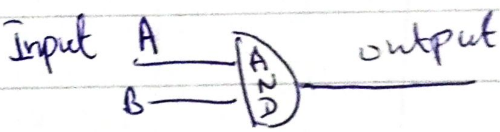
DLD #01

AND Gate:



Truth table: 2^2 => base = 2-binary digits - 1, 0
power = number of inputs
=> $2^2 = 4$ => number of entries in table

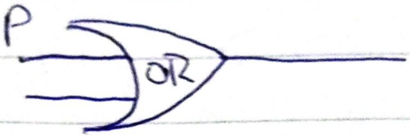
	A	B	O/P
1.	0	0	0
2.	0	1	0
3.	1	0	0
4.	1	1	1

table always starts from 0

$2^2 = 4/2 = 2$

$\downarrow$

$2/2 = 1$

OR Gates I/P  O/P

for example there are 4 ~~examples~~ inputs
then, $2^4 = 16 \Rightarrow$ entries

$16/2 = 8 \Rightarrow 8$ (0s and 1s)

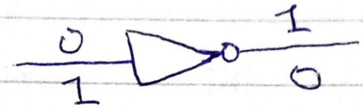
$8/2 = 4 \rightarrow 4/2 = 2$

$2/2 = 1$

	A	B	C	D	O/P
1.	0	0	0	0	0
2.	0	0	0	1	1
3.	0	0	1	0	1
4.	0	0	1	1	1
5.	0	1	0	0	1
6.	0	1	0	1	1
7.	0	1	1	0	1
8.	0	1	1	1	1
9.	1	0	0	0	1
10.	1	0	0	1	1
11.	1	0	1	0	1
12.	1	0	1	1	1
13.	1	1	0	0	1
14.	1	1	0	1	1
15.	1	1	1	0	1
16.	1	1	1	1	1

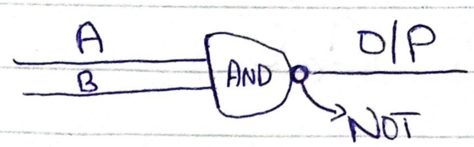
LAB #02

- Not Gate:



Universal gates: NAND & NOR

- NAND Gate:

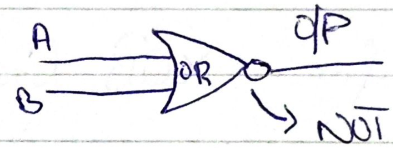


A	B	AND	O/P
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

∴ Reverse of AND

$A'B' + A'B + AB'$

- NOR Gate:



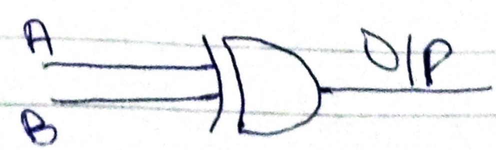
A	B	OR	O/P
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

∴ Reverse of OR

- XOR Gate: (Exclusive OR Gate) = $A \oplus B$

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

∴ Output will be zero when two same bits



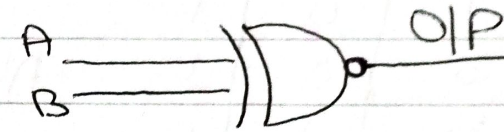
To make the expression from the truth table, we will consider only 1s output. Take compliment on the 0 input.

XNOR Gate :

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

row 1 row 4

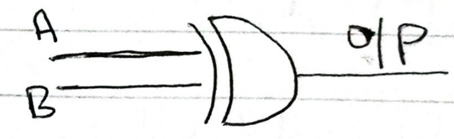
$A'B' + AB$



Expression for XOR Gate :

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

$f = A'B + B'A$



for Multiplication : AND Gate/IC

for Addition : OR Gate/IC

DLD LAB #04

⇒ Random Custom Truth table:

A	B	(O/P) → your choice
0	0	1 ✓
0	1	0
1	0	1 ✓
1	1	1 ✓

$$\Rightarrow \boxed{F = A'B' + A'B + AB}$$

Inputs: $2^n \Rightarrow 2^2 \Rightarrow 4$ entries in K-map

K-Map:

A \ B	0	1
0	00	01
1	10	11

⇒ Now implement the equation into K-map to reduce it.

$$\Rightarrow F = \underset{(00)}{A'B'} + \underset{(01)}{A'B} + \underset{(11)}{AB}$$

A \ B	0	1
0	1	1
1		1

Now, next step is pairing from K-map.

A \ B	0	1
0	1	1
1		1

pair overlapping is allowed.

Now, look at which variable is constant in a pair, not in the

Block, but outside the block, that are row and column.

In the first pair where the row of A is constant, that is 0.

So, $F = A'$ and in the second pair, the column (B) is constant, 1

So, $\boxed{F = A' + B}$ well done! equation is reduced.

It is give the same answer as that of the original equation.

=> Random Equation of your choice:

=> 3 inputs => $F = A'BC' + AB'C' + ABC + A'B'C' + A'B'C$

Now, K-mapping: 2^n => n = number of inputs

2^3 => 8 entries/blocks in K-map.

A \ BC	00	01	11	10
0	000	001	011	010
1	100	101	111	110

← Gray Code & LSB will change, one bit at a time.

421

Usual sequence:

00 → 0
01 → 1
10 → 2
11 → 3

=> But gray don't follow this, it only changes one bit at a time at LSB (Least significant Bit)

Now, map the equation:

$$\Rightarrow F = \overbrace{A'BC'}^{(010)} + \overbrace{AB'C'}^{(100)} + \overbrace{ABC}^{(111)} + \overbrace{A'B'C'}^{(000)} + \underbrace{A'B'C}_{(001)}$$

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

$$\Rightarrow \boxed{F = B'C' + A'B' + ABC + A'BC'}$$

=> Write the single/non-pair elements as it is.

TASK #01

$F = A'B'C' + A'B'C + A'BC' + AB'C + ABC'$
$+ ABC + AB'C'$

$\therefore 2^n = 2^3 = 8$ entries

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

$\Rightarrow F = \underset{(000)}{\underline{A'B'C'}} + \underset{(001)}{\underline{A'B'C}} + \underset{(010)}{\underline{A'BC'}} + \underset{(101)}{\underline{AB'C}}$
$+ \underset{(110)}{\underline{ABC'}} + \underset{(111)}{\underline{ABC}} + \underset{(100)}{\underline{AB'C'}}$

K-map : $2^n \Rightarrow 2^3 = 8$ blocks in K-map

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

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

$\Rightarrow f = B' + AC + BC'$

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

$\boxed{f = B' + A + C'}$ ←

Task # 02

Truthtable → Random :

A	B	C	O/P
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1 ✓
1	0	0	1 ✓
1	0	1	1 ✓
1	1	0	1 ✓
1	1	1	1 ✓

Equation : $f = \underset{(011)}{A'BC} + \underset{(100)}{AB'C'} + \underset{(101)}{AB'C} + \underset{(110)}{ABC'} + \underset{(111)}{ABC}$

$\therefore 2^n = 2^3 = 8$ blocks

K-map :

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

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

Pairing : $\boxed{f = A + BC}$

DLD LAB #05

Adders :

=> Half adders :

A	B	Sum	Carry
0	0	0	0
0	1	1 ✓	0
1	0	1 ✓	0
1	1	0	1 ✓

Equations of :

=> $Sum = \underbrace{A'B + AB'}$

∴ XOR = $A \oplus B$

=> $Carry = AB$

=> Full Bit Adder :

A	B	Cin	Sum	Carry
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

=> $Sum = A'B'Cin + A'BCin' + AB'Cin' + ABCin$

$$\Rightarrow Sum = A'(\overbrace{\underbrace{B'Cin + BCin'}_{B}}^{XOR}) + A(\overbrace{\underbrace{B'Cin' + BCin}_{B'}}^{XNOR})$$

$$\Rightarrow B = XOR = B'Cin + BCin' = B \oplus Cin$$

$$\Rightarrow B' = XNOR = B'Cin' + BCin = \overline{B \oplus Cin}$$

$$\Rightarrow Sum = A'B + AB'$$
$$= A \oplus B.$$
$$= A \oplus B \oplus Cin$$

$$\Rightarrow Carry = A'BC + AB'C + ABC' + ABC$$
$$= C(\underbrace{A'B + AB'}_{XOR = A \oplus B}) + AB(\underbrace{C' + C}_{0+1=1})$$
$$= C(A \oplus B) + AB(1)$$
$$= C(A \oplus B) + AB$$

911. Osama Bin Laden.

DLD LAB # 06

"Subtractors"

Half subtractor:

A	B	diff	Borrow
0	0	0	0
0	1	1 ✓	1 ✓
1	0	1 ✓	0
1	1	0	0

=> Difference $= A'B + B'A$

$= A \oplus B$

=> $\boxed{\text{Borrow} = A'B}$

Full Subtractor:

A	B	C	difference	Borrow
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

$$\text{Diff} = A'B'C + A'BC' + AB'C' + ABC$$

$$= A'(\underbrace{B'C + BC'}_{XOR}) + A(\underbrace{B'C' + BC}_{XNOR})$$

$XOR = B'C + BC' = B \oplus C$
$XNOR = B'C' + BC = \overline{B \oplus C}$

let $B \oplus C = X$, $\overline{B \oplus C} = X'$

$$diff = A'X + AX' \quad \therefore A \oplus X = XOR$$
$$= A \oplus X \quad \therefore X = B \oplus C$$
$$= A \oplus B \oplus C$$

$$Borrow = A'B'C + A'BC' + A'BC + ABC$$
$$= A'(\underbrace{B'C + BC'}_{XOR}) + BC(\underbrace{A' + A}_{0+1=1})$$

$$= A'(B \oplus C) + BC$$

LAB # 07

Comparatives: → 1-Bit

Input		output		
A	B	$Y_1 = (A<B)$	$Y_2 (A=B$	$Y_3 (A>B)$
0	0	0	1 ✓	0
0	1	1 ✓	0	0
1	0	0	0	1 ✓
1	1	0	1 ✓	0

LAB # 07

Equations: $Y_1 = A'B$, $Y_2 = A'B' + AB$
$Y_3 = AB'$ $\quad = \overline{A \oplus B}$

2-Bit:

2-bit. $A = A_1, A_0$, $B = B_1, B_0$

A_1	A_0	B_1	B_0	$Y_1(A<B)$	$Y_2(A=B)$	$Y_3(A>B)$
0	0	0	0	0	1 ✓	0
0	0	0	1	1 ✓	0	0
0	0	1	0	1 ✓	0	0
0	0	1	1	1 ✓	0	0
0	1	0	0	0	0	1 ✓
0	1	0	1	0	1 ✓	0
0	1	1	0	1 ✓	0	0
0	1	1	1	1 ✓	0	0
1	0	0	0	0	0	1 ✓
1	0	0	1	0	0	1 ✓
1	0	1	0	0	1 ✓	0
1	0	1	1	1 ✓	0	0
1	1	0	0	0	0	1 ✓
1	1	0	1	0	0	1 ✓
1	1	1	0	0	0	1 ✓
1	1	1	1	0	1 ✓	0

- 2-Bit: $A = 2^1 < B = 2^1$
$\quad\quad 0 \; 1 < 1 \; 0$

Equation : $Y_1 = A_1'A_0'B_1'B_0 + A_1'A_0'B_1B_0' +$
$A_1'A_0'B_1B_0 + A_1'A_0B_1B_0' +$
$A_1'A_0B_1B_0 + A_1A_0'B_1B_0$

$\Rightarrow Y_2 = A_1'A_0'B_1'B_0' + A_1'A_0B_1'B_0 + A_1A_0'B_1B_0'$
$+ A_1A_0B_1B_0$

$\Rightarrow Y_3 = A_1'A_0B_1'B_0' + A_1A_0'B_1'B_0' + A_1A_0'B_0'B_1 +$
$A_1A_0B_1'B_0' + A_1A_0B_1'B_0 + A_1A_0B_1B_0'$

K-mapping :

$Y_1 =$

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

$Y_1 = A_1'B_1 + A_1'A_0'B_0 + A_0'B_1B_0$

$Y_2 =$

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	1			
01		1		
11			1	
10				1

$\Rightarrow$ Not possible
no pairs

$Y_3 =$

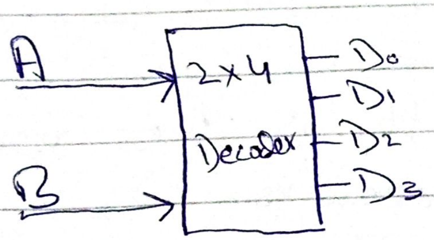
$A_1A_0 \backslash B_1B_0$	00	01	11	10
00				
01	1			
11	1	1		1
10	1	1		

$$Y_3 = A_1B_1' + A_0B_1'B_0' + A_1A_0B_0'$$

LAB # 08

- 74LS138 = Decoder
- 74LS148 = Encoder

Decoders 2 to 4 lines:

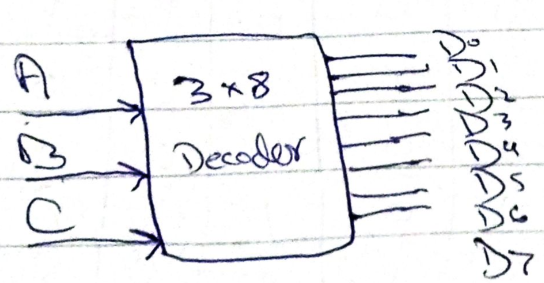


inputs		outputs			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

2 to 4 $= 2^n = 2^2 = 4$
inputs outputs

$D_0 = A'B'$ $D_1 = A'B$
$D_2 = AB'$ $D_3 = AB$

Decoders 3 to 8 lines:



inputs			Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
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✓

$D_0 = A'B'C'$ $\quad D_1 = A'B'C$ $\quad D_2 = A'BC'$

$D_3 = A'BC$ $\quad D_4 = AB'C'$ $\quad D_5 = AB'C$

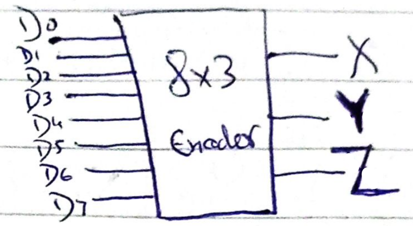
$D_6 = ABC'$ $\quad D_7 = ABC$

Encoders 8 to 3 & It is the reverse of decoders.

∴ 74LS148 = IC

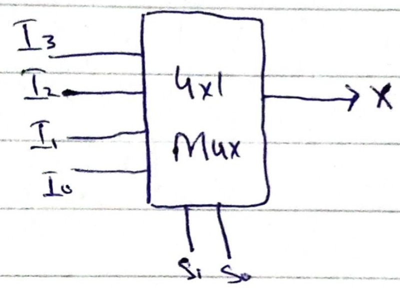
inputs								output		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	X	Y	Z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$X = D_4 + D_5 + D_6 + D_7$
$Y = D_2 + D_3 + D_6 + D_7$
$Z = D_1 + D_3 + D_5 + D_7$



LAB # 09

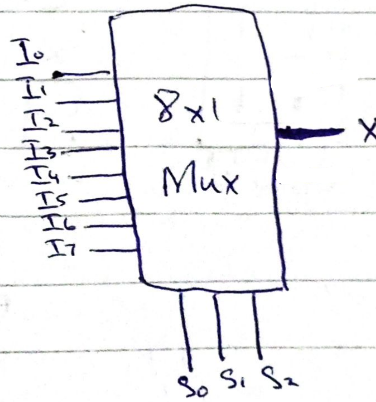
⇒ 4 x 1 Mux :



S_1	S_0	X
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

∴ 4 inputs & 1 output
∴ S_1 & S_0 = selection pins
∴ 74LS153 (mux) 4 inputs

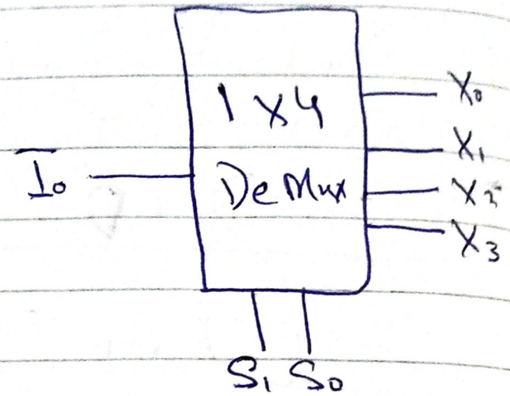
⇒ 8 x 1 Mux :



C	B	A	
S_2	S_1	S_0	X
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

∴ 74LS151 (Mux)
8 input pins
∴ On trainer, check first four pins, and then other four

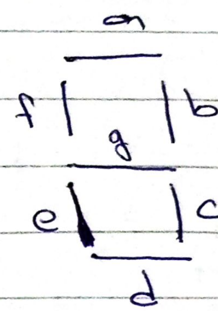
De Multiplex : 1 x 4

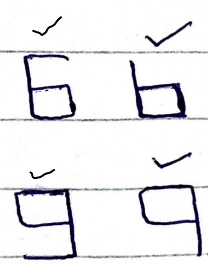


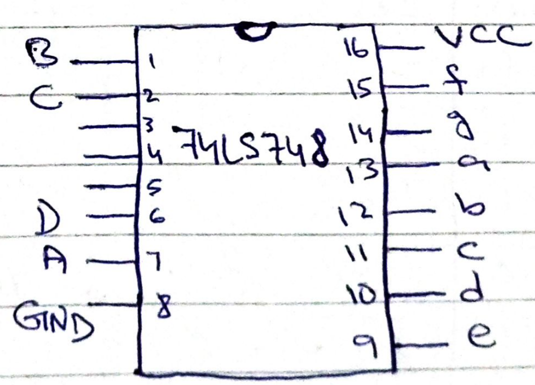
S_1	S_0	X_3	X_2	X_1	X_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

LAB # 10

7 segment Display :







	Inputs				Output						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

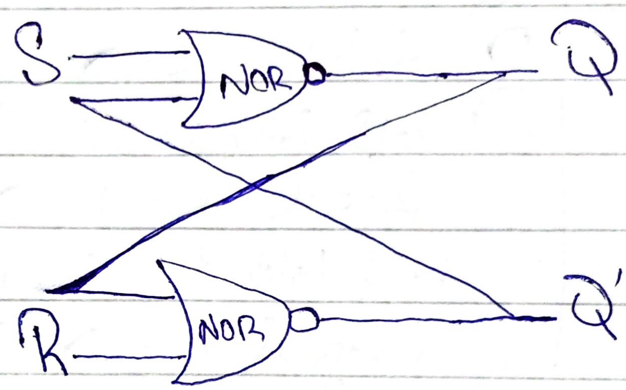
LAB # 11

Latches: S.R. latch using NOR

NOR

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

Sequential vs combinational



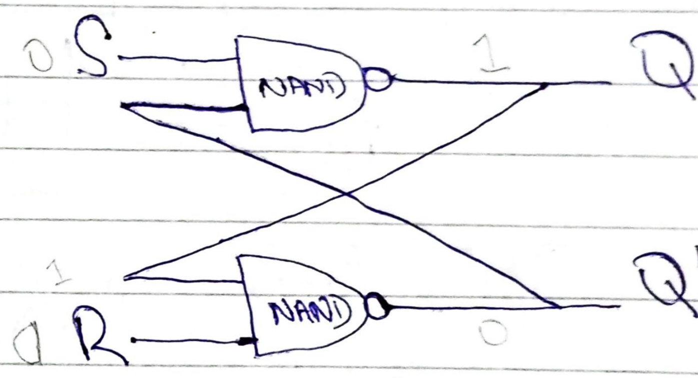
→ Wrong Truth Table

Inputs S	Inputs R	Outputs Q	Outputs Q'	
0	1	1	0	→ Set State
0	0	1	0	→ Hold latched
1	0	0	1	→ Reset State
0	0	0	1	→ Hold latched
1	1	0	0	→ undefined

⇒ S.R. latch using NAND

NAND

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



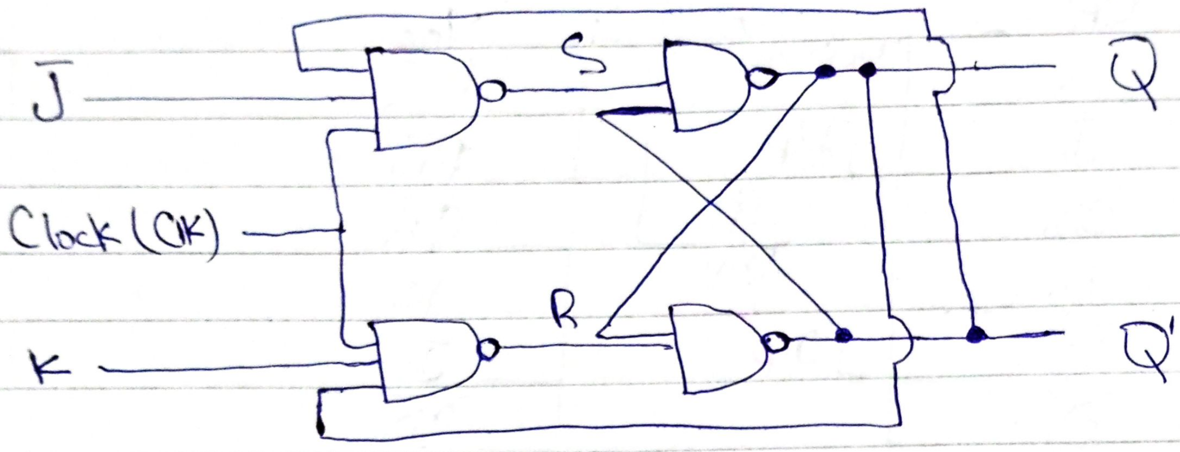
Inputs S	Inputs R	Output Q	Output Q'
0	1	1	0
0	0	1	0
1	0		
0	0		
1	1		

Correct Truth Table

	Inputs S	Inputs R	Output Q	Output Q'
Set ←	0	1	1	0
hold latch ←	1	1	1	0
Reset ←	1	0	0	1
hold latch ←	1	1	0	1
undefined ←	0	0	1	1

LAB # 12

JK Flip Flops : (Sequential)



CLK	J	K	S	R	Q	Q'
0	X	X	1	1	Memory	State
1	1	0	0	1	1	0
1	0	1	1	0	0	1
1	1	1	0	1	1	0
1	1	1	1	0	0	1

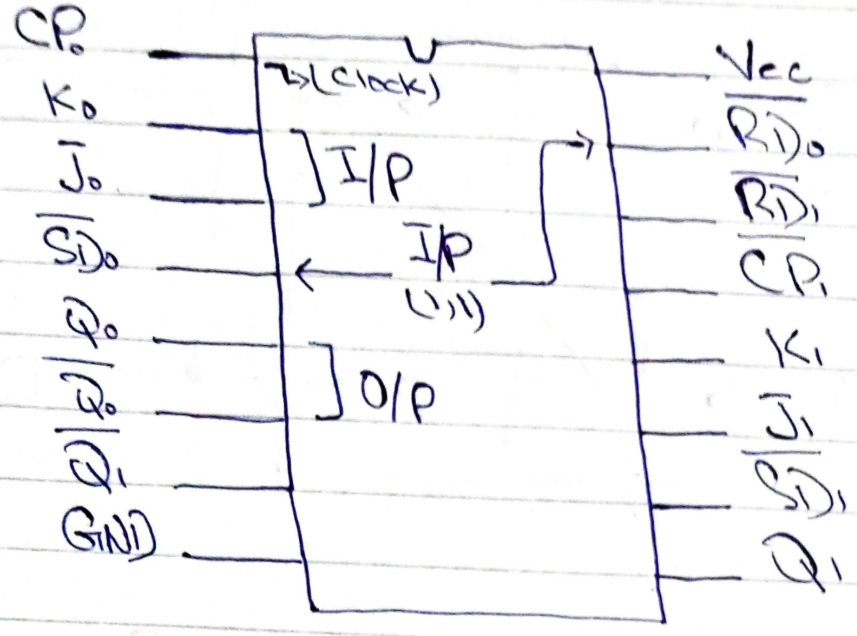
Toggle

NAND

0	0	1
0	1	1
1	0	1
1	1	0

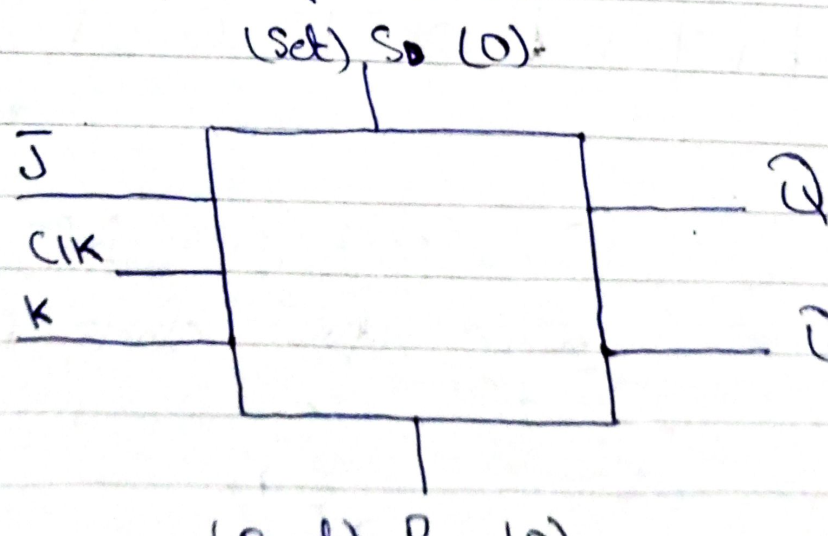
∴ Use Pulser switches on trainer for clock A' or B'

Multisim: 74LS112

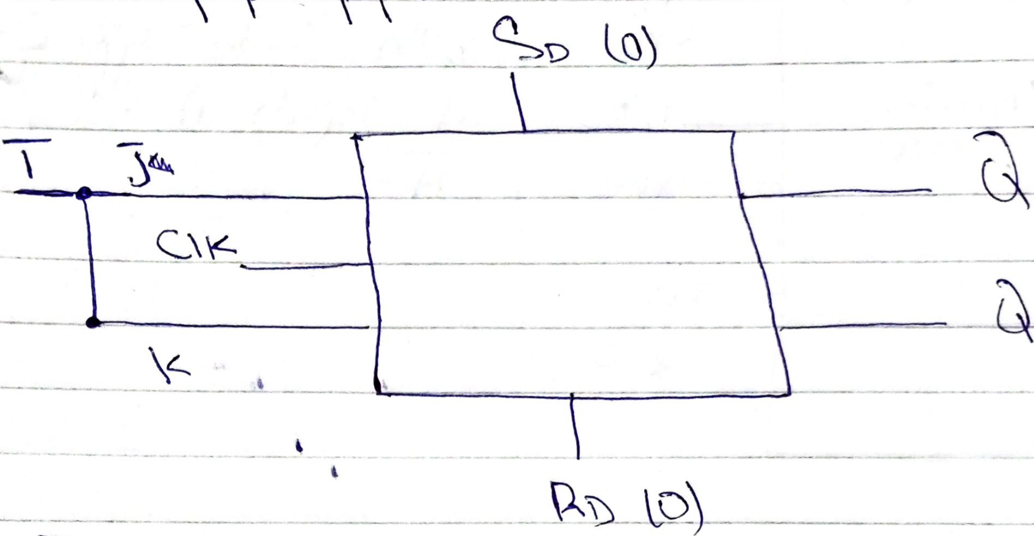


2- flip flops in one IC

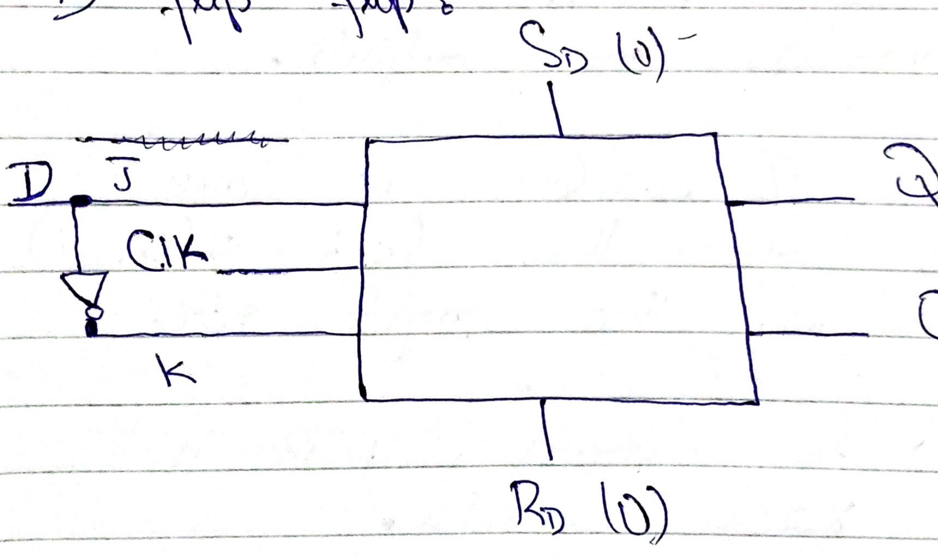
⇒ JK flip flop:



T flip flop:



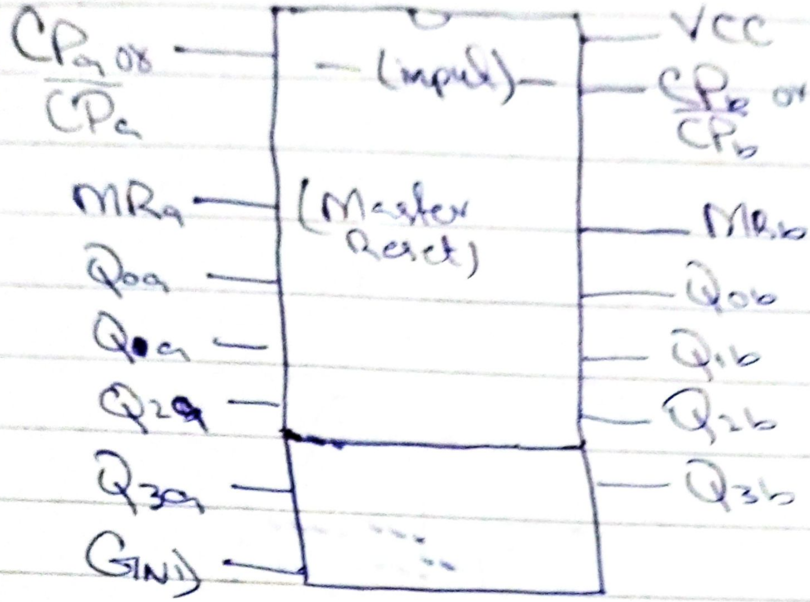
D flip flop:



LAB # 13

Counters: 74LS393

→ 4-bit counter
0-15 Decimal



∴ It will work with both type of inputs A and $\overline{A}$.

∴ CP is input ; MR is master reset ; Q0 – Q3 are outputs.

∴ There are 2- counters in one IC, one on the left side, the other on the right side.

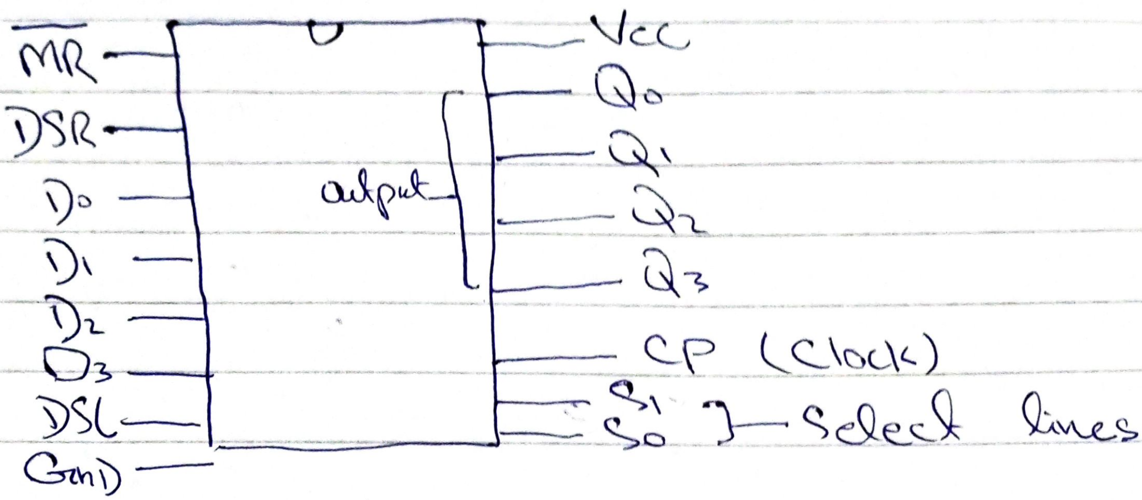
∴ CP is to count, MR is used to reset counter.

Shift Registers :

S_1	S_0	O/P - Modes
0	0	No change
0	1	Right shift – one bit at
1	0	Left shift – a time
1	1	Parallel – 4-bits altogether

74LS194

=>



- DSR = Shift Right Input
- DSL = Shift Left Input
- $D_0 - D_3$ = Parallel inputs

Give clocks for every input

=> Keep reset to HIGH so the bar will make it LOW.