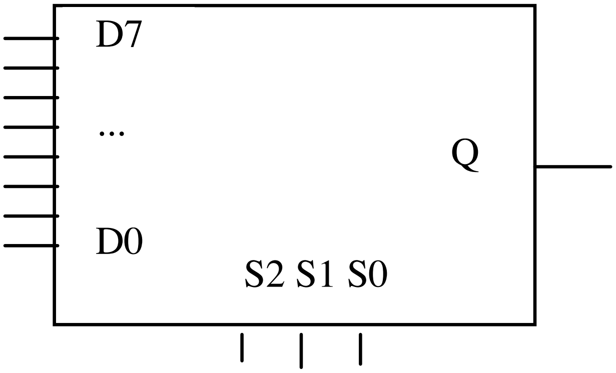
CS 240 Lab 3
Basic Digital Circuits

- **Multiplexer**
- **Decoder**
- **Adder**
- **Two’s Complement and Overflow**

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.

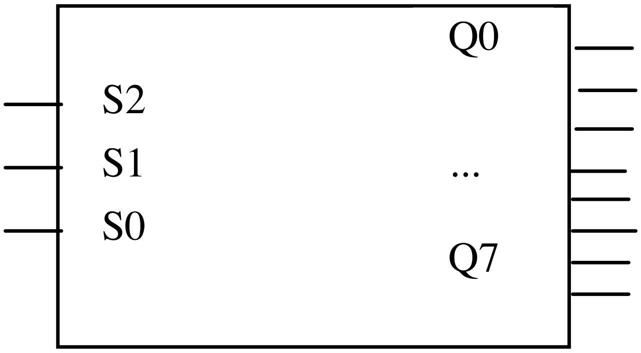


S2	S1	S0	Q
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

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

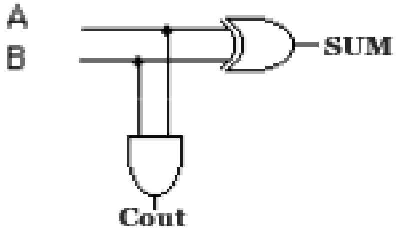
Decoder

- 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.



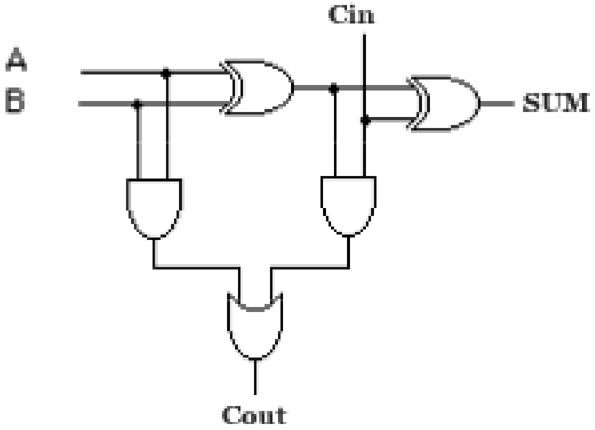
S2	S1	S0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

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

Two’s Complement and Overflow

Given n bits, the range of binary values which can be represented using

Unsigned representation: $0 \rightarrow 2^{n} - 1$

Signed representation: $-2^{n-1} \rightarrow 2^{n-1} - 1$ because one bit is used for sign

Two’s Complement (signed representation):
Most significant /leftmost bit (0/positive,1/negative)

Example: given a fixed number of 4 bits:
1000_2 is negative.
0111_2 is positive.

Overflow
Given a fixed number of n available bits:
Overflow occurs if a value cannot fit in n bits.

Example: given 4 bits:
The largest negative value we can represent is -8_{10} (1000_2).
The largest positive value we can represent is $+7_{10}$ (0111_2).

Overflow in Addition
When adding two numbers with the same sign which each can be represented with n bits, the result may cause an overflow (not fit in n bits).

An overflow occurs when adding if:

Two positive numbers added together yield a negative result, or
Two negative numbers added together yield a positive result, or
The carry-in and carry-out bits to the most significant pair of bits being added are not the same.

An overflow cannot result if a positive and negative number are added.

Example: given 4 bits:

$$\begin{array}{rl} & 0111 \\ + & 0001 \\ \hline & 1000 \end{array} \quad \text{overflow} \qquad \text{NOTE: there is not a carry-out!}$$

In two’s complement representation, a carry-out does not indicate an overflow, as it does in unsigned representation.

Example: given 4 bits,

$$\begin{array}{rl} & 1001\ (-7) \\ + & 1111\ (-1) \\ \hline 1 & 1000\ (-8) \end{array} \quad \text{no overflow, even though there is a carry-out}$$