

6x8x3x13

K-Map

- In K Map representation, successive cell will differ by only one bit.
- In this gray code representation is used.
- Two variable

		LSB \rightarrow B	0	1
		MSB \rightarrow A	0	1
0	0	00	01	
	1	10	11	

Three variable

		LSB	
		BC	A
0	0	000	001
	1	100	101
1	0	111	110
	1	110	111

four variable

		CP	
		AB	00 01 11 10
00	00	0	1 3 2
	01	4	5 7 6
01	12	13	15 14
	11	8	9 10
10	8	9	11 10
	10	12	13 14

Some times Redundant group is taken to avoid the HAZARDS.

* Group of adj. cells and each cell is represented with Group of literals.

Pairs 1 Quad 2 Octet 3

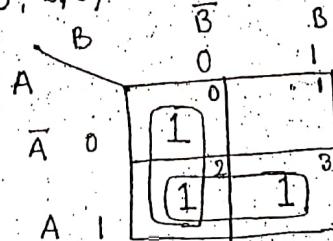
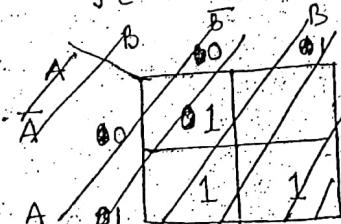
* In a 6 variable K map Octet give $5 \cdot 3 = 2$ literals

* If all cells of a K map are 1 then the K map is represent as 1 or K=1. It indicates that C.R. not required.

SOP form

(i) Two variable

$$f(A, B) = \sum m(0, 2, 3)$$



$$\bar{A}C + BC + AB$$

Without BC also it is complete hence without BC also it is complete.

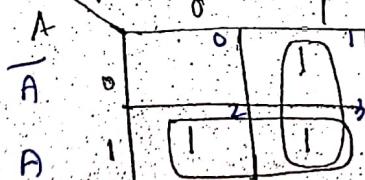
So

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

$$\bar{B}(A + \bar{A}) + A(B + \bar{B})$$

(ii) $f(AB) = \sum m(1, 2, 3)$

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



$$Y = A + B$$

Ques $f(A, B) = \sum m(0, 2) + \sum d(3)$

A	B	0	
0	0	1	
1	1	X	

$$Y = \overline{B} + A$$

→ don't care are used if it is proving minimized expression.

Ques $f(A, B) = \sum m(0, 3) + \sum d(1)$

A	B	0	1	
0	0	1	X	
1	1	X	1	

$$Y = \overline{A} + B$$

Ques $f(A, B) = \sum m(1, 3) + \sum d(0, 2)$

A	B	0	
0	0	X	1
1	1	1	X

$$Y = 1$$

also

Y	Y	
X	X	

$$Y = 0$$

Three Variables

$$f(A, B, C) = \sum m(0, 1, 3, 5, 7)$$

A	BC	00	01	11	10
0	0	1	1	1	1
1	1	1	1	X	1

$$Y = \overline{A} \cdot \overline{B}$$

$$\overline{C} [A \oplus B]$$

$$B(A \oplus C)$$

A	BC	$\overline{B}C$	$\overline{B}C$	$B\overline{C}$
0	00	1	1	1
1	01	1	1	1

$$\overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{C} + A \cdot \overline{C}$$

Ques $f(A, B, C) = \sum m(0, 1, 2, 7)$

A	BC	00	01	11	10
0	0	1	1	1	1
1	1	1	1	1	1

$$Y = \overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{C} + A \cdot \overline{B} \cdot \overline{C}$$

$$\begin{aligned} & \overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{C} \\ & + \overline{B} \cdot \overline{C} + A \cdot \overline{B} \cdot \overline{C} \\ & + \overline{A} \cdot B \cdot \overline{C} + A \cdot B \cdot \overline{C} \end{aligned}$$

A	BC	00	01	11	10
0	0	1	1	1	1
1	1	1	1	1	1

$$Y = \overline{A} \cdot \overline{C} + A \cdot B$$

$$\hookrightarrow \text{Implicant } \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C}$$

$$\text{Prime } II \dots = \overline{A} \cdot C + A \cdot B + B \cdot C$$

$$\text{Minterm } II \dots = A \cdot C + A \cdot B$$

Procedure

- ① Octet
 - ② Guard
 - ③ Pairs
 - ④ Single
 - ⑤ Remove redundant terms
- * * * Hazards - Because of different propagation delays of a combination of inputs there exists hazards in the output. Hazards are of two types - static & dynamic.
- The Redundant pairs in combination of inputs will give hazard free output.

Ques $f(A, B, C) = \sum m(0, 1, 5, 6, 7)$

A	BC	00	01	11	10
0	1	1	0	0	0
1	1	0	1	1	0

static 0

static 1

dynamics



$$Y = \bar{A}\bar{B} + A\bar{B} + \bar{B}C$$

$$\text{or } Y = \bar{A}\bar{B} + AB + AC$$

→ K-map will provide minimized expression but not necessarily unique. (K-map not unique necessarily)

Ques

A	BC	$\bar{B}C$	$B\bar{C}$	BC	$B\bar{C}$
0	1	1	0	0	0
1	1	0	1	1	0

$$Y = \bar{B}C + \bar{B}C + \bar{A}\bar{B} + \bar{A}\bar{C}$$



Ques

$$f(A, B, C) = \sum m(0, 1, 3, 7) + \sum d(2, 5)$$

A	BC	$\bar{B}C$	$B\bar{C}$	BC	$B\bar{C}$
0	1	1	1	X	X
1	1	X	1	1	1

$$Y = C + A\bar{C}$$

$$Y = A + C$$

$$- \quad A\bar{B} + ACD + \bar{A}BC + \bar{A}\bar{C}D$$

Ques

$$f(A, B, C) = \sum m(0, 1, 6, 7) + \sum d(3, 4, 5)$$

A	BC	$\bar{B}C$	$B\bar{C}$	BC	$B\bar{C}$
0	1	1	X	1	2
1	4	X	5	1	1

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

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

Ques

$$f(A, BC) = \sum m(0, 1, 6, 7) + \sum d(3, 4, 5)$$

A	BC	$\bar{B}C$	$B\bar{C}$	BC	$B\bar{C}$
0	1	1	X	X	1
1	X	X	1	1	1

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

four variable

$$\text{Ques } f(A, B, C, D) = \sum m(0, 1, 3, 5, 7, 8, 9, 11, 13, 15)$$

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$A\bar{B}$	1	1	1		
$\bar{A}B$		1	1	1	
AB			1	1	
$A\bar{B}$	1	1	1		

$$Y = D + \bar{C}\bar{B}$$

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$A\bar{B}$	1	1	1		
$\bar{A}B$		1	1	1	
AB			1	1	
$A\bar{B}$	1	1	1		

$B0 [AO]$

$AC(C\bar{B}\bar{D})$

$$\text{Ques } f(A, B, C, D) = \sum m'(0, 1, 4, 5, 8, 9, 13, 15)$$

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$A\bar{B}$	1	1	1		
$\bar{A}B$		1	1	1	
AB			1	1	
$A\bar{B}$	1	1	1		

$$Y = \bar{C}\bar{D} + \bar{C}\bar{B} + ABD$$

$$Y = \bar{C}\bar{B} + \bar{C}\bar{A} + ABD$$

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$A\bar{B}$	1	1	1		
$\bar{A}B$		1	1	1	
AB			1	1	
$A\bar{B}$	1	1	1		

$\bar{A}B (C \oplus D)$

$AB (C \odot D)$

P.O.S form

Rules are same.

$$\text{Ques } f(A, B) = \prod M(0, 2, 3)$$

A	B	B_0	\bar{B}_1
0	0	0	1
1	0	0	0

$$Y = B \oplus A$$

$$\text{Ques 2 } f(A, B) = \prod M(0, 1) + \prod d(8)$$

A	B	B_0	\bar{B}_1
0	0	0	1
1	0	X	1

if not used

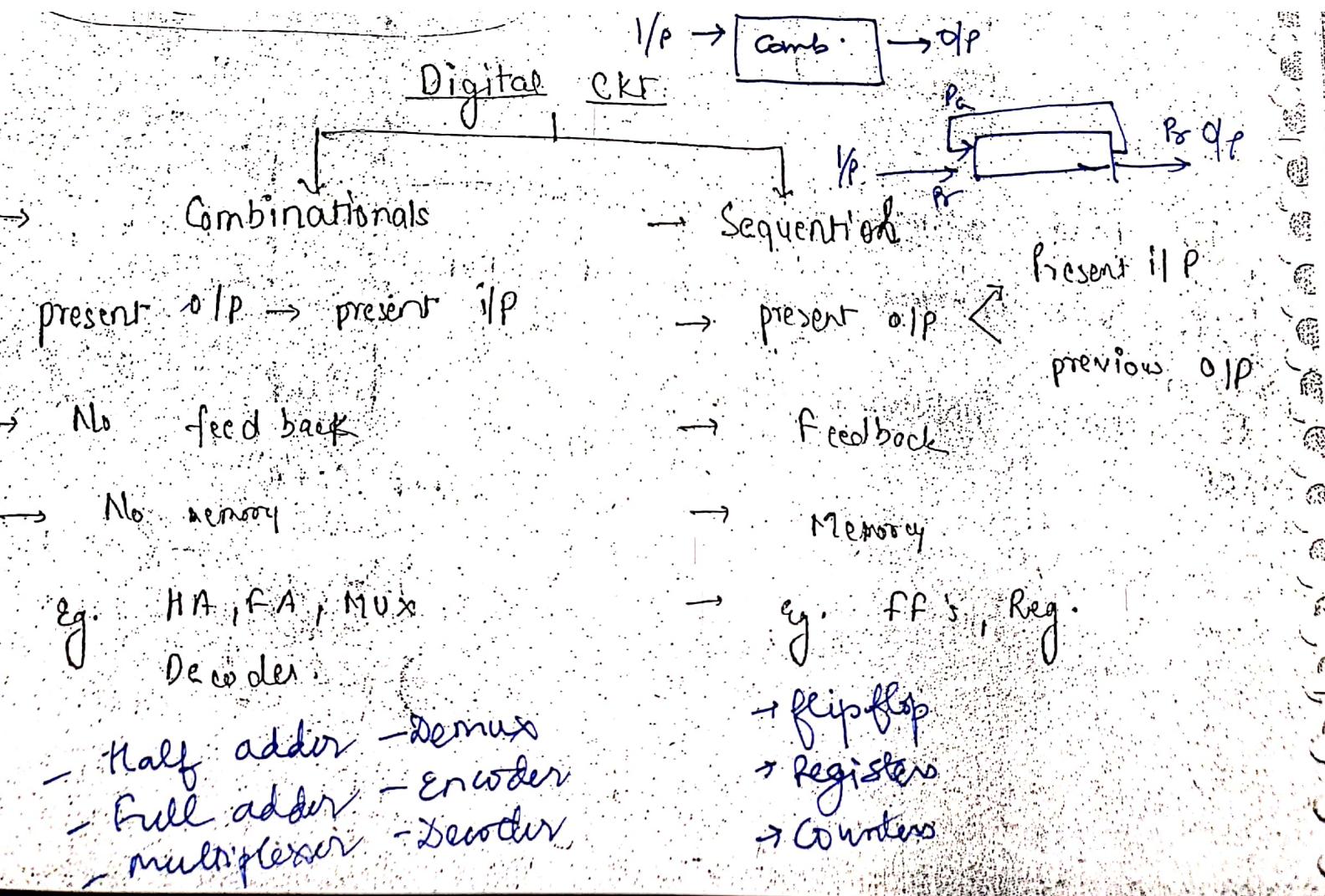
$$Y = A$$

$$\text{Ques } f(A, B) = \prod M(0, 3) + \prod d(1)$$

A	B	B_0	\bar{B}_1
0	0	0	1
1	0	X	0

$$Y = A \cdot \bar{B}$$

value of don't care is 0, we are using it]



Combinational Ckt

- ① Identify inputs and output
- ② Construct truth table
- ③ Write logical expression .. in SOP / POS
- ④ Minimize
- ⑤ Implement the ckt.

Arithmetic cks

① Half adder

① Identify



② Truth table

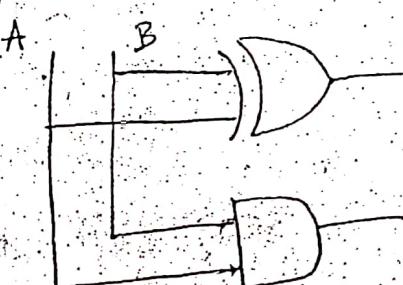
A	B	Sum Out	Carry Out	Sum Out	Carry Out
0	0	0	0	0	0
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	1	0	1

③ Logical Exp.

$$\text{for Sum} \dots = \bar{A}B + A\bar{B} = A \oplus B$$

$$\text{for Carry} \dots = AB$$

④ Logical ckt



$$\text{Sum} = A \oplus B$$

[EX-OR]

$$\text{Carry} = A \cdot B$$

[AND]

B	0	1
A	0	1

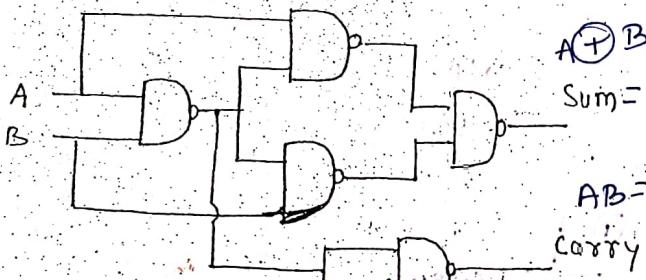
$$\bar{A}B + A\bar{B}$$

B	0	1
A	0	1

11.28

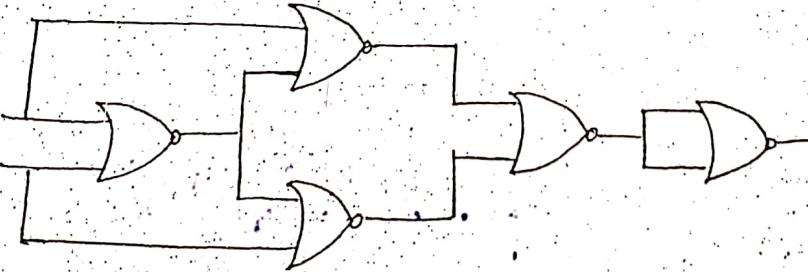
- i) logical exp. for sum = $A \oplus B$
- ii) logical exp. for carry = AB
- iii) minimum no. of NAND = 5
- iv) minimum no. of NOR = 5
- v) No. of MUX
- vi) No. of DeMUX

NAND GATE

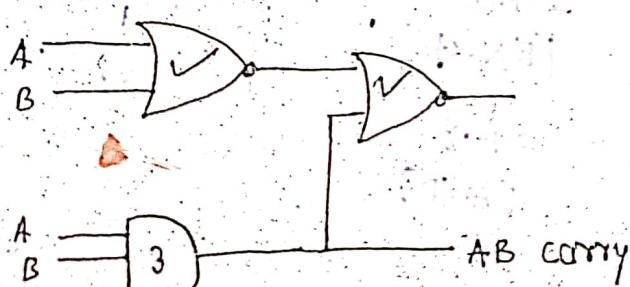


= 5 nand gates

NOR GATE require



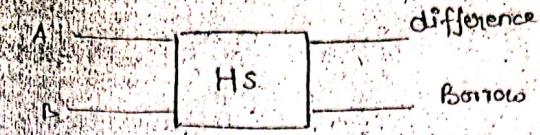
for minimum



= 5 nor gates.

Half Subtractor

i) Identify



ii) Truth Table

A	B	Diff.	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$\begin{array}{r} \textcircled{1} \\ \begin{array}{r} 10 \\ -1 \\ \hline 1 \end{array} \\ 00 \end{array}$$

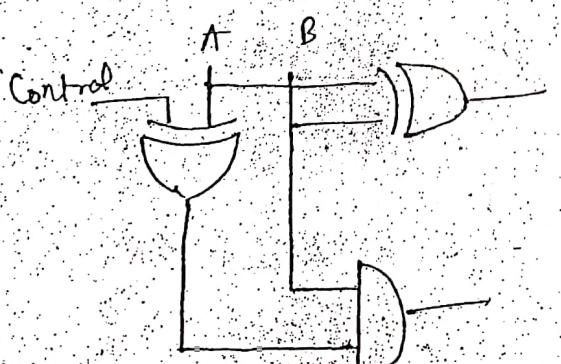
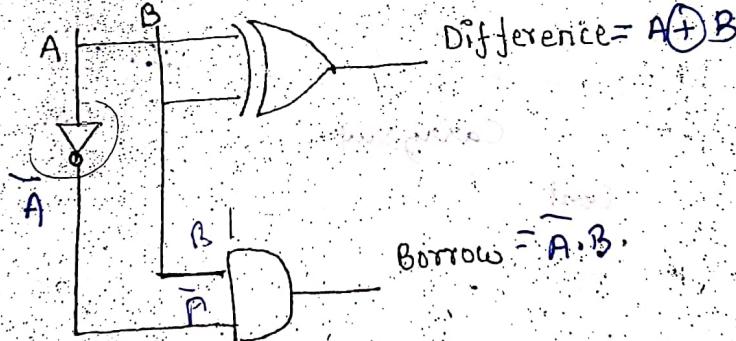
$$0 - 1 = 1 \quad 1$$

iii) Logical exp.

$$\text{Diff.} = \overline{A}B + A\overline{B} = A \oplus B$$

$$\text{Borrow} = \overline{A}B$$

iv) Logical ckt.



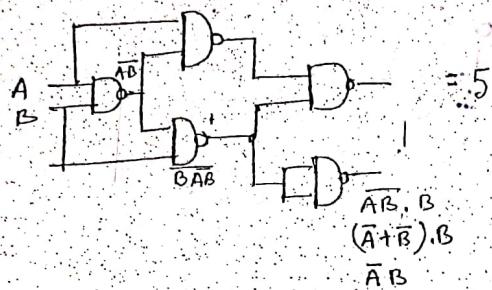
Control = 0	\rightarrow	HA
Control = 1	\rightarrow	HS

HS 80

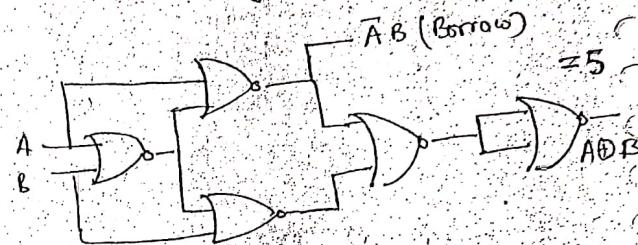
-) Logical exp. Diff = $A \oplus B$
-) Logical exp. of Borrow = $\bar{A}B$
- ③ minimum no. of NAND = 5
- ④ " " NOR = 5

- ⑤ No. of MUX
- ⑥ " " Decoder

HS using NAND

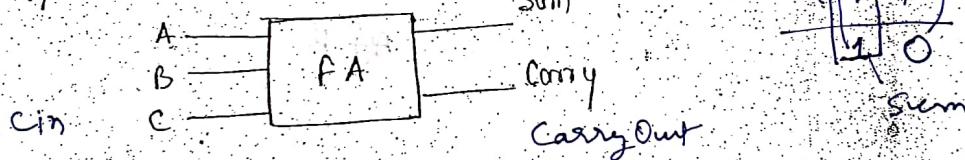


HS using NOR



Full Adder

Identify



Truth table

	A	B	Cin	Sum	Carry
0	0	0	0	0	0
1 →	0	0	1	1	0
2 →	0	1	0	1	0
3 →	0	1	1	0	1
4 →	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7 →	1	1	1	1	1

$$\text{Sum} = f(A, B, C) = \Sigma m(1, 2, 4, 7)$$

$$1 + 1 = 10 \\ 10 + 1 = 11$$

$$\text{Carry} = f(A, B, C) = \Sigma m(3, 5, 6, 7)$$

22

$$\text{Sum} = A \oplus B \oplus C$$

$$= \Sigma m(1, 2, 4, 7)$$

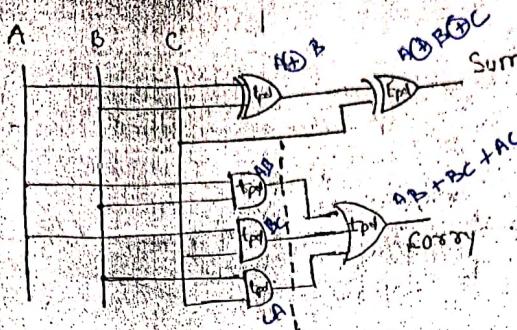
$$\text{Carry}' = \bar{A}BC + A\bar{B}C + ABC$$

$$+ ABC$$

$$= \Sigma m(3, 5, 6, 7)$$

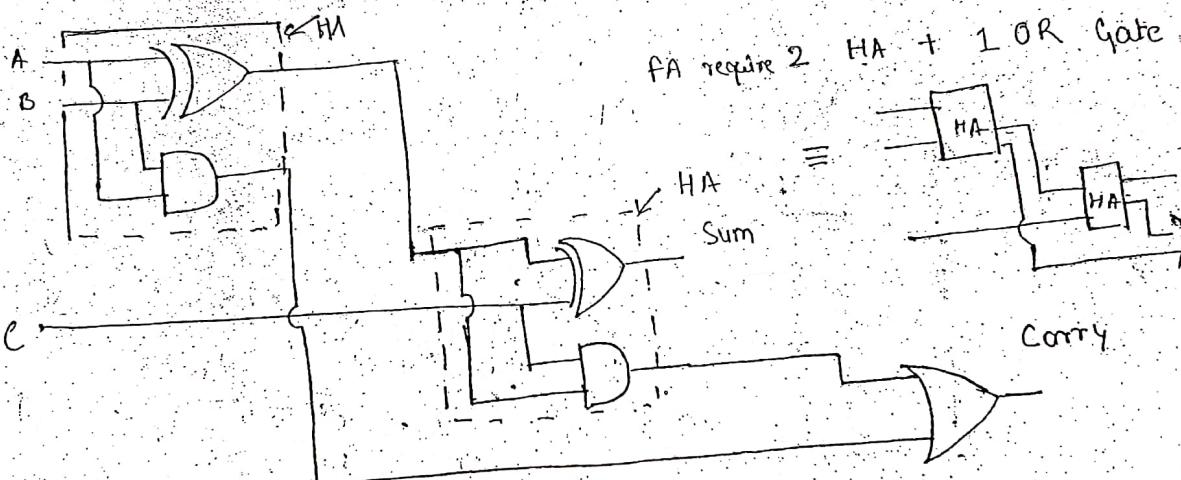
$$\text{Carry} = AB + BC + AC$$

Implementation



In FA to provide sum or carry output, two logic gates' delays require 2 tpd.

$$\begin{aligned} \text{also } \text{Carry} &= \overline{A}BC + A\overline{B}C + A\overline{B}\overline{C} + ABC \\ &= AB + C(\overline{A}B + A\overline{B}) \\ &= AB + C(A \oplus B) \end{aligned}$$

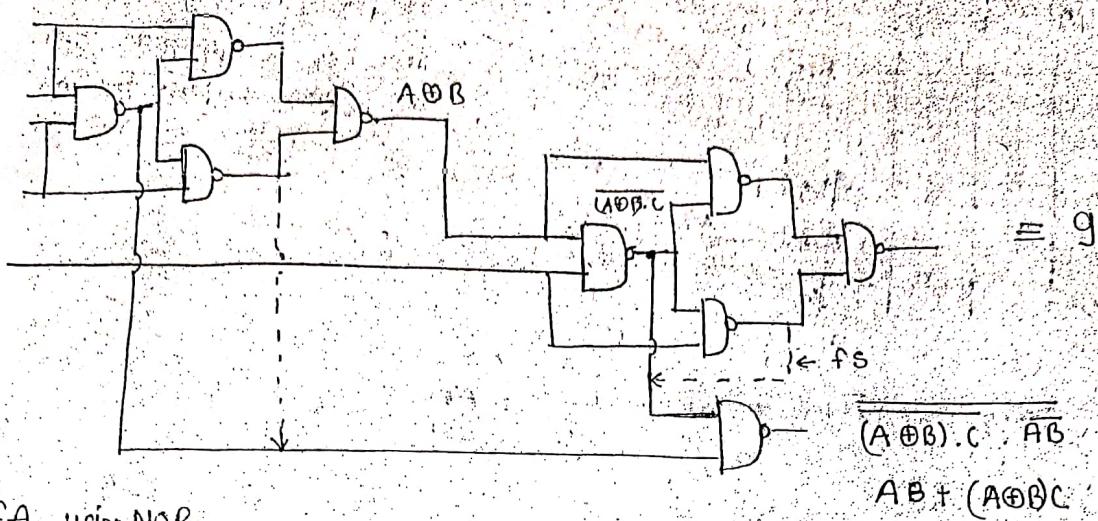


FA

- (0) ~~Sum~~ Sum = $A \oplus B \oplus C$
- (1) ~~Carry~~ Carry = $AB + BC + AC$
- (2) No of HA and OR = 2, OR = 1
- (3) NAND = 9
- (4) NOR = 9
- (5) MUX
- (6) Decoders

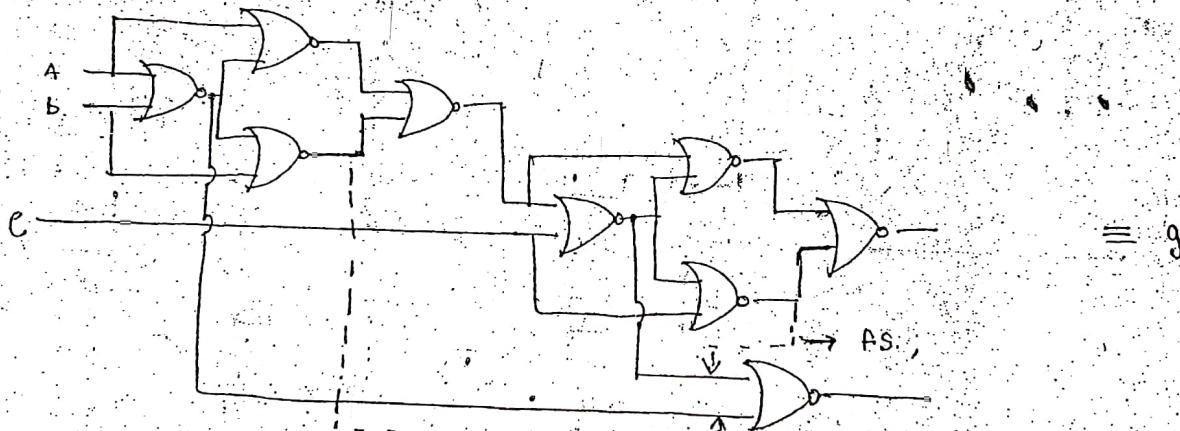
receiving NS $\Rightarrow 2^N, 6^N$

FA using NAND



FA using NOR

Just replace each NAND gate by NOR

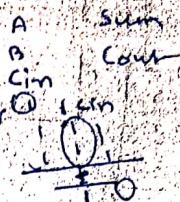


→ HA → 5
→ HS → 5
→ FA → 9
→ FS → 9

Full subtraction



Diff
Borrow



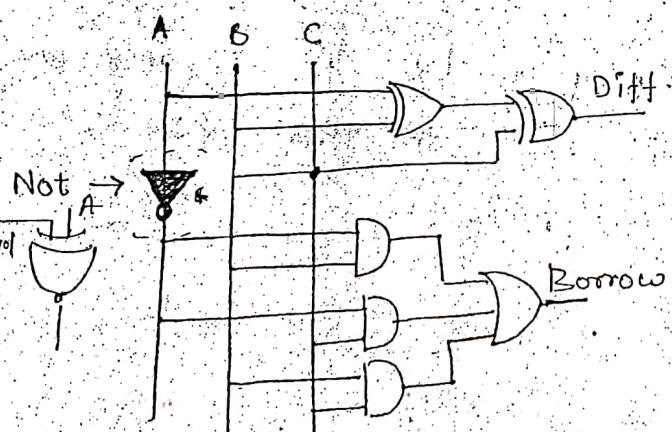
Truth table

A	B	C	(A-B)-C	Diff	Borrow
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	1	1	1

$$\text{Diff} = \sum m(1, 2, 4, 7)$$

$$\text{Borrow} = \sum m(1, 2, 3, 7)$$

Ckt



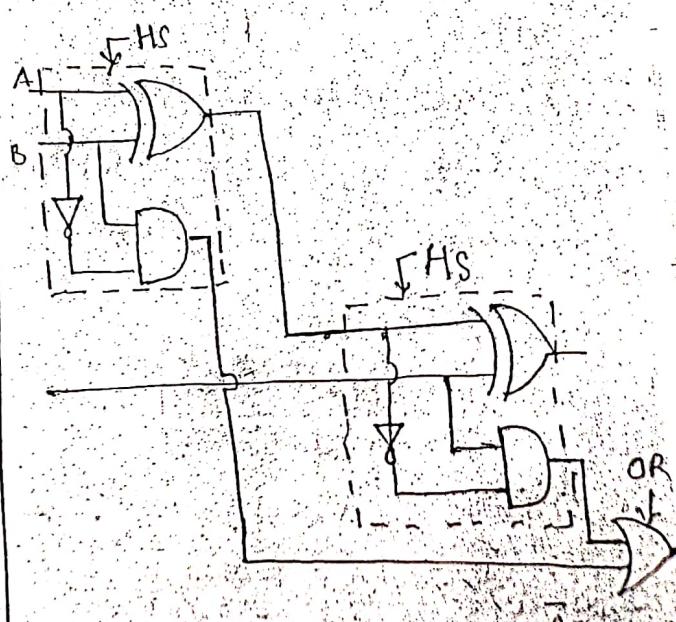
Control 0 \rightarrow FA

Control 1 \rightarrow FS

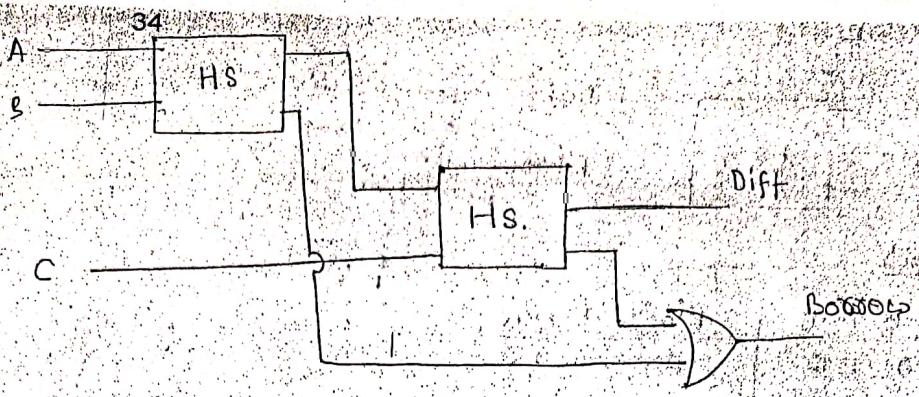
$$\text{also } \text{Borrow} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$$

$$= \bar{A}B + (\bar{A}\bar{B} + AB)C$$

$$= \bar{A}B + (A \oplus B)C$$



$$\bar{A}B + (A \oplus B)$$



$HA / HS \rightarrow$ Sum / Diff $\rightarrow A \oplus B$ ⑤
 \rightarrow carry / borrow
 $AB \quad \bar{A}B$

$FA / FS \rightarrow$ Diff / sum $\rightarrow A \oplus B \oplus C$ ⑨
 \rightarrow carry / Borrow

$$\begin{array}{c|c} AB + [A \oplus B]C & \bar{A}B + [A \oplus B]C \\ AB + BC + AC & AB + \bar{A}C + B\bar{C} \end{array}$$

Group Adder

- 1) Serial adder (Slowest)
 - 2) Parallel adder ($2^{n-1}pd$ delay)
 - 3) Look ahead adder (Fastest)
- Serial adder is used to add group of bits, in this only one

full adder is used to add the bits. In Serial adder to add n bits it require n clock pulse. It is the slowest adder.

Parallel adder - Used to add group of bits. In this to add

4 bit

n bits if required:

3 FA and 1 HA

$\xrightarrow{*} (n-1) FA$ and one HA

4 FA

or

7 HA and 3 OR

n FA

or

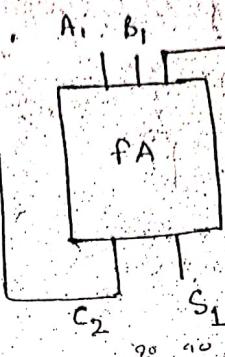
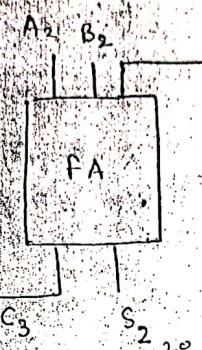
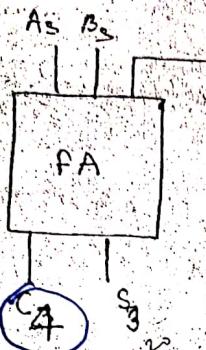
$\xrightarrow{*} (2n-1) HA$ and $(n-1) OR$ gates.



4 bit parallel adder using 4 FA

$$\begin{array}{cccc} A_3 & A_2 & A_1 & A_0 \\ \hline B_3 & B_2 & B_1 & B_0 \end{array} \quad C_0$$

→ Parallel adder is also known as Ripple carry adder



Ripple carry adder

Actual carry → $C_4 \quad [S_3 \ S_2 \ S_1 \ S_0]$ Sum

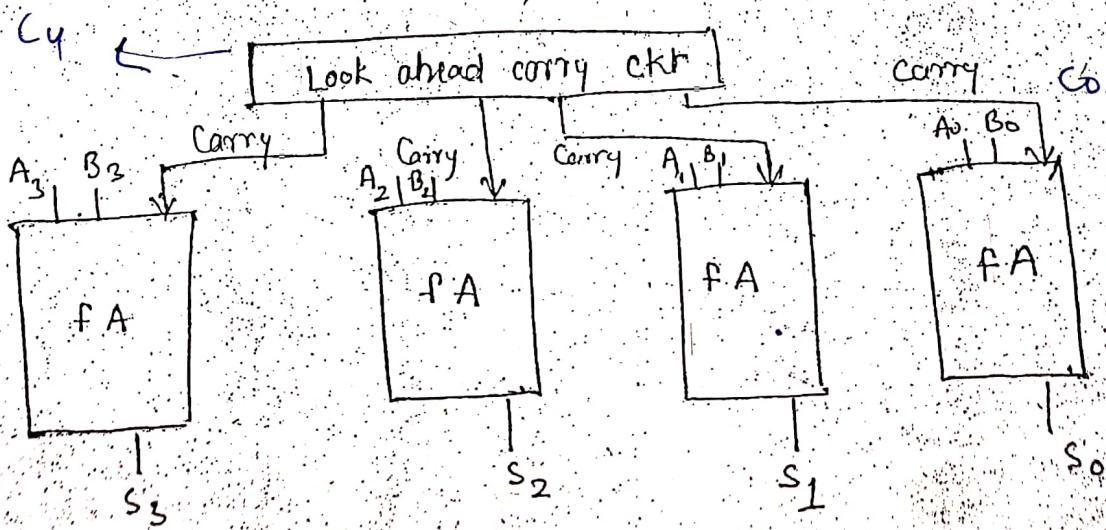
→ In Parallel adder each full adder will provide ~~two~~ 2 logic gate delays. In n bit parallel adder if require $2n$ tpd delay to provide final result

Disadvantage of Parallel

~~wp~~ → carry propagation delay will present, due to this as number of bits increases speed of operation decreases.

Look ahead carry adder (fastest)

It is used as it is the fastest adder



$C_4 \ S_3 \ S_2 \ S_1 \ S_0$

Comparisons



<u>$A \geq B$</u>		$A > B$	$A = B$	$A < B$
A	B	X	Y	Z
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

$$\therefore x = \overline{AB}$$

$$\dots Y = \bar{A}\bar{B} + AB = A \odot B$$

$$Z = \overline{A} B$$

A circuit diagram for a half-adder. It has two inputs, A and B, represented by vertical lines. Input A is connected to the left input of a top AND gate and the left input of a bottom OR gate. Input B is connected to the right input of the top AND gate and the left input of a third OR gate. The outputs of the top AND gate and the bottom OR gate are connected to the right input of a second OR gate. This second OR gate has its output labeled X, representing the sum. Its other output is connected to the right input of the third OR gate. The outputs of the bottom OR gate and the third OR gate are connected to the right input of a fourth OR gate. This fourth OR gate has its output labeled Y, representing the carry. Its other output is connected to the right input of a fifth OR gate. The outputs of the third OR gate and the fifth OR gate are connected to the right input of a sixth OR gate. This sixth OR gate has its output labeled Z, representing the sum of the two inputs.

$$7. \text{ Duty Cycle} = \frac{t_{on}}{t_{on} + t_{off}} \times 100$$

Page 8 W.B
9.5

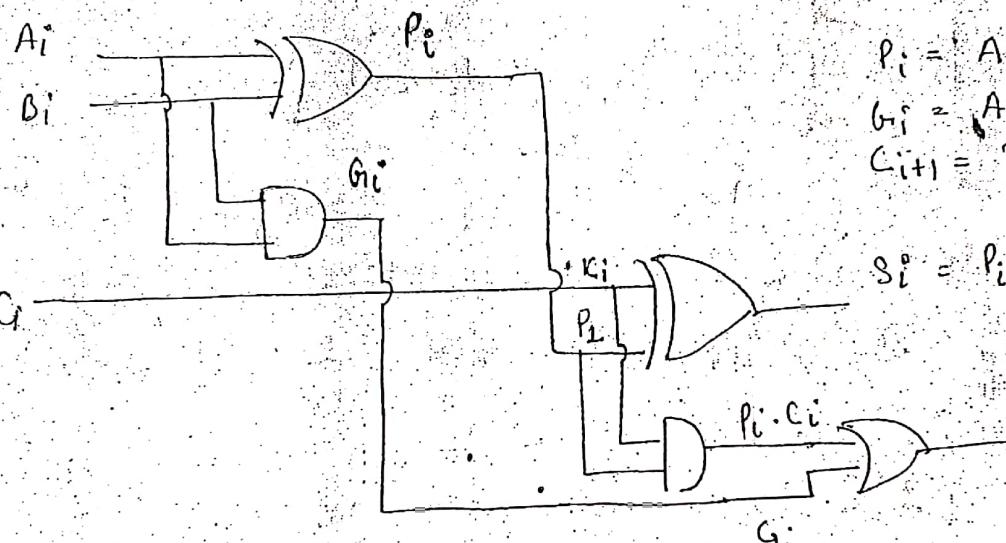
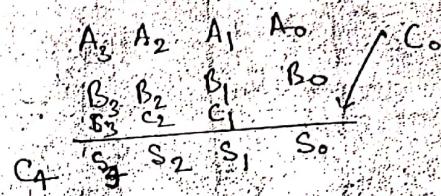
$$t_{on} = 1.7 - 0.4 = 1.3$$

$$T = 2.2 - 0.4 = 1.8$$

$$D = \frac{1.3}{1.8} \times 100 = 72.2\%$$

Look Ahead Carry Adder - 3

4 bit look ahead carry adder



$$P_i = A_i \oplus B_i$$

$$G_i = A_i \cdot B_i$$

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

$$S_i = P_i \oplus C_i$$

$$C_{i+1}$$

$$P_0 = A_0 \oplus B_0$$

$$G_0 = A_0 \cdot B_0$$

$$P_1 = A_1 \oplus B_1$$

$$G_1 = A_1 \cdot B_1$$

$$P_2 = A_2 \oplus B_2$$

$$G_2 = A_2 \cdot B_2$$

$$P_3 = A_3 \oplus B_3$$

$$G_3 = A_3 \cdot B_3$$

$$S_0 = P_0 \oplus C_0$$

$$S_1 = P_1 \oplus C_1$$

$$S_2 = P_2 \oplus C_2$$

$$S_3 = P_3 \oplus C_3$$

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

$$C_1 = P_0 C_0 + G_1$$

$$C_2 = P_1 C_1 + G_2 \Rightarrow P_1 (P_0 C_0 + G_1) + G_2$$

$$C_3 = P_2 C_2 + G_3 \Rightarrow P_2 (P_1 (P_0 C_0 + G_1) + G_2) + G_3$$

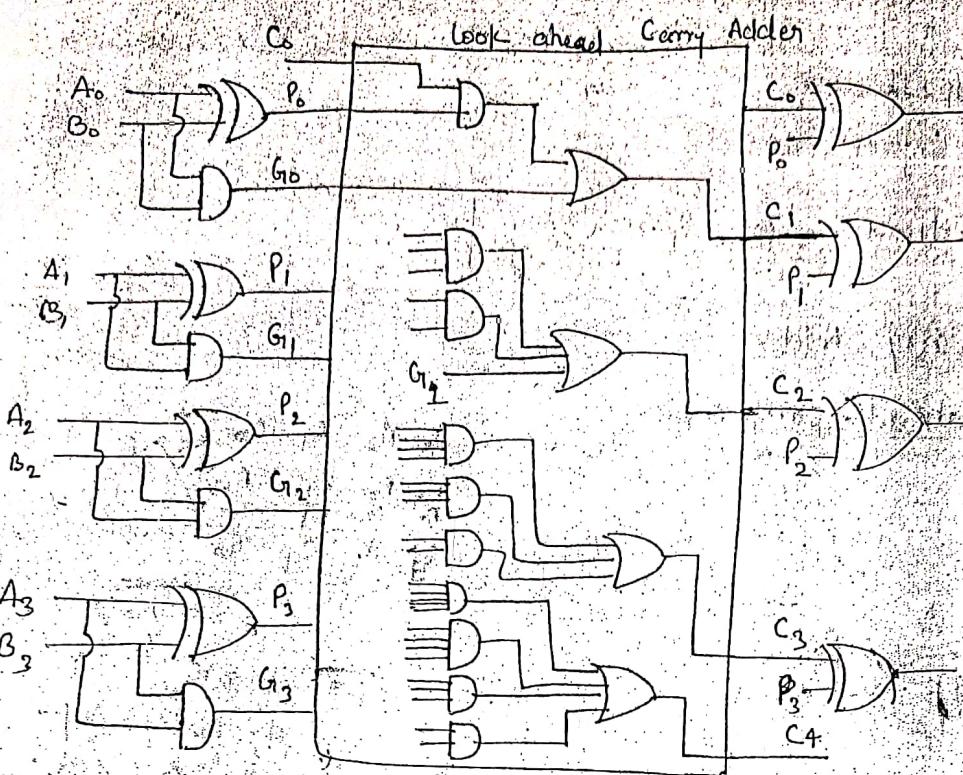
$$C_4 = P_3 C_3 + G_4 \Rightarrow P_3 (P_2 (P_1 (P_0 C_0 + G_1) + G_2) + G_3) + G_4$$

$$= P_1 (P_0 C_0 + G_1) + G_2$$

$$= P_1 P_0 C_0 + P_1 P_0 G_1 + G_2$$

$$= P_2 P_1 P_0 C_0 + P_2 P_1 P_0 G_1 + P_2 G_2 + G_3$$

$$= P_3 P_2 P_1 P_0 C_0 + P_3 P_2 P_1 P_0 G_1 + P_3 P_2 G_2 + P_3 G_3 + G_4$$



- 1) Look ahead carry adder is fastest adder (carry is gen. 1 level)
- 2) In this carry ckt. is implemented with 2 level AND-or gate n/w.
- 3) In this to provide carry output it require 3 logic gate delay and to provide sum output 4 logic gate delays.
- 4) Number of AND gates used in n -bit look ahead carry network is $\frac{n(n+1)}{2}$
- No. of OR gates for n -bit LACA is m