Module-4

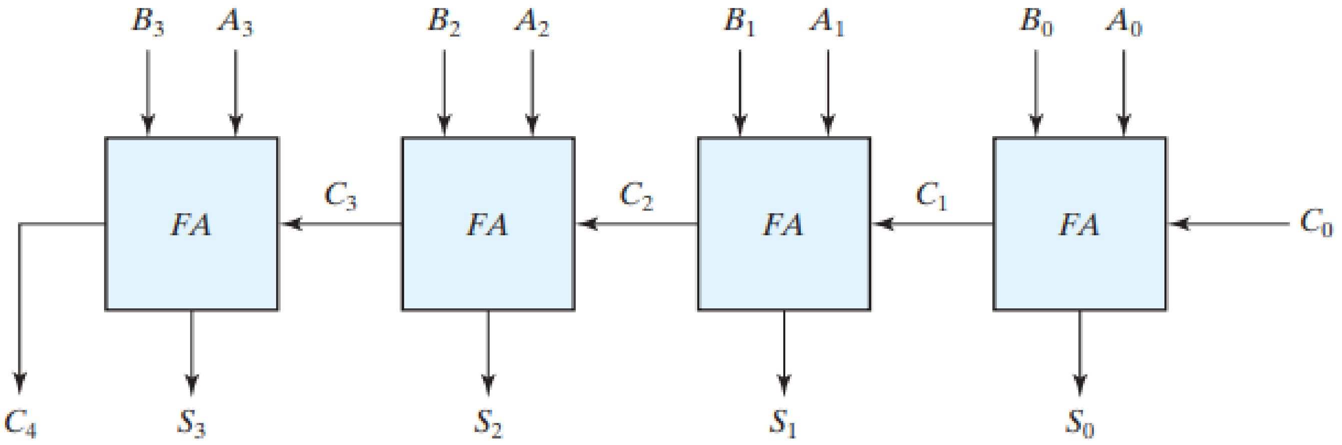
Design of data path circuits

N-bit Parallel Adder/Subtractor, Carry Look Ahead Adder, Unsigned Array Multiplier, Booth Multiplier, 4-Bit Magnitude comparator. Modeling of data path circuits using Verilog HDL

Parallel binary adder(4-Bit binary ripple carry adder)

Let augend bit of A be A3,A2,A1,A0 and the addend bit of B be B3,B2,B1,B0. Let the output are sum and carry.The sum bit are S3,S2,S1,S0.

Consider the binary number A=1001 and B=1101 Here C0=0.The four bit adder perform the addition as follows.



Carry C		1	0	0	1	0
Augend A			1	0	0	1
Addend B			1	1	0	1
Sum	S		0	1	1	0
			↓	↓	↓	↓
			S3	S2	S1	S0

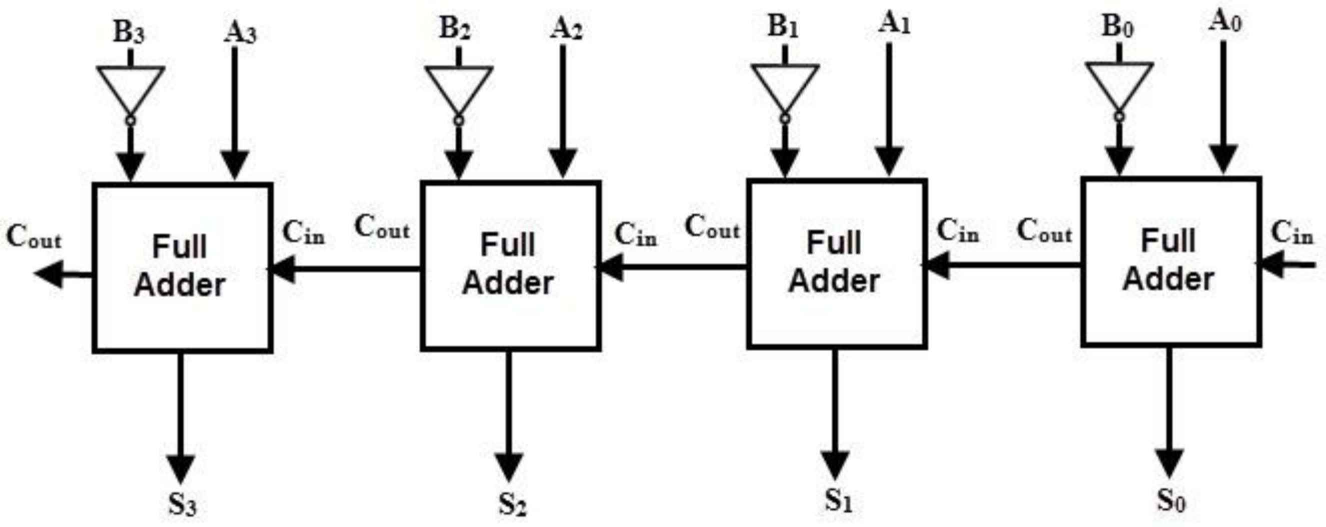
Parallel Binary Subtractor:

The parallel subtractor perform the subtraction operation based on 2s complement subtraction. The subtraction of A-B can be done by taking the 2s complement of B and adding it to A.

The 2s complement of B can be done by taking ones complement of B and adding 1 to the least significant bit.

The one complement can be implemented with inverter and 1 can be added through input carry Co.

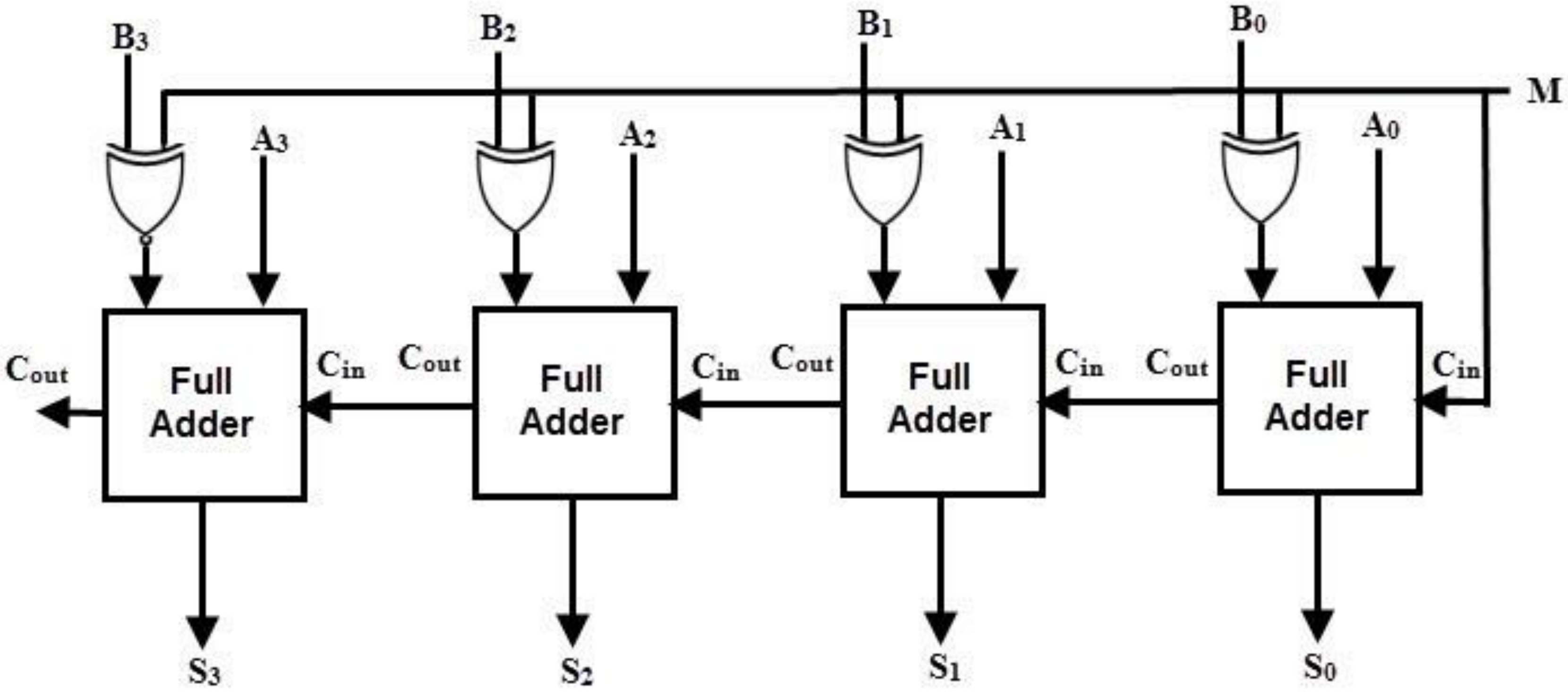
Minuend A= 1100
Subtrahend B= 0101



Carry C	1	0	0	0	1
Augend A		1	1	0	0
$\bar{B}$		1	0	1	0
Difference D		0	1	1	1

Parallel binary adder /Subtractor

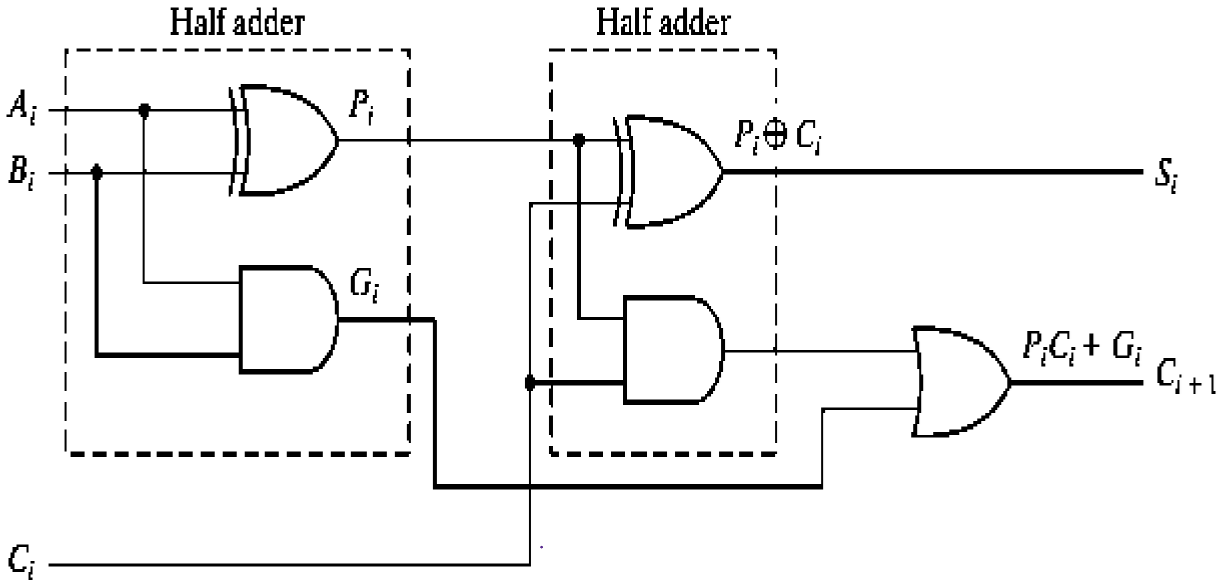
The Mode of input M control the operation of the circuit. When M=0 Co=0 the full adder get the input A and B . Hence addition operation is performed. When M=1,Co=1 the full adder gets the inputs A and B’. Hence subtraction operation is performed.



Sum and carry output of any stage cannot be produced until the input carry occur. This leads to time delay in addition process. This delay is known as carry propagation delay.

It can be eliminated by using carry look ahead adder.

Carry Look ahead adder:



$$C_0 = \text{input carry}$$

$$C_1 = G_0 + P_0C_0$$

In parallel adder output of each full adder stage is connected to carry input of the next higher order stage. Sum and carry output cannot be produced until the input carry occurs. This is called carry propagation delay.

Normally full adder has a propagation delay of 30 ns. So the total time required to perform the addition is 120ns.For a 16 bit number carry propagation delay be 480 ns.

One method of speedup the process by eliminate interstage carry delay is called look ahead carry addition.

It has two functions

1.Carry propagation
2.Carry generate

$$P_i = A_i \oplus B_i$$
$$G_i = A_i B_i$$

Output sum and carry,

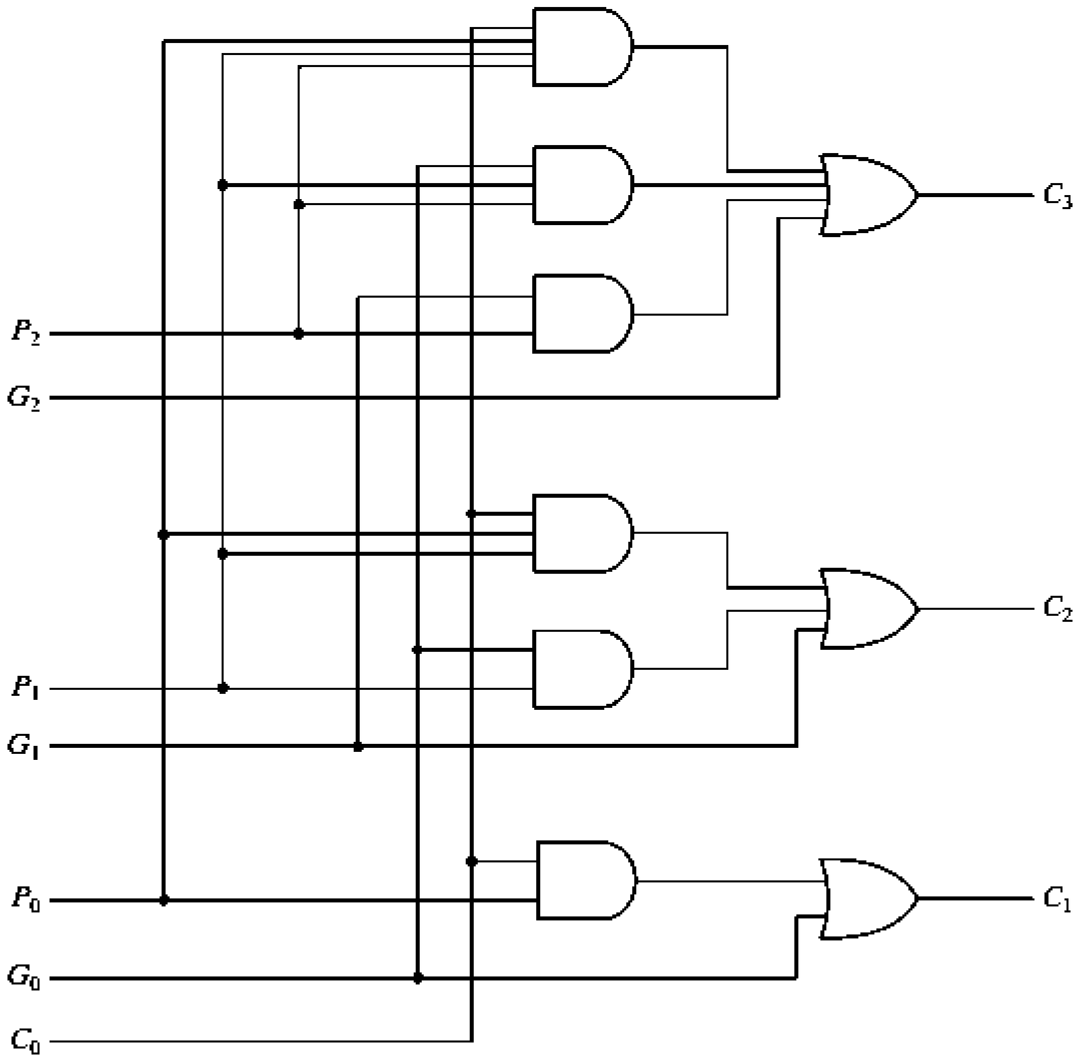
$$S_i = P_i \oplus C_i$$
$$C_{i+1} = G_i + P_i C_i$$

Gi-carry generate and it produce a carry when Ai & Bi are 1

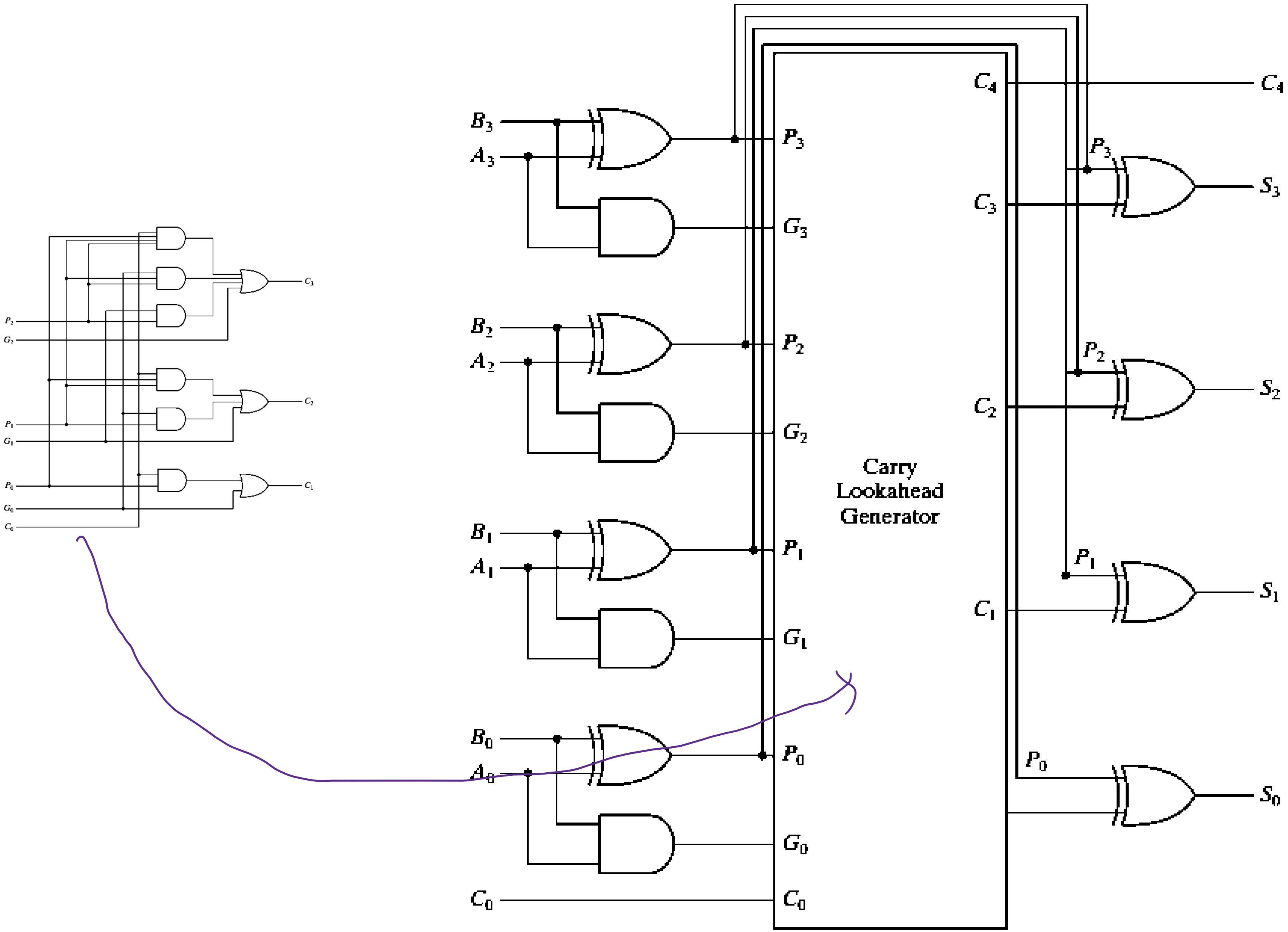
Pi- carry propagate ,it associated with propagation of carry from Ci to Ci+1

$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0)$
$C_3 = G_2 + P_2 C_2 = G_2 + P_2 (G_1 + P_1 G_0 + P_0 P_1 C_0)$

Logic diagram for carry generator:



Integrated to IC



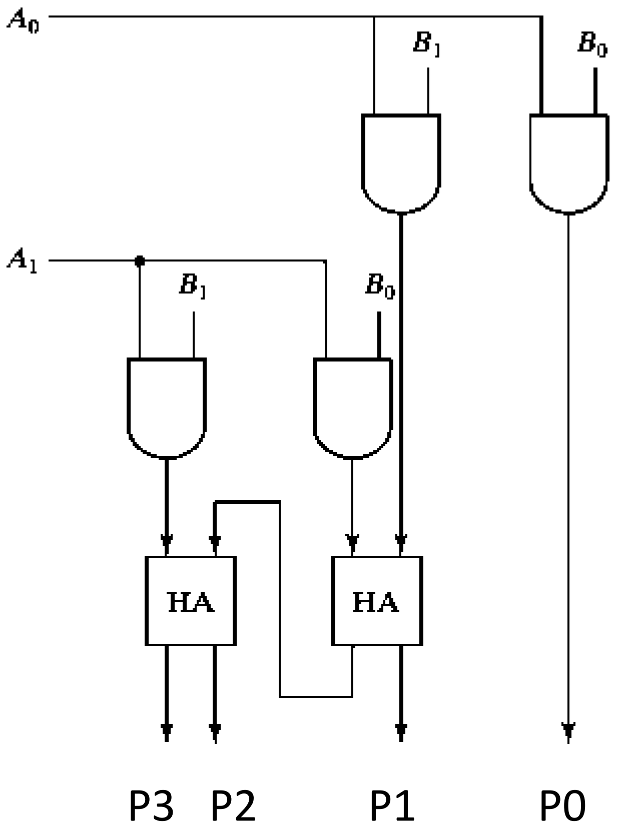
Drawback :
Circuit will complex while implementing higher bit.

Unsigned Array multiplier:

Binary multiplication is performed similar to decimal multiplication. The multiplicand is multiplied by each bit of the multiplier starting from LSB bit. Consider the binary numbers A=110 and B=101 The binary multiplication performed as follows,

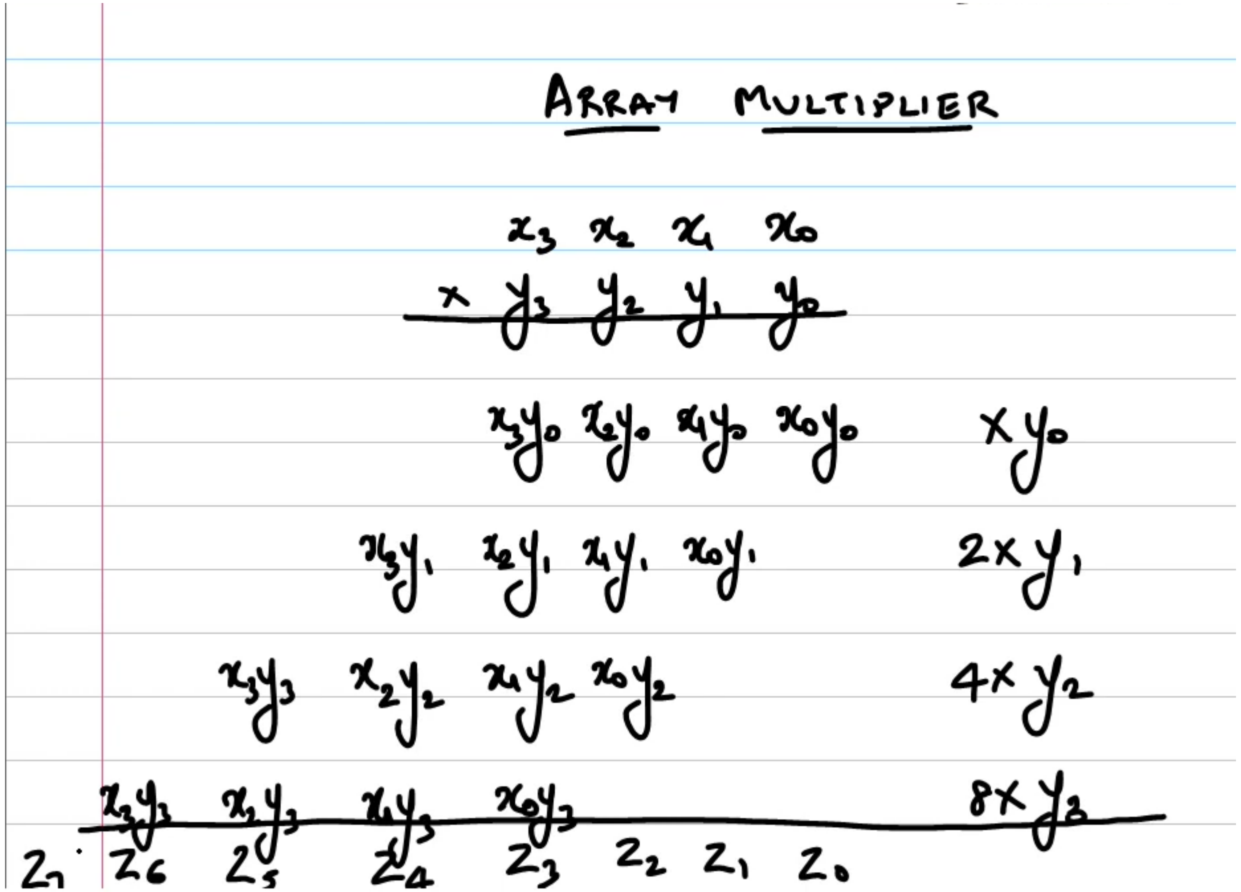
$$\begin{array}{cccc} & & B_1 & B_0 \\ & & A_1 & A_0 \\ \cline{3-4} & & A_0B_1 & A_0B_0 \\ & A_1B_1 & A_1B_0 & \\ \hline P3 & P2 & P1 & P0 \end{array}$$

$$\begin{array}{ccccc} & & 1 & 1 & 0 \\ & & 1 & 0 & 1 \\ \hline & & 1 & 1 & 0 \\ & 0 & 0 & 0 & \\ 1 & 1 & 0 & & \\ \hline 1 & 1 & 1 & 1 & 0 \\ P4 & P3 & P2 & P1 & P0 \end{array}$$



P0=A0B0
P1=B0A1+B1A0
P2=B1A1+Carry out of P1
P3=carry out of P2

Array Multiplier



ARRAY MULTIPLIER

$x_3 \quad x_2 \quad x_1 \quad x_0$

$\times \quad y_3 \quad y_2 \quad y_1 \quad y_0$

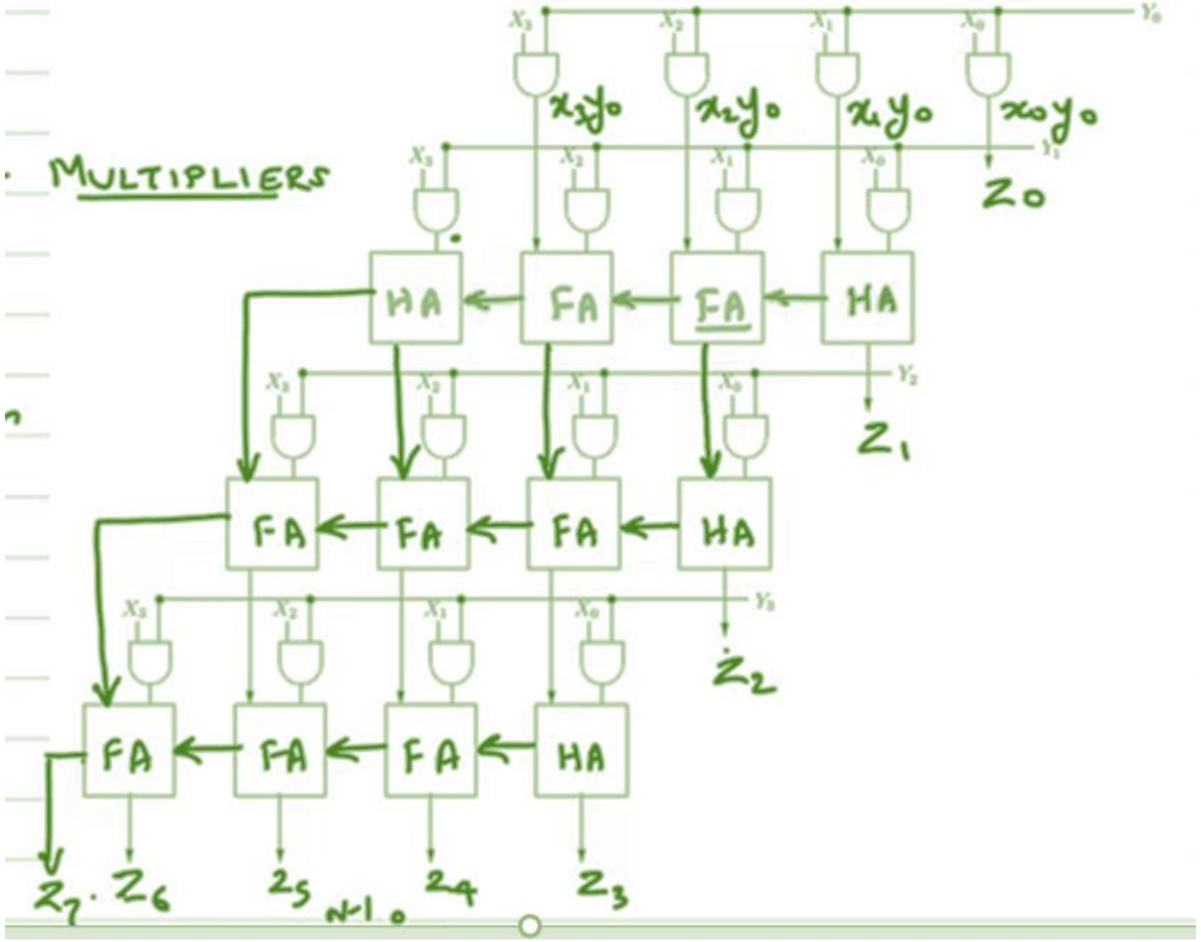
$x_3y_0 \quad x_2y_0 \quad x_1y_0 \quad x_0y_0 \qquad \times y_0$

$x_3y_1 \quad x_2y_1 \quad x_1y_1 \quad x_0y_1 \qquad 2\times y_1$

$x_3y_3 \quad x_2y_2 \quad x_1y_2 \quad x_0y_2 \qquad 4\times y_2$

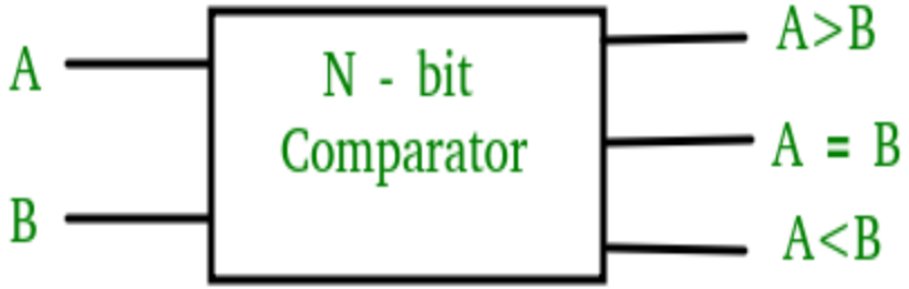
$x_3y_3 \quad x_2y_3 \quad x_1y_3 \quad x_0y_3 \qquad 8\times y_3$

$z_7 . \; z_6 \quad z_5 \quad z_4 \quad z_3 \quad z_2 \quad z_1 \quad z_0$

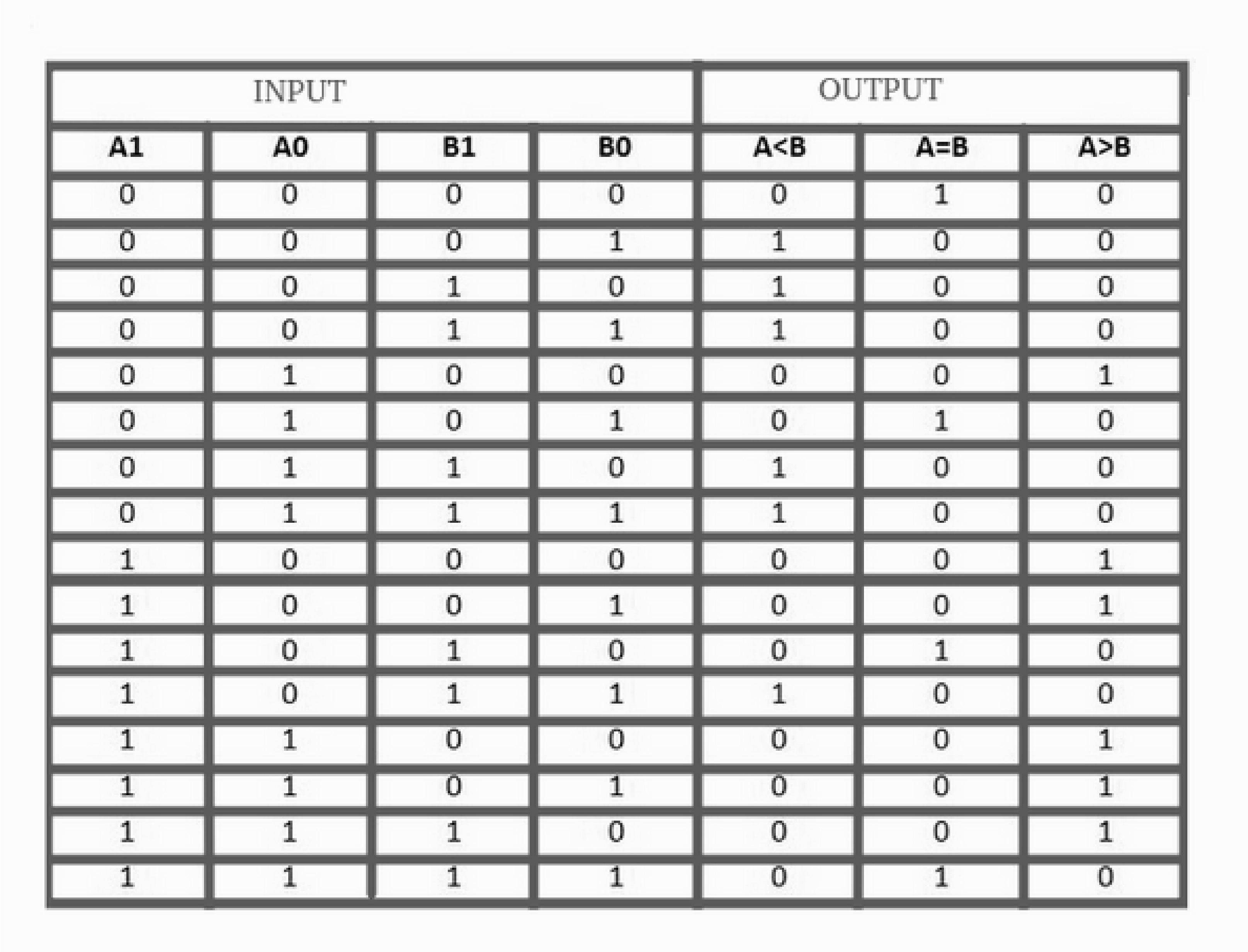


2-bit magnitude comparator:

- A comparator used to compare two binary numbers each of two bits is called a 2-bit Magnitude comparator. It consists of four inputs and three outputs to generate less than, equal to, and greater than between two binary numbers.

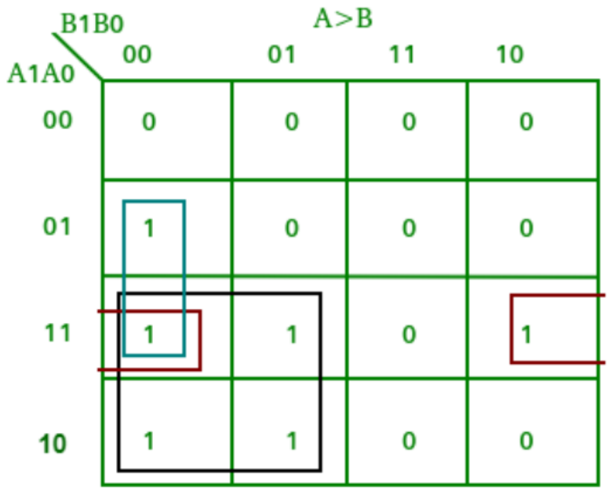


Truth table:



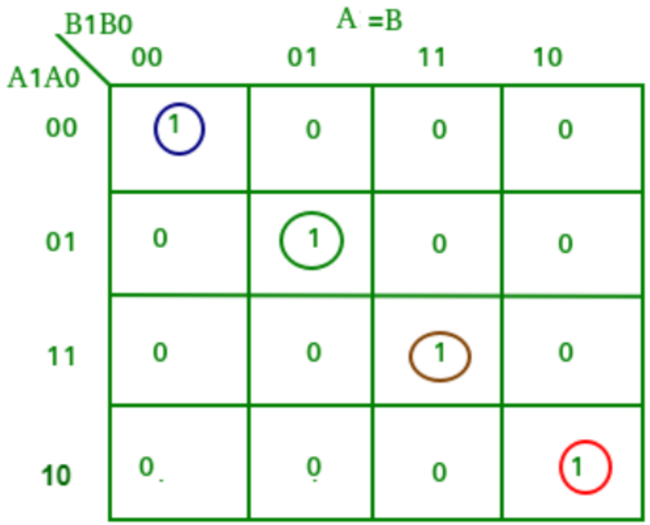
INPUT				OUTPUT		
A1	A0	B1	B0	A<B	A=B	A>B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

K-map Simplification:



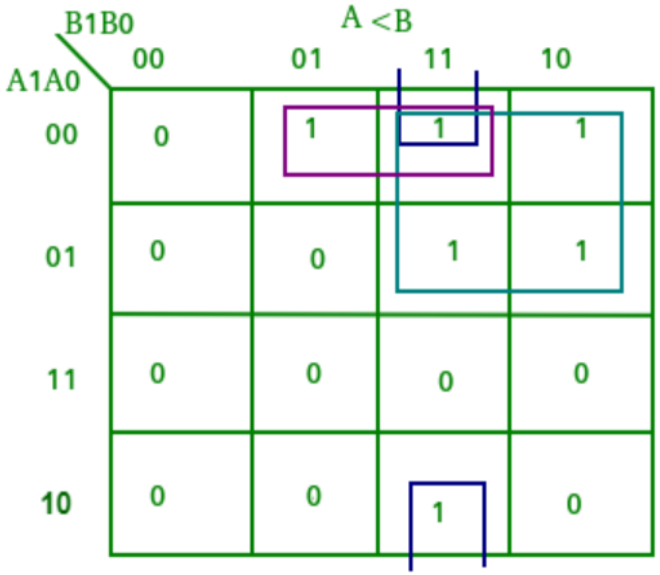
A>B

B1B0 A1A0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0



A =B

B1B0 A1A0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

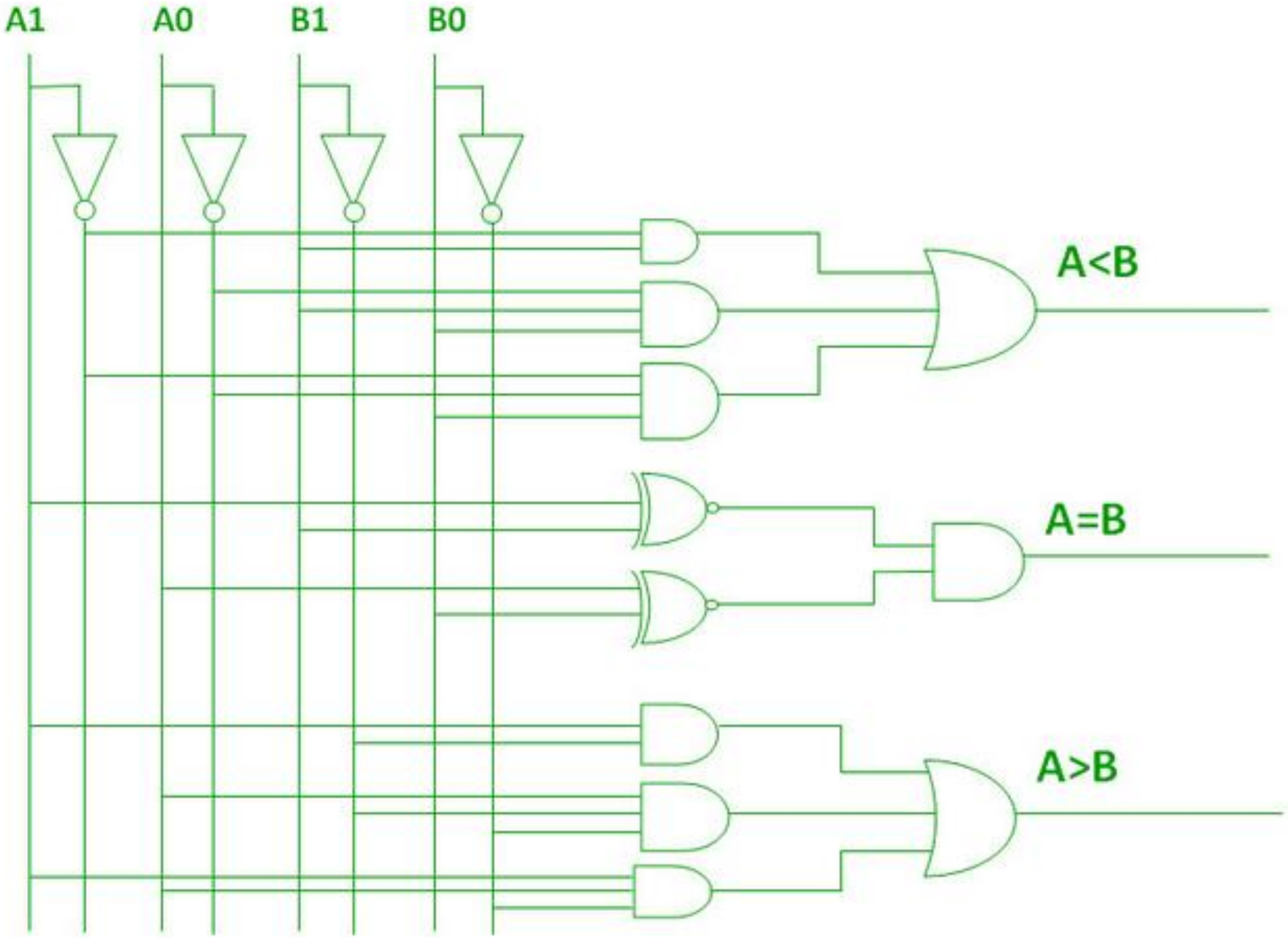


A <B

B1B0 A1A0	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

```
A>B:A1B1’ + A0B1’B0’ + A1A0B0’
A=B: A1’A0’B1’B0’ + A1’A0B1’B0 + A1A0B1B0 + A1A0’B1B0’
   : A1’B1’ (A0’B0’ + A0B0) + A1B1 (A0B0 + A0’B0’)
   : (A0B0 + A0’B0’) (A1B1 + A1’B1’)
   : (A0 Ex-Nor B0) (A1 Ex-Nor B1)
A<B:A1’B1 + A0’B1B0 + A1’A0’B0
```

Logic Diagram for 2 bit comparator:

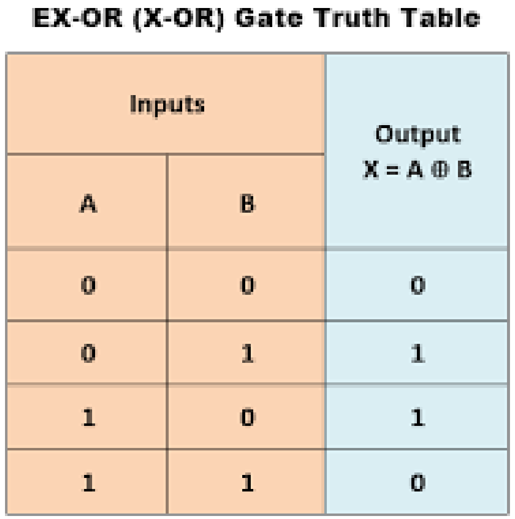


4-bit magnitude comparator:

For A<B

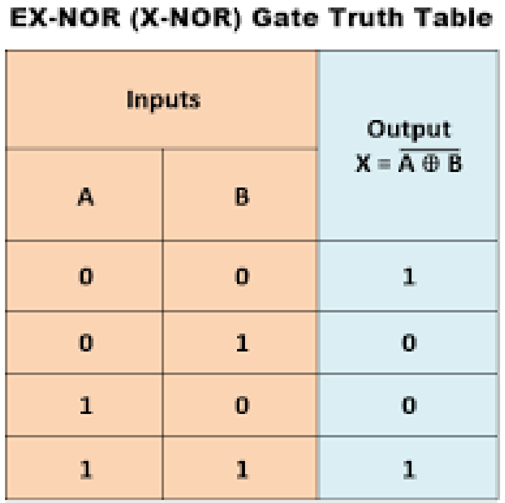
$B_3 > A_3$,
$A_3 = B_3$, B2>A2 $_,$
$A_3 = B_3$, $A_2 = B_2$ $_,$ $B_1 > A_1$ $_,$
$A_3 = B_3$, $A_2 = B_2$ $_,$ $A_1 = B_1$ $_,$ $B_0 > A_0$

$$(A < B) = A'_3 B_3 + x_3A'_2 B_2 + x_3x_2A'_1B_1 + x_3x_2x_1A'_0B_0$$



EX-OR (X-OR) Gate Truth Table

Inputs		Output $X = A \oplus B$
A	B	
0	0	0
0	1	1
1	0	1
1	1	0



EX-NOR (X-NOR) Gate Truth Table

Inputs		Output $X = \overline{A \oplus B}$
A	B	
0	0	1
0	1	0
1	0	0
1	1	1

4 bit Binary numbers A and B represented as,
$A = A_3A_2A_1\ A_0$
$B = B_3\ B_2\ B_1B_0$

For A=B,
$A_3 = B_3$, $A_2 = B_2$ $_,$ $A_1 = B_1$ $_,$ $A_0 = B_0$

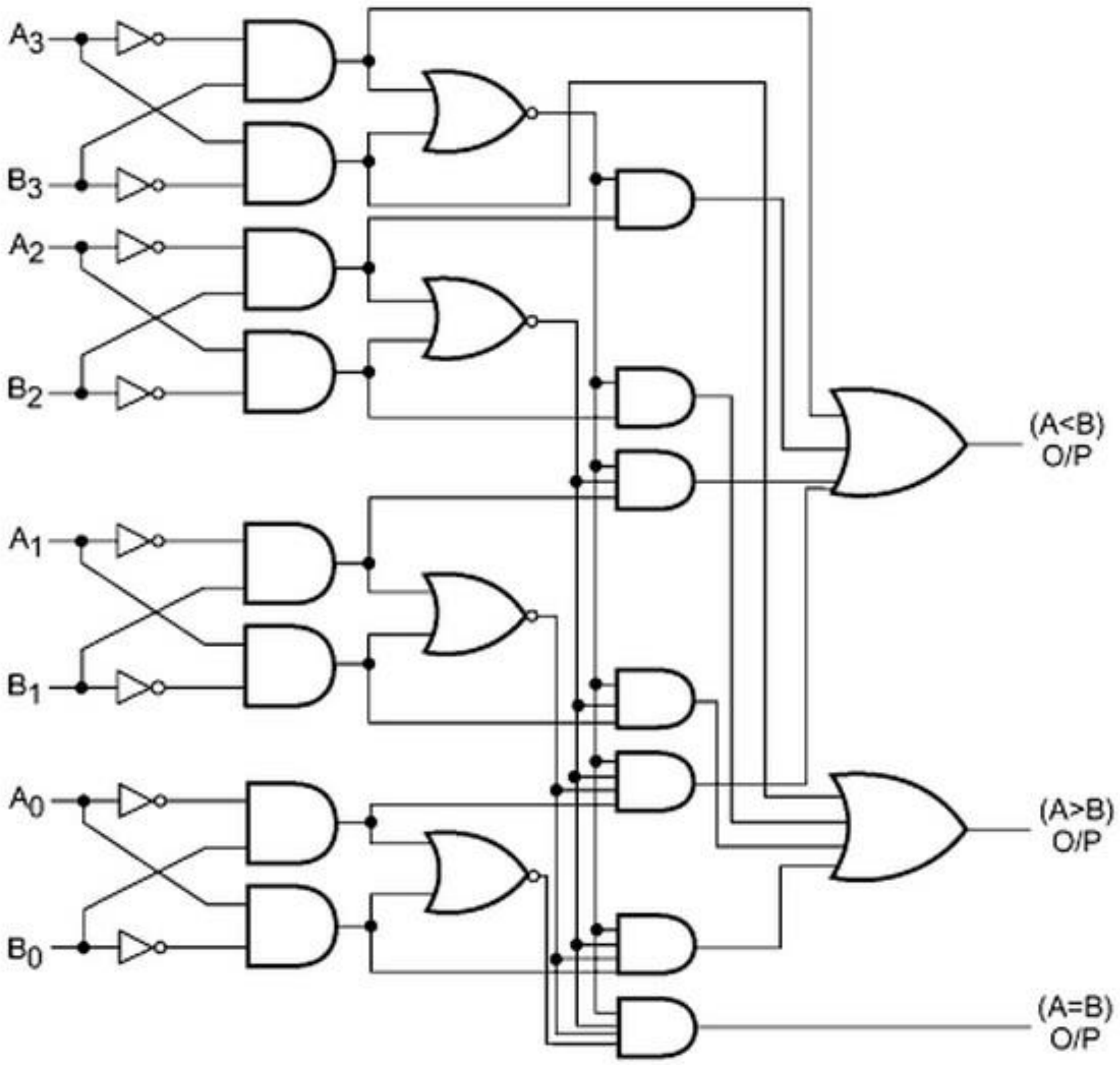
$$(A = B) = x_3\ x_2\ x_1\ x_0$$

For A>B

$A_3\ > B_3$,
$A_3 = B_3$, A2>B2 $_,$
$A_3 = B_3$, $A_2 = B_2$ $_,$ $A_1 > B_1$ $_,$
$A_3 = B_3$, $A_2 = B_2$ $_,$ $A_1 = B_1$ $_,$ $A_0 > B_0$

$$(A > B) = A_3\ B'_3 + x_3A_2\ B'_2 + x_3x_2A_1B'_1 + x_3x_2x_1A_0B'_0$$

Logic Diagram for 4 bit magnitude Comparator:



Applications of Comparators:

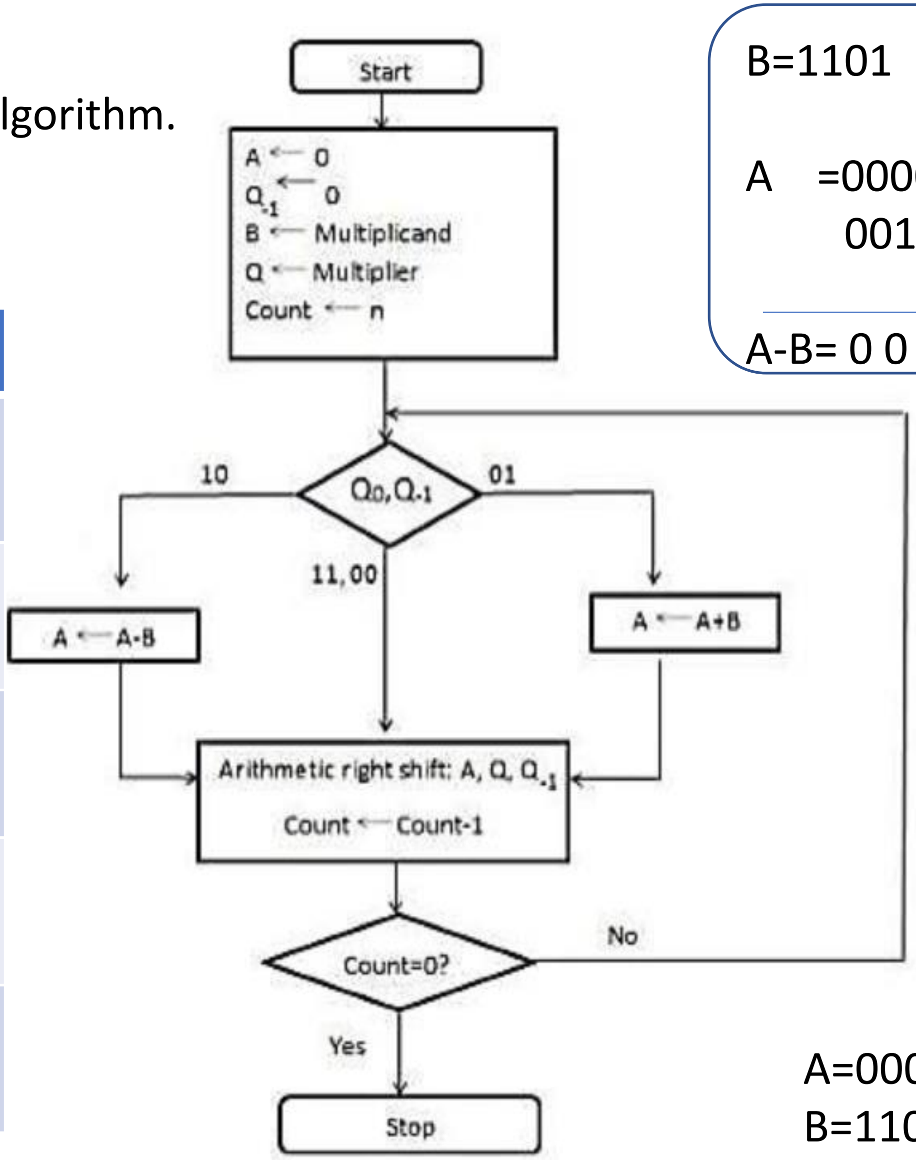
1. Comparators are used in central processing units (CPUs) and microcontrollers (MCUs).
2. These are used in control applications in which the binary numbers representing physical variables such as temperature, position, etc. are compared with a reference value.
3. Comparators are also used as process controllers and for Servo motor control.
4. Used in password verification and biometric applications.

Booth Multiplier:

Multiply signed binary number -3 and 4 using Booth multiplier algorithm.

Q=4(Multiplier) A=0 B=multiplicand B=1101 (-3)

Steps	A	Q	Q-1	Process
	0 0 0 0	0 1 0 0	0	initial
1.	0 0 0 0	0 0 1 0	0	Right shift
2.	0 0 0 0	0 0 0 1	0	Right shift
3.	0 0 1 1 0 0 0 1	0 0 0 1 1 0 0 0	0 1	A-B ,right shift
4.	1 1 1 0 1 1 1 1	1 0 0 0 0 1 0 0	1 0	A+B,right shift



B=1101

A =0000
0010
1

A-B= 0 0 1 1

A=0001
B=1101
1110

Serial adder: