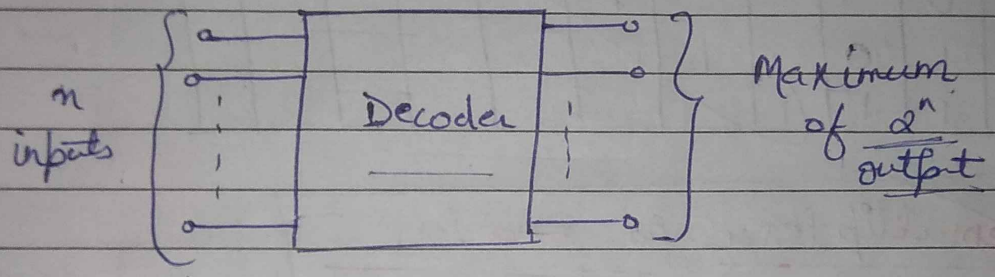
Page
Date

Decoders and Encoders

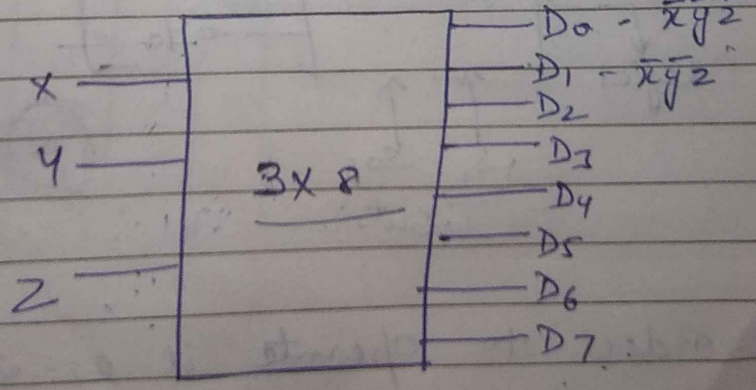
Decoder is a combinational circuit that converts n lines of input to 2^n lines of output. It is a multiple input and multiple output device.

Input – n
Output – 2^n



Applications of decoder are – $n \times 2^n$
i) Binary to octal – 3×8
ii) Binary to Hexadecimal – 4×16
iii) Binary to decimal – 4×10

Ex- If we have to design 3×8 decoder



$2^n \rightarrow 16$

$n \rightarrow 4$

Page
Date

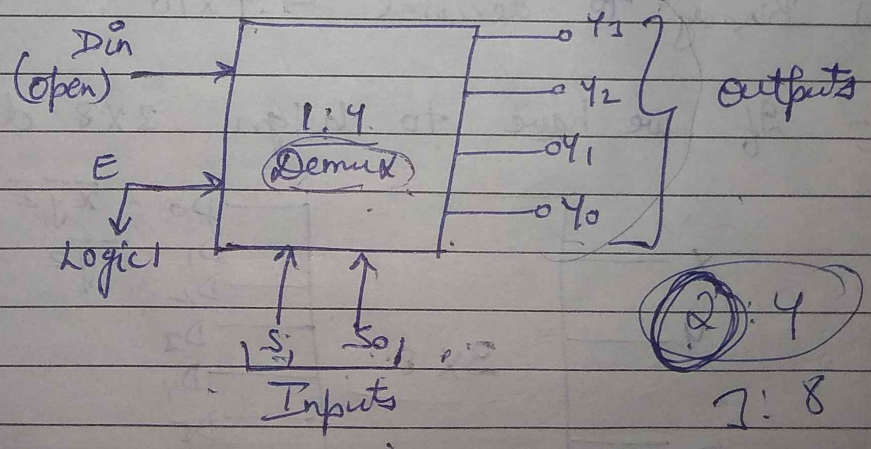
Truth Table

x	y	z	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0					1	1		
0	1	1					1			
1	0	0				1				
1	0	1			1					
1	1	0		1						
1	1	1	1							

Demultiplexer as Decoders

→ We can use a demultiplexer as a decoder.

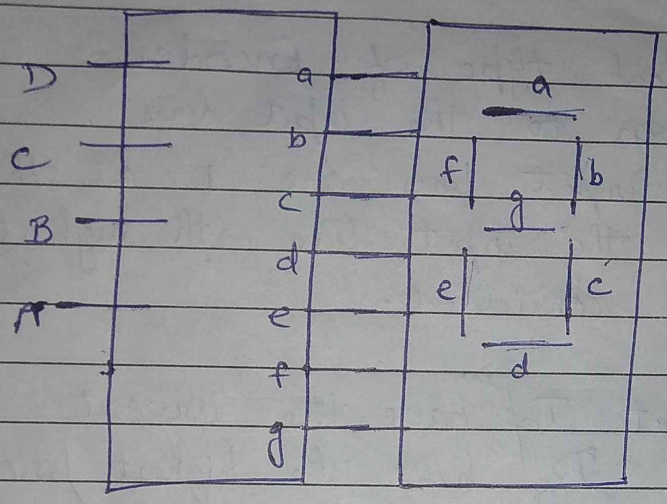
→ We will be using 1:4 Demux as 2:4 decoder.



→ In order to operate it as 2:4 decoder, we have to use S_1 S_0 as inputs, keep Din Open and use Y_3 to Y_0 as outputs.

Page
Date

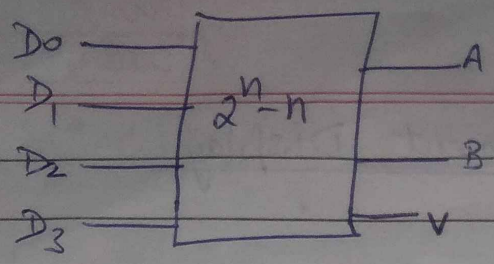
BCD to Seven-Segment Display



LED display

D.N.	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	1	1					
2	0	0	1	0							
3	0	0	1	1							
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0							
9	1	0	0	1							

Priority Encoders



→ This is a special type of Encoder
→ Priorities are given to the input lines. If two or more input lines are '1' at the same time, then the input line with highest priority will be considered.

→ We assume that I_0 / D_0 have the lowest priority and D_3 have the highest priority

Truth Table

D_3	I_2	D_1	I_0 / D_0	Y_1	Y_0
0	0	0	0	x	x
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

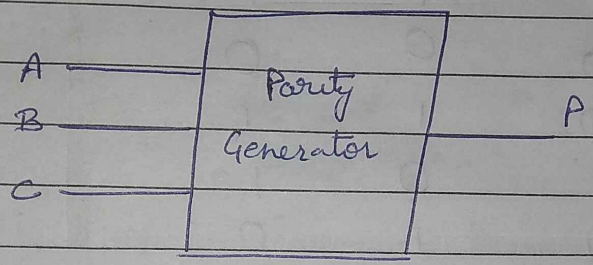
Parity Generator

0 0 → I_0
0 1 → I_1
1 0 → I_2 ✓
1 1 → I_3

Page
Date

Parity Generator

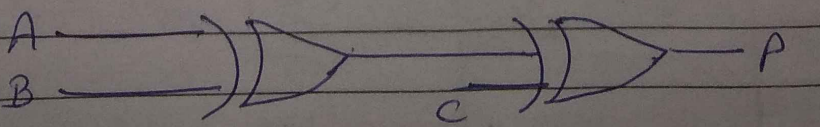
We already know about
- Even Parity
- Odd Parity



Even Parity Generator

A	B	C	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

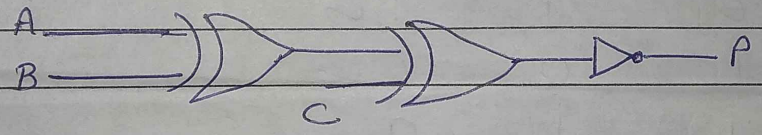
$$P = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$$
$$= \bar{A}(B \oplus C) + A(\overline{B \oplus C})$$
$$P = A \oplus B \oplus C$$



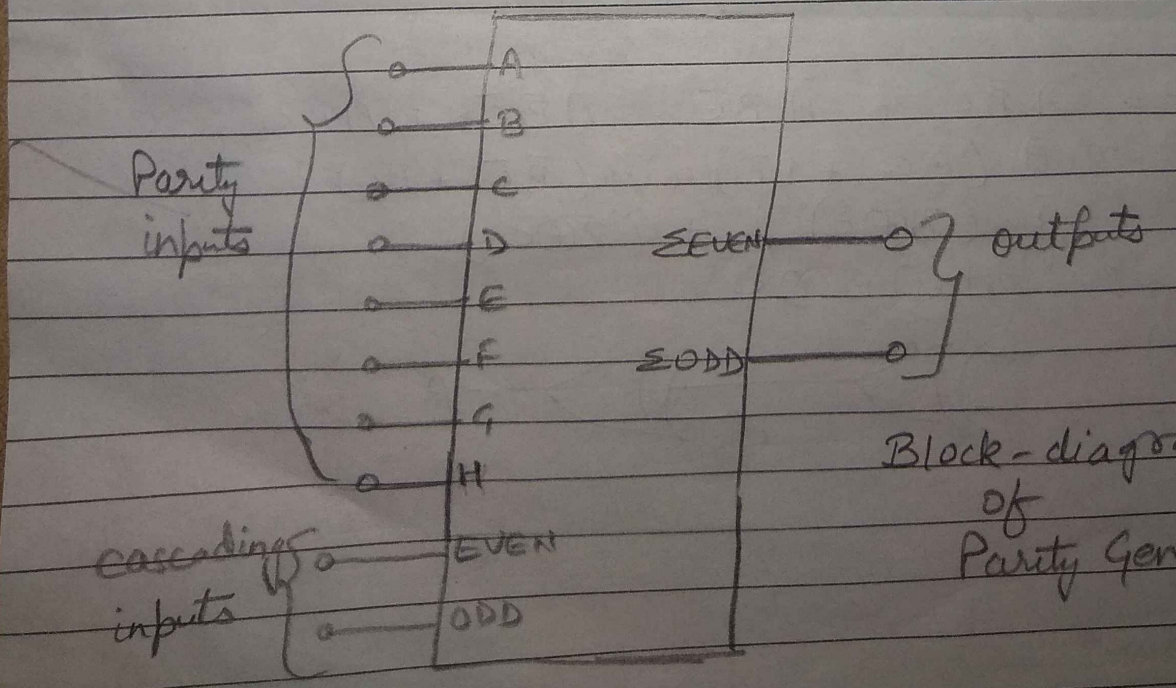
Odd - Parity Generator

A	B	C	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$$P = \overline{A \oplus B \oplus C}$$



Block Diagram

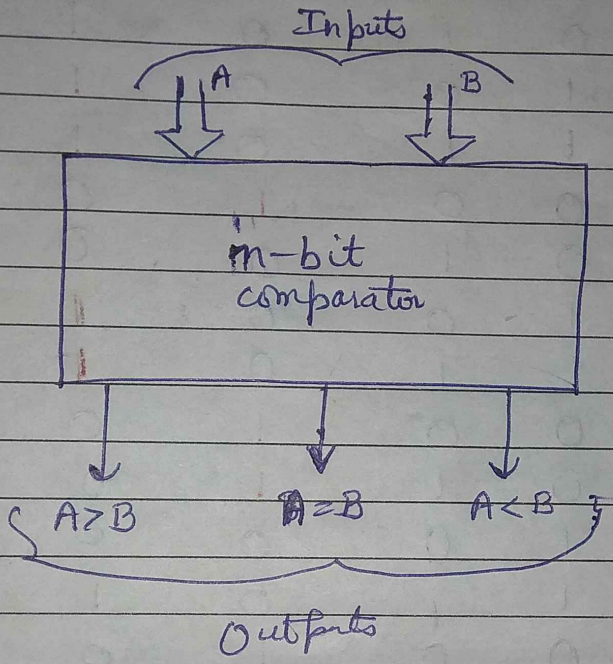


Block-diagram of Parity Generator

Page
Date

Digital comparators

Comparators can be designed for comparing multibit numbers:



A	B	Y
0	0	A=B
0	1	A<B
1	0	A>B
1	1	A=B

$A<B = \bar{A}B$

$A=B = \bar{A}\bar{B} + AB$

$A>B = A\bar{B}$

Now, a 2-bit digital comparator

$A_1, A_0 \rightarrow$ 00, 01, 10, 11

$B_1, B_0 \rightarrow$ 00, 01, 10, 11

A		B		Y
A_1	A_0	B_1	B_0	
0	0	0	0	

P.T.O

Page
Date

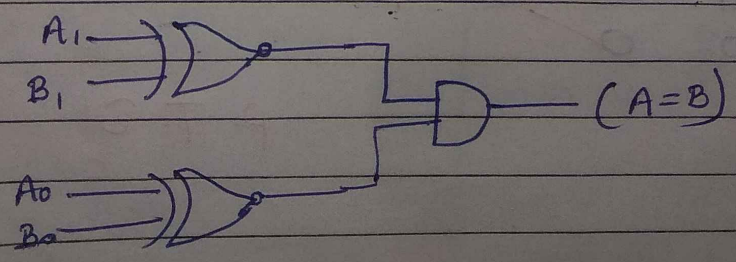
Truth Table of 2-bit comparator

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

Thus, for a n-bit comparator —
We have
$2n$ variables
and 2^{2n} rows.

For,

$$A=B = (A_1 \odot B_1)(A_0 \odot B_0)$$



Page
Date

For $A > B \Rightarrow A_1 > B_1$ or
if $A_1 = B_1$ and $A_0 > B_0$

Case 1
$A = 1\ 0$
$B = 0\ 1$

Case 2:
$A = 1\ 0$
$B = 1\ 0$

For $A < B \rightarrow A_1 < B_1$ or
if $A_1 = B_1$ and $A_0 < B_0$

Case 1
$A = 0\ 1$
$B = 1\ 1$

Case 2
$A = 1\ 0$
$B = 1\ 1$

Page
Date

Code Converters

There is a wide variety of binary codes used in digital systems such as BCD, excess-3, gray, octal, hexadecimal etc.

Often it is required to convert from one code to another.

Ex - 4 bit binary ⟶ BCD conversion

TRUTH TABLE

Binary No.				Decimal	BCD				
B_3	B_2	B_1	B_0		D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	1
0	0	1	0	2	0	0	0	1	0
0	0	1	1	3	0	0	0	1	1
0	1	0	0	4	0	0	1	0	0
0	1	0	1	5	0	0	1	0	1
0	1	1	0	6	0	0	1	1	0
0	1	1	1	7	0	0	1	1	1
1	0	0	0	8	0	1	0	0	0
1	0	0	1	9	0	1	0	0	1
1	0	1	0	10	1	0	0	0	0
1	0	1	1	11	1	0	0	0	1
1	1	0	0	12	1	0	0	1	0
1	1	0	1	13	1	0	0	1	1
1	1	1	0	14	1	0	1	0	0
1	1	1	1	15	1	0	1	0	1

Page
Date

Boolean Equation – Solving for D_0, D_1, D_2, D_3 & D_4 by using K-Map –

$$D_0 = B_0$$

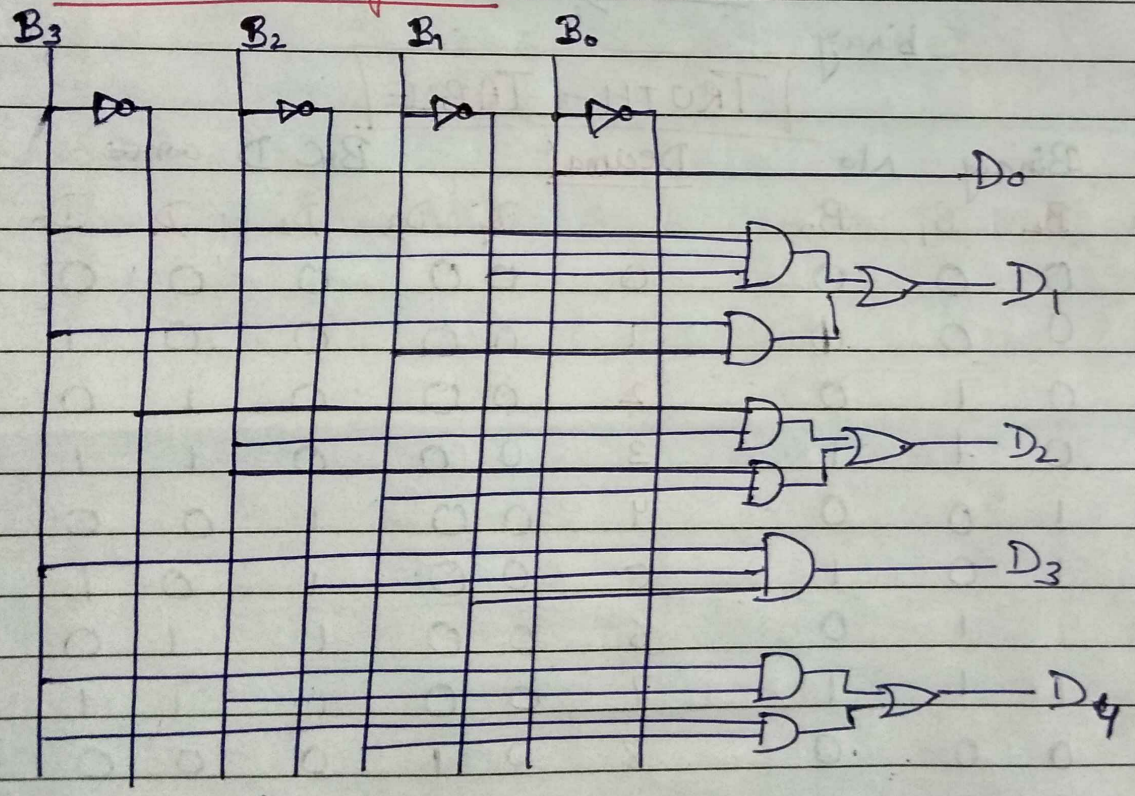
$$D_1 = B_3 B_2 \overline{B_1} + B_3 B_1$$

$$D_2 = \overline{B_3} B_2 + B_2 B_1$$

$$D_3 = B_3 \overline{B_2} \overline{B_1}$$

$$D_4 = B_3 B_2 + B_3 B_1$$

Circuit diagram



Page
Date

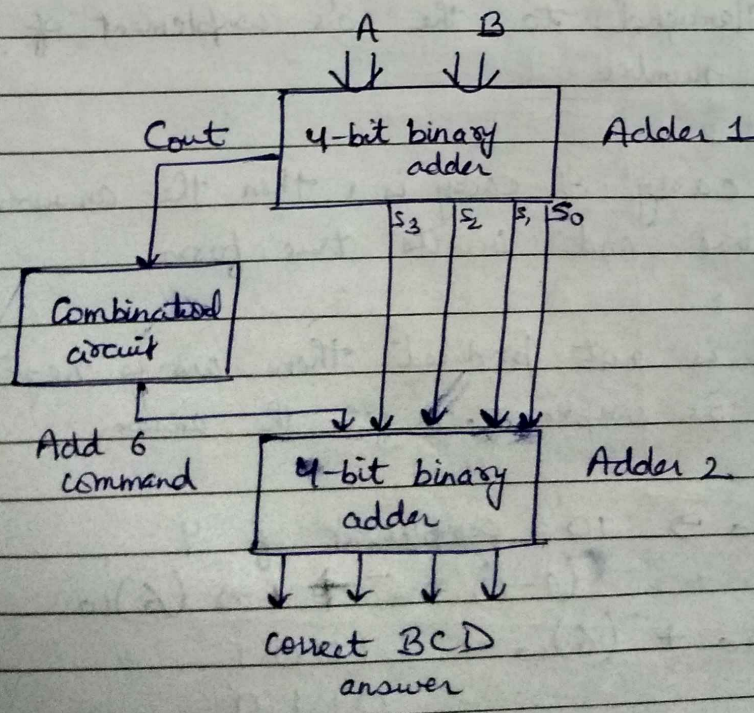
, D_3 & D_4

BCD Arithmetic

i) BCD Addition

- BCD adder adds two BCD digits and produces a BCD digit.
- But, a BCD digit can't be greater than 9.
- If sum is less than or equal to 9 and carry = 0, then no correction is necessary. The sum obtained is correct and in the true BCD form.
- But if sum is invalid BCD or carry = 1, then result is wrong and needs correction.
- The wrong result can be corrected by adding six i.e (0110) to it.

Block Diagram of BCD Adder



ii) BCD Subtractor

BCD subtraction can be performed using 2 methods →
i) Using 9's complement
ii) Using 10's complement

→ 9's complement of a BCD number is given by 9 minus that number

Ex-

9's complement of 7 → 0111 is
9-7=2 ⇒ 0010

→ 10's complement of a BCD no.
⇒ 9's complement of the number + 1

BCD Subtraction using 10's complement

Step 1 → Obtain the 10's complement of 'number to be subtracted'.

Step 2 → Add Maruend to the 10's complement of the number.

Step 3 → Discard carry. If carry is 1 then the answer is positive and in its true form.

Step 4 → If carry is not produced then ans. is negative. So, take 10's complement to get the answer.

Ex- $(9)_{10}-(4)_{10}$ → 10's complement of 4
⇒ (9-4) = 5 → 1 ⇒ $(6)_{10}$
→ $(9)_{10}+(6)_{10}$ ⇒ 1001
0110
1111 ← Invalid BCD & carry = 0

Add $(0110)_{10}$ for correction

Page
Date

Adding $(0110)_{10}$ for correction

⇒

```
  1 1 1 1
  0 1 1 0
-------------
1 0 0 0 1
(5)
```

Discard the carry

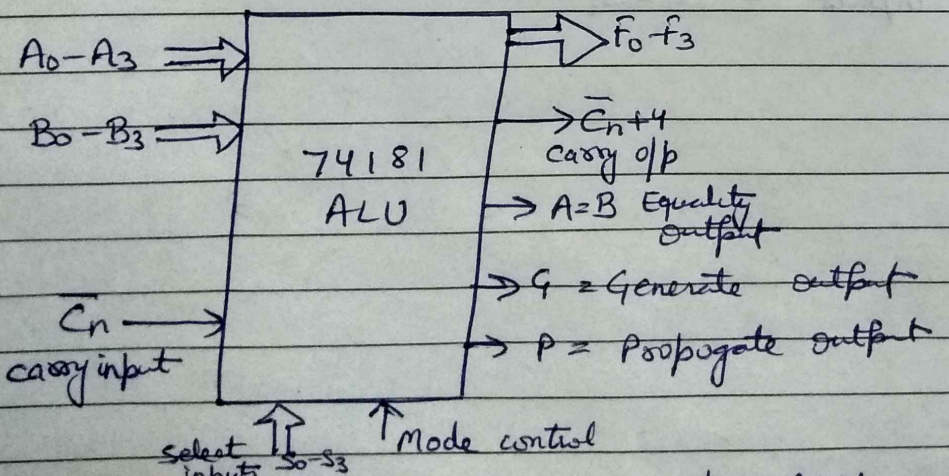
Answer is $(5)_{10}$

Arithmetic Logic Unit (ALU)

→ ALU is a combinational circuit which performs arithmetic and logical operation.

→ Arithmetic operation includes - Addition, Subtraction, increment, decrement etc.
Logical operation - NOT, NAND, OR, AND, NOR, XOR

→ ALU is the most important part of processor.



Block diagram of ALU

→ $A_0 - A_3$ and $B_0 - B_3$ are the two 4-bit numbers as inputs

→ $\overline{C_n}$ → carry input

If $\overline{C_n} = 1$ means there is no carry.

$\overline{C_n} = 0$ means there is carry

→ $F_0 - F_3$ ⇒ 4-bit o/p data

→ $\overline{C}_{n+4}$ → Carry o/p

If we do the addition and we get one extra bit as o/p, then that bit will be shown over here.

→ A = B Equality o/p

If A = B, then Equality o/p = 1

→ G and P pins are used for cascade mode. These pins are used when 2 or more ALUs are used.

→ Mode control, M

M = 0 → arithmetic operation

M = 1 → Logical operation

→ Select inputs - $S_0 - S_3$

Page
Date

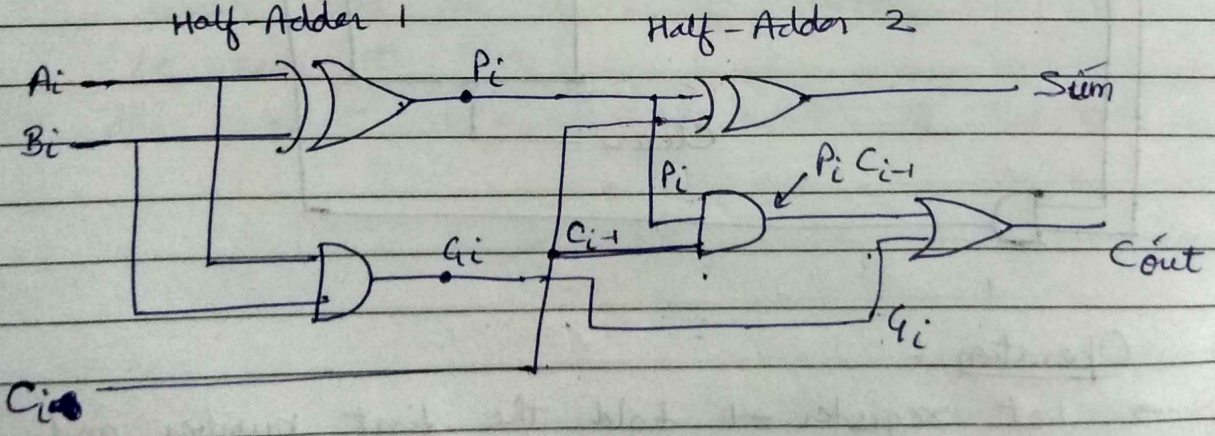
Carry Look Ahead Adder

→ In parallel adders, where n adders are connected in sequency to add the numbers, the carry out of the previous stage is connected to carry in of the next stage.

→ The carry is said to be propogated like ripple, this phenomenon is called ripple carry propogation.

→ Due to this ripple carry propagation time delay is introduced in the addition process. This time delay is called as the propogation delay.

→ The problem of propagation delay can be eliminated by using this addition technique of carry look-ahead adder.



$$P_i = A \oplus B$$
$$G_i = AB$$

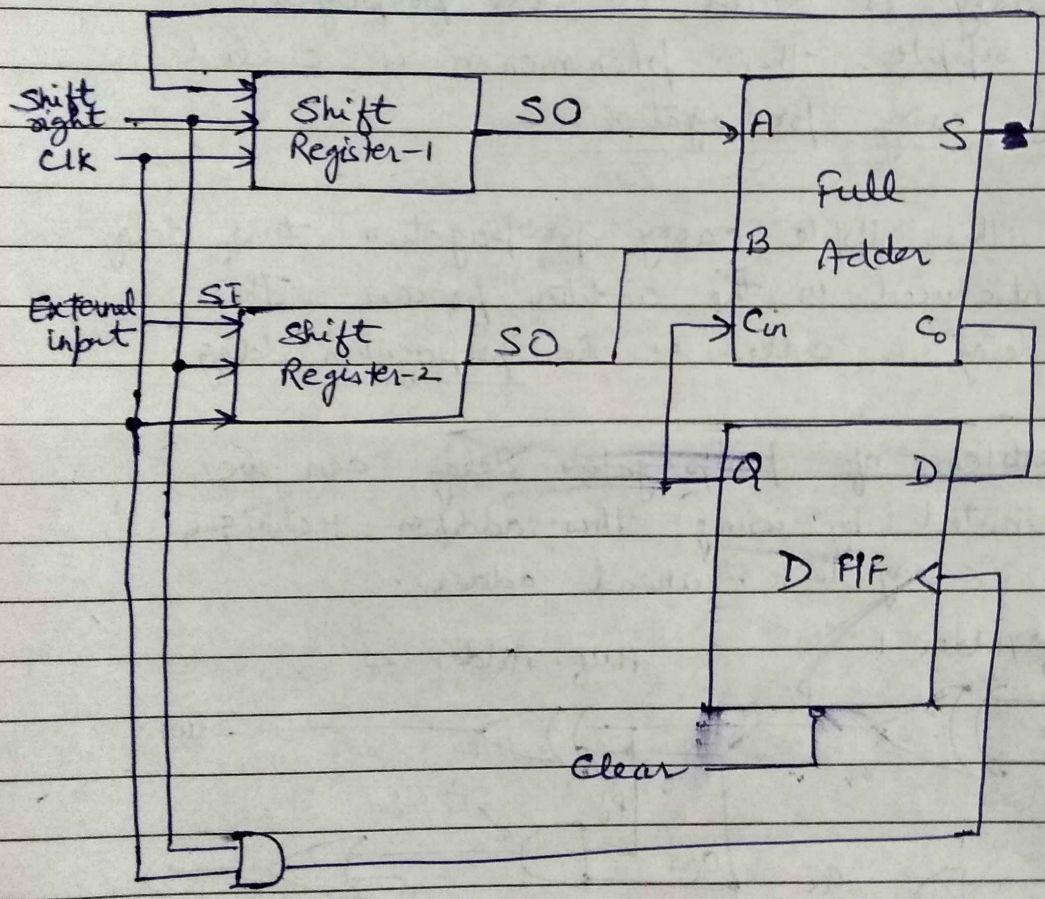
$$S = P_i \oplus C_i$$
$$C_{out} = G_i + P_i C$$

Serial Adder

→ Serial Adder is used to add two n-bit numbers.

→ Unlike the parallel adder, its ~~speed~~ is slow & it requires ~~less equipments~~ more components.

→ It is a sequential circuit means ~~the~~ it is memory-based.



Operation :

→ Let register-1 hold the first number and register-2 hold the other number.

→ The D flip-flop is cleared initially so Q=0 and Cin = 0

Page
Date

→ The serial outputs (SO) of the two registers will provide the LSBs of the two numbers. They will act as bits A and B for the full adder.

→ Full Adder will add these bits & produce sum S and carry out C_o.

→ Now, a clock pulse is applied to both the shift registers. So, the two numbers are right shifted by one bit each.

→ The same operation will continue and we will get the addition of two numbers in a bit by bit manner.

Disadvantages of Serial Adder

i) More time is required to complete the addition
ii) A complicated circuit is required.
iii) The sum & carry are available in the serial form so result of addition cannot be seen at once.
iv) The circuit contains more number of components.