23/7/2018

Number System

* Radix Complement:

Number = N, base = r, digits = n

$\boxed{(r-1)\text{'s complement of } N \text{ is defined as } [(r^n-1)-N]}$.

$r^n = 10^n = $ (1 followed by n 0's) -1

$= 9999\ldots\ldots$ n times

* 9's complement of 2432 = 9999 - 2432

$\boxed{1\text{'s complement} = [(2^n-1)-N]}$

* r's complement:

$\boxed{r\text{'s complement of } N = r^n - N}$

$= [(r^n-1)-N]+1$

$= (r-1)$'s complement + 1

10's complement:

9's complement of 2389 = 9999 - 2389 = 7610

10's complement = 7610+1 = 7611

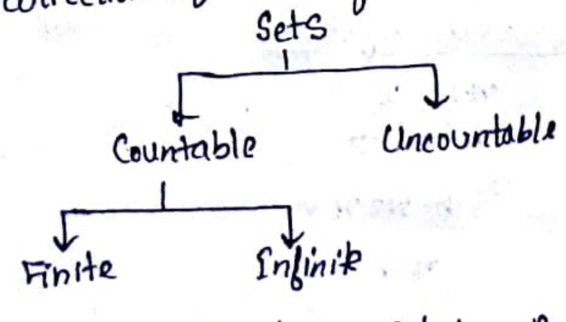
1's complement of 1010 = 0101

2's complement = 0101+1 = 0110

27/7/2018

Boolean Algebra:

Set: A collection of well defined objects.



Algebraic Structure: A set associated with a binary operation.
eg. Set of integers wrt addition (I, +).

Properties of algebraic structure:

i) Closure Property: $(A,*) \rightarrow x_1, x_2 \in A \rightarrow x_1 * x_2 \in A$. eg- $(I,-)$

ii) Associative Property: If A is any set and * is any arbitrary binary operation, $(A,*)$, then

$$x_1 * (x_2 * x_3) = (x_1 * x_2) * x_3$$

iii.) Identity Property: There should be an element such that $x+e=x \ \forall\, x \in A$ and $e \in A$, $(N,*)$

iv.) Inverse Element: $x+x^{-1}=e \quad \exists\, x^{-1}\in A$ for all $x \in A, x \neq 0$

eg. $(I,+)$, $(Q,*)$

Axiomatic Definition of Boolean Algebra:

Boolean algebra is an algebraic structure defined on a set of element B together with two binary operators + and · provided the following postulates are satisfied :

1. a.) Closure wrt + operation b.) Closure wrt · operation
2. a.) An identity element wrt + designated by 0 : $x+0=0+x=x$.
   b.) An identity element wrt · designated by 1 : $x\cdot 1=1\cdot x=x$
3. a.) Commutative wrt + b.) Commutative wrt ·
4. a.) · is distributive over + , $x\cdot(y+z)=(x\cdot y)+(x\cdot z)$
   b.) + is distributive over · , $x+(y\cdot z)=(x+y)\cdot(x+z)$
5. For every element $x \in B$, there exists an element $x' \in B$ (called the complement of x) such that $x+x'=1$, $x\cdot x'=0$.
6. There exists at least two elements $x, y \in B$ such that $x \neq y$.

Two valued Boolean Algebra:

$B=\{0,1\}$ $+,\cdot$

x	y	$x\cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

x	x'
0	1
1	0

Postulates and Theorems of Boolean Algebra:

Postulate 2	$x+0=x$	$x\cdot 1=x$
Postulate 5	$x+x'=1$	$x\cdot x'=0$
Theorem1	$x+x=x$	$x\cdot x=x$
Theorem2	$x+1=x$	$x\cdot 0=0$
Theorem3	$(x')'=x$	
Postulate 3	$x+y=y+x$	$xy=yx$
Theorem 4 Associative	$x+(y+z)=(x+y)+z$	$x(yz)=(xy)z$

De Morgan's Theorem: $\overline{x+y}=x'\cdot y'$

Absorption Theorem: $x+xy=x$ $\quad x(x+y)=x$

$$x+x=(x+x)\cdot 1=(x+x)(x+x')=x+xx'=x+0=x$$

28/1/19

Postulates of Boolean Algebra:

5 a) $x + x' = 1$

6. $x \cdot x' = 0$

Principle of Duality:
Every algebraic expression deducible from the postulates of boolean algebra remains true if the operator and identity element are interchanged.

* $x \cdot x = x$

$x \cdot x = x \cdot x + 0 = x \cdot x + x \cdot x' = x(x + x') = x$

* $x + 1 = 1$

$x + 1 = 1 \cdot (1 + x) = (x + x') \cdot (1 + x) = x + x' \cdot 1 = x + x' = 1$

By the principal of duality, $x \cdot 0 = 0$

* $x + xy = x$

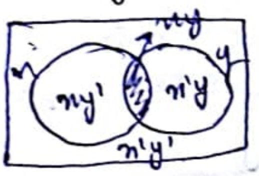
$x \cdot 1 + xy = x(1 + y) = x \cdot 1 = x$

by principal of duality $= x$

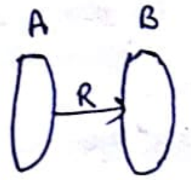
Operator Precedence:

1. Paranthesis 2. Not 3. And 4. OR

Venn Diagrams:



Boolean Function:



$R \subseteq A \times B$

Let $A = \{1, 2\}$, $B = \{3, 4\}$

$A \times B = \{(1,3), (1,4), (2,3), (2,4)\}$

Total no. of relations $= 2^{n^2}$

Total no of functions $= n^n$

* $\boxed{2^{n^2} \geqslant n^n}$

B		f1	f2	f3	f4 ...
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	0	1	1
1	1	0	1	1	1

Total no. of functions $= 2^{2^n}$

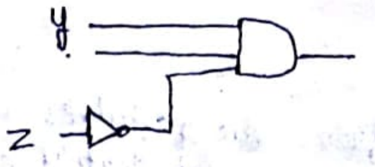
Find $f_1 = xyz'$, $f_2 = x + y'z$, $f_3 = x'y'z + xy'$

$f_4 = xy' + x'z$

x	y	z	f1	f2
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

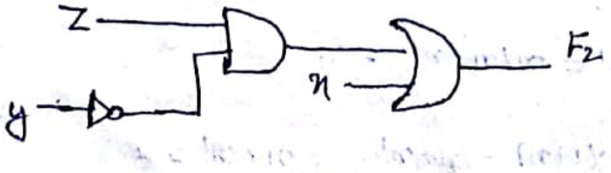
30/1/2018

Implementation of Boolean Functions with gates



$F_1 = (x+y)\cdot z'$

$F_2 = x+y'z$



Literal: Any primed or unprimed variable.

eg. $\underset{3}{\underline{x}+\underline{x'y}} = (x+x')(x+y) = 1\cdot(x+y) = \underset{2}{\underline{x+y}}$ {expr. with 3 literals converted into 2}

* $x\cdot(x'+y) = xx'+xy = 0+xy = xy$

* $\underline{xy} + \underline{x'z} + yz = xy+x'z+yz(x+x')$ {No. of terms reduced}

$= xy+x'z+xyz+x'yz$

$= xy(1+z)+x'z(1+y)$

$= \underline{xy}+\underline{x'z}$

Complement of a Function:

x	y	$x'y$	$(x'y)' = (x')'+y' = x+y'$	
0	0	0	1	1
0	1	1	0	0
1	0	0	1	1
1	1	0	1	1

In general,

$(A+B+C+D+\dots)' = A'B'C'D'\dots$

$(ABCD\dots)' = A'+B'+C'+D'\dots$

* Min terms and Maxterms:

x	y	z	Minterm	Maxterm	designation
0	0	0	$x'y'z'$	$x+y+z$	M_0
0	0	1	$x'y'z$	$x+y+z'$	M_1
0	1	0	$x'yz'$	$x+y'+z$	M_2
0	1	1	$x'yz$	$x+y'+z'$	
1	0	0	$xy'z'$	$x'+y+z$	
1	0	1	$xy'z$	$x'+y+z'$	
1	1	0	xyz'	$x'+y'+z$	
1	1	1	xyz	$x'+y'+z'$	

For n variables, no of minterms and maxterms = 2^n each

x	y	z	f1	f2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	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

Minterms:

$f_1 = x'y'z + xy'z' + xyz$

$f_2 = x'yz + xy'z + xyz' + xyz$

Maxterms:

~~[illegible]~~

$f_1 = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')$
$(x'+y'+z)$ ~~[illegible]~~

$f_2 = (x+y+z)(x+y+z')(x+y'+z)$
$(x'+y+z)$

Standard Form:

Sum of min terms:

$F = A+B'C = A(B+B') + B'C.(A+A')$
$= AB+AB'+AB'C+A'B'C$
$= AB(C+C')+AB'(C+C') + B'CA+B'CA'$
$= ABC+ABC'+AB'C+AB'C'+AB'C+A'B'C$

$F(A,B,C) = \Sigma(1,4,5,6,7)$
$= x'y'z + xy'z'+xy'z+xyz'+xyz$

Product of max terms:

$F = \Sigma(1,3,6,7) = \pi(0,2,4,5,$.

3/8/18
6 * Subtraction with complements:

Let M and N be two n digit unsigned numbers with base r, then to calculate M-N follow the steps given below:

$M + r^n - N$

(i) Add M to r's complement of N. ⇒ $M + r^n - N$

(ii) If M>,N, sum will produce an end carry r^n which is discarded.

(iii) If M<N, sum does not produce an end carry and is equal to $r^n-(N-M)$ which is r's complement of N-M. To obtain the answer in a familiar form, take the r's complement of sum and precede with a negative sign.

Q. Using 10's complement find 72532-3250.

M = 72532, N = 03250, r=10

⇒ 10's complement of N = ~~[illegible]~~ $10^5 - 03250$ = 9's complement + 1
= 96749+1
= 96750

$M + r^n - N$ = 72532+96750
= 169282

1 is carry.

⇒ Ans = 169282 − 100000 = 69282

Q. 3250 − 72532

M = 3250, N = 72532

10's complement of N = 10^5 − 72532 = 27467 +1 = 27468

$M + r^n - N$ = 30718

10's complement of 30718, = 69281 + 1 = 69282

Result = − 69282

10001

carry

* Q. X = 1010100, Y = 1000011. Perform subtraction (i) X−Y (ii) Y−X using 2's complement.

i) 2's complement of Y = 0111100 + 1 = 0111101

$X + r^n - Y$ = 1010100 + 0111101 = 10010001

1 is carry

Ans = 10010001 − 10000000 = 0010001

ii) 2's complement of X = 0101011 + 1 = 0101100

$Y + r^n - X$ = 1101111

2's complement of $Y + r^n - X$ = 0010000 + 1 = 0010001

Ans = − 10001

Signed Binary Numbers:

Sign is written in binary form against the no. −10

1 = −ve, 0 = +ve

Signed magnitude notation: Signed 1's complement notation:

+9 = 01001 ⇒ −9 = 1's complement of 9 = 10110

$$\begin{array}{r} 10110 \\ +1 \\ \hline 10111 \end{array}$$ (2's comp)

−9 with 8 bits → $\underset{\text{Sign}}{\underline{1}}$ 0001001

+9 in 8 bits = 00001001

1's complement of −9 = 11110110

Signed 2's complement representation:

11110110 + 1 = 1111 0111

6/8/19

* Binary Codes:

* No. of different patterns possible with n bits of 0 and 1 = 2^n

If we have 2^n different objects, then we need at least n bits to distinguish between them.

i) BCD (Binary Coded Decimal)

0	0000
1	0001
2	:
3	:
:	:
9	

BCD code for 10 = 00010000 ≥ 1010 ← Binary

for 11 = 00010001 = 1011

(not equivalento to binary)

ii) Excess 3:

9 = 1100

0	0011
1	0100
2	0101
:	:
9	1100

iii) 84-2-1:

0	0000	
1	0111	
2	0110	1×4−1×2

iv) 2421:

0	0000
1	0001
2	0010 / 1000

* BCD code is used when internal arithmetic in the computer is done in decimal form.

eg 12+13 = 00010010
+00010011

* The excess 3, 2421 and 84-2-1 are self complementing codes. 9's complement of a no. is obtained by changing 1's to 0's and 0s to 1s.

* 1001 → 9 (BCD), 6 (excess 3), 7 (84-2-1), 3 (2421), −1 (sign mag rep)
−6 (signed 1's complement), −7 (signed 2's complement)

Q. 23−18 = ?

$$\begin{array}{r} 23 \\ 82 \\ \hline 105 \end{array}$$ Ans = 5

Q. 1011 − 0100 = ? 84-2-1

9's complement of 0100 = 1011

* Error Detection Codes:

i) Parity Scheme:

Odd Parity		Even Parity	
Message	Parity	Message	Parity
0000	1	0000	0
0001	0	0001	1

Can not detect more than one error.

ii) Gray Code:

Gray Code	Decimal equivalence
0000	0
0001	1
0011	2
0010	3
0110	4
0111	5
0101	6

There will be change in only one bit b/w succesive decimal equivalents.

10/8/18

* Two and Three Variable Maps (K-Maps):

x	y		
0	0	$x'y'$	m_0
0	1	$x'y$	m_1
1	0	xy'	m_2
1	1	xy	m_3

x \ y	0	1
0	m_0	m_1
1	m_2	1 m_3

F = ~~xy~~ xy

x \ yz	00	01	11	10
0	m_0	m_1	1 m_3	1 m_2
1	1 m_4	1 m_5	m_7	m_6

$$F(x,y,z) = \Sigma(2,3,4,5) = m_2+m_3+m_4+m_5$$

$$x'y + xy' = x \oplus y$$

$$F(x,y,z) = \Sigma(3,4,6,7)$$

x \ yz	00	01	11	10
0			1	
1	1		1	1

$yz + xz'$

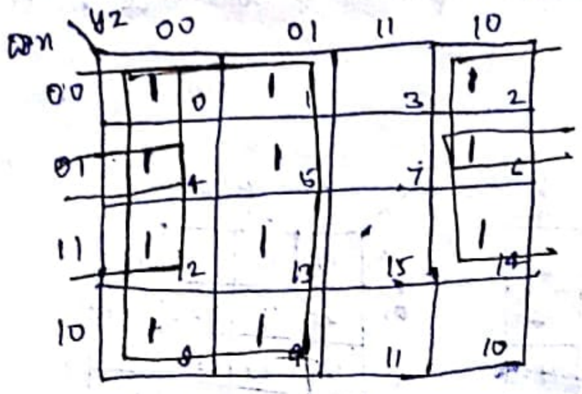
$$F(x,y,z) = \Sigma(0,2,4,5,6)$$

x \ yz	00	01	11	10
0	1			1
1		1		1

~~[illegible]~~

$z' + xy'$

$F(w,x,y,z) = \Sigma(0,1,2,4,5,6,8,9,12,13,14)$



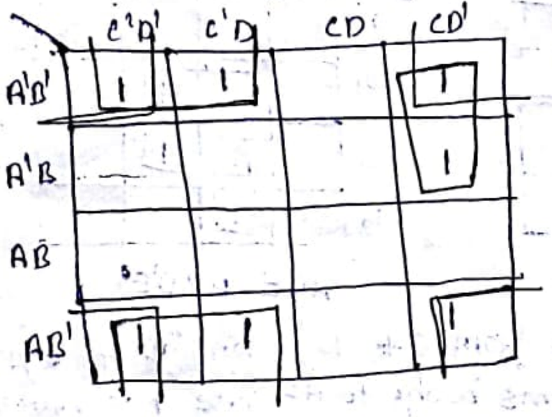
wx \ yz	00	01	11	10
00	1 0	1 1	3	1 2
01	1 4	1 5	7	1 6
11	1 12	1 13	15	1 14
10	1 8	1 9	11	10

$y' + w'z' + xz'$

Simplify the Boolean function $F = A'B'C' + B'CD' + A'BCD' + AB'C'$.

$F = A'B'C'(D+D') + (A+A')B'CD' + A'BCD' + AB'C'(D+D')$

$= A'B'C'D + A'B'C'D' + AB'CD' + A'B'CD' + A'BCD' + AB'C'D + AB'C'D'$



	C'D'	C'D	CD	CD'
A'B'	1	1		1
A'B				1
AB				
AB'	1	1		1

$B'C' + B'D' + A'CD'$

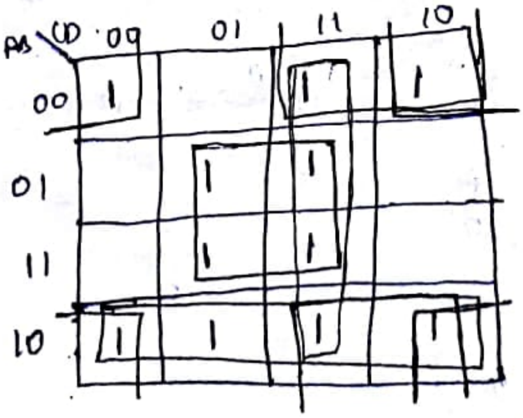
20/8/2018

* Prime Implicant:

A prime implicant is a product term obtained by combining the maximum possible no. of adjacent squares in the map. A single 1 on a map represents a prime implicant if it is not adjacent to any 1.

eg. $F(A,B,C,D) = \Sigma(0,2,3,5,7,8,9,10,11,13,15)$

$\bar{B}\bar{D} + BD + A\bar{B} + \bar{B}C$



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

BD, B'D' are essential PI.s.

AD, AB', CD, B'C are PI's.

$F = BD + B'D' + CD + AD = BD + B'D' + CD + AB'$

$= BD + B'D' + B'C + AD = BD + B'D' + B'C + AB'$

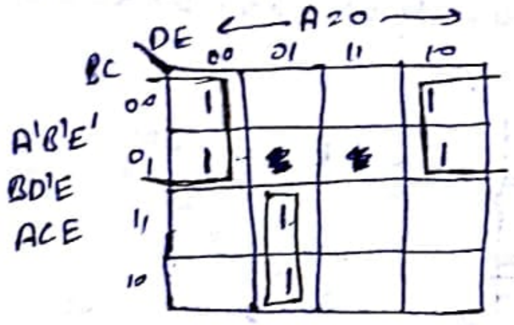
* If a minterm in a square is covered by only one prime ~~implication~~ implification, that implicant is said to be an essential prime implicant.

Five Variable Maps:

BC \ DE (A=0)	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

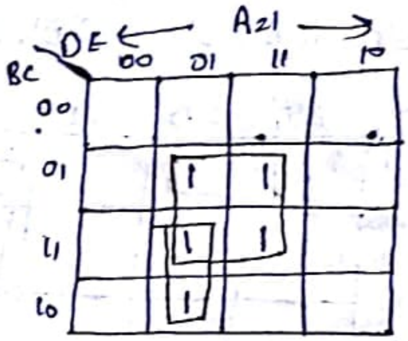
A=1	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

eg. F(A, B, C, D, E) = (0, 2, 4, 6, 9, 13, 21, 23, 25, 29, 31)



BC \ DE	00	01	11	10
00	1			1
01	1			1
11		1		
10		1		

~~A'B'E', A'BD'E~~



BC \ DE	00	01	11	10
00				
01		1	1	
11		1	1	
10		1		

ACE, ABD'E

There are 6 minterms from 0 to 15 belonging to A=0 part of the map. Other 5-terms belong to A=1. The two squares in Column 01 and the last two rows are common to both parts of the map.

31/8/18

Circuit:

There are two types of circuit:

i) Combinational

ii) Sequential

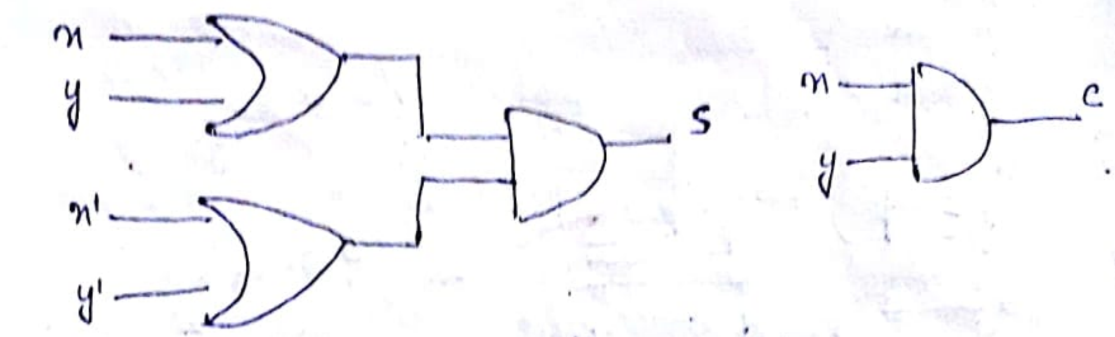
Adder Design:

Bit 1		Bit 2	sum	carry
0	+	0	0	0
0	+	1	1	0
1	+	0	1	0
1	+	1	0	1

S = x'y + xy'

C = xy

Two OR Gates, Two AND Gate:



$S = (x+y)(x'+y')$

$= xx' + xy' + yx' + yy'$

$= xy' + x'y$

* $S = (C + x'y')'$

$C = xy$

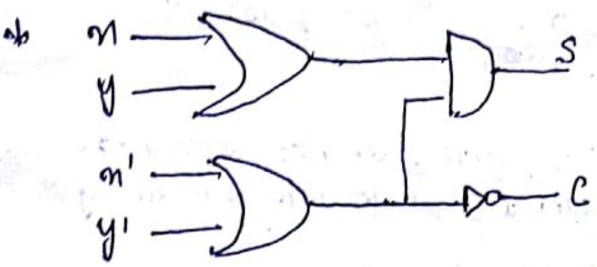
$(A+B)' = A'B'$

$= (x'+y')(x+y)$

$= x'x + x'y + y'x + y'y$

$= x'y + y'x$

*



$C = (x'+y')' = xy$

$S = (x+y)(x'+y')$

$= xy' + x'y$

* Full Adder:

Half adder

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

* $S = x'y'z' + x'yz' + xy'z' + xyz$

$C = x'yz + xy'z + xyz' + xyz$

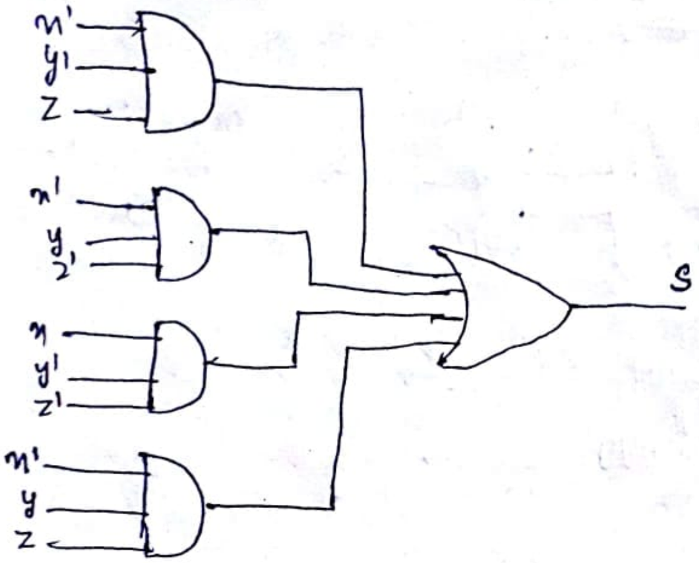
carry $= xz + xy + yz$

x \ yz	00	01	11	10
0		1		1
1	1		1	

S

x \ yz	00	01	11	10
0			1	
1		1	1	1

C



7/9/18

Code Conversion:

Input BCD				O/P excess -3 code			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0

0101 - 5
0110 - 6
0111 - 7
1000 - 8
1001 - 9

W

AB \ CD	00	01	11	10
00				
01		1	1	1
11	12 X	13 x	15 x	14 x
10	1	1	x 11	x 10

Values after 9 are not defined. They are called don't care and denoted by x.

$W = A + BD + BC$

X

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

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

Y

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

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

Z

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

$Z = CD + C\bar{D}$

Exclusive OR:

Error Detection:

Parity bit ensures that parity is same.

Single Error Detection

Message	Parity bit
1111	1
1011	

Exclusive OR:

$x \oplus 0 = x$ $\quad$ $x \oplus 1 = \overline{x}$

$x \oplus x = 0$ $\quad$ $x \oplus \overline{x} = 1$

$x \oplus \overline{y} = (x \oplus y)'$ $\quad$ ~~x~~ $x' \oplus y = (x \oplus y)'$

Commutative and Associative:

$a \oplus b = b \oplus a$

$a \oplus (b \oplus c) = (a \oplus b) \oplus c$

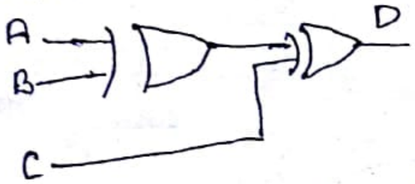
No need of Parenthesis.

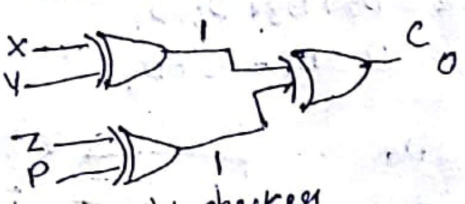
~~$A \oplus B \oplus C = A \oplus B'C \oplus BC'$~~

$A \oplus B \oplus C = AB'C' + A'BC' + ABC + A'B'C$

$\quad\quad\quad\quad\quad\quad 100 \quad\; 010 \quad \overset{111}{\cancel{000}} \quad 001$

Its value is 1 when any of the 4 is 1.





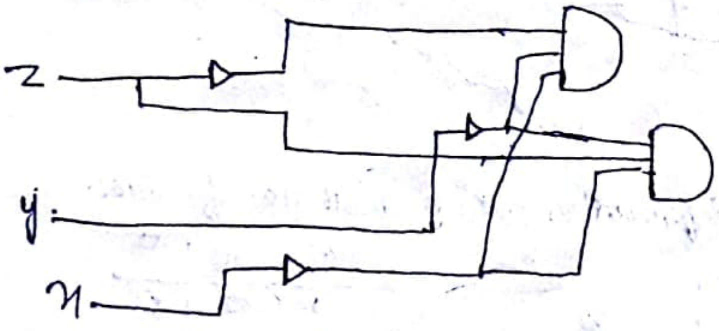
Parity checker

14/9/18

Decoders and Encoders:

Standard combinational circuits

A decoder is a combination at circuit that converts binary information from n input lines to a maximum of 2^n output lines.



$D_0 = x'y'z' \; (000)$

$D_1 = x'y'z \; (001)$

$D_2 = (x'yz') \; (010)$

⋮

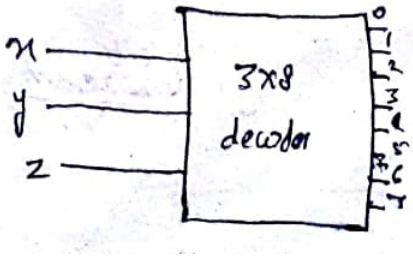
D_7

Full adder with a decoder and two OR Gates:

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

$S(x,y,z) = \Sigma(3,5,6,7)$

$x'y'z + x'yz' + xy'z' + xyz$



Encoder:

2^n input - n outputs

An encoder is a digital circuit that performs the inverse operation of a decoder.

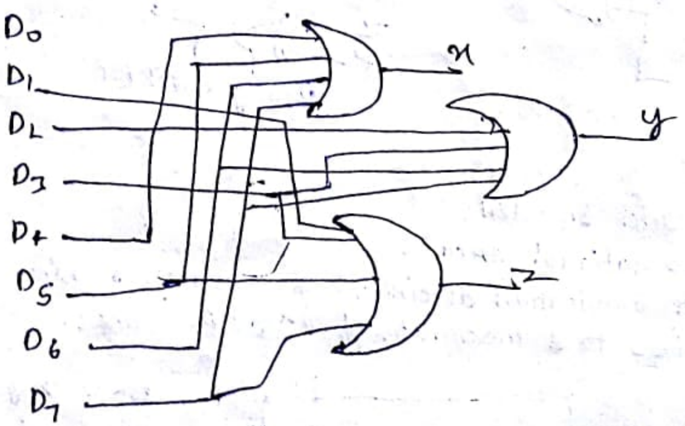
Truth table for octal to binary encoder:

Inputs								Encoder output		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
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

$xg = D_4 + D_5 + D_6 + D_7$

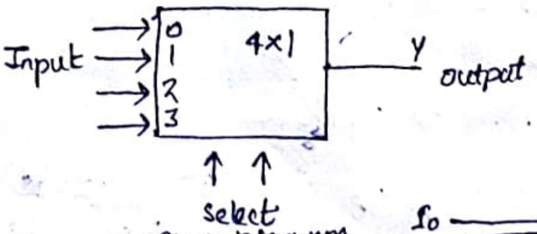
$y = D_2 + D_3 + D_6 + D_7$

$z = D_1 + D_3 + D_5 + D_7.$



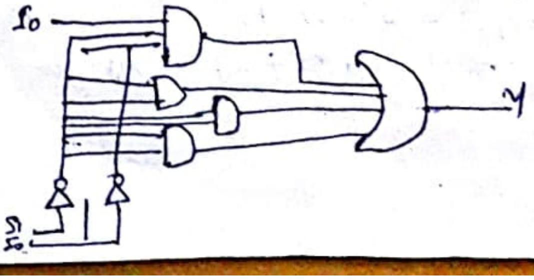
* Multiplexer:

Transmitting a large information over a small no. of lines.



Block Diagram

S1	S0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



5/10/18

Digital Circuits:

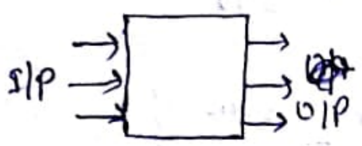
1. Combinational — o/P depends only on I/P

2. Sequential — o/P depends on the I/P as well as previous o/P

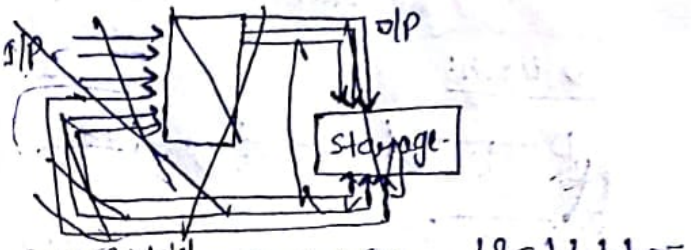
Importance of sequential circuits:

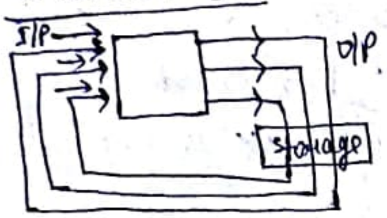
- For incrementing a no. by 1
- Traffic light controller

* Sequential circuits are circuits having a memory element.

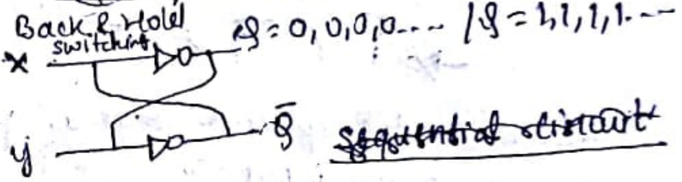


Combination ckt



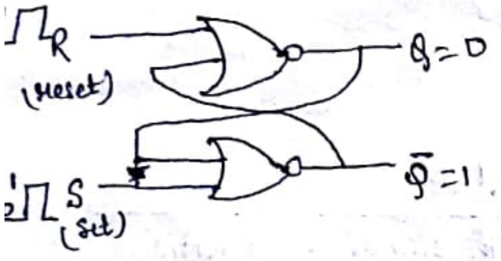


Sequential circuit



~~sequential circuit~~

SR/RS Latch:

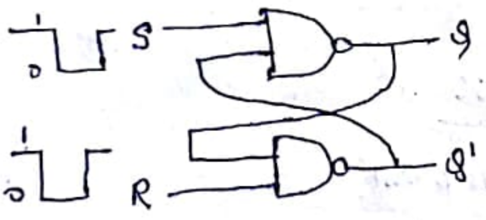


A	B	O/P
0	0	1
0	1	0
1	0	0
1	1	0

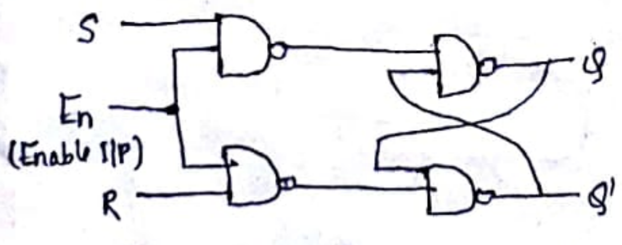
S	R	Q	Q̄
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0
0	0	1	0
		OR	
		0	1

O/P is unpredictable

SR Latch with the help of NAND Gates:



S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after S=1, R=0)
0	1	1	0	
1	1	1	0	(after S=0, R=1)
0	0	1	1	(forbidden)



SR Latch with control input

En	S	R	Next stage of Q
0	x	x	No change
1	0	0	No change
1	0	1	Q = 0 reset state
1	1	0	Q = 1 set state
1	1	1	Indeterminate

Enable input decides the enability of circuit. ~~If~~ it is 1, circuit will work, if it is close, circuit will not respond

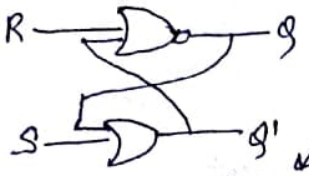
6/10/18

~~SR Latch~~

Two Realizations:

SR Latch

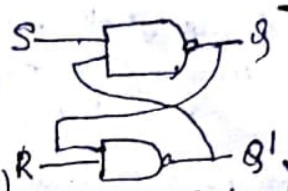
NOR Gate



S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0(forbidden)

Quiesent state

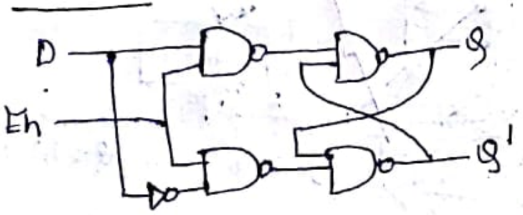
NAND Gate



S	R	Q	Q'
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1(forbidden)

Quiesent state

D Latch:



En	D	next state of Q
0	X	No change
1	0	Q = 0, reset state
1	1	Q = 1, set state

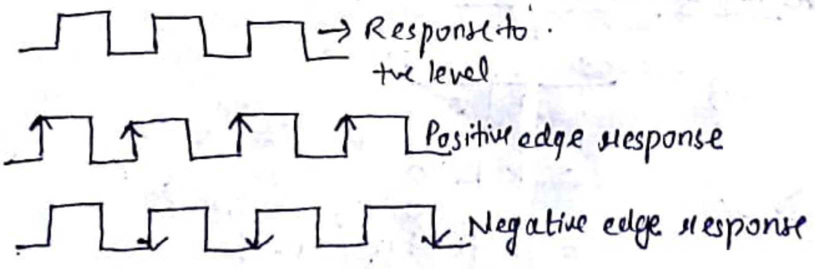
Also called transparent latch as gives the same o/p as I/P. To put on hold enable input = 0.

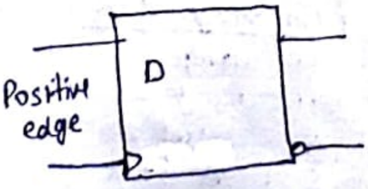
The undesirable O/P (11) is not obtained as a complimentary I/P is used.

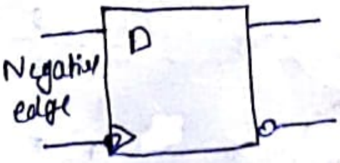
Difference b/w Latch and Flip flop:

Both are storage elements

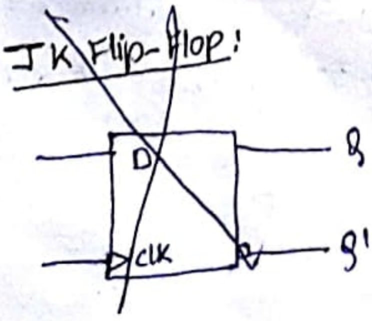
Latch	Flip-flops.
Operate with signal levels.	Operate with signal transitions



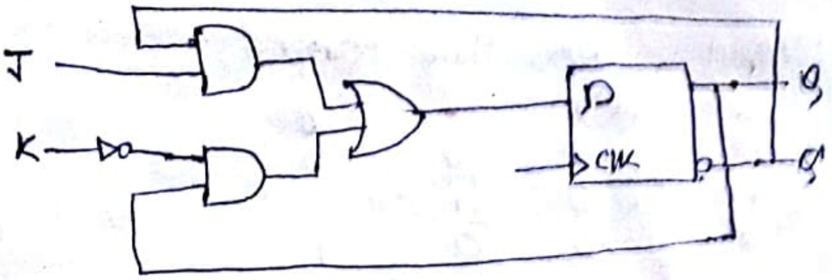




~~JK Flip-flop:~~



JK Flip-Flop:

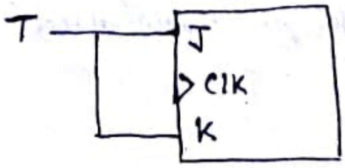


J	K	Q(t+1)
0	0	Q(t) no change
0	1	0 reset
1	0	1 set
1	1	Q'(t) complement

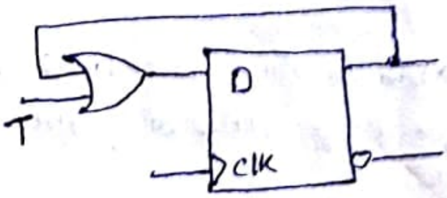
$$D = JQ' + K'Q$$

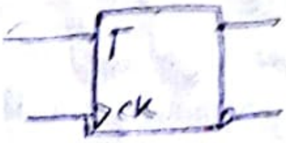
T Flip-Flop:

T=Toggle



from JK flip-flop.



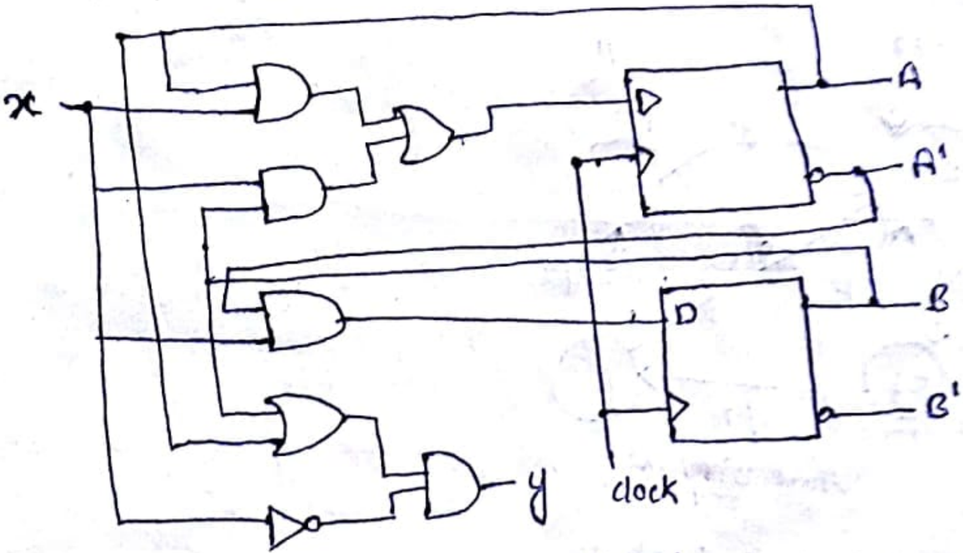


Symbol

T	Q(t+1)
0	Q(t) no change
1	Q'(t) complement

8/9/18

Analysis of Sequential Circuits:



$$A(t+1) = A(t)\,x(t) + B(t)\,x(t)$$
$$B(t+1) = A'(t)\,x(t)$$
$$y(t) = [A(t)+B(t)]\,x'(t)$$

⇐ State equations

$$\begin{cases} A(t+1) = Ax + Bx \\ B(t+1) = A'(x) \\ y = (A+B)x' \end{cases}$$

State Table:

Present state		Input	Next state		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1			
⋮					

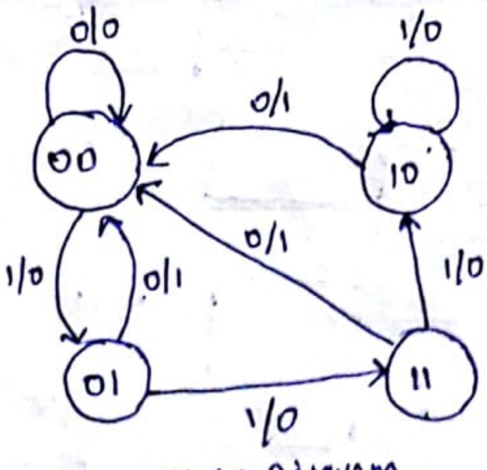
State Diagram:

The information available in the state table can be represented graphically in the form of state diagram.

Second Form of State Table:

Present state		Next state				Output	
		x = 0		x = 1		x = 0	x = 1
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

This form makes it easy to draw the pictorial representation of the state table.



State Diagram

The four circles represent 4 states.

$\text{i/p} \leftarrow 0/0 \rightarrow \text{o/p}$

Steps to analyse sequential ckts

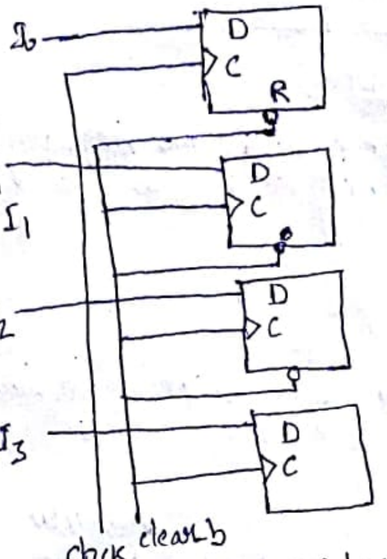
ckt diagram → equations → state table → state diagram

26/10/2018

Registers and Counters:

Register is a group of flip-flops, each of which share a common clock and is capable of storing 1 bit of information. An n bit register has n flip-flops.

Counter is basically a register that goes through a predetermined sequence of states.



4 bit register

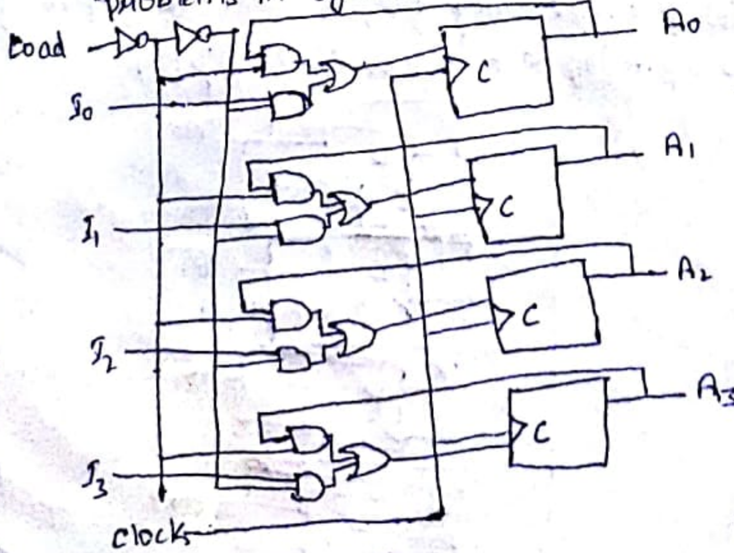
The common clock input triggers all flip-flops at the positive edge and the binary data available at the four inputs in transferred into the register.

Clear_b - input is useful for clearing the register to all 0's before the input is present.

Register will Parallel Load:

i) All the inputs should appear at the output simultaneously.

ii) Using clock input for creating the register with parallel load creates problems in synchronization.



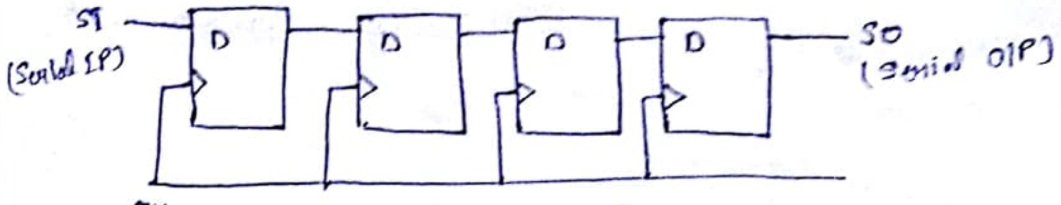
Load - 1
Load - 0

When the load input is 1, the data in the four inputs are transferred into the register with next positive edge of the clock.

When the load input is 0, the outputs of the flip-flops are connected to their respective inputs. The feedback connection from output is input is necessary because D flip-flops do not have a no change condition.

Shift Register:

A register capable of shifting its binary information is called a shift register.



4 bit shift register

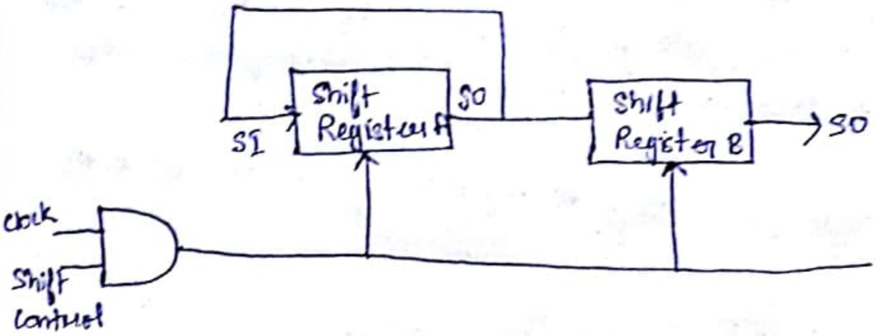
Each clock pulse shifts the contents of the register 1 bit position right. The serial input determines what goes into the leftmost flip-flop during the shift.

Serial Transfer:

A digital system is said to operate in serial mode when information is transferred and manipulated one bit at a time.

Parallel Transfer:

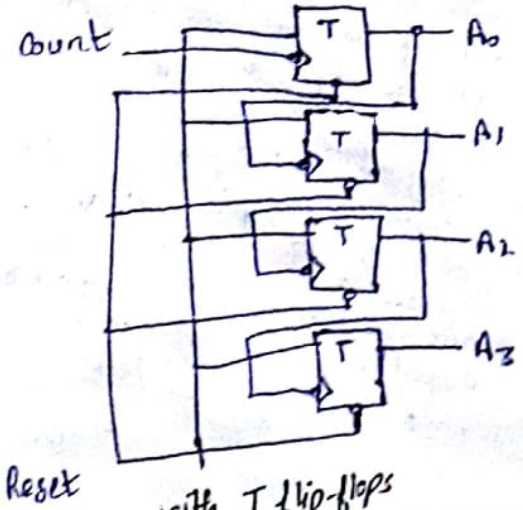
All the bits of the register are transferred, at the same time.



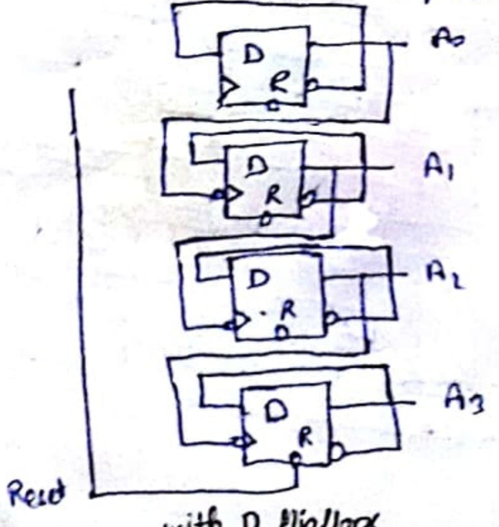
24/10/18

Binary Ripple Counter:

A binary ripple counter is a series of complementing flip-flops with the output of each flip-flop connected to the C input of next higher cycles flip-flop.



with T flip-flops



with D flipflops

11/9/18

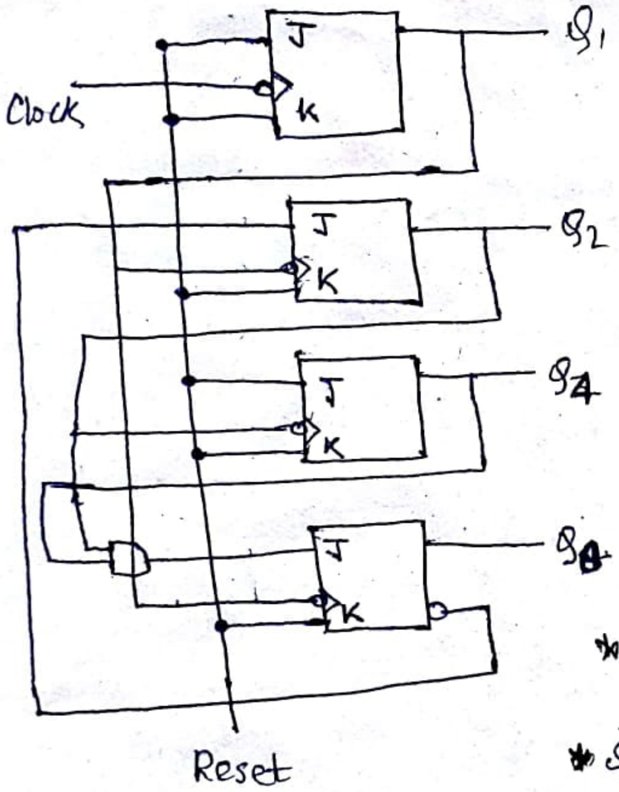
Queue:

It is a linear data structure. Follows FIFO. Insertion opposite ends.

Deletion:

① Both F and R are -1

Insertion:



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

- Q_1 changes state after each clock pulse.
- Q_2 complements every time Q_1 goes from 1 to 0, as long as $Q_8 = 0$.
- When Q_8 becomes 1, Q_2 remains at 0.
- Q_4 complements every time Q_2 goes from 1 to 0.
- Q_8 remains at 0 as long as Q_2 or Q_4 is 0.