BASIC ARITHMETIC OPERATIONS

Object: To design and implement logic circuit for basic arithmetic operations.

Theory:

An important part of the central processor of any computer is the arithmetic unit in which binary addition, subtraction, division and multiplication are carried out.

Subtraction however can be performed by adding complemented numbers. Multiplication can also be performed by repeated addition .Division can be also achieved by repeated subtraction. This means that the adder is the centre piece of the arithmetic unit. There are two types of the addition:

1. ***Half - Adder (H.A) :***

It is a device that adds two bits of binary data. In other words, the half adder performs the operation s:

$0 + 0 = 0$

$0 + 1 = 1$ ……. (4.1)

$1 + 0 = 1$

$1 + 1 = 0$, Carry = 1

The last operation is, of course, 1+1=0, which is 0 with a carry 1 to the next bit position. Equation (4.1) may be expressed n the form of a truth table as shown in table (4.1)

Input		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

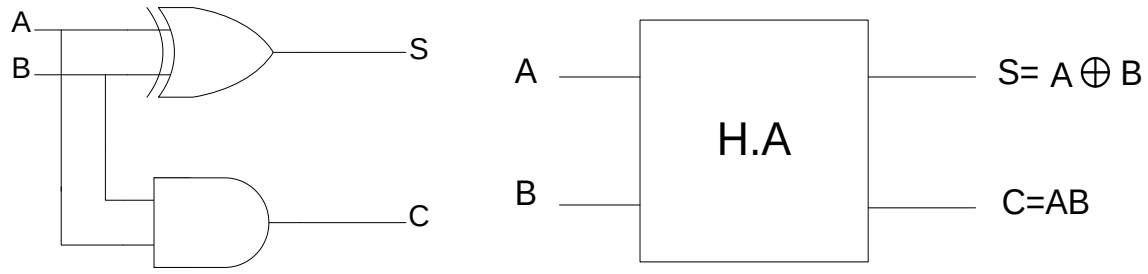
From the truth table we see that

$$S = A\bar{B} + \bar{A}B \qquad \text{…….. (4.2)}$$

$$= A \oplus B$$

and $$C = A.B \qquad \text{…….. (4.3)}$$

So the H.A adds only two bits at a time, so that it cannot be used to add two bits and a carry bit from a previous step, as is generally required in adding tow binary numbers the symbol for the H.A is given in Fig (5.1.b).



(a) (b)

Fig. (4.1) (a) Half Adder circuit diagram

(b) Half Adder block diagram

2. ***Full- Adder (F.A):***

A half adder is not very useful on its own, and a third input is often required for carries. Adding numbers that have two bits or more requires a full adder (F.A) which is capable of the previous order. The symbol of full-adder is shown in Fig. (4.2).

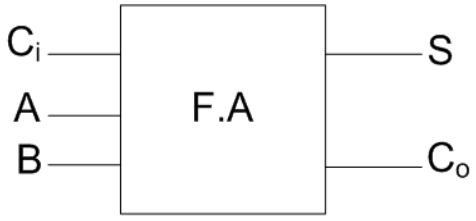


Fig. (4.2) Full –adder block diagram

Where:

C_i: carry- in from the previous addition.

C_o: carry- out to the next addition.

The truth table for a full-adder is (F.A) is determined by the 8 possible combinations of the inputs A, B and C_i, the corresponding values of S and C_o is given in table (4.2) from which we may write

$$S = \overline{A} \cdot \overline{B} \cdot C_i + \overline{A} \cdot B \cdot \overline{C_i} + A \cdot \overline{B} \cdot \overline{C_i} + A \cdot B \cdot C_i \quad \ldots\ldots\ldots \quad (4.4.a)$$

$$= \left(\overline{A} \cdot \overline{B} + A \cdot B\right) \cdot C_i + (\bar{A} \cdot B + A \cdot \bar{B}) \cdot \overline{C_i} \quad \ldots\ldots\ldots \quad (4.4.b)$$

$$C_o = \overline{A} \cdot B \cdot C_i + A \cdot \bar{B} \cdot C_i + A \cdot B \cdot \overline{C_i} + A \cdot B \cdot C_i \quad \ldots\ldots\ldots \quad (4.5.a)$$

$$= A \cdot B + C_i(A \oplus B) \quad \ldots\ldots\ldots \quad (4.5.b)$$

Input	Output	
A B C_i	S	C_o
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

Procedure:

A. Half-Adder (H.A)

1. Implement a H.A logic equation for sum and carry using NAND gates only then verify the truth table.
2. Design a Half-Subtractor (H.S) network, and verify its truth table.

B. Full-Adder (F.A):

1. Verify the truth table of F.A by means of using NAND gates only.
2. Design a Full-Subtractor (F.S) network, and verify its truth table.

Discussion:

1. By means of H.A block diagram build a F.A.
2. By means of H.S block diagram build a F.S.
3. Build a H.A using NOR gates only.
4. Use only two 2-inlput EX-OR gates and three 2-input NAND gates to build F.A.
5. Use the block diagrams of F.A to show the addition process of the binary numbers110&111.
6. By means of F.A block diagram, EX-OR gates and external switch x, design a 4-bit adder/subtractor.
7. What is meant by Parallel binary adders? For the parallel adder shown in Fig (4.3), determine the sum by analysis of the logical operation of the circuit.

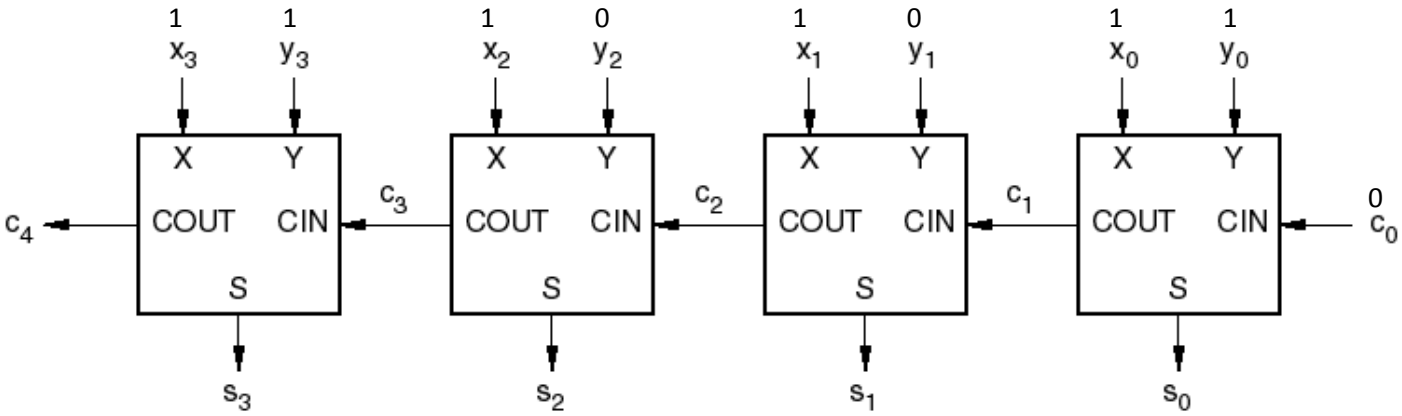


Fig. (4.3)