Chapter 03: Computer Arithmetic

Lesson 03:

Arithmetic Operations–

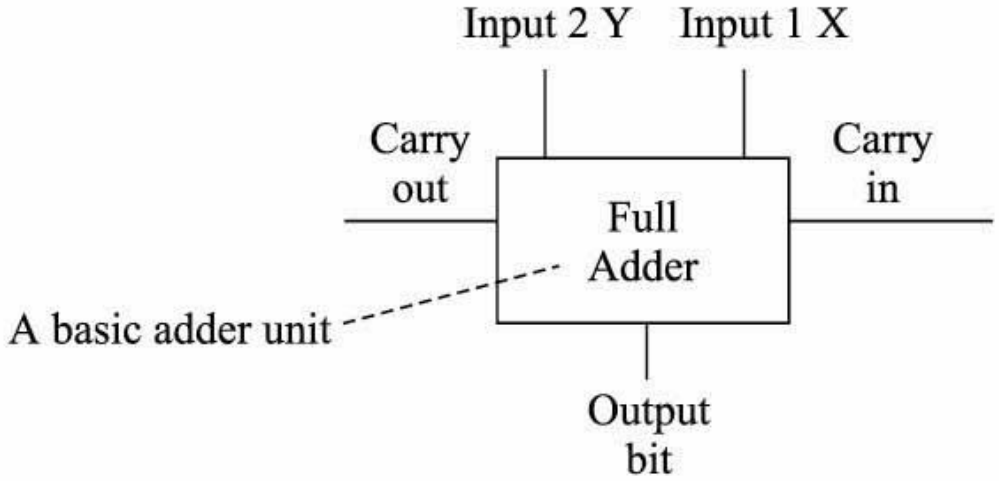
Adder and Subtractor circuits Design

Objective

- To understand adder circuit
- Subtractor circuit
- Fast adder circuit

Adder Circuit

Full Adder Circuit



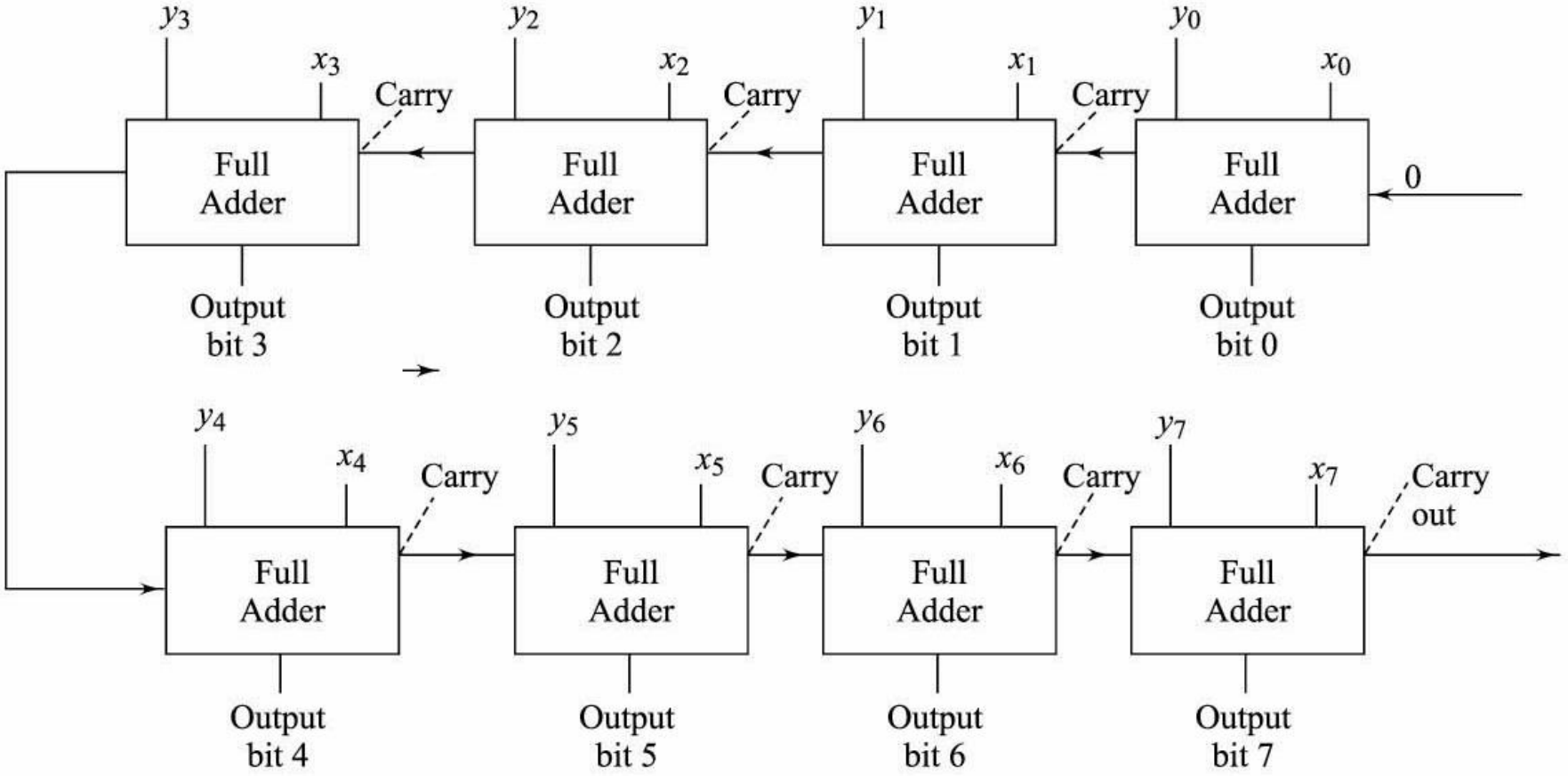
8-bit of X = $x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0$

8-bit of Y = $y_7, y_6, y_5, y_4, y_3, y_2, y_1, y_0$

Ripple carry propagation addition method using full adder circuit

- Compute $x_i + y_i + C_{i-1}$
- Each bit output z_i based on three bits— the bits at the inputs and carry-in
- Carry-in— Generated by the next-lower bit of computation

Ripple carry propagation in 8-bit adder circuit based on 8 full adders



Speed of Adder

- Speed of the circuit determined by time it takes for the carry signals to propagate through all the full adders
- Each full adder can't perform its part of the computation until all the full adders to the right of it have completed their parts
- The computation time grows linearly with the number of bits in the inputs

Subtractor Circuit

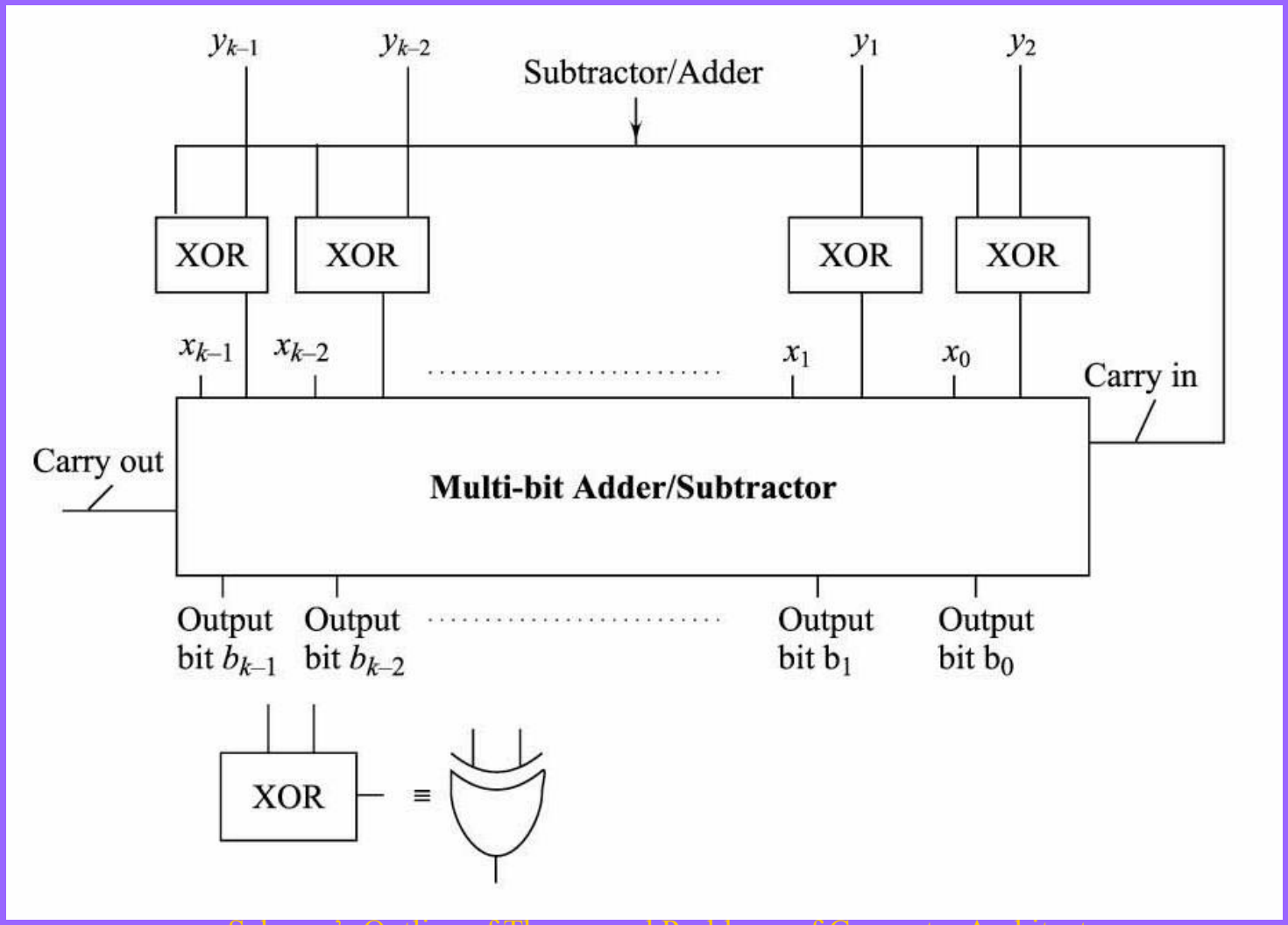
Subtraction

- Handled by similar methods, using adder modules that compute the difference between *two* numbers by adding the *X* and two's complement of *Y*
- An adder-based adder-cum-subtractor circuit based on an adder circuit
- Uses additional circuit of XOR gates

XOR gate in adder-cum-subtractor

- Finds the 1's complement when subtraction of *Y* is to be done
- Does nothing when addition of *Y* is to be done used in the adder circuit

k-bit adder/subtractor Using Two's complement converter circuit with Y



Fast Adders (High Speed Adders) and Carry Look Ahead Addition

Design of Fast Adders (High Speed Adders) and Carry Look Ahead Addition

- Ripple carry adder computation time grows linearly with the number of bits in the inputs
- Use of binary logic cells
- Output stage sum bit = s_i = x_i.XOR. y_i . XOR. C_i
- Output stage carry bit for input to (i + 1)th stage
- = C_i+1 = (x_i. AND. y_i). OR. (x_i. AND. C_i). OR . (y_i. AND. Ci)
- = C_i+1 = (x_i.AND. y_i). OR. [(x_i.OR. y_i.).AND. C_i)]

Design of Fast Adders (High Speed Adders) and Carry Look Ahead Addition

- Output stage sum bit = s_i = x_i.XOR. y_i . XOR. C_i
- Output stage carry bit for input to (i + 1)th stage

= C_i+1 = (x_i. AND. y_i). OR. (x_i. AND. C_i). OR . (y_i. AND. Ci)

= C_i+1 = (x_i.AND. y_i). OR. [(x_i.OR. y_i.).AND. C_i)]

- Ripple carry adder computation time grows linearly with the number of bits in the inputs

Design of Fast Adders (High Speed Adders) and Carry Look Ahead Addition

- Use of binary logic cells
- Use k-full adders 0th to (k – 1)th for k-bit addition
- An i-th stage adder circuit two logic terms─ Generating (using x bit and y bit) first term of expression
- Generating component─ $g_i = x_i$. AND. y_i

Propagating component

- Propagating component (us carry, x and y) second term of expression
- $p_i = x_i$ OR y_i,
- An output stage carry, $C_{i+1} = g_i + pi.C_i$
- C_i depends on C_{i-1}
- C_{i-1} depends on C_{i-2}, and so on
- C_{i-1} is $g_{i-1} + p_{i-1}.C_{i-1}$

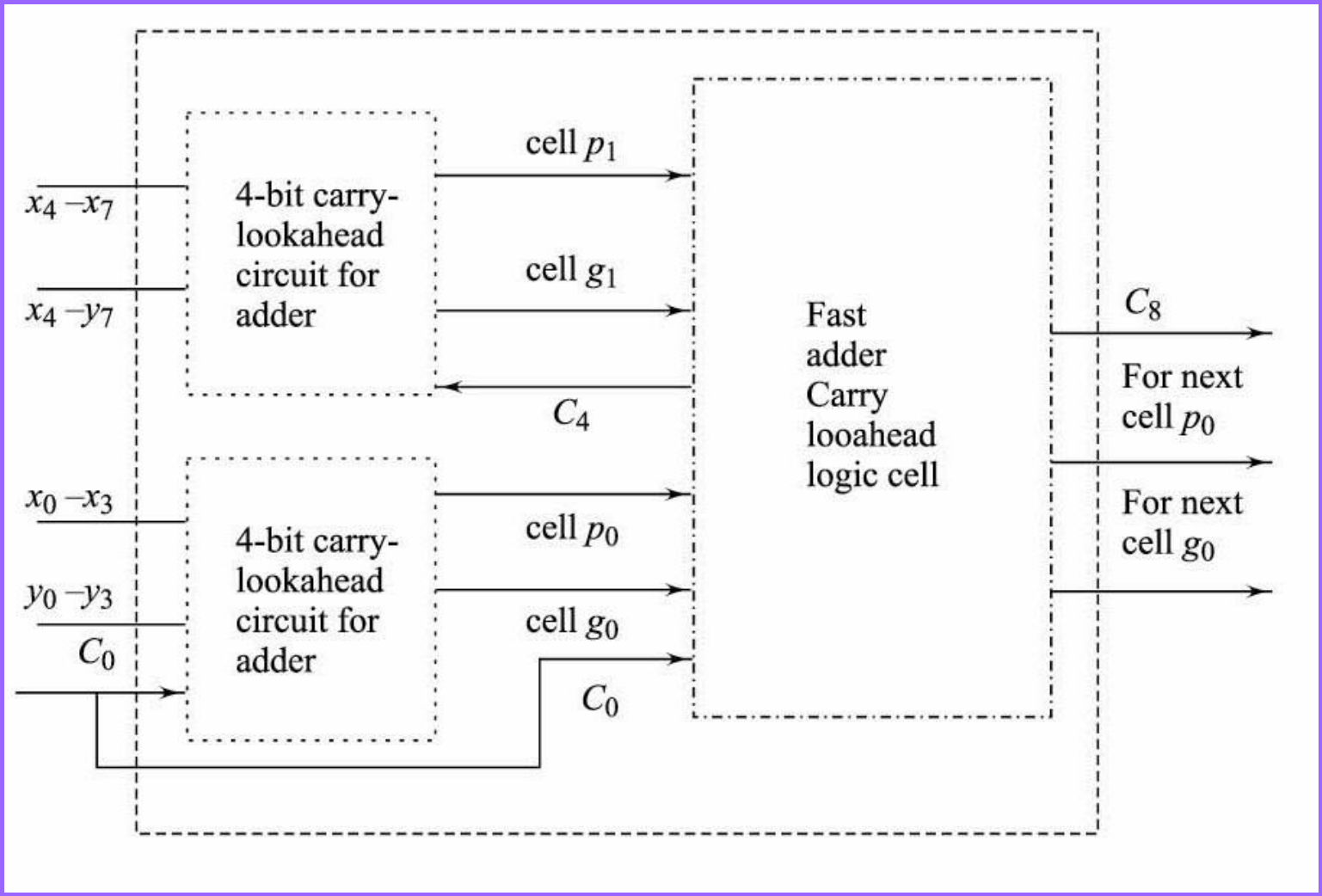
Generating and propoagating components and Carry in CGPS cell

- $g_i = x_i$. AND. y_i and $p_i = x_i$ OR y_i,
- $C_{i+1} = g_i + p_i . g_{i-1} + p_i . p_{i-1} . g_{i-2}$

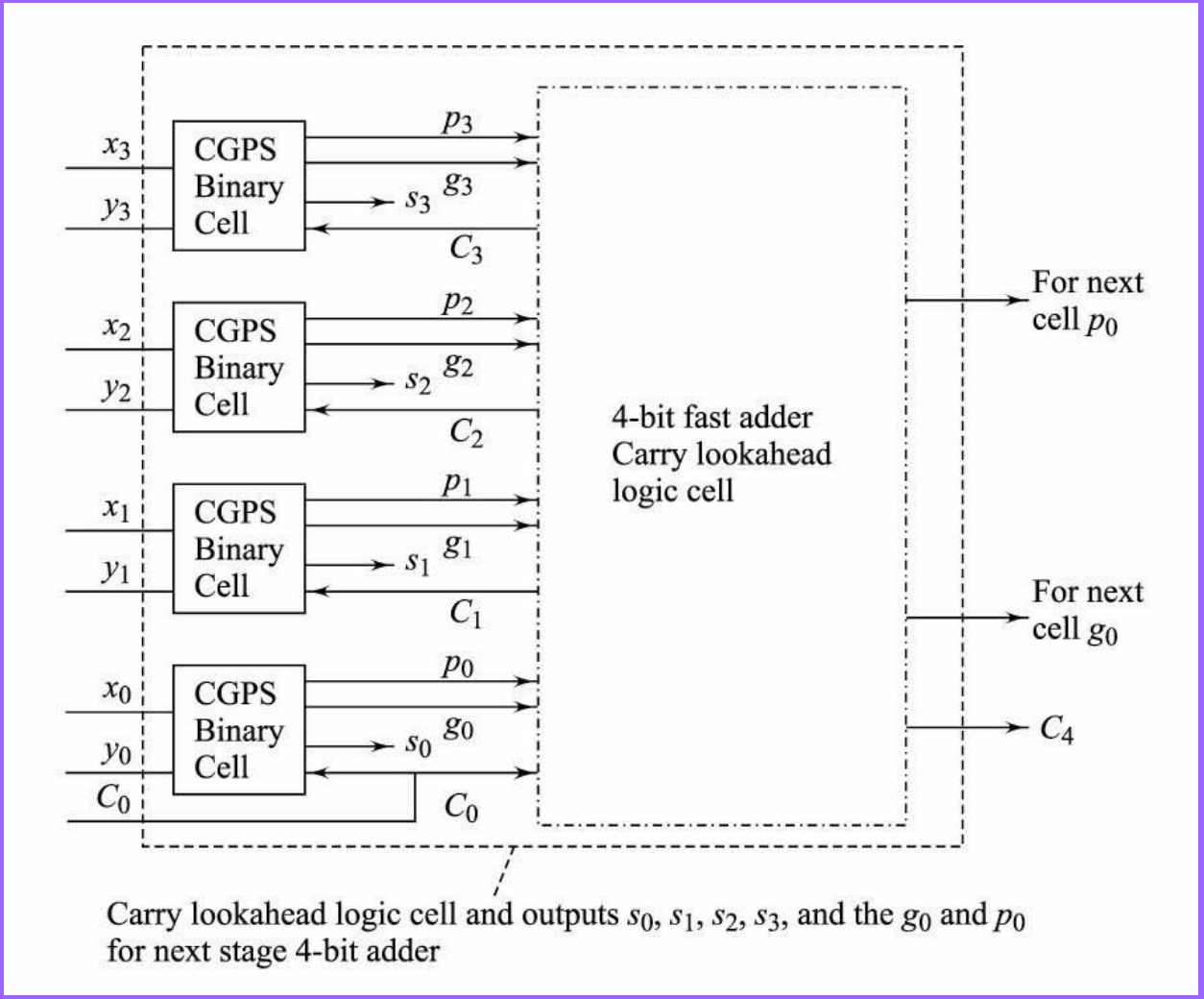
$+ p_i . p_{i-1} . p_{i-2} . g_{i-3} + \ldots . + p_i . p_{i-1} . p_{i-2} . \ldots p_0 C_0$

- C_4 can be computed ahead using a binary logic cell having p_is and g_is in the inputs
- C_4 input to next binary logic cell

Using Binary Logic Cell outputs g_0, p_0, p_1, g_1, …



Use of 4-binary logic cells



Summary

We Learnt

- Full adder— Each bit output z_i based on three bits— the bits at the inputs and carry-in
- Ripple carry adder computation time grows linearly with the number of bits in the inputs
- Adder cum subtractor using XORs for two's complement generation
- Fast addition by carry-look-ahead computations

End of Lesson 3 on
Arithmetic Operations–
Adder and Subtractor circuits Design