Arithmetic Circuits Design and Implementation

Introduction:

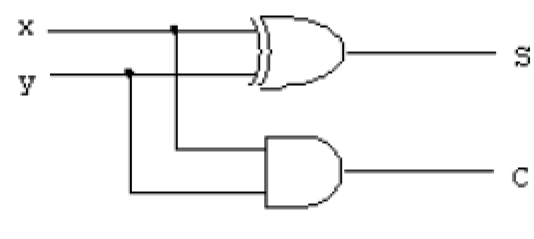
Addition is just what you would expect in computers. Digits are added bit by bit from right to left, with carries passed to the next digit to the left, just as you would do by hand. Subtraction uses addition: the appropriate operand is simply negated before being added.

Objectives:

- To understand the concept of Half and Full Adders.
- Design and build Ripple Carry Adder .
- Introduce 4-bit magnitude comparator.
- Design and implement binary multiplier

Half Adder:

Half adder is a combinational circuit that adds only two one bit numbers ,Since there are two inputs (x and y), only four possible combinations of inputs can applied . These four possibilities, and the resulting sums are shown in following truth table.



Logic Diagram of the Half-Adder

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table of the Half-Adder

Figure(1)

$$S = x \oplus y = x'y + xy'$$

$$C = xy$$

Full Adder:

Full adder is a combinational circuit that adds three bits and generates a sum and carry.

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Truth Table of the Full-Adder

.

From the truth table, we can obtain the Boolean expression of C & S outputs as follows :

$$S = x'y'z + x'yz' + xy'z' + xyz$$

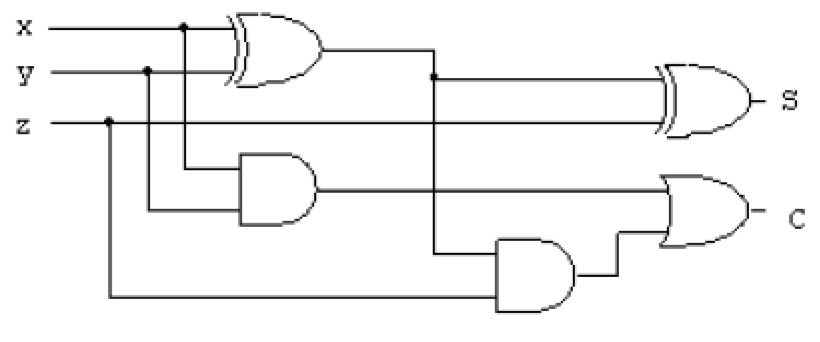
$$C = x'yz + xy'z + xyz' + xyz$$

Using Map-simplification method, we can get the simplified forms as follows :

$$S = x \oplus y \oplus z$$

$$C = xy + yz + xz$$

Now, we can construct the full-adder circuit based on the simplified Boolean expression of S and C outputs



Logic Diagram for the Full-Adder

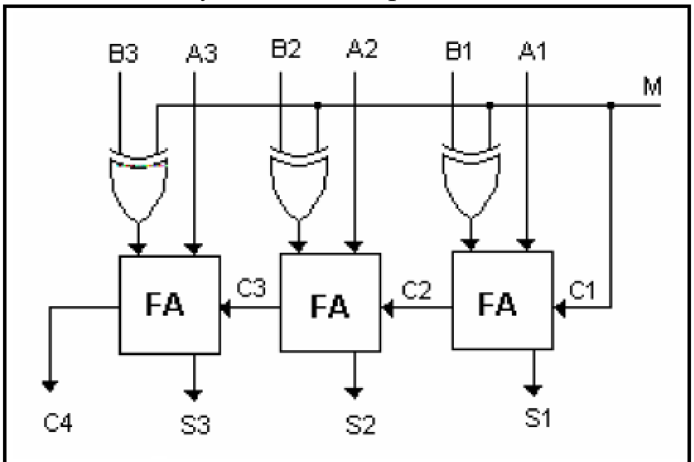
Figure(2)

Ripple Adder

Two binary numbers, each of n bits, can be added using a ripple adder, a cascade of n full adders; each full adder handles one bit. Each Cout of a full adder is connected to the Cin of the higher full adder. The Cin of the least significant full adder is set to 0.

Adder-Subtractor circuit

The subtraction of two binary numbers can be done by taking the 2's complement of the subtrahend and adding it to the minued. The 2's complement can be obtained by taking the 1's complement and adding 1. To perform A - B, we complement the four bits of B, add them to the four bits of A, and add 1 to the input carry.
We may use XOR gate as an inverter if placing a logic "1" at one of the inputs. This helps in getting the 1's complement of the subtrahend; then we add "1" to get the 2's complement; which in turn is added to the minued to get the final result of the subtraction.
Figure below shows adder-subtractor circuit; the mode input M controls the operation; when M=0, the circuit is an adder. When M=1, the circuit becomes a subtractor. This circuit can be cascaded for any number of inputs.



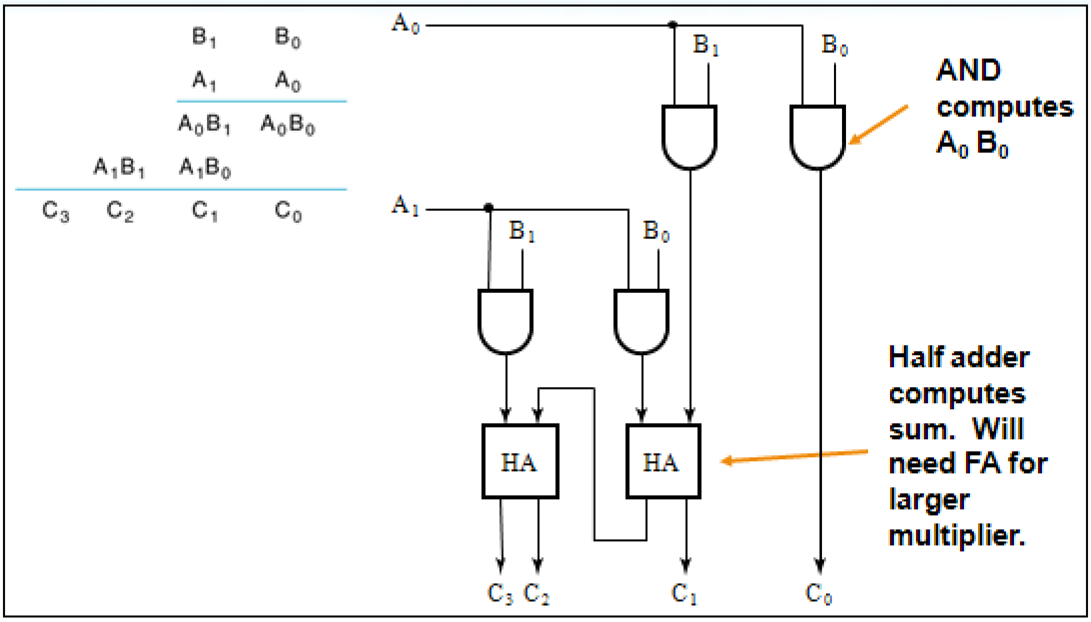
Logic Diagram of 3-bit Adder-Subtractor Circuit

Figure(3)

Multiplier:

If we want to multiply tow numbers(A,B) ,each of them is consist of two bit as follows:
B = {B1 B0},
A = {A1 A0}
Then we multiply by doing single-bit multiplications and shifts.

		B_1	B_0
		A_1	A_0
		A_0B_1	A_0B_0
	A_1B_1	A_1B_0	
C_3	C_2	C_1	C_0



Figure(4)

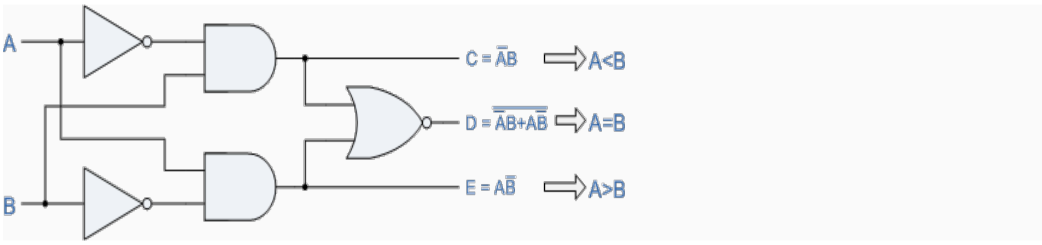
Comparator:

Another common and very useful combinational logic circuit is that of the **Digital Comparator** circuit. Digital or Binary Comparators are made up from standard AND, NOR and NOT gates that compare the digital signals at their input terminals and produces an output depending upon the condition of the inputs. For example, whether input A is greater than, smaller than or equal to input B etc.

Digital Comparators can compare a variable or unknown number for example A (A1, A2, A3, ... An, etc) against that of a constant or known value such as B (B1, B2, B3, ... Bn, etc) and produce an output depending upon the result. For example, a comparator of 1-bit, (A and B) would produce the following three output conditions.

$$A > B, \quad A = B, \quad A < B$$

1-bit Comparator



Then the operation of a 1-bit digital comparator is given in the following Truth Table.

Truth Table

Inputs		Outputs		
B	A	A > B	A = B	A < B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0