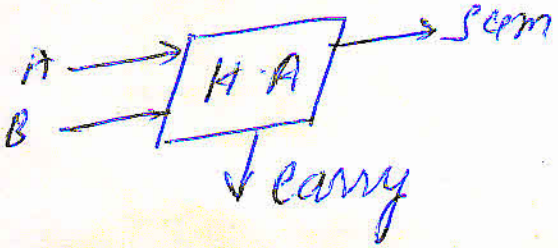
Combinational circuit

There is no feedback element. present O/P depend only up on present I/P : (NO. memory elemt)

* Design of combinational circuit
1. Identify I/P and O/P
2. Construct truth table
3. write logical operation
4. Simplify logical "
5. Construct the logic ckt

Arthematic circuit -- HA. HS. FA. FS

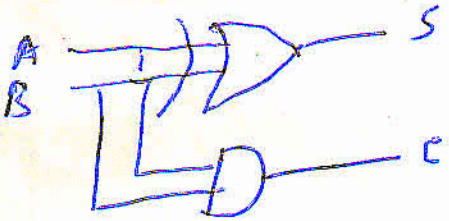
⇒ Half Adder :- perform addition of two bits,



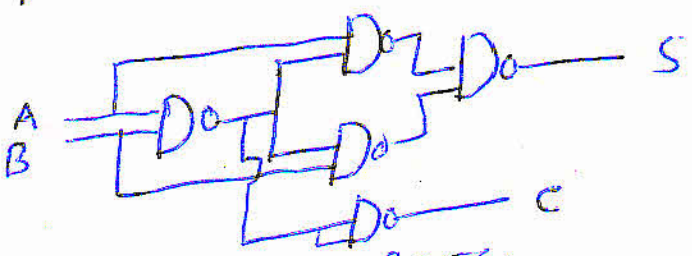
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = A \oplus B$

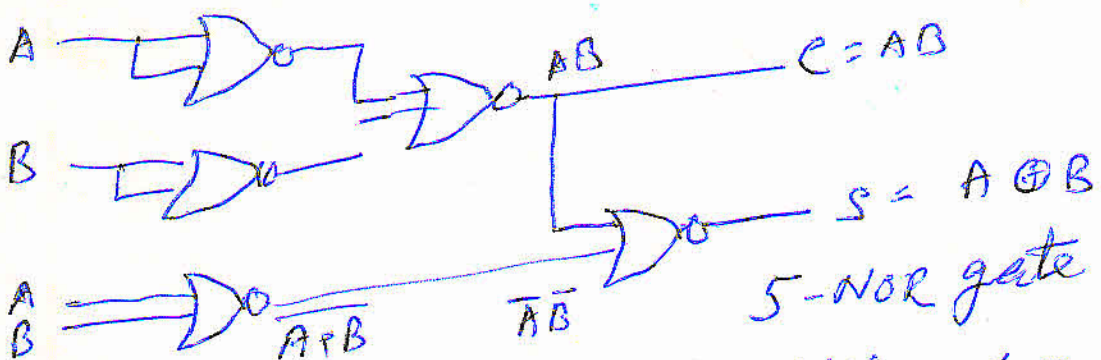
$C = AB$



⇒



5 NAND - GATE.



5-NOR gate

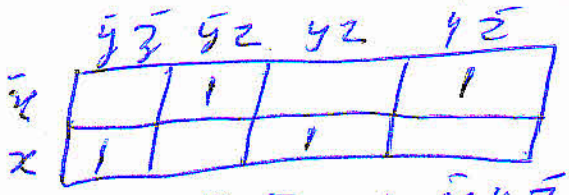
⇒ Full Adder → perform addition of 3 bits Two Input ## and ## represent two significant bits to be added and third Input Z represent Carry from lower Significant position.

⇒

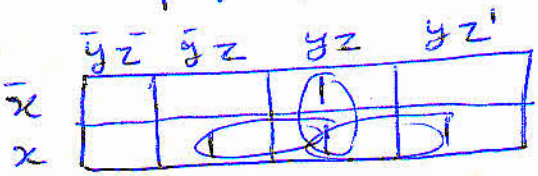
x	y	z	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

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

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



$S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$



for Carry

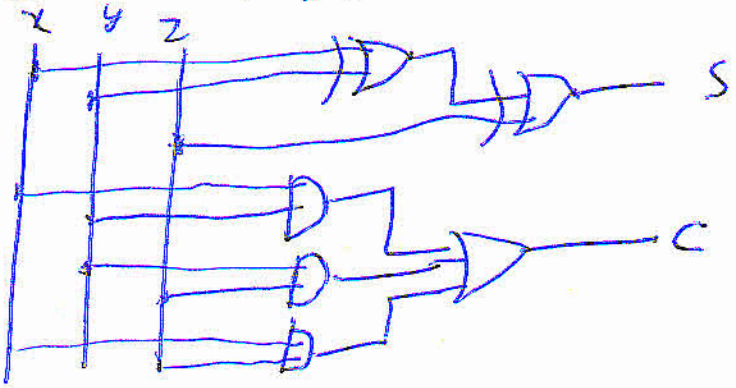
$C = xy + yz + zx$

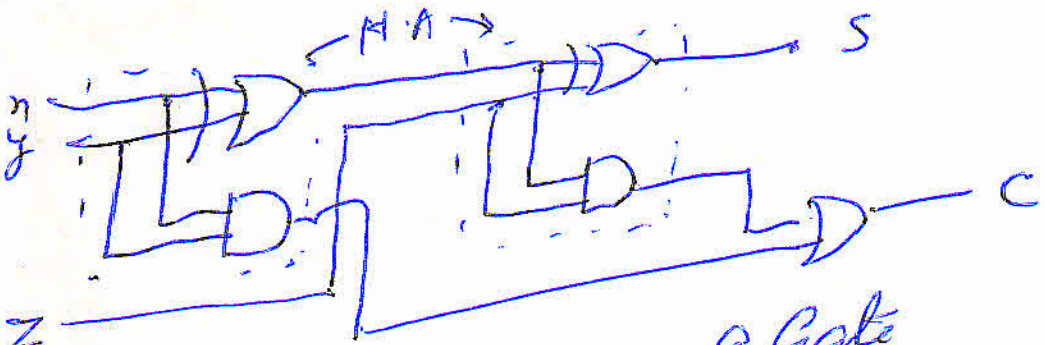
$= xy + \bar{x}yz + x\bar{y}z$

$= xy + (x \oplus y)z$

$= \bar{x}(y \oplus z) + x(y \odot z)$

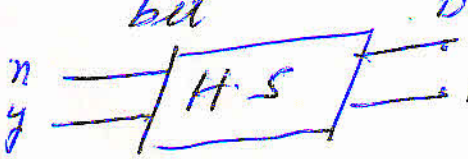
$S = x \oplus y \oplus z$





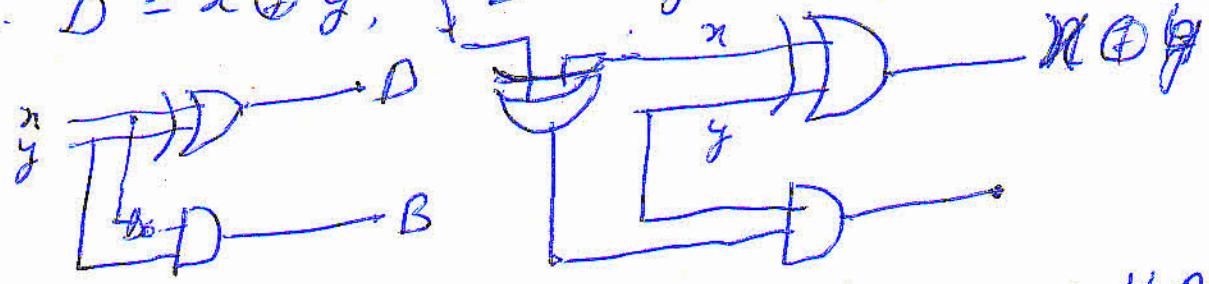
* F.A by NAND — 9 Gate
NOR — 9 Gate

⇒ Half Subtracter — perform subtraction of two bit



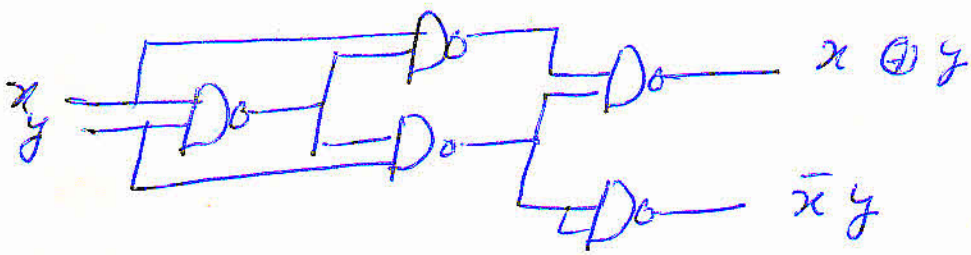
x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$\Rightarrow$ $D = x \oplus y$, $B = \bar{x}y$

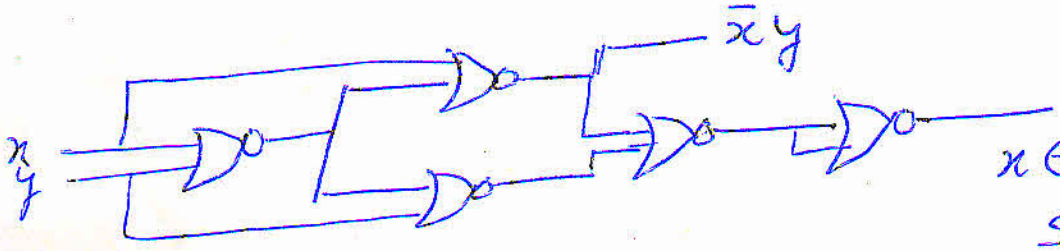


If Control = 0 $\rightarrow$ HA
= 1 $\rightarrow$ HS

$\Rightarrow$ H.S using NAND



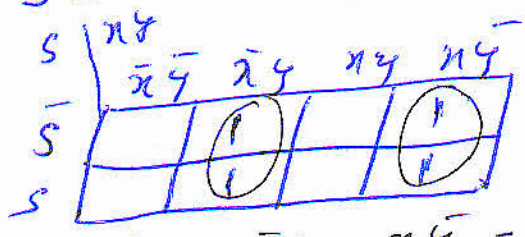
5 NAND



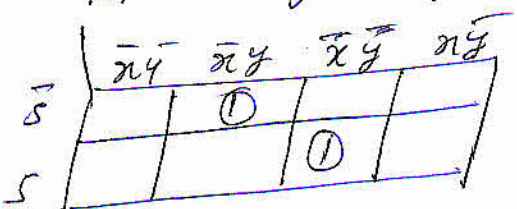
5 - NOR

$\Rightarrow$ Design a HA/H.S controlled i.e it behave as Half Subtractor for $S = 0$ and HA for $S = 1$

	S	x	y	S/D	B/C
HS	0	0	0	0	0
	0	0	1	1	1
	0	1	0	1	0
	0	1	1	0	0
HA	1	0	0	0	0
	1	0	1	1	0
	1	1	0	1	0
	1	1	1	0	1



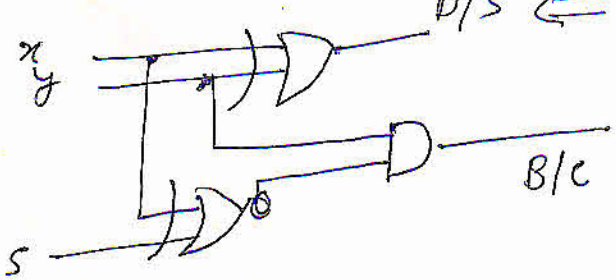
$D/S = \bar{x}y + x\bar{y} = x \oplus y$



$B/C = \bar{S}\bar{x}y + S\,x\,y$

$= y \cdot \overline{x \oplus S} = y \cdot x \odot S$

$D/S \Leftarrow$

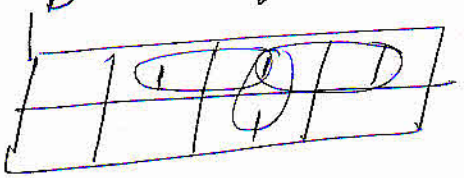


Full Subtractor → perform subtraction Between two bits taking in to account that a 1 have been borrowed from a lower significant stage

Min	Sub	previous Borrow	D	B
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

x \ yz	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$	0	1	0	1
x	1	0	1	0

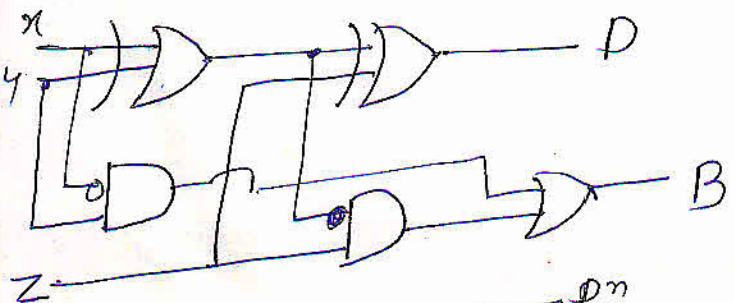
$$D = x \oplus y \oplus z$$

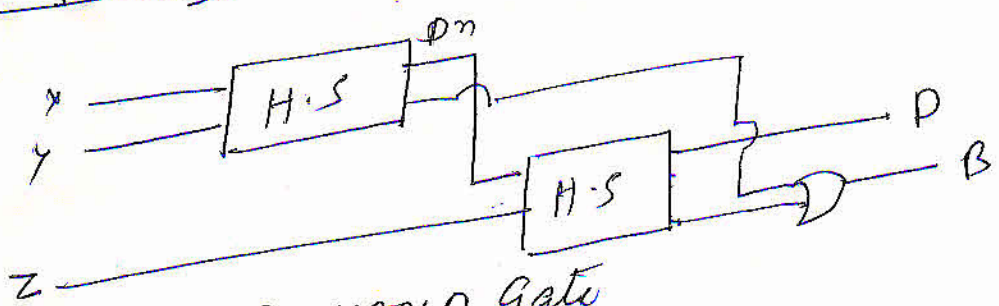


$$B = \bar{x}z + \bar{x}y + yz$$

$$B = \bar{x}y + \bar{x}\bar{y}z + xyz$$

$$B = \bar{x}y + [\overline{x \oplus y}]z$$

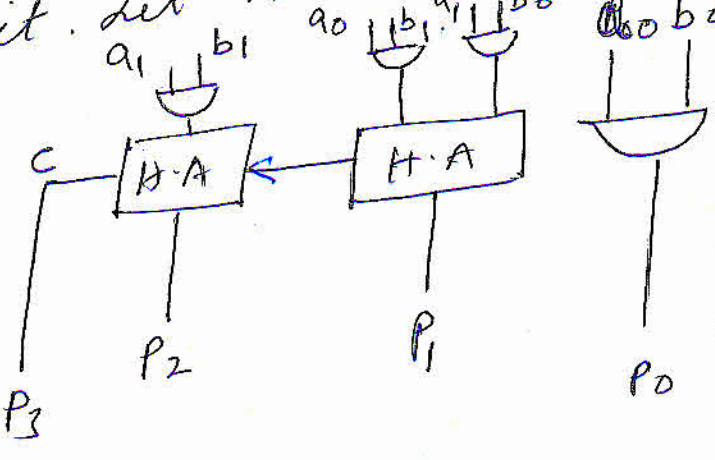




⇒ F.S → 9 NAND Gate
→ 9 NOR Gate.

Two bit multiplier ⇒ In general A is of m bit ($m \neq 1$) and B is of n bit ($n \neq 1$) then product of AB will have $(m+n)$ bit. Let $A = a_1 a_0$ $B = b_1 b_0$

$$A \cdot B = \begin{array}{r} a_1\ a_0 \\ b_1\ b_0 \\ \hline a_1 b_0 \quad a_0 b_0 \\ + \quad a_1 b_1 \quad b_1 a_0 \quad X \\ \hline P_3 \quad P_2 \quad P_1 \quad P_0 \end{array}$$



Three bit multiplier

$A = a_2\ a_1\ a_0$

$B = b_2\ b_1\ b_0$

---

a_2b_0 a_1b_0 a_0b_0

a_2b_1 a_1b_1 a_0b_1 X

a_2b_2 a_1b_2 a_0b_2 X

P_5 P_4 P_3 P_2 P_1 P_0

1 1 1 HA H.A AND

1 1 FA 1 FA

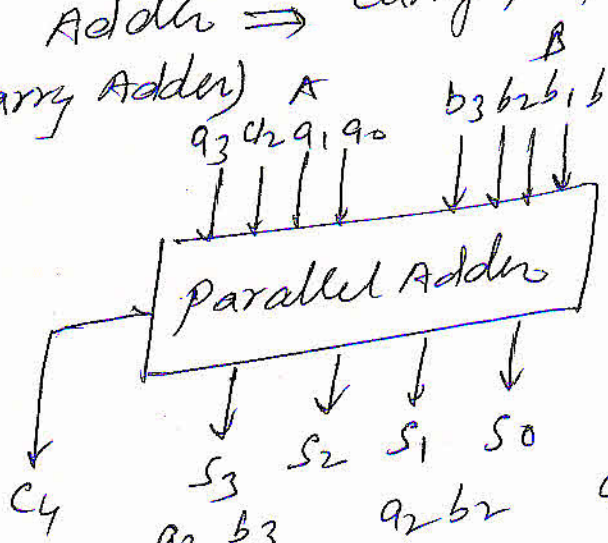
HA

TWO → FA

Three → H.A

one → And GATE

Parallel Adder ⇒ Carry propagate adder (Ripple Carry Adder)

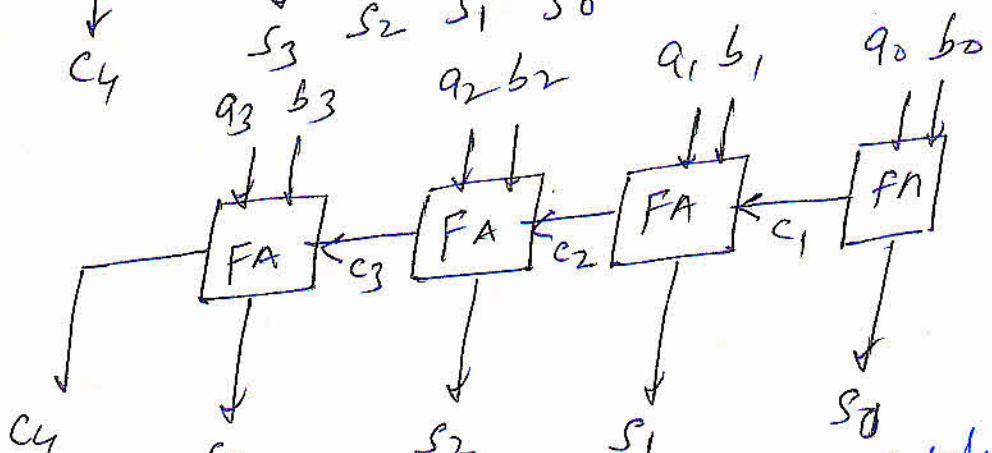


C_3 C_2 C_1

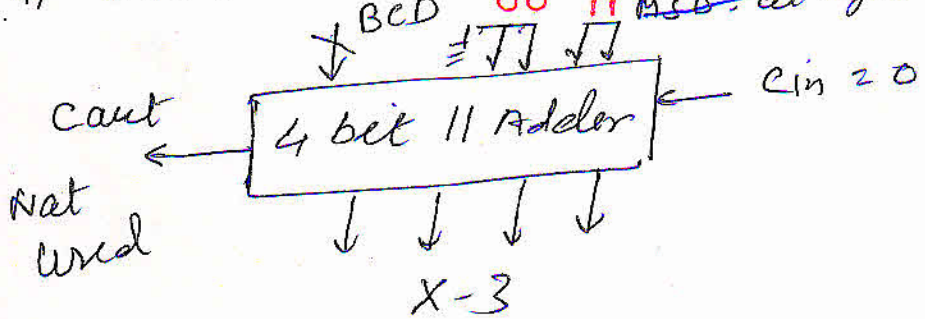
Augend A a_3 a_2 a_1 a_0

Addend B b_3 b_2 b_1 b_0

C_4 S_3 S_2 S_1 S_0



BCD to X-3 Converter ⇒ Using 4 bit 11 Adder



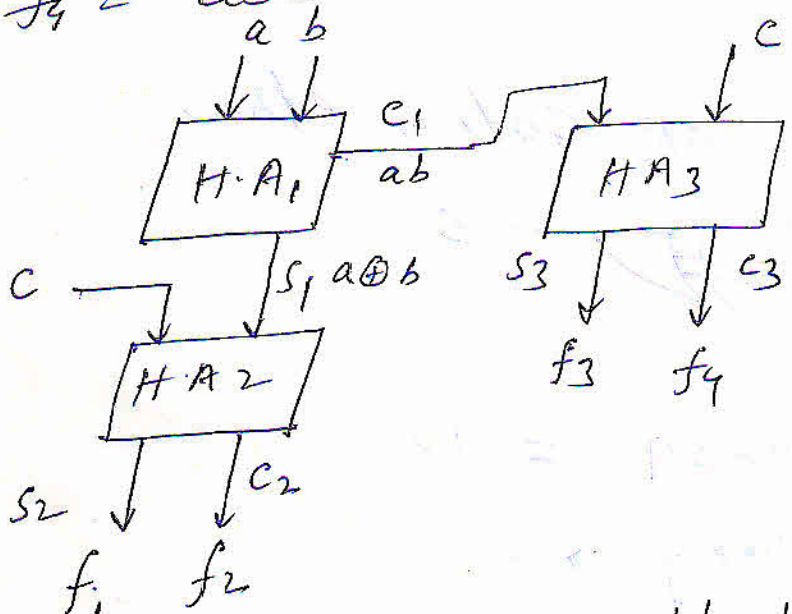
⟹ Implement following 4 functn. using H.A.

$f_1 = a \oplus b \oplus c$

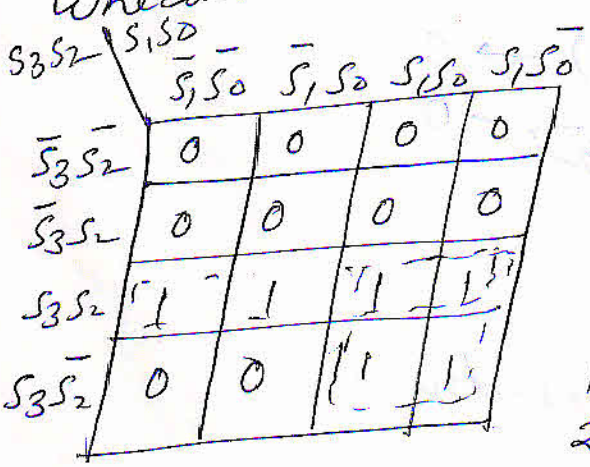
$f_2 = \bar{a}bc + a\bar{b}c = [a \oplus b]c$

$f_3 = ab\bar{c} + (\bar{a}+\bar{b})c = ab\bar{c} + \overline{ab}c = ab \oplus c$

$f_4 = abc$



BCD Adder :- It adds two BCD digits and provide the result in BCD (Sum digit). It must include correction logic if Sum > 9 or carry is generated on 11 adder than 0110 is to be added to the result. The following k-map gives the design of a ckt that decide Whether the Sum is a valid BCD code or Not.



$f = S_3S_2 + S_3S_1$

$f = S_3(S_1 + S_2)$

How Many dont care terms will be in BCD Adder.

Total term. $2^9 = 512$ (A B: 0 0 to 9 9)

A and B can vary from. $= 100$

100 with $C_{in} = 0$

100 " $C_{in} = 1$, will contribute

200 valid terms.

Dont care $= 512 - 200 = 312$

If Sum > 9
When ever the o/p of AND gate is 0 the Carry from first adder is 1 (or vice versa)

Here 0 ⇉ 1
1

If Sum ≤ 9
then Input to Ex-OR Gate is either 0,0, or 1,1 Hence 0,0 ⇉ 0
1,1

Example

9 ⇒ 1001
3 ⇒ 0011
1100 Sum > 9 = 12
Carry → 0
$S_3(S_2 + S_1)$ → 1
0 ⇉ 1
1

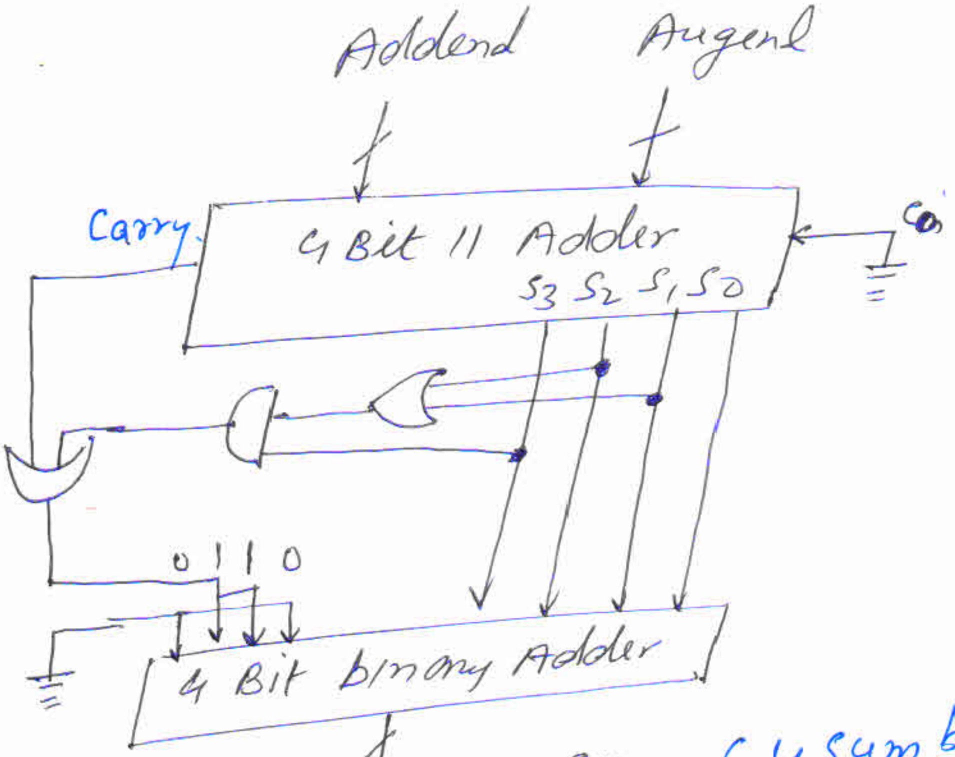
Correction

0110
0001 0010
BCD BCD
⇒ 0110
0011
1001

Sum < 9
Carry → 0
$S_3(S_2 + S_1)$ → 0
0 ⇉ 0
0

⇒ 9 → BCD → 1001
9 − " → 1001
1 ← Carry → 10010
$S_3(S_2 + S_1)$ → 0 + 0110 → Correction
1 ⇉ 1
0 10000
BCD of 1 BCD → 8 ⇒ 18

⇒



⇐

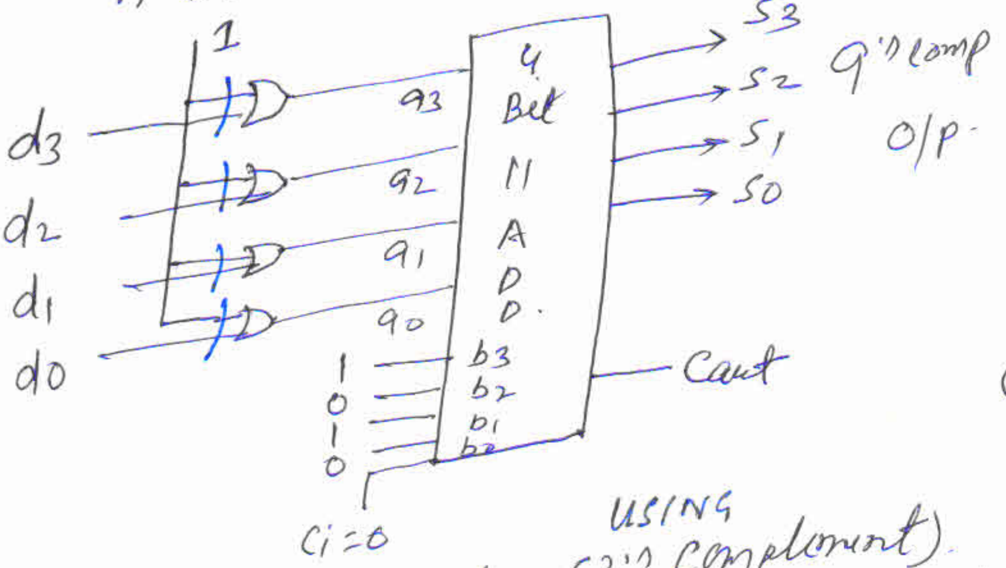
BCD sum. (4 sum bit and one carry bit).

9's complement of a BCD data =

9 - in decimal. = 1's complement + 1010

1's comp. of a binary data can be achieved by -

$A \oplus 1 = \bar{A}$



BCD of 4
0100
9's complement = 5

in binary 1's comp. → 1011

```
  1011
+ 1010
------
1 0101  = 5 in BCD
```
Cout

Adder / subtracter USING (2's complement).

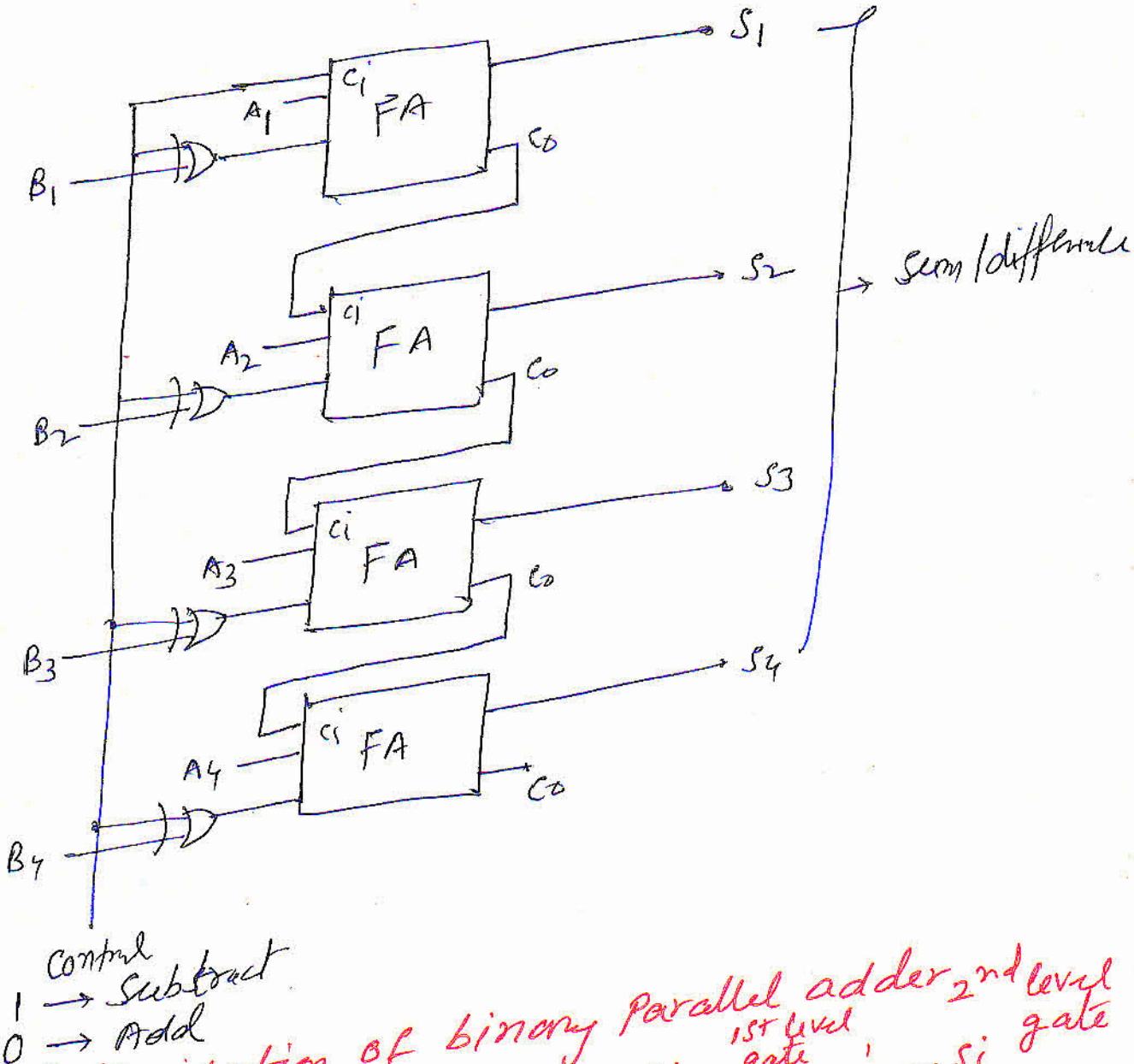
* EX-OR acts as inverter for control = 1
* EX-OR acts as buffer " " = 0

A_4 A_3 A_2 A_1 A_0

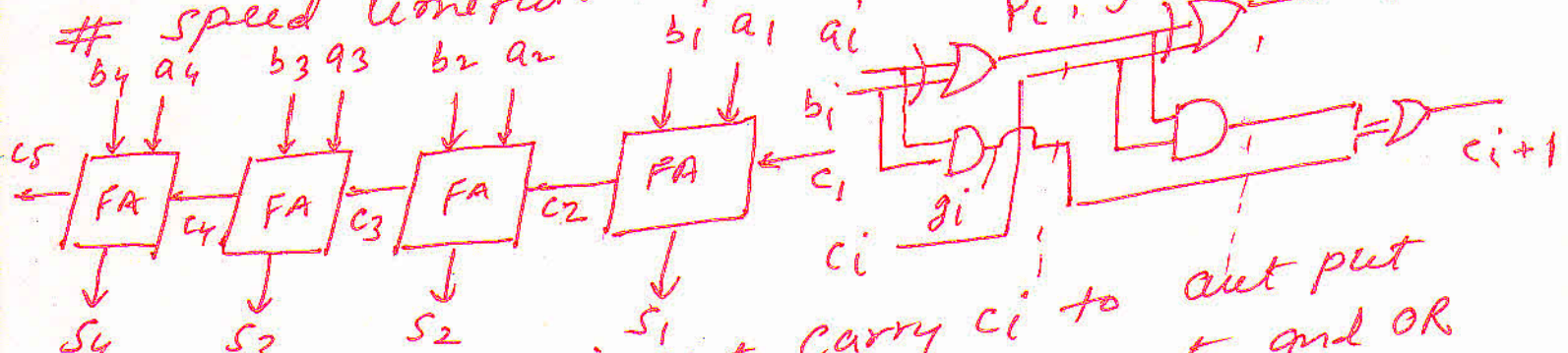
B_4 B_3 B_2 B_1 B_0

+/-

⇒



speed limitation of binary parallel adder



* The signal from input carry C_i to out put C_{i+1} propagate through an ~~and~~ AND gate and OR gate which constitute two gate levels.
* If there are 4 FAs in n adder o/p carry C_5 would have $2 \times 4 = 8$ gate levels from C_1 to C_5
* a_i and b_i are simultaneously available at each FA for S_i to be in steady state. The previous

carry has to propagate through two gate levels.

* The signals P_i and g_i settle to their steady state value after propagation through respective gates i.e. only one level (HA). P_i and g_i are common to all FAs and depend on only a_i and b_i, FA_1, FA_2, FA_3, FA_4 all provide P_i and g_i simultaniously after one gate delay

⇒ Total propagation time = Delay of one (HA) + 2 × n × each gate Delay

in general total propagation delay = (2n+1) times delay of each gate

for 4 bit II Adder / = (2×4+1) 10ns = 90ns

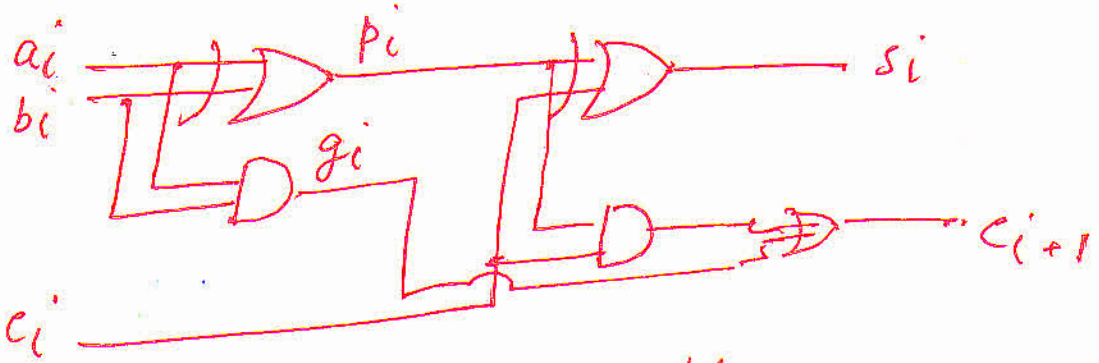
each gate delay = 10ns

C_1 to C_2 = 30ns
C_2 to C_3 = 20ns
C_3 to C_4 = 20ns
C_4 to C_5 = 20ns = 90ns

⇒ The Carry propagation time can be reduced by using faster gates but there is always a physical limit.

⇒ Mother way is to increase hardware complexity so as to reduce Carry propagation time

Carry look ahead Adder — in this type of adder all carries get generated simultaniously.



Let us define two variable

$p_i = a_i \oplus b_i$ → is called carry propagate

$g_i = a_i b_i$ → is " generates "

$s_i = p_i \oplus c_i$

$c_{i+1} = g_i + p_i c_i$

$c_2 = g_1 + p_1 c_1$

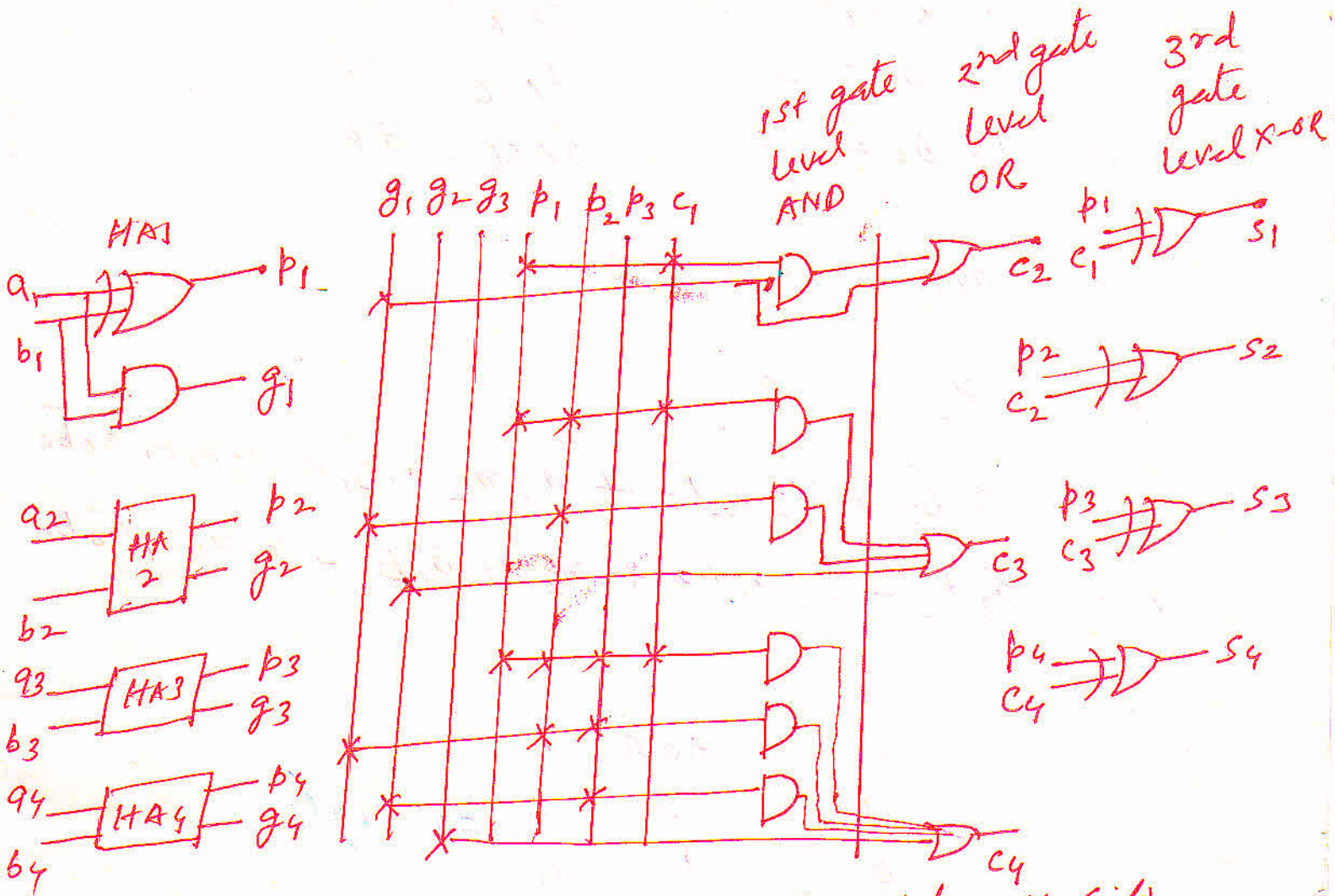
$c_3 = g_2 + p_2 c_2 = g_2 + p_2 (g_1 + p_1 c_1)$

$c_3 = g_2 + p_2 g_1 + p_2 p_1 c_1$

$c_4 = g_3 + p_3 c_3 = g_3 + p_3 (g_2 + p_2 g_1 + p_2 p_1 c_1)$

$c_4 = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 c_1$

* Note that c_2, c_3, c_4 can be expressed in SOP and has two level AND-OR / Two level NAND-NAND implementation.
* Also note that c_4 does not have to wait for c_3 and c_2 to propagate. In fact c_4 is propagate at the same time as c_2 and c_3
* p_i is Responsible for propageting carry from c_i to c_{i+1}
* g_i is independent of c_i



propagation delay of Carry look ahead Adder (4 bit)

= Delay of one HA + Delay of 3 gate levels.
(AND - OR - EX-OR)

$P.D_{EX-OR} = 20ns$ and $P.D_{AND/OR} = 10ns$

Than $P.D_{Carry\ look\ ahead} = \underbrace{20}_{HA} + 10 + 10 + 20 = 60ns$
(EX-OR, AND, OR, EX-OR)

$$Delay_{H.A} = Delay_{X-OR}$$

While $P.D_{Parallel\ adder}$ = Delay of one H.A + 4(AND +OR)

= 20 + 4(10 +10)

= 100ns.

Magnitude Comprector :—

Consider a 4 bit no.

$A = q_3\ q_2\ q_1\ q_0$

$B = b_3\ b_2\ b_1\ b_0$

one bit comprector

$f_{A=B} = \bar{a}\bar{b} + \bar{a}b = a \odot b$

$f_{A>B} = a\bar{b}$

$f_{A<B} = \bar{a}b$

let $x_i = a_i \odot b_i \quad (i = 0, 1, 2, 3)$

$f_{A=B} = x_3 x_2 x_1 x_0$

$f_{A>B} = q_3 \bar{b_3} + x_3 a_2 \bar{b_2} + x_3 x_2 a_1 \bar{b_1} + x_3 x_2 x_1 a_0 \bar{b_0}$

$f_{A<B} = \bar{q_3} b_3 + x_3 \bar{a_2} b_2 + x_3 x_2 \bar{q_1} b_1 + x_3 x_2 x_1 \bar{q_0} b_0$

one bit

q	B	A=B	A<B	A>B
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

$\Rightarrow$ logic

$\Rightarrow$ If $A = B$

A = 9 1001

B = 9 1001

$a_i \odot b_i = x_i$

A > B

A = 13 1101

B = 11 1011

$x_3 = 1$

$x_2 = 0$

$x_1 = 0$

$x_0 = 1$

$\Rightarrow$ If A < B

1011

1101

$x_3 = 1$

$x_2 = 0$

$x_1 = 0$

$x_0 = 0$