Unit - II

Combinational Logic Circuits

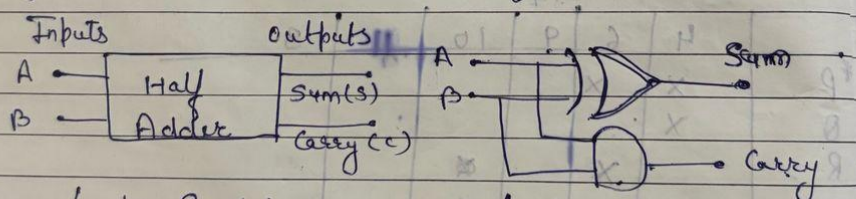
A digital System Consists of two types of circuits, Namely

Combinational logic Circuit.
Sequential logic Circuit.

In a Combinational Circuit, the output at any time depends only on the input value at that time.

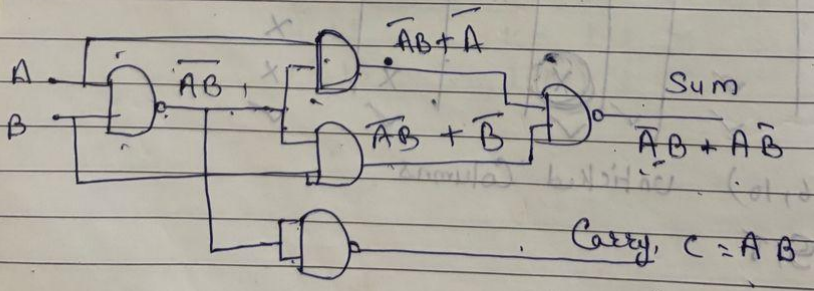
Half Adder :

The logic circuit that add two bits producing a Sum and a carry to be used in the next higher position is called half adder.



Logic Symbol

Logic Diagram



Using NAND Gate

Truth Table of

Inputs	
A	B
0	0
0	1
1	0
1	1

In the truth

AB = 01

in the logic

Therefore,

Full Add

A full

adds

bit and

It has

A

B

C_{in}

Truth Table of Half Adder

Inputs		Outputs	
A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

In the truth Table, the Sum output is 1, when AB = 01 and AB = 10, therfore the Sum is

$$S = \bar{A}B + A\bar{B} \quad -(1) \qquad A \oplus B$$

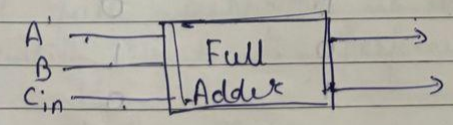
in the logic truth table, the Carry is 1 when AB=11 therefore, the Carry is

$$C = AB \quad -(2)$$

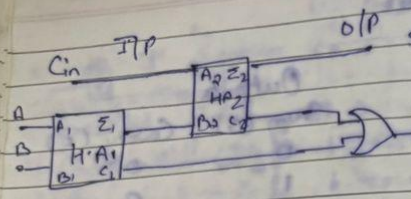
Full Adder :

A full Adder is a Combinational circuit that adds three bits & Carry and output a Sum bit and a carry bit.

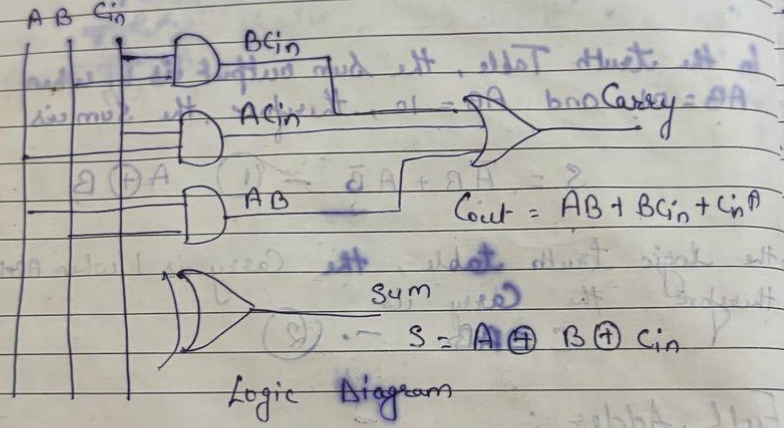
It has three inputs and two outputs



Logic Symbol



Symbol using 2 half-adders



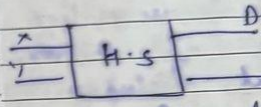
Logic Diagram

Truth Table of full Adder

I/P			O/P	
A	B	Cin	Sum	Carry output
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

Half Subtractor

A half subtra
which is us
two bits.



Logic Symbol

Truth table of

Inputs	
X	Y
0	0
0	1
1	0
1	1

D =
B

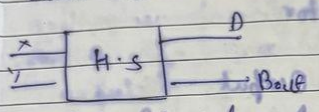
Full Subtra

A full
which u
& bits.

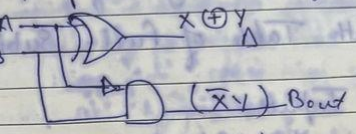
X
Y
Bin

Half Subtractor:

A half subtractor is a combinational circuit which is used to perform subtraction of two bits.



Logic Symbol



Logic Diagram

Truth table of Half Subtractor

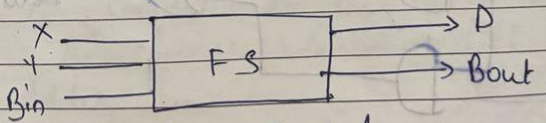
Inputs		Output	
X	Y	Difference (D)	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = \bar{X}Y + X\bar{Y} = X \oplus Y$

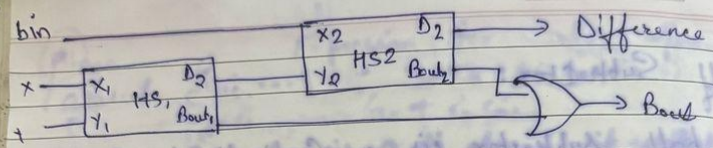
B ~~Sum~~ $= \bar{X}Y$

Full Subtractor

A full subtractor is a combinational circuit which used to perform subtraction of three bits.



Logic Symbol



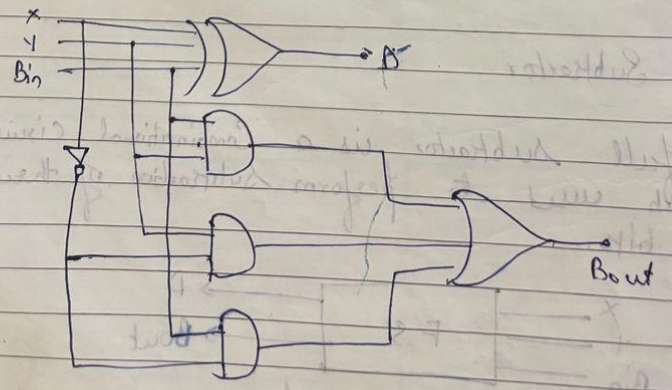
Using half Subtractors

Truth Table of Full- Subtractor

Inputs			Output	
X	Y	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

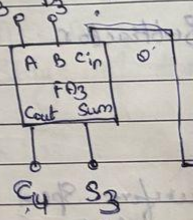
$D = X \oplus Y \oplus Bin$

$Bout = \overline{X}Y + \overline{X}Bin + YBin$



4-Bit Parallel

A single full a
one bit numbers
adder is a di
arithmetic Sum
greater than o
Corresponding p
Consists of ful
where the ou
is Connected
higher order
n-bit parall
to perform



Working of p

As shown
FA, adds
C_1 to ge
which is
chain .
full Ad

→ Difference

→ Bout

Bout
0
1
1
0
0
0
1

A = 1 0 1 1
B = 1 0 1 0

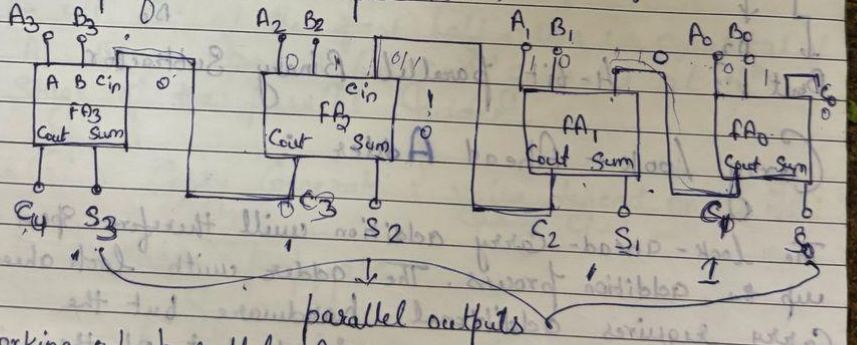
Cin

F.A

HSB LSB
A = 0 1 0 1
B = 1 0 1 0

4-Bit Parallel Adder

A single full adder performs the addition of two one bit numbers and an input Carry. But a parallel adder is a digital circuit capable of finding the arithmetic sum of two binary numbers that is greater than one bit in length by operating on corresponding pairs of bits in parallel. It consists of full adders connected in a chain where the output Carry from each full adder is connected to the Carry input of the next higher order full adder in the chain. A n-bit parallel adder requires n full adders to perform the operation.

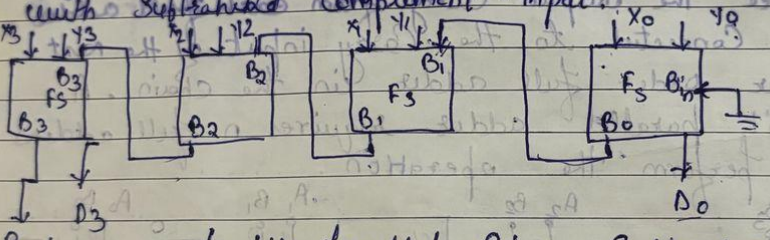


Working of parallel Adder :-

As shown in the figures, firstly the full Adder FA_1 adds A_1 and B_1 along with the Carry C_1 to generate the Sum S_1 & the Carry C_2 which is connected to the next adder in chain. The process continue till the last full Adder.

Parallel Subtractor :

A parallel Subtractor is a digital circuit capable of finding the arithmetic difference of two Binary numbers that is greater than one bit in length by operating on Corresponding pairs of bits in parallel. The parallel subtractor can be designed in several ways including Combination of half and full Subtractors, with Subtrahend Complement input.



4-bit parallel Binary Subtractor

Carry Look-Ahead Adder :

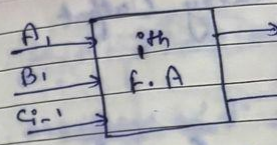
The look-ahead-carry addition will therefore speed up the addition process. The adder with look ahead carry requires additional hardware but the speed of this adder is independent of the number of bits.

$P_i = A_i \oplus B_i$

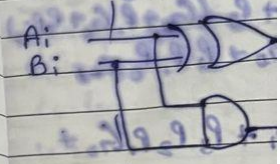
$G_i = A_i \cdot B_i$

$S_i = P_i \oplus C_{i-1} = A_i \oplus B_i \oplus C_{i-1}$

$C_i = G_i + P_i C_{i-1}$



Block Dia



Half adder

The o/p G_i e

$A_i = B_i = 1$

Stage of the

G_i = Car

P_i = Car

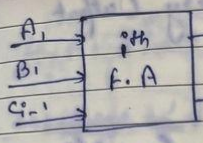
We write

adder.

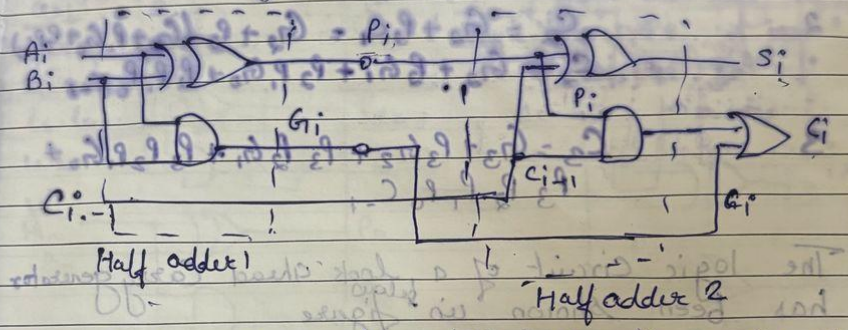
P_i =

FA – 2

4.



Block Diagram i^{th} full adder



The o/p G_i of the first half adder is equal to 1 if $A_i = B_i = 1$ and a "Carry" is generated at i^{th} Stage of the parallel adder.

G_i = Carry Generate { value depend on the i/p Carry C_{i-1}

P_i = Carry propagate

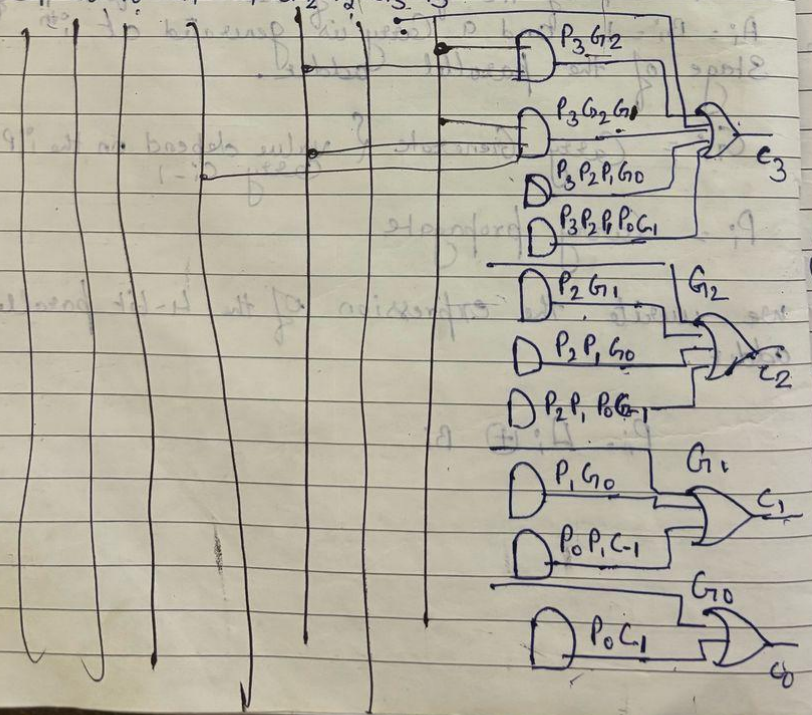
We write the expression of the 4-bit parallel adder.

$$P_i = A_i \oplus B_i$$

Stage	Expression for Carry Output
0	$C_0 = G_0 + P_0 C_{-1}$
1	$C_1 = G_1 + P_1 C_0 = G_1 + P_1(G_0 + P_0 C_{-1})$ $C_1 = G_1 + P_1 G_0 + P_0 P_1 C_{-1}$
2	$C_2 = G_2 + P_2 C_1 = G_2 + P_2 (G_1 + P_1 G_0 + P_0 P_1 C_{-1})$ $C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1}$
3	$C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{-1}$

The logic circuit of a look ahead Carry generator has been shown in figure

C_{-1} G_0 P_0 G_1 P_1 G_2 P_2 G_3 P_3



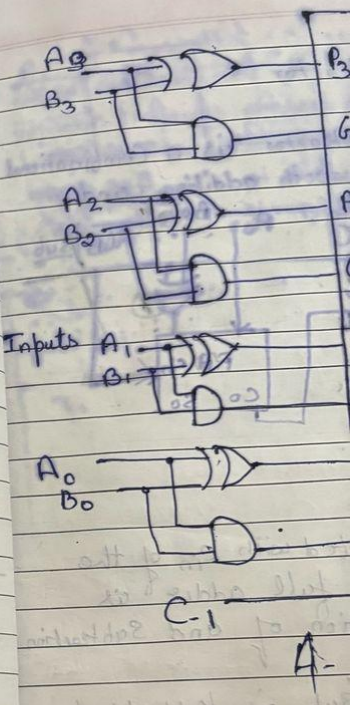
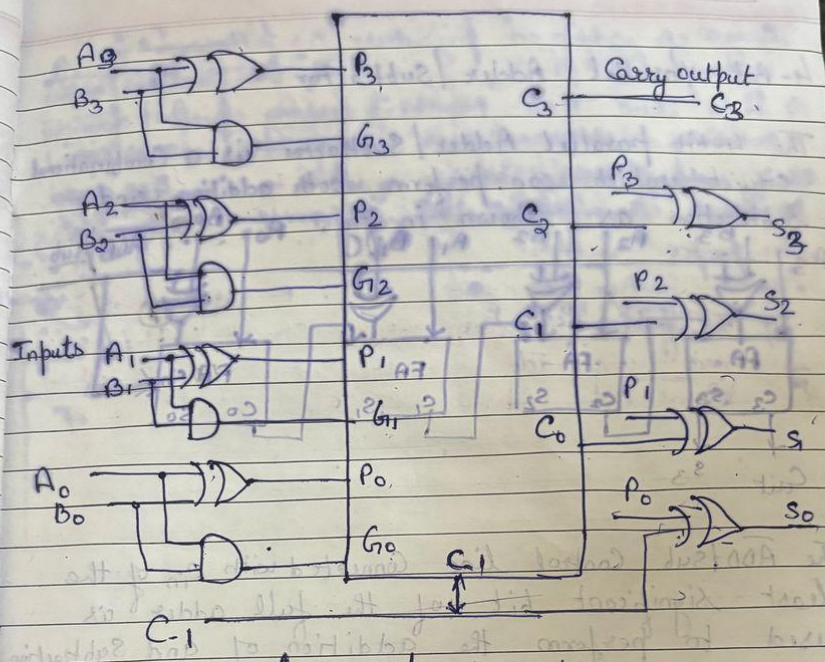


Fig. Shows the
bit parallel ad
generator. The P
from the inpu
Ex-OR and
C_0 through
by the look a
Carry output
to produce th

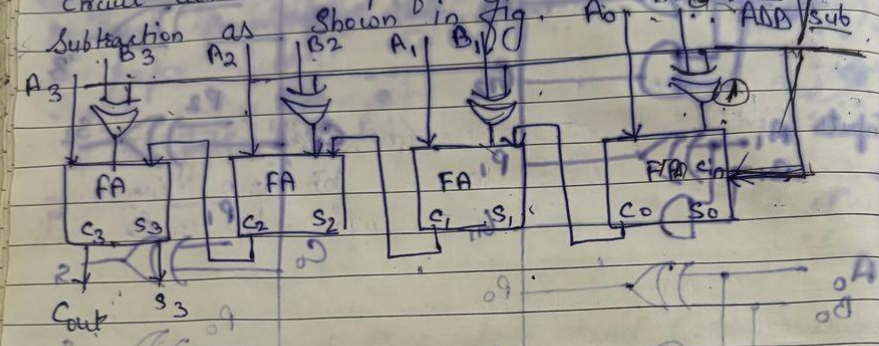


4-bit parallel Adder

Fig. shows the block diagram of a four bit parallel adder using the look-ahead carry generator. The P_i and G_i variables are obtained from the inputs A_i and B_i by using the Ex-OR and AND gates rly. The carry outputs C_0 through C_3 are produced simultaneously by the look ahead carry generator. These carry outputs and P_i variables are Ex-ORed to produce the sum outputs S_0 through S_3.

4-Bit Parallel Adder / Subtractor

The 4-bit parallel Adder / Subtractor is a Combinational circuit which can perform both addition and Subtraction as shown in fig.



The $\overline{ADD}$/Sub Control line Connected with C_{in} of the least Significant bit of the full adder is used to perform the addition of and Subtraction

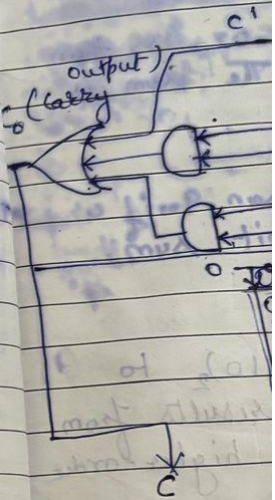
To perform Subtraction, $\overline{ADD}$ Sub is kept high (i.e = 1)

Now Controlled inputs produce the 1's Complement of addend (i.e B_3, B_2, B_1, B_0) Since 1 is added in 1's Complement which produce 2's complement of the addend before addition. Now the data ~~added~~ A_3, A_2, A_1, A_0 will be added to the 2's Complement of B_3, B_2, B_1, B_0 which is equivalent to Subtraction.

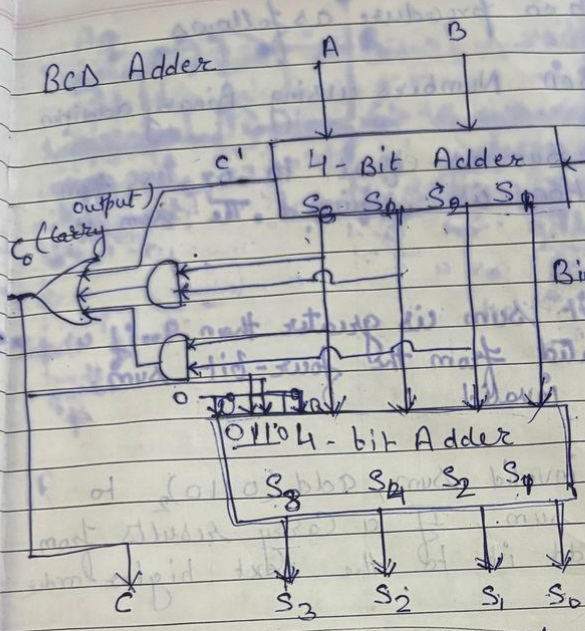
To perform Addition $\overline{ADD}$/Sub is kept Low (i.e. 0)

400 111
011 111
011

BCD Adder



The digital
form of B
BCD add
digits ar
BCD numbe
represented
each BCD
Binary Nu
526,



Block Diagram of BCD Adder

The digital Systems handles the decimal Number in the form of Binary Coded decimal Number (BCD). A BCD adder is a circuit that adds two BCD digits and produces a Sum digit also in BCD. BCD number uses 10 digits, 0 to 9 which are represented in the binary form 0000 to 1001 i.e each BCD digit is represented as a 4-bit Binary Number, For Ex- we write BCD Number 526;

5	2	6
↓	↓	↓
0101	0010	0110

The BCD addition procedure as follows

1) Add two BCD Numbers using Binary addition

2. If four-bit sum is equal to or less than 9, no Correction is required. The sum is in proper BCD.

3. If the four-bit sum is greater than 9 or if a Carry is generated from the four-bit sum, the Sum is invalid.

4. To Correct the invalid Sum, add $(0110)_2$ to the four-bit Sum. If a Carry results from this addition, add it to the next higher order BCD digit.

Decimal Digit	Uncorrected BCD sum					Corrected BCD Sum				
	C_3	S_3	S_2	S_1	S_0	Carry	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1

	S_3
12	1
13	1
14	1
15	1

Correction

S_1S_0	S_3S_2 00
00	
01	
11	
10	

C_n =

with this de
block diagr

Block

as shown in
together wi
in the
a binary 8
to zero. i.
is added
one (i.e
added to th
binary ad
the bottom

	S_3	S_2	S_1	S_0	Cout	S_3	S_2	S_1	S_0
12	1	1	0	0	1	0	0	1	0
13	1	1	0	1	1	0	0	1	1
14	1	1	1	0	1	0	1	0	0
15	1	1	1	1	1	0	1	0	1

Correction Required

S_1S_0 \ S_3S_2	00	01	11	10
00			12	
01			13	
11			14	16
10			15	10

$$C_n = S_3S_2 + S_1S_3$$

K-Map Simplification

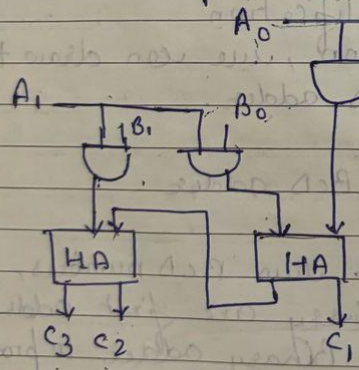
With this design equation, we can draw the block diagram of BCD adder.

Block Diagram of BCD adder

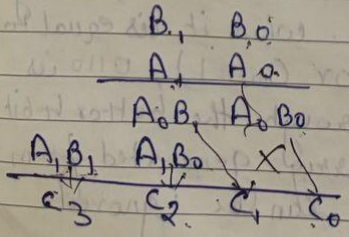
As shown in the fig, the two BCD numbers, together with input Carry are first added in the top 4-bit Binary adder to produce a binary Sum. When the o/p Carry is equal to zero. i.e when (Sum $\leq$ 9 and Cout = 0) nothing is added to the binary sum. when it is equal to one (i.e when Sum > 9 or Cout = 1), 0110 is added to the binary Sum through the bottom 4-bit binary adder. The o/p Carry generated from the bottom binary adder can be ignored.

Binary Multiplier :

A Binary Multiplier is a Combinational logic circuit or digital device used for Multiplying two Binary Numbers. The two numbers are More specifically known as Multiplicand and Multiplier and the result is known as a product. Binary Multiplication Method is same as decimal Multiplication. Binary Multiplication of more than 1-bit numbers contains 2 steps. The 1st step is single bit wise Multiplication known as partial product & the 2nd step is adding all partial products into a single product

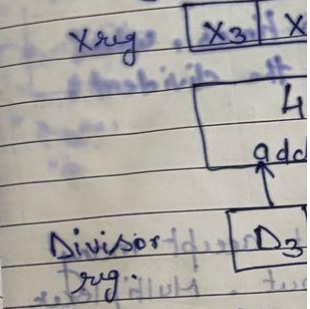


2-Bit by 2-bit Binary Multiplier



Binary Divider

A Block diagr
using restoring
in figure.



Here, the divide
Y (Y3, Y2 Y1 Y0
divisor regist
X register (

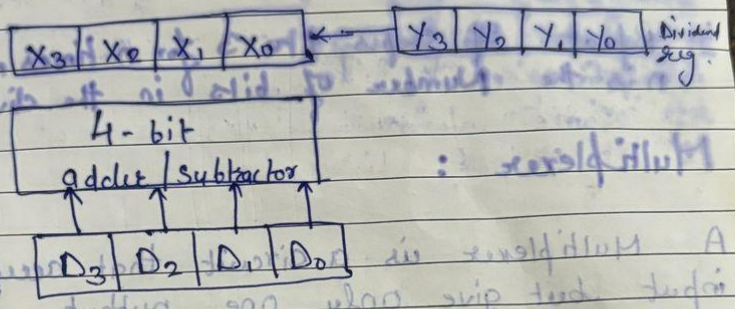
The procedure
circuit sh
as follows :

1. Shift the
the left

2. Perform tri
of D-reg

Binary Divider :

A Block diagram of Binary divider unit using restoring technique for division is shown in figure.



Block Diagram of a 4-bit binary divider

Here, the dividend is stored in the dividend register Y (Y_3, Y_2, Y_1, Y_0), the divisor is stored in the divisor register D ($D_3 D_2 D_1 D_0$) and initially the X register ($X_3 X_2 X_1 X_0$) is cleared.

The procedure for division operation using the circuit shown in above figure is explained as follows:

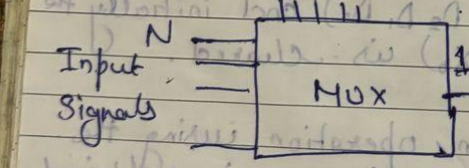
1. Shift the Combined Content of X & Y registers to the left by one bit.
2. Perform Trial Subtraction by subtracting the Content of D-register from the Content of X-register.

3. If there is no borrow in the previous Subtraction put 1, in the LSB of Y register, else restore the Original Content of X register by adding the Contents of A register with the Contents of X register.

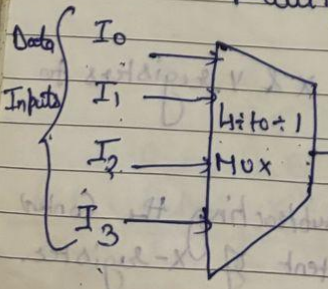
4. Repeat Sth Steps 1 to 3 for n times, where n is the Number of bits in the divident.

Multiplexer :

A Multiplexer is a circuit that accept many input but give only one output. Multiplexer means Many into One. A Multiplexer is a circuit used to select and route any one of the Several input signals to a Single output. M control line



Multiplexer Pin Diagram



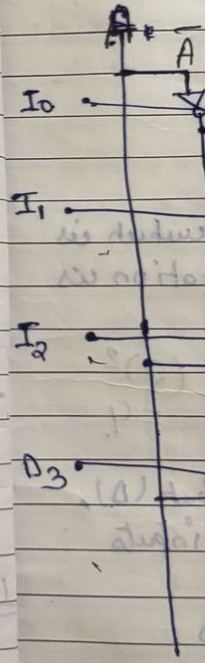
A	B	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$m_0 = A'B'$

Z =

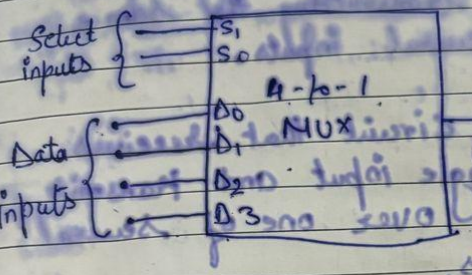
Select inputs

Data inputs

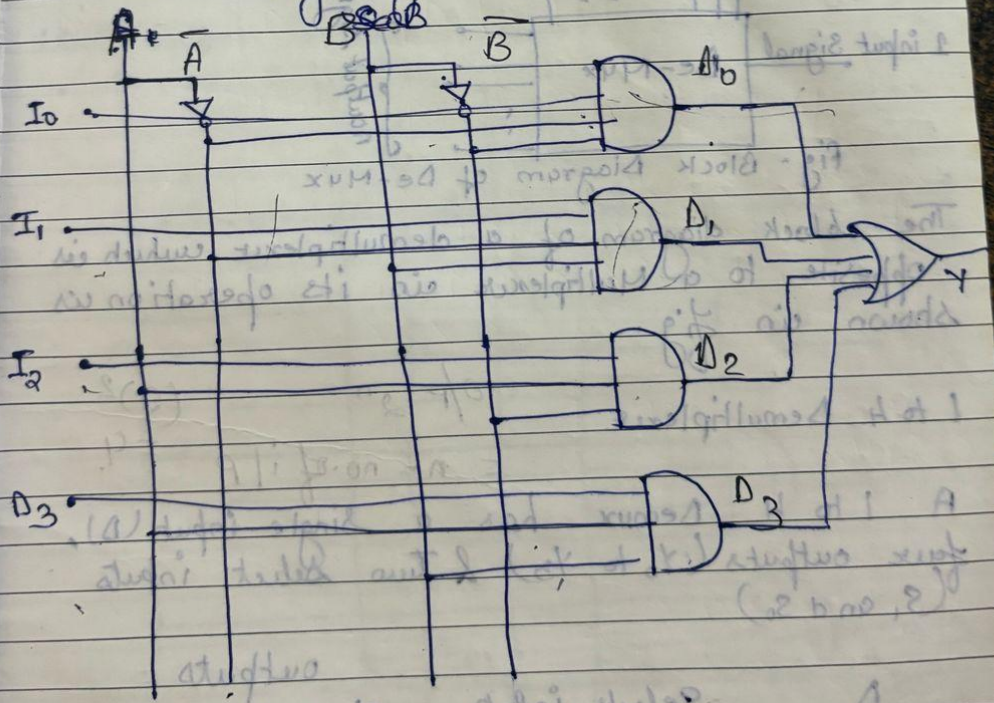


$m_0 = A'B'$, $m_1 = A'B$, $m_2 = A \cdot B'$ $m_3 = A \cdot B$

$Z = A' \cdot B' I_0 + A' B I_1 + A \cdot B I_2 + A \cdot B I_3$



Logic Symbol



(b) Logic Diagram 4-to-1 MUX

De-Multiplexer:

The word "demultiplexer" means one into Many Demultiplexing is the process of taking information from one input & transmitting the same over one of several inputs.

A De-mux is a logic circuit that receives information on a single input and transmit the same information over one of several (2^m) output lines

fig. Block Diagram of De-Mux

The block diagram of a demultiplexer which is opposite to a Multiplexer in its operation is shown in fig.

o/p 2^n $(2)^2$

= n = no. of i/p =4.

1 to 4 Demultiplexer

A 1 to 4 Demux has a single input (D), four outputs (Y_0 to Y_3) & Two Select inputs (S_1 and S_0)

D	Selects inputs S_1	S_0	outputs Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	D
D	0	1	0	0	D	0
D	1	0	0	D	0	0
D	1	1	D	0	0	0

Also, from the Outputs can

Y_0 =
Y_1 =
Y_2
Y_3

The 1-to-4 using four gates as

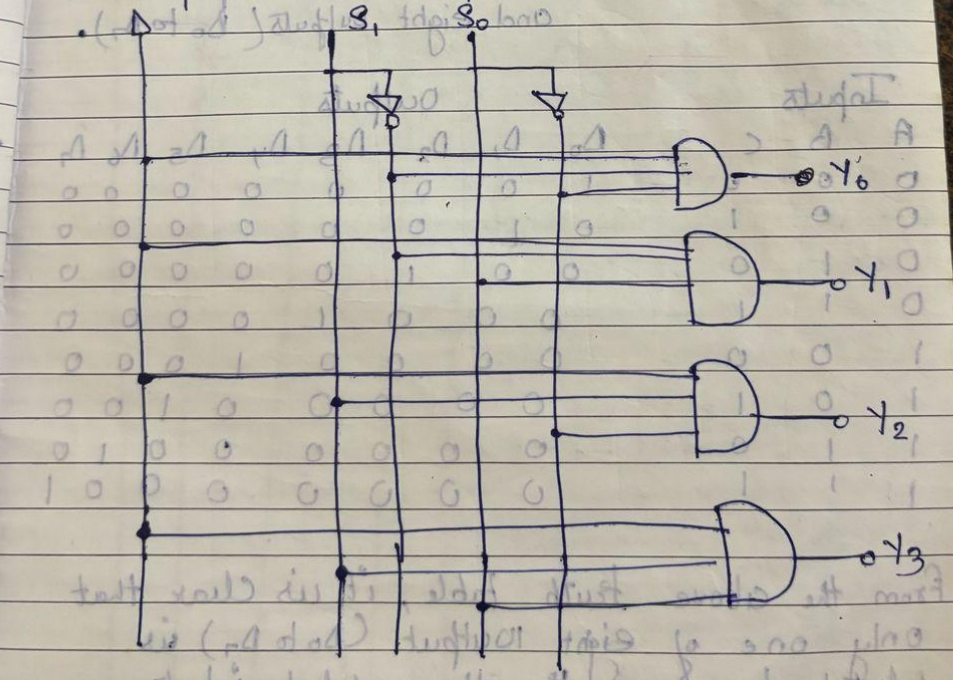
Data input

Logic

Also, from the truth table, the expressions for outputs can be written as follows:

$$Y_0 = \bar{S}_1 \bar{S}_0 D$$
$$Y_1 = \bar{S}_1 S_0 D$$
$$Y_2 = S_1 \bar{S}_0 D$$
$$Y_3 = S_1 S_0 D$$

The 1-to-4 demultiplexer can be implemented using four 3-input AND gates and two NOT gates as shown in figure



Logic Diagram of 1-to-4 demultiplexer

1:4 1:8 $(2)^3 = 8$

Decoder :

A decoder is a logic circuit that converts an n-bit binary input code (data) into 2^n output line, such that each output line will be activated for only one of the possible combinations of inputs.

In a decoder, the number of output is greater than the number of inputs.

3 to 8 Decoder : It has three inputs (A, B, C) and eight outputs (D_0 to D_7).

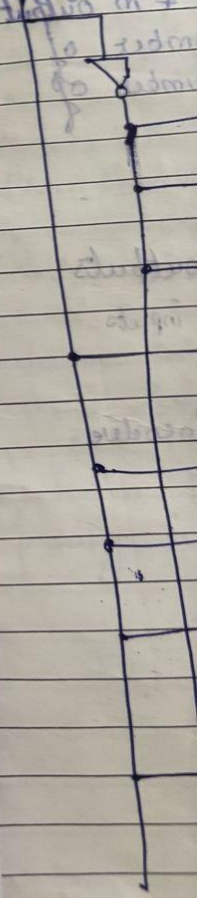
Inputs			Outputs							
A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

From the above truth table, it is clear that only one of eight output (D_0 to D_7) is selected based on the three select inputs.

The logic expres
as follows

$D_0 = \bar{A}\bar{B}\bar{C}$, D_1
$D_4 = A\bar{B}\bar{C}$; D_5

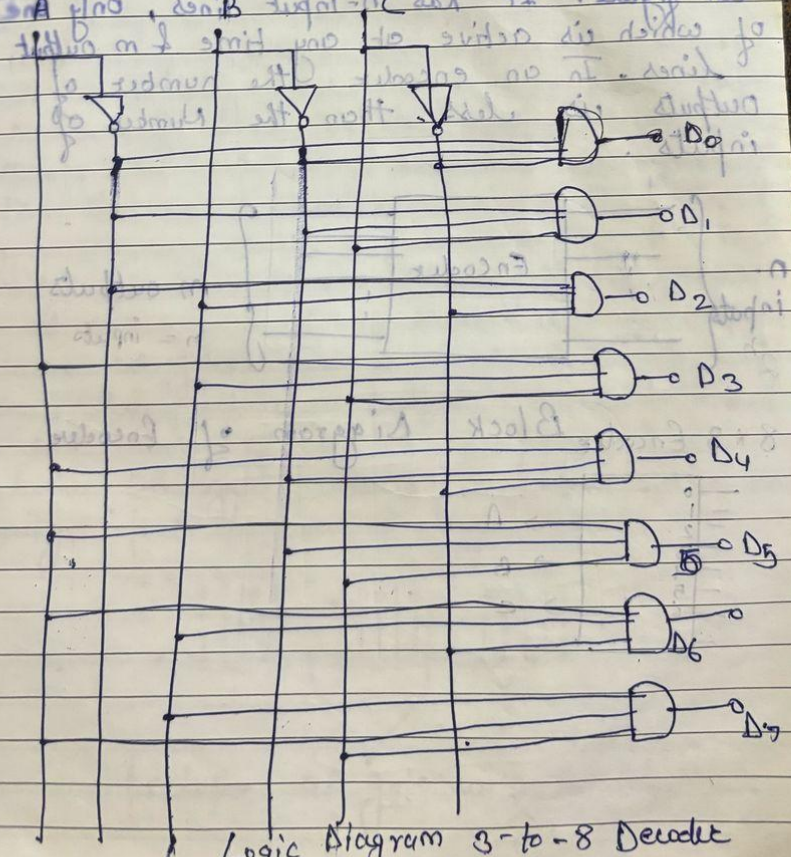
The circuit of
by using 3 N
as shown in



The logic expression for the output can be written as follows

$D_0 = \bar{A}\bar{B}\bar{C}$, $D_1 = \bar{A}\bar{B}C$, $D_2 = \bar{A}B\bar{C}$, $D_3 = \bar{A}BC$
$D_4 = A\bar{B}\bar{C}$; $D_5 = A\bar{B}C$; $D_6 = AB\bar{C}$, $D_7 = ABC$

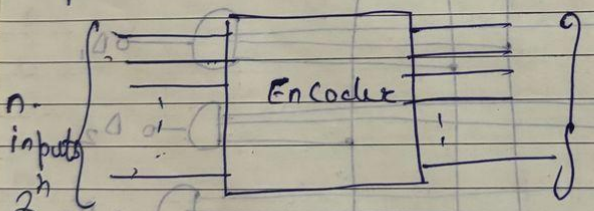
The circuit of a 3 to 8 decoder can be implemented by using 3 NOT gates & eight 3 input AND gates as shown in fig.



Logic Diagram 3-to-8 Decoder

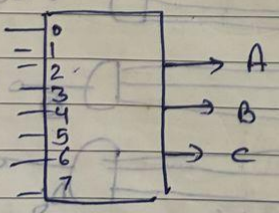
Encoder : An encoder is a digital circuit that performs the inverse operation of a decoder. Hence the opposite of the decoding process is called encoding. An encoder is a combinational logic circuit that converts an active input signal into a coded output signal.

The basic block digram of an encoder is with n-inputs and m-outputs as shown in figure. It has n-input lines, only one of which is active at any time & m output lines. In an encoder the number of outputs is less than the number of inputs.



Block Diagram of Encoder

8 x 3 Encoder



Parity Genera

When digita
to another
receiving
of error.
parity bit
transmitte

There are
even parity

The even
number
means
1's fo

Parity

exclusive
Because
has odd
exclusive
parity

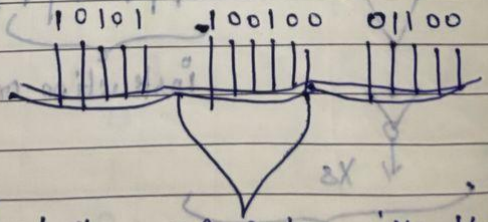
Parity Generator / Checker

When digital data is transmitted from one location to another, it is necessary to know at the receiving end whether the received data is free of error. This addition bit is known as the parity bit & it decides whether the data is transmitted is error free or not.

There are two types of parity bits, namely even parity and Odd parity.

The even parity means an n-bit input has an even number of 1's for ex 11001. Odd parity means an n-bit input has an Odd Number of 1's for ex 10001.

Parity Generator :- For checking the parity of a Binary Number we use the exclusive OR gates. These are the ideal checking purpose Because they produce an Output 1 when the input has odd Number of 1's. The even parity input to an exclusive OR gate produces a low output while odd parity input produces a high output.

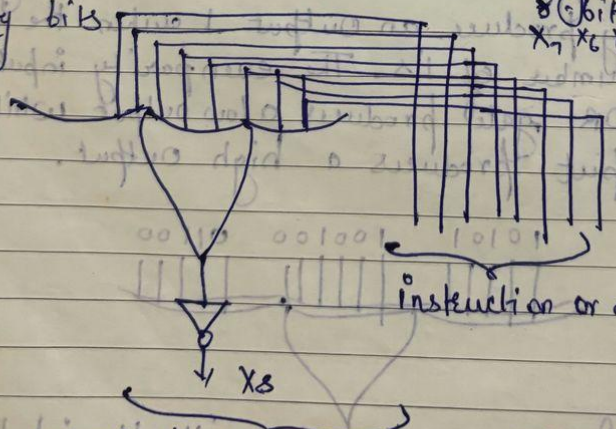


Exclusive OR gate with 16-inputs

Parity Generator :

A Binary Number may represent an instruction that tells the Computer to add, subtract and so on. The binary number may also represent data to be processed like a number, letter, etc., in a Computer. In either case, an extra bit is added to the original binary Number to produce a binary Number with even or odd parity. Such an extra bit can be easily generated using an Ex-nor gate.

Consider the odd parity generator using an Ex-OR gate and a NOT gate shown in fig. Let the 8-bit Number $X_7, X_6, X_5, X_4, X_3, X_2, X_1, X_0$ be equal to 01000001. This number has even parity. It means that, when it is applied to an exclusive-OR gate it will produce an output of 0. Because the inverted $X_8 = 1$ and the final 9-bit output is 101000001, it has odd parity. Thus, an Ex-NOR can be used to generate parity bits.



9-bit Number with odd parity

Odd - Parity Generator

Odd Convert

Magnitude

A magnitude circuit tha numbers in binary Num the ... o design a ... two inputs have three Cond?, one Cond?

A

B

1- Bit Ma

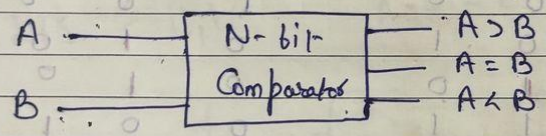
A compar called a two input and three & greate

A	B
0	0
0	1
1	0
1	1

Code Converter

Magnitude Comparator

A magnitude digital Comparator is a Combinational circuit that Compares two digital or Binary numbers in order to find out whether one binary Number is equal, less than or greater the other Binary Number. We logically design a Circuit for which one must have two inputs one for A and other for B and have three output terminals, One for A > B Cond?, one for A = B Condition and one for A < B Cond?



1- Bit Magnitude Comparator

A comparator used to compare two bits is called a Single bit Comparator. It Consists of two inputs each for two Single bit numbers and three Outputs to generate less than equal to & greater than between two binary numbers.

A	B	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

The logic expression for each output can be expressed as follows:

$A > B : AB'$

$A < B : A'B$

$A = B : A'B' + AB$

Example 2- bit Comparator

A_1	A_0	B_1	B_0	$A > B$	$A = B$	$A < B$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

$B < A$	$B = A$	$B > A$	B	A
0	1	0	0	0
0	0	1	1	0
1	0	0	0	1
0	1	0	1	1