$0 \ldots 9 \; (\;)_{10} \rightarrow$ Decimal $\quad (16)_{10} \quad 16$

$0, 1 \; (\;)_2 \rightarrow$ Binary $\quad (0101011)_2$

$0, 1, \ldots 7 \; (\;)_8 \rightarrow$ Octal $\quad (16)_8$

$0 \ldots 9 \; (\;)_{16} \rightarrow$ Hexadecimal $\quad (16)_{16} \rightarrow 16H$

A, B, C, D, E, F

M	A	D	D	A	D
↓	↓	↓	↓	↓	↓
multiply	Any base	Decimal	Decimal	Any base	Divide

$(25)_{10} \rightarrow (\ \ )_2$

$$\begin{array}{r|l}
2 & 25 \\ \hline
2 & 12 - 1 \\ \hline
2 & 6 - 0 \\ \hline
2 & 3 - 0 \\ \hline
 & 1 - 1
\end{array}$$

$(11001)_2$

MSB LSB

$(1011)_2 = (\ \ )_{10}$

$$\begin{array}{cccc}
1 & 0 & 1 & 1 \\
\times & \times & \times & \times \\
2^3 & 2^2 & 2^1 & 2^0
\end{array}$$

$8 + 0 + 2 + 1 = 11$

$(\ \ )_{10} = (\ \ )_8$

$(\ \ )_8 = (\ \ )_{10}$

Hexadecimal $(\ \ )_{16}$ 0 .... 9, A, B, C, D, E, F

$(\ \ )_{10} = (\ \ )_{16}$

$(172)_{10} = (\ \ )_{16}$

$$\begin{array}{r|l}
16 & 172 \\ \hline
 & \underline{10} - \underline{12} \\
 & \ \ A \quad\ \ C
\end{array}$$

10 - A

$(AC)_{16}$

$$\begin{array}{r|l} 16 & 1616 \\ \hline 16 & 101 - 0 \\ \hline & 6 - 5 \end{array}$$

$(650)_{16}$

$(650)_{16} = (\quad)_{10}$

$$6 \times 16^2 \quad 5 \times 16^1 \quad 0 \times 16^0$$

$$\begin{array}{r} 256 \\ \times 6 \\ \hline \cancel{336} \\ 1536 \end{array}$$

$1536 + 80 + 0 = 1616$

(5) $(2532)_{16} = (\quad)_2$

$(\;)_{16} \rightarrow (\;)_{10} \rightarrow (\;)_2$

(II) 2(4)

$$\begin{array}{cccc} 2 & 5 & 3 & 2 \\ \downarrow & \downarrow & \downarrow & \downarrow \end{array}$$

$(0010\ 0101\ 0011\ 0010)_2$

	Binary 8 4 2 1	Hexad 16
0	0000	0
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

$(713)_8 = (\quad)_2$ $\qquad 2^3 = 8$

I $\quad (\ )_8 \to (\ )_{10} \to (\ )_2$

II $\quad 7 \to 111, \quad 1 \to 001, \quad 3 \to 011$

$111 \quad 001 \quad 011$

$(111001011)_2$

$(\ )_5 = (\ )_2$

$(\underbrace{000}_{0}\underbrace{110}_{6}\underbrace{010}_{2})_2 = (062)_8$ $\qquad 2^3 = 8$

$(\underbrace{\ \cdot\ }_{4}\ \underbrace{\ \cdot\ }_{4})_2 = (\ )_{16}$ $\qquad 2^4 = 16$

CODES

1. Gray
2. BCD
3. ASCII
4. Excess-3 [XS-3]

(I) BCD [Binary coded Decimal]

Every Digit as a group of 4 binary bits

Decimal		Binary	BCD
8	→	1000	1000
12	→	$\boxed{1100}$	$\boxed{0001\ 0010}$

12.3   00010010.0011

(II) Excess -3 XS 3

(12) → XS_3

+2

(I) BCD 0001 0010

(II) XS-3 + 0011 0011

(0100 0101)$_{XS-3}$

		S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

1
1
1 1
1

2|2
1-0
C S

2|3
1-1

33
33
66

(0110 0110)$_{XS-3}$

8 9
+3 3
1110 10
$\boxed{1101\ 1100}$ XS-3 code

③ Gray code

① $(12)_{10} \rightarrow$

Ⅰ Binary

Ⅱ Gray code

A	B	⊕
0	0	0
0	1	1
1	0	1
1	1	0

1 1 0 0

1 0 1 0 → Gray code

② 25

2	25
2	12 − 1
2	6 − 0
2	3 − 0
	1 − 1

999
1001

$(11001)_2$

10101 → Gray code.

ASCII (128) 0 - 127

A → 65 a → 97 ϕ → 32 0 → 48

B → 66 b → 98 1 → 49

(7 bit code)

Logic Gates

AND NAND Ex-OR

OR NOR

NOT

Basic Gates

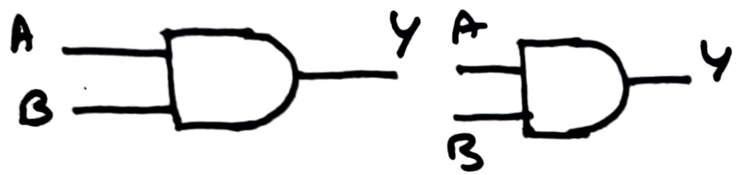
Universal Gates

Combinator

AND Gate

2 I/p 1 o/p Gate

Symbol



o/p Expression

$Y = A \cdot B$

logic table

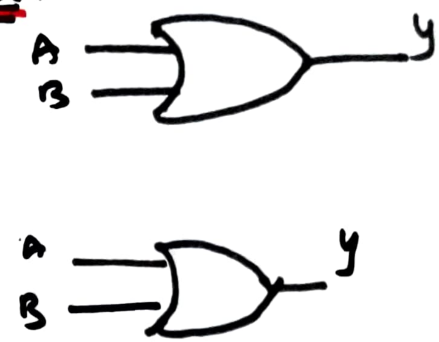
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

logic IC: 7408

OR Gate

2 I/p 1 o/p Gate.

Symbol



o/p Expression

$y = A + B$

logic table

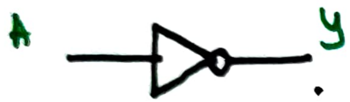
A	B	y
0	0	0
0	1	1
1	0	1
1	1	1

logic IC :- 7432

Not Gate

1 I/p 1 o/p Gate.

Symbol



logic expression

$y = \overline{A}$

logic table

A	y
0	1
1	0

logic IC 7404

NAND Gate

2 I/p 1 o/p Gate

Symbol

Expression

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

logic table

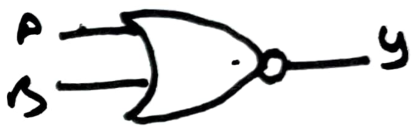
A	B	y
0	0	1
0	1	1
1	0	1
1	1	0

logic IC :- 7400

NOR Gate

2 I/p 1 o/p Gate

Symbol



Expression

$y = \overline{A + B}$

logic table

A	B	y
0	0	1
0	1	0
1	0	0
1	1	0

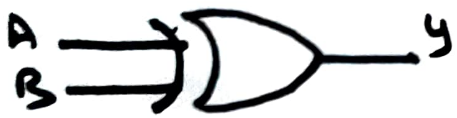
logic IC 7402

Ex-OR

Ex-NOR

2 I/P 1 O/P Gate

Symbol



Expression

$y = A \oplus B$ $\boxed{y = \bar{A}B + A\bar{B}}$

table

A	B	y
0	0	0
0	1	1
1	0	1
1	1	0

IC : 7486

"Addition of two numbers"

Arch

1. ADD → Instruction

2. Registers

   Memory address

   MOV B, C

   MVI B, 21

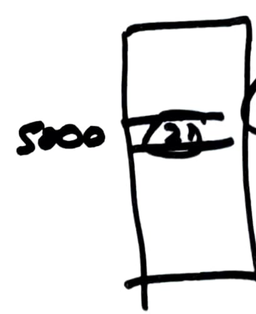
   LDA 5000

3. Addressing

Organisation

1. Dedicate hardware
   or
   will share some existing H/w

2. memory allocation



3. compiler
   or
   hardwired. logic

① Binary addition

A	B	A+B	C
0	0	0	0
0	0	1	0
1	0	1	0
1	1	0	1

8 4 2 1

10 + 7 in Binary

1010 + 111

$$
\begin{array}{r l}
1\,0\,1\,0 & \leftarrow A \\
0\,1\,1\,1 & \leftarrow B \\
1 \qquad\quad & \leftarrow \text{carry} \\
(1\,0\,0\,0\,1)_2 &
\end{array}
$$

Try $(1111\ 0010)_2 + (01110)_2$

Binary subtraction

	A-B		
A	B	R	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$(0011010)_2 - (001100)_2$

$$
\begin{array}{r l}
 & 0011010 \rightarrow 26 \\
- & 0001100 \rightarrow 12 \\
\hline
 & 11 \\
\hline
 & 0001110 \rightarrow 14
\end{array}
$$

$$
\begin{array}{r}
{}^{1}0 \\
-\;1 \\
\hline
1
\end{array}
$$

$11 \rightarrow 3$

Try

$(1100\ 1100)_2 - (0011\ 0011)_2$

Binary multiplication

A	B	A × B
0	0	0
0	1	0
1	0	0
1	1	1

$(1100)_2 \times (100)_2$

```
     1100  ← 12
      100  ← 4
   ------
     0000
    0000×
   1100××
   ------
   110000  ← (48)10
```

TW $(13)_{10} \times (5)_{10}$ using Binary multiplication

Binary division

$(101010)_2 \div (110)_2$

$(42)_{10} \div (6)_{10} =$ q = 7, r = 0

```
       111      <- quotient
    ________
110 ) 101010
    -  110
    --------
      01001
    -   110
    --------
     000110
        110
    --------
        000  <- rem
```

Try

$(27)_{10} \div (5)_{10}$ using Binary Division.

$110 \overline{)10010}$

$\underset{6}{110} \overline{)\underset{7}{111}}$

Signed Binary

7 bit

5 → 101

sign

+5 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

Sign Bit
1 → -ve
0 → +ve

-5 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

-5

2 bit

S		
0	11	+3
0	10	+2
0	01	+1
0	00	+0
1	00	-0
1	01	-1
1	10	-2
1	11	-3

X

3 bit 00

1's complement

3bit -3

+3 → 011

-3 → 100 (1's complement)

+7 (4bit) 0111 → 0111 → 1's complement)

+7

3 bit

+6 to −6

~~+7~~	111	
+6		
+5		
+4	100	
+3 →	011	
+2 →	010	
+1 →	001	
0 →	000	
−0 →	111	
−1 →	110	
−2 →	101	
−3 →	100	
−4	011	
−5		
−6		
−7	000	

×

SW

obtain 1's complement for the following number using 7 bit format

1> −12 → Bin 0100000

2> −32 → ~~0011111~~ → 1011111 (−32)

3> +7 → 0000111 (+7)

4> −19

2's complement (used for -ve)

Steps

1> obtain 1's complement of the binary number

2> Add 1 to the 1's complement

Ex -7 in 2's complement (4 bit)

Binary for +7 → 0111

1> obtain 1's complement

1000

2> Add 1 to 1's compleme

$$\begin{array}{r} 1000 \\ +\ \ 1 \\ \hline 1001 \end{array}$$ ← -7 in 2's complement)

-34

+34 100010

011101 1's comp

+ 1 Add 1

011110 → 2's comp

2's complement 011110

TW

obtain 2's complement (7 bit)

a) ① −33

② −15

④ +16

1>

4 bit → $\boxed{\begin{matrix} 0000 \\ 0000 \end{matrix}}$ → +0, −0

$$\begin{array}{r} 0000 \\ 1111 \\ +\quad 1 \\ \hline \boxed{1}\,0000 \end{array}$$

ignore

end around carry (1's comp)

2's

1. take 9's complement of subtrahend (-ve)
2. Add BCD of 9's complement with minuend
3. Check result for Invalid BCD & apply correction
4. Shift carry to next bit
5. if carry from MSB then end around carry

BCD Arithematic

Step 1

42

$\boxed{0100 \quad 0010}$

42

8421
0010
0100

2458

0010 0100 0101 1000

case 1 Add 2 BCD result is valid & no carry

case 2 [result is valid BCD] but carry generated

apply correction i.e [6]

case 3 [result is Invalid] apply correction i.e 6

BCD Addition 8421

1> 3 + 2

0011 0010

$$\begin{array}{r} 0011 \\ +\ 0010 \\ \hline 0101 \\ \hline 5 \end{array}$$

2) 22 + 43 = 65

22 $\xrightarrow{BCD}$ 0010 0010

43 $\xrightarrow{BCD}$ 0100 0011

0110 0101

6 5

Valid

0	0000
⋮	
9	1001

Inv

10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Case I result is valid BCD number

case 2

result is invalid BCD

① 6 + 7 = 13

6 → 0110

7 → 0111

1101

\+ 0110

0001 0011

1 3

② 19 + 12 = ~~2~~ 31

19 $\xrightarrow{BCD}$ 0001 1001

12 $\xrightarrow{BCD}$ 0001 0010

0010 1011

✓ ×

\+ 0110

0010 + 1 0001

0011 0001

3 1

case 3 result is valid but carry generated
Apply correction

1) + 9 → 1001
+ 9 → 1001

$\boxed{1}$ 0010 ×
0110

0001 1000
1 8

2) 27 + 29 = 56
27 → 0010 0111
29 → 0010 1001

$\boxed{1}$
0100 0000 ×
✓ 0110

0100 0110
+1

0101 0110
5 6

Perform BCD Addition

336 + 376 = 712

336 → 0011 0011 0110

376 → 0011 0111 0110

11 111 11

0100 1010 1100

0110 0110

1 11 1 1

0110 0000 0010

1 1

0111 0001 0010

7 1 2

BCD subsbraction

A − B   9's complement

A + (−B)

98 − 81 = 98 + (−81) = 99 − 81 = 18 → 9's complement of −81

98 → minuend, 81 → subtrend

98 $\xrightarrow{BCD}$	1001	1000
18 → +	0001	1000
	1010	0000 → C
C	0110	0110
1	0000	0110
		+1
	0001	0111
	1	7

98 − 63 = 98 + (−63)

99 − 63 = (36) 9's complement (−63)

336 - 623 = 336 + (-623) = -287

minuend: 336; subtrahend: 623

= 336 + 376

```
  999
- 623
-----
  376 → [9's complement of -623]
```

```
    336 → 0011 0011 0110
    376 → 0011 0111 0110
  +
  ----------------------
         0111 ✓  1010 ✗  1000 ✗
              0110    0110
  +
  ----------------------
         0110   0000   0010
         1 ←    1 ←
  +
  ----------------------
         0111   0001   0010
          7      1      2   [9's complement form]
```

No carry from MSB (result is -ve)

```
  999
- 712
-----
 -287
```

-287 is 9's comp of 712

Hexadecimal addition [Hex Addition]

✓ $(\ )_{16}$ ✓ ends with H ✓ starts with 0X

Hex	Dec
0	0
⋮	⋮
9	9
A	10
B	11
C	12
D	13
E	14
F	15

39H 1AH

$$\begin{array}{r} 3\ 9 \\ 1\ A \\ \hline {}^{1} \\ \hline (5\ 3)_{16} \end{array}$$

19 − 16 = 03

$16 \overline{)19}$ $\boxed{1-3}$

$(BC)_{16} + (19)_{16}$

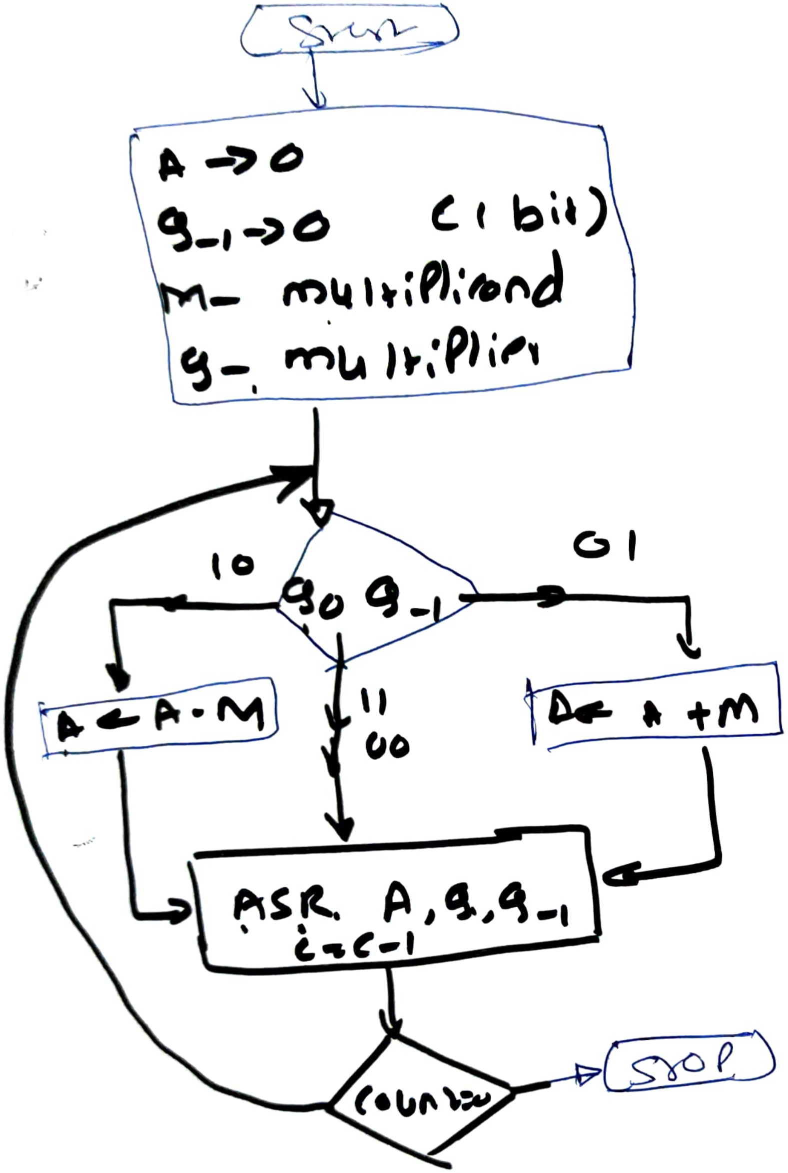
$$\begin{array}{r} B\ C \\ 1\ 9 \\ \hline {}^{1} \\ \hline (D\ 5)_{16} \end{array}$$

21 − 16 = 5

$16 \overline{)21}$ 1 − 5

16 > ... −16

$(FAF)_{16} + (ABC)_{16}$



Booth's Algorithm

~~5×7~~ 7×3

$A \rightarrow 0000$ $Q \rightarrow 0011$ $Q_{-1} = 0$

$M = 0111$ $-M \rightarrow 1001$

A	Q	Q_{-1}	Count	Remark
0000	0011	0	4	Initilise
+ 1001				
1001	0011	0		A=A-M
1100	1001	1	3	ASR A,Q,Q_{-1} count=count-1
1110	0100	1	2	ASR
+ 0111				A=A+M
0101	0100	1		
0010	1010	0	1	ASR c=c-1
$\boxed{0001 \quad 0101}$		0	0	

21

M Q

7 × -3 = -21

Initilise A → 0000

Q_{-1} → 0

M → 0111 -M → 1001

Q → 1101

3

0011 1100

1.101 +1

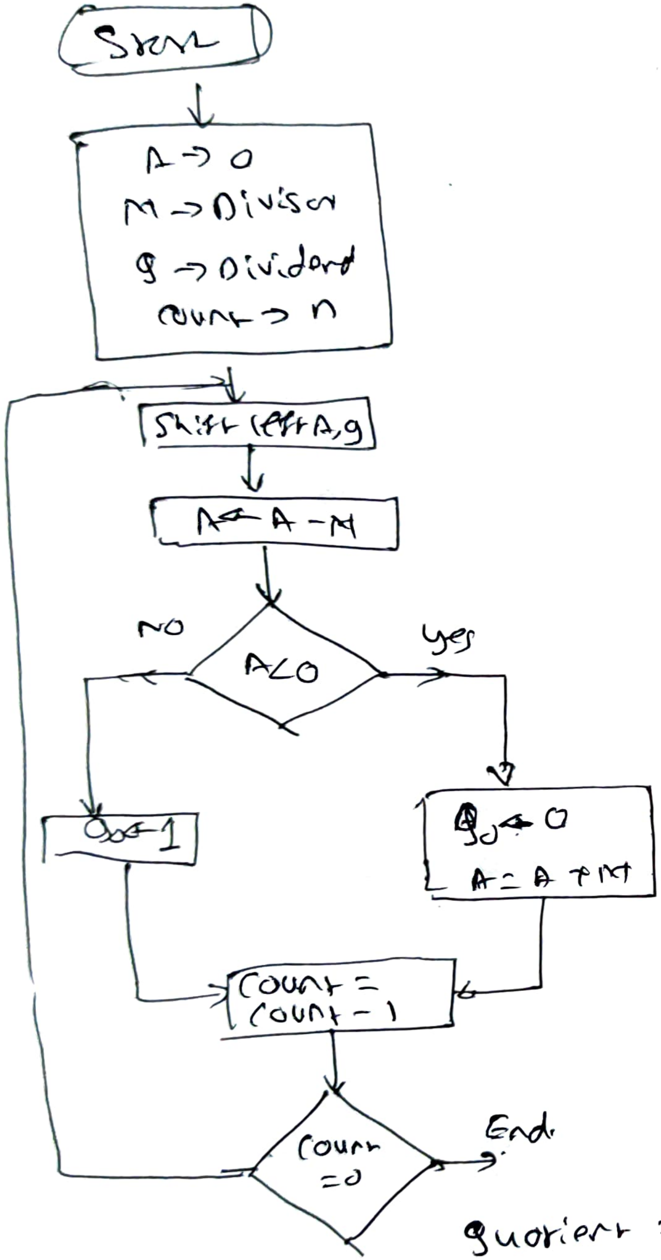
1101 (-3)

A	Q	Q_{-1}	count	Remark
0000	1101	0	4	Initilise
+ 1001				A = A - M
1001	1101	0		
1100	1110	1	3	ASR A, Q, Q_{-1}
+ 0111				A = A + M
0011	1110	1		
0001	1111	0	2	ASR A, Q, Q_{-1}
+ 1001				A = A - M
1010	1111	0		
1101	0111	1	1	ASR A, Q, Q_{-1}
$\boxed{1110}$	$\boxed{1011}$	1	0	ASR A, Q, Q_{-1}

−21 (2's complement of 21)

$21 \rightarrow 0001\ 0101 \rightarrow \boxed{1110\ 1011}$ −21

Restoring



Quotient = Q

Remainder = A

~~7/8~~ 6/2

Q → 0110

M = 0010

−M = 1110

A	Q	count
0000	0110	4
0000	1100	
+ 1110		
1110	~~0~~100	
0010		
0000	~~0~~100	3
0000	1000	
1110		
1110	~~0~~100	
0000	~~0~~100	2

Shift left

A = A − M

Q = 0

A = A + M

6/2

Q → 0110 M = 0010 -M = 1110

count	A	Q	
4	0000	0110	
	0000	1100	SLAQ
	1110		A = A - M
	1110	1100	
	-0010		
	0000	1100	SLAQ
3	0001	1000	A = A - M
	+1110		
	1111	1000	
	+0010	1000	
	0001	1000	
2	0011	0000	SLAQ
	+1110		A = A - M
	0001	0001	
1	0010	0010	SLAQ
	+1110		A = A - M
	0000	0011	
0	remainder	quotient	

6/2

6 - 2 = 4 = 1

4 - 2 = 2 = 2

2 - 2 = 0 = 3

rem

7 / 3

7 - 3 = 4 = 1

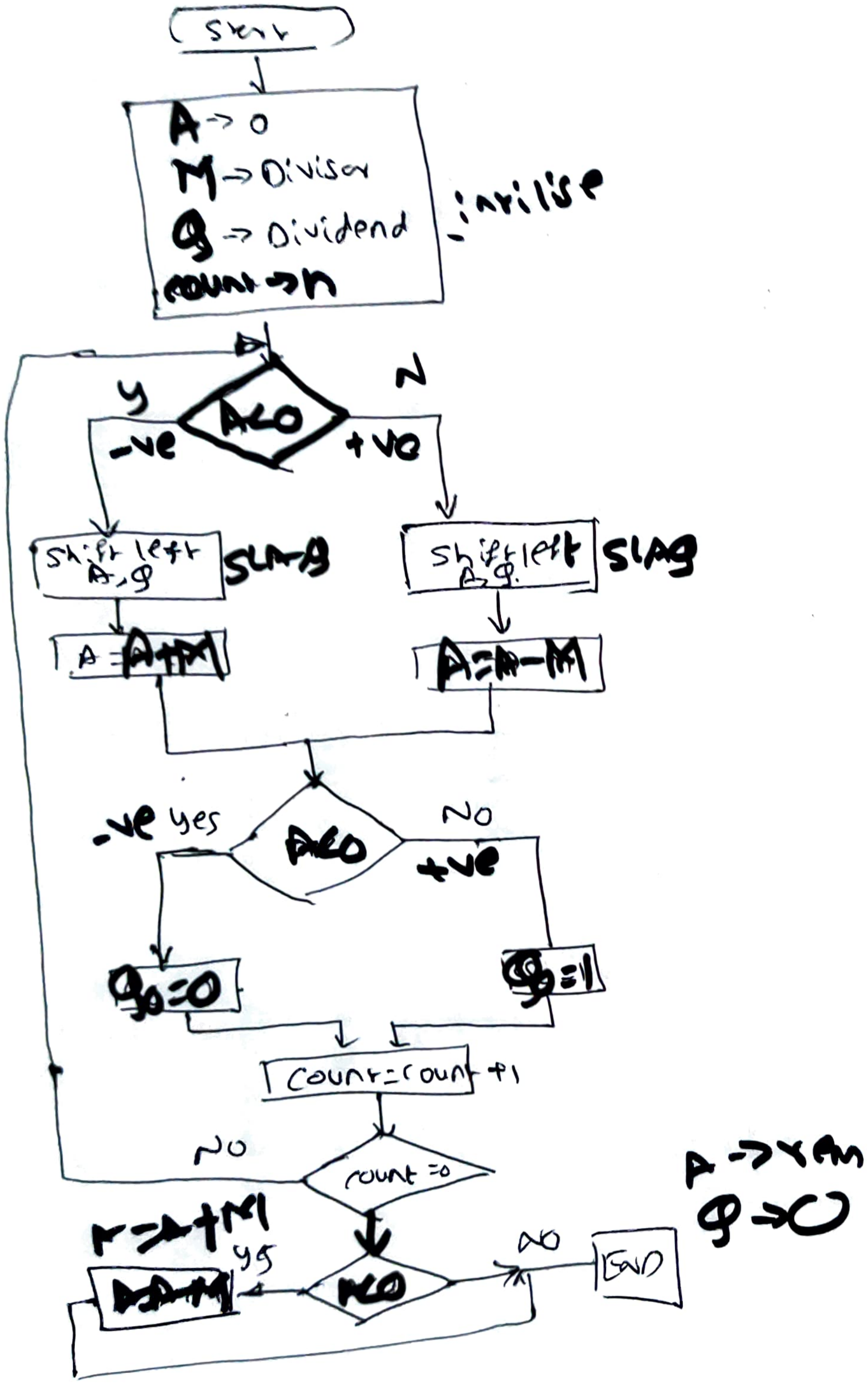
4 - 3 = 1 = 2 q.

1 - 3 = -2

-2 + 3 = $\boxed{1}$

rem

Non - Restoring



11/3 → non-restoring

A → 0   Q → 01011   M = 00011   −M = 11101

Round	A	Q	Remark
5	00000	01011	
	00000	10110	SLAQ
	−+ 11101		A = A − M
4	11101	10110	Q = 0
	11001	01101	SLAQ
	00011		A = A + M
3	11110	01100	Q = 0
	11100	11001	SLAQ
	+ 00011		A = A + M
2	11111	11000	Q = 0
	11111	10001	SLAQ
	+ 00011		A = A + M
1	00010	10001	Q = 1
	00101	00010	SLAQ
	+ (−M) 11101		A = A − M
0	00010	00011	
	rem	Q	

36

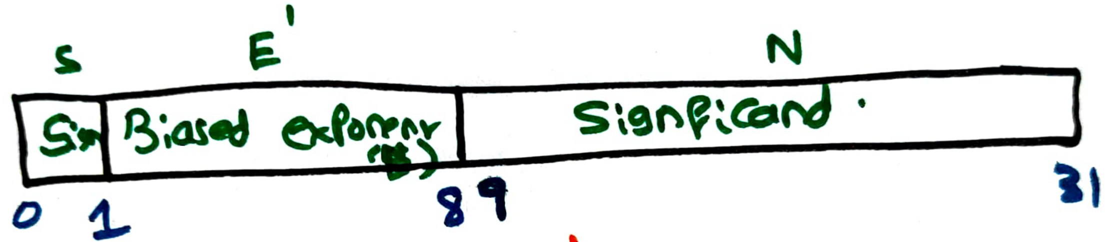
11/3

q = ~~[illegible]~~

q = 01011 M = 00011 -M = 11101 count = 5

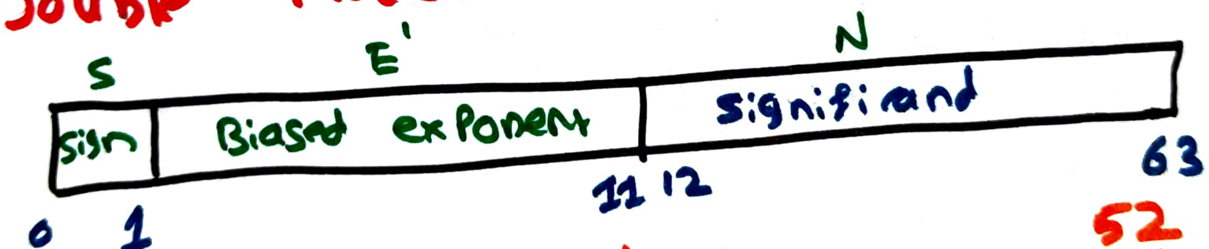
IEEE 754 standard for Floating point numbers.

① Single Precision Format (32 bit)



S -> 1 bit E' -> 8 bits N - 23 bits

② Double Precision Format (64 bit)



S -> 1 bit E' -> 11 bits N - 52 bits

200.5 in IEEE 754 standards [32 bits & 64 bits] 39

① convert the number in Binary

$$\begin{array}{r|l} 16 & 200 \\ \hline & 12 - 8 \end{array}$$

$(200)_{10} = (C8)_{16}$

$(11001000)_2$

$0.5 \times 2 = 1.0 \quad 1$

$0.0 \times 2 = 0.0 \quad 0$

$(0.5) = (.10)_2$

$(200.5) = (11001000.10)_2$

$0.32 \times 2 = 0.64$	0
$0.64 \times 2 = 1.28$	1
$0.28 \times 2 = 0.56$	0
$0.56 \times 2 = 1.12$	1
$0.12 \times 2 = 0.24$	0
$0.24 \times 2 = 0.48$	0
$0.48 \times 2 = 0.96$	0
$0.96 \times 2 = 1.92$	1

$0.32 \; (0.01010001)_2$

Step 2 Normalization

$$(200.5)_{10} = (11001000.10)_2$$

$$= 1.\underbrace{1001000010}_{N} \times 2^{7}$$

2572.173

$\boxed{2.572173 \times 10^{3}}$ — Normalised

0.0007283

7.283×10^{-4}

$(0.00000101001)_2$

1.01001×2^{-6}

Step 3 Calculate Biased Exponent (E')

Actual Exponent = 7

single

$E' = E + 127$

$E' = 7 + 127$

$= 134$

double

$E' = E + 1023$

$E' = 7 + 1023$

$= 1030$

$$\begin{array}{r|l} 16 & 1030 \\ 16 & 64 - 6 \\ & 4 - 0 \end{array}$$

$(406)_{16}$

$$\begin{array}{r|l} 16 & 134 \\ & 8 - 6 \end{array}$$

$(134) = (86)_{16} = (10000110)_2$

$(0100\ 0000\ 0110)_2$

32 bits

S → 0

E' → 10000110

N → 100100010

64 bits

S → 0

E' = 10000000110

N = 100100010

IEEE 754 single Precision Format

for 200·5

S	E'	N
0	10000110	10010001000 0

IEEE 754 double Precision Format

for 200·5

	E'	N
0	10000000110	100100010 ....
0 1	11 12	63

Try (−0·125) in 32 bit & 64 IEEE754 Standards mul or division

$A \times E^{x} \;\;|\;\; B \times E^{y}$

multi: $A \times B \quad E^{x+y}$

Div $\frac{A}{B}\, E^{x-y}$

$[25 \times 10^{3}] \times [5 \times 10^{2}]$

$[125 \times 10^{5}]$

$\frac{25 \times 10^{3}}{5 \times 10^{2}}$

$= 5 \times 10^{1} = 50$

add / sub

$2.7 \times 10^3 + 0.2 \times 10^4$

$A \times E^x + B \times E^y$

[Addition or sub] is possible only if $x = y$

$\Rightarrow A + B \times E^{(x \text{ or } y)}$ [add]

$A - B \times E^{(x \text{ or } y)}$ [sub]

$2.7 \times 10^3 + 0.2 \times 10^4$

$0.27 \times 10^4 + 0.2 \times 10^4 = 0.47 \times 10^4$

Half Adder

Table

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

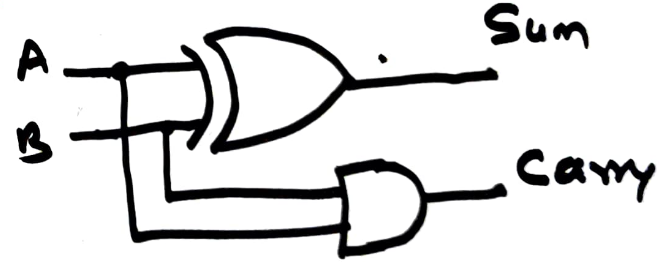
Block

Exp

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

$C = A \cdot B$

Circuit

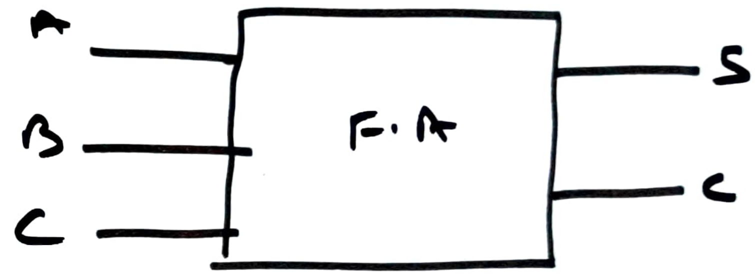


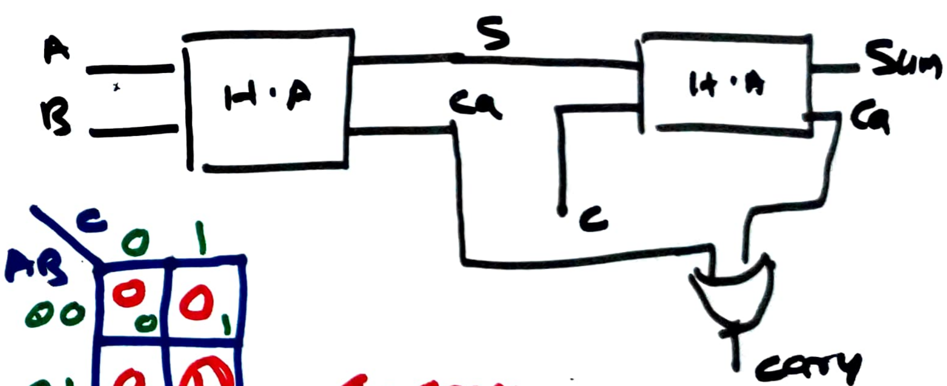
IC's → 7486 - EX-OR
7408 - AND

FULL ADDER

table

A	B	C	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





K-maps

S =

A \ BC	00	01	11	10
0	0_0	1_1	0_3	1_2
1	1	0	1	0

$$\boxed{S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC}$$

AB \ C	0	1
00	0_0	0_1
01	0_2	1_3
11	1	1
10	0_4	1

← carry

$$\boxed{\text{carry} = AB + BC + AC}$$

4 variable K-map

$2^4 = 16$

$$\boxed{y = \bar{A}\bar{B} + C}$$

AB \ CD	00	01	11	10
00	1 (0)	1 (1)	1 (3)	1 (2)
01	0 (4)	0 (5)	1 (7)	1 (6)
11	0 (12)	0 (13)	1 (15)	1 (14)
10	0 (8)	0 (9)	1 (11)	1 (10)

	8 A	4 B	2 C	1 D	y
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

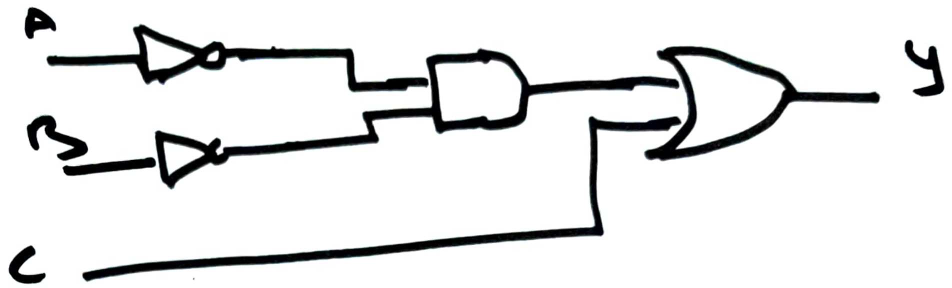
$\boxed{1 \rightarrow \text{min terms}}$ $y = \Sigma m(0, 1, 2, 3, 6, 7, 10, 11, 14, 15)$

$A = 0 \rightarrow \bar{A}$ $\quad$ $A = 1 \rightarrow A$

$\boxed{0 - \text{max terms}}$ $y = \Pi m(4, 5, 12, 13, 8, 9)$

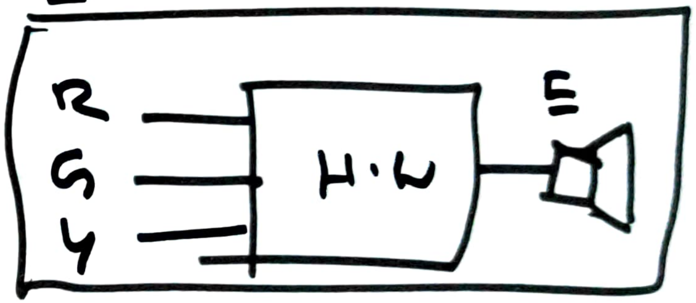
$A = 0 \rightarrow A$ $\quad$ $A = 1 \rightarrow \bar{A}$

$y =$

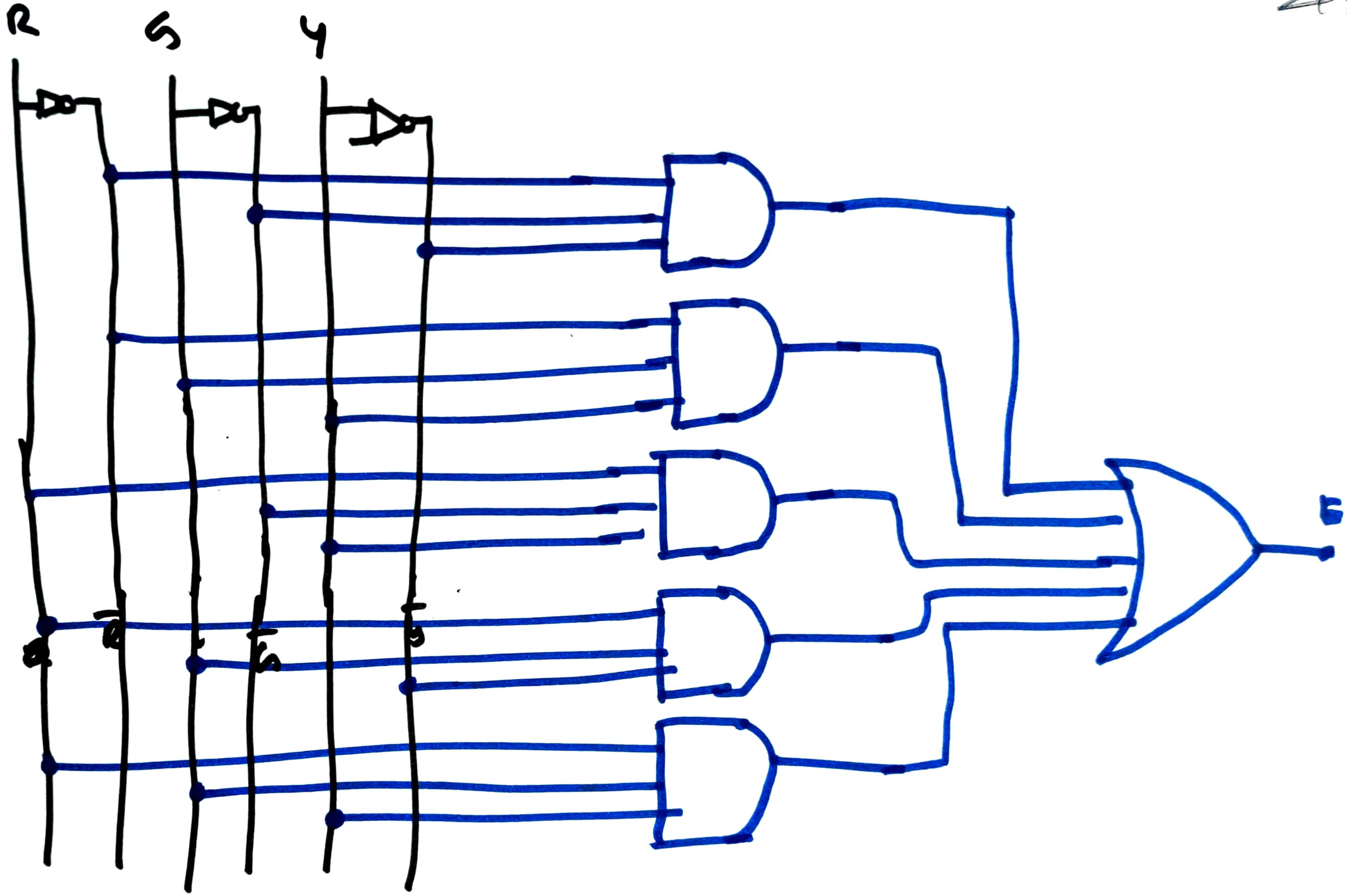


signal system

R	G	Y	Error
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	1

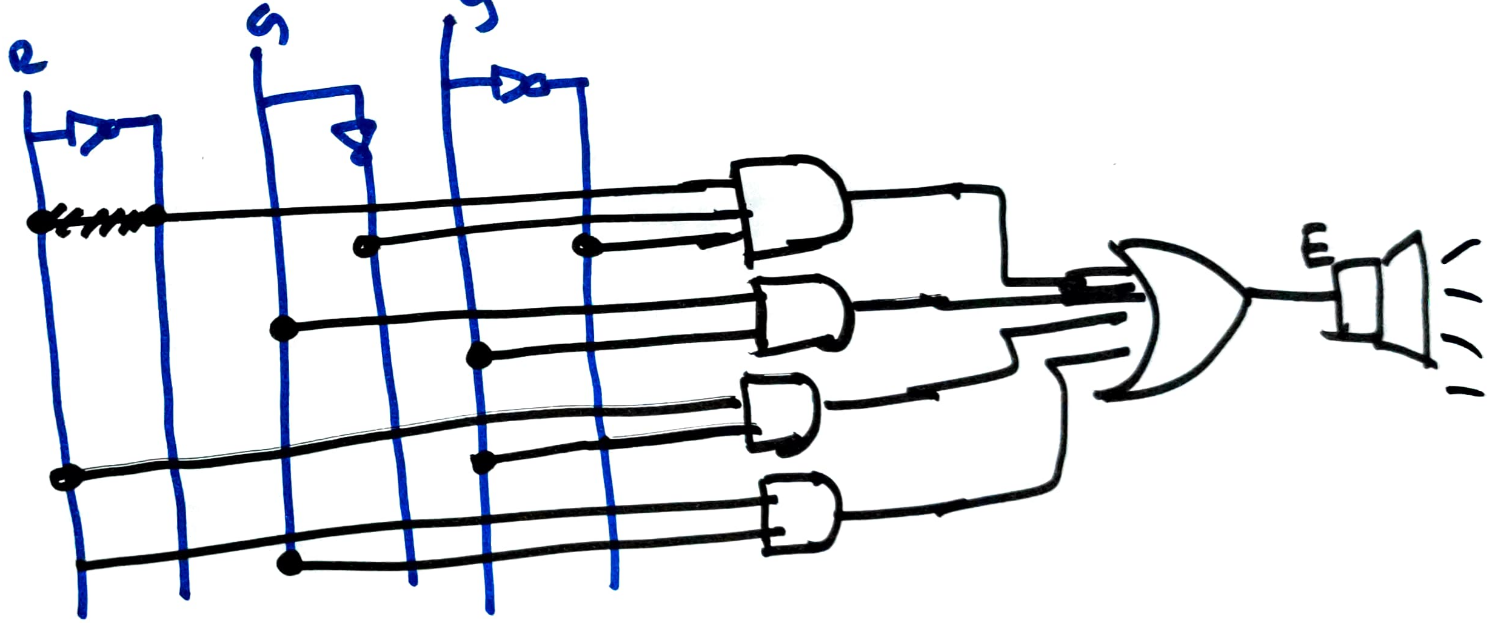


$$E = \bar{R}\bar{G}\bar{Y} + \bar{R}GY + R\bar{G}Y + RG\bar{Y} + RGY$$


R
G
Y
E

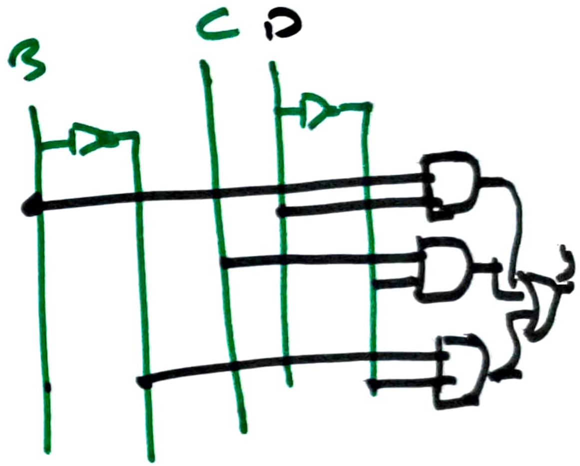
R \ GY	00	01	11	10
0	1	0	1	0
1	0	1	1	1

$$E = \bar{R}\bar{G}\bar{Y} + GY + RY + RG$$



y

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	1	1	1
11	0	1	1	1
10	1	0	0	1



$$y = BD + C\bar{D} + \bar{B}\bar{D}$$

$\cancel{A + \bar{A}}$

$\cancel{BD + \bar{B}\bar{D}}$

$$y(A,B,C,D) = \Sigma m(0, 1, \text{[redacted]}, 2, 4, 5, 6, 8, 9, 10, 12, 13, 14)$$

AB \ CD	00	01	11	10
00	1_0	1	0_3	1_2
01	1_4	1_5	0_7	1_6
11	1_8	1_{13}	0_{15}	1_{14}
10	1_8	1_9	0_{11}	1_{10}

$y = \bar{C} + \bar{D} = \overline{CD}$

Boolean Algebra

$$y = A + 1 = 1$$

$$y = B \cdot 0 = 0$$

$$y = A + 0 = A$$

$$\boxed{\begin{aligned} y &= A + \bar{A} = 1 \\ y &= A \cdot \bar{A} = 0 \end{aligned}}$$

$$y = \overline{CD}$$

De-morgan law

$$\boxed{\begin{aligned} y &= \overline{A \cdot B} = \bar{A} + \bar{B} \\ y &= \overline{A + B} = \bar{A}\,\bar{B} \end{aligned}}$$

Break Line Chang Sign

			LHS			RHS		
A	B	AB	$\overline{AB}$	$\bar{A}$	$\bar{B}$	$\bar{A}+\bar{B}$	$\overline{A+B}$	$\bar{A}\cdot\bar{B}$
0	0	0	1	1	1	1	1	1
0	1	0	1	1	0	1	0	0
1	0	0	1	0	1	1	0	0
1	1	1	0	0	0	0	0	0

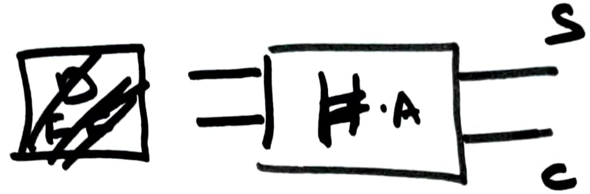
1 2

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

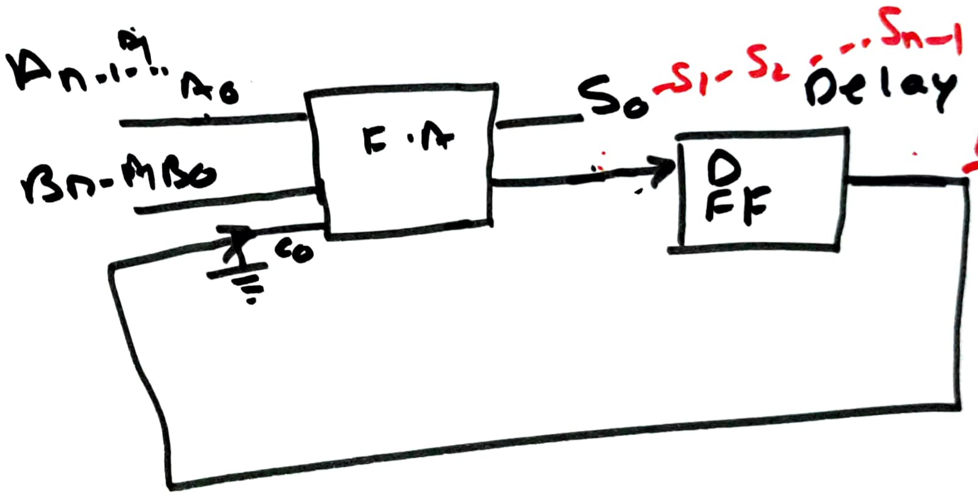
$$\overline{A+B} = \bar{A}\cdot\bar{B}$$

$$\begin{array}{rl} & A_5 \quad\quad\quad A_0 \\ A \rightarrow & 101101 \\ & B_5 \quad\quad\quad B_0 \\ B \rightarrow & 100010 \\ & \quad\quad\quad\; C_0\ 0 \\ \hline \rightarrow & \quad\quad\quad\quad S_0 \end{array}$$

n- bit adder



Serial → Parallel



Delay FlipFlop

Adv → low cost

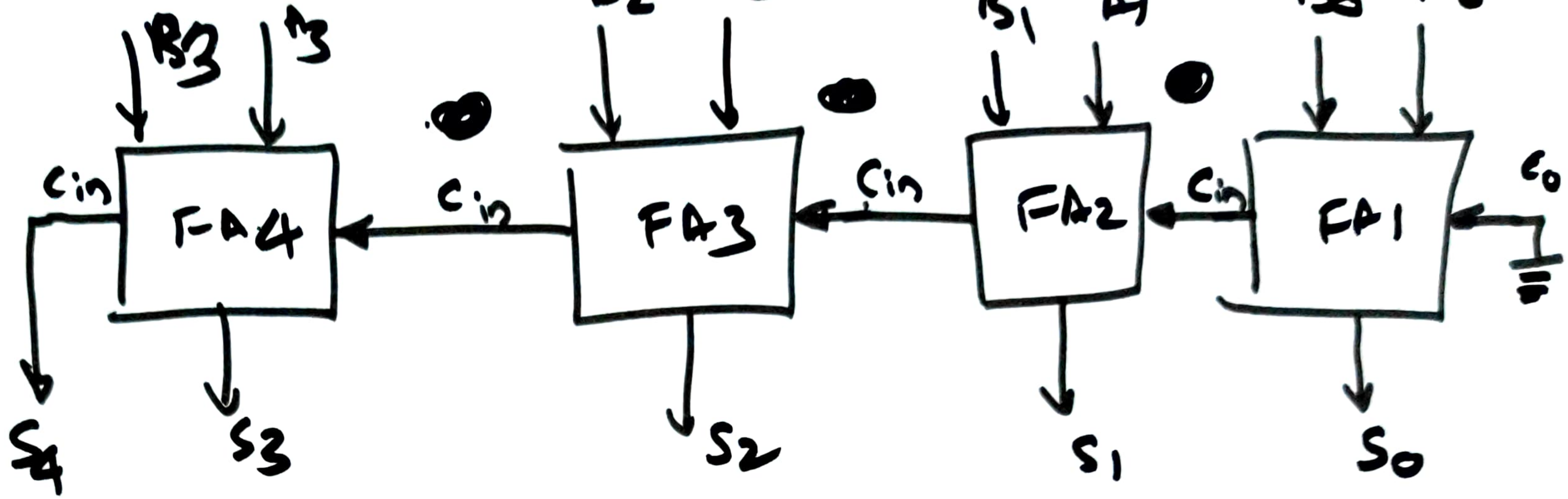
Drawback

large amount of time

$\boxed{C1 = A_0 \cdot B_0} + A_0 \oplus B_0 \cdot 0$

n-bit Parallel Adder

A3 A0
1011
1100
B B0



Progation delay (Ripple carry) of carry

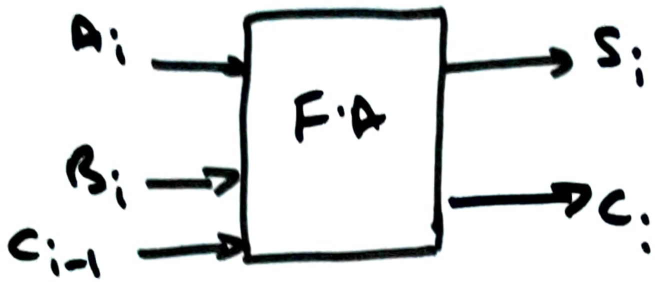
Adv
Process is faster

Drawbacks
1) Hardware is expensive
2) Propogation Delay

7483 -IC → Four Bit Adder IC

16 Pin IC → 4 I/P - A, 4 I/P - B, 4 O/P → Sum, Vcc, Gnd, C0, C4

look ahead carry Generator

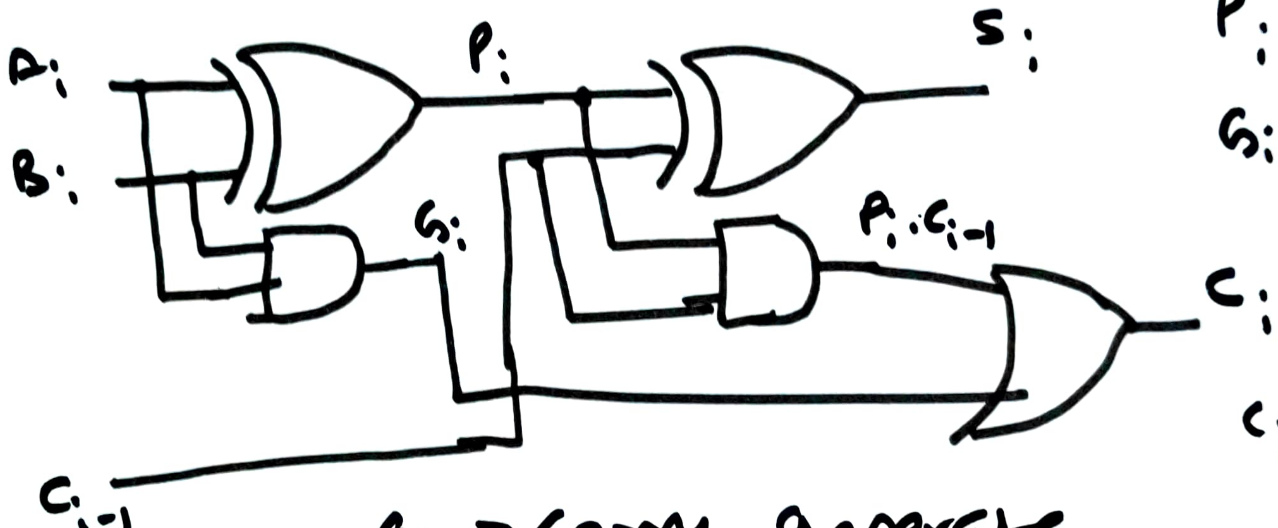


$S_i = P_i \oplus C_{i-1}$

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

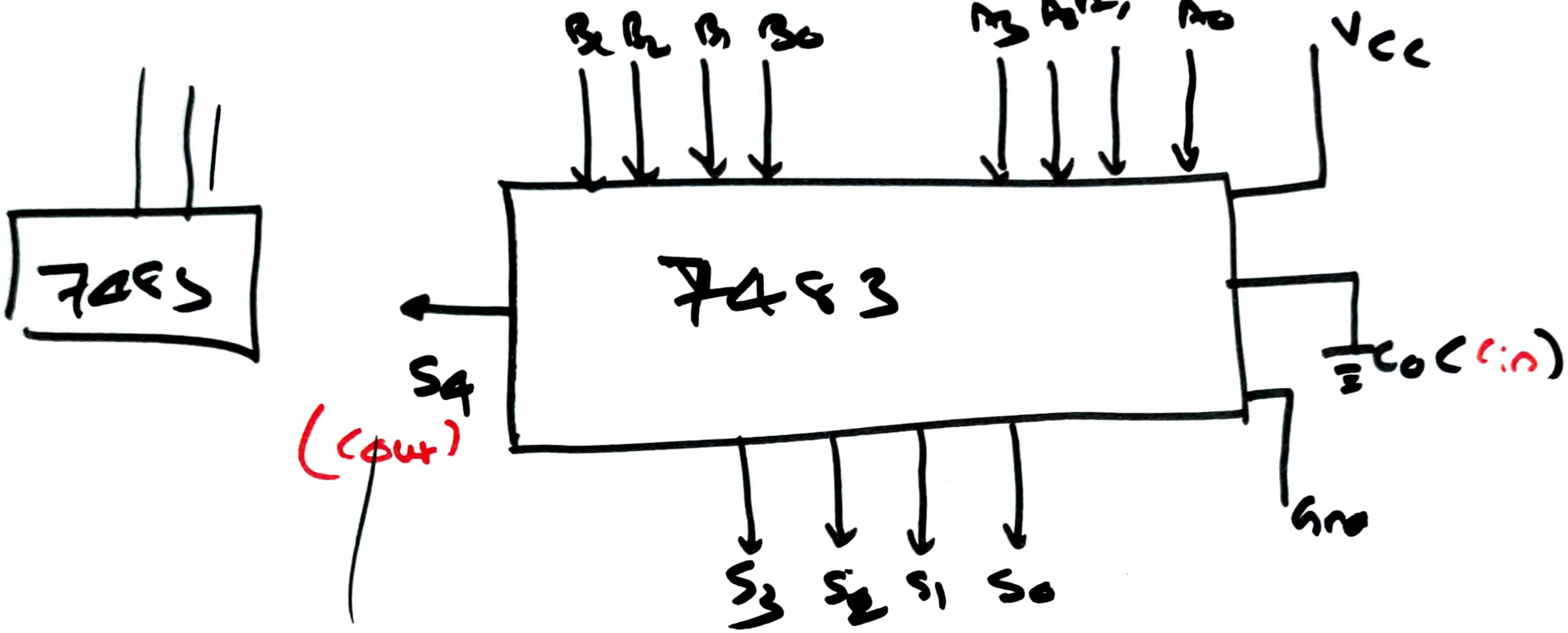
$P_i = A_i \oplus B_i$

$G_i = A_i \cdot B_i$

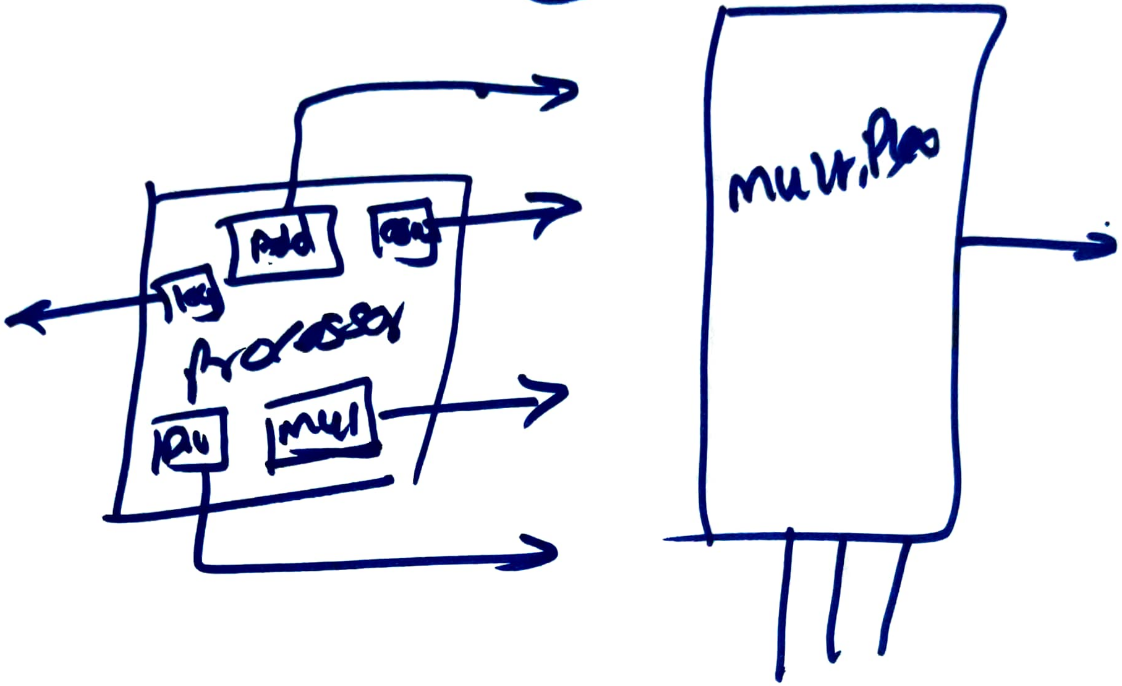


G_i -> carry generate

P_i -> carry Propogate

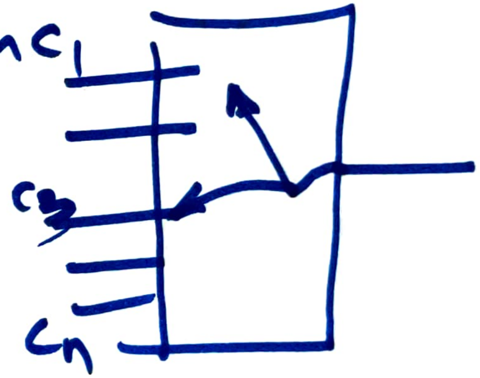

7483
B3 B2 B1 B0
A3 A2 A1 A0
Vcc
7483
S4
(Cout)
C0 (Cin)
Gnd
S3 S2 S1 S0

10

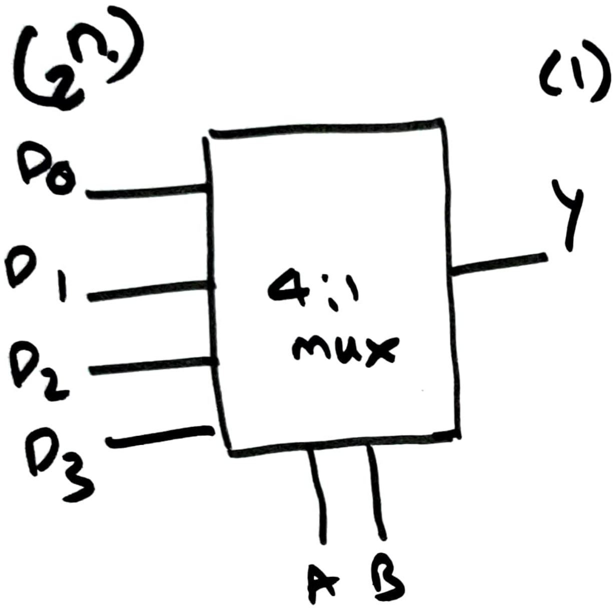


Select lines

telephone systems



$\boxed{2^n : 1}$



n → select lines

4:1 mux

Select line		o/p
A	B	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$Y = \bar{A}\,\bar{B}\cdot D_0 + \bar{A}\,B\cdot D_1 + A\,\bar{B}\,D_2 + A\,B\,D_3$$

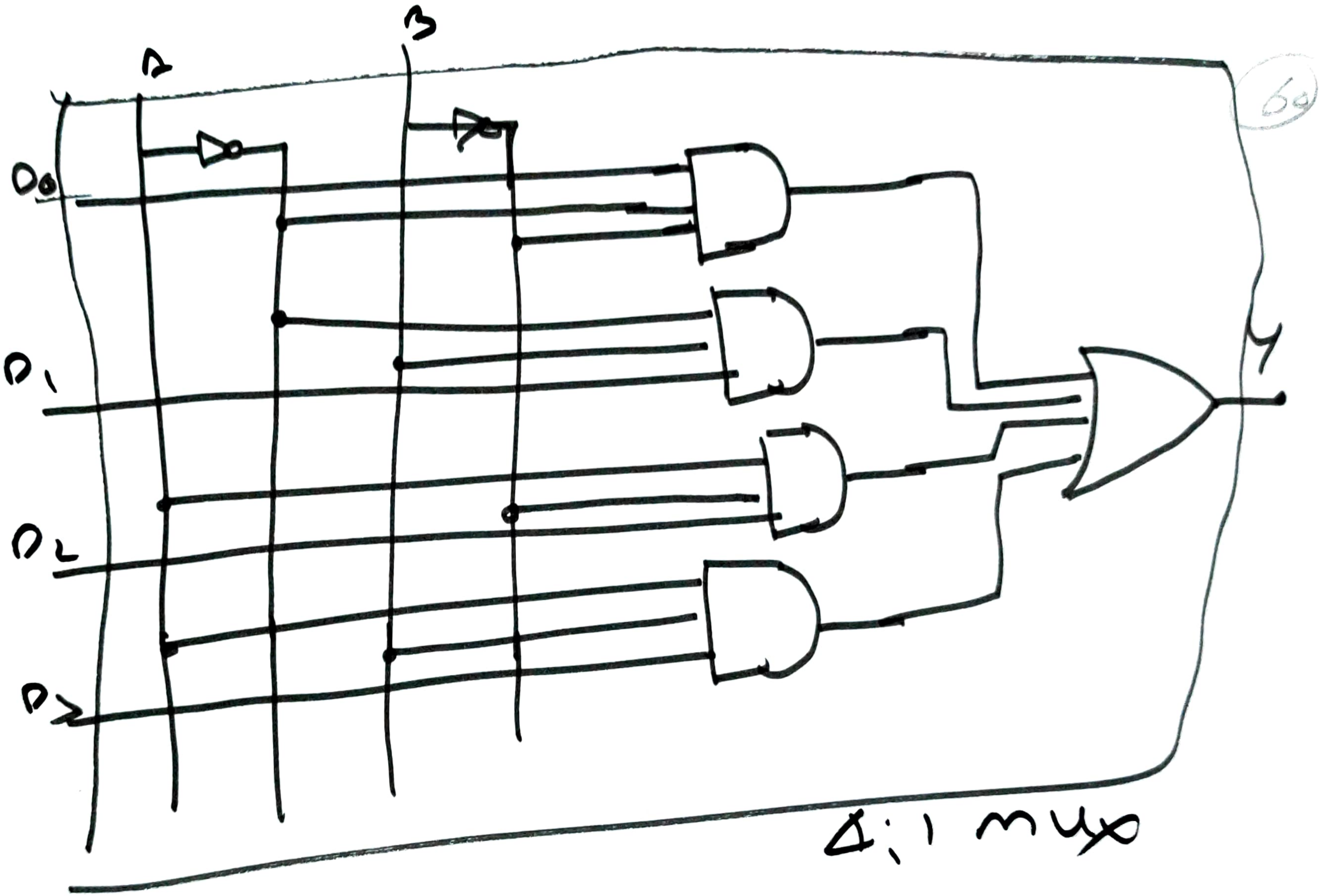
$$= 1\ 1\ 1 + 1\cdot 0\cdot 1 + 0\cdot 1\cdot 1 + 0\cdot 0\cdot 1$$

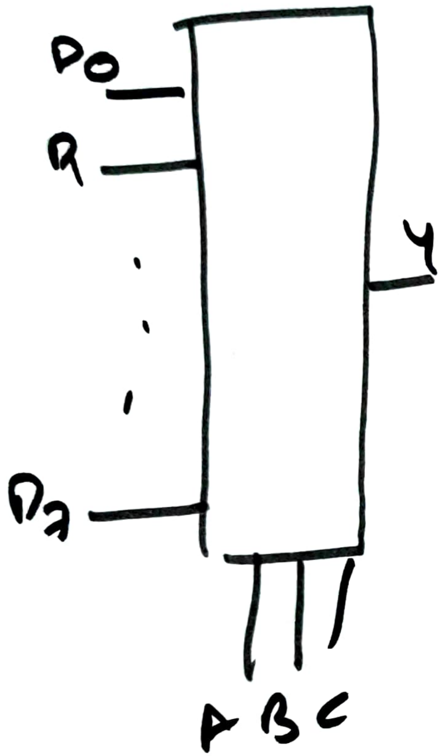
$$= 1 + 0 + 0 + 0$$

$A = 0$, $B = 0$, $D_0 = 1$

$\boxed{Y = 1}$

$\boxed{Y = 1}$


A
B
D0
D1
D2
D3
Y
4:1 mux



A	B	C	Y
0	0	0	D_0
0	0	1	D_1
0	ϕ	0	D_2
	⋮		
1	1	1	D_7

$$Y = \bar{A}\cdot\bar{B}\cdot\bar{C}\cdot D_0 + \bar{A}\;\bar{B}\;C\;D_1$$

$$+ \cdot \quad - \cdot$$

$$\cdot \quad - \quad -$$

$$+ A\,B\,C\,D_7$$

	R	G	Y	(S) O/P (Speaker)
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

$\boxed{\text{min terms}} \rightarrow$ SOP

$E = \sum m(0, 3, 5, 6, 7)$

$E = \bar{R}\bar{G}\bar{Y} + \bar{R}GY + \ldots\ldots\ldots$

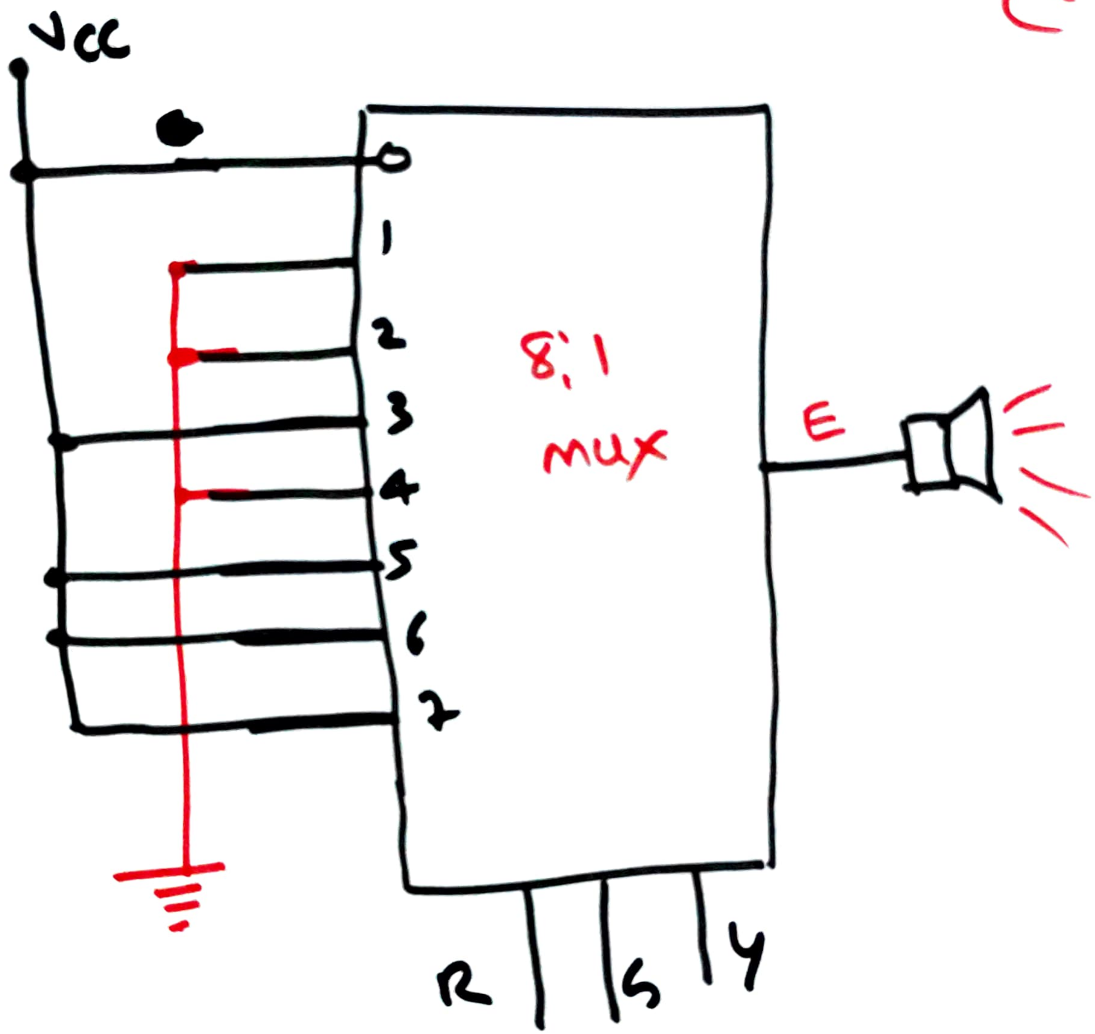
SOP [~~Produc~~ ~~as~~ Sum of Products]

$\boxed{\text{max terms}} \rightarrow$ POS

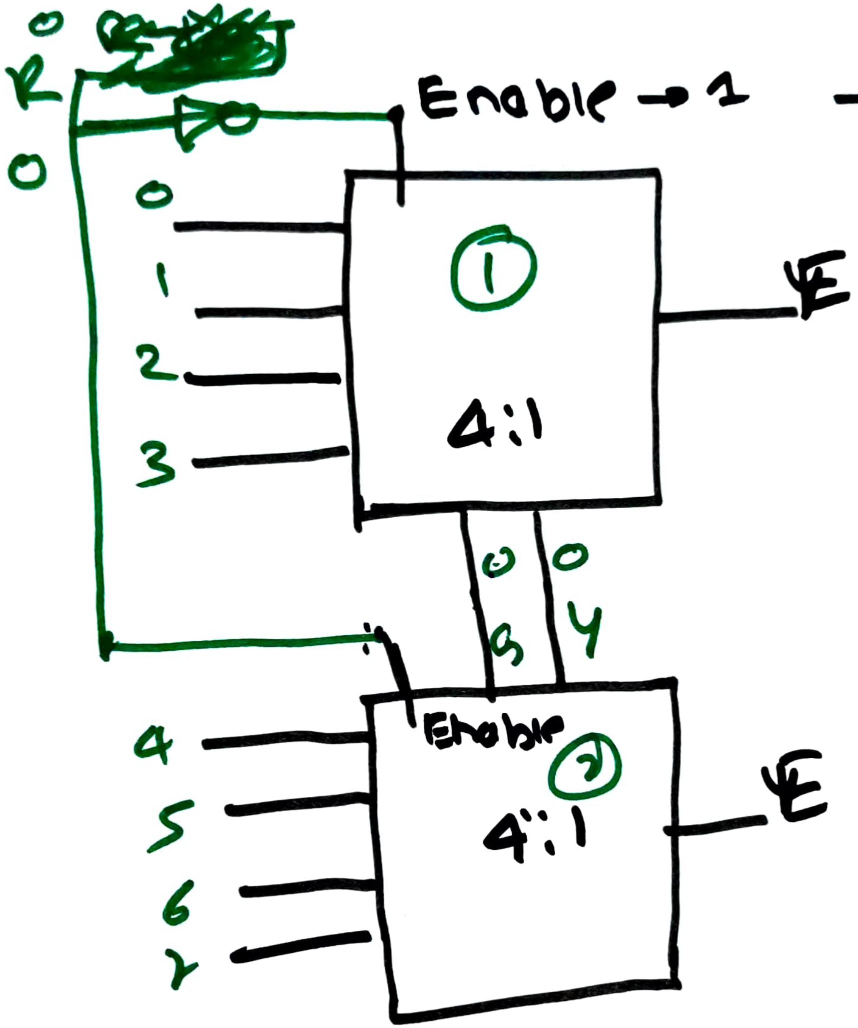
$E = \prod M(1, 2, 4)$

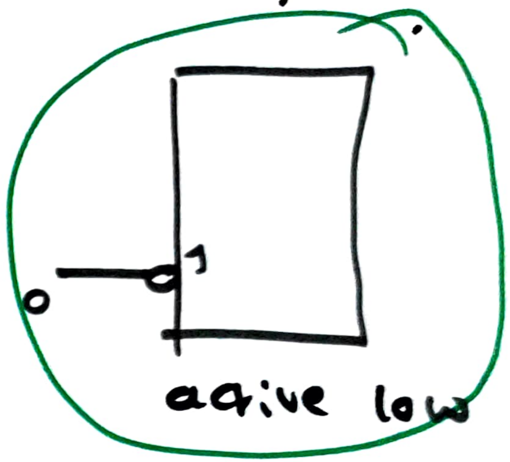
$E = (R + G + \bar{Y}) \cdot (R + \bar{G} + Y)(\bar{R} + G + Y)$

$\sum m(0, 3, 5, 6, 7)$ (8:1)



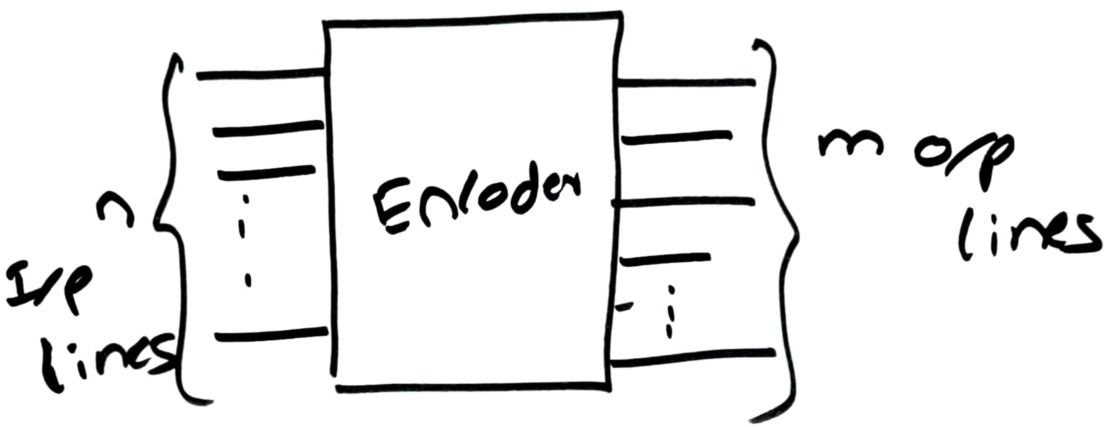
Enable → 1 → active high

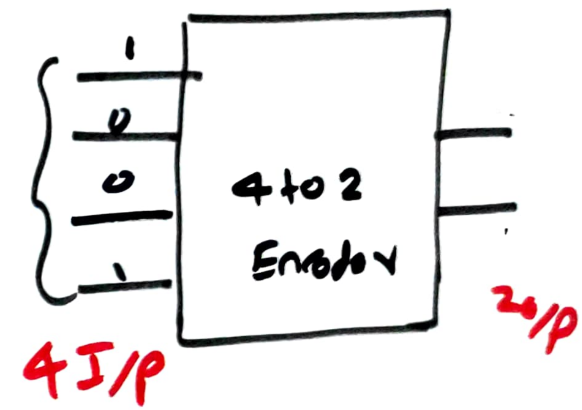




Encoder & Decoder

Encoder



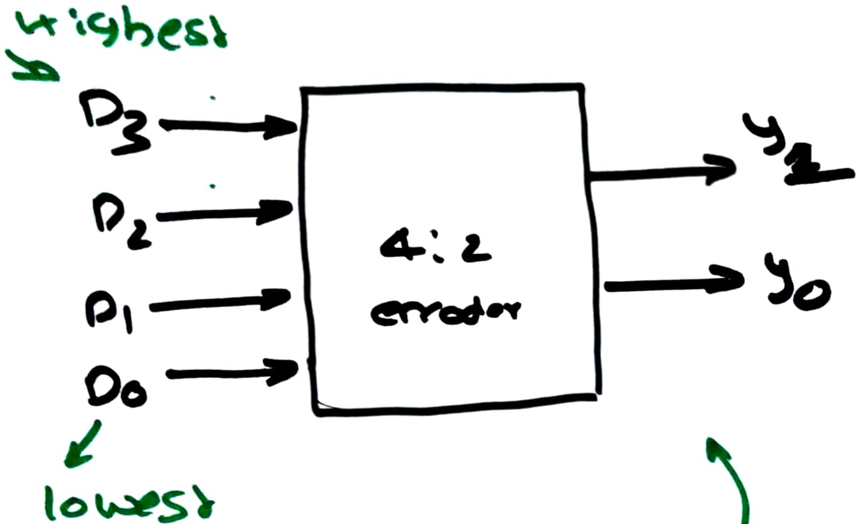


$$\boxed{n = 2^m}$$

Types

1> Priority Encoder
2> Octal to binary
3> Hexadecimal to binary
4> Decimal to BCD encoder

Priority Encoder



Block Diagram

X → Don't care

I/P (Highest → D_3, lowest → D_0)

D_3	D_2	D_1	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

Truth table

$\boxed{y_1}$

D_3D_2 \ D_1D_0	00	01	11	10
00	X	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

Group 1 = D_2

Group 2 = D_3

SOP

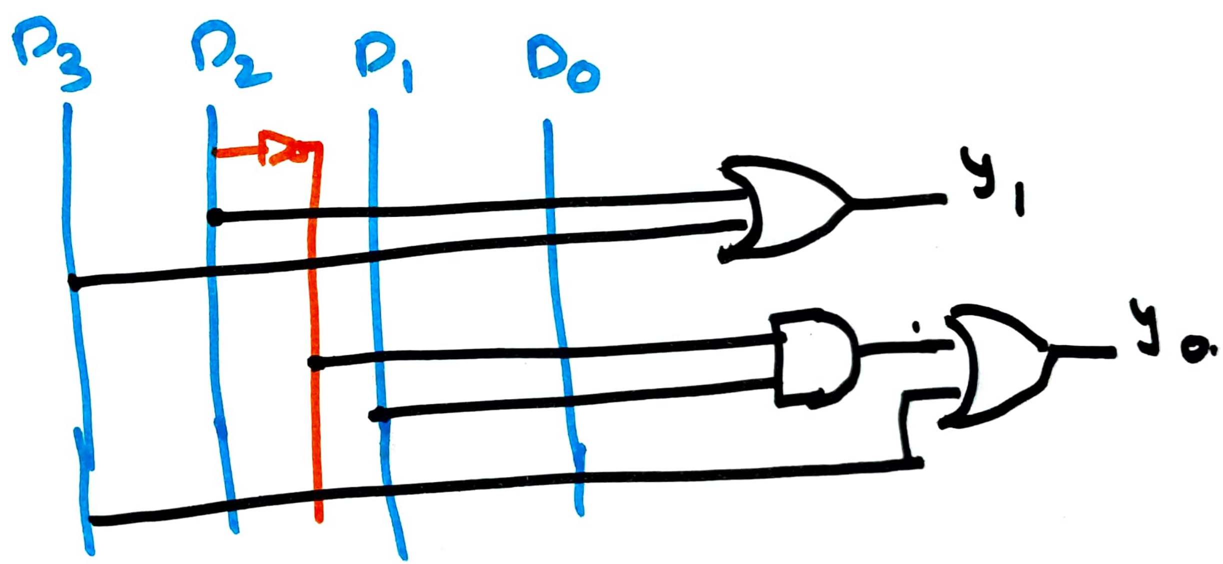
$$\boxed{y_1 = D_2 + D_3}$$

$D_3D_2 \backslash D_1D_0$	00	01	11	10
00	X	0	1	1
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

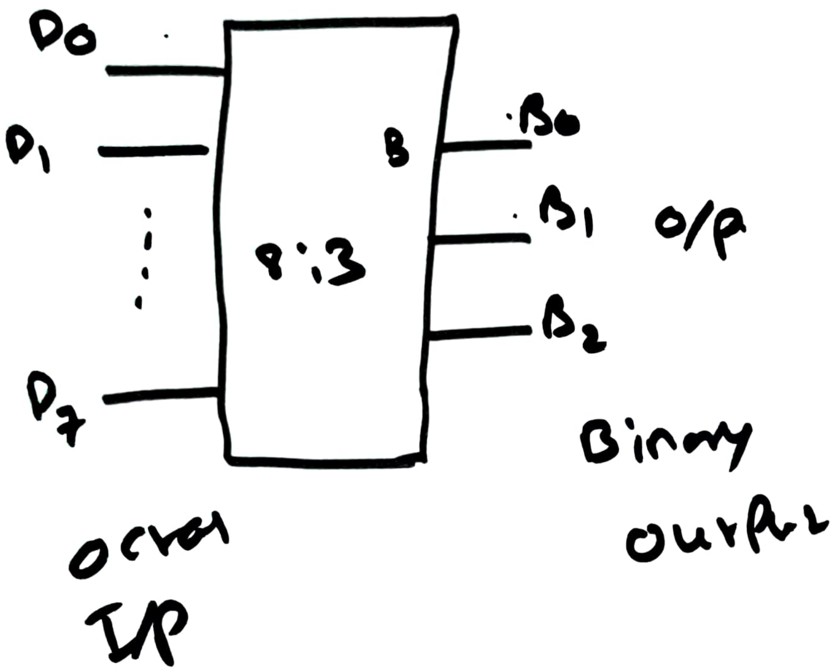
Group 2 = $\cancel{\overline{D_2} \cdot D_1}$ = $\overline{D_2} \cdot D_1$

Group 1 = D_3 = D_3

$$\boxed{y_0 = D_3 + \overline{D_2} \cdot D_1}$$



Octal to Binary number



D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	B_0	B_1	B_2
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$B_0 = D_4 + D_5 + D_6 + D_7$$

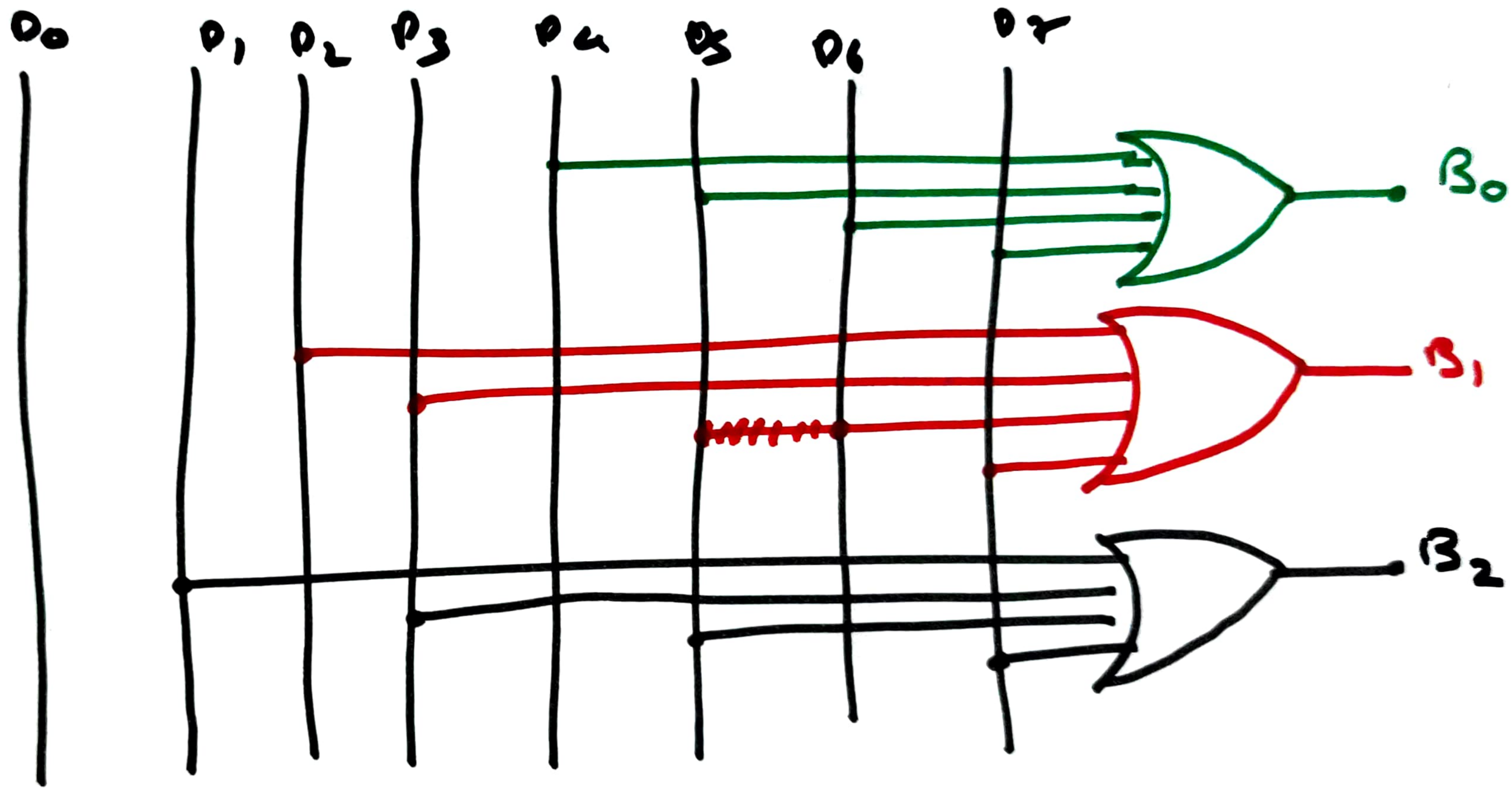
$$B_1 = D_2 + D_3 + D_6 + D_7$$

$$B_2 = D_1 + D_3 + D_5 + D_7$$

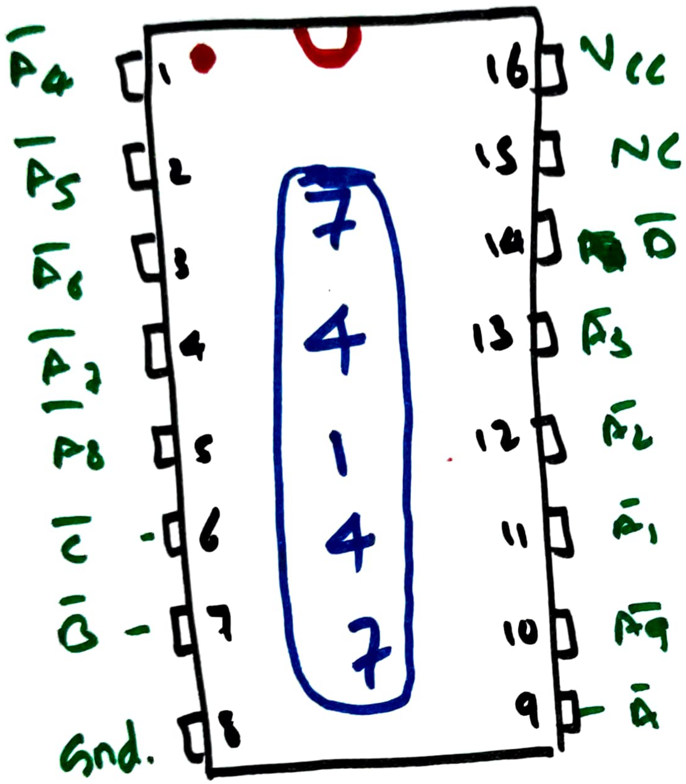
$$B_0 = D_4 + D_5 + D_6 + D_7$$

$$B_1 = D_2 + D_3 + D_6 + D_7$$

$$B_2 = D_1 + D_3 + D_5 + D_7$$



Decimal to BCD encoder (74147)



$\bar{A}_1 \ldots \bar{A}_9$ → Inputs

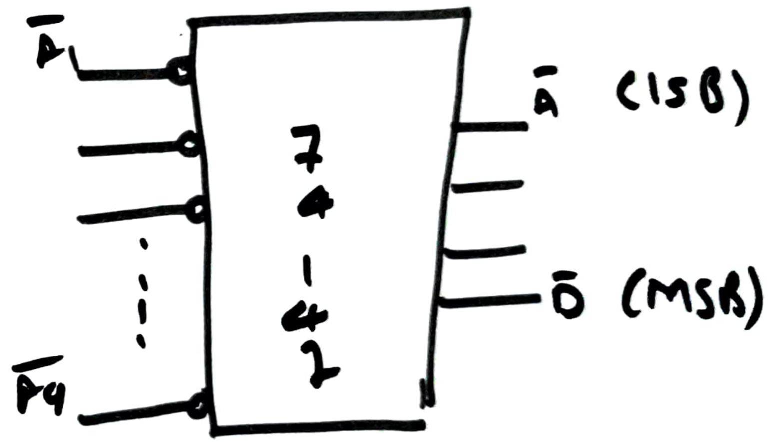
$\bar{A}$ $\bar{B}$ $\bar{C}$ $\bar{D}$ → outputs

NC → not connected

Vcc → Power supply

Gnd → Ground

NC → not connected.

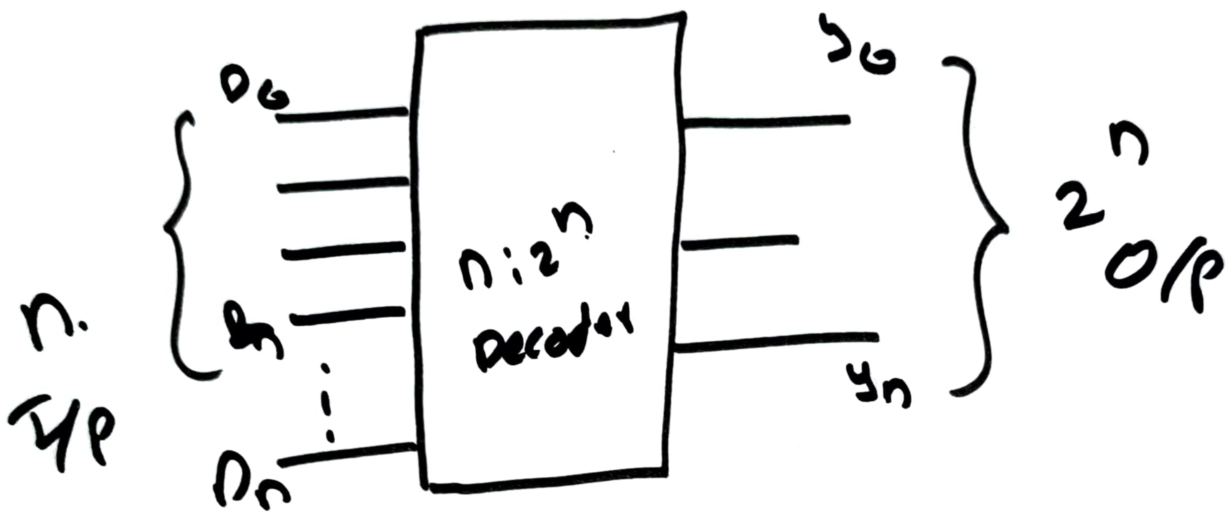


$\bar{A}_1$	$\bar{A}_2$	$\bar{A}_3$	$\bar{A}_4$	$\bar{A}_5$	$\bar{A}_6$	$\bar{A}_7$	$\bar{A}_8$	$\bar{A}_9$	$\bar{D}$	$\bar{C}$	$\bar{B}$	$\bar{A}$
1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0
ϕ	0	1	1	1	1	1	1	1	1	1	0	1

active low inputs

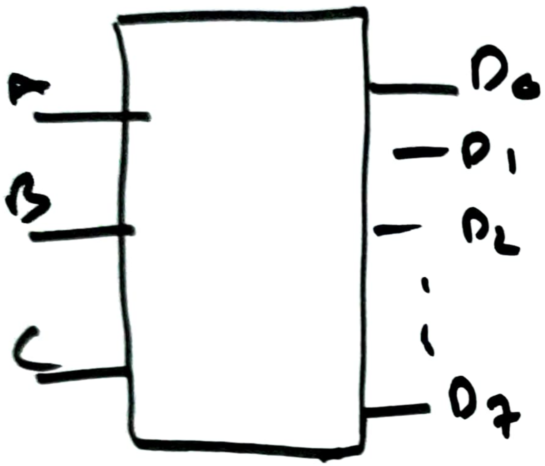
active low O/P's

Decoder



2 to 4 Decoder
3 to 8 Decoder

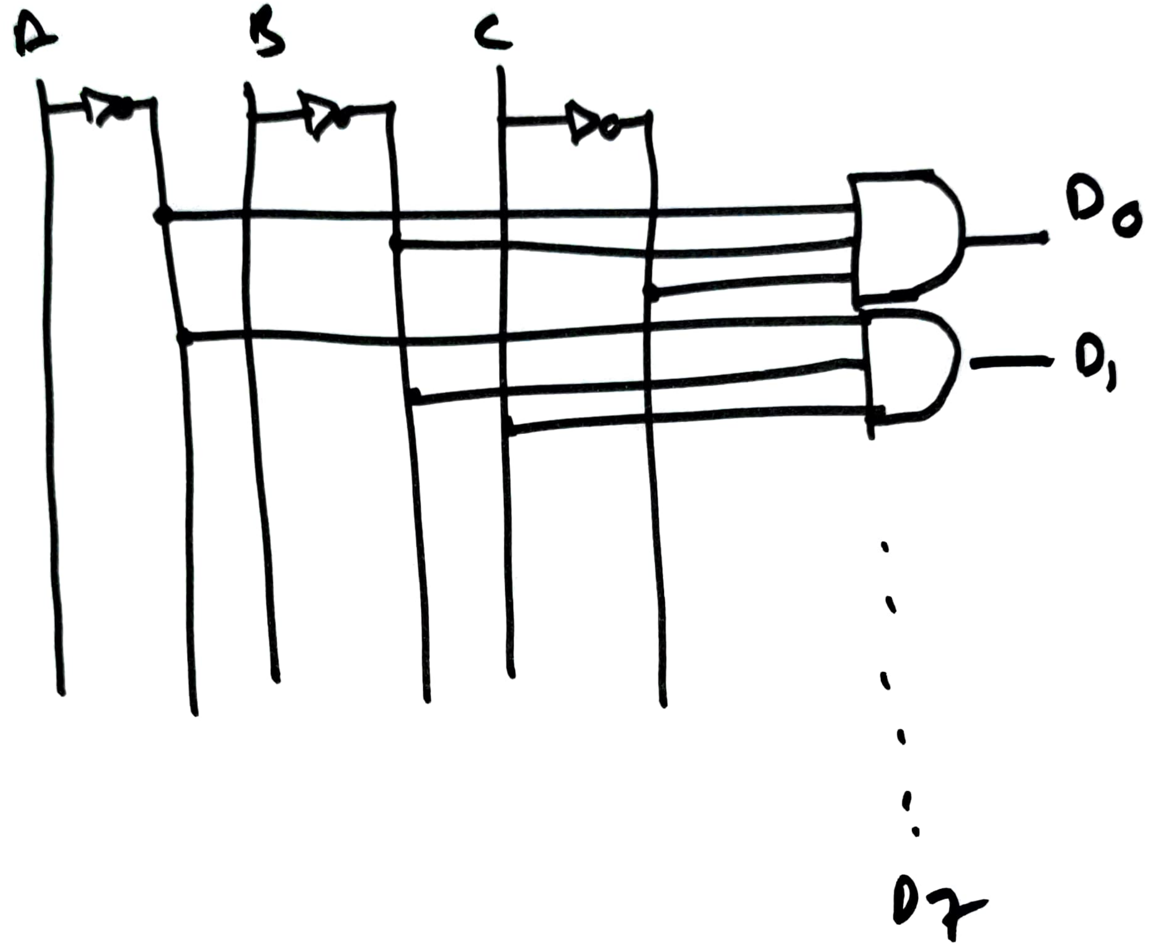
Design 3 to 8 Decoder



I/p			O/p							
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

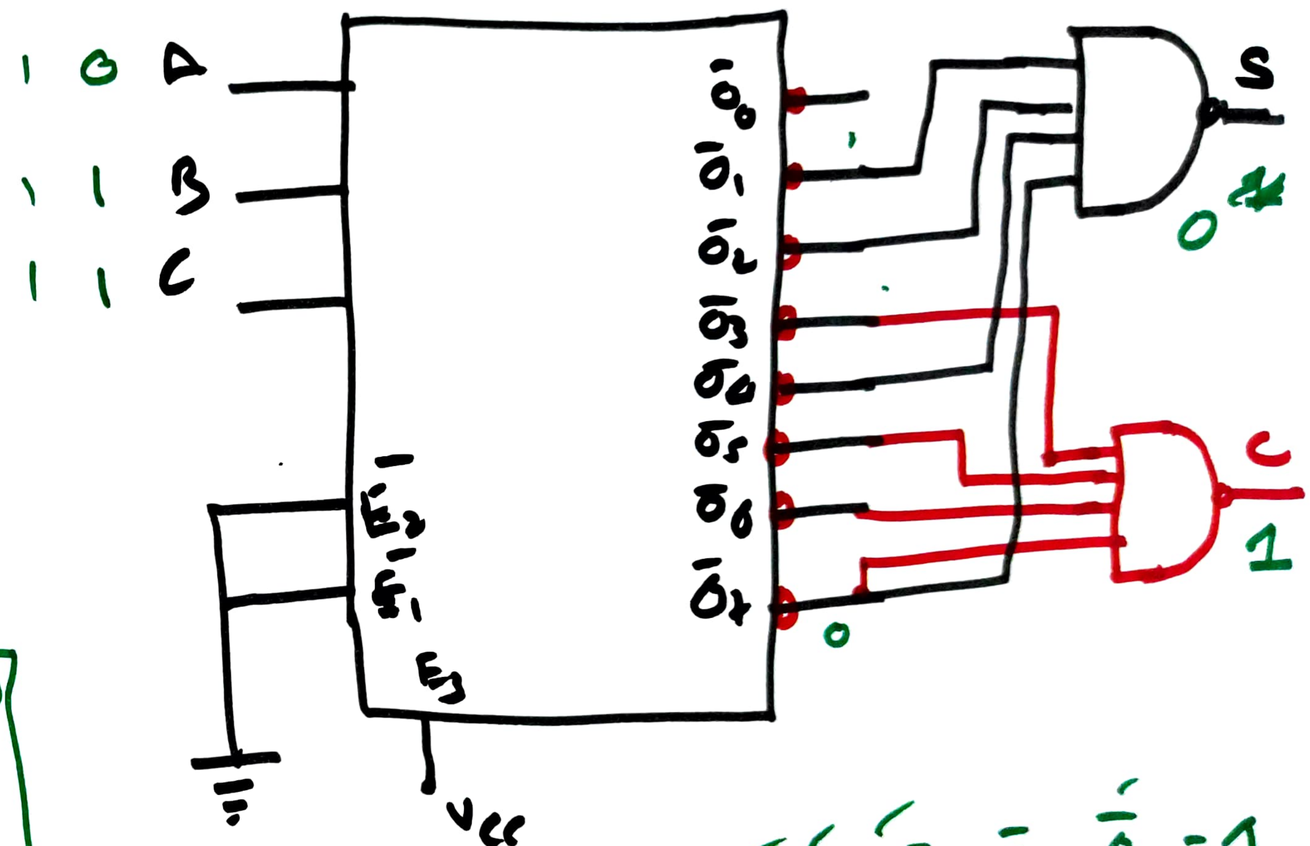
$D_0 = \bar{A}\,\bar{B}\,\bar{C}$ | $D_2 = \bar{A}\,B\,\bar{C}$ | $D_4 = A\,\bar{B}\,\bar{C}$ | $D_6 = A\,B\,\bar{C}$

$D_1 = \bar{A}\,\bar{B}\,C$ | $D_3 = \bar{A}\,B\,C$ | $D_5 = A\,\bar{B}\,C$ | $D_7 = ABC$


A
B
C
D_0
D_1
D_7

implement Full adder using 74138 decoder

A	B	C	S	Ca
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



		AND	NAND
A	B	AB	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

$A=0 \quad B=1 \quad C=1$

$\bar{O}_0, \bar{O}_1, \bar{O}_2, \bar{O}_4, \bar{O}_5, \bar{O}_6, \bar{O}_7 = 1$

$\bar{O}_3 = 0$

Test

on

Chapter - 1

[Computer Fundamenta's]

Test on Computer Fundamentals [Chapter 1]

Q1. convert (5m)

a. $(256)_8 = (?)_{10}$

b. $(101101)_2 = (?)_{10}$

c. $(ABC)_{16} = (\quad)_{10}$

d. $(777)_8 = (?)_2$

e. $(110110110)_2 = (?)_{16}$

Q2.

a. convert $(252)_{10}$ to its equivalent Gray code (1m)

b. convert $(32)_{10}$ to its Exress -3 code & Gray code (2m)

c. Find the value using Boolean Algebra (2m)

i) $A \cdot \bar{A} = ?$

ii) $A + 1 = ?$

iii) $A + \bar{A} = ?$

(iv) $A + A = ?$

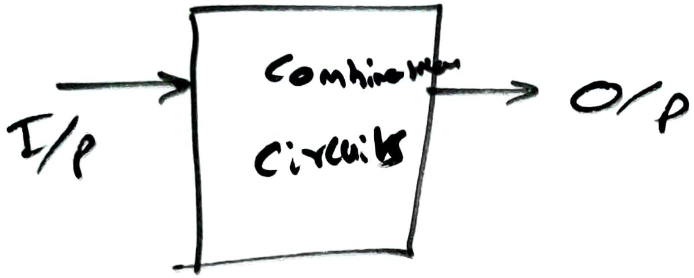
Q.3. Draw the symbol & write the truth table for the Gates (5m)

i) AND ii) OR iii) NOT

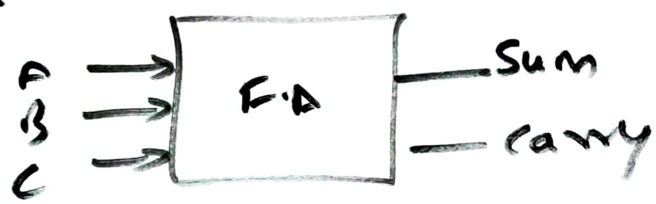
iv) NOR v) NAND

Q.4 Difference between computer organization and computer Architecture with an example **4m**

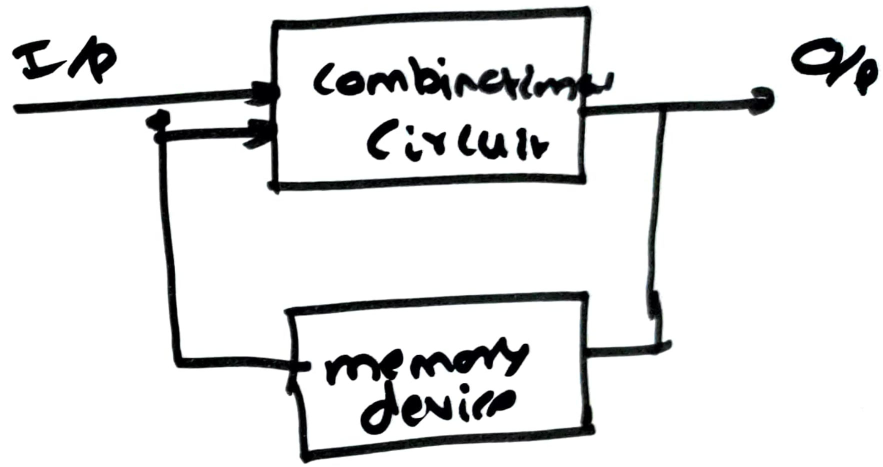
Q.5 Explain the Von-neumann model with a neat ~~block~~ Diagram **6m** combinational circuits



F. A



Sequential circuits



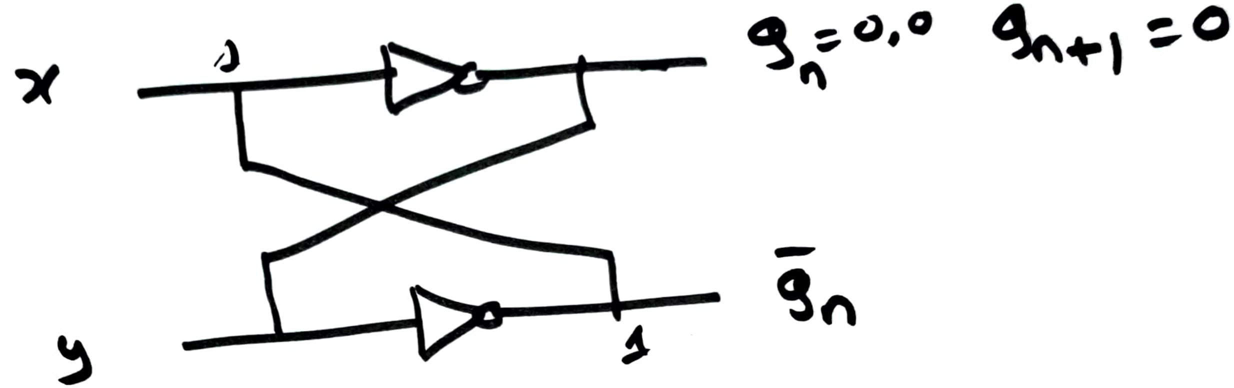
Ex

Flip Flops

Registers

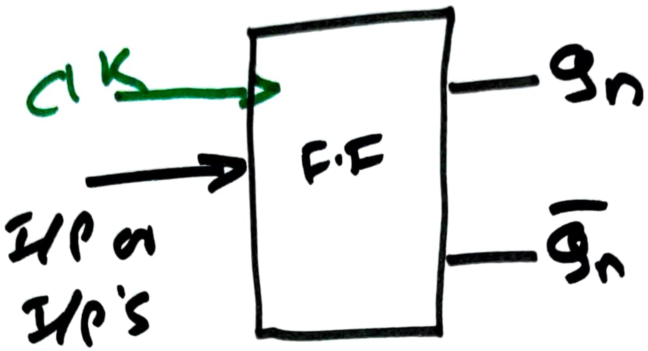
counters

latch (hold & lock)

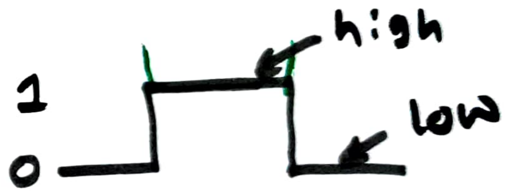


- ✓ latches as building Blocks of FlipFlops
- ✓ one bit

Flip Flop

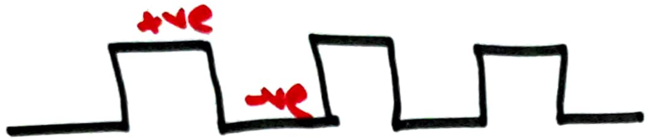


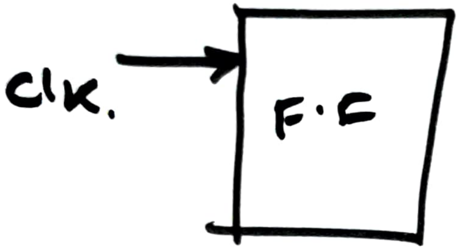
- capability to store 1 bit
- types (SR, JK, D, T)
- Bistable (0, 1)



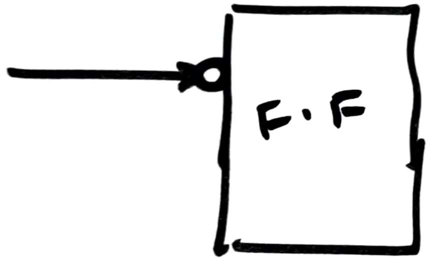
- clock synchronization

level triggered or edge triggered

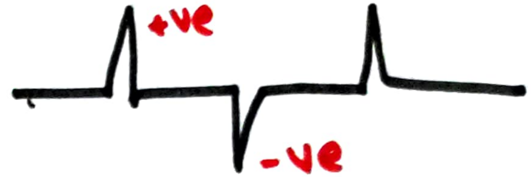


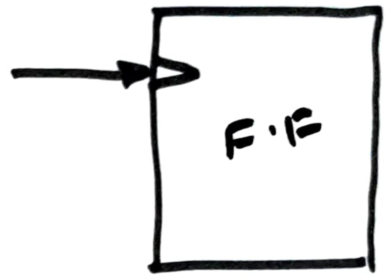


+ve level triggered F·F

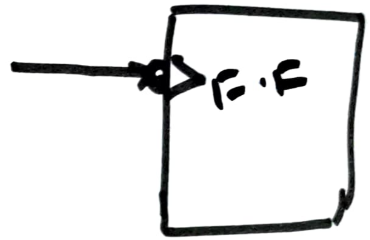


-ve level triggered F·F





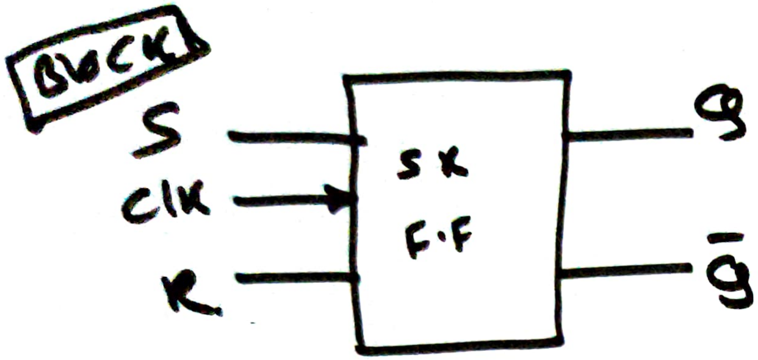
+ve edge triggered F·F



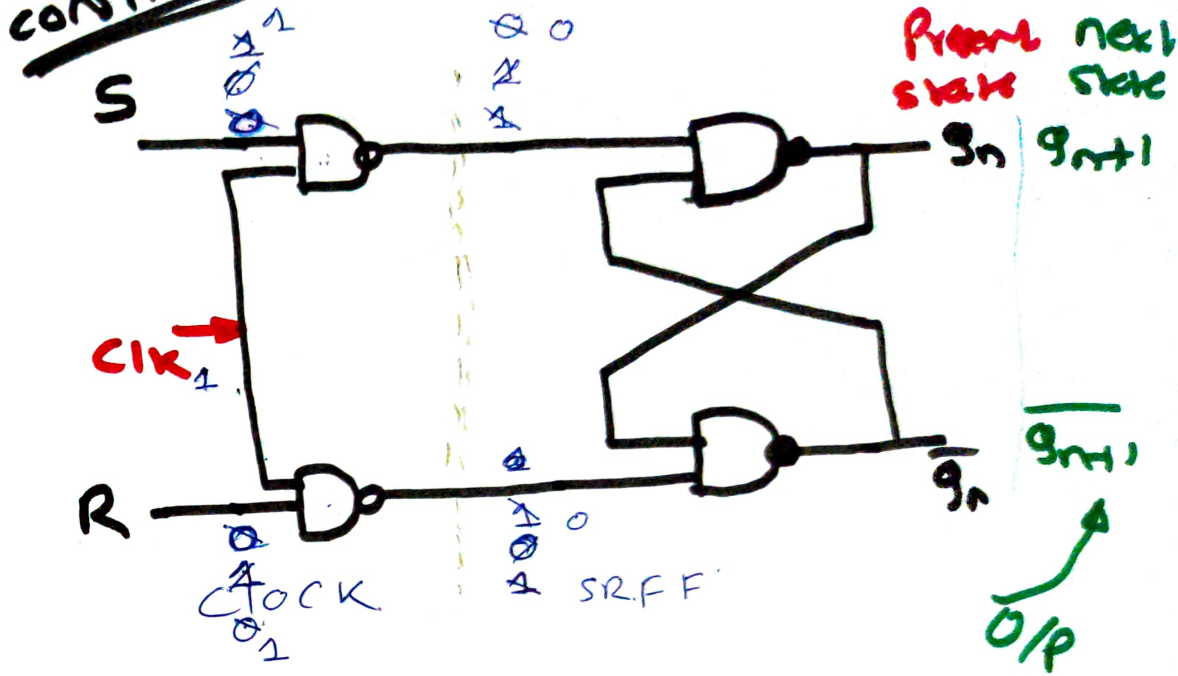
-ve edge triggered F·F

S.R Flip Flop

Block



Construction



Truth table

Clk	S	R	q_n Present state	q_{n+1} next state
↑	0	0	x	q_n
↑	0	1	x	0 → Reset
↑	1	0	x	1 → set
↑	1	1	x	Indeterminate state

case 1

$S=0 \quad R=0$

$q_{n+1} = \overline{1 \cdot \bar{q}_n} = \bar{\bar{q}}_n = q_n$

$\overline{q_{n+1}} = \overline{1 \cdot q_n} = \bar{q}_n$

case-2

$S=0 \quad R=1$

$q_{n+1} = \overline{1 \cdot \bar{q}_n} = \bar{\bar{q}}_n = q_n = 0$

$\overline{q_{n+1}} = \overline{0 \cdot q_n} = \bar{0} = 1$

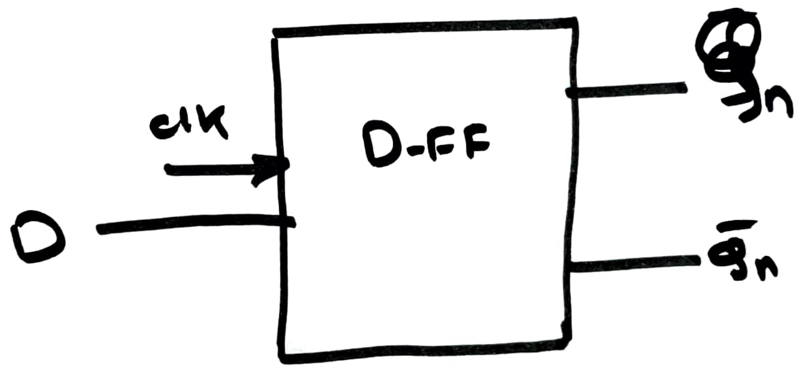
case 3

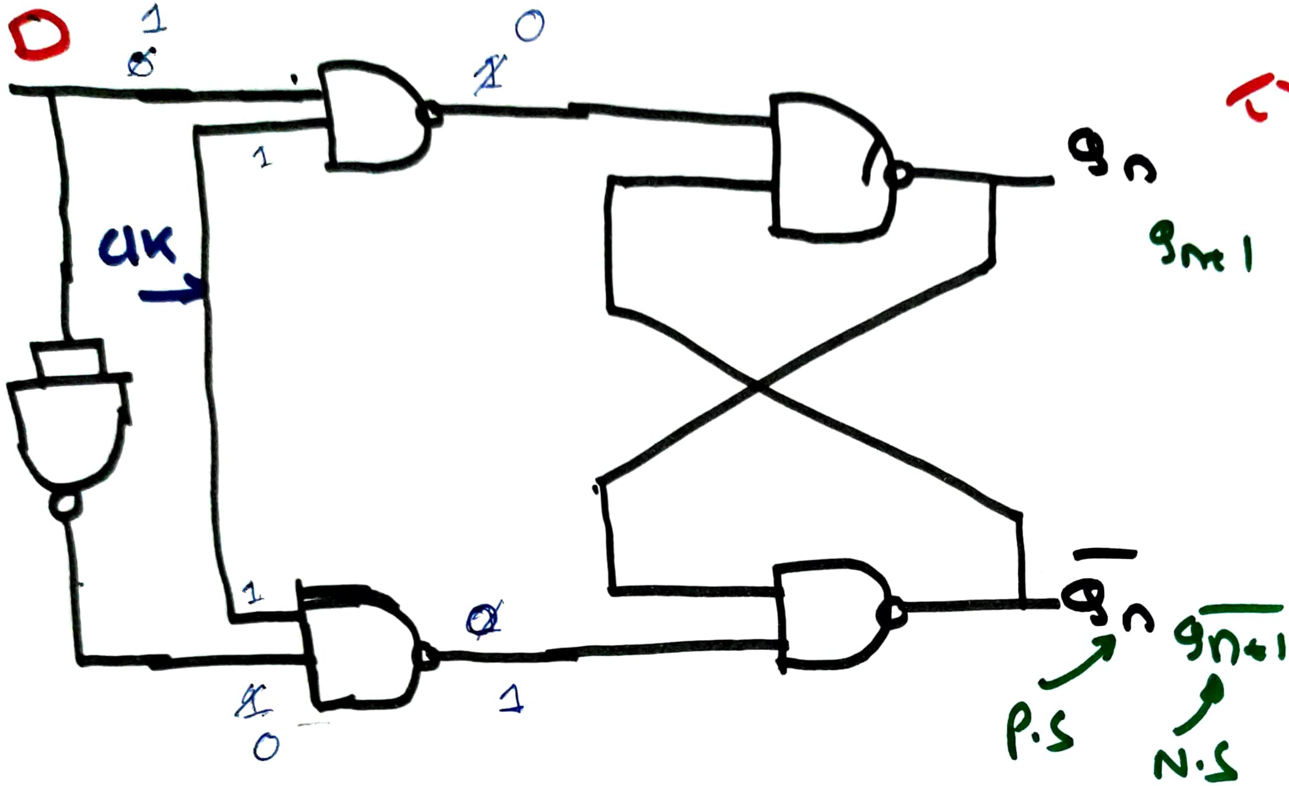
$$q_{n+1} = \overline{0 \cdot \overline{q_n}} = \overline{0} = 1$$

case 4

$$\left.\begin{array}{l} q_{n+1} = \overline{0 \cdot \overline{q_n}} = \overline{0} = 1 \\ \overline{q_{n+1}} = \overline{0 \cdot q_n} = \overline{0} = 1 \end{array}\right\} \text{Indetermin state}$$

D- Flip Flop





Truth table

D	q_n	q_{n+1}
0	X	0
1	X (0, 1)	1

Case 1 D = 0

$$q_{n+1} = \overline{1\cdot\bar{q}_n} = \bar{\bar{q}}_n = q_n \quad 0$$

$$\overline{q_{n+1}} = \overline{0\cdot q_n} = \bar{0} = 1$$

Case 2 D = 1

$$q_{n+1} = \overline{0\cdot\bar{q}_n} = \bar{0} = 1$$

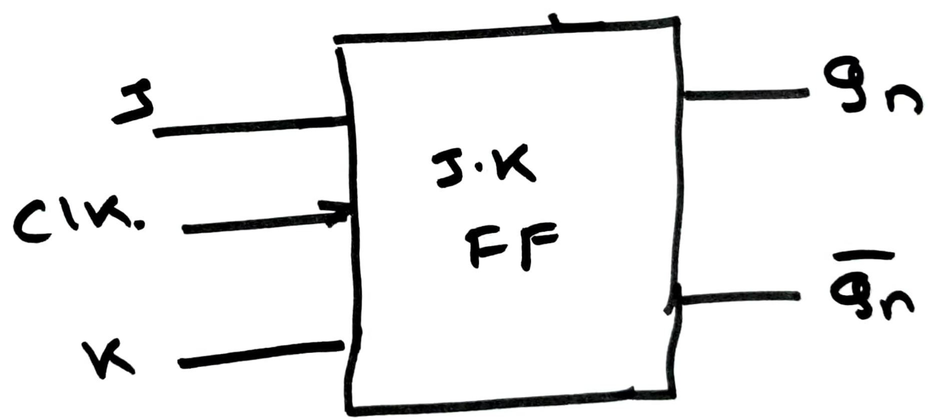
$$\overline{q_{n+1}} = \overline{1\cdot q_n} = \bar{\bar{q}}_n = 0$$

expression for D-F.F

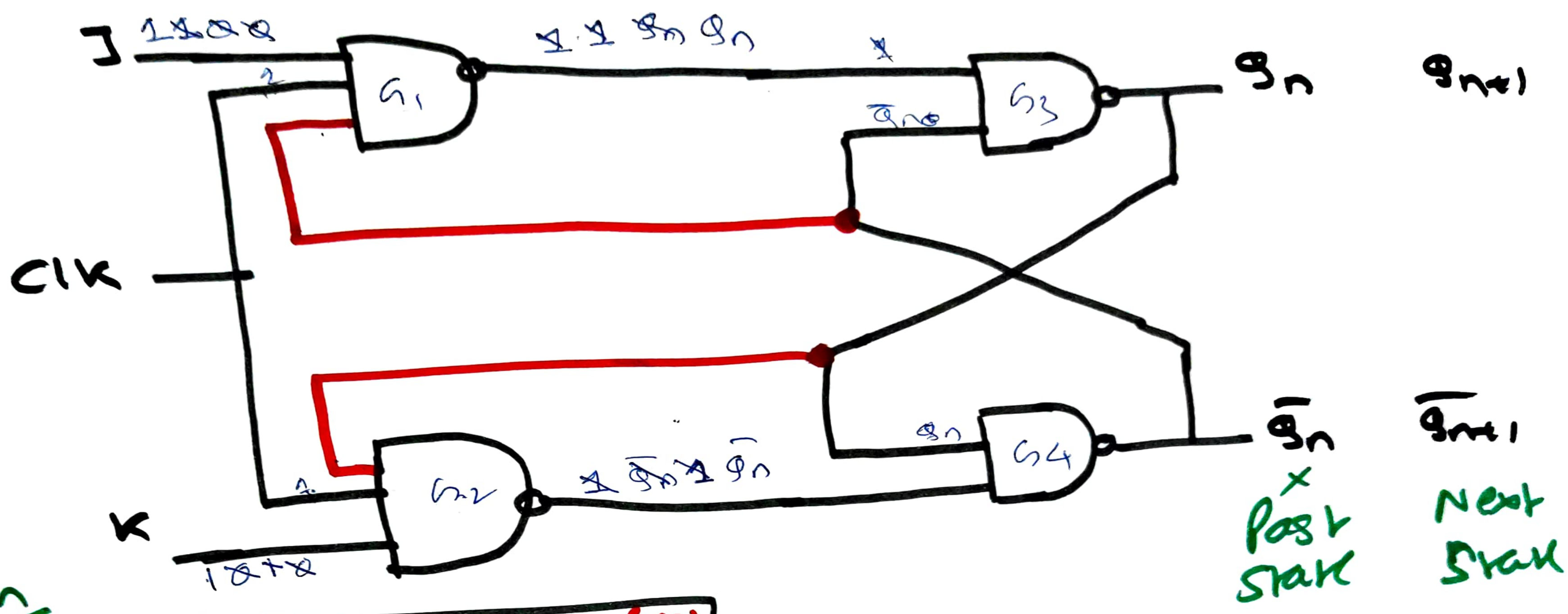
D \ Q_n	0	1
0	0	0
1	1	1

$$\boxed{Q_{n+1} = D}$$

J. K. Flip Flop



Construction



Truth table

J	K	q_n (PS)	q_{n+1} (NS)
0	0	x	q_n
0	1	x	0
1	0	x	1
1	1	x	$\bar{q}_n$

case 1 J=0 K=0

$$q_{n+1} = \overline{1 \cdot \bar{q}_n} = \bar{\bar{q}}_n = q_n$$

$$\overline{q_{n+1}} = \overline{1 \cdot q_n} = \bar{q}_n$$

case 2 J=0 K=1

$$q_{n+1} = \overline{1 \cdot \bar{q}_n} = \bar{\bar{q}}_n = q_n = 0$$

$$\overline{q_{n+1}} = \overline{q_n \cdot \bar{q}_n} = \bar{0} = 1$$

case 3 J=1 K=0

$$q_{n+1} = \overline{q_n \cdot \bar{q}_n} = \bar{0} = 1$$

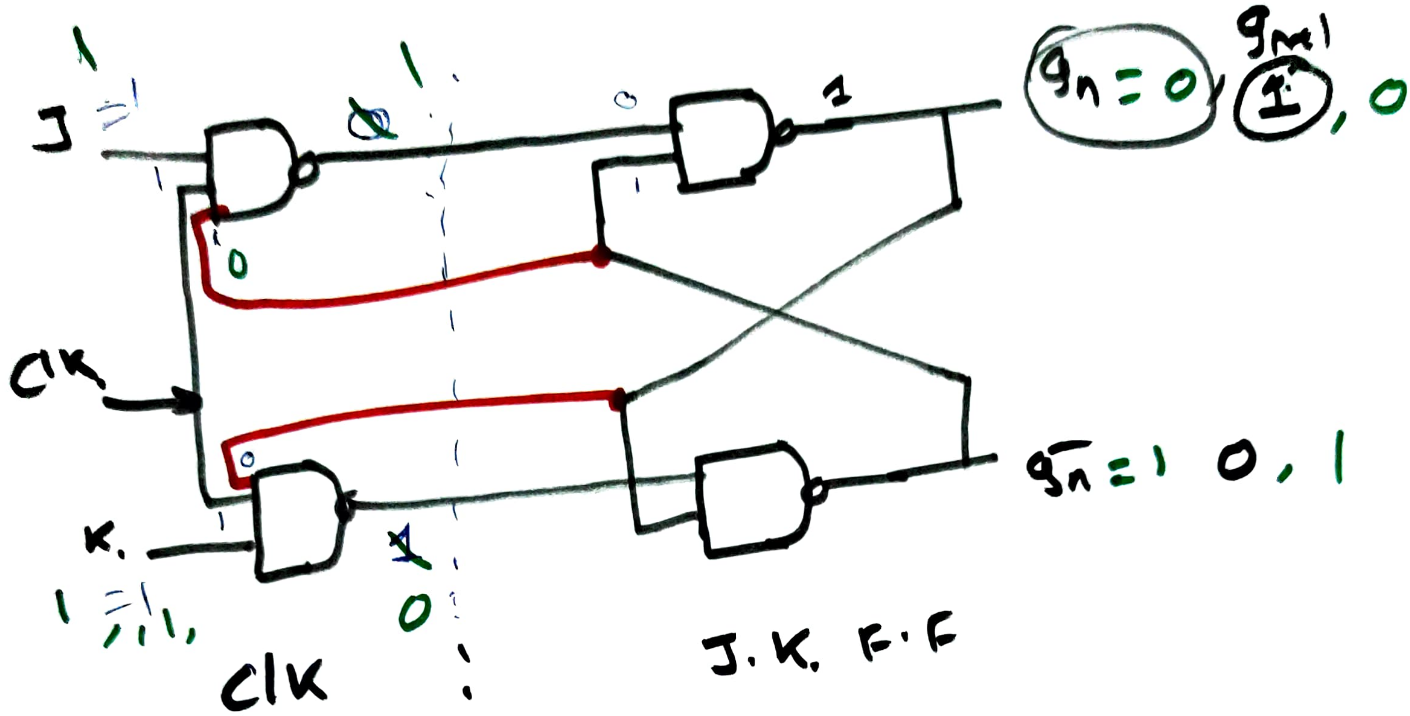
$$\overline{q_{n+1}} = 0$$

case-4 J=1 K=1

$$\overline{q_{n+1}} = \overline{q_n \cdot \bar{q}_n} = \bar{0} = 1 = \bar{q}_n$$

$$\cancel{q_{n+1}} = \cancel{\overline{q_n \cdot \bar{q}_n}} = \bar{q}_n$$

$$\boxed{\begin{array}{l} J=1 \quad K=1 \quad q_n=0 \Rightarrow \overline{1 \cdot q_n} = \bar{q}_n \\ J=1 \quad K=1 \quad q_n=1 = 1 = q_n \end{array}}$$



J = 1 K = 1

$$q_{n+1} = \overline{q_n}$$

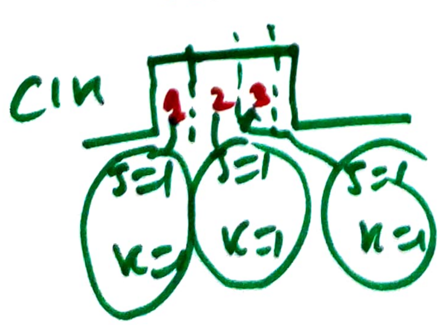
J = 1 K = 1

Race around condition

$q_n = 1$ $q_{n+1} = 0$

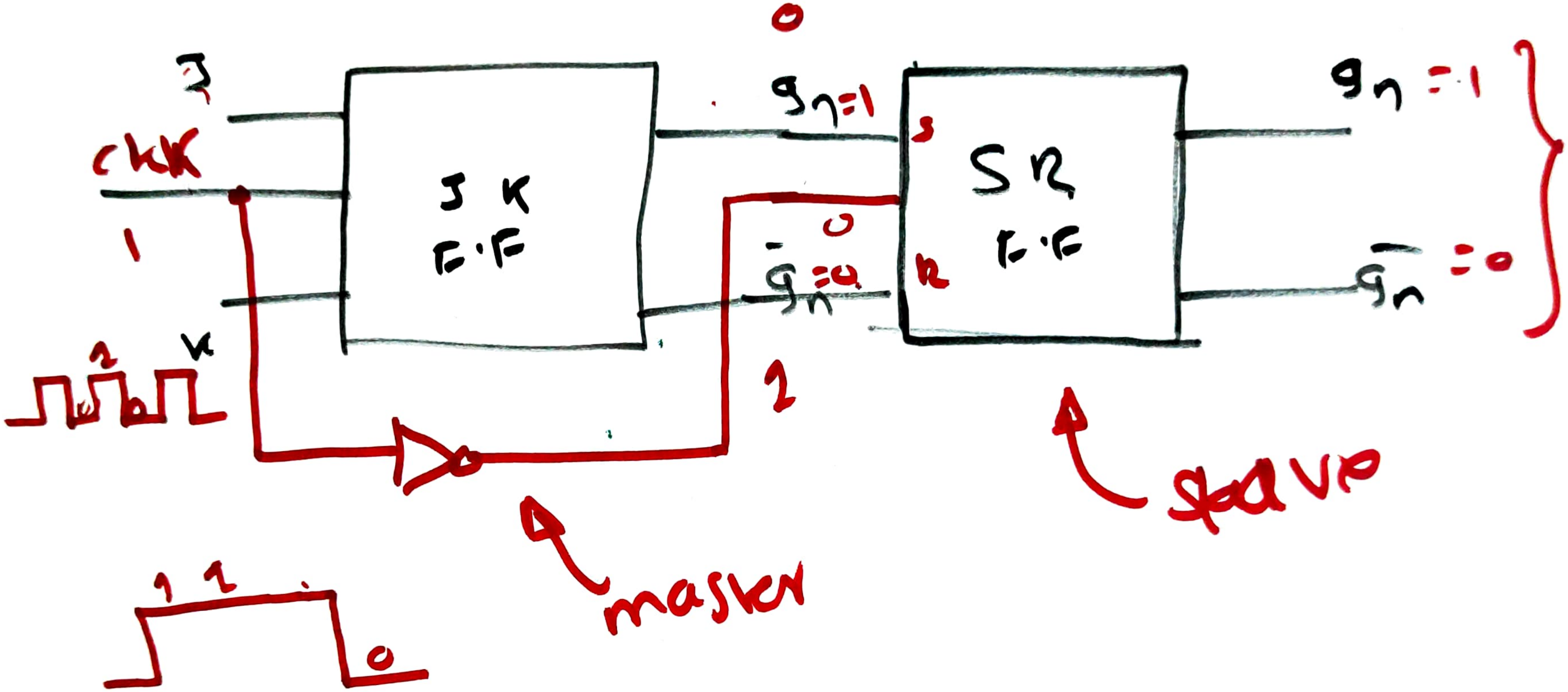
J = 1

K = 1



$q_n = 1$ $q_{n+1} = 0$ = q_{n+1} 1 = q_{n+1} 0

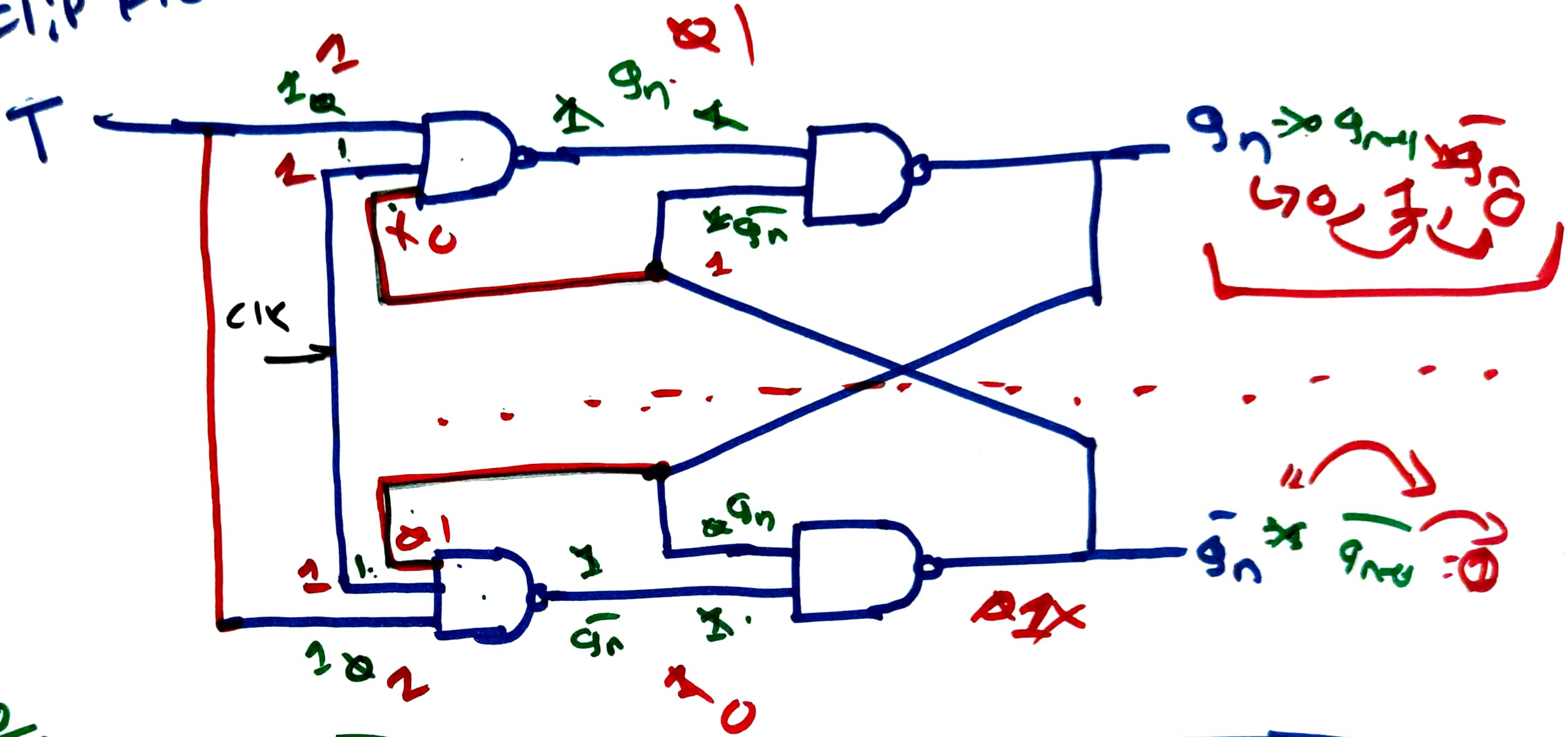
master slave J·K F·F



J=1 J=1

K=1 K=1

T-Flip Flop



$\underline{\underline{T=0}}$

$g_{n+1} = \overline{1\cdot\overline{g_n}} = \overline{\overline{g_n}} = g_n$

$\overline{g_{n+1}} = \overline{1\cdot g_n} = \overline{g_n}$

$\underline{\underline{T=1}}$

$g_{n+1} = \overline{g_n\cdot\overline{g_n}} = \overline{0} = 1$

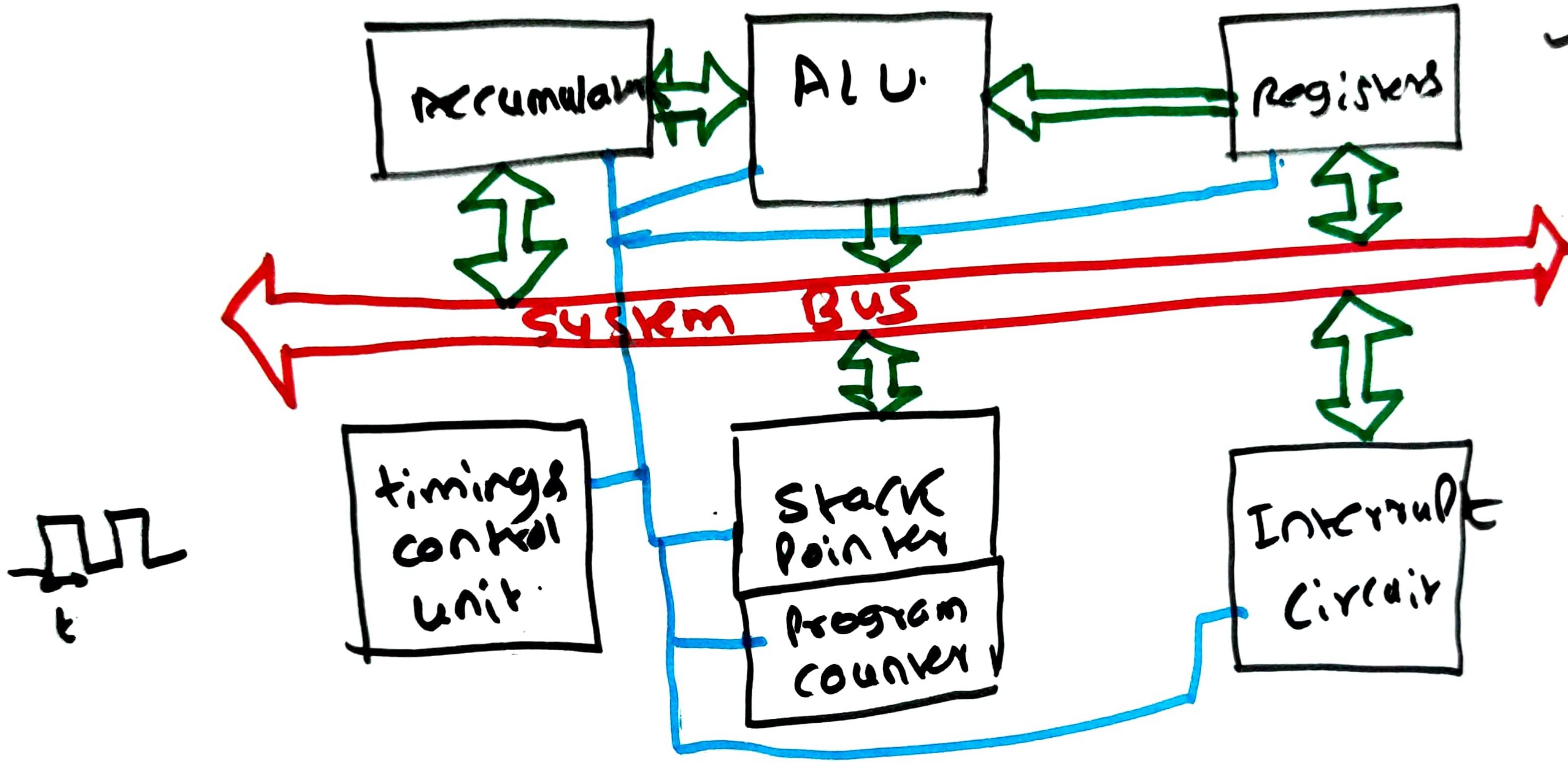
$\overline{g_{n+1}} = 0$

T	g_n	g_{n+1}
0	x	g_n
1	x	1

Characteristic table

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

CPU Architecture & Processor orginasation



8085
8086
80186
80286
80386
80486
PI, PII
C2D, DC
i5 i10...

System - Address Bus
- Data Bus
- Control Bus

single bit
multibit

Registers

1) Temporary

2) General Purpose Registers

3) Special Purpose Registers