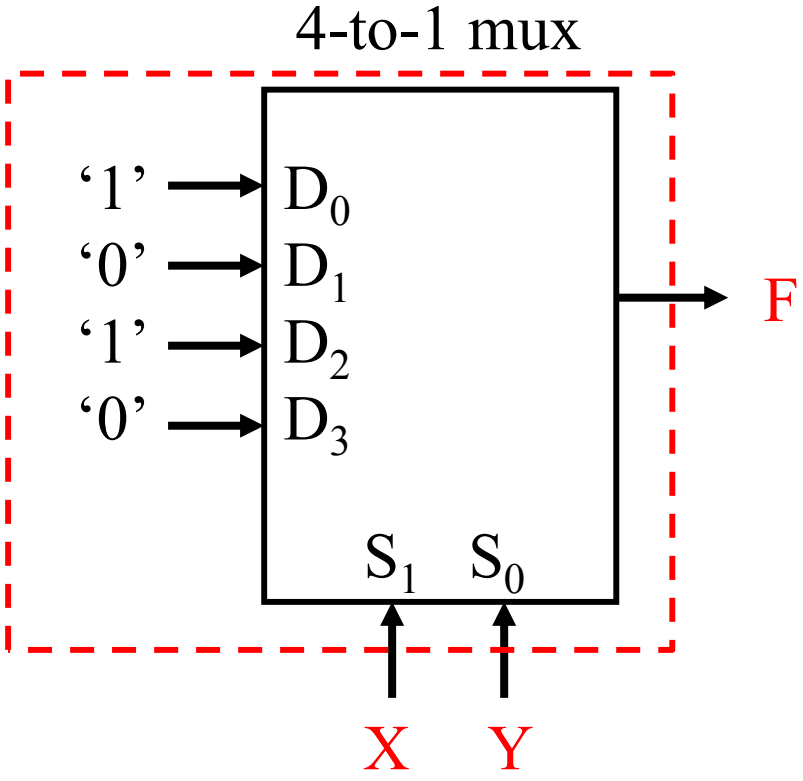
Every possible combination of the control lines is fed to an AND gate, generating all of the minterms for a two variable function. Individual minterms can be selected by the data inputs.

A multiplexer can be used to synthesise the logic function of an *n* input truth table. Consider the following expression.

$F = \bar{X}\bar{Y} + X\bar{Y}$

$\left(F(X,Y) = m_0 + m_2 = \Sigma(0,2) \right)$

X Y	F
0 0	1
0 1	0
1 0	1
1 1	0



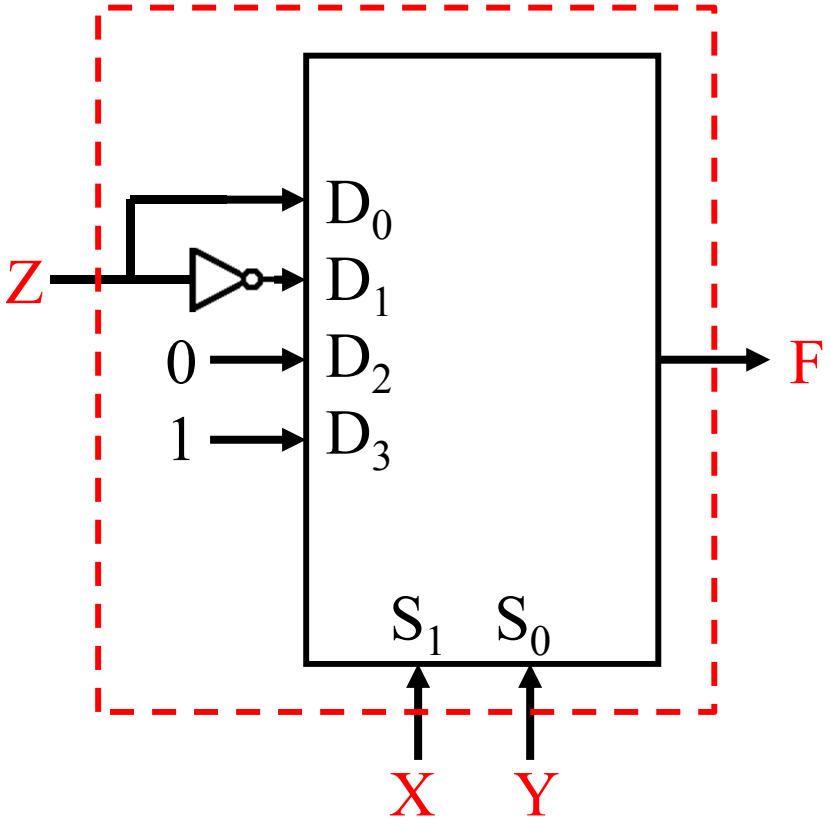
The truth table shown could be implemented by setting D_0 and D_2 to '1' and D_1 and D_3 to '0'. F is selected by X and Y.

A more efficient method exists to implement a Boolean function of n variables with a multiplexer that has (n -1) selection lines.

Consider the function $F(X,Y,Z) = \Sigma(1,2,6,7)$

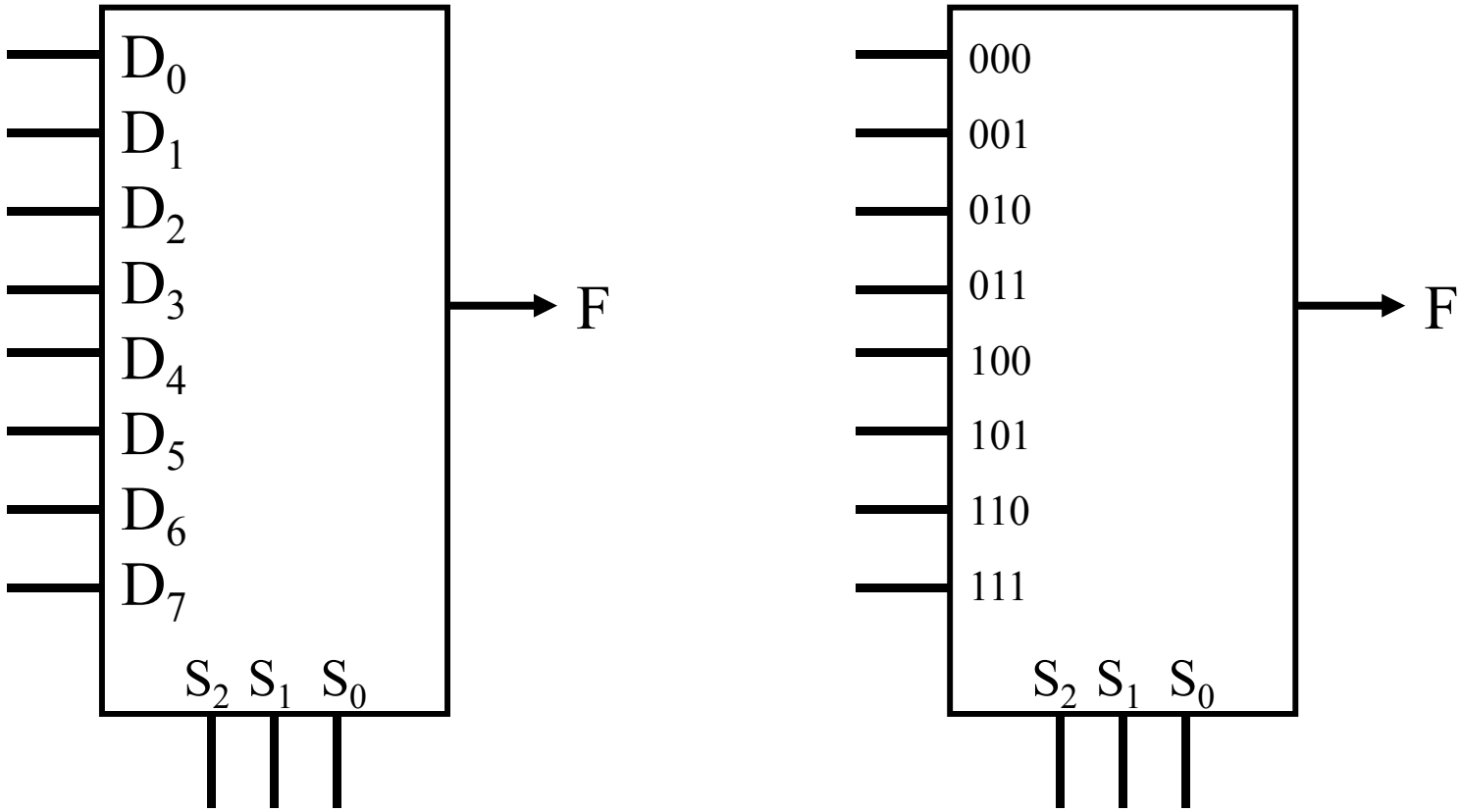
The function can be implemented with a 4-1 multiplexer as shown.

X Y Z	F	
0 0 0	0	F = Z
0 0 1	1	
0 1 0	1	$F = \bar{Z}$
0 1 1	0	
1 0 0	0	F = 0
1 0 1	0	
1 1 0	1	F = 1
1 1 1	1	



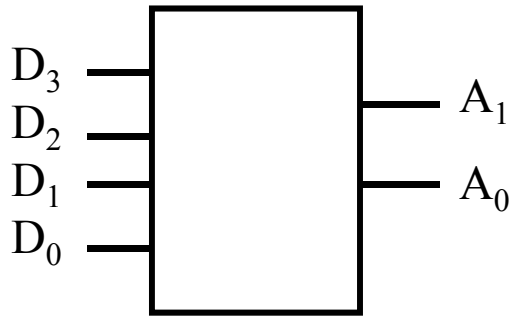
Alternative notation for multiplexer:

Sometimes, instead of labelling the inputs of a mux with a letter and subscript, it is clearer to put the select values corresponding to that data path as shown for an 8-to-1 mux.



Encoders

The output lines of an encoder generate the binary code corresponding to a single active input value. For 2^n inputs there are *n* outputs. Encoders can be used to compress data.



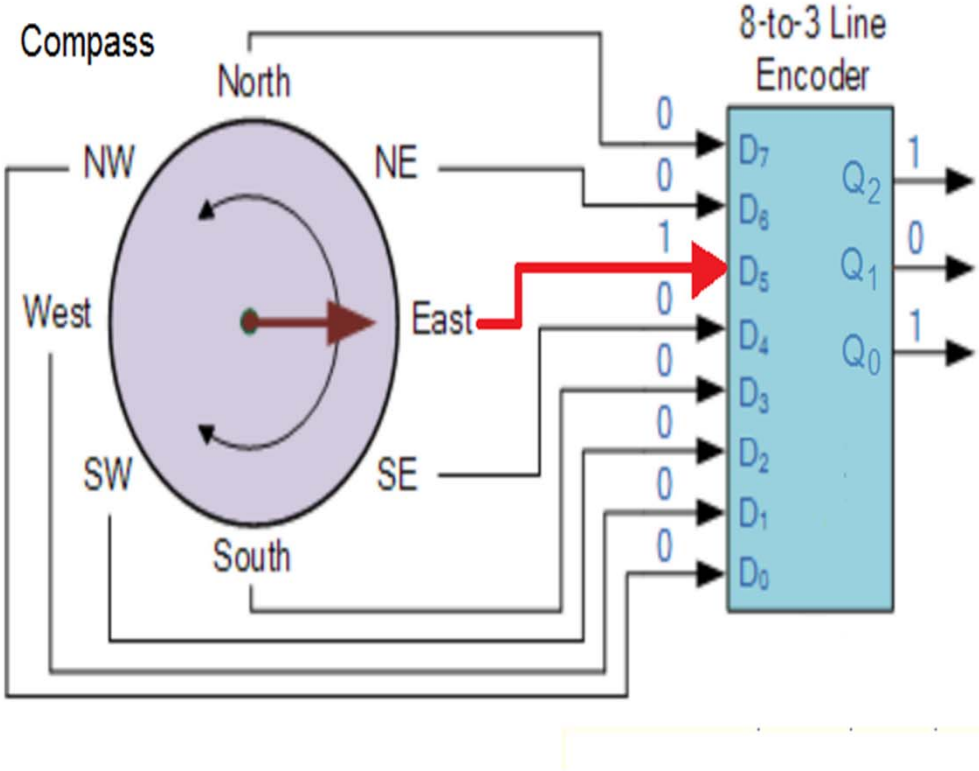
D_3	D_2	D_1	D_0	A_1	A_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

For n = 3, $2^n = 8$

8 inputs can be encoded to 3 output bits etc.

Only one input can be active, in this 1 which is called **active high**.

Encoder Application:



Compass Direction	Binary Output		
	Q_2	Q_1	Q_0
North-West	0	0	0
West	0	0	1
South-West	0	1	0
South	0	1	1
South-East	1	0	0
East	1	0	1
North-East	1	1	0
North	1	1	1

The compass needle will activate only one input line.
The eight compass readings have been encoded into three bits.

Priority Encoder

If more than one input line is active, a **priority encoder** will give precedence to the input with the highest priority. They are often used in computers as interrupt controllers so that the microprocessor can service the most important interrupt request (IRQ).

D_3	D_2	D_1	D_0	A_1	A_0
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1

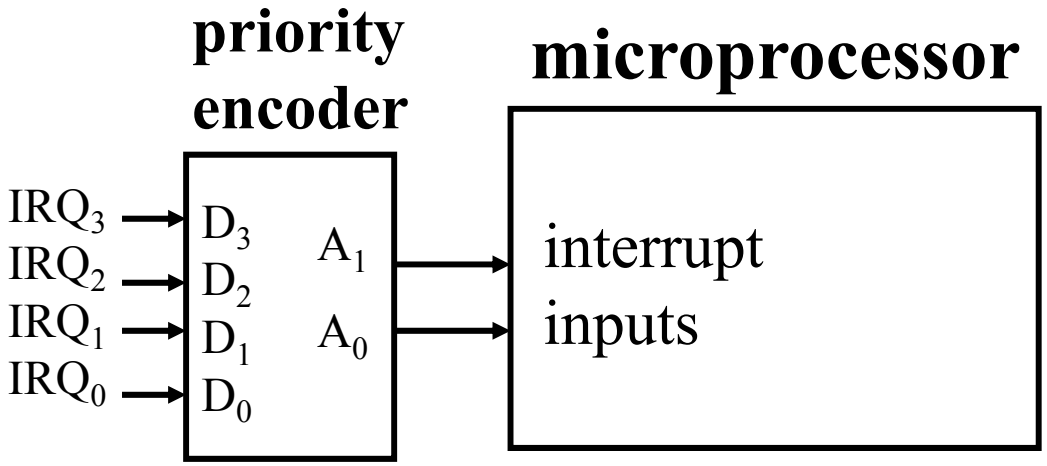
In this example, D_3 has the highest priority, followed by D_2 then D_1 then D_0

IRQ_3 – Aircraft about to stall

IRQ_2 – Change direction

IRQ_1 – Turbulence, seat belts ON

IRQ_0 – Toilet Engaged



Summary

- Multiplexers can be used to implement Boolean logic functions
- Encoders have been examined.