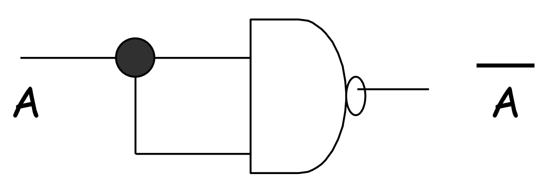
NAND and NOR are universal gates

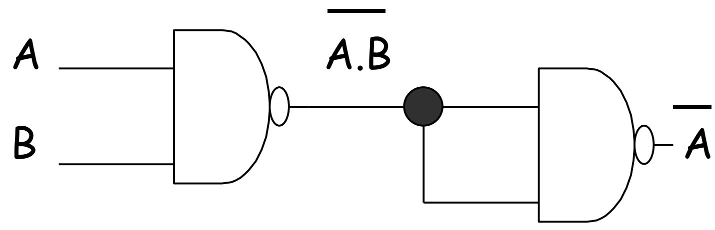
Any function can be implemented using **only NAND** or **only NOR** gates. How can we prove this?

(Proof for NAND gates) Any boolean function can be implemented using AND, OR and NOT gates. So if AND, OR and NOT gates can be implemented using NAND gates only, then we prove our point.

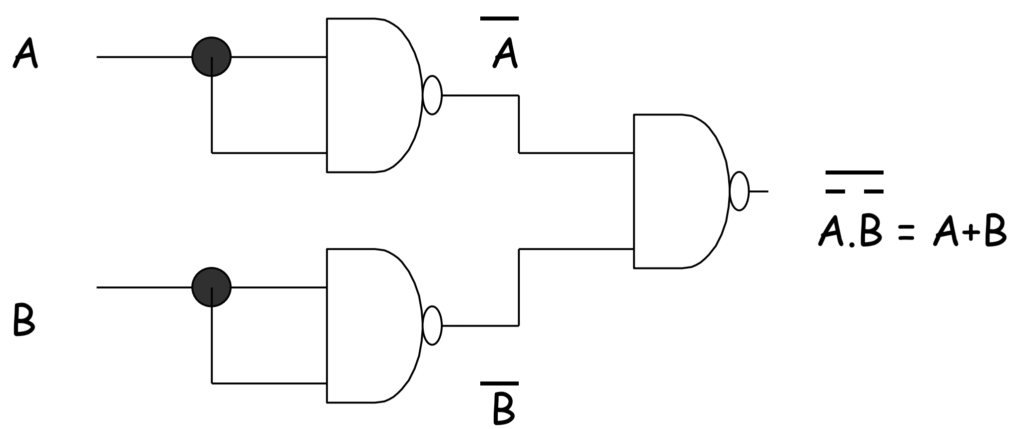
1. Implement NOT using NAND



2. Implementation of AND using NAND



1. Implementation of OR using NAND



(Exercise) Prove that NOR is a universal gate.

Additional properties of XOR

XOR is also called modulo-2 addition.

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

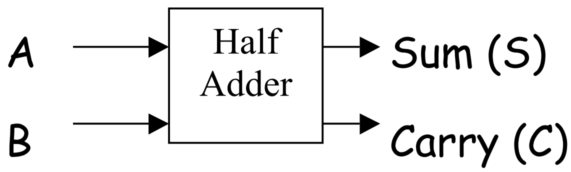
$A \oplus B = 1$ only when there are an odd number of 1's in (A,B). The same is true for $A \oplus B \oplus C$ also.

$$\left.\begin{aligned} 1 \oplus A &= \overline{A} \\ 0 \oplus A &= A \end{aligned}\right\}$$

Why?

Logic Design Examples

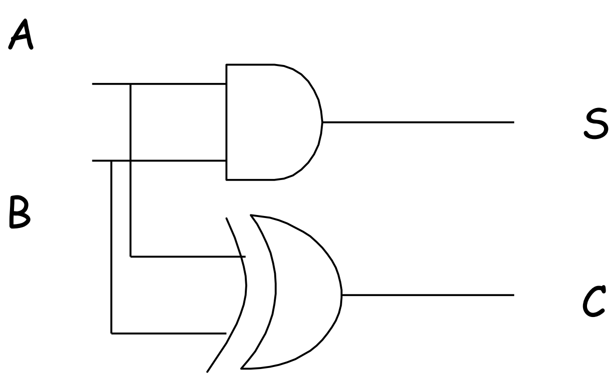
Half Adder



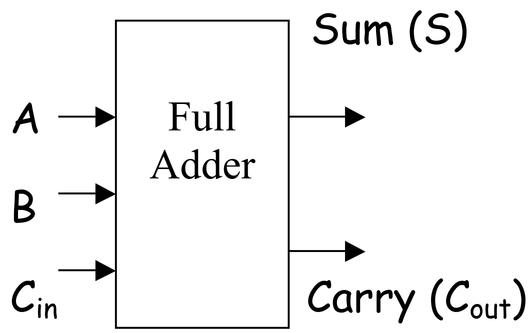
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = A \oplus B$

$C = A.B$



Full Adder



A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

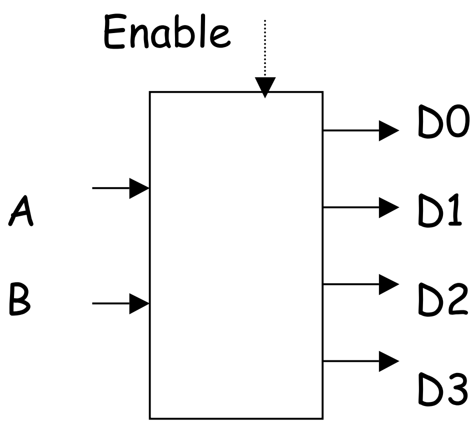
$S = A \oplus B \oplus C_{in}$

$C_{out} = A.B + B.C_{in} + A.C_{in}$

Can you design a full adder using two half-adders (and a few gates if necessary)?

<u>Decoders</u>

A typical decoder has n inputs and 2^n outputs.



A	B	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

A 2-to-4 decoder and its truth table

$D3 = A.B$

$D2 = A.\overline{B}$

$D1 = \overline{A}.B$

$D0 = \overline{A}.\overline{B}$

Draw the circuit of this decoder.

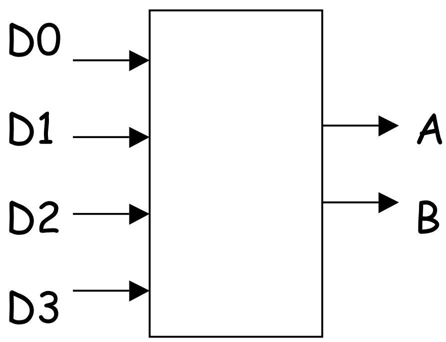
The decoder works per specs when (**Enable = 1**). When **Enable = 0**, all the outputs are 0.

Exercise. Design a 3-to-8 decoder.

Question. Where are decoders used?

Encoders

A typical encoder has 2^n inputs and n outputs.



D0	D1	D2	D3	A	B
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

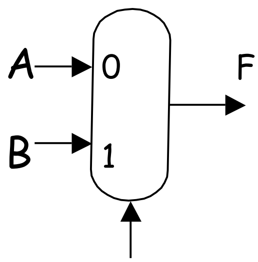
A 4-to-2 encoder and its truth table

$$A = D1 + D3$$

$$B = D2 + D3$$

Multiplexor

It is a **many-to-one switch**, also called a **selector**.



Control S

S = 0, F = A

S = 1, F = B

Specifications of the mux

A 2-to-1 mux

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

Exercise. Design a 4-to-1 mux.