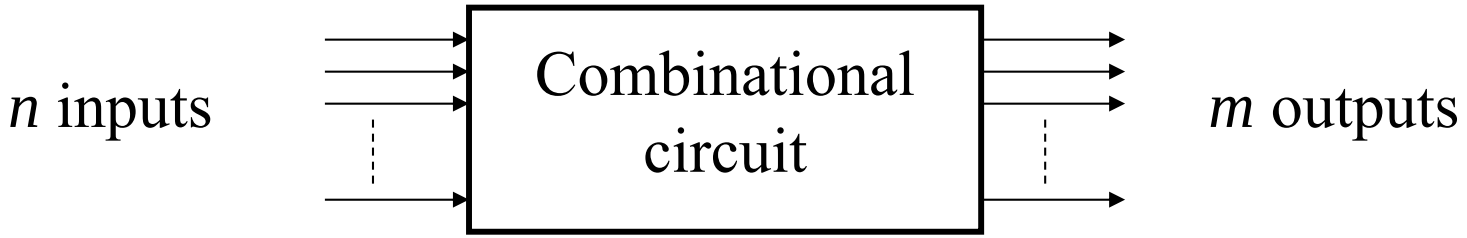
Combinational Logic Circuits (I)

- Timing Diagrams
- Universal Gates
- Simple XOR Circuits
- Multiplexers

Combinational Logic

The outputs of a combinational circuit can be determined at any time from the present combination of inputs.

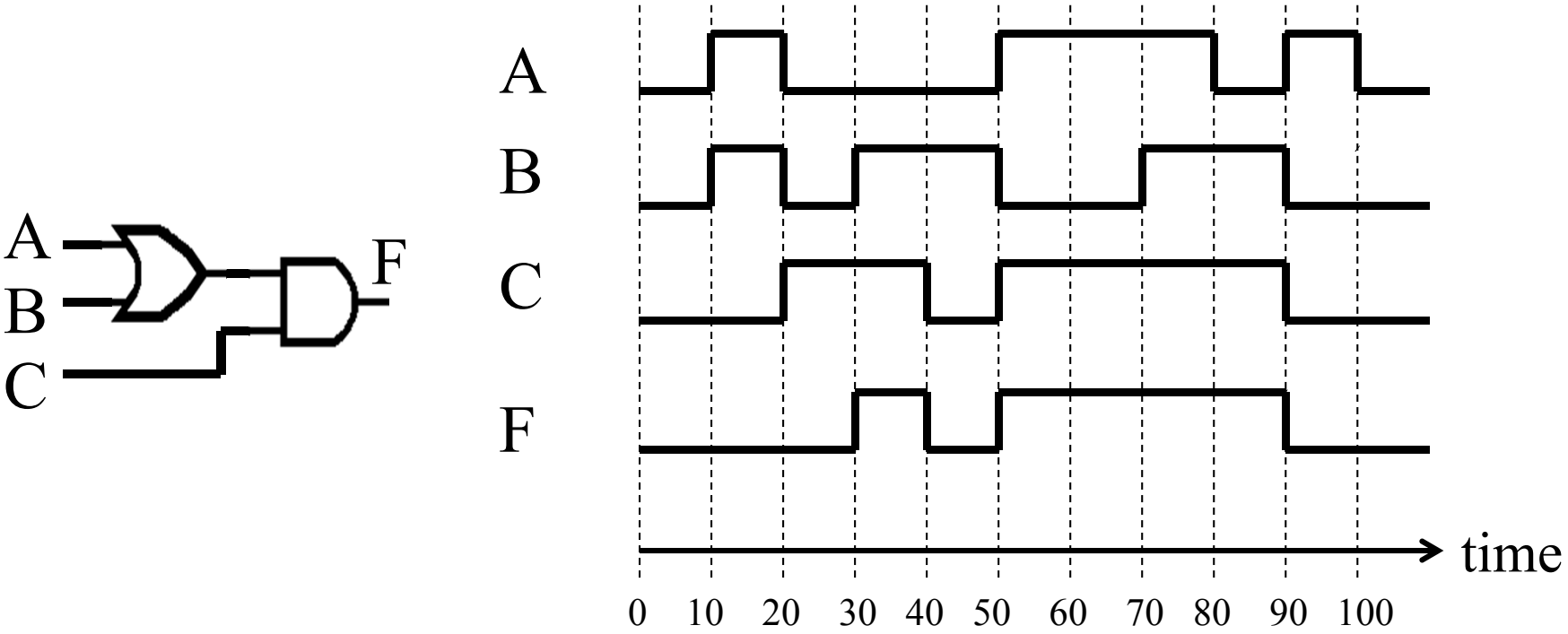


The combinational circuit transforms the binary input data to the required binary output data.

For a circuit with *n* input variables, there are 2^n possible binary input combinations. Each of the *m* output values can be expressed as a function of the n input variables.

Timing Diagrams

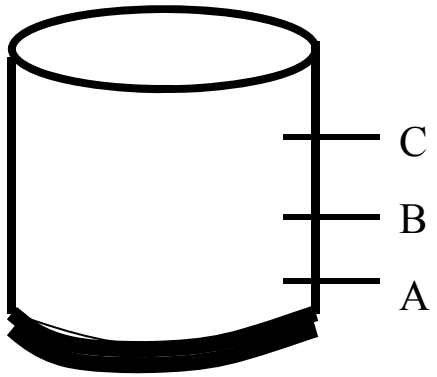
Timing diagrams can be used to show how the outputs of a circuit vary as a function of time.



The diagram shows the output F of the circuit, in response to a time sequence of inputs A,B,C.

Incompletely Specified Functions

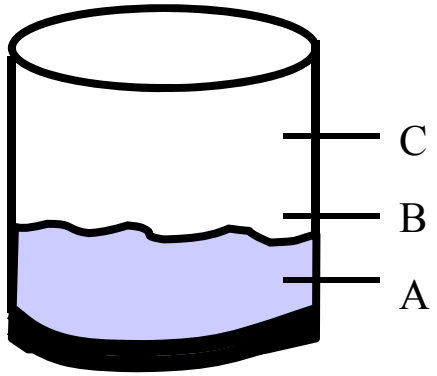
Consider a fluid tank with three level detectors. The detector output is normally ‘0’. It becomes ‘1’ when the fluid reaches it.



C B A
0 0 0

Incompletely Specified Functions

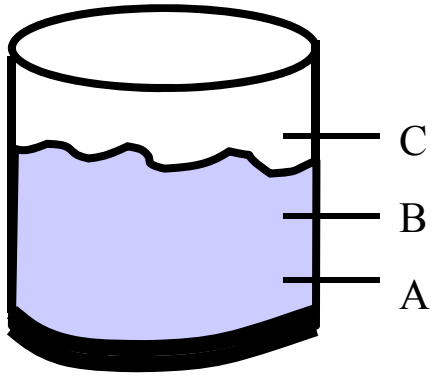
Consider a fluid tank with three level detectors. The detector output is normally ‘0’. It becomes ‘1’ when the fluid reaches it.



C B A
0 0 1

Incompletely Specified Functions

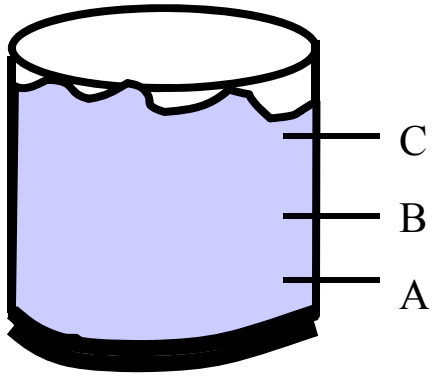
Consider a fluid tank with three level detectors. The detector output is normally ‘0’. It becomes ‘1’ when the fluid reaches it.



C B A
0 1 1

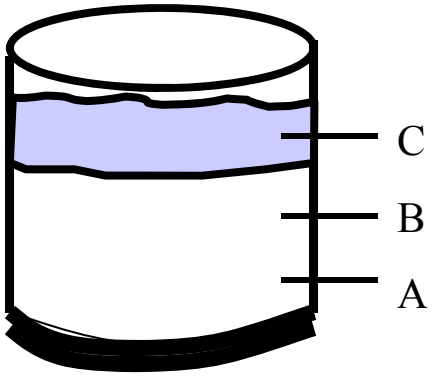
Incompletely Specified Functions

Consider a fluid tank with three level detectors. The detector output is normally ‘0’. It becomes ‘1’ when the fluid reaches it.



C B A
1 1 1

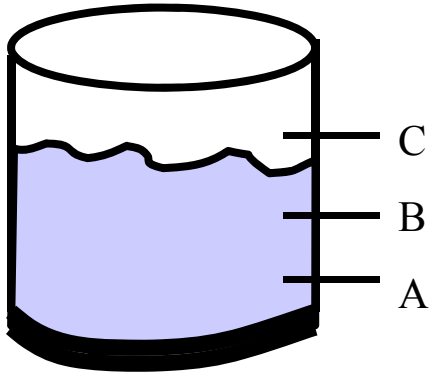
The liquid can't float in mid air, so the combination shown for CBA below will never occur.



C B A
1 0 0

The output for certain inputs does not matter. (these input conditions will never occur in the problem). These 'don't care outputs' are written as X and can take either a 1 or 0. This function is **incompletely specified.**

Design a circuit to give a warning when the liquid level is between B and C.



C	B	A	F
0	0	0	0
0	0	1	0
0	1	0	X
0	1	1	1
1	0	0	X
1	0	1	X
1	1	0	X
1	1	1	0

$$F = \overline{C}.B.A$$

Solution requires a three input AND gate.

Consider making the don't care output at CBA = 010 a '1'.

C B A	F
0 0 0	0
0 0 1	0
0 1 0	X
0 1 1	1
1 0 0	X
1 0 1	X
1 1 0	X
1 1 1	0

C B A	F
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	X
1 0 1	X
1 1 0	X
1 1 1	0

$$F = \overline{C}.B.\overline{A} + \overline{C}.B.A$$

$$F = \overline{C}.B.(\overline{A} + A)$$

$$F = \overline{C}.B$$

Solution now requires a two input AND gate.

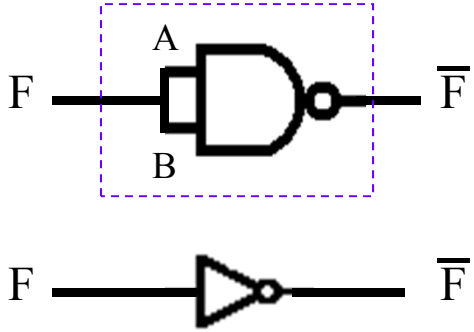
The don't care terms can be used to help with minimisation.

Universal Gates

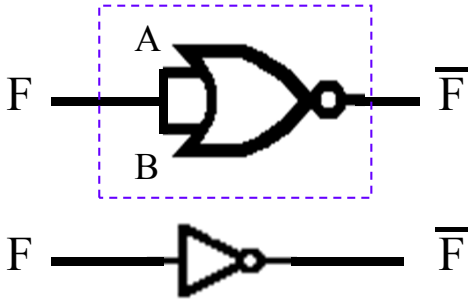
Universal gates are gates from which any digital circuit can be built.

There are two such gates, the NAND gate and the NOR gate. An inverter can be formed by connecting together the inputs of either gate.

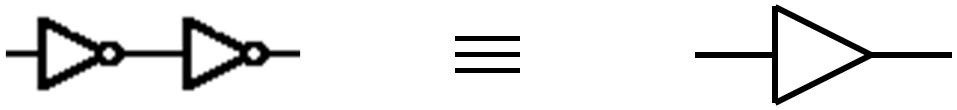
A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0



A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0



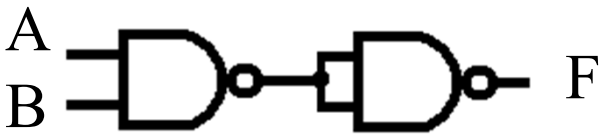
A logic buffer can be formed by cascading two inverters.

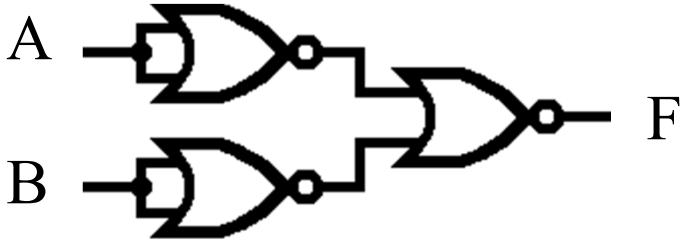


Examples

How can an AND gate be made from(a) NAND gates (b) NOR gates?

$F = A.B = \overline{\overline{A.B}}$ applying De Morgan, $F = \overline{\overline{A} + \overline{B}}$





Examples

How can an OR gate be made from(a) NOR gates (b) NAND gates?

Examples

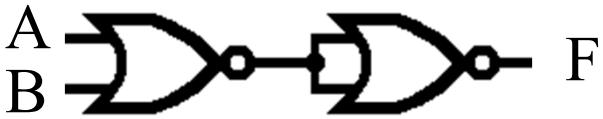
How can an OR gate be made from(a) NOR gates (b) NAND gates?

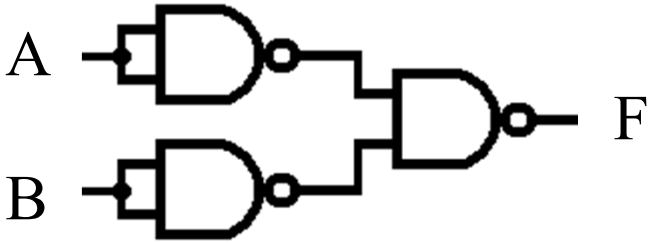
$F = A + B$

Examples

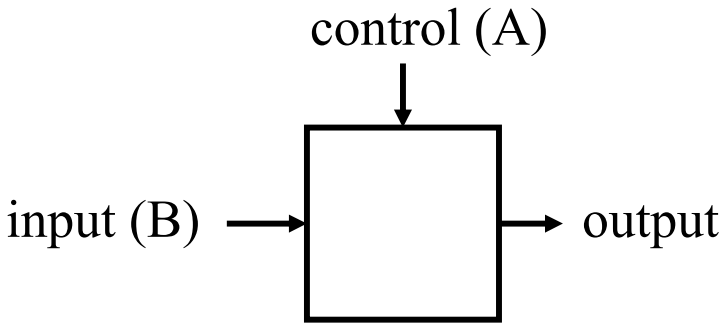
How can an OR gate be made from(a) NOR gates (b) NAND gates?

$F = A + B = \overline{\overline{A + B}}$ applying De Morgan, $F = \overline{\overline{A}.\overline{B}}$





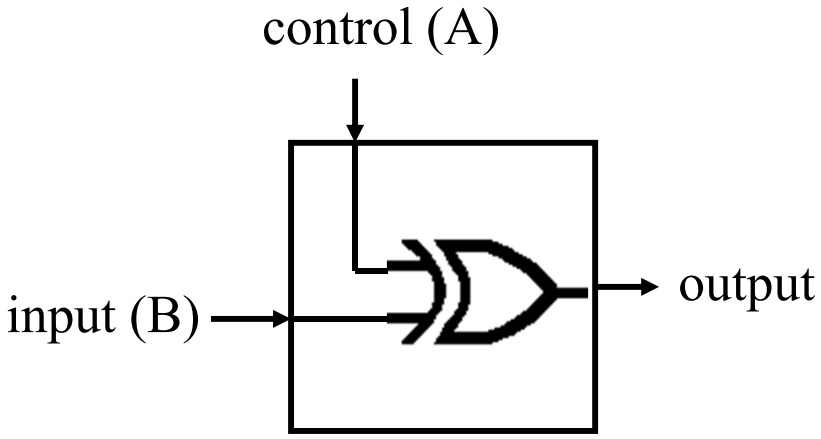
Controlled Inverter



when control (A) = **0**, output = input

when control (A) = **1**, output = $\overline{\text{input}}$

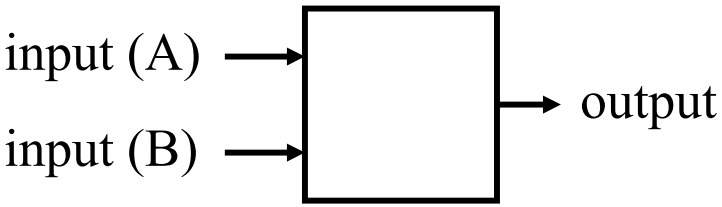
A	B	output
0	0	0
0	1	1
1	0	1
1	1	0



$\mathbf{0 \oplus B = B}$

$\mathbf{1 \oplus B = \overline{B}}$

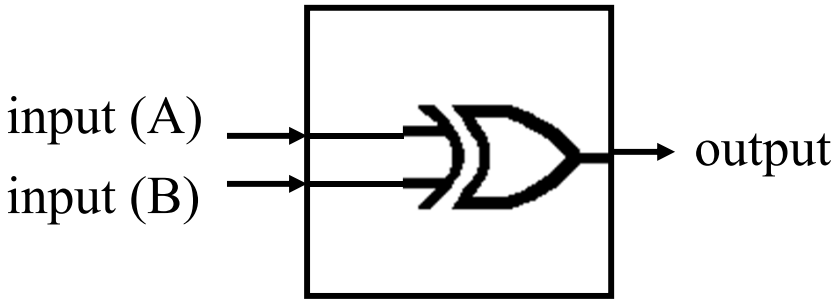
Bitwise Comparator



when the input bits are the same,
output = **0**

when the input bits differ,
output = **1**

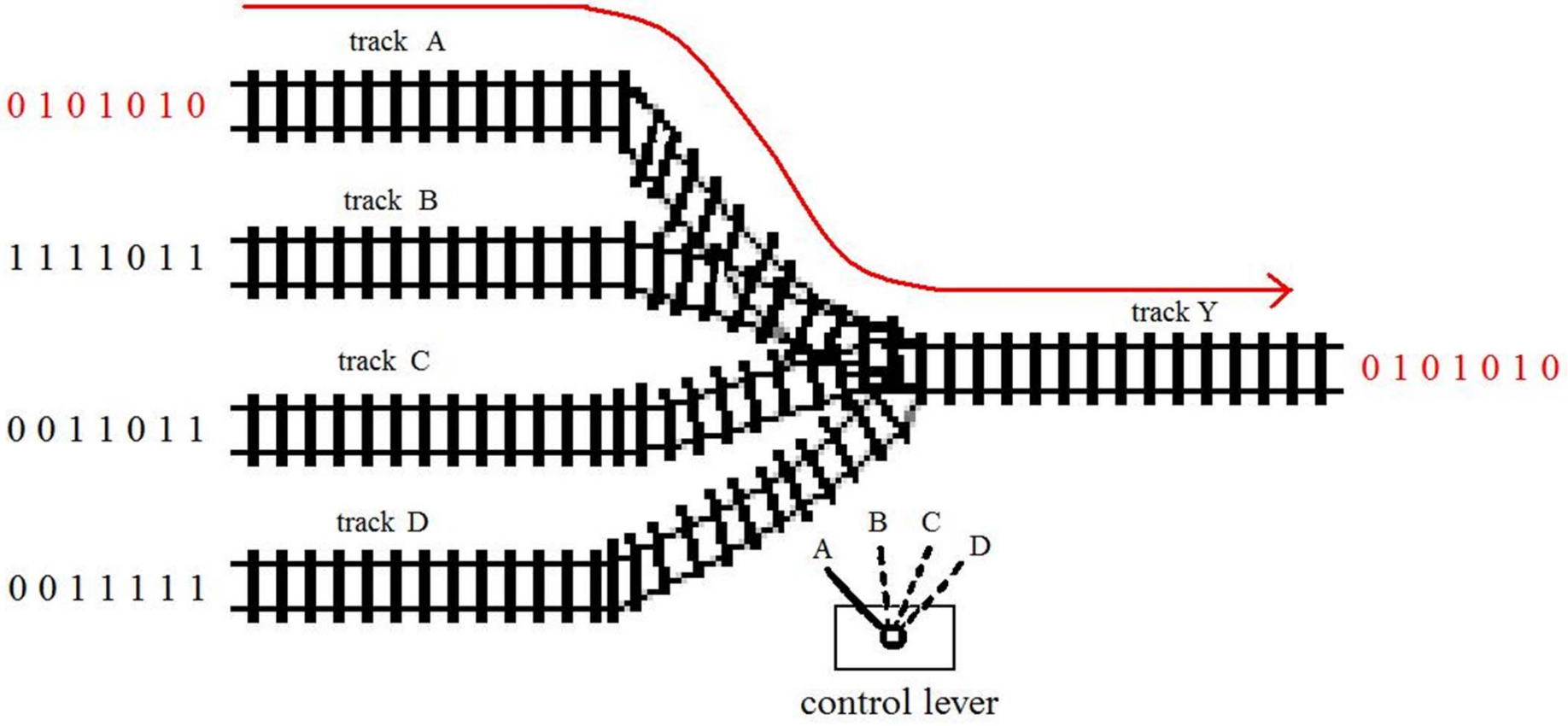
A	B	output
0	0	0
0	1	1
1	0	1
1	1	0



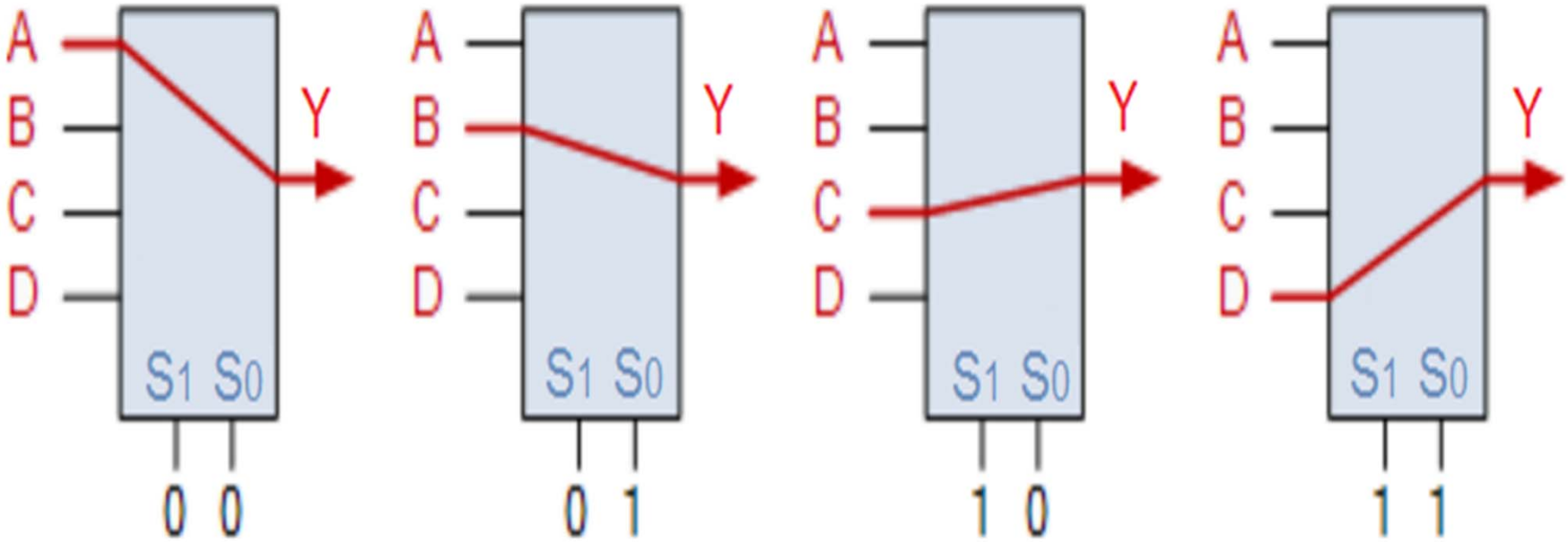
$X \oplus X = 0$

$X \oplus \overline{X} = 1$

Multiplexer (Mux) - a digital switch



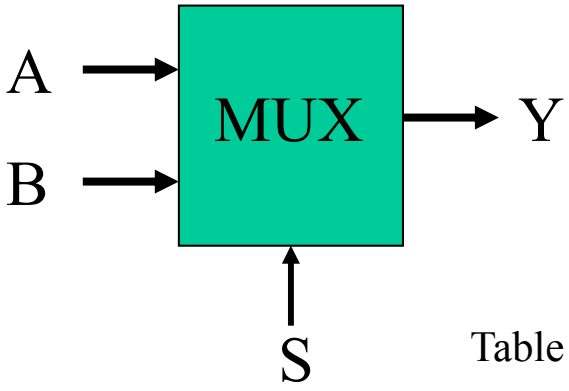
4-to-1 Multiplexer



Two select lines can choose between four data paths because there are four combinations or two bits.

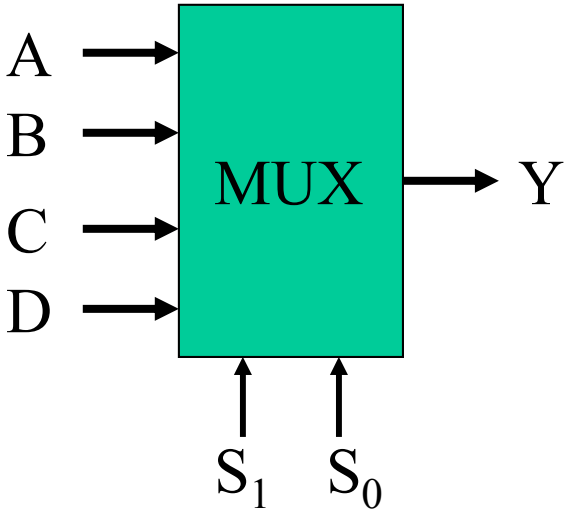
Multiplexers

A multiplexer is basically a switch that selects binary data from one of several input lines and directs it to a single output.

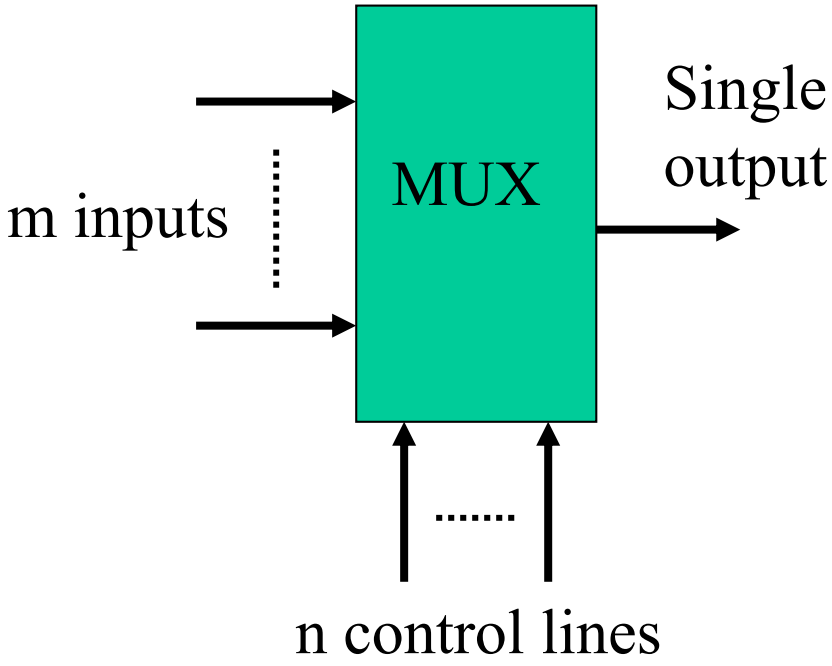


S	Y
0	A
1	B

Table entered variable



S_1	S_0	Y
0	0	A
0	1	B
1	0	C
1	1	D

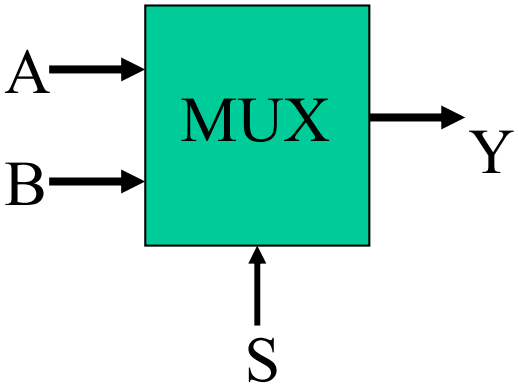


$$m = 2^n$$

Multiplexer Logic

2-1 mux truth tables

S	Y
0	A
1	B



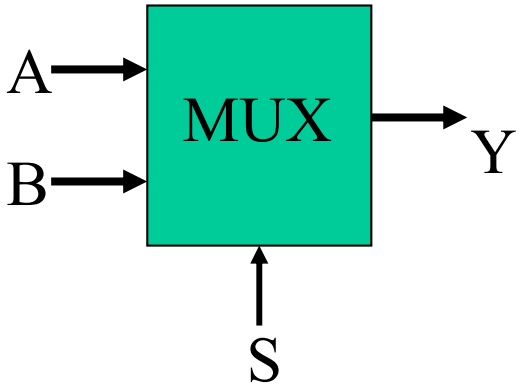
S	A	B	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Multiplexer Logic

2-1 mux truth tables

S	Y
0	A
1	B

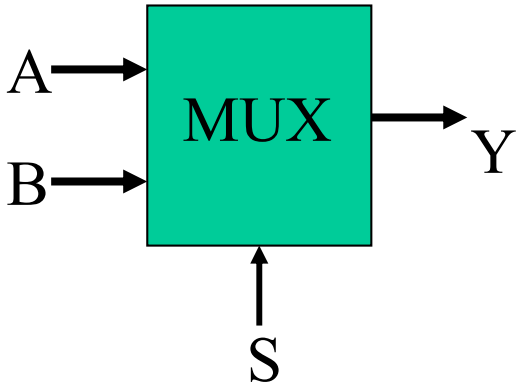
S	A	B	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Multiplexer Logic

2-1 mux truth tables

S	Y
0	A
1	B



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

2-1 mux logic expression

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

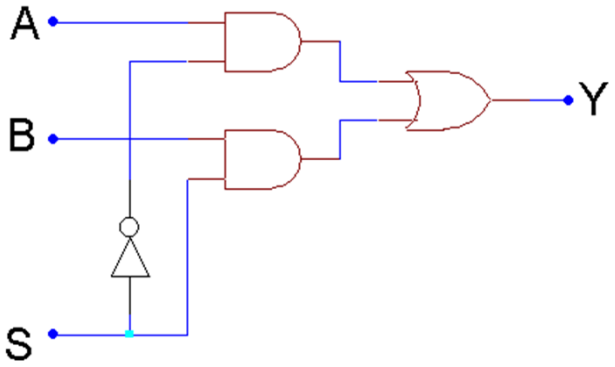
Simplify, trying to separate A and B

$$Y = \overline{S}.A.(\overline{B} + B) + S.B.(\overline{A} + A)$$

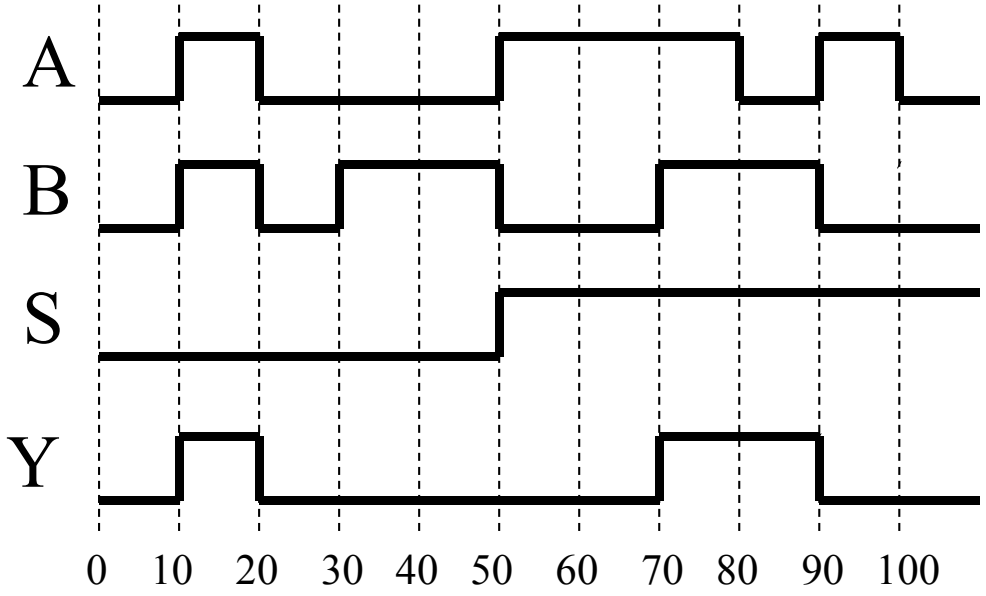
$$\boxed{Y = \overline{S}.A + S.B}$$

Multiplexer circuit

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



Timing diagram for a sample input sequence.



AND Function

X	Y	X.Y
0	0	0
0	1	0
1	0	0
1	1	1

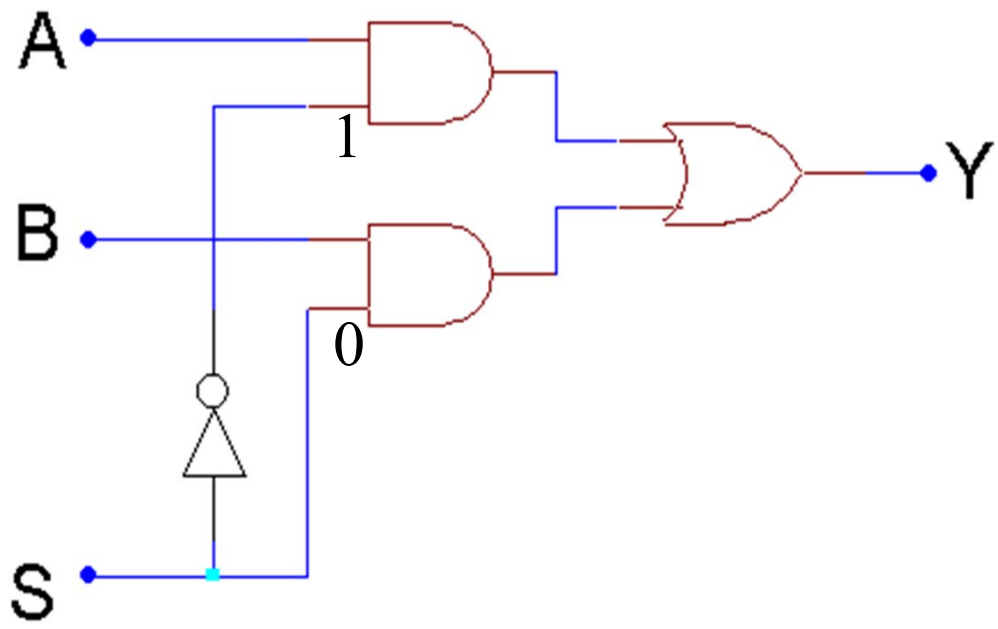
AND Function

X	Y	X.Y
0	0	0
0	1	0
1	0	0
1	1	1

Multiplexer circuit

S	Y
0	A
1	B

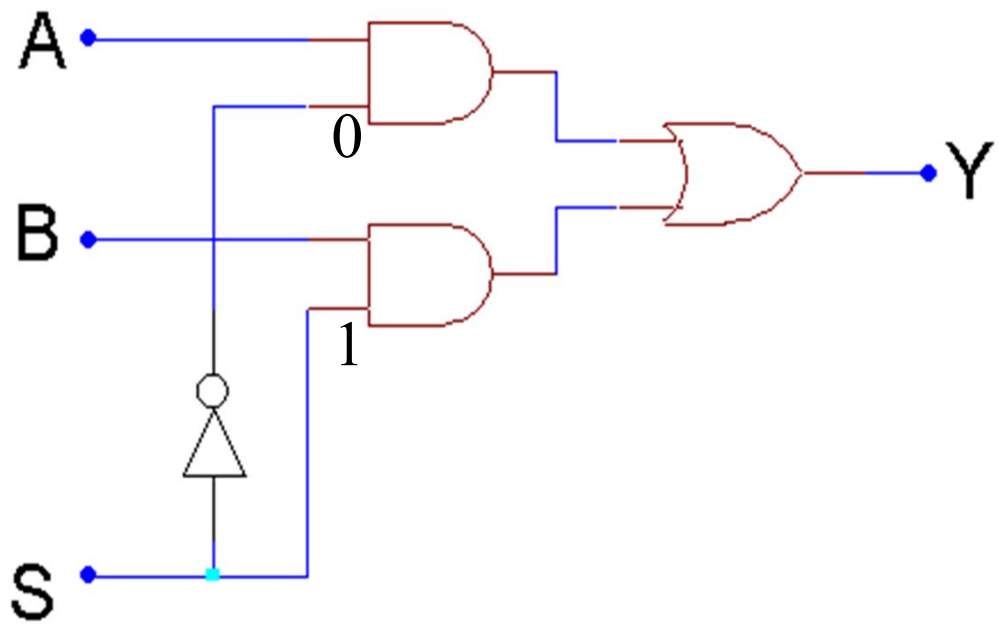
S = 0



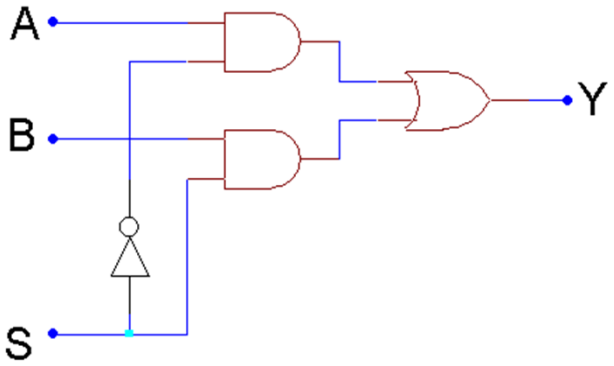
Multiplexer circuit

S	Y
0	A
1	B

S = 1



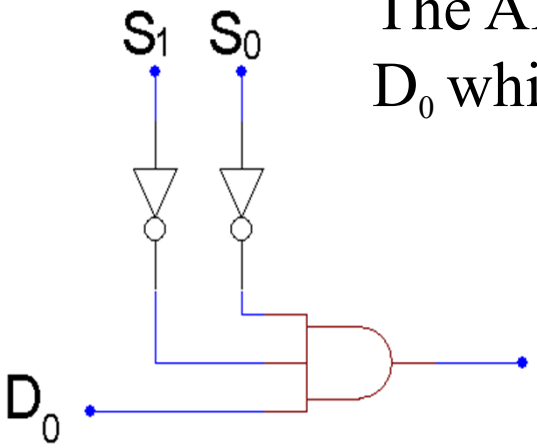
Multiplexer circuit



Only one of the AND gates is ever activated.

The OR gate is used to feed this to the output.

If there are two control lines, S_1 and S_0, they can be decoded to activate a 3-input AND gate.

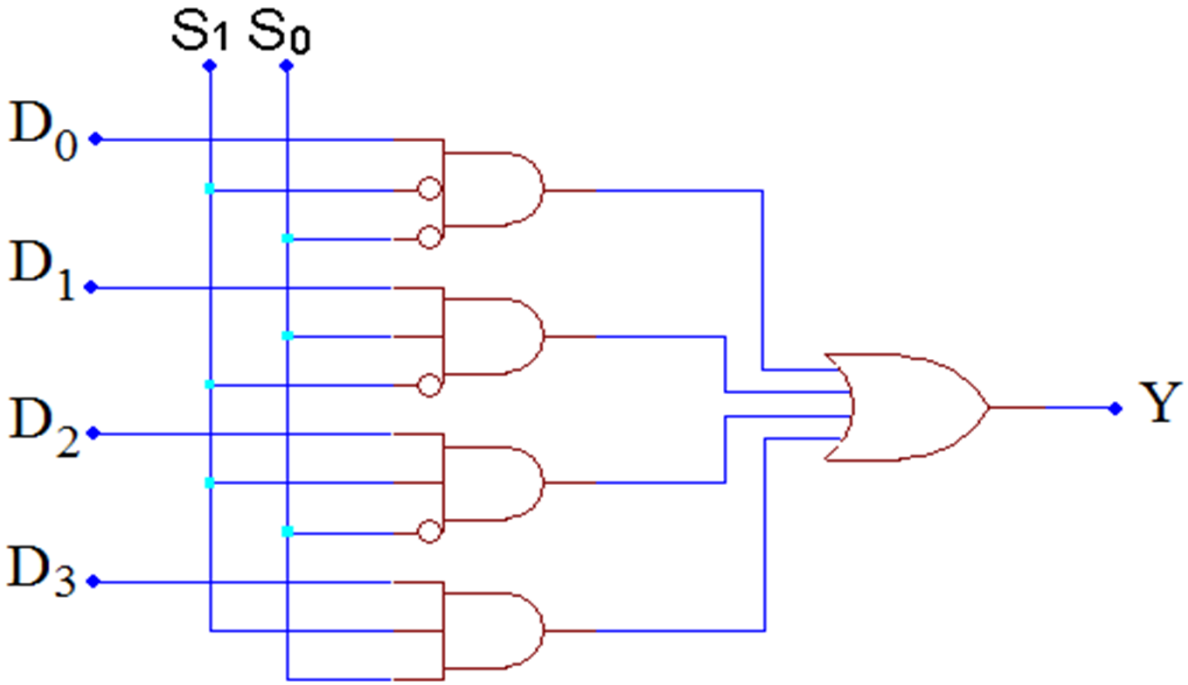


The AND gate shown controls the passage of data bit D_0 which will be selected for an input of S_1= **0** S_0= **0**

The presence of **00** on the select lines effectively activates the gate, passing D0. Any other input combination will produce a **0** on the AND gate output.

Circuit diagram for a 4-1 multiplexer

S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3



The same technique can be used for 3 or more select lines.

Summary

- Timing diagrams can be used to show the operation of digital logic circuits.
- NAND and NOR gates can be used to implement any logic function.
- XOR gates can be used as controlled inverters and bitwise comparators.
- Multiplexers can be used to steer data to a single output