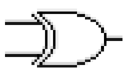
Laboratory 6
Basic Digital Circuits

Exclusive OR (XOR)

$F = AB' + A'B = A \oplus B$

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

Available on IC as a gate, useful for comparison problems



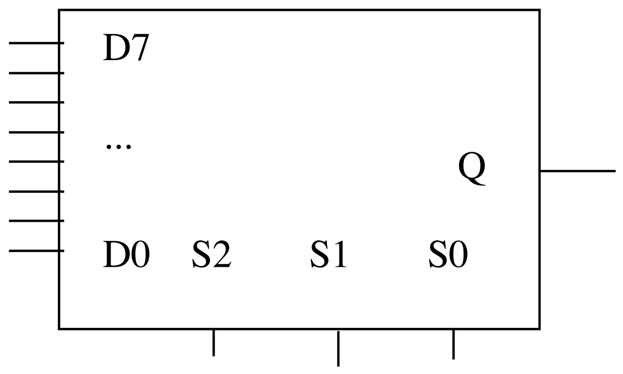
Example: Even parity $F = A \oplus B \oplus C$

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

Multiplexer

- n select lines
- 2^n input lines
- 1 output

One of the possible 2^n inputs is chosen by the n select lines, and gated through to the output of a multiplexer.

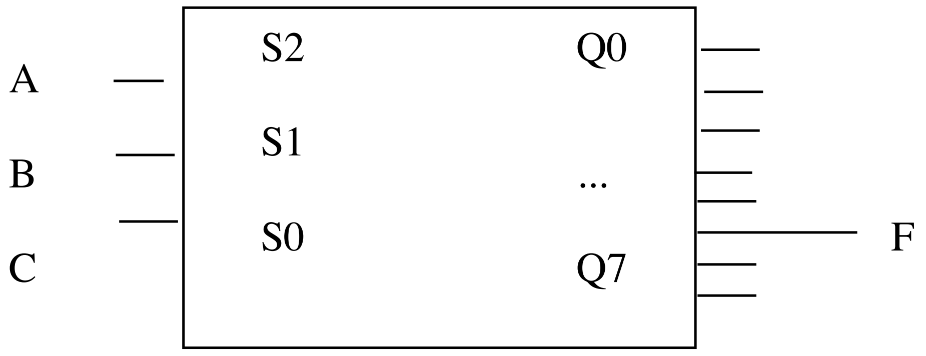


S2	S1	S0	Q
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

Mulitplexers are usually used for **selection**, but can also act as code detectors.

Decoder

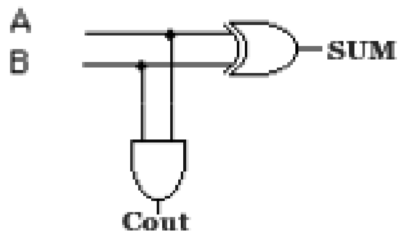
- n input/select lines
- 2^n outputs
- only one of the outputs is active at any given time, based on the value of the n select lines.



S2	**S1**	**S0**	**Q0**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

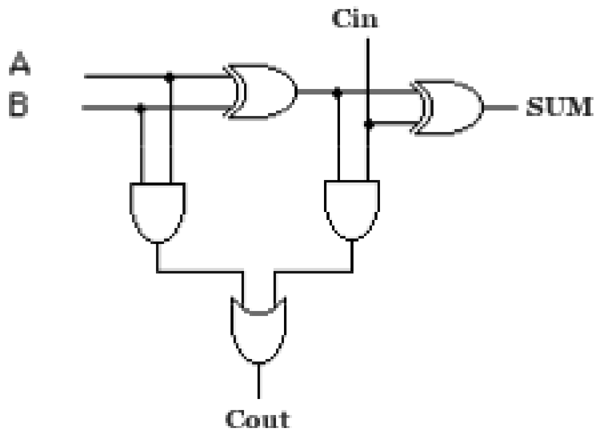
Decoders may also be used as code detectors

Half-Adder — adds two one-bit values



A	**B**	**Sum**	**Cout**
0	0		
0	1		
1	0		
1	1		

Full Adder — incorporates a carry-in



A	**B**	**Cin**	**Sum**	**Cout**
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

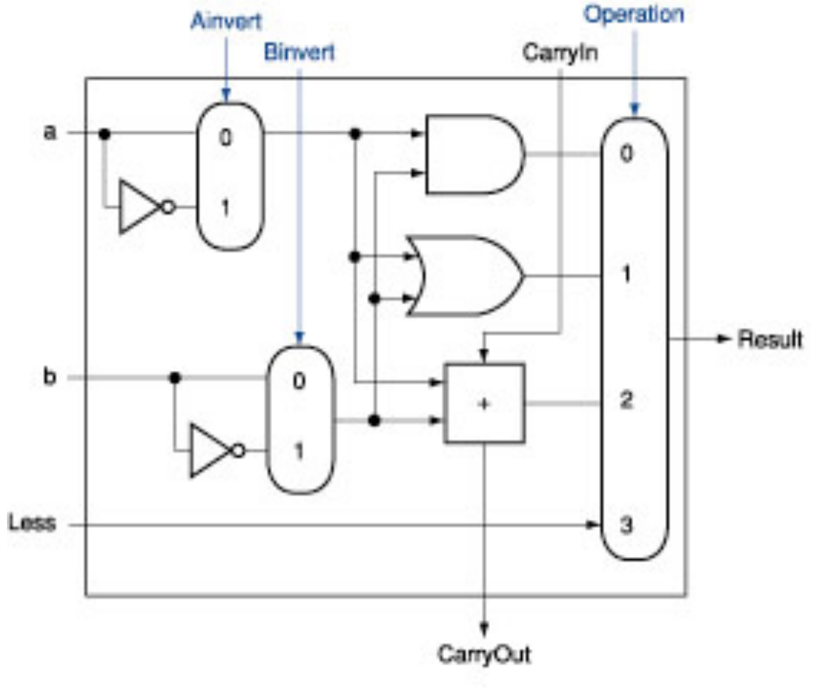
$\text{Sum} = A \oplus B \oplus \text{Cin}$

$\text{Cout} = AB + (A \oplus B)\text{Cin}$

n-bit adder = n 1-bit adders

Carry-out of each adder = Carry-in of the adder for next two most significant bits being added

Arithmetic Logic Unit (ALU)



Ainvert	Binvert	Cin	Op1	Op0	Result
0	0	X	0	0	a AND b
0	0	X	0	1	a OR b
0	0	X	1	0	a + b (add)
0	1	1	1	0	a – b
0	1	1	1	1	Set on Less Than

addition (a + b) **subtraction** (invert b, c_{in} = 1, a + b)

AND gate (a AND b) **OR** gate (a OR b)

NOR (invert a, invert b, a AND b)

SLT (invert b, c_{in} = 1, so that a – b is performed by the adder; sign bit and overflow bit used to set Less bit which goes to LSB of result)

```
if a < b
  LSB of result = '1', other result bits = '0'
else
  all bits of result = 0
```