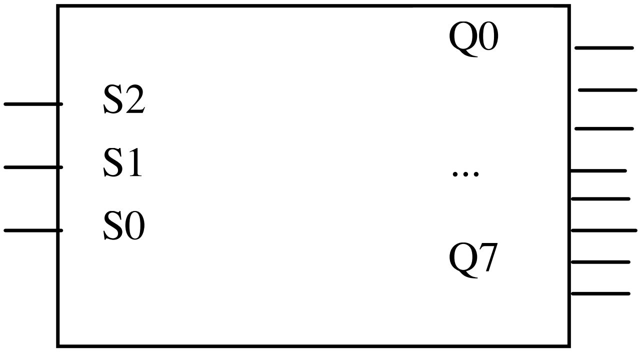
Laboratory 3

Basic Digital Circuits

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.

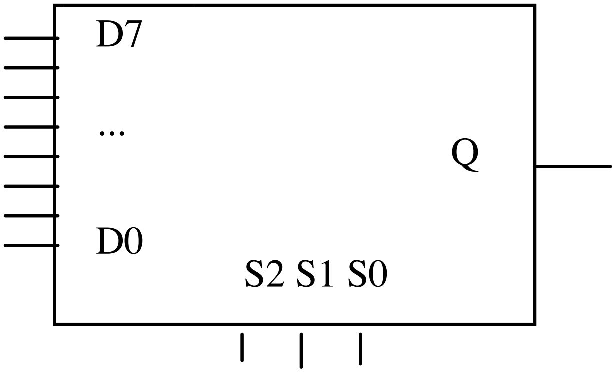


S2	S1	S0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Multiplexer

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

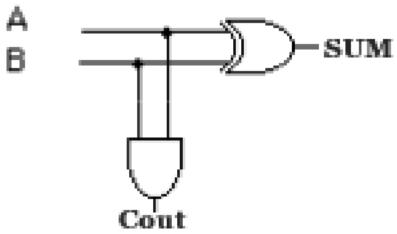
One of the possible 2^n inputs is chosen by the n select lines, and gated through to the output of a multiplexer.



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

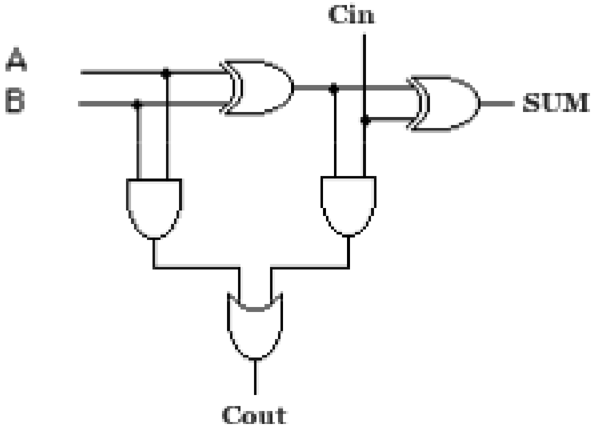
Mulitplexers are usually used for **selection**, but can also act as code detectors.

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

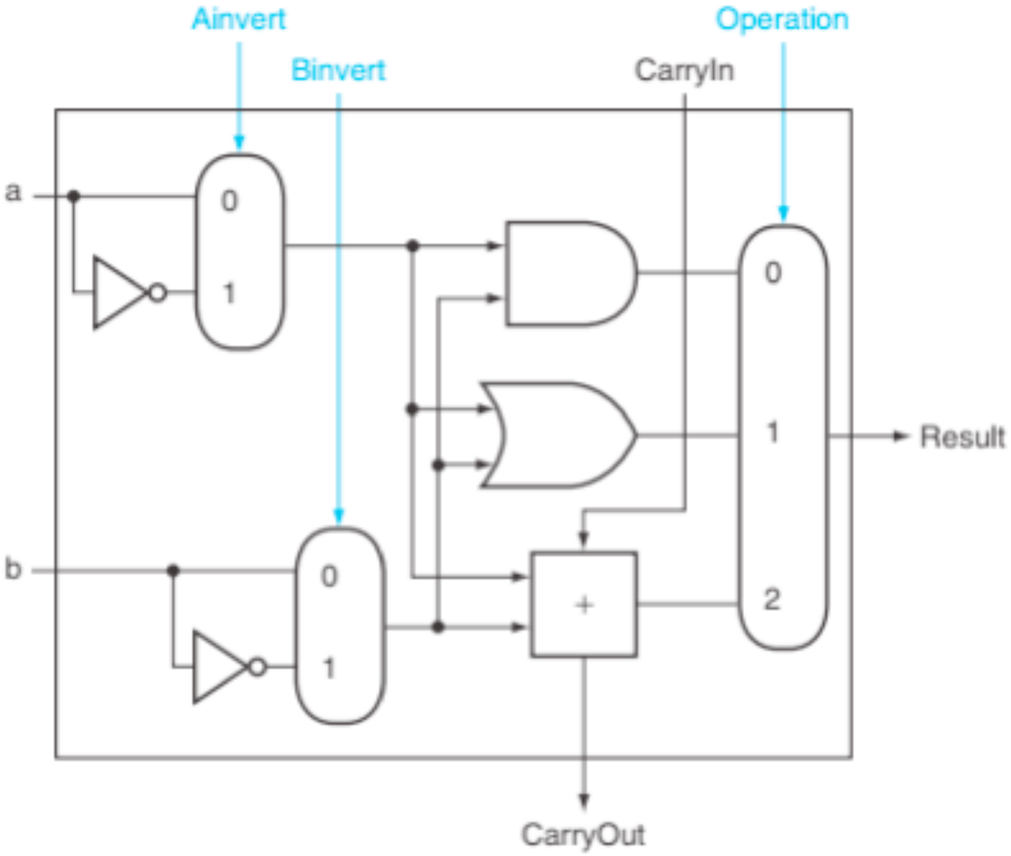
$\text{Sum} = A \oplus B \oplus \text{Cin}$

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

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)



Ainv	Binv	Cin	Op1	Op0	Result	Cout	Zero	Overflow
0	0	X	0	0	a AND b			
0	0	X	0	1	a OR b			
0	0	0/1	1	0	a + b			
0	1	1	1	0	a — b			
1	1	X	0	0	a NOR b			

add (a + b + Cin) **sub** (invert b, Cin = 1, a + b + Cin)

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

Cout = 1 if (adder produces a carry-out == 1)

Zero = 1 if (all bits of result == 0)

Overflow = 1 if (Cin XOR Cout == 1)