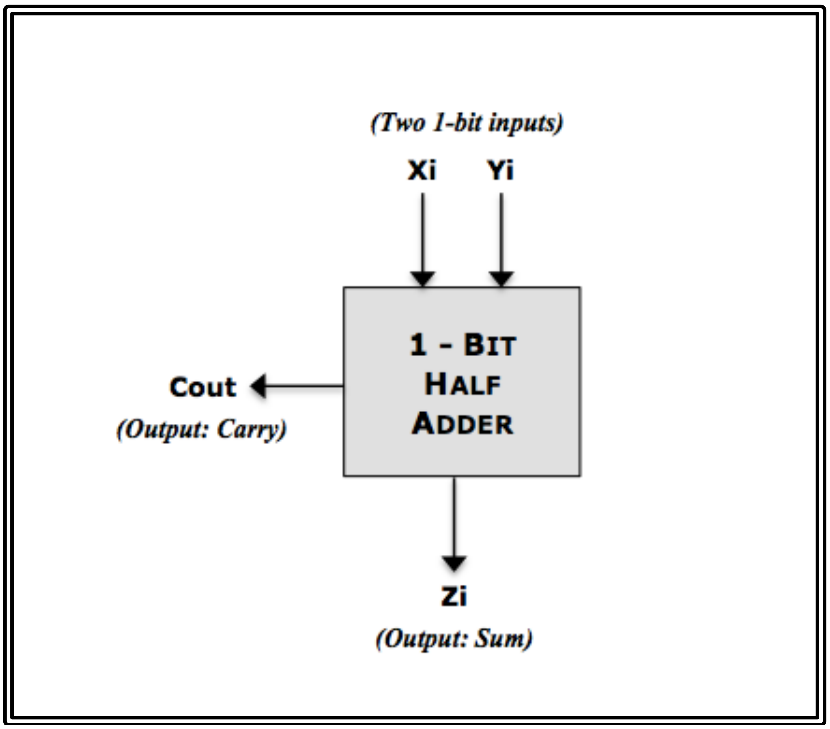
INTEGER DATA COMPUTATION

ONE BIT ADDITION: HALF ADDER

1) It is a simple 1-bit adder circuit.
2) It adds two 1-bit inputs Xi & Yi and produces a sum Zi and a Carry Cout.
3) As it does not consider any carry input, it can't be combined to add large numbers.
4) Hence it is called a Half Adder.



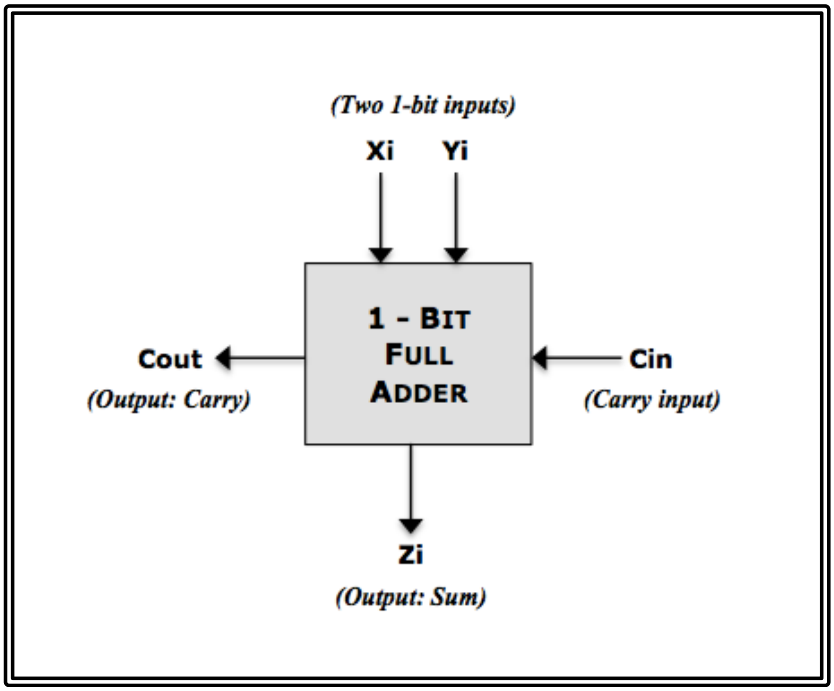
Inputs bits: Xi and Yi.

Output (Sum): Zi
Output (Carry): Cout

Formula:
Sum (Zi) = Xi Ex-Or Yi.
Carry (Cout) = Xi.Yi

ONE BIT ADDITION: FULL ADDER

1) It is a **1-bit adder** circuit.
2) It adds **two 1-bit inputs Xi & Yi**, along with a **Carry Input Cin**.
3) It produces a **sum Zi** and a Carry output **Cout**.
4) As it considers a carry input, it can be used in **combination to add large numbers**.
5) Hence it is called a **Full Adder**.



Inputs bits: Xi and Yi.
Input Carry: Cin

Output (Sum): Zi
Output (Carry): Cout

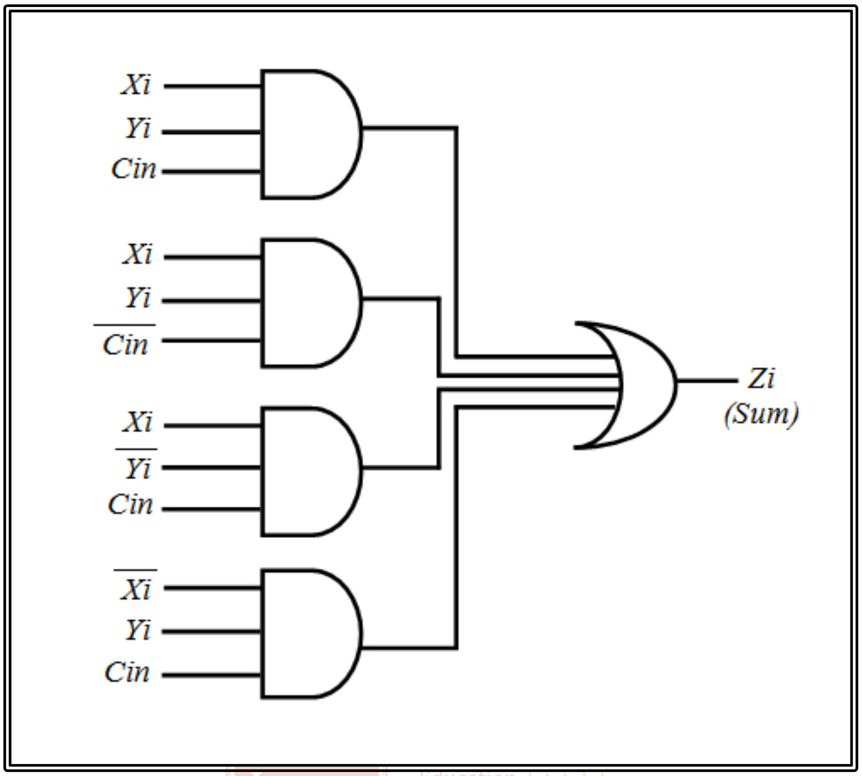
Formula for Sum (Zi)

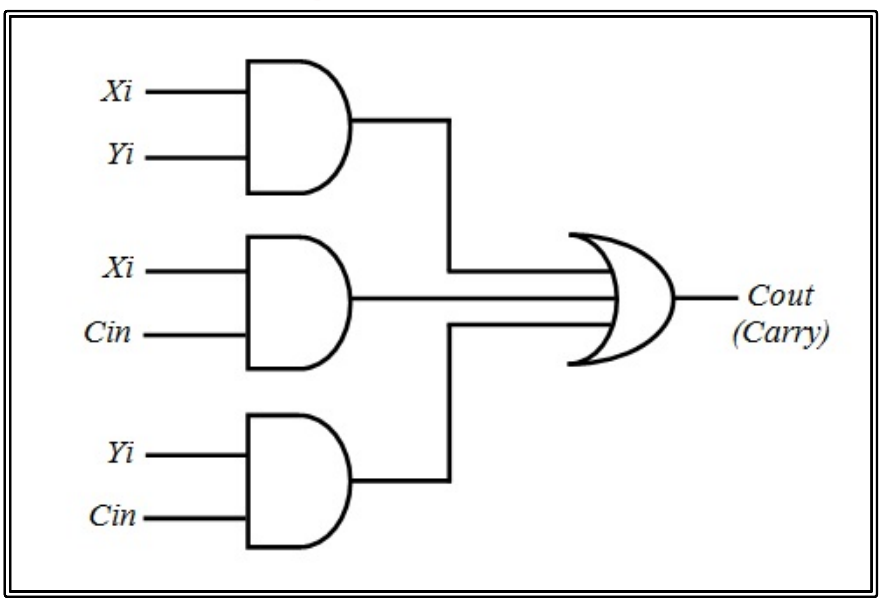
$$Zi = Xi \oplus Yi \oplus Cin$$
$$\therefore Zi = Xi \cdot Yi \cdot Cin + Xi \cdot Yi \cdot \overline{Cin} + Xi \cdot \overline{Yi} \cdot Cin + \overline{Xi} \cdot Yi \cdot Cin$$

Formula for Carry (Cout)

$$Cout = Xi \cdot Yi + Xi \cdot Cin + Yi \cdot Cin$$

CIRCUIT FOR A FULL ADDER





Multiple Bit Addition: Serial Adder / Ripple Carry Adder

1) A Full Adder can add two "1-bit" numbers with a Carry input.
2) It produces a "1-bit" Sum and a Carry output.
3) **Combining many of these Full Adders, we can add multiple bits**.
4) One such method is called Serial Adder.
5) Here, bits are **added one-by-one from LSB onwards**.
6) The **Carry of each stage is propagated (Rippled) into the next stage**.
7) Hence, these adders are also called **Ripple Carry Adders**.
8) Advantage: They are very **easy to construct**.
9) Drawback: As addition happens **bit-by-bit**, they are **slow**.
10) **Number of cycles** needed for the addition is equal to the **number of bits to be added**.

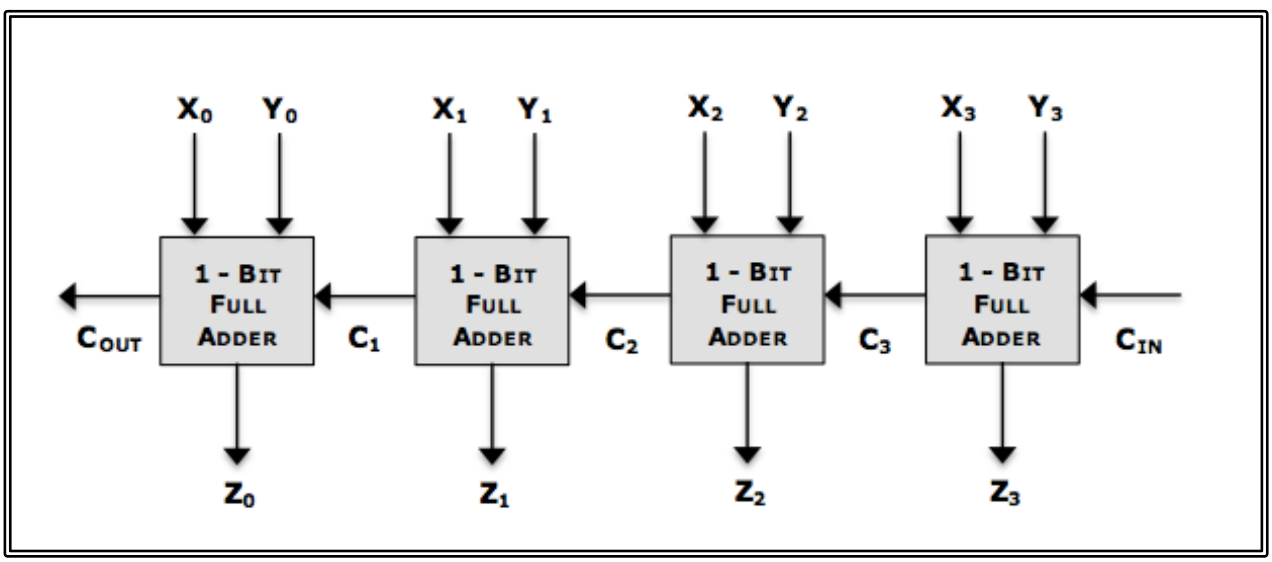
Inputs:
Assume X and Y are two "4-bit" numbers to be added, along with a Carry input C_{IN}.

$X = X_0\ X_1\ X_2\ X_3$ (X_0 is the MSB … X_3 is the LSB)
$Y = Y_0\ Y_1\ Y_2\ Y_3$ (Y_0 is the MSB … Y_3 is the LSB)
C_{IN} = Carry Input

Outputs:
Assume Z to be a "4-bit" output, and C_{OUT} to be the output Carry

$Z = Z_0\ Z_1\ Z_2\ Z_3$ (Z_0 is the MSB … Z_3 is the LSB)
C_{OUT} = Carry Output

Circuit for 4-bit Serial Adder/ Ripple Carry Adder



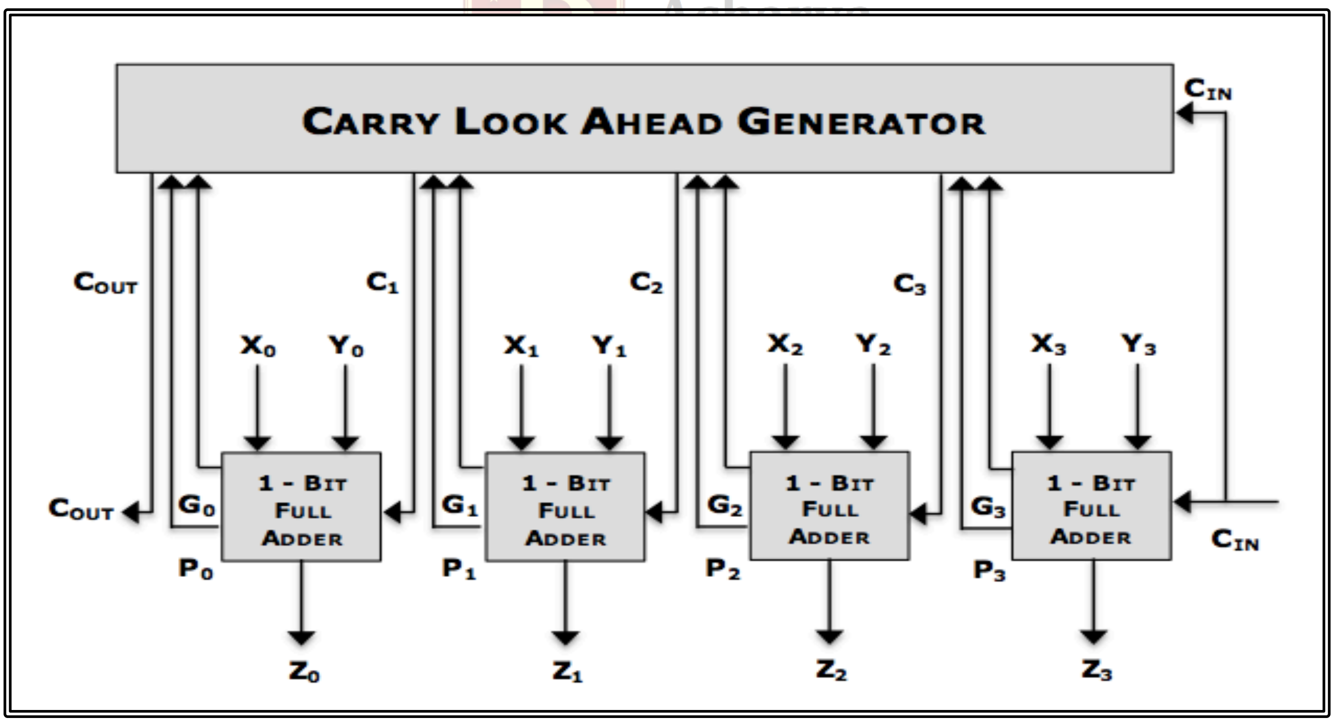
MULTIPLE BIT ADDITION: CARRY LOOK AHEAD ADDER / PARALLEL ADDER

1) It is used to add multiple bits **simultaneously**.
2) While adding multiple bits, the main issue is that of the **intermediate carries**.
3) In Serial Adders, we therefore added the bits one-by-one.
4) This allowed the carry at any stage to propagate to the next stage.
5) But this also made the process **very slow**.
6) If we "**PREDICT**" the **intermediate carries**, then all bits can be added **simultaneously**.
7) This is done by the **Carry Look Ahead Generator** Circuit.
8) Once all carries are determined beforehand, then all bits can be **added simultaneously**.
9) Advantage: This makes the addition process **extremely fast**.
10) Drawback: Circuit is **complex**.

Inputs:
Assume X and Y are two "4-bit" numbers to be added, along with a Carry input C_{IN}.
X = X_0 X_1 X_2 X_3 (X_0 is the MSB … X_3 is the LSB); Y = Y_0 Y_1 Y_2 Y_3 & C_{IN} = Carry Input

Outputs:
Assume Z to be a "4-bit" output, and C_{OUT} to be the output Carry
Z = Z_0 Z_1 Z_2 Z_3 & C_{OUT} = Carry Output

Circuit for 4-bit Serial Adder/ Ripple Carry Adder



CALCULATIONS

We can "Predict" (Look Ahead) all the intermediate carries in the following manner.

The Carry at any stage can be calculated as:

$C_i = X_i.Y_i + X_i.C_{in} + Y_i.C_{in}$
$C_i = X_i.Y_i + C_{in}(X_i + Y_i)$

$$\boxed{C_i = G_i + P_i.C_{IN}}$$

Here Gi = $X_i.Y_i$ … (Generate)
And Pi = X_i+Y_i … (Propagate)

We need to predict the Carries: C_3, C_2, C_1 and C_0

$$C_3 = G_3 + P_3C_{IN} \quad \text{… I}$$

$C_2 = G_2 + P_2C_3$

Substituting the value of C3, we get:

$$C_2 = G_2 + P_2G_3 + P_2P_3C_{IN} \quad \text{… II}$$

$C_1 = G_1 + P_1C_2$

Substituting the value of C2, we get:

$$C_1 = G_1 + P_1G_2 + P_1P_2G_3 + P_1P_2P_3C_{IN} \quad \text{… III}$$

$C_0 = G_0 + P_0C_1$

Substituting the value of C1, we get:

$$C_0 = G_0 + P_0G_1 + P_0P_1G_2 + P_0P_1P_2G_3 + P_0P_1P_2P_3C_{IN} \quad \text{… IV}$$

From the above four equations, it is clear that the values of all the four Carries (C_3, C_2, C_1, C_0) can be determined beforehand even without doing the respective additions. To do this we need the values of all G's ($X_i.Y_i$) and all P's (X_i+Y_i) and the original carry input C_{IN}. This is done by the Carry Look Ahead Generator Circuit.

ADDER / SUBTRACTOR CIRCUIT:

1) **Subtraction** in binary numbers is simply performed by **addition of two's complement**.
2) That means, a special circuit for subtraction is not needed.
3) The same circuit that is used for Addition, can also be used for subtraction.
4) The following circuit is called **Adder/ Subtractor** circuit.
5) It can perform Addition as Z = X + Y.
6) It can also perform subtraction as Z = X + (2's Complement of Y)
7) The Variable "S" determines if Addition or Subtraction will be performed.
8) **If S = 0, then Addition will be performed**.
9) **If S = 1, then Subtraction will be performed**.
10) If S = 1, then the operation is **Z = X + (1's Complement of Y) + 1**. Hence Z = X – Y.