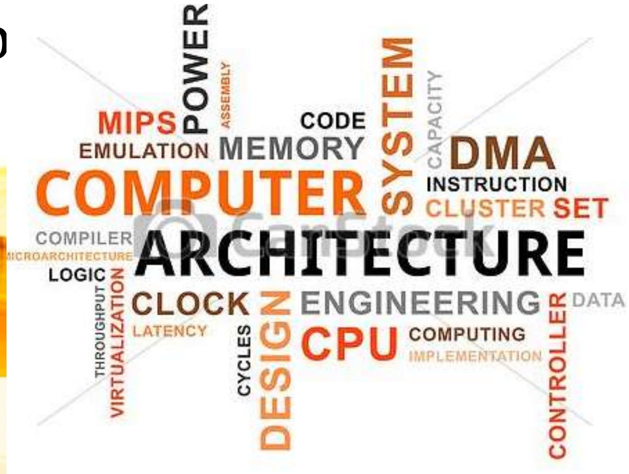
UNIT II
ARITHMETIC OPERATIONS

Addition and subtraction of signed numbers – **Design of fast adders** – Multiplication of positive numbers - Signed operand multiplication- fast multiplication – Integer division – Floating point numb



Recap the previous Class

Binary Adders

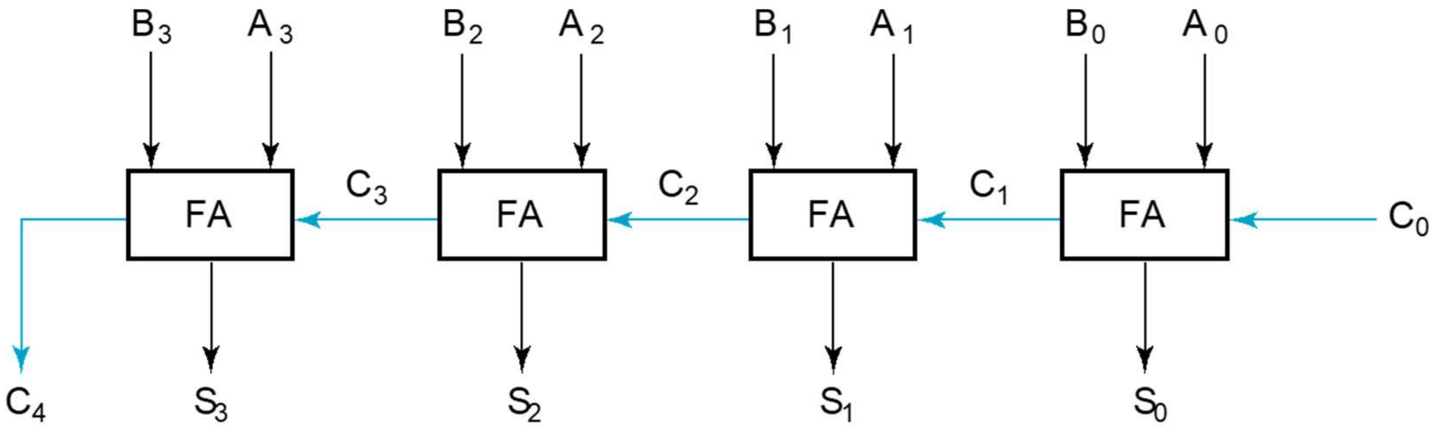
Description	Subscript 3 2 1 0	Name
Carry In	0 1 1 0	Ci
Augend	1 0 1 1	Ai
Addend	0 0 1 1	Bi
Sum	1 1 1 0	Si
Carry out	0 0 1 1	Ci+1

Note:

carry out of cell i becomes carry in of cell i + 1

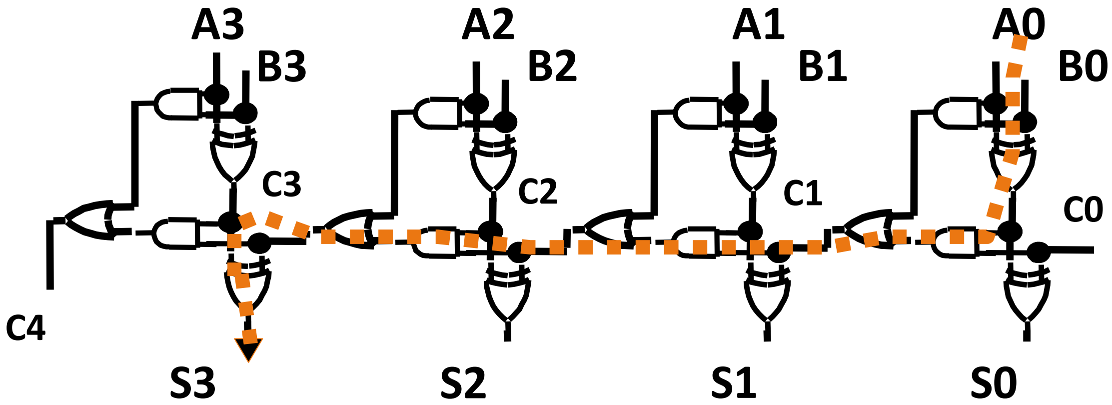
4 bit Ripple carry Adder

- A four-bit Ripple Carry Adder made from four 1-bit Full Adders



Carry Propagation & Delay

- One problem with the addition of binary numbers is the length of time to propagate the ripple carry from the least significant bit to the most significant bit.
- The gate-level propagation path for a 4-bit ripple carry adder of the last example:



Carry Lookahead Adder

$S_i = x_i \oplus y_i \oplus c_i$

$C_{i+1} = x_i y_i + x_i c_i + y_i c_i$

Factorizing

$C_{i+1} = x_i y_i + (x_i + y_i) c_i$

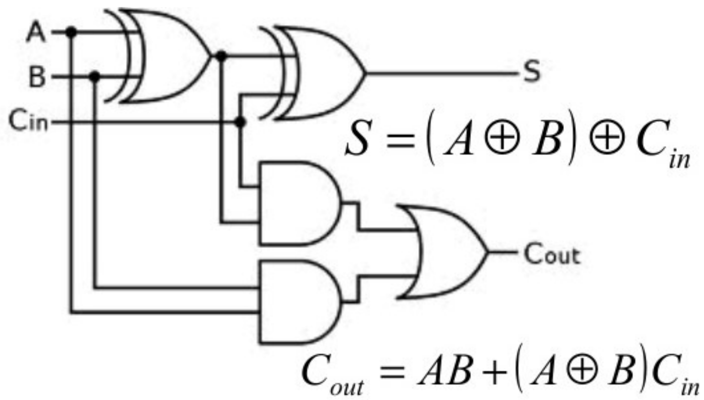
We can write

$C_{i+1} = G_i + P_i c_i$

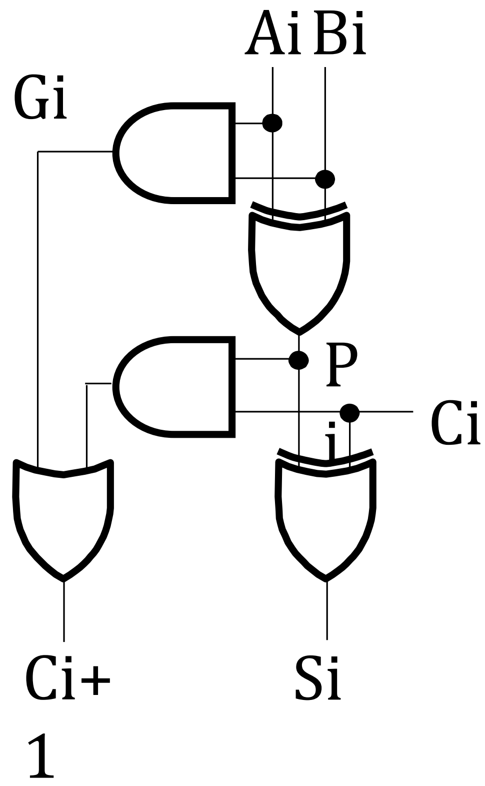
Where

$G_i = x_i y_i$

$P_i = x_i + y_i$



These two signal conditions are called *generate,* denoted as G_i, and *propagate,* denoted as P_i respectively



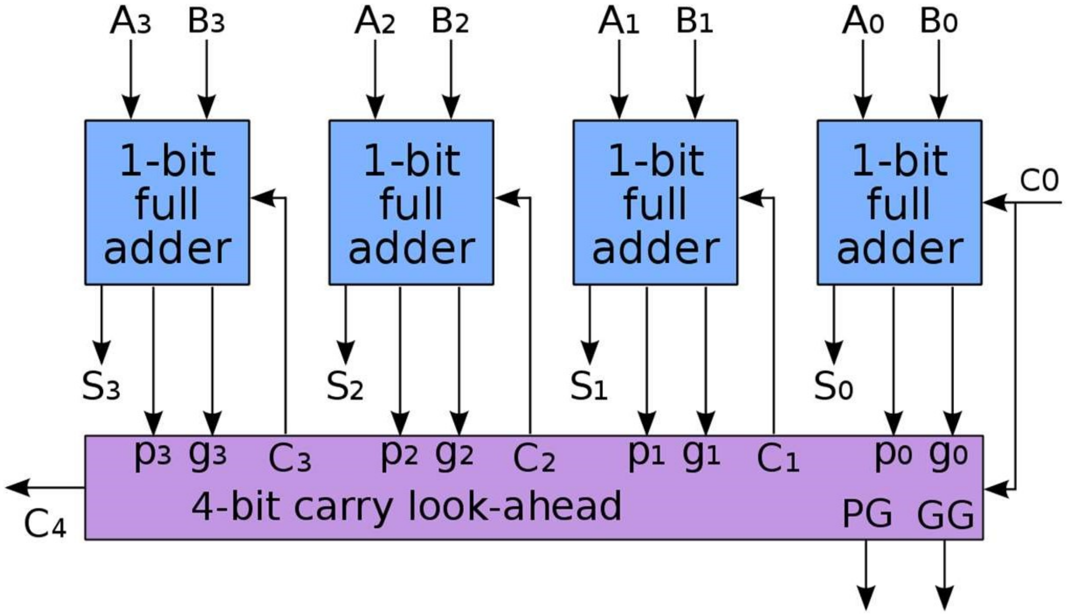
Carry Lookahead Adder

- In the ripple carry adder:
  - Gi, Pi, and Si are local to each cell of the adder
  - Ci is also local each cell
- In the carry lookahead adder, in order to reduce the length of the carry chain, Ci is changed to a more global function spanning multiple cells
- Defining the equations for the Full Adder in term of the P_i and G_i:

$$P_i = A_i \oplus B_i \qquad G_i = A_i B_i$$

$$S_i = P_i \oplus C_i \qquad C_{i+1} = G_i + P_i C_i$$

Carry Lookahead Adder



C1 = G0 + P0 C0

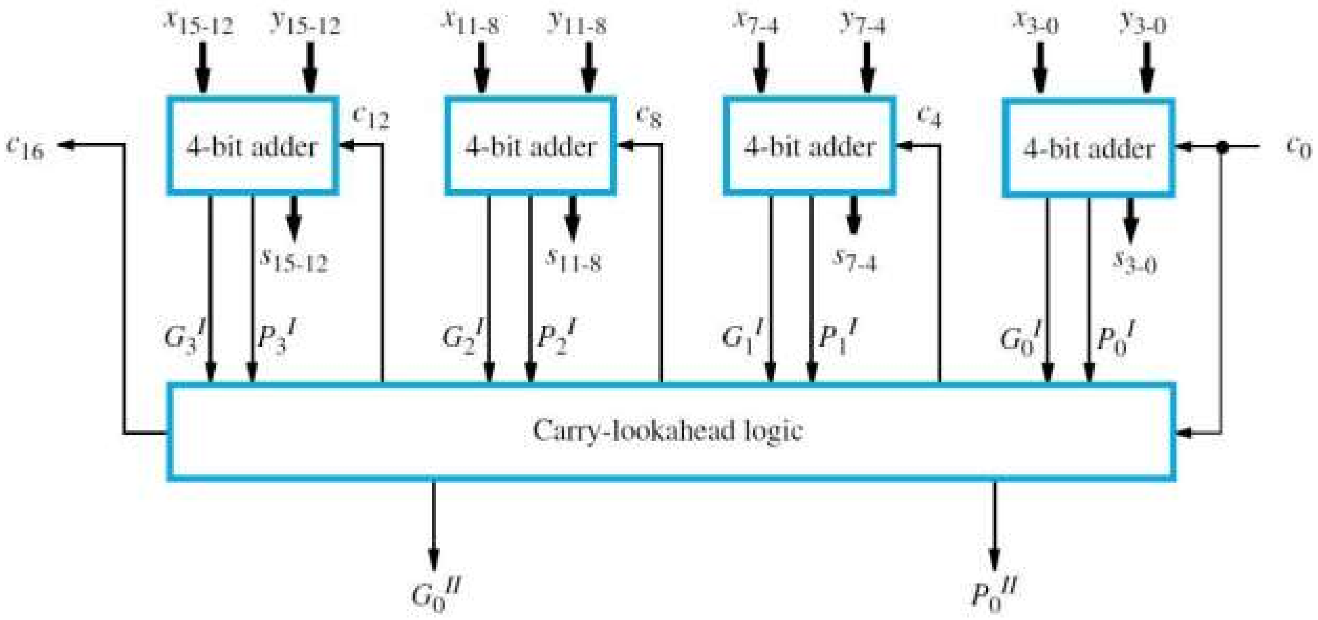
C2 = G1 + P1 C1 = G1 + P1(G0 + P0 C0)
= G1 + P1G0 + P1P0 C0

C3 = G2 + P2 C2 = G2 + P2(G1 + P1G0 + P1P0 C0)
= G2 + P2G1 + P2P1G0 + P2P1P0 C0

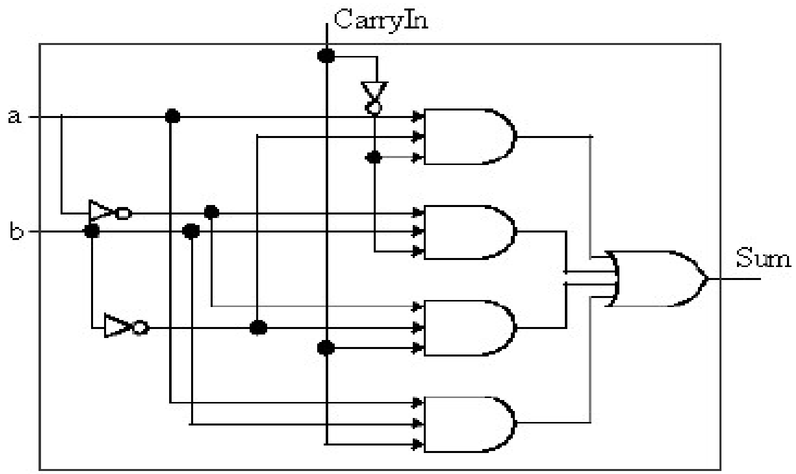
C4 = G3 + P3 C3 = G3 + P3G2 + P3P2G1
+ P3P2P1G0 + P3P2P1P0 C0

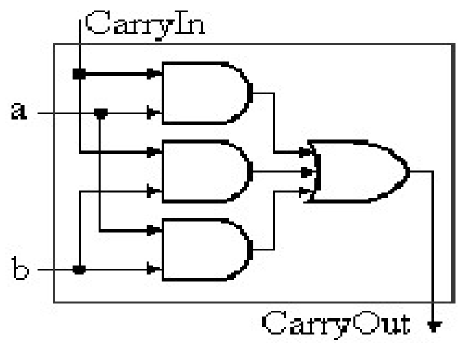
$$C_{i+1} = G_i + P_i\, G_{i-1} + P_i\, P_{i-1}\, c_{i-1}$$

16 bit Carry Lookahead Adder



Assessment





Carryout = (b.CarryIn)+(a.CarryIn) +(a.b)
Sum = (a.b’.CarryIn’)+ (a’.b.CarryIn’)+ (a’.b’.CarryIn)+ (a.b.CarryIn)


Thank You