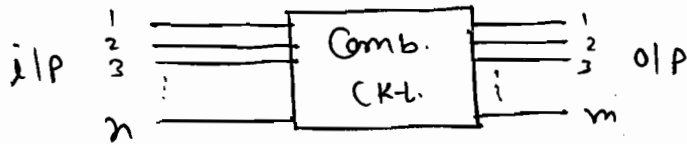


★

Digital Circuits

Combinational Ckt

Sequential Ckt.



→ Outputs = $F(\text{Present I/P})$

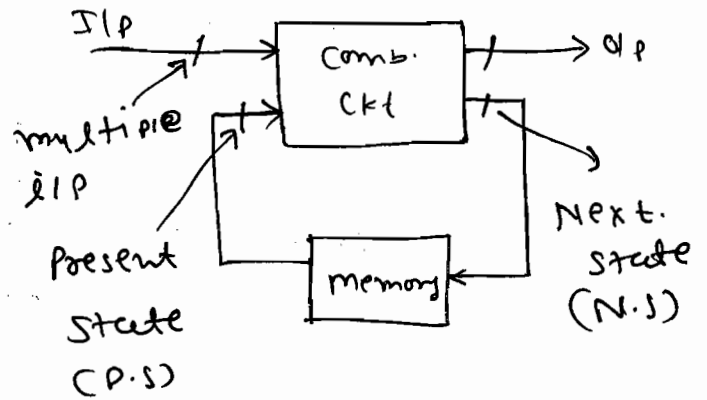
E.g. = Code Converter,

Adder & Subtractor,

Decoder, Mux,

Mag. Comparators, PLA,

✓ Rom



$O/P = F(\text{present state, present inputs})$

$\text{Next state} = F(\text{present state, present I/P.})$

→ Mooore Seq Ckt.

$O/P = F(\text{present state only})$

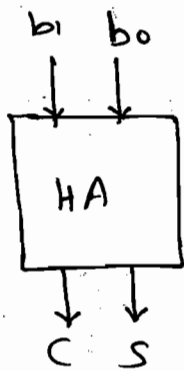
$N.S. = F(\text{present Inputs, present state}).$

E.g. Shift register,
Counters, calculators,
MP, PC.

* Arithmetic Combinational Ckt.

53

(1) Half Adder:



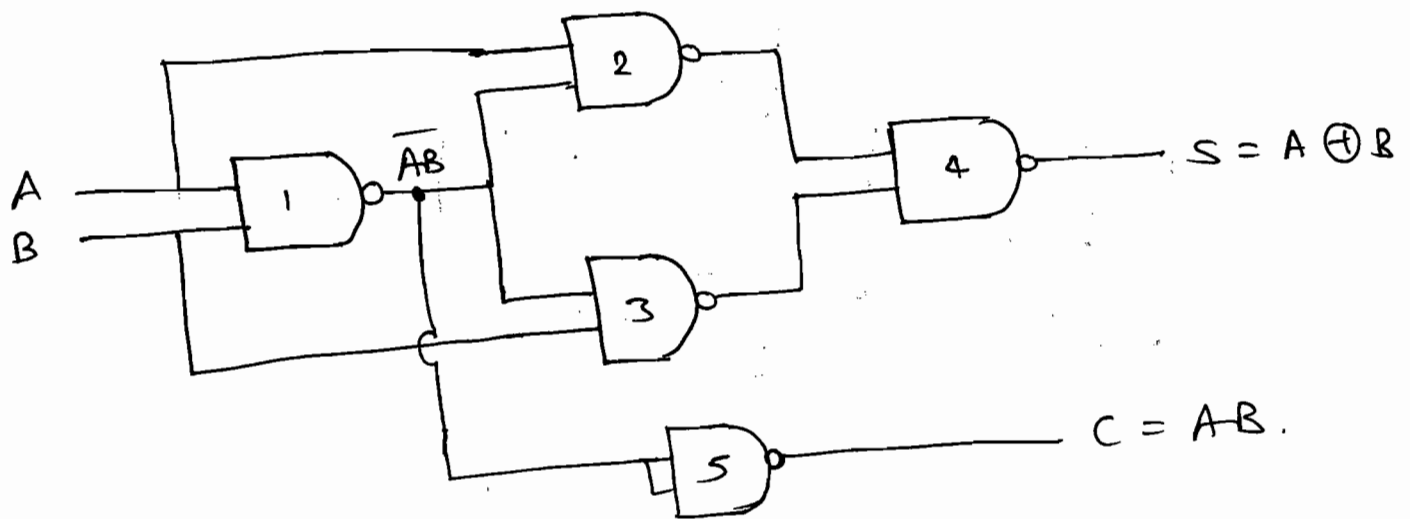
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

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

$$S = A \oplus B$$

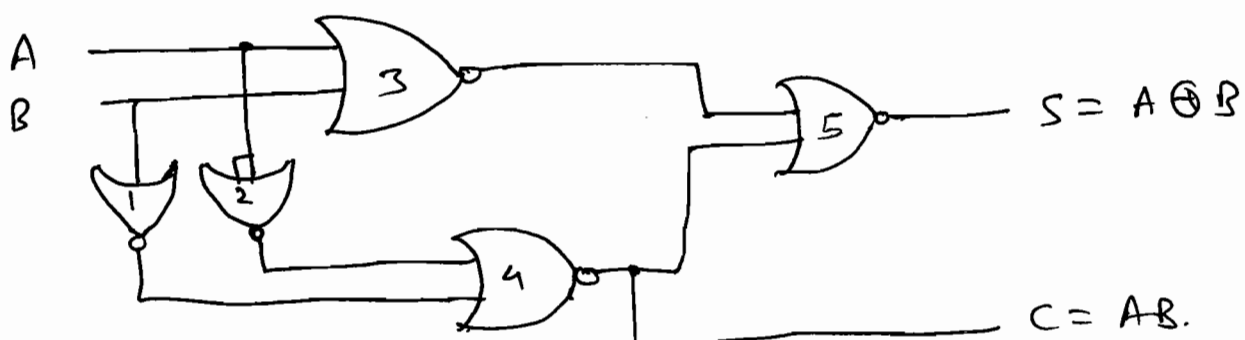
$$C = A \cdot B$$

⇒ Half Adder using NAND Gates.



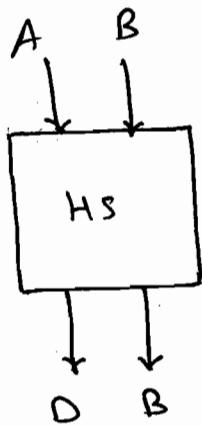
So, ⑤ NAND gate required.

⇒ Half Adder using NOR gate



So, ⑤ NOR gate required

(2) Half Subtractor:



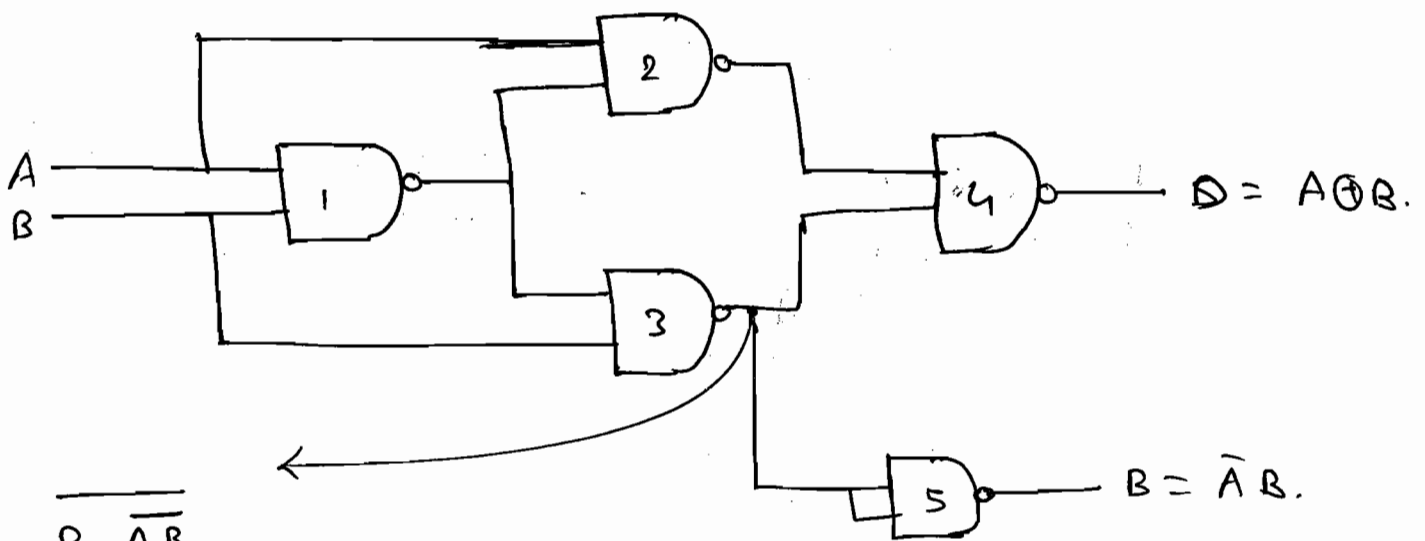
A	B	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$D = A \oplus B$$

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

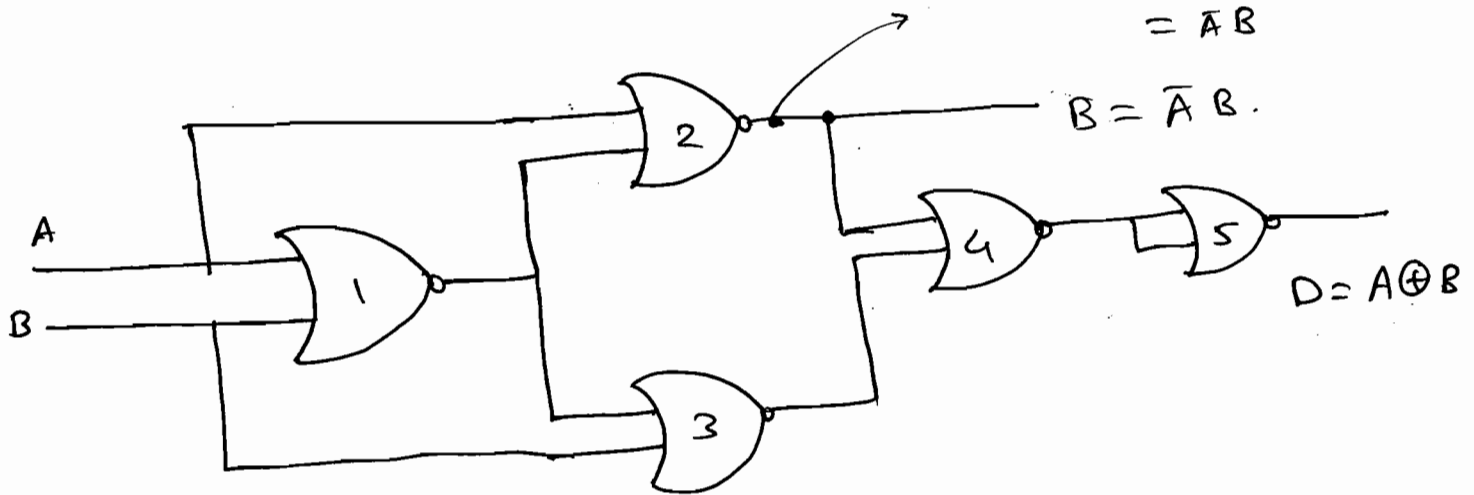
$$\begin{array}{r} 10 \\ B - 1 \\ \hline 1 \rightarrow D \end{array}$$

⇒ Half Subtractor ~~using~~ using NAND gate only:

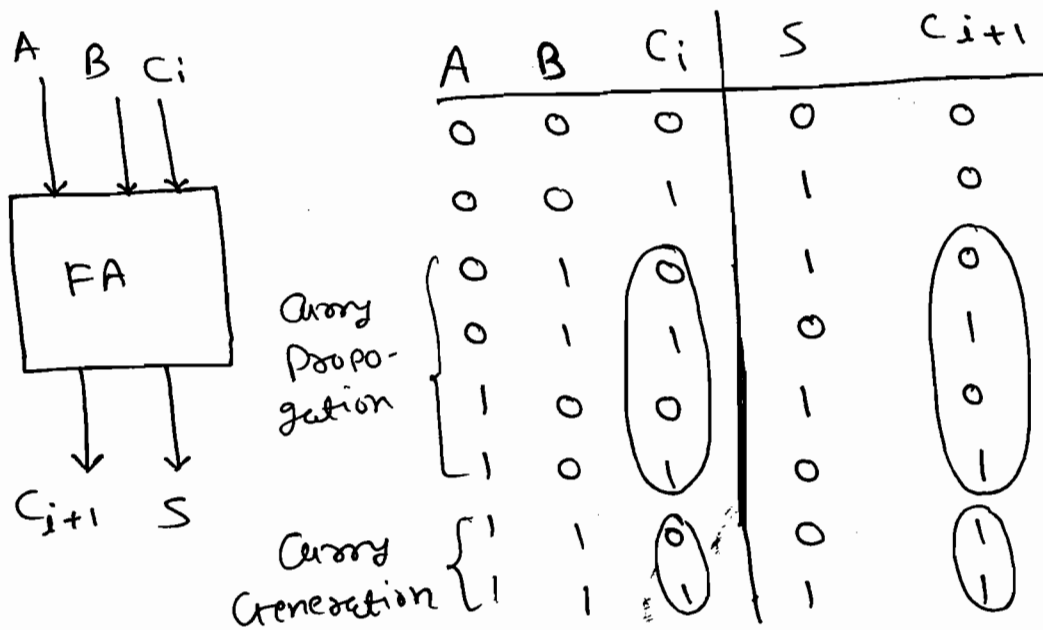


$$\begin{aligned} & \overline{B \cdot \bar{A} B} \\ &= \bar{B} + AB \\ &= A + \bar{B} = \overline{\bar{A} \cdot B} \end{aligned}$$

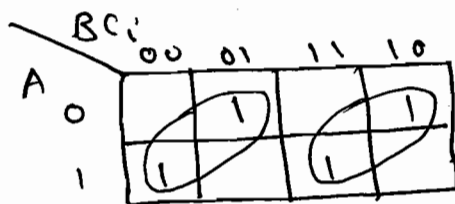
⇒ Half Subtractor using NOR gate only. 65



(3) Full Adder: (1 bit adder).



$$S(A, B, C_i) = \sum m(1, 2, 4, 7)$$



$$S = A \oplus B \oplus C_i \quad (\text{or}) \quad A \odot B \odot C_i$$

$$C_{i+1} = \sum m(3, 5, 6, 7)$$

$$= \bar{A}BC_i + A\bar{B}C_i + AB\bar{C}_i + ABC_i$$

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

$$\therefore C_{i+1} = C_i (A \oplus B) + AB$$

	BC_i			
	00	01	11	10
A				
0			1	
1		1	1	1

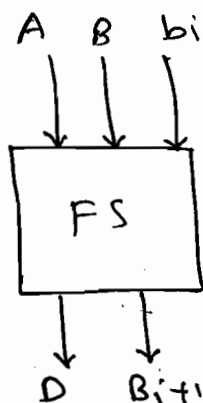
$$C_{i+1} = AB + BC_i + AC_i$$

NOTE: Full adder

- (i) Require 9 NAND gate
- (ii) Require 12 NOR gate.

(4) Full Subtractor:

→



A	B	bi	D	bi+1
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

$$\rightarrow D(A, B, b_i) = \sum m(1, 2, 4, 7).$$

	Bb_i			
	00	01	11	10
A				
0	0	1	1	1
1	1	1	1	1

$$D = A \oplus B \oplus b_i \text{ (or) } A \odot B \odot b_i.$$

$$\rightarrow b_{i+1}(A, B, b_i) = \sum m(1, 2, 3, 7).$$

$$b_{i+1} = m_1 + m_2 + m_3 + m_7$$

$$= \bar{A}\bar{B}b_i + \bar{A}B\bar{b}_i + \bar{A}Bb_i + ABb_i$$

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

$$b_{i+1} = b_i(A \odot B) + \bar{A}B.$$

	Bb_i			
	00	01	11	10
A				
0	0	1	1	1
1	0	1	1	0

$$\therefore b_{i+1} = \bar{A}B + Bb_i + \bar{A}b_i$$

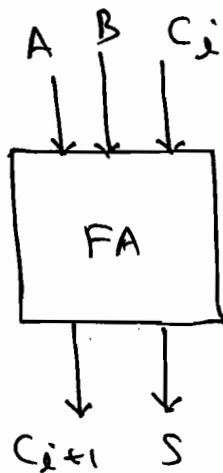
NOTE: Full subtractor

(i) Required 9 NAND gate

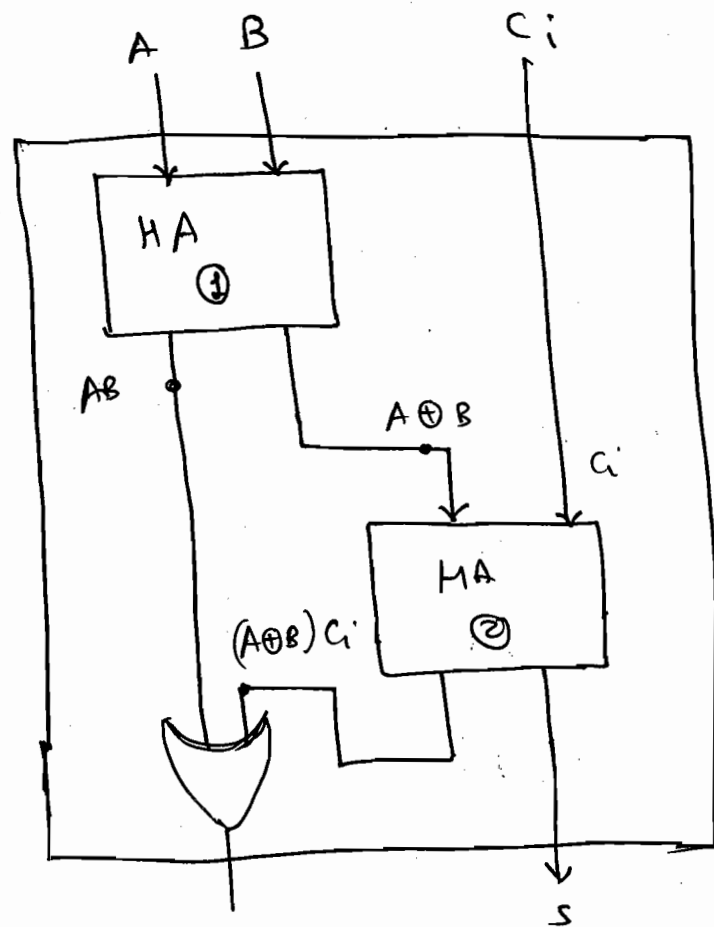
(ii) Required 12 NOR gate.

Ex-1 Implement Full Adder using HAs and Logic gate:

Ans:



$\equiv$



$$C_{i+1} = AB + C_i(A \oplus B)$$

1 Full adder = 2 Half adder + 1 OR gate

Ex-2 How many HA required to implement the following bⁿ.s.

$$F_1 = \bar{A}C + AB\bar{C} + \bar{B}C.$$

$$F_2 = A \oplus B \oplus C.$$

$$F_3 = \bar{A}BC + A\bar{B}C.$$

Ans: (i) $F_1 = \bar{A}C + AB\bar{C} + \bar{B}C.$

$$= \overline{AB} \cdot C + AB\bar{C}$$

$$= (\overline{A \cdot B}) \cdot C$$

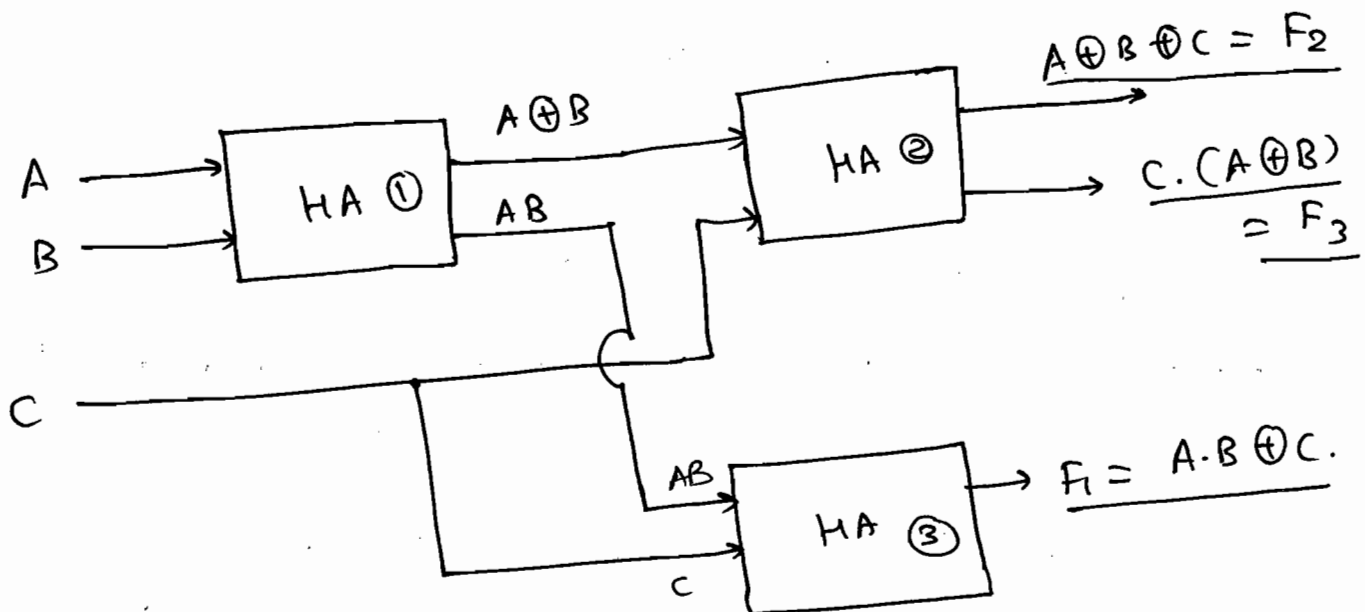
$$X = \overline{AB}$$

$$= \bar{X} \cdot C + X\bar{C}$$

$$F_1 = A \cdot B \oplus C \rightarrow$$

For (ii) $F_2 = A \oplus B \oplus C.$

(iii) $F_3 = (\bar{A}B + A\bar{B})C = C \cdot (A \oplus B)$



* Binary Adder:

- (1) Parallel binary Adder:
- (2) Carry Look Ahead Adder:
- (3) Serial Adder.

(1) Parallel binary Adder: (4 bit):

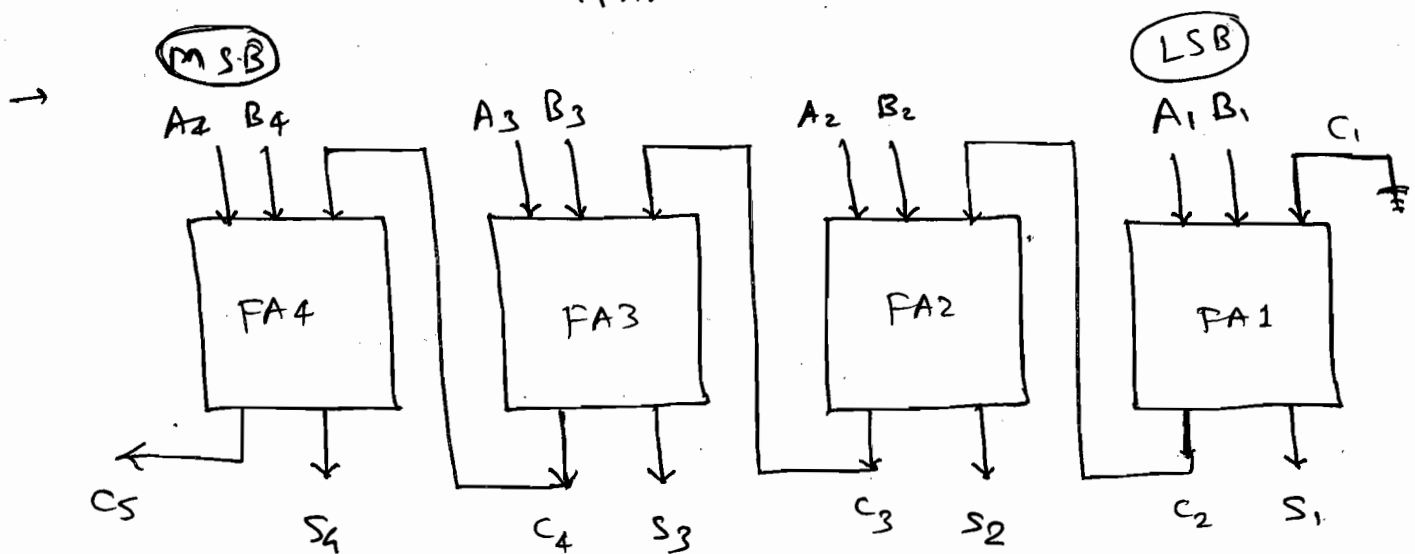
→ ~~IC = 74LS83~~

[IC = 74LS83]

$$\begin{array}{r}
 A = A_4 \ A_3 \ A_2 \ A_1 \\
 + B_4 \ B_3 \ B_2 \ B_1 \\
 \hline
 C_5 \ S_4 \ S_3 \ S_2 \ S_1
 \end{array}$$

$\xrightarrow{\text{FA}}$
 $\xrightarrow{\text{HA (OR) FA}}$

~~IC = 74LS83~~ + 3 FA + 1 HA (OR)
4 FA



Q: how many HA. required to implement 4-bit parallel adder

Ans: 7 HA &
3 OR gate

Q.

→ In a 4 bit parallel binary adder a FA takes 32 ns to produce the sum and 14 ns to produce the carry. Determine

- Time required for addition.
- the addition rate of the adder.

Ans:

a)

Time required for Addition in N-bit parallel adder = $T = (N-1)t_c + \max(t_s, t_c)$.

$$\Rightarrow T = (4-1)14 + \max(32, 14)$$

$$= 42 + 32$$

$$T = 74 \text{ ns}$$

b)

$$\text{Addition Rate} = \frac{1}{T} = \frac{1}{74 \times 10^{-9}} \text{ Hz}$$

→ This can be used upto 4 bit.

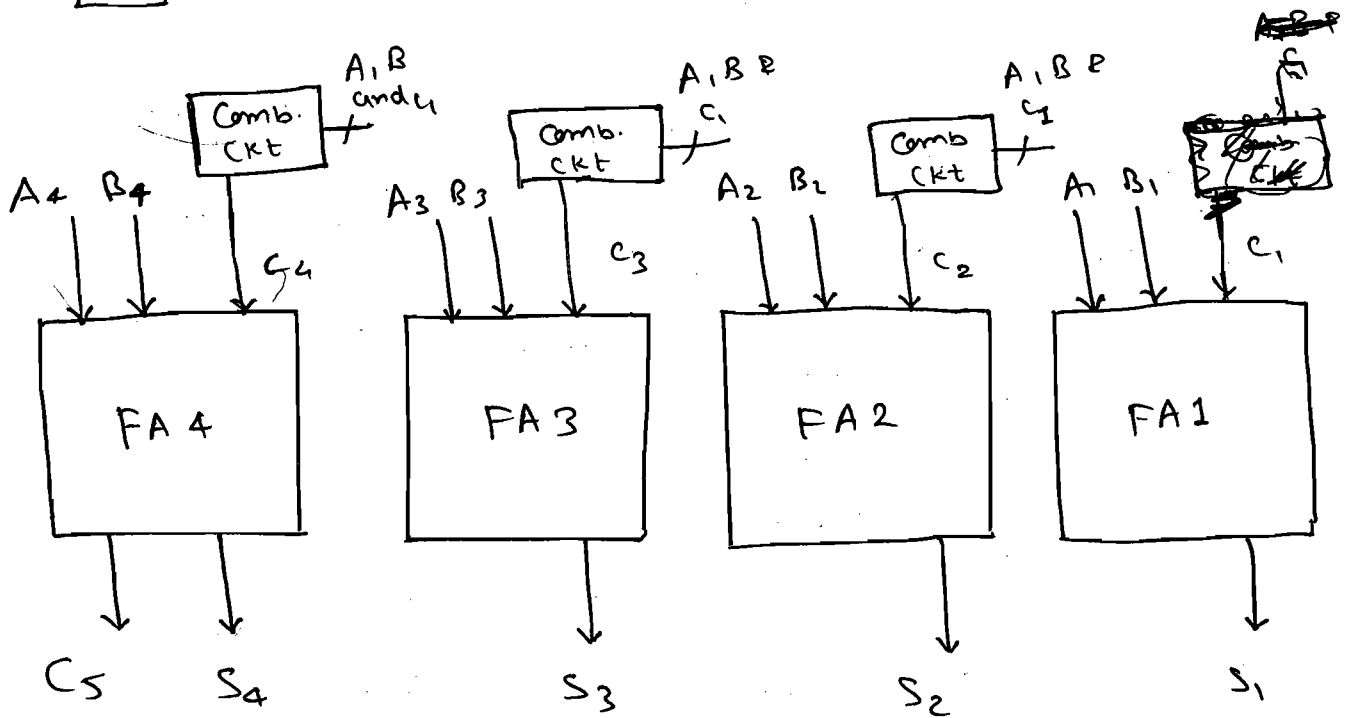
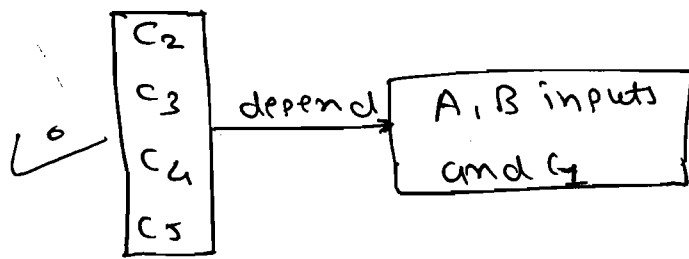
→ Disadvantages:

→ as the size of the Adder increases the Speed operation decreases as the carry

✓ has to propagate through all the FAs to overcome this we use Carry Look ahead Adder.

(2) Carry Look Ahead Address:

→ Principle:



$$\rightarrow C_{i+1} = C_i (A \oplus B) + AB$$

$$P_i = A \oplus B = \text{Propagation of carry}$$

$$C_i = AB = \text{Carry generation}$$

$$\therefore C_{i+1} = C_i P_i + C_i$$

$$\rightarrow \boxed{C_2 = C_1 P_1 + C_1} \quad - \textcircled{1}$$

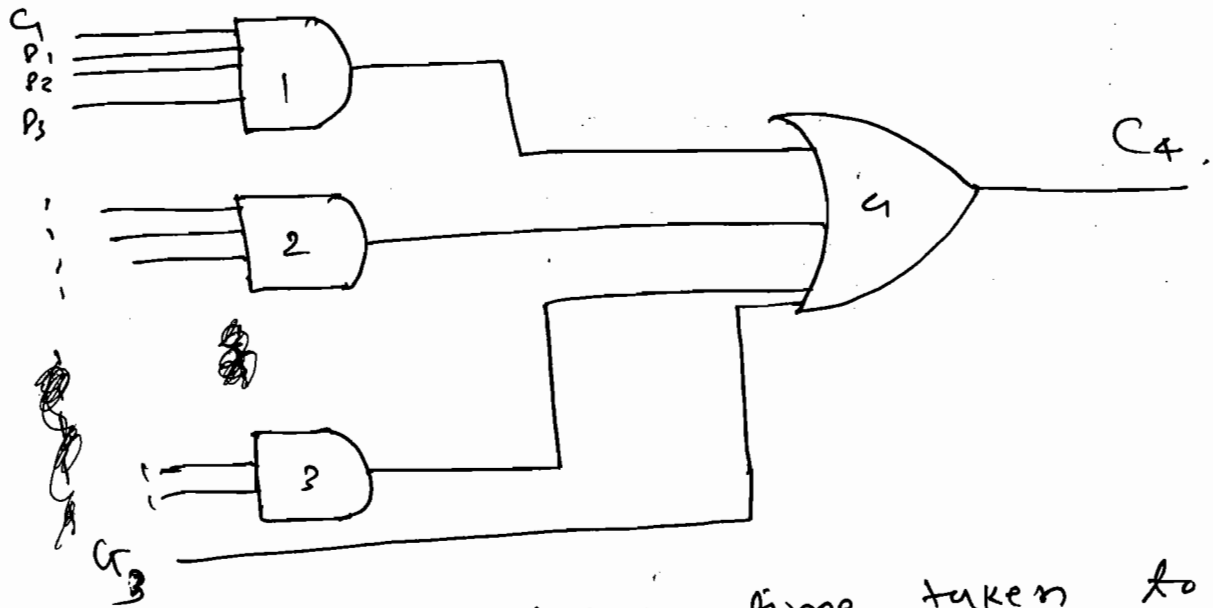
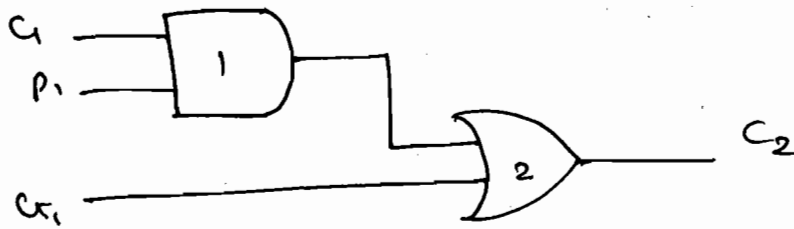
$$\rightarrow C_3 = C_2 P_2 + C_2$$

$$\boxed{C_3 = C_1 P_1 P_2 + C_1 P_2 + C_1} \quad - \textcircled{2}$$

$$\rightarrow C_4 = C_3 P_3 + C_3$$

→ $C_4 = C_1 P_1 P_2 P_3 + C_1 P_2 P_3 + C_2 P_3 + C_3$ — ③ 73

*



→ In Carry Look-Ahead time taken to generate the carries C_2, C_3, C_4 is same as they are implemented by 2 level Logic.

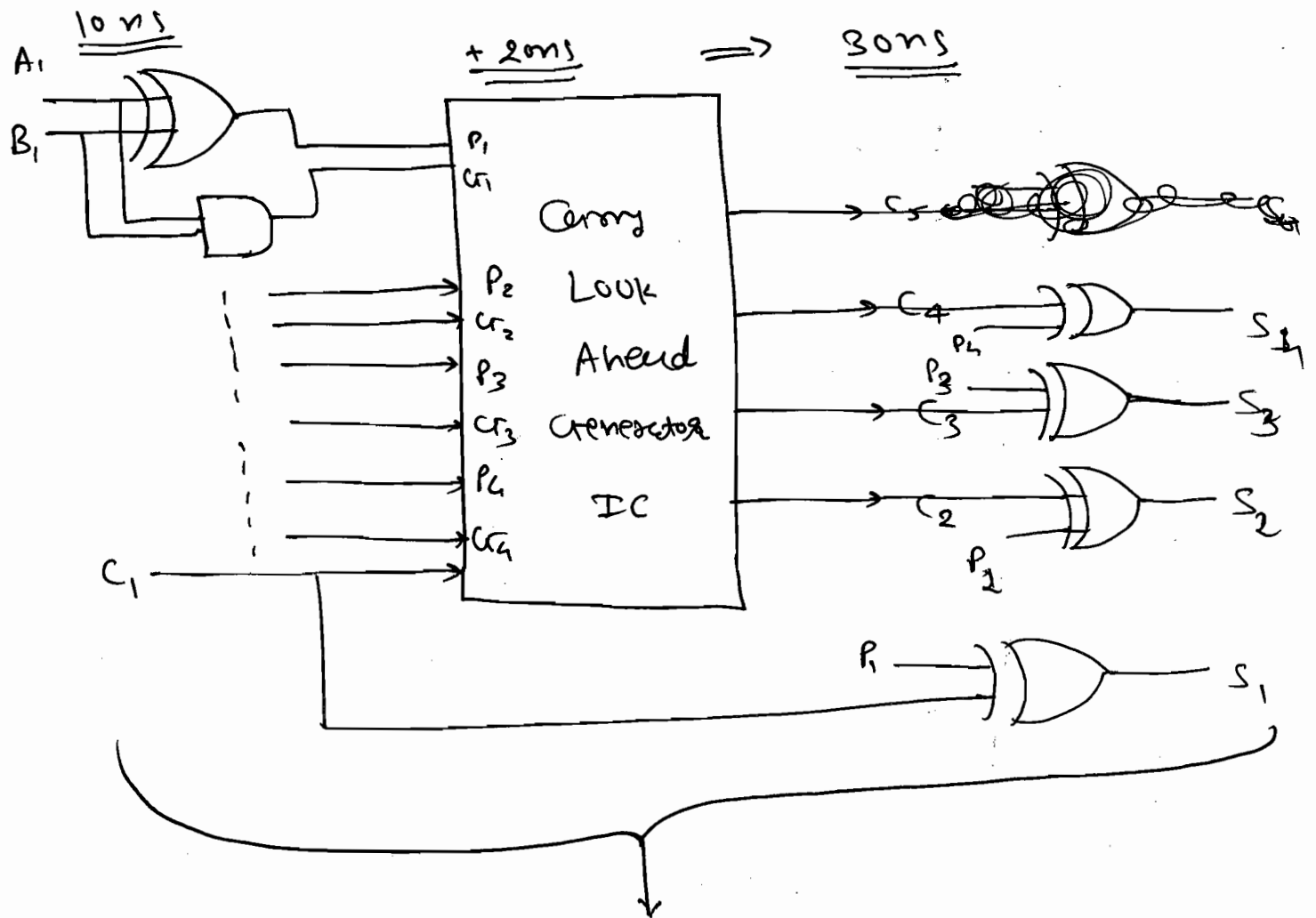
* Advantage:

→ Its Speed of operation is very high and doesn't depend on the size of the adder.

* Disadvantage:

→ It has more hardware complexity. To overcome this we use carry look ahead generator IC.

⇒ Carry Look Ahead generator IC



Carry Look Ahead Adder

$$S_i = A_i \oplus B_i \oplus C_i$$

$$S_i = P_i \oplus C_i$$

$$S_1 = P_1 \oplus C_1$$

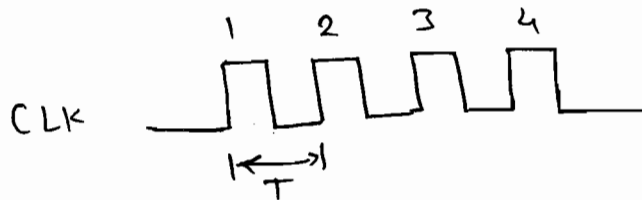
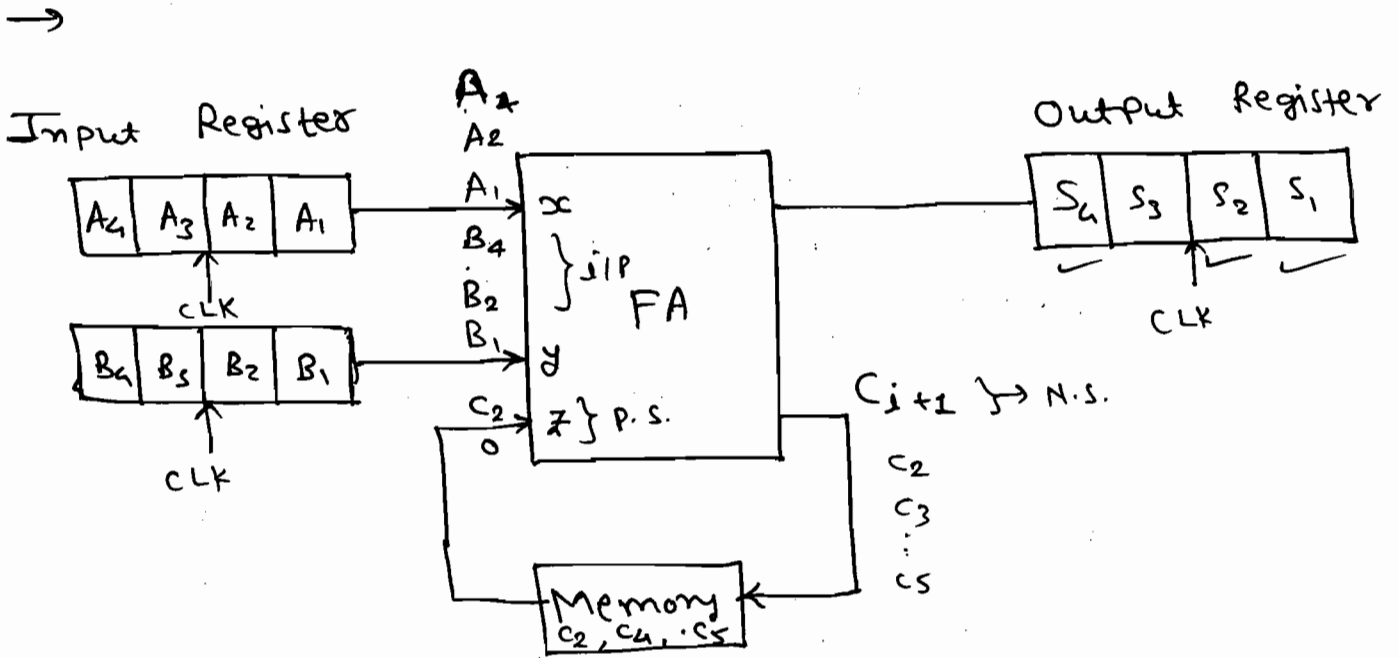
$$S_2 = P_2 \oplus C_2$$

$$S_3 = P_3 \oplus C_3$$

$$S_4 = P_4 \oplus C_4$$

(3) Serial Adder: (sequential CKT)

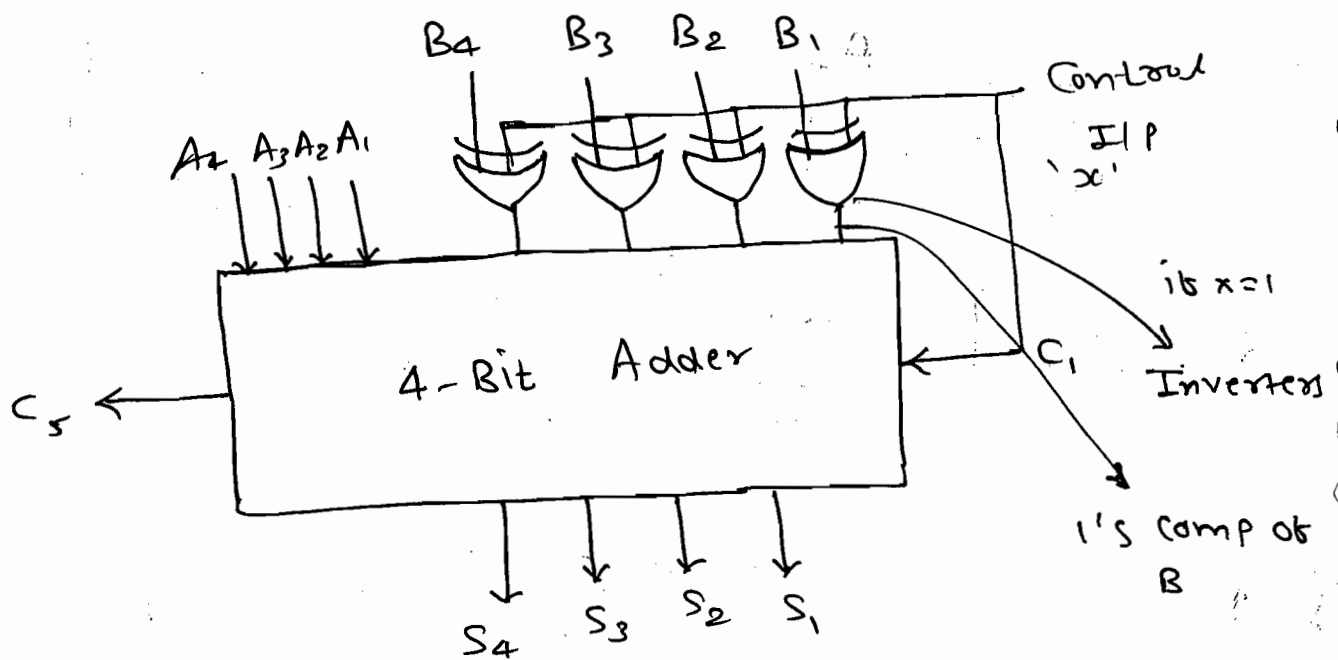
75



$T = \text{CLK period}$

Time for 4-bit serial addition = $4T$

* 4-bit Parallel Adder/Subtractor



(1) If $x=1 \Rightarrow$ Ex-OR gate $\rightarrow$ Inverters

$\rightarrow A + 1's \text{ Comp of } B + (C_1=1)$

$\rightarrow A + (2's \text{ Comp of } B)$

$\Rightarrow \boxed{A - B} \rightarrow \boxed{\text{Binary Subtraction}}$

(2) If $x=0 \Rightarrow$ Ex-OR gate $\Rightarrow$ Buffer

$\rightarrow A + B + (C_1=0)$

$= \boxed{A + B} \rightarrow \boxed{\text{Binary Adding}}$

Ex-1 Determine the f^n of the above 77 circuit if the inputs are as given below:

① $x=1$; $A = \text{Ex-3 code}$; $B = 0011 \Rightarrow \text{Function}=?$

$$\begin{aligned}\Rightarrow A - B &= \text{Ex-3} - 0011 \\ &= \text{Ex-3} + (1100) + 1 \\ &= \text{BCD code.}\end{aligned}$$

Ex-3 to BCD code converter.

$\therefore$ let, $\text{Ex-3} \xrightarrow{\text{or } 4} 0111 \Rightarrow 0100 \rightarrow \text{BCD}$

$\uparrow$
Ex-3

$$\begin{array}{r} 111 \\ 0111 \\ + 1100 \\ + 1 \quad (\because C=1) \\ \hline 0100 \end{array} \Rightarrow 4_{10} = (0100)_{\text{BCD}}$$

② $x=1$, $A = 1001$; $B = \text{BCD} \Rightarrow \text{Function}=?$

Ans: $A = 1001$, let, $B = \text{BCD of } 3$
 $B = 0011$.

$$\begin{array}{r} A \\ + 1 \\ \hline A - B = \begin{array}{r} 1001 \\ + 1100 \\ + 1 \\ \hline 10110 \end{array} \end{array}$$

$= 6_{10}$ which is 9's comp. of 3_{10}

$\therefore$ So, BCD to 9's Comp. of BCD.

$$A + 1's \text{ comp of } B + C_1 = 1.$$

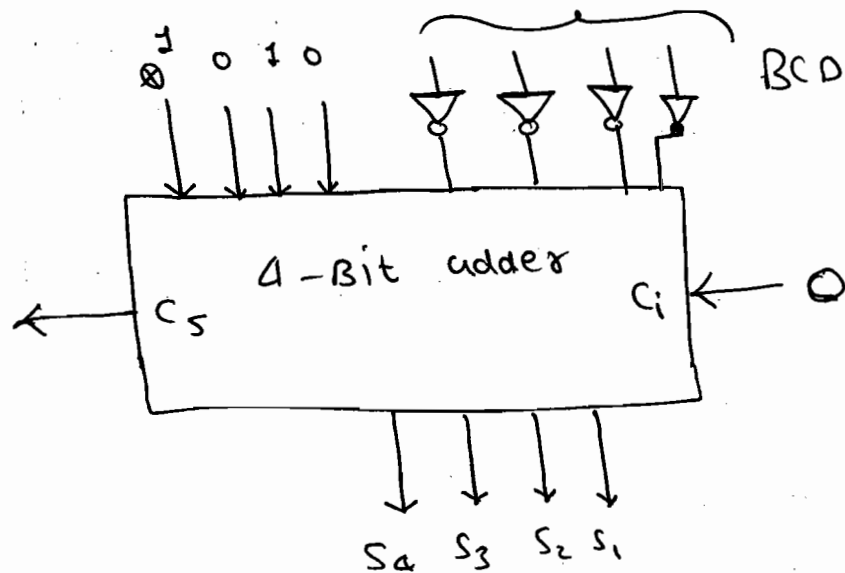
$$= A + (2's \text{ comp of } B).$$

$$= A - \text{BCD.}$$

$$= 9_{10} - \text{BCD}$$

$$= 9's \text{ complement of BCD.}$$

(c)



$$\Rightarrow 1010 + 1's \text{ Comp. of } BCD \quad \boxed{C_i = 0}$$

$$= \cancel{1001} + 1 + 1's \text{ Comp. of } BCD$$

$$\Rightarrow 1001 + 2's \text{ Comp. of } BCD.$$

$$= 9_{10} - BCD.$$

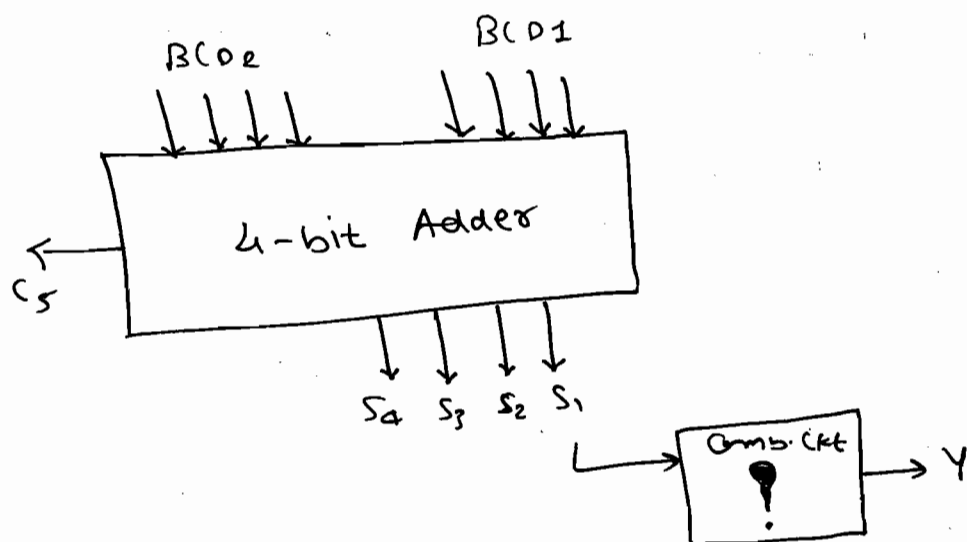
→ 9's Complement of CKt .

NOTE: The above, 9's Complement CKt can be

Converted to a 10's Complement either
by choosing $C_i = 1$ (or) by choosing

A value as $A = 1011$

* BCD Adder



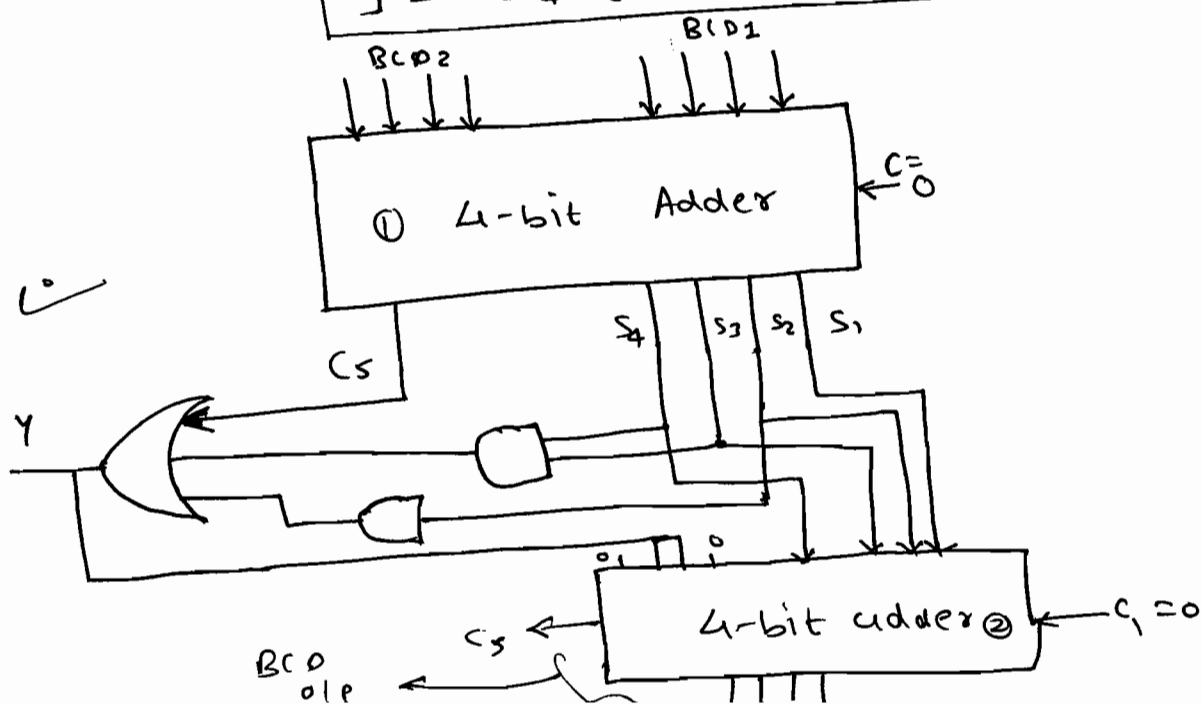
* Output is invalid BCD if

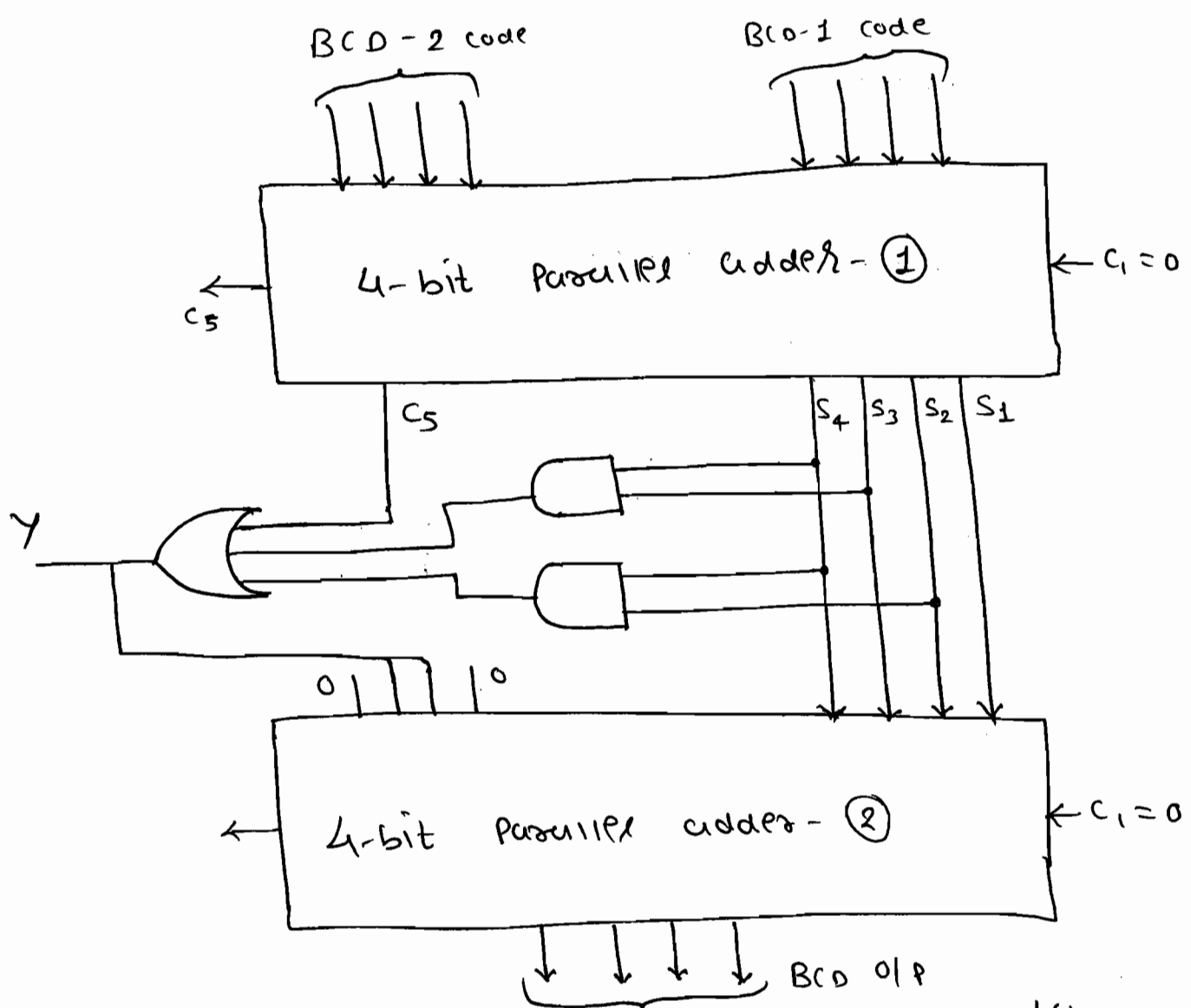
$$S_4S_3S_2S_1 > 9 \text{ (or) } C_5 = 1.$$

S_2S_1	00	01	11	10
S_4S_3				
00				
01				
11	1	1	1	1
10			1	1

→ i.e. output is invalid if

$$Y = S_4S_3 + S_4S_2 + C_5$$



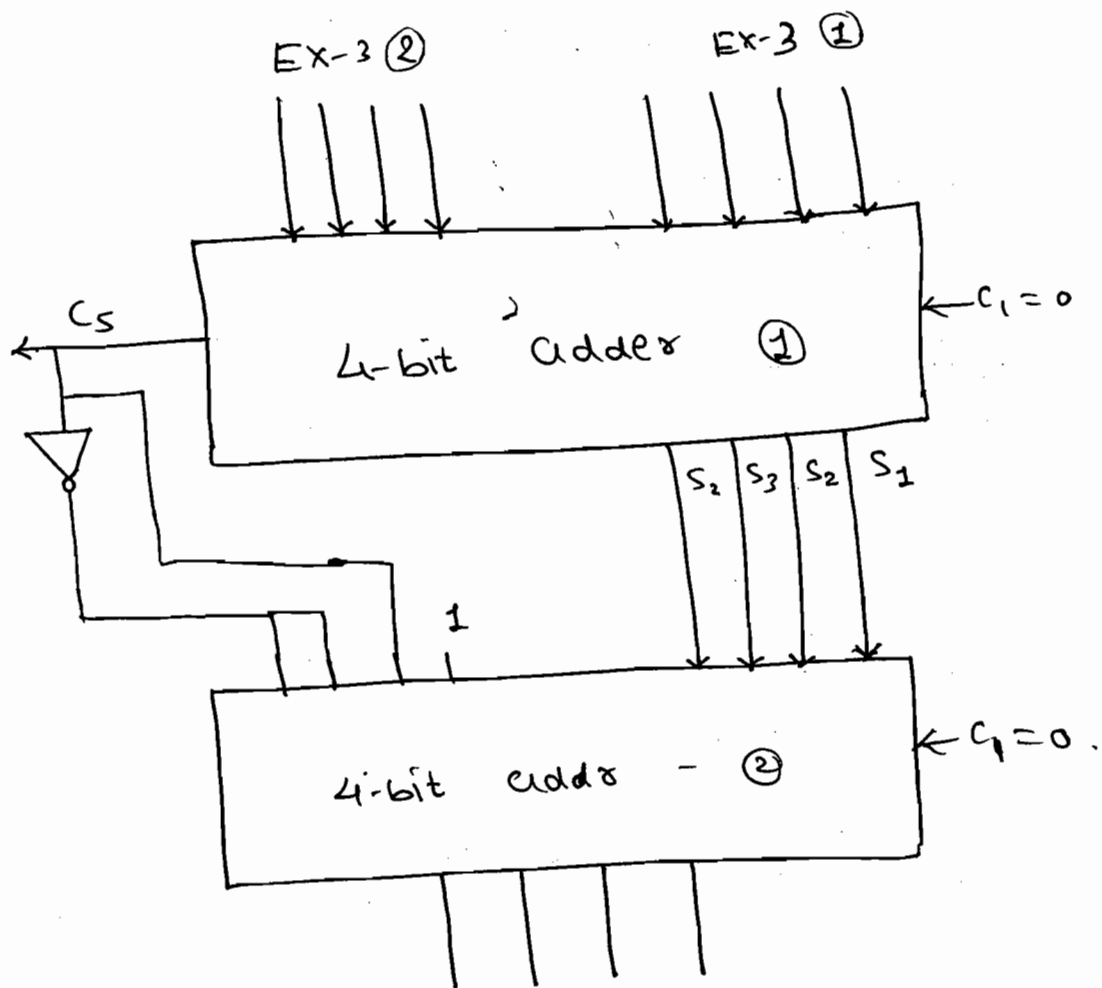


→ Hence, one BCD adder requires ^{two} 4-bit parallel adders

* Ex-3 Adder

(1) If $C_5 = 1 \Rightarrow$ Add $\begin{array}{ccc} \bar{C}_5 & \bar{C}_5 & C_5 \\ 0 & 0 & 11 \\ \hline \end{array}$

(2) If $C_5 = 0 \Rightarrow$ Subtract 0011
 $=$ Add 2's Comp of 0011
 $=$ Add $\begin{array}{ccc} \bar{C}_5 & \bar{C}_5 & C_5 \\ 1 & 1 & 01 \\ \hline \end{array}$



* 2-Bit Magnitude Comparator:

→ The truth table of n bit magnitude Comparator is not preferred for the design as the no. of rows in the table is 2^{2n} .

n -bit Mag Comparator $\rightarrow$ No. of rows = 2^{2n}
 $= 2^{2n}$

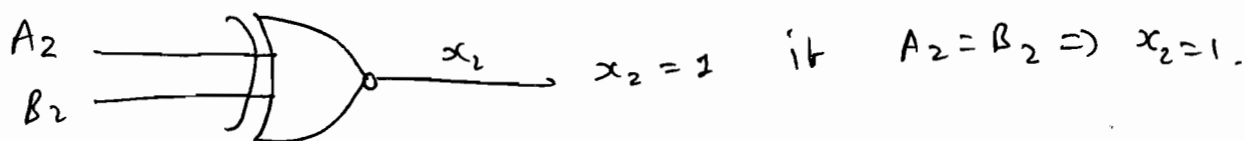
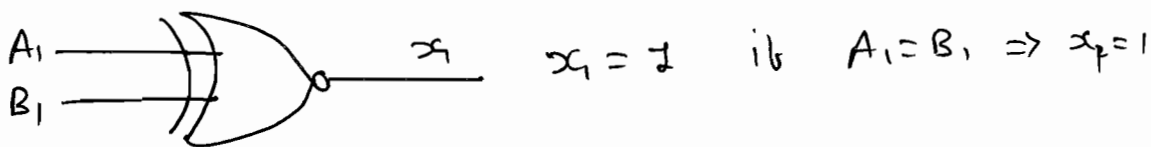
*Q → How many possible way in 2 bit mag. Comparator that $A > B$?

Ans:

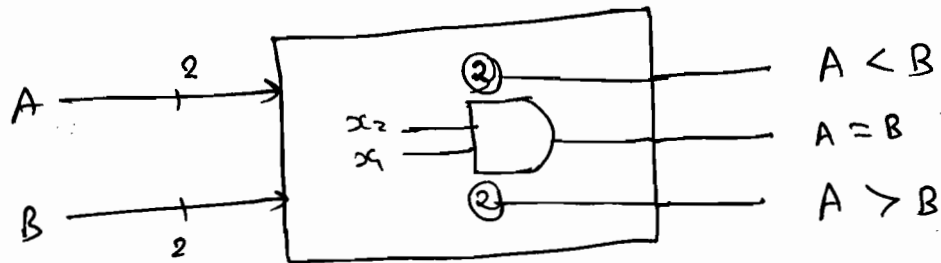
$A_2 A_1$	$B_2 B_1$	
0 0	—	
0 1	0 0	$\rightarrow$ ①
1 0	0 0 0 1	$\rightarrow$ ②
1 1	0 0 0 1 1 0	$\rightarrow$ ③

So, 6 possibility that $A > B$.

*



- ① $A=B$ if $A_2=B_2$ and ~~$x_2=1$~~ $A_1=B_1$ 83
i.e. $A=B$ if $x_2=1$ and $x_1=1$.
i.e. $A=B$ if $x_1 \cdot x_2 = 1$.

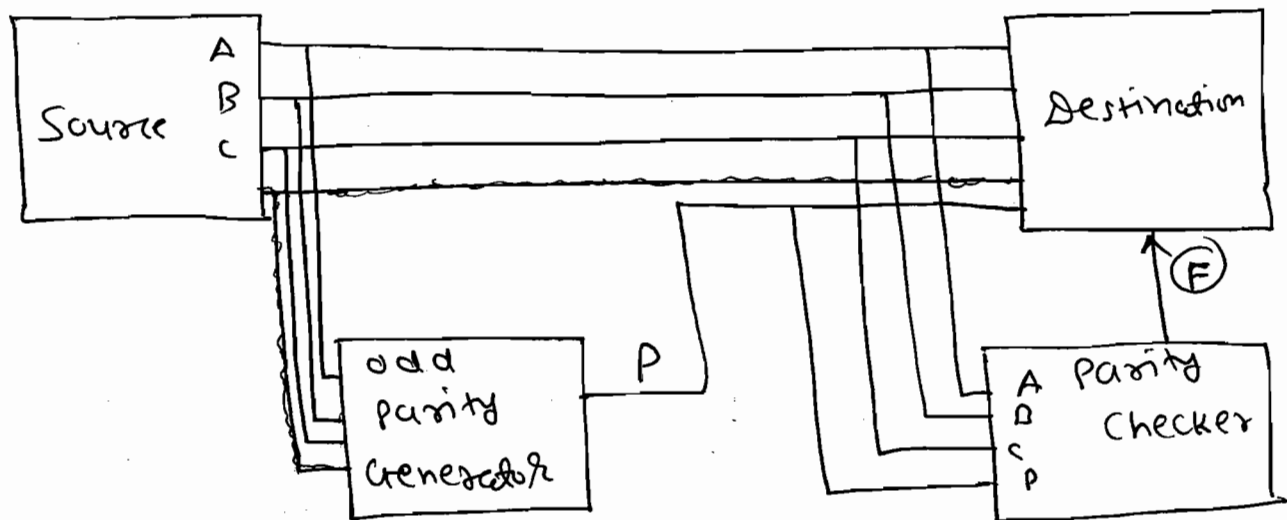


$$A = A_2 B_1$$

$$B = B_2 B_1$$

- ② $A > B$ if $A_2 > B_2$ (or) $A_2 = B_2$ and $A_1 > B_1$.
i.e. $A > B$ if $A_2 \bar{B}_2 + x_2 \cdot A_1 \bar{B}_1 = 1$ — ②
- ③ $A < B$ if $A_2 < B_2$ (or) $A_2 = B_2$ and $A_1 < B_1$.
i.e. $A < B$ if $\bar{A}_2 B_2 + x_2 \cdot \bar{A}_1 B_1 = 1$ — ③

* Odd parity generator and parity checker.



P is chosen so that
 $A, B, C, P \rightarrow \text{odd parity.}$

$F = 1$ if even parity
 $F = 0$ if odd parity
 occurs for $A, B, C, P.$

$\rightarrow$ 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

$$P(A, B, C) = \sum m(0, 3, 5, 6)$$

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

$$P = A \oplus B \oplus C$$

(or)

$$P = A \odot B \oplus C$$

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

$$F = \sum m(A, B, C, P)$$

$$= \sum m(0, 3, 5, 6, 9, 10, 12, 15).$$

		C D			
		00	01	11	10
A B	00	1		1	
	01		1		1
	11	1		1	
	10		1		1

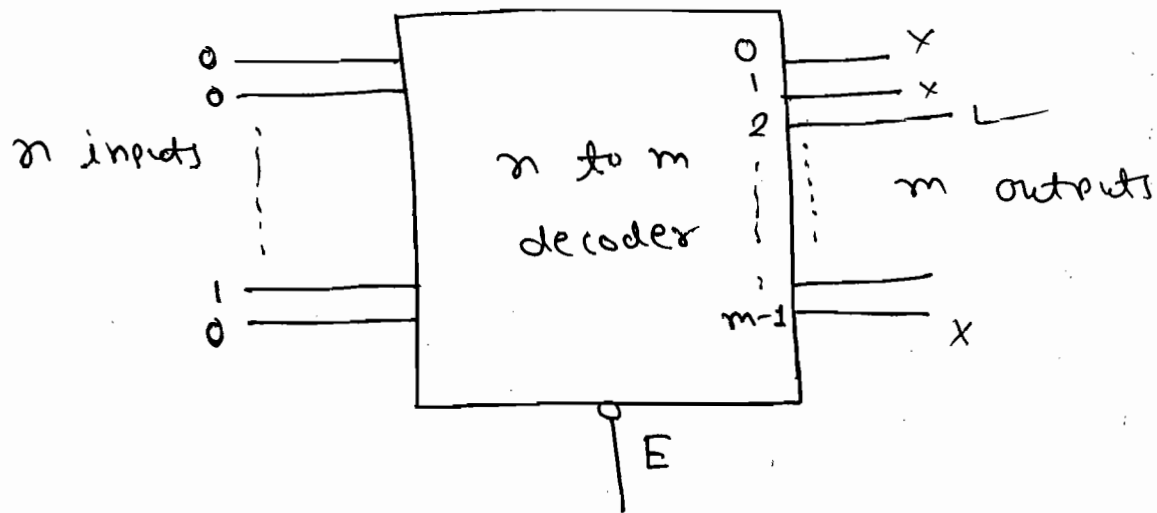
* All minterms have
Even no of 0's.

$$\Rightarrow F = A \oplus B \oplus C \oplus P.$$

* Decoder:

→ It converts the binary information on i/p lines to one of many o/p lines.

→ n to m decoder = (1 out of m decoder).

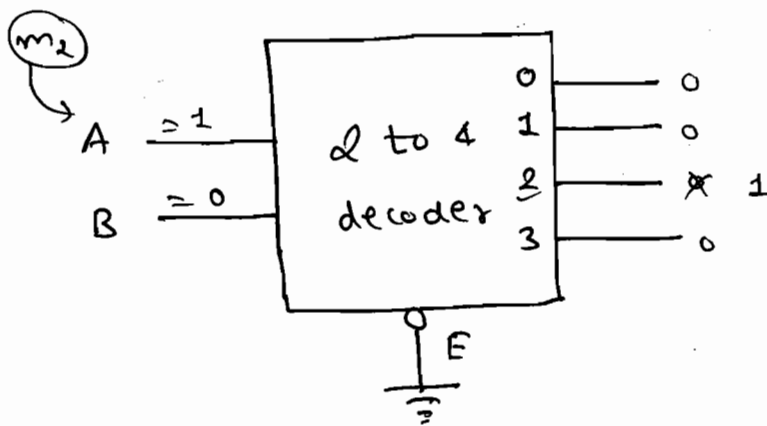


$E=0 \Rightarrow$ Decoder is enabled

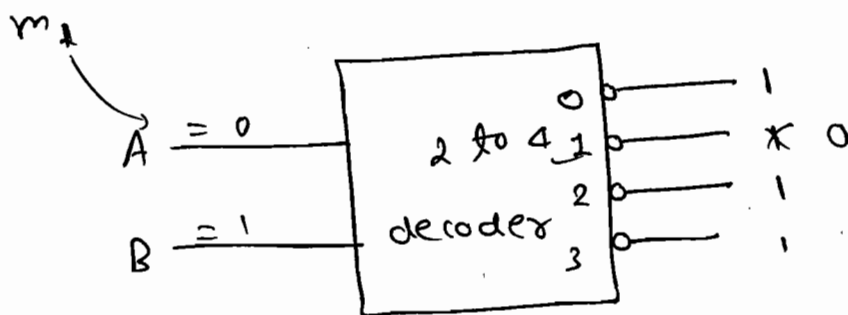
$E=1 \Rightarrow$ Decoder is disabled

$$m \leq 2^n$$

* 2 to 4 decoder (Active High O/p).

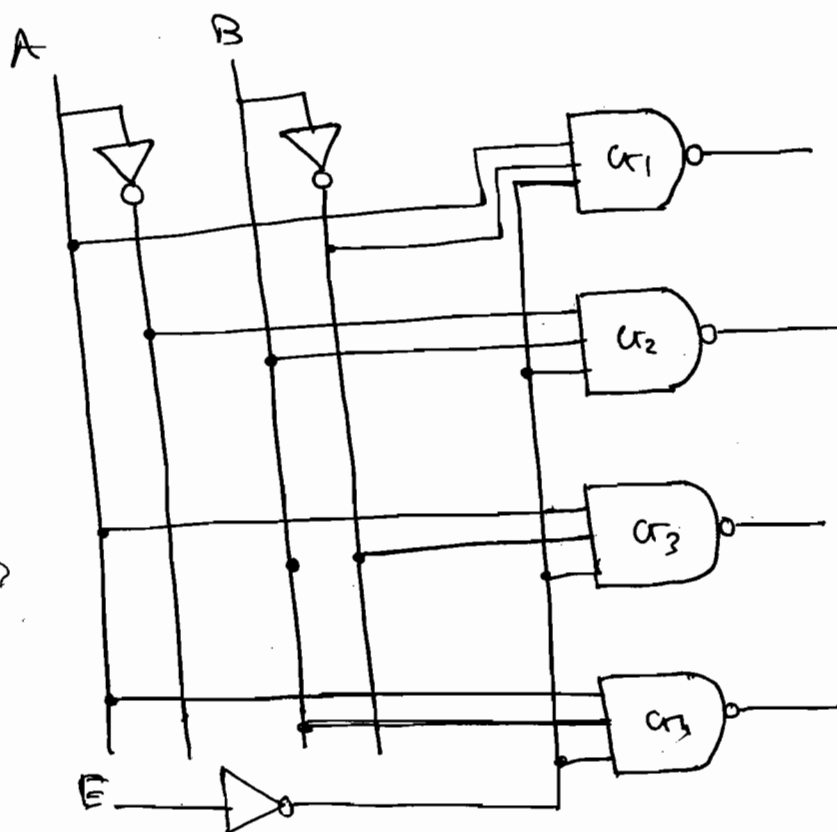


* 2 to 4 decoder (Active Low O/p).



Most widely used in practice. because it generate less noise compare to Active High Decoder.

⇒ Internal circuits:



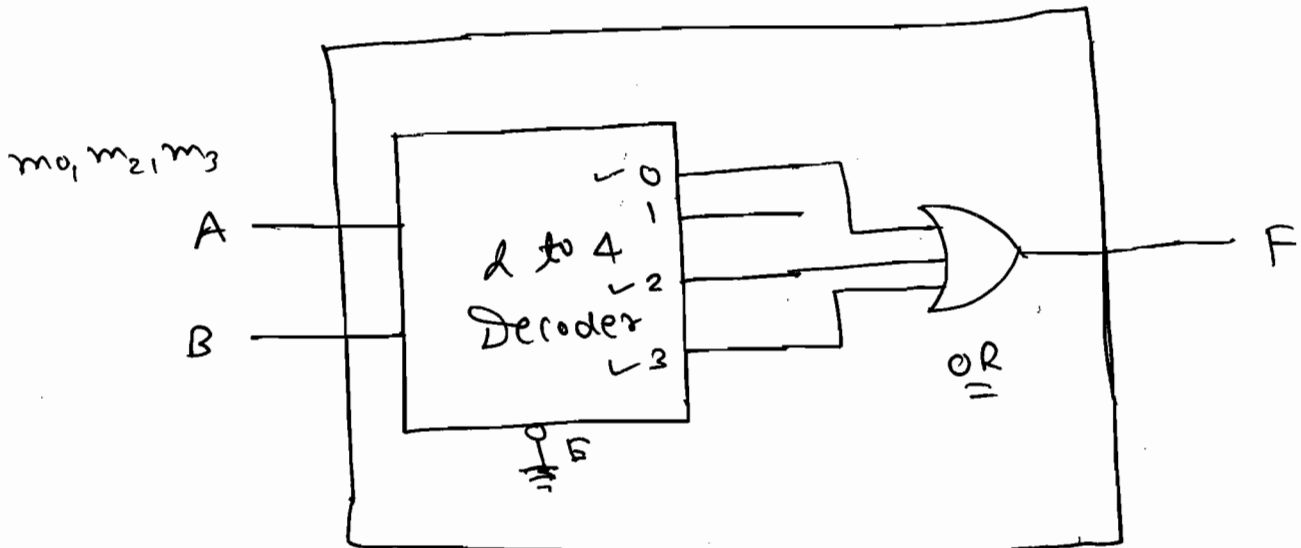
* Decoders
⇒ AND, NAND gates.

Ex-1 Implement $F(A,B) = \sum m(0,2,3)$ using
a decoder (a) with active high o/p.s
(b) with active Low o/p.s.

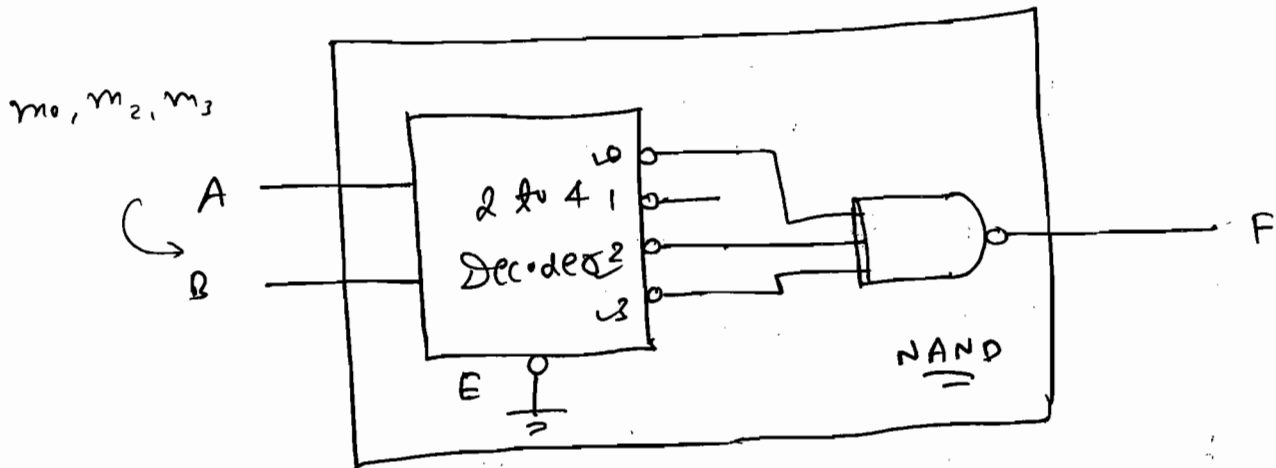
Ans: (a) With active high o/p.

A	B	F
0	0	1
0	1	0
1	0	1
1	1	1

m_0
 m_2
 m_3



(b) With active Low o/p.

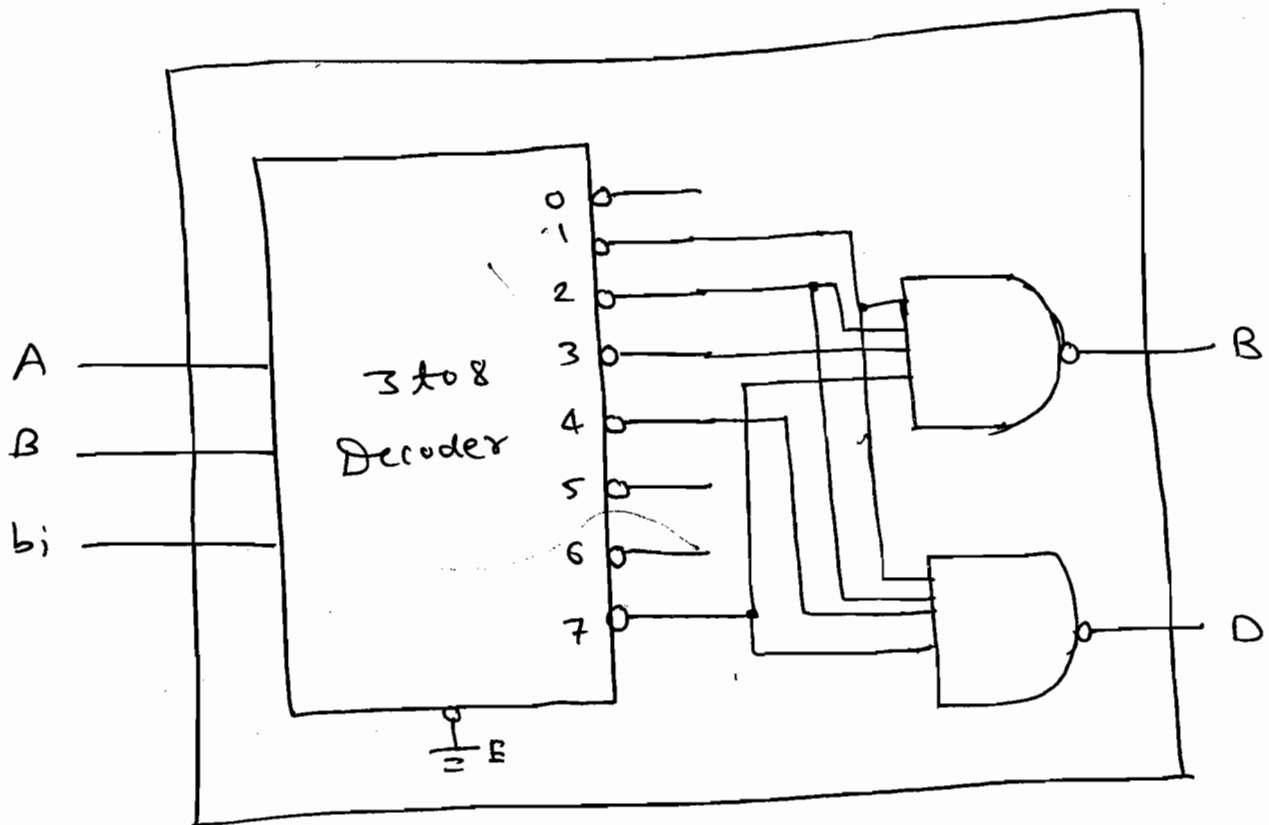


Ex-2 Implement a full Subtractor using a decoder with active Low outputs.

Ans:

$$D(A, B, b_i) = \sum m(1, 2, 4, 7).$$

$$B(A, B, b_i) = \sum m(1, 2, 3, 7).$$

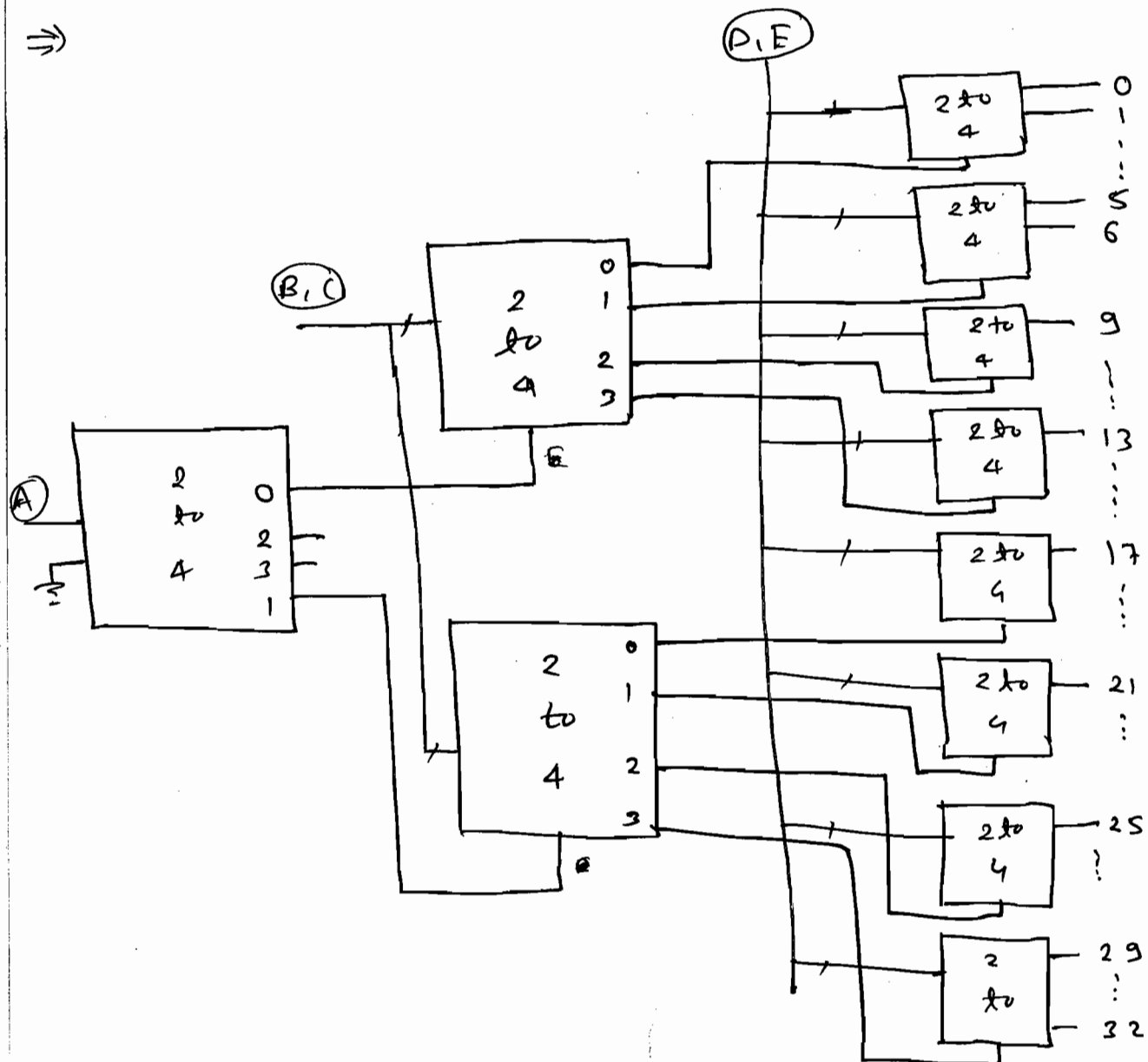


Ex-3 How many 2 to 4 decoder required to construct 1 out of 32 decoder.

Ans: Short cut:

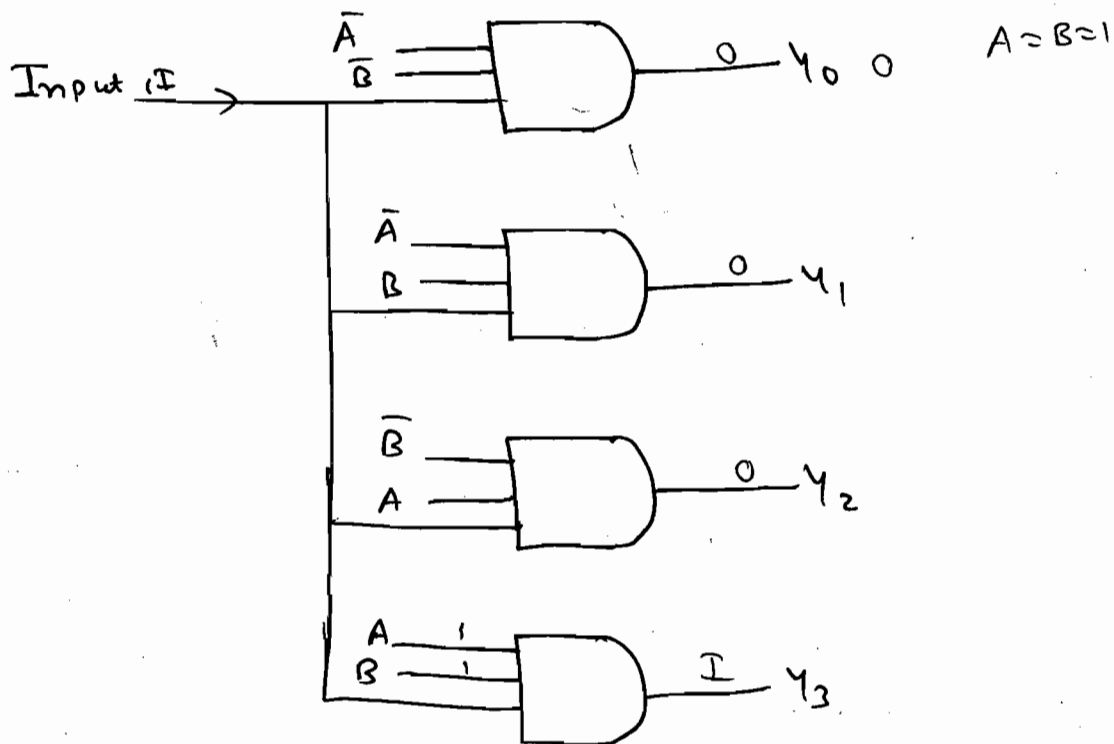
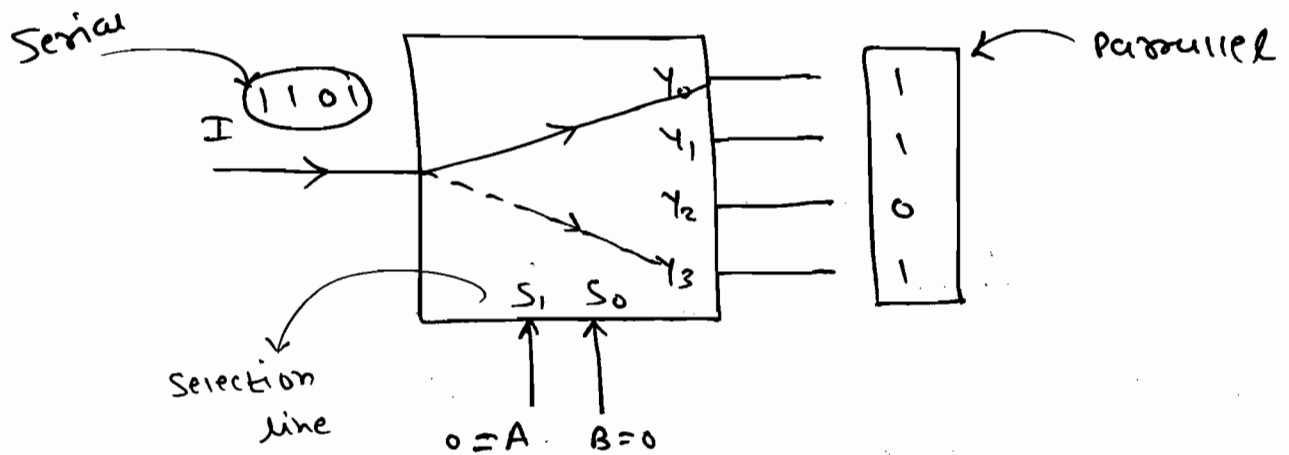
Required No. of O/P	Given no. of o/p	No. of Decoders
$\frac{32}{4} = 8$		8
$\frac{8}{4} = 2$		+ 2
$\frac{2}{4} = 0.5 \approx 1$		+ 1
		<u>11</u>

3 levels.



* Demultiplexer (one to many ckt, Serial to parallel converter). 91

1:4 Demux



→ A Demux is similar to a decoder.

→ A 1:4 Demux is converted to a 2 to 4

Decoder by making two changes.

(1) Selection of Demux are converted to the inputs of 2 to 4 decoder.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840

3



1



1

C.

11)

$$A = 0_4 + 0_5 + 0_6 + 0_7.$$

$$B = 0_2 + 0_3 + 0_6 + 0_7.$$

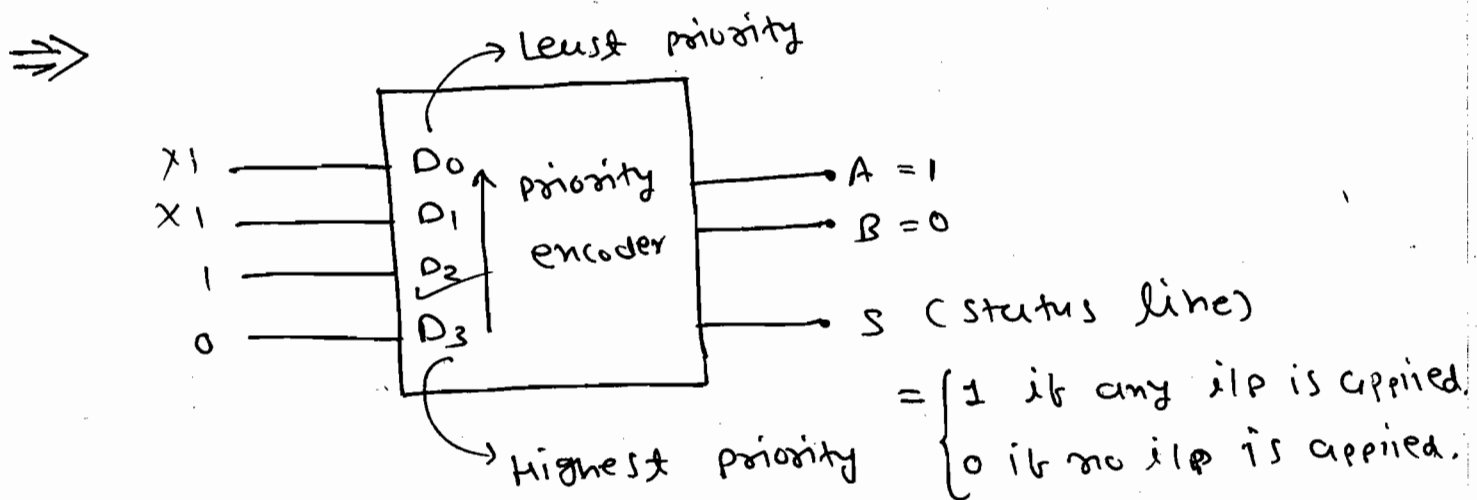
$$C = 0_1 + 0_3 + 0_5 + 0_7.$$

NOTE:

→ The encoder follows the OR Logic.

→ The limitation of encoder is it can't performed the coding if more than 1 input is active simultaneously. To overcome this, we use priority encoder.

* Priority Encoder:



⇒

	D_3	D_2	D_1	D_0	A	B	S
$m_0 \leftarrow$	0	0	0	0	0	0	0
$m_1 \leftarrow$	0	0	0	1	0	0	1
$m_2, m_3 \leftarrow$	0	0	1	X	0	1	1
$m_4 \text{ to } m_7 \leftarrow$	0	1	X	X	1	0	1
$m_8 \text{ to } m_{15} \leftarrow$	1	X	X	X	1	1	1

(b) Priority Encoder table.

$$\rightarrow B(D_3, D_2, D_1, D_0) = \sum m(2, 3, \cancel{4}, \cancel{5}, \cancel{6}, \cancel{7}, 8, 9, 10, \dots, 15).$$

		$D_1 D_0$			
		00	01	11	10
$D_3 D_2$	00			1	1
	01				
	11	1	1	1	1
	10	1	1	1	1

$$B = D_3 + \overline{D_2} D_1$$

$$\rightarrow A(D_3, D_2, D_1, D_0) = \sum m(4, 5, 6, 7, \dots, 15).$$

		$D_1 D_0$			
		00	01	11	10
$D_3 D_2$	00				
	01	1	1	1	1
	11	1	1	1	1
	10	1	1	1	1

$$A = D_2 + D_3$$

$$\rightarrow S = \sum m(1, 2, 3, \dots, 15).$$

		$D_1 D_0$			
		00	01	11	10
$D_3 D_2$	00	1	1	1	1
	01	1	1	1	1
	11	1	1	1	1
	10	1	1	1	1

$$\therefore S = \overline{D_3} + \overline{D_2} + \overline{D_1} + \overline{D_0}$$

$$S = D_3 + D_2 + D_1 + D_0$$

NOTE:

74LS138 $\Rightarrow$ $\begin{cases} 3 \text{ to } 8 \text{ decoder} \\ (\text{Active low outputs}). \end{cases}$

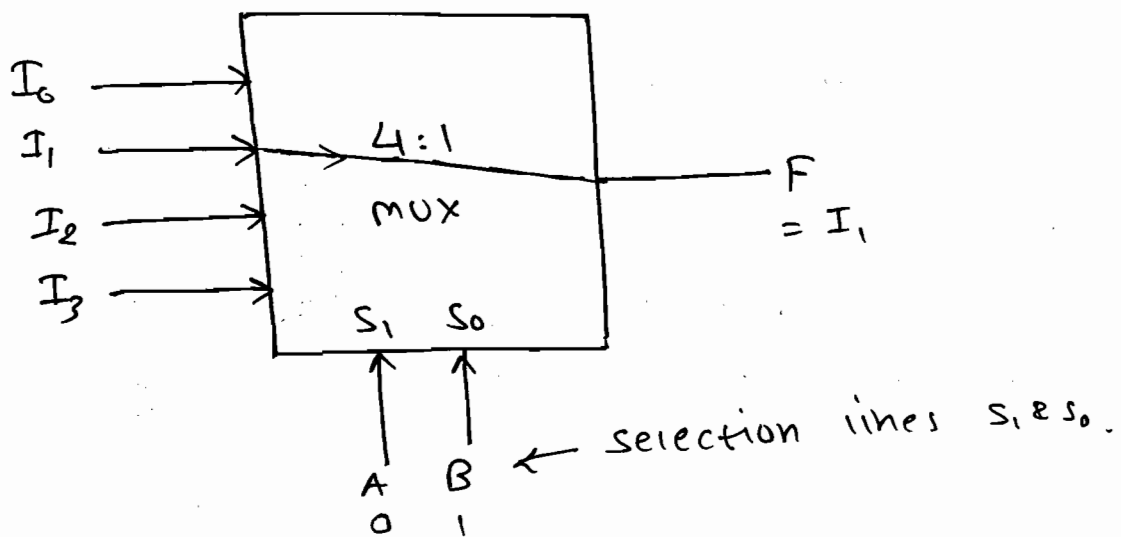
74LS148 $\Rightarrow$ $\begin{cases} 8\text{-input, 3-output} \\ \text{priority encoder.} \end{cases}$

* Multiplexer:

(Many to one circuit,
parallel to serial converter).

*

4:1 MUX.



$\therefore$

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

$\Rightarrow$ 4:1 MUX $\Rightarrow F = \bar{A}\bar{B}I_0 + \bar{A}BI_1 + A\bar{B}I_2 + ABI_3$

$$\therefore F = m_0 I_0 + m_1 I_1 + m_2 I_2 + m_3 I_3.$$

* E.g.: Given $I_0 = I_1 = 1$; $I_2 = I_3 = 0$.

$$\therefore F = m_0 \cdot 1 + m_1 \cdot 1 + 0 + 0.$$

$$\therefore F = m_0 + m_1$$

$$\therefore F = \sum m(0, 1).$$

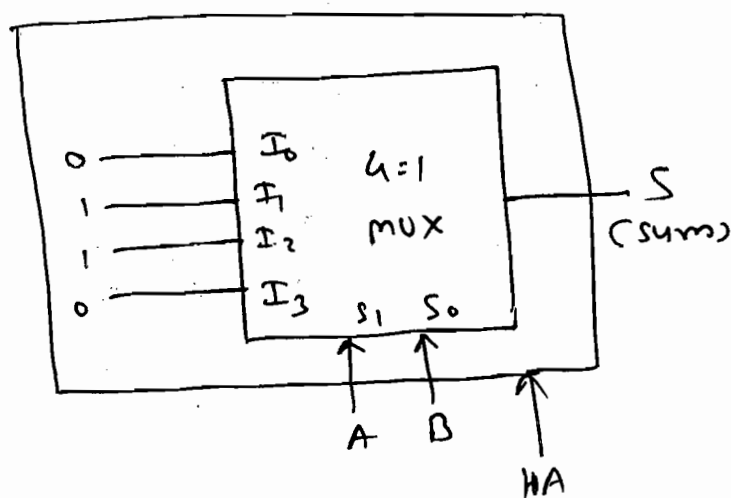
Ex-1 Implement the Sum of half Adder using 4:1 MUX.

Ans:

$$S(A, B) = \sum m(1, 2).$$

$$S(A, B) = m_1 + m_2$$

Choose $I_1 = I_2 = 1$ &
 $I_0 = I_3 = 0$

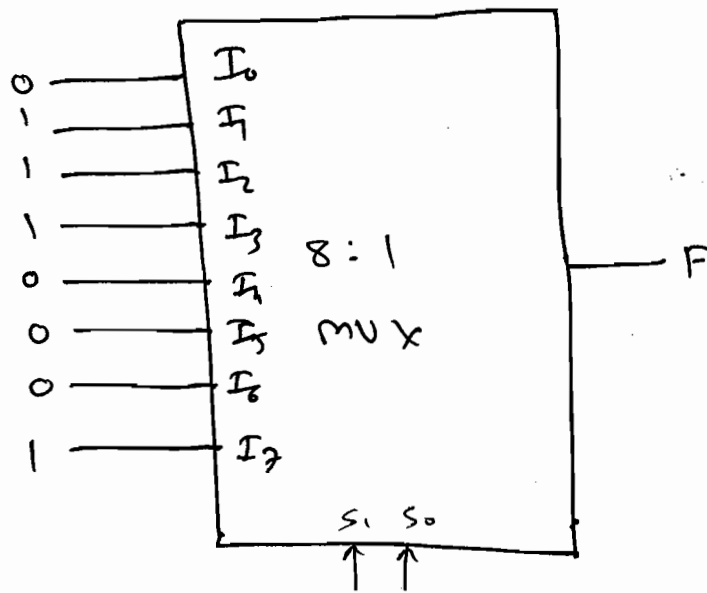


Ex-2 Which of the following is implemented by the following mux.

- (a) Sum output of Full adder.
- (b) Carry output of Full adder.
- (c) Difference output of Full Subtractor.
- ✓ (d) Borrow output of Full Subtractor.

Q Sum up the given circuit.

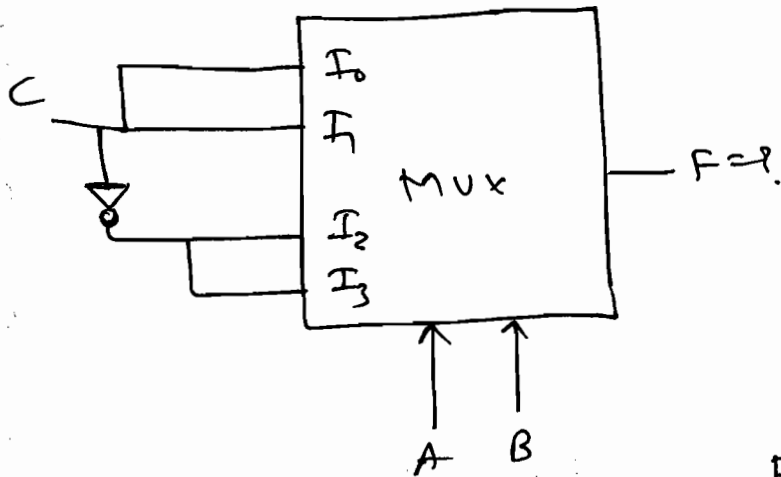
97



$$F = \sum m(1, 2, 3, 7)$$

Ex-2 What are the Logic gates represented by the following mux circuits.

Q



$$F = m_0 C + m_1 C + m_2 \bar{C} + m_3 \bar{C}$$

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

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

$$F = (A \oplus B)C$$

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

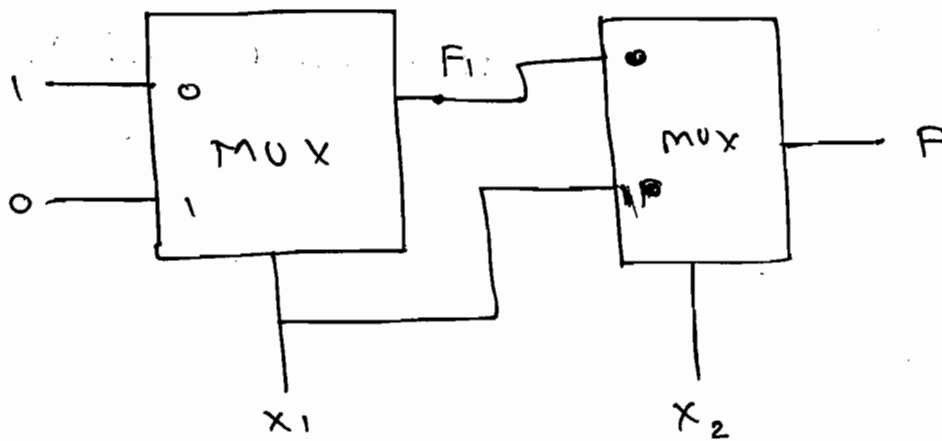
$$F = (\bar{A}\bar{B} + \bar{A}B)C + (A\bar{B} + AB)\bar{C}$$

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

$$F = A \oplus C$$

∴ XOR gate

(b)



$$\therefore F_1 = m_0 = \bar{x}_1$$

$$\therefore F_1 = \bar{x}_1$$

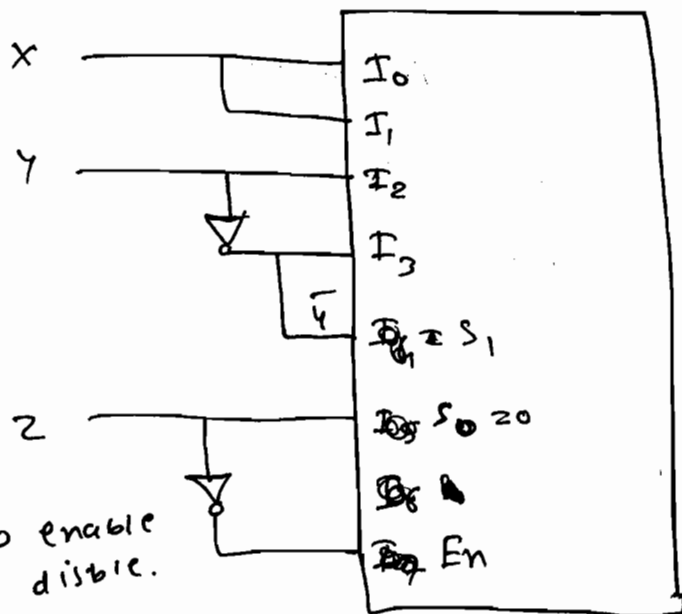
$$\therefore F = F_1 \bar{x}_2 + x_2 x_1$$

$$= \bar{x}_1 \bar{x}_2 + x_2 x_1$$

$$\therefore \boxed{F = x_1 \oplus x_2}$$

Ex-4 Determine the output of the following MUX.

Ans.



$Z=0$ enable
 $Z=1$ disable.

$$F = (m_0 + m_1) X$$

$$+ m_2 Y + m_3 \bar{Y}$$

$$= (\bar{S}_1 \bar{S}_0 + \bar{S}_1 S_0) X$$

$$+ S_1 \bar{S}_0 Y + S_1 S_0 \bar{Y}$$

$$= \bar{S}_1 X + S_1 \bar{S}_0 Y + S_1 S_0 \bar{Y}$$

$$= \bar{S}_1 X + S_1 Y + 0$$

$$= XY + 0$$

$$= XY$$

$\Rightarrow$

$$F = x \cdot y \cdot \bar{z}$$

 $\Rightarrow z=0 \rightarrow \text{enable.}$
 $z=1 \rightarrow \text{disable.}$

Ex-5 How many 4:1 mux required to construct 128:1 MUX.

Ans: $\frac{\text{Required No. of Inputs}}{\text{Given No. of I/P}}$

$$= \frac{128}{4} = 32 \xrightarrow{\text{No. of MUX}} 32$$

$$\downarrow +$$

$$\therefore \frac{32}{4} = 8 \xrightarrow{\text{No. of MUX}} 8$$

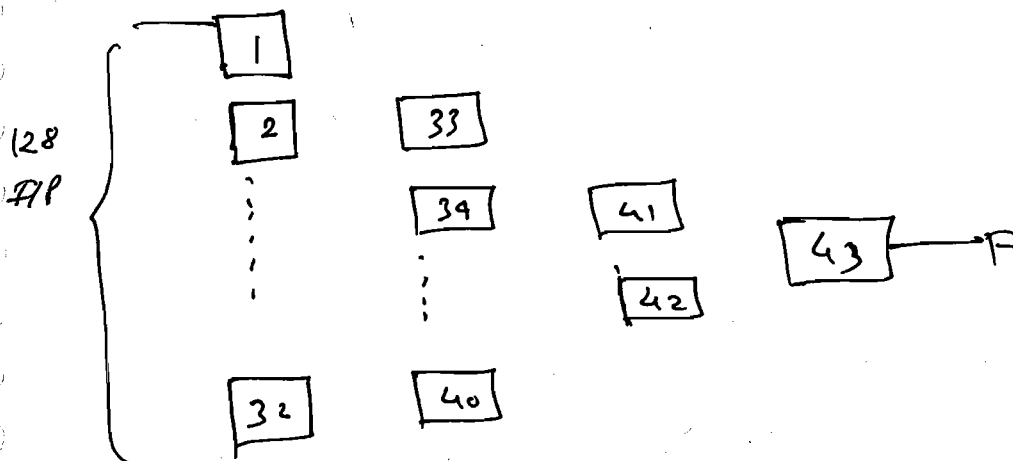
$$\downarrow +$$

$$\frac{8}{4} = 2 \xrightarrow{\text{No. of MUX}} 2$$

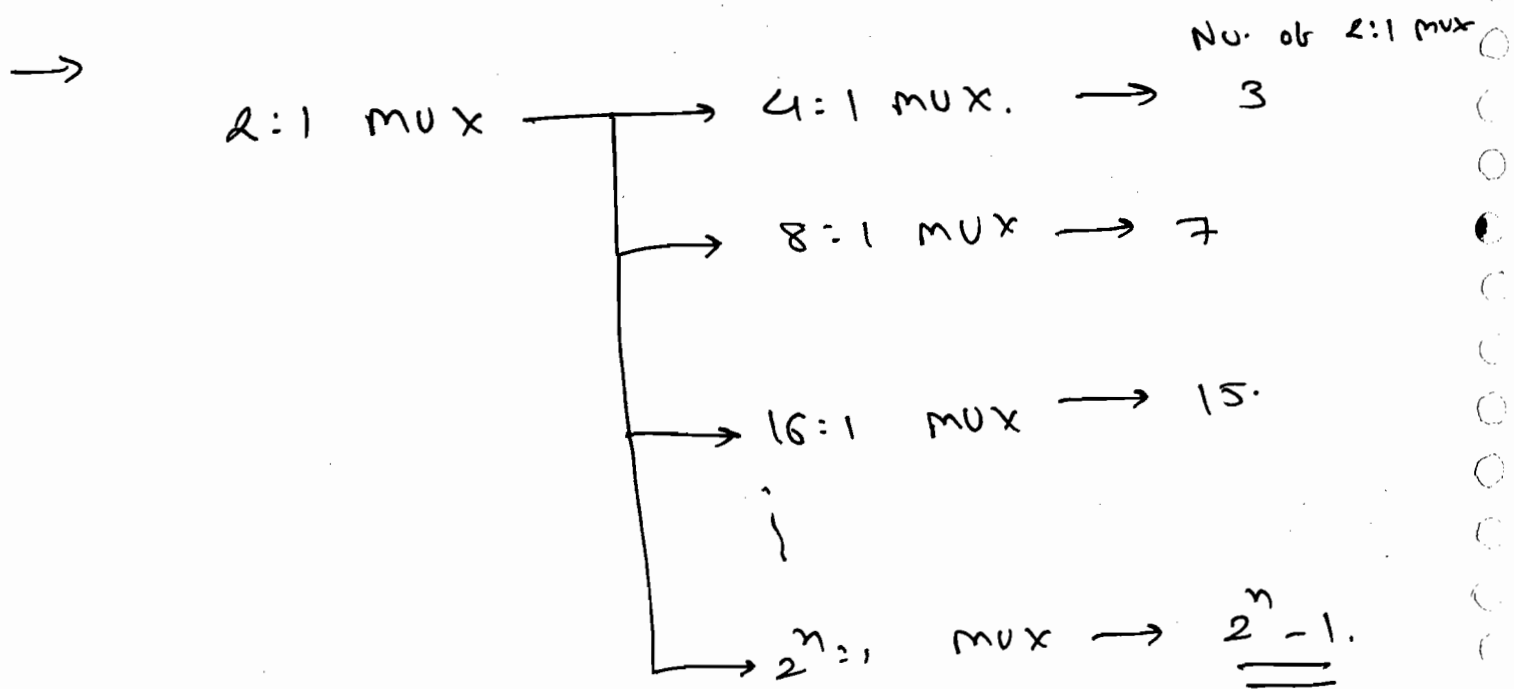
$$\downarrow +$$

$$\frac{2}{4} = 0.5 \xrightarrow{\text{No. of MUX}} 1$$

$$\underline{43}$$



(b) $2:1 \text{ MUX} \longrightarrow 2^n:1 \text{ MUX.}$



So, $\boxed{2^n - 1}$ $2:1$ MUX required for $2^n:1$ MUX.

Ex-5 How many $2:1$ MUX are req. to realize

- (a) AND gate
- (b) OR gate
- (c) EX-OR gate.

Ans:

$$F = \bar{A}I_0 + AI_1$$

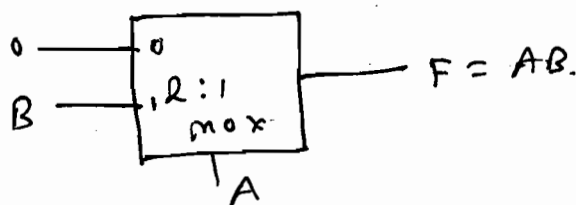
$$F = AB$$

$$F = AB + 0 \cdot \bar{A}$$

$$F = 0 \cdot \bar{A} + B \cdot A$$

$$F = \bar{A}I_0 + AI_1$$

So, $I_0 = 0, I_1 = B.$



(b) OR gate:

$$\therefore F = \bar{A}I_0 + AI_1$$

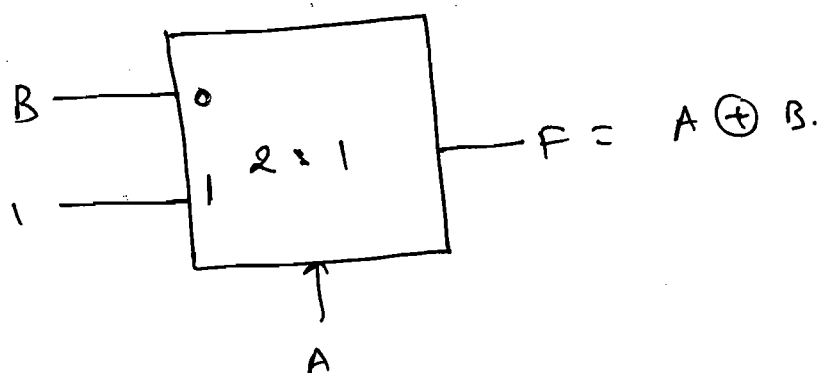
$$F = A + B.$$

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

$$F = \bar{A}B + A \cdot 1.$$

$$\therefore F = \bar{A}I_0 + A \cdot I_1$$

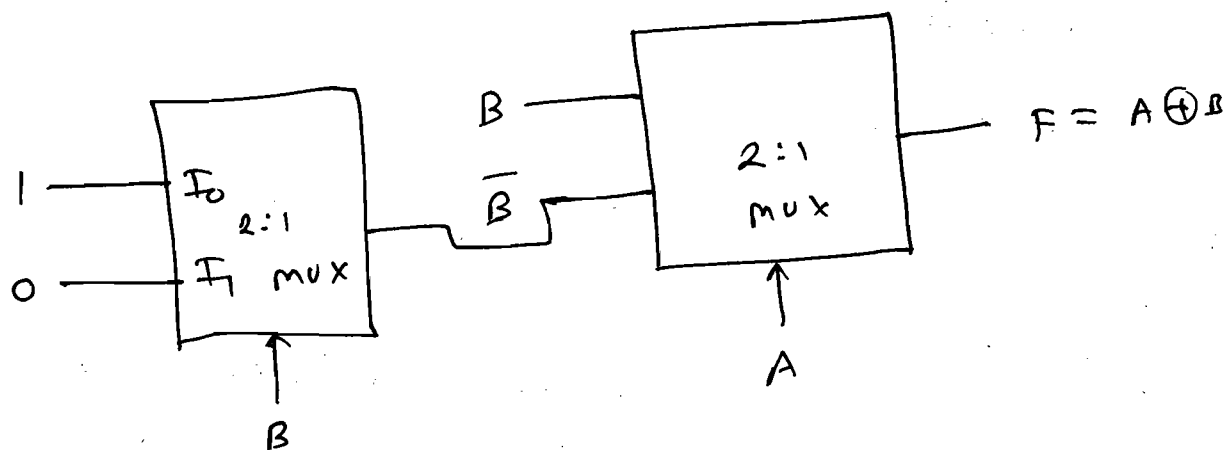
$\therefore$ So, Choose $I_0 = B, I_1 = 1.$



(c) Ex-or gate:

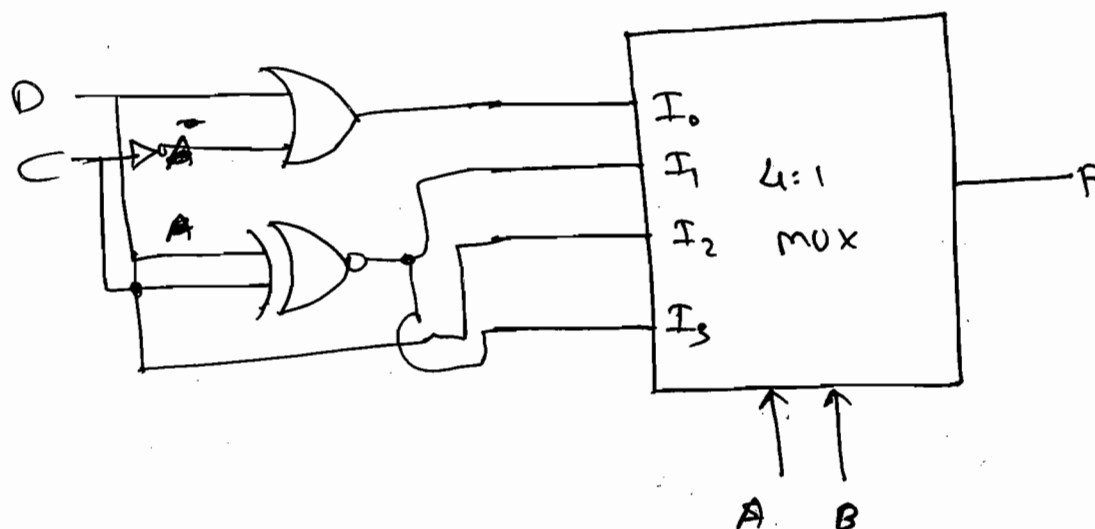
$$\rightarrow F = \bar{A}B + A\bar{B} = A \oplus B.$$

$\uparrow$ $\uparrow$
 I_0 I_1



Ex-6 Implement $F(A, B, C, D) = \sum m(0, 1, 3, 5, 6, 10, 11, 13, 14)$ using 4:1 MUX.

Ans:



Method - 1

AB =		00	01	10	11
		I_0	I_1	I_2	I_3
00	$\bar{C}\bar{D}$	(0)	4	8	12
01	$\bar{C}D$	(1)	(5)	9	(13)
10	$C\bar{D}$	2	(6)	(10)	(14)
11	CD	(3)	7	(11)	15
		$\bar{C}\bar{D} + \bar{C}D + CD = \bar{C} + D$	$C\bar{D}$	C	$C \oplus D$

Method :- 2

If A, B are selection line

AB	CD				
	00	01	11	10	
00	1	1	1		$\rightarrow I_0 = \bar{C}\bar{D} + \bar{C}D + CD = \bar{C} + D$
01		1		1	$\rightarrow I_1 = \bar{C}D + C\bar{D} = C \oplus D$
10		1	1	1	$\rightarrow I_2 = \bar{C}D + C\bar{D} = C \oplus D$
11			1	1	$\rightarrow I_3 = CD + C\bar{D} = C$

→ NOTE: If C & D are selection lines

103

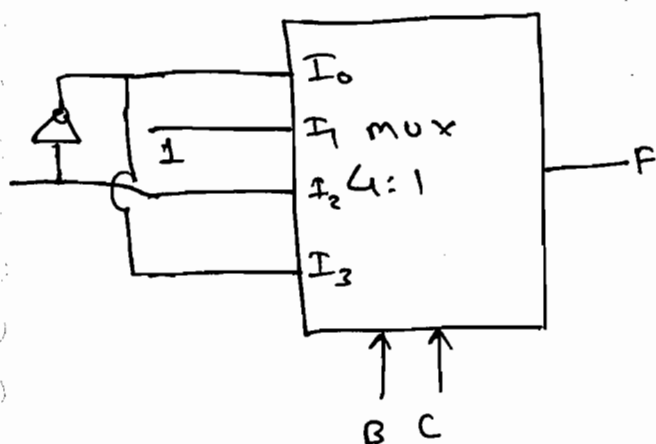
CD \ AB	00	01	11	10
00	1	1	1	
01		1		1
11		1		1
10			1	1

$I_0 = \bar{A}\bar{B}$ $I_1 = \bar{A}B$ $I_2 = A\bar{B}$ $I_3 = AB$
 $I_0 + I_1 = \bar{A}$ $I_2 + I_3 = B$

Ex-1 $F(A, B, C) = \sum m(0, 3, 5, 6)$ using 4:1 mux.

Ans:

Method-I:



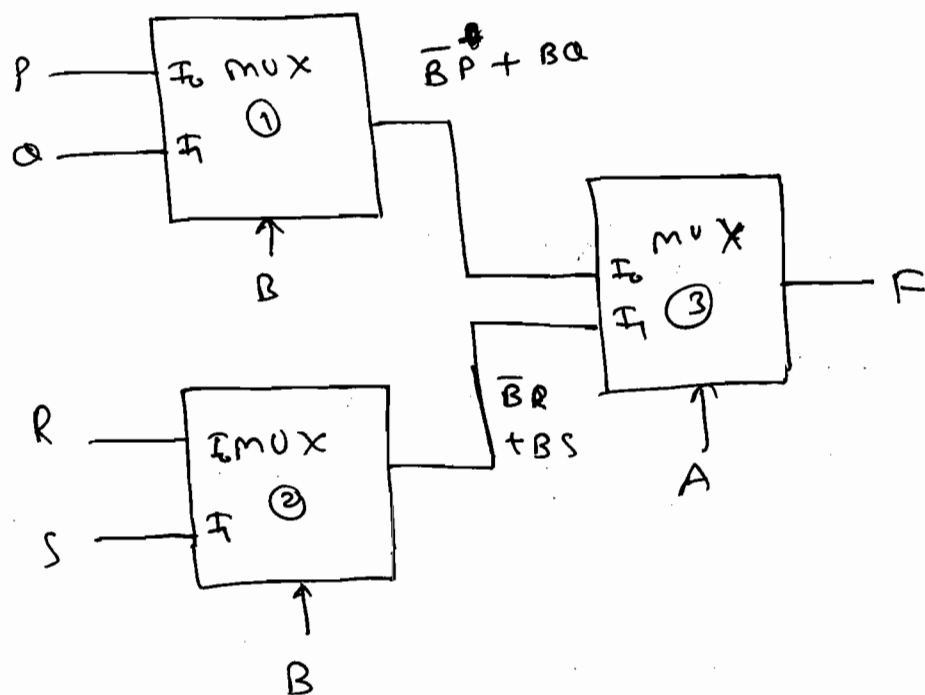
	I_0	I_1	I_2	I_3
$\bar{A}$	0	1	2	3
A	4	5	6	7
	$\bar{A}$	1	A	$\bar{A}$

Method-II (K-Map)

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

$I_0 = \bar{A}$ $I_1 = 1$ $I_2 = \bar{A}$ $I_3 = A$

Ex-2 In the following Mux tree, determine the values of $P, Q, R, S = ?$
 Where, $F(A, B, C) = \sum m(1, 2, 4, 5, 6)$.



$$\therefore F = \bar{A}(\bar{B}P + BQ) + A(\bar{B}R + BS)$$

$$\therefore F = \bar{A}\bar{B}P + \bar{A}BQ + A\bar{B}R + ABS$$

AB		00	01	11	10
AC	0	0	1	1	1
	1	1	3	7	1

$$I_0 = C \quad I_1 = \bar{C} \quad I_2 = \bar{C} \quad I_3 = 1$$

$\uparrow \quad \quad \uparrow \quad \quad \uparrow \quad \quad \uparrow$
 $P \quad \quad Q \quad \quad R \quad \quad S$

$$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + ABC$$

$$\therefore F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A$$

Ex-2 Using $n=1$ MUX we can implement 105
all ' $\log_2 N$ ' variable function and
Some of " $\log_2 N + 1$ " variable functions
[T/F].

Ans: let, $n=1$ MUX

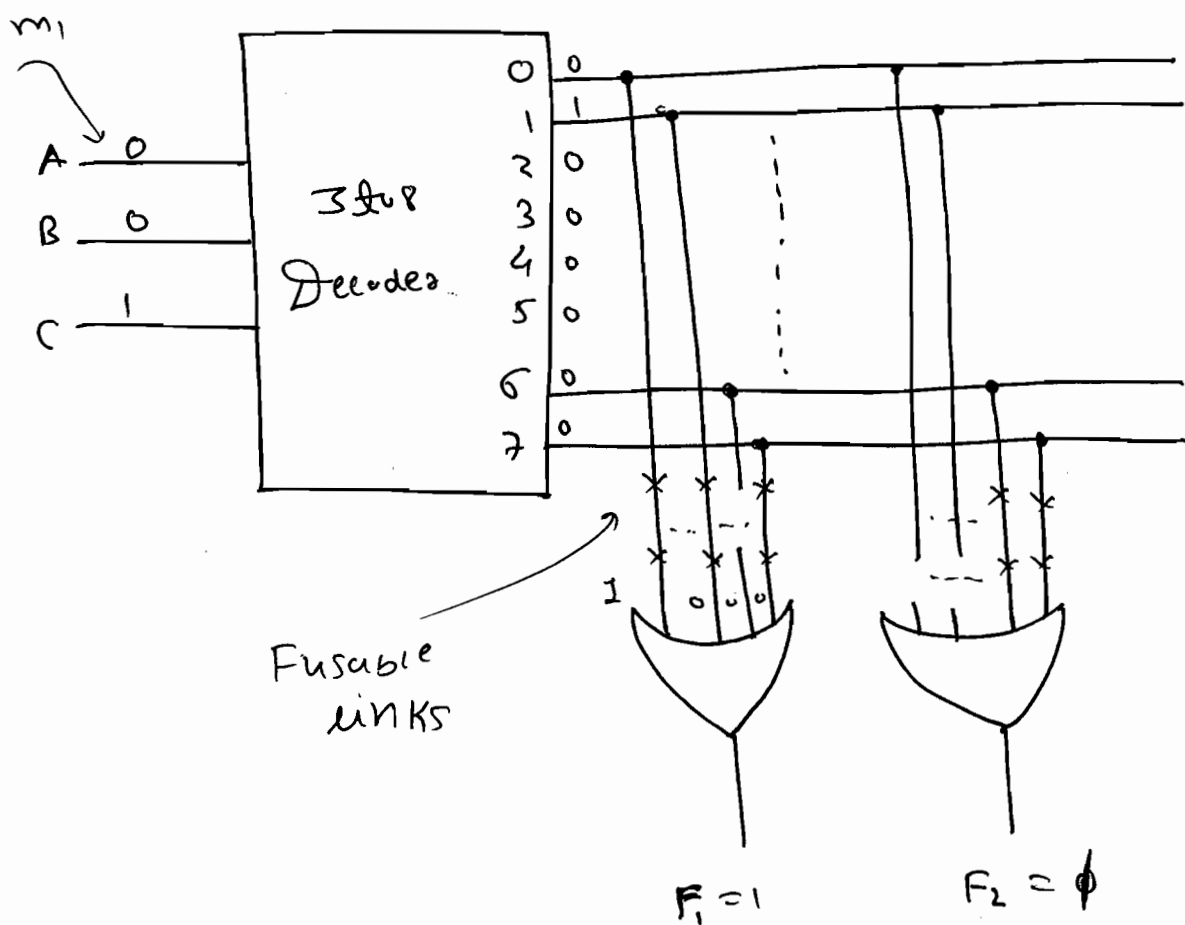
$$\text{So, } \log_2 4 = 2$$

→ True : because the other functions required
external logic gates along with MUX.

* Rom (Read only Memory).

$\Rightarrow$ Decoder + Prog. OR gates } = Rom.
(Fixed AND gates) (encoder)

$\rightarrow$ Rom is a Combinational circuits and we can use it to implement sum of minterms expression as shown in the following example:



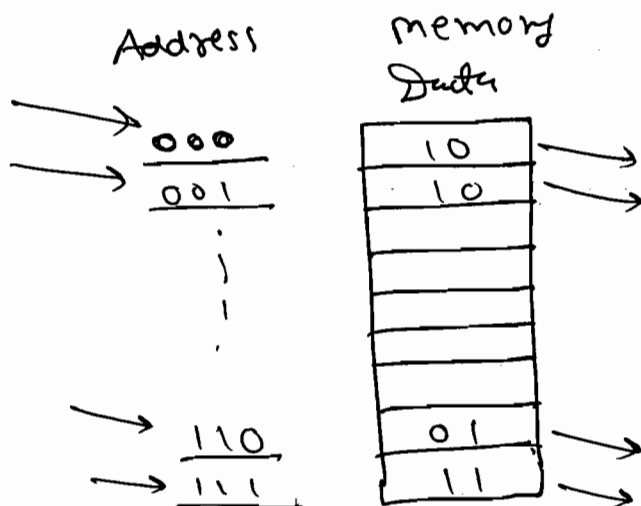
$$F_1(A, B, C) = \sum m(0, 1, 6, 7).$$

$$F_2(A, B, C) = \sum m(2, 3, 4, 5).$$

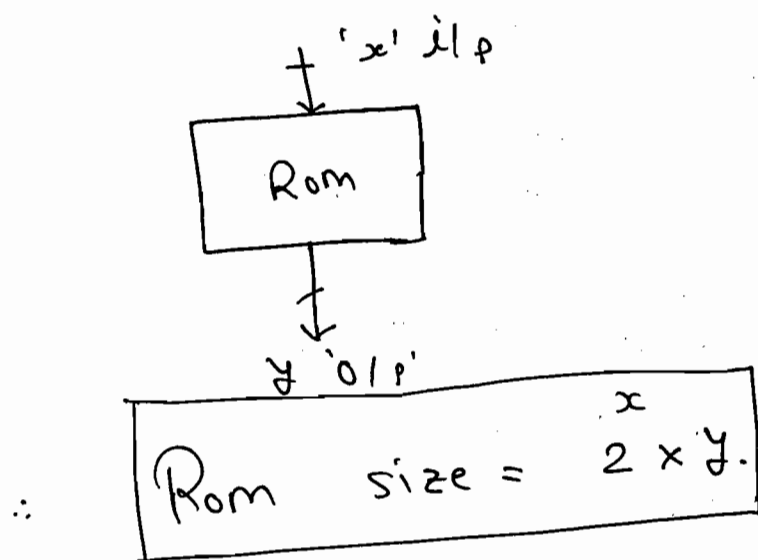
$$\text{size} = 2^3 \times 2 = 16.$$

⇒

Read only memory



⇒

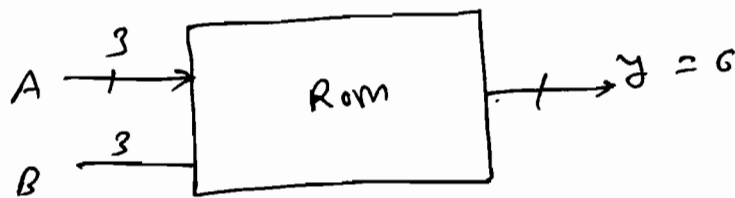


→ Rom size indicates that how many bits can be stored.

Ex-1 Determine the size of the Rom for the following functions.

① Rom as 3 bit binary multiplier.

②



$$x = 3 + 3 = 6$$

$$y = 6$$

$$7_{10} * 7_{10} = 49_{10}$$

$$2^y > 49$$

$$y = 6$$

$$\therefore \text{Rom size} = 2^x \times y = 2^6 \times 6 = 384.$$

② Rom as 4-bit squarer.

Ans:

$$x = 8$$



~~64~~

1111

1111

~~64~~

~~64 x 64~~

~~4 x 4~~

~~2 x 2~~

~~y = 8~~

$$x = 4$$

$$2^4 \times 8 = 16 \times 8 = 128$$

$$15 \times 15 = 225_{10}$$

$\therefore$

$$2^y > 225$$

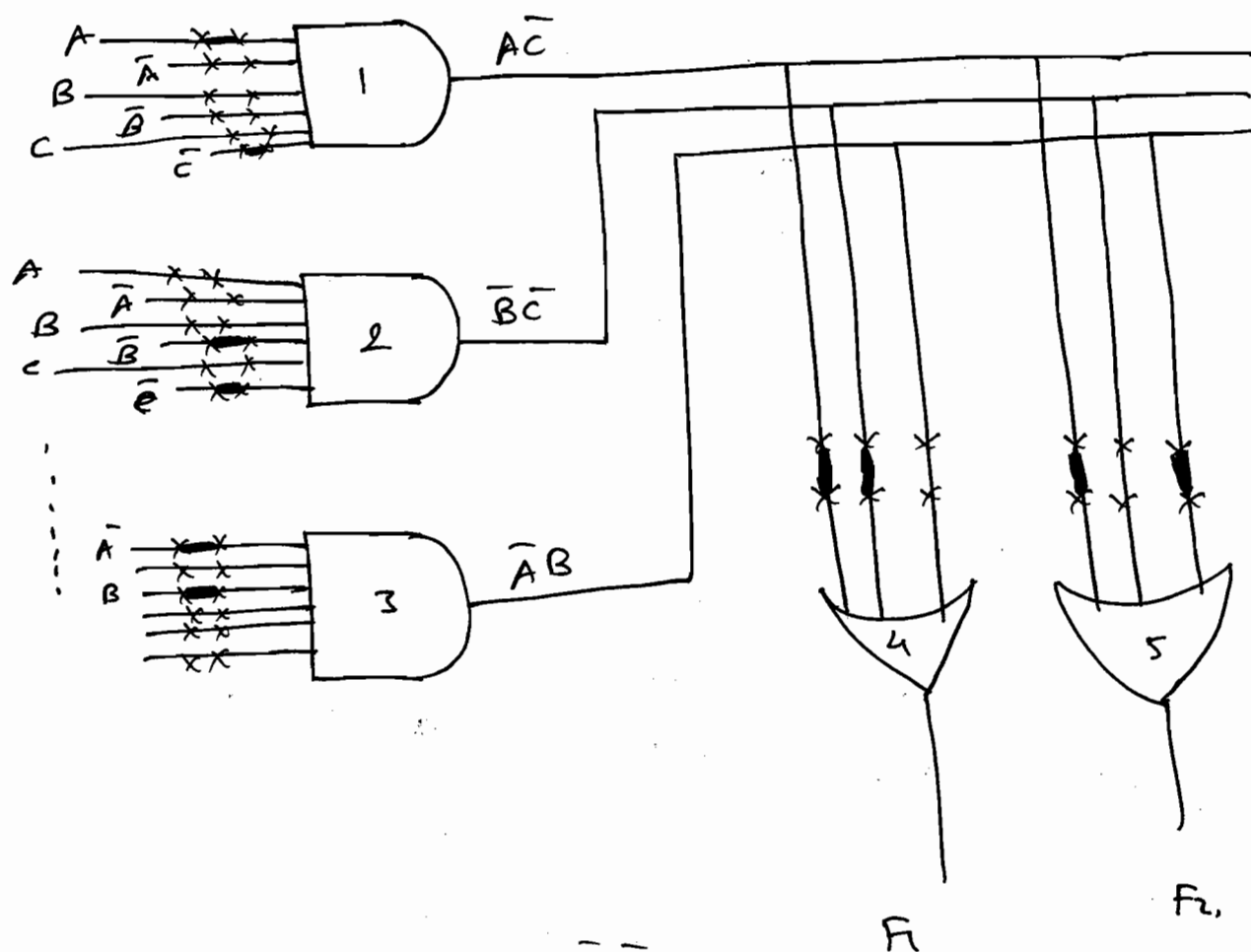
$$y = 8, x = 4$$

$$\therefore \text{size} = 2^4 \times 8 = 128.$$

* PLA (Prog. Logic Array):

109

⇒ Prog. AND gates + Prog. OR gates.



$$F_1(A, B, C) = A\bar{B}\bar{C} + \bar{A}B\bar{C}$$

$$F_2(A, B, C) = \bar{A}B + A\bar{C}$$

Product terms $\rightarrow$

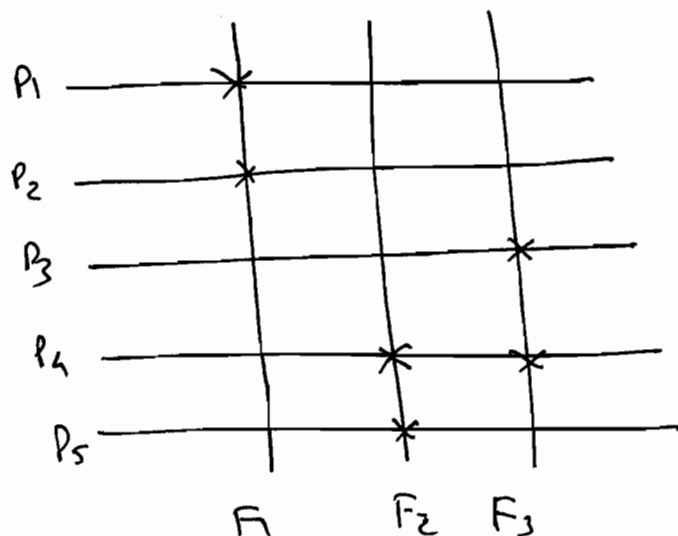
$$P_1 = A\bar{B}\bar{C}$$

$$P_2 = \bar{A}B\bar{C}$$

$$P_3 = \bar{A}B$$

PLA size $\Rightarrow$ (3 inputs, 3 product terms, 2 outputs).

Ex-1 In the following PLA Determine the product terms P_1, P_2, P_3, P_4, P_5



$$F_1(A, B, C) = \bar{A}B\bar{C} + A\bar{B}C + AB$$

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

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

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

$$F_1(A, B, C) = \bar{A}B\bar{C} + AB + AC = \bar{A}B\bar{C} + AC$$

$$F_2(A, B, C) = \bar{A}\bar{B}\bar{C} + ABC + \bar{A}C$$

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

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

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

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

$$F_2 = \bar{A}\bar{C} + \bar{A}B + BC = \bar{A}\bar{C} + BC$$

$$F_3 = \bar{A}B\bar{C} + BC + A\bar{B}C$$

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

$$= AC + BC + \bar{A}B\bar{C}$$

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

$$= AC + BC + \bar{A}B$$

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

$F_1 =$

	BC			
	00	01	11	10
A	0			1
1		1		1

$$F_1 = \underline{AC} + BC$$

$F_2 =$

	BC			
	00	01	11	10
A	0	1		
1			1	

$$F_2 = \bar{A}\bar{B} + BC$$

$F_3 =$

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

$$F_3 = \underline{AC} + \bar{A}B$$

$$\therefore P_2 = AC, P_1 = BC, P_3 = \bar{A}B$$

$$P_4 = \bar{A}B, P_5 = BC \text{ or}$$

$$P_5 = \bar{A}B, P_4 = BC$$

size = (3, 5, 3)
 (input product output term)

* PAL : Prog. AND gate + fixed OR gate.

GRAL : Generic Array logic

= Prog. AND gates + fixed OR gate

+ Prog. Output Logic

(E.g. Tri-state O/P, Normal d.p etc).

Ex-1 How many invalid 10 combinations occur at input of BCD adder?

Ans: No. of Invalid BCD combinations at Input.

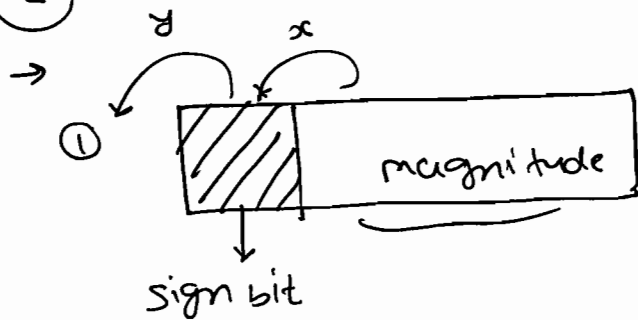
= Total Input combinations
- No. of Valid combinations.

$$= (16 \times 16) - (10 \times 10) \\ = 156.$$

* Overflow:

- overflow is used in signed arithmetic.
- overflow occurs whenever the result exceeds the range of signed no.s.
- The overflow condition can be verified by the following methods:

①



If $x=1$ and $y=0$ (or) $x=0$ and $y=1$ } overflow occurred.



② Overflow occurs if 2 +ve no.s are added the result is -ve or 2 -ve no.s are added the result is +ve.

E.g. Consider 2's comp. nos.

⑥ $\begin{array}{r} x=0 \\ y=1 \end{array}$

$$\begin{array}{r} 1 \ 0 \ 1 \ (-3) \\ + \ 1 \ 1 \ 0 \ (-2) \\ \hline 0 \ 0 \ 1 \end{array}$$

Overflow occurred
(or)

Two -ve nos are added result is +ve

⑦ $\begin{array}{r} y=0 \rightarrow x=1 \\ 0 \ 1 \ 0 \ (2) \\ + \ 0 \ 1 \ 1 \ (3) \\ \hline 1 \ 0 \ 1 \ (-3) \end{array}$

Overflow occurred
(or)

Two +ve nos are added result is -ve

⑧ $\begin{array}{r} y=1 \rightarrow x=1 \\ 1 \ 1 \ 0 \ (-2) \\ + \ 1 \ 1 \ 0 \ (-2) \\ \hline 1 \ 0 \ 0 \ (-4) \end{array}$

No overflow

Ex-1

$$\begin{array}{r} x \\ + y \\ \hline z \end{array}$$

overflow

If $x=1, y=1$ but $z=0$
(or)

$x=0, y=0$ but $z=1$

Expression for overflow i.e. $\bar{x} \cdot \bar{y} \cdot z + x \cdot y \cdot \bar{z}$