DIGITAL LOGIC

EXAMPLE PROBLEMS

A	B	C	O
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

O = f(A, B, C)

How do I express this truth table?

A’B’C + A’BC’ + AB’C’ + AB’C + ABC’ + ABC = O

This is the sum of products

O = f(A, B, C)

A	B	C	O
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

An alternative way is to express the product of sums.

For the rows where the output is zero, we take the sum of the inverse of the inputs.

(A+B+C)*(A+B'+C') = O

The two are equivalent:
A'B'C + A'BC' + AB'C' + AB'C + ABC' + ABC = (A+B+C)*(A+B'+C') = O

Just for fun

The two are equivalent:

A'B'C + A'BC' + AB'C' + AB'C + ABC' + ABC = (A+B+C)*(A+B'+C') = O

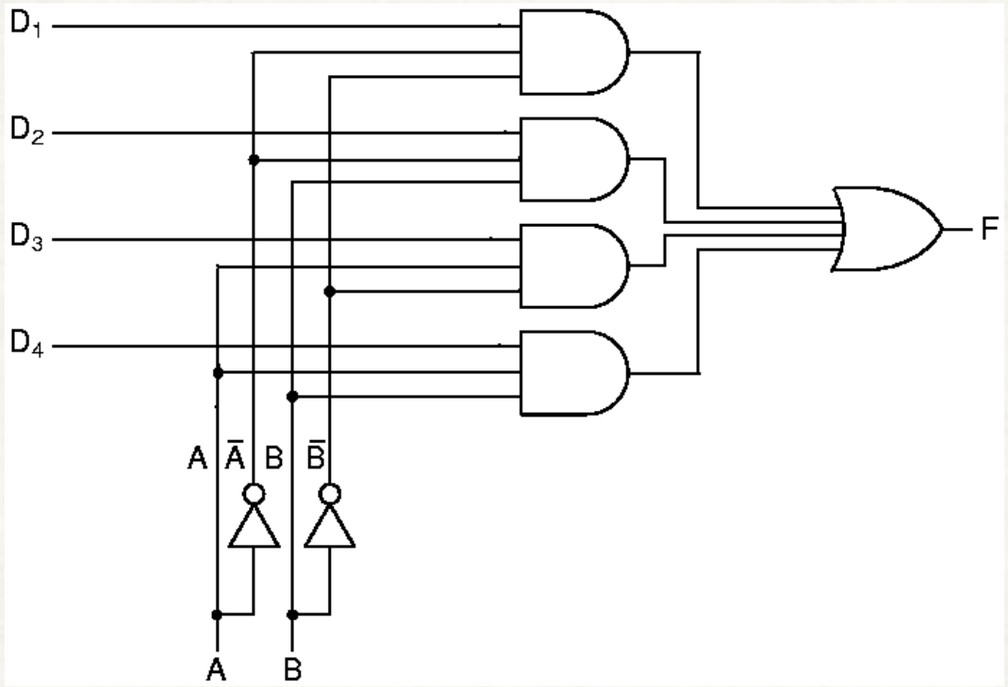
A'B'C + A'BC' + AB'C' + AB'C + ABC' + ABC = (A'+A)(B'C+BC')+A(BC+B'C'+B'C+BC') = B'C + BC' + A

(A+B+C)*(A+B'+C') = A+AB'+AC' + AB+BC' + AC+B'C

= A+A(B'+B)+A(C'+C) + BC' + B'C = A + B'C + BC'

Multiplexer circuit

1. ***Selects*** **one data channel (from many input channels)**
2. ***Maps*** **the selected channel to the output channel**



Sum-of-Products circuit

$D_1 \ldots D_4$	:	Data channels
A, B	:	Control lines
F	:	Output channel

If we want a multiplexer that chooses one of eight possible outputs, how many control signal bits do we need?

If we want a multiplexer that chooses one of three possible outputs, how many control signal bits do we need?

Decoder circuit

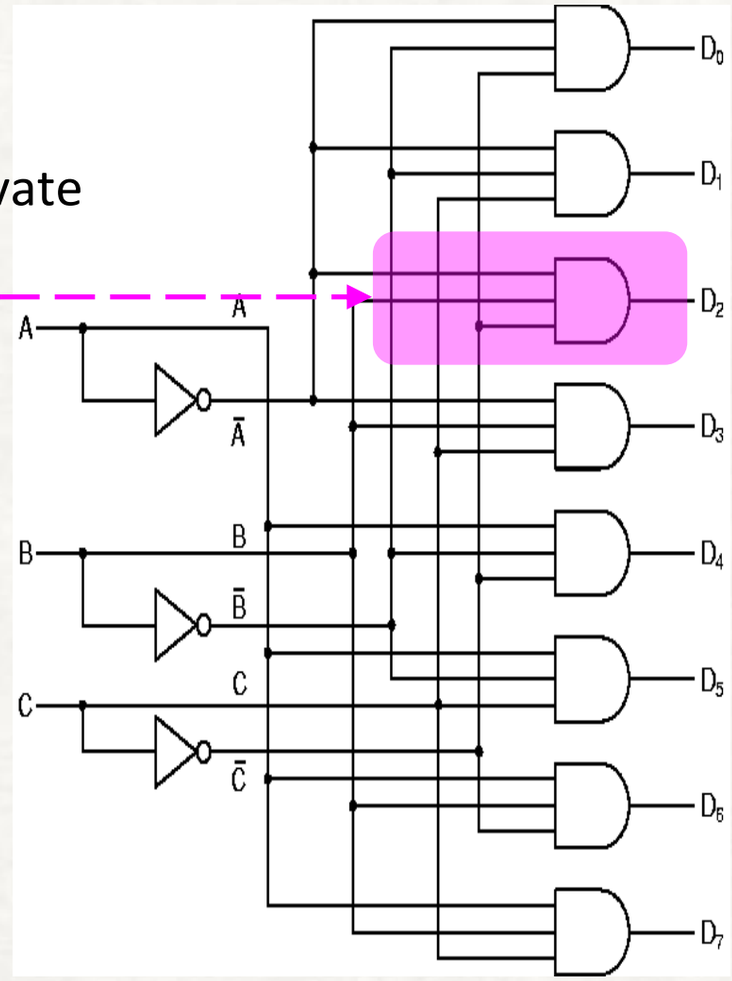
1. **Given** *n* control lines
2. **Select** one of 2^n outputs to activate

Simple AND-gate circuit

$D_1 \ldots D_8$: Output channels

A, B, C : Control lines

Note: *n* control lines, 2^n outputs



Laws of Boolean Logic

Regulate circuit function Identity

Law $1 \wedge A = A,\ A \in \{0,1\}$

Example: Select D_2: A,C=0; B=1

How many control signal bits are needed to choose one of 32 possible outputs

What should the control signal bits be to choose output 30

Majority Function

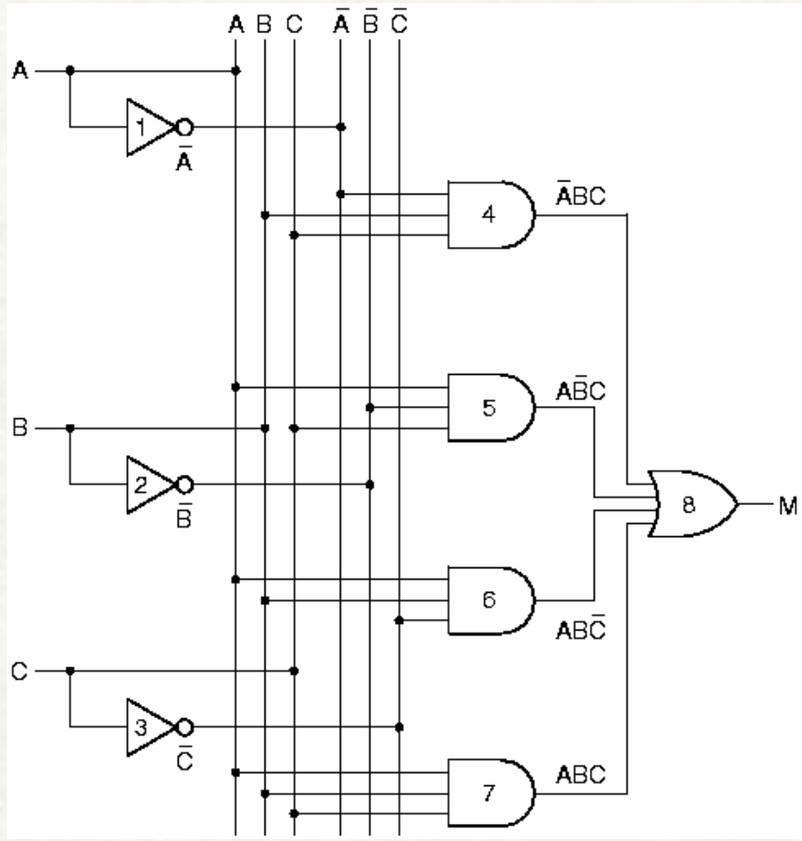
- Returns 1 if a majority of inputs are high (=1)
- Formula: $M(A,B,C) = A'BC + AB'C + ABC' + ABC$

Truth Table

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

⇒ **SOP Circuit:**

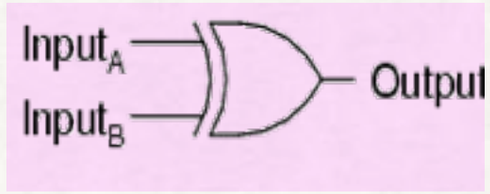
[Note: $\overline{A}$ = NOT(A)]



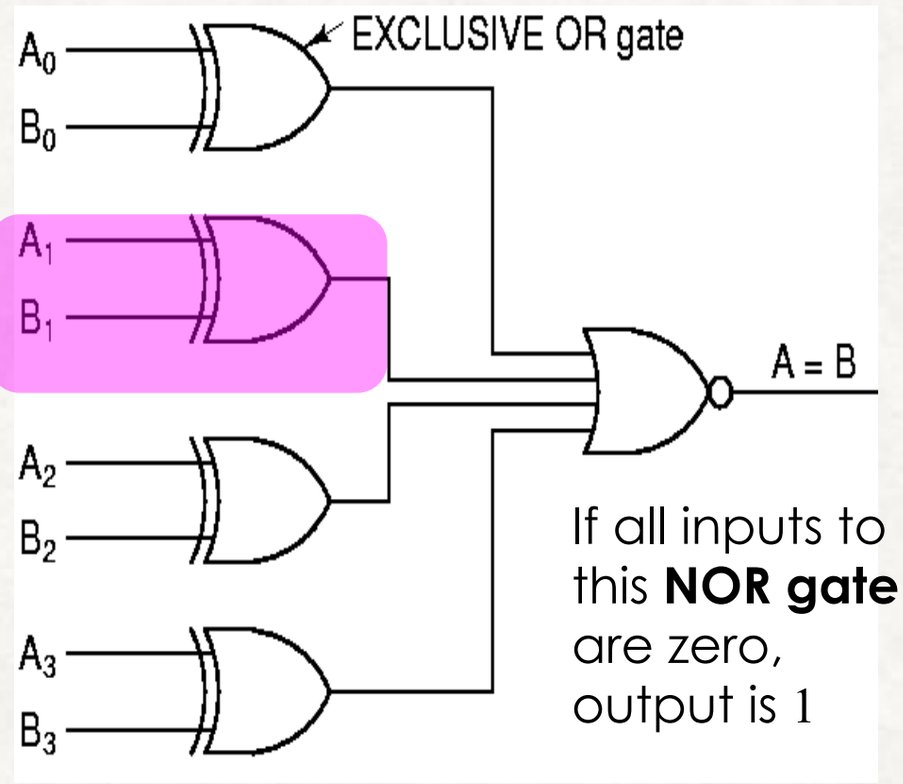
Application: In the Space Shuttle, 3 redundant circuits each compute the same function *f*. If a MAJORITY of circuit outputs agree (=1), then the 1-valued result is assumed to be correct.

Comparator circuit

1. **Given** two *n*-bit vectors **A** and **B**
2. **Output** 1 if **A = B**, or 0 otherwise



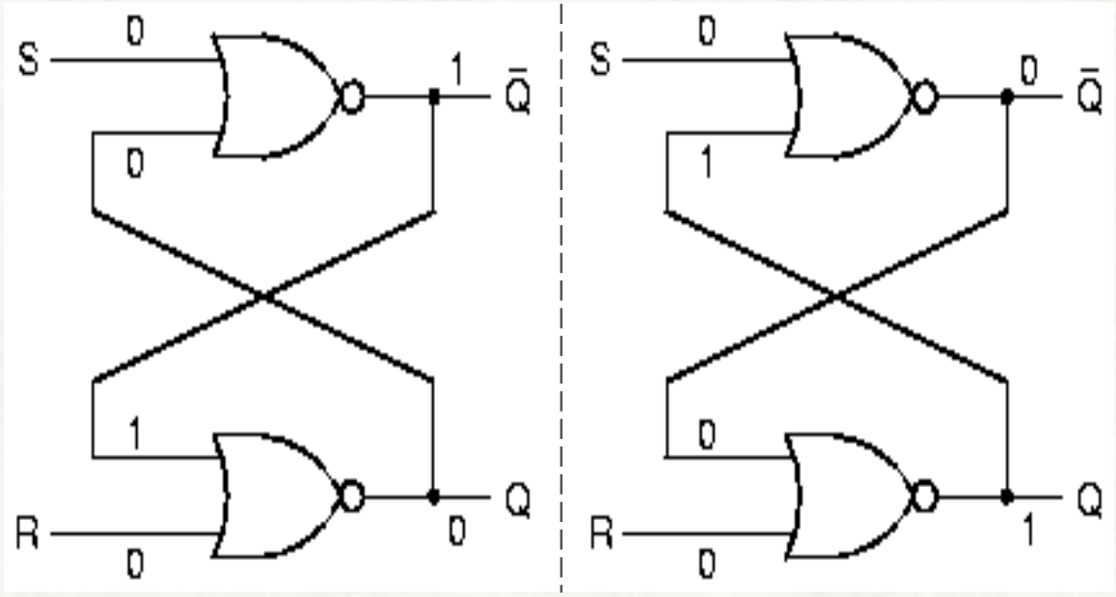
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



Laws of Boolean Logic

Example: Comparator for Bit i outputs 0 if $A_i = B_i$

NOR S-R Latch

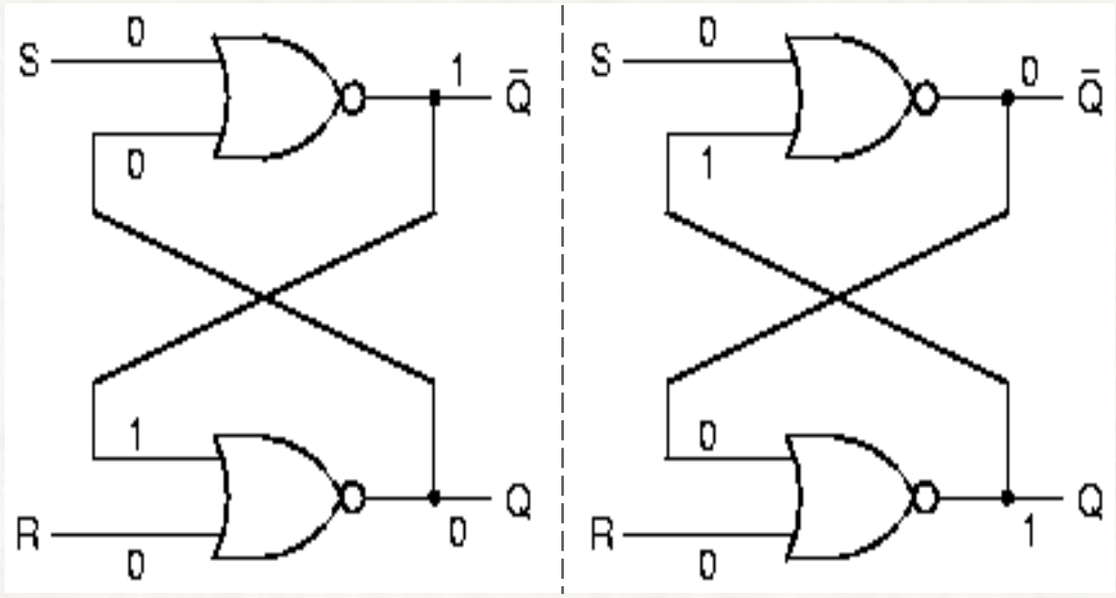


What is this used for?

Memory cell.

How does it work

The output becomes one when S is asserted. It remains at 1 after S is de-asserted and until R is asserted



An S-R latch begins in the state on the left, with Q = 0 and Q'=1

What is the value of Q when S is set to 1
What is the value of Q when S is subsequently set to 0

What is the value of Q when R is subsequently set to 1
What is the value of Q when R is subsequently set to 0