SUBTRACTOR & RIPPLE CARRY ADDER & CARRY LOOKAHEAD ADDER

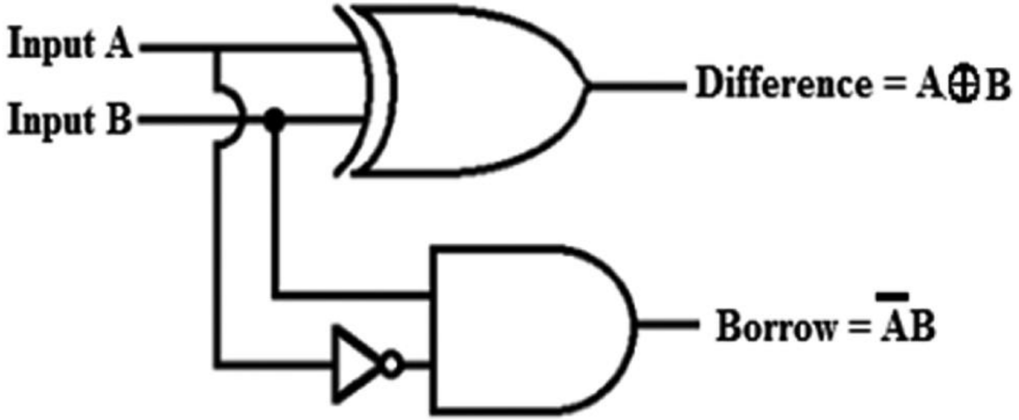
Presented by Nabanita Das

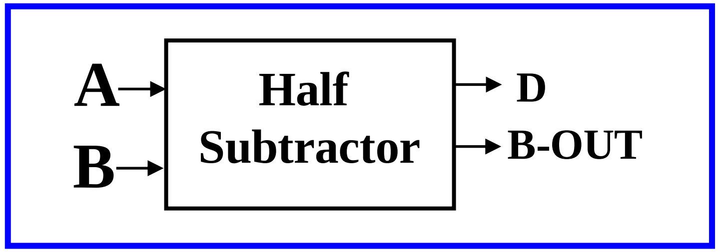
HALF SUBTRACTOR

- Subtracting a single-bit binary value Y from X (I.e. A -B) produces a difference bit D and a borrow out bit B-out.
- This operation is called half subtraction and the circuit to realize it is called a half subtractor.

Half Subtractor Truth Table:

Inputs		Outputs	
A	**B**	**D**	**B-out**
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0





D(A,B) = Σ (1,2)

D = A'B + AB'

D = A ⊕ B

B-out(A, B) = Σ (1)

B-out = A'B

FULL SUBTRACTOR

- Subtracting two single-bit binary values, Y, B-in from a single-bit value X produces a difference bit D and a borrow out B-out bit. This is called full subtraction.

Truth Table:

Inputs			Outputs	
X	**Y**	**B-in**	**D**	**B-out**
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

S(X,Y, B-in) = Σ (1,2,4,7)

B-out(x, y, B-in) = Σ (1,2,3,7)

Difference D

B-in \ XY	00	01	11	10
0	0	2 1	6	4 1
1	1 1	3	7 1	5

X, Y, B-in

D = X'Y'(B-in) + X'Y(B-in)' + XY'(B-in)' + XY(B-in)

D = X ⊕ Y ⊕ (B-in)

Borrow B-out

B-in \ XY	00	01	11	10
0	0	2 1	6	4
1	1 1	3 1	7 1	5

X, Y, B-in

B-out = X'Y + X'(B-in) + Y(B-in)

FULL SUBTRACTOR CIRCUIT

D = X ⊕ Y ⊕ (B-in)

B-out = X’Y + X’(B-in) + Y(B-in)

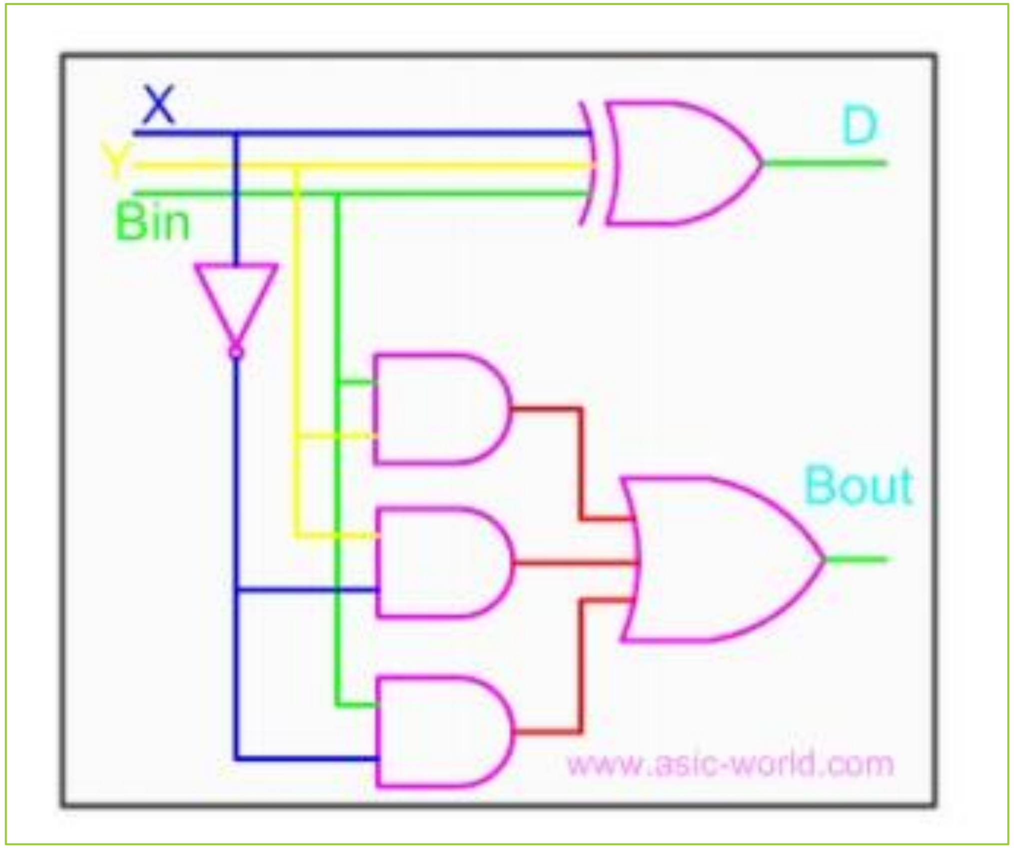
$$D = X'Y'Bin + X'YBin' + XY'Bin' + XYBin$$

$$= (X'Y' + XY)Bin + (X'Y + XY')Bin'$$

$$= (X \oplus Y)'Bin + (X \oplus Y)Bin'$$

$$= X \oplus Y \oplus Bin$$

$$Bout = X'.Y + X'.Bin + Y.Bin$$



RIPPLE CARRY ADDER

- An n-bit adder used to add two n-bit binary numbers can built by connecting in series n full adders.
  - Each full adder represents a bit position j (from 0 to n-1).
  - Each carry out C-out from a full adder at position j is connected to the carry in C-in of the full adder at the higher position j+1.
  - In parallel adder circuits, the carry output of one stage serves as the carry input of the succeeding stage, thus being called the ‘ripple’ carry adder.

Ripple’? Why called so?

What happens when we throw a stone in the water? The water around the stone-hit area flows ahead in the form of ripples. Similarly, in the Ripple Carry Adder, the Carry bit ‘ripples’ forward into the system.

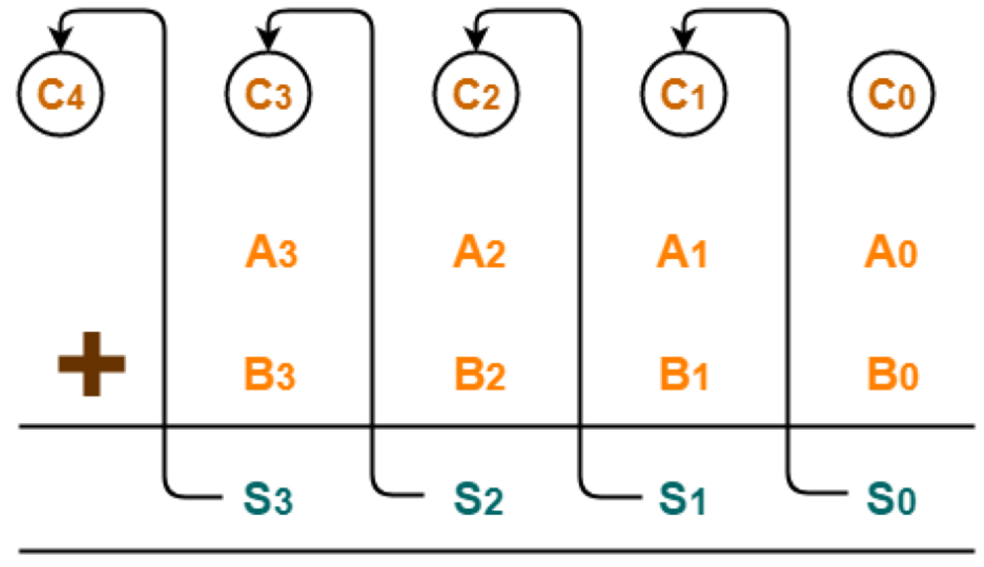
MATHEMATICALLY WHEN WE ADD TWO NUMBERS-

Consider two 4-bit binary numbers A3A2A1A0 and B3B2B1B0 are to be added.

Mathematically, the two numbers will be added as-

From here, we have-

C1 = C0



Adding two 4-bit Numbers

RIPPLE CARRY ADDER

In Ripple Carry Adder,

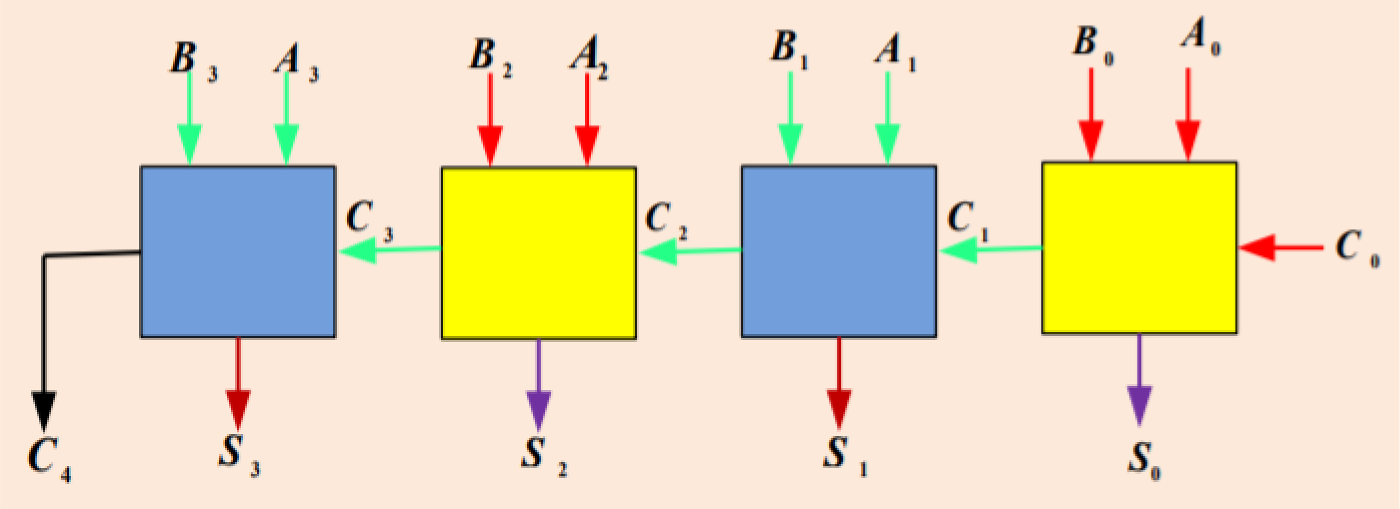
Each full adder has to wait for its carry-in from its previous stage full adder.

Thus, nth full adder has to wait until all (n-1) full adders have completed their operations.

This causes a delay and makes ripple carry adder extremely slow.

The situation becomes worst when the value of n becomes very large.

To overcome this disadvantage, Carry Look Ahead Adder comes into play.



Four-bit Adder (Ripple Carry Adder)

CARRY LOOKAHEAD ADDER

A look ahead carry adder is fast adder which improves speed by reducing the amount of time required to determine carry bits. It reduces the time which are delayed at each stage.

Let us consider a full adder. We have the inputs signals A, B, and Cin. If we consider the addition of these three variables in every possible case, we get a truth table like the one below.

INPUT OUTPUT						
Row	**A**	**B**	**Cin**	Sum	**Cout**	
0	0	0	0	0	0	No carry generation C_{out} =0
1	0	0	1	1	0	
2	0	1	0	1	0	Carry propagation $C_{out} = C_{in}$
3	0	1	1	0	1	
4	1	0	0	1	0	
5	1	0	1	0	1	
6	1	1	0	0	1	Carry generation C_{out} = 1
7	1	1	1	1	1	

EXPRESSION FOR CARRY GENERATION & PROPAGATION

From truth table , carry generation in row 6th and 7th is given by:-

$$G_i = A_iB_i$$

Similarly the carry propagation Pi occur with either Ai= 1 and Bi= 0 or vice versa.

$$P_i = A_i \oplus B_i$$

Gi is known as the carry Generate signal

Pi is known as the carry propagate signal

The full adder circuit we will give two outputs named as Sum, Carry out.

$$S = A \oplus B \oplus (C\text{-in})$$

$$C\text{-out} = (A \oplus B)(Cin) + AB$$

The new expressions for the output sum and the carryout are given by:-

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_iC_i$$

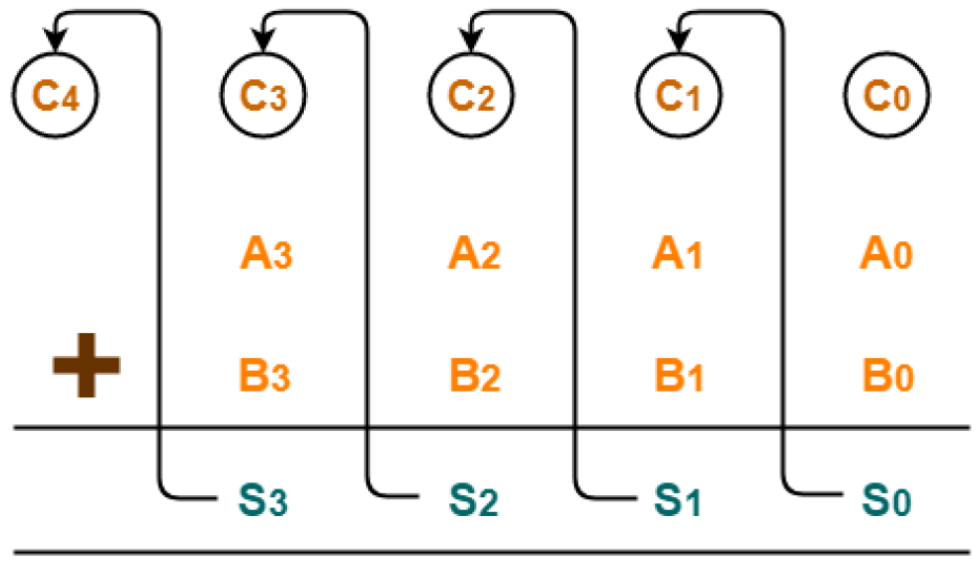
MATHEMATICALLY WHEN WE ADD TWO NUMBERS-

Consider two 4-bit binary numbers A3A2A1A0 and B3B2B1B0 are to be added.

Mathematically, the two numbers will be added as-

From here, we have-

C1 = C0



Adding two 4-bit Numbers

CONTD.

Sum = Pi ⊕ Ci

Carry = Gi + Pi . Ci

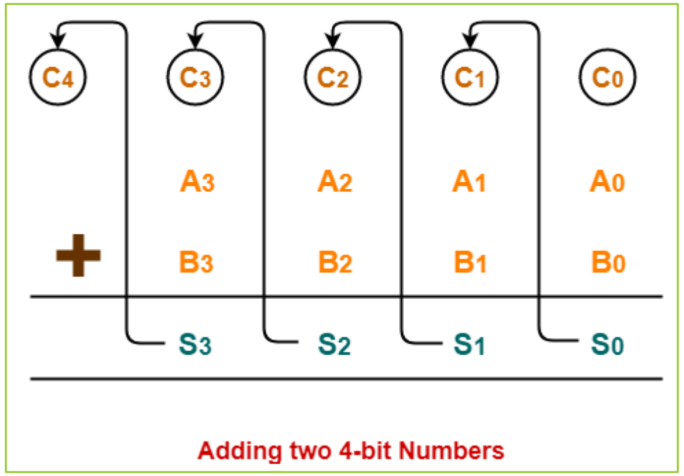
We can calculate the output carry C1, C2, C3, and C4 using the above derived equations as:

$C_1 = G_0 + P_0C_0$

$C_2 = G_1 + P_1C_1 = G_1 + P_1 (G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0$

$C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$

$C_4 = G_3 + P_3C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$



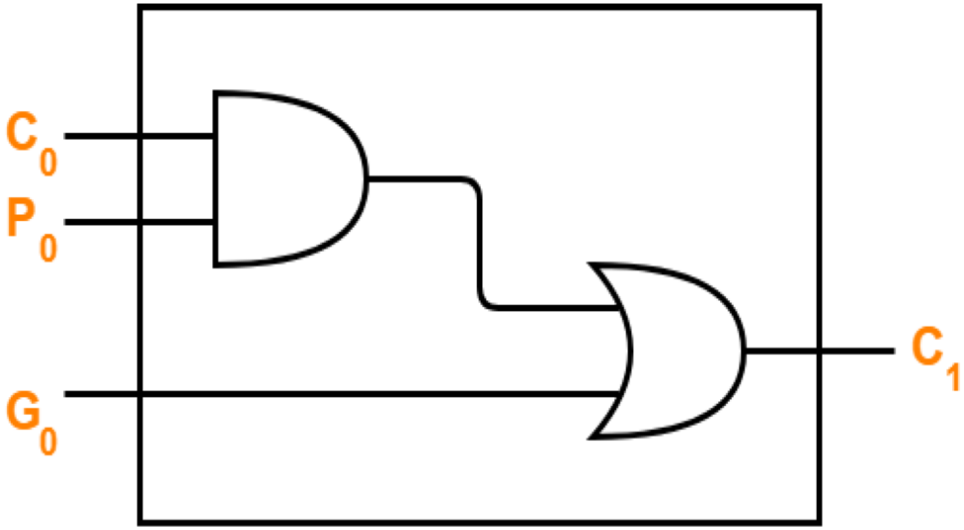
IMPLEMENTATION OF CARRY GENERATOR CIRCUITS-

Implementation Of C1–

The carry generator circuit for C1 is implemented as shown below.

It requires 1 AND gate and 1 OR gate.

C1 = C0P0 + G0

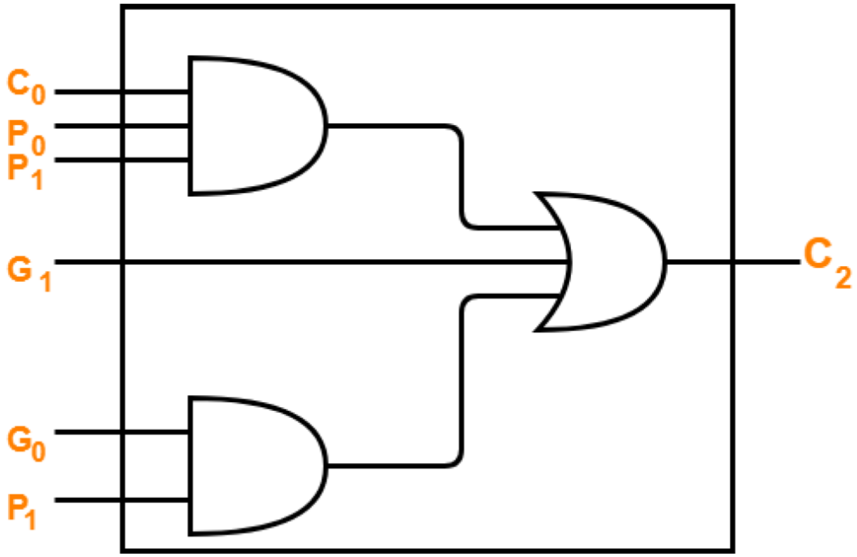


Similarly, implementation Of C2–

The carry generator circuit for C2 is implemented as shown below.

It requires 2 AND gates and 1 OR gate.

C2 = C0P0P1 + G0P1 + G1



4 BIT CARRY LOOKAHEAD ADDER CIRCUIT