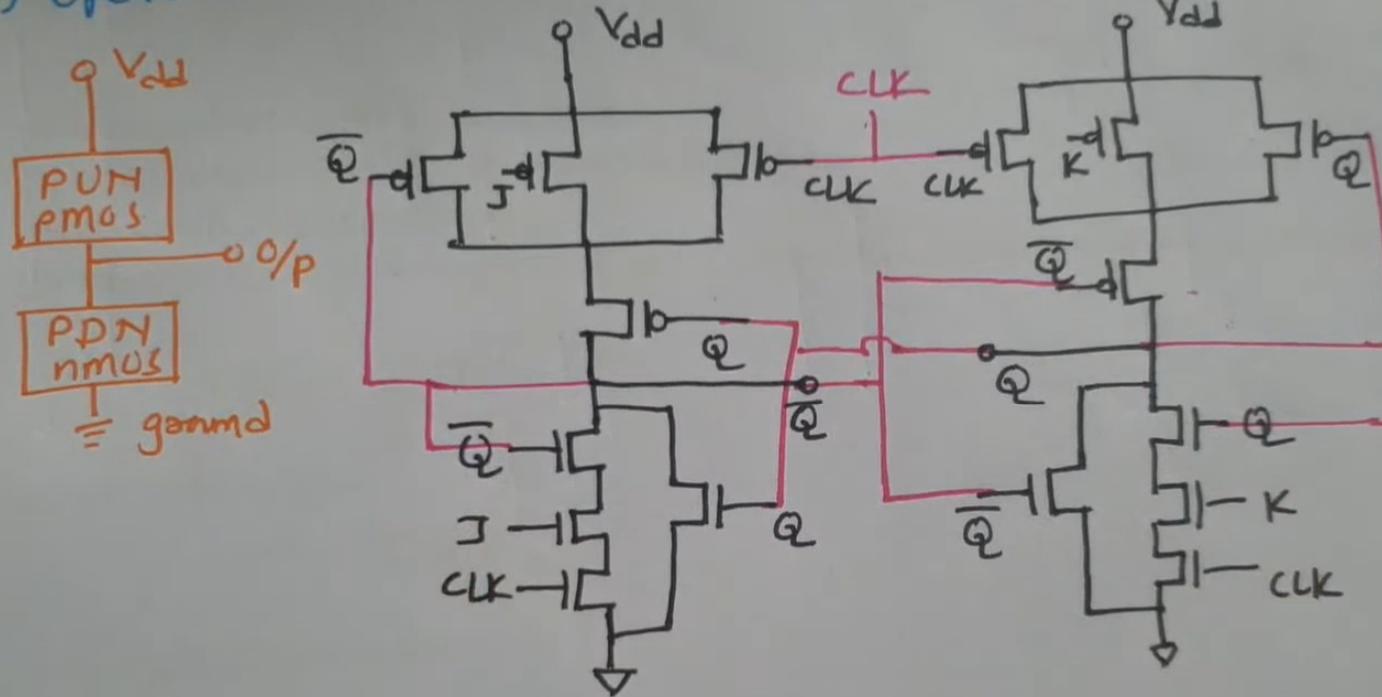


## CMOS JK Flip Flop Using NOR Gates: Circuit, Rules, Working, Implementation & Truth Table

*(+) Operation*  $\rightarrow$  pmos (parallel)  $\rightarrow$  nmos (series)  
*(-) Operation*  $\rightarrow$  nmos (series)  $\rightarrow$  pmos (parallel)



11:51 / 12:14

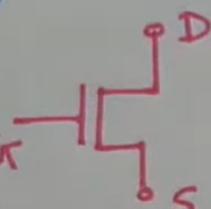
CMOS JK Flip Flop using NOR Gates implementation >

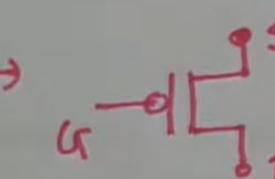


## CMOS Logic Circuit rules

- CMOS

 $\begin{cases} \text{nmos} \\ \text{pmos} \end{cases}$ 

nMOS  $\rightarrow$    $\rightarrow$  If  $G=0$ , OFF, (D to S as O.C.)  
 $\rightarrow$  If  $G=1$ , ON, (D to S as S.C.)

pMOS  $\rightarrow$    $\rightarrow$  If  $G=0$ , ON, (S to D as S.C.)  
 $\rightarrow$  If  $G=1$ , OFF, (S to D as O.C.)

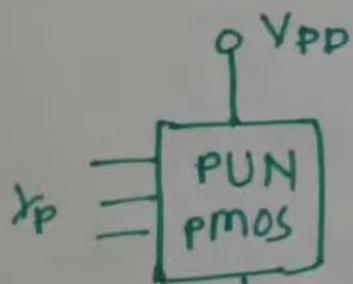
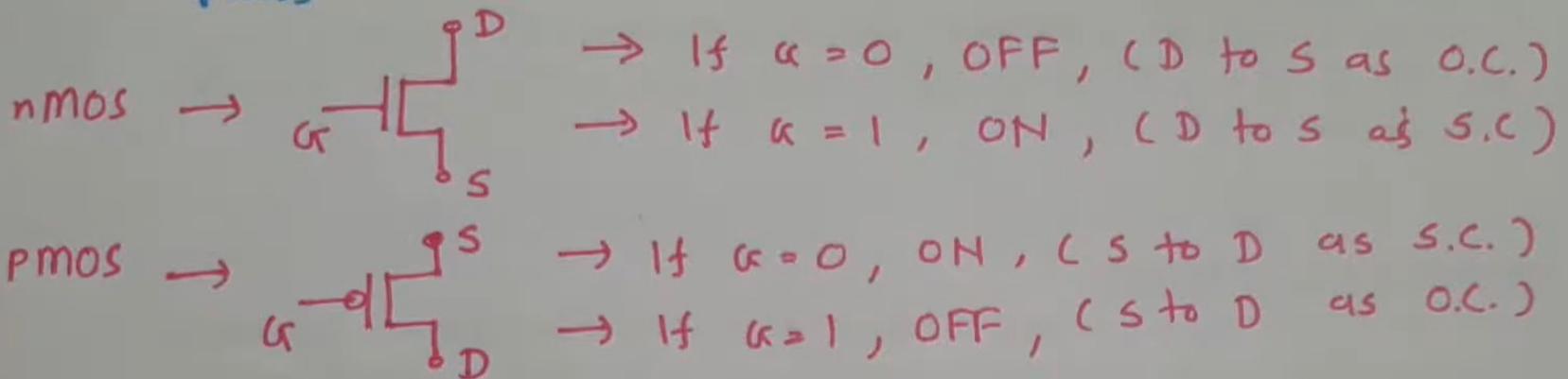


3:35 / 10:17

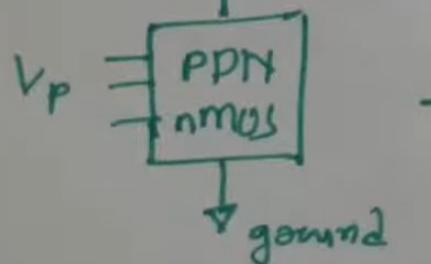
pMOS and nMOS working



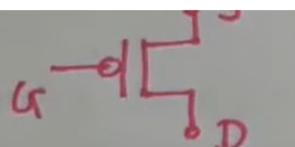
$L_{pmos}$

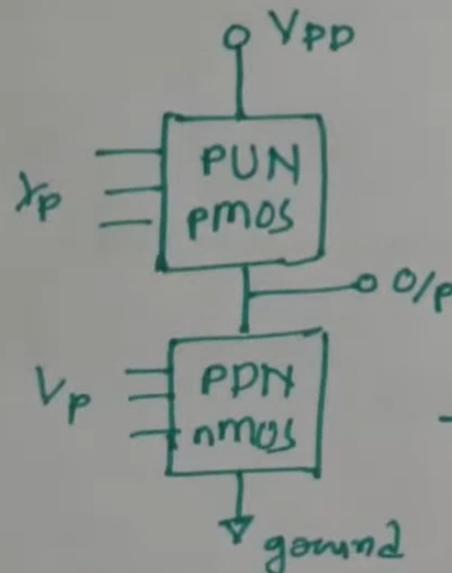


$\rightarrow$  PMOS Transistor can easily pass logic high.



$\rightarrow$  NMOS Transistor can easily pass logic low.

$\text{PMOS} \rightarrow$    $\rightarrow$  If  $S=0$ , ON, (S to D as S.C.)  
 $\rightarrow$  If  $S=1$ , OFF, (S to D as O.C.)



$\rightarrow$  pMOS Transistor can easily pass logic high.

$\rightarrow$  nMOS Transistor can easily pass logic low.

- For logic circuit  $(\cdot) \rightarrow \text{AND}$ ,  $(+) \rightarrow \text{OR}$ .

-  $(\cdot)$  operation -

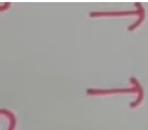
- pMOS  $\rightarrow$  1kΩ

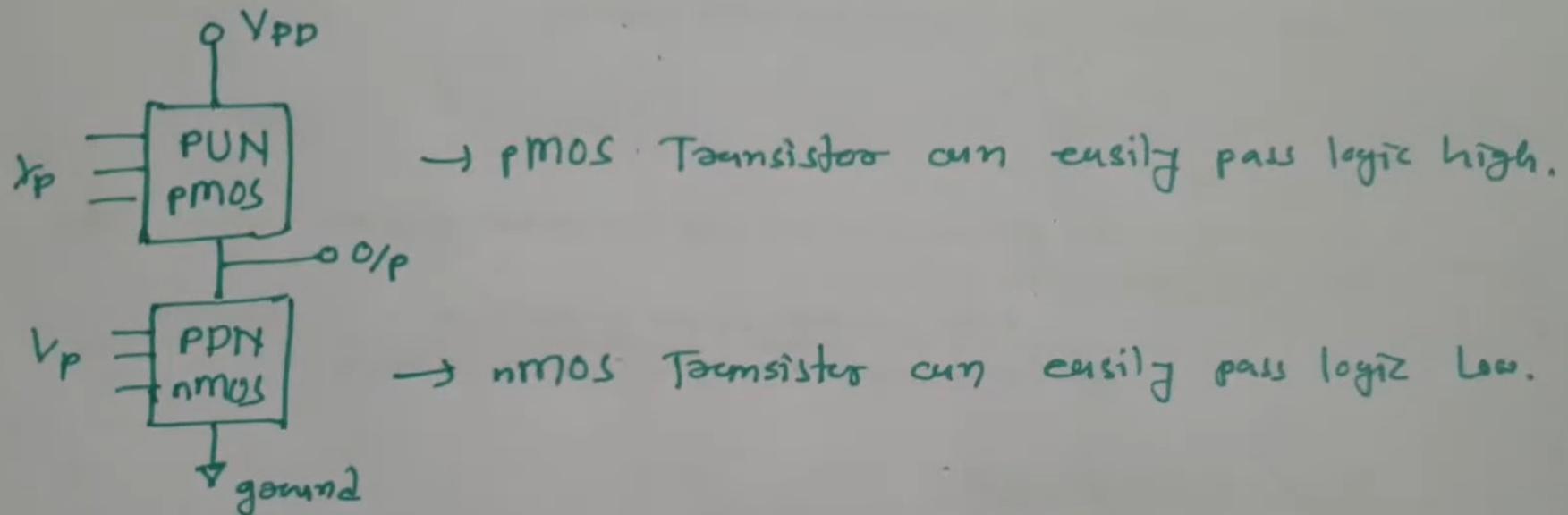
- nMOS  $\rightarrow$  1kΩ

-  $(+)$  operation

- pMOS  $\rightarrow$  Series

- nMOS  $\rightarrow$  Parallel

$\text{PMOS} \rightarrow$    $\rightarrow$  If  $G=0$ , ON, (S to D as S.C.)  
 $\rightarrow$  If  $G=1$ , OFF, (S to D as O.C.)



- For logic circuit  $(\cdot) \rightarrow \text{AND}$ ,  $(+) \rightarrow \text{OR}$ .

- $(\cdot)$  operation
- pmos  $\rightarrow$  parallel
- nmos  $\rightarrow$  series.

- $(+)$  operation
- pmos  $\rightarrow$  series
- nmos  $\rightarrow$  parallel

## Design of CMOS NAND gate

→ Boolean eqn.

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

→ Symbol



→ Truth Table

Inputs		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

→ (.) Operation

pMOS - parallel

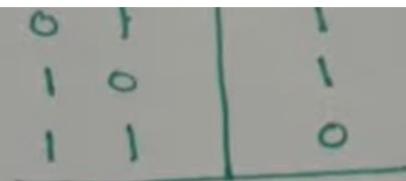
nMOS - series



2:49 / 12:22

CMOS Circuit Rules >

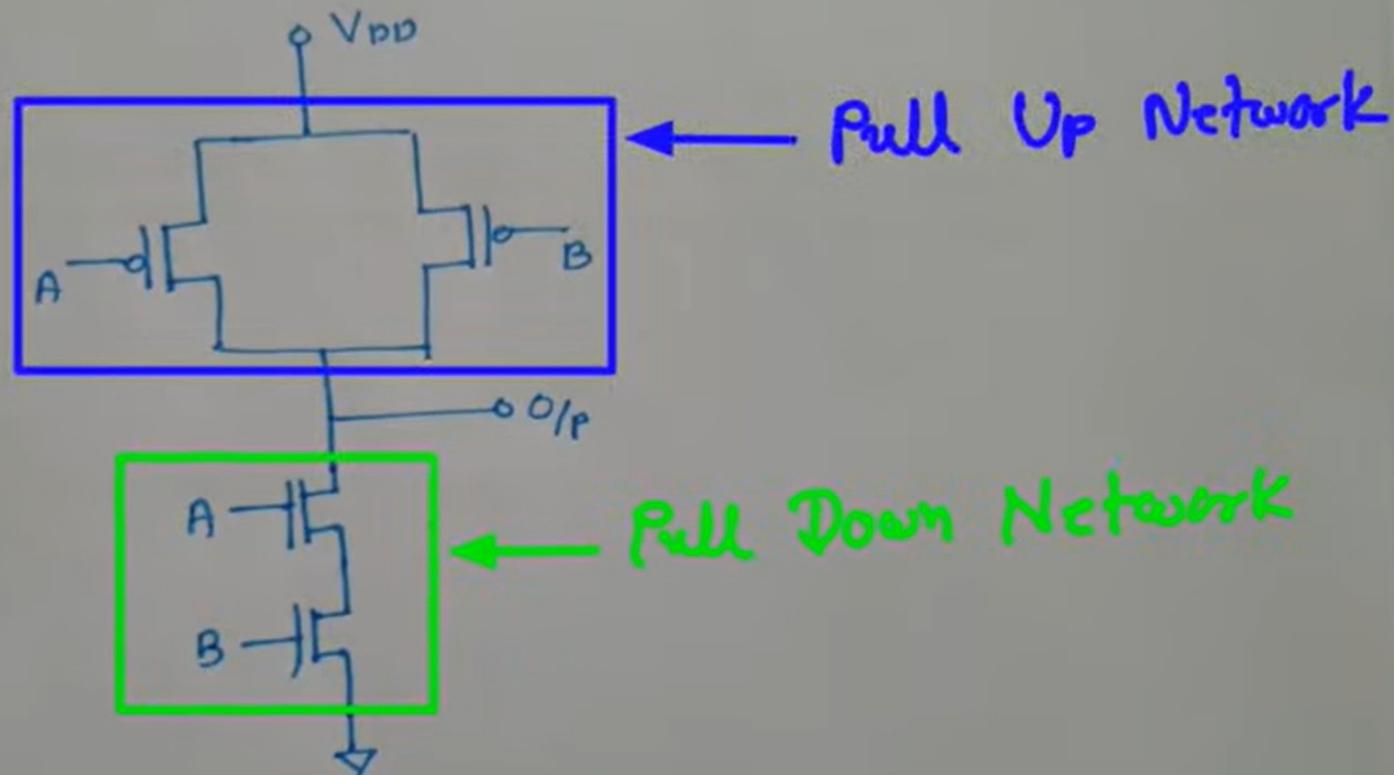
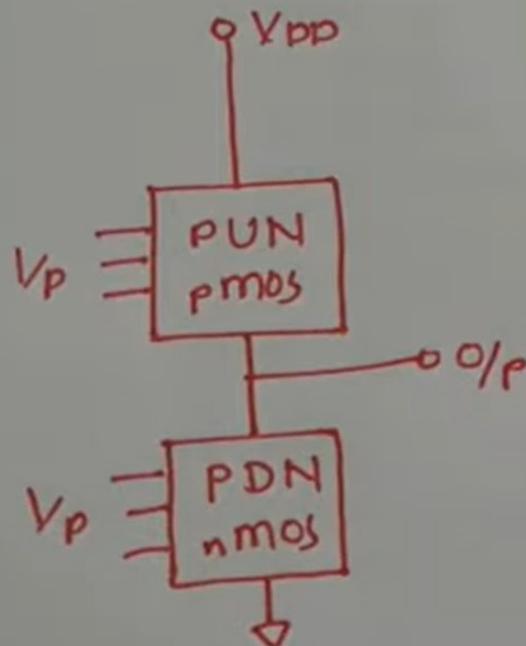




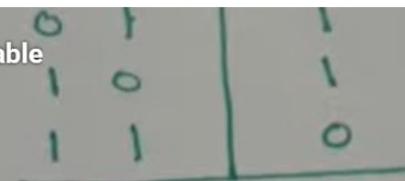
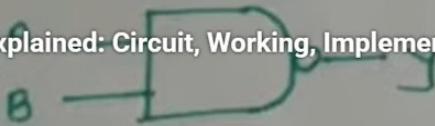
→ (.) Operation

pMOS - parallel

nMOS - series



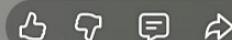
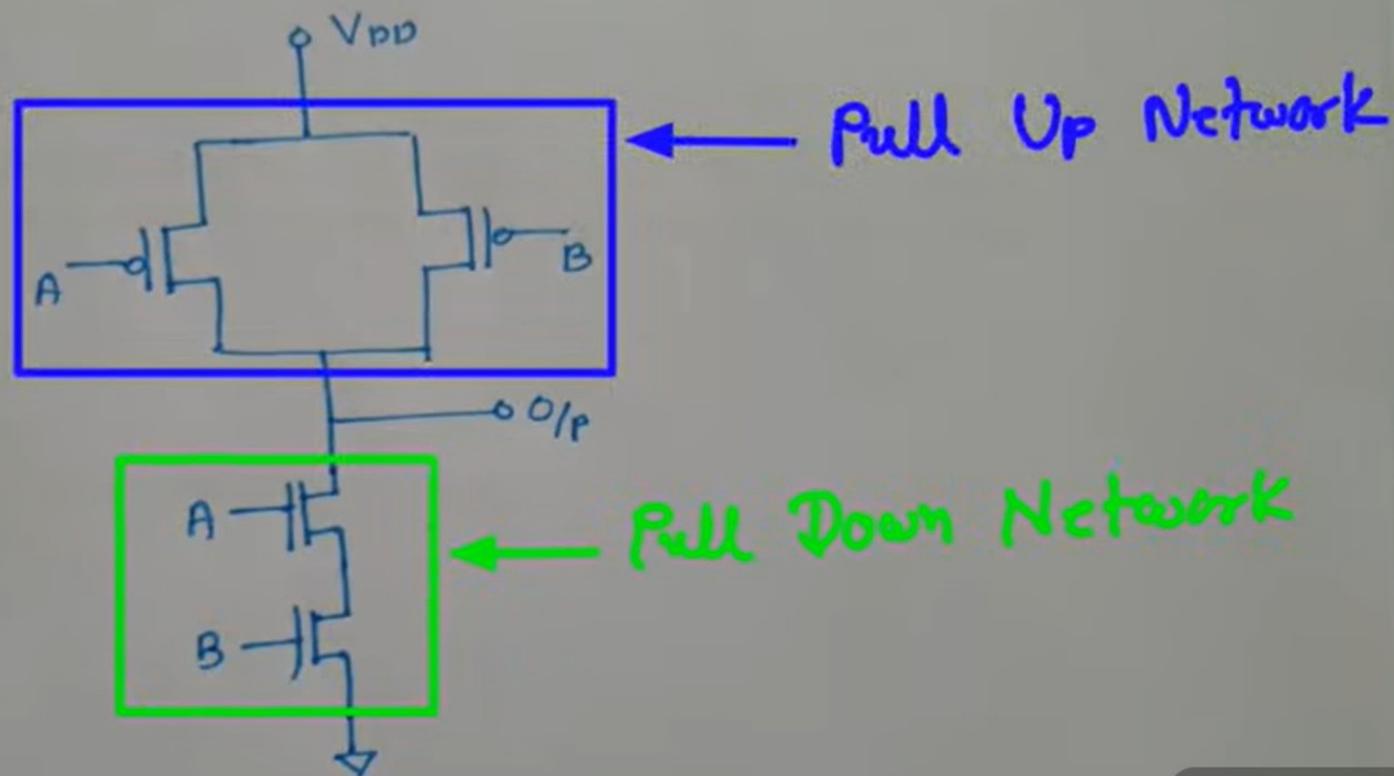
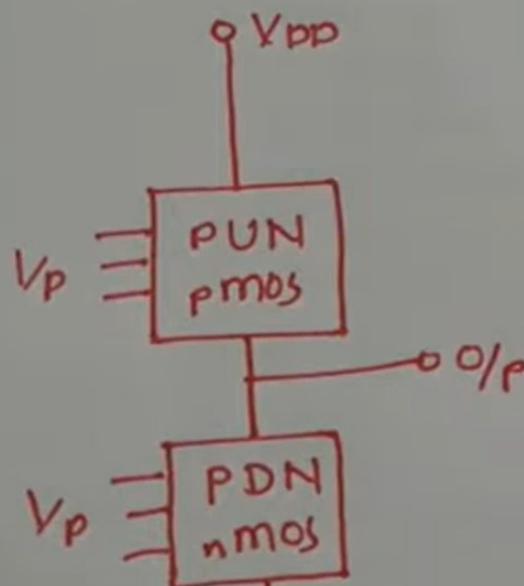
## CMOS NAND Gate Explained: Circuit, Working, Implementation, and Truth Table



→ (.) Operation

pMOS - parallel

nMOS - series



6:07 / 12:22

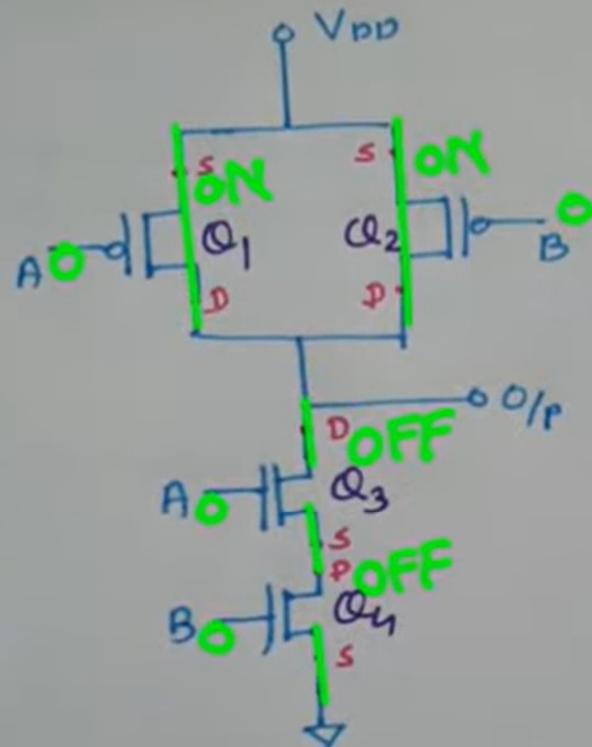
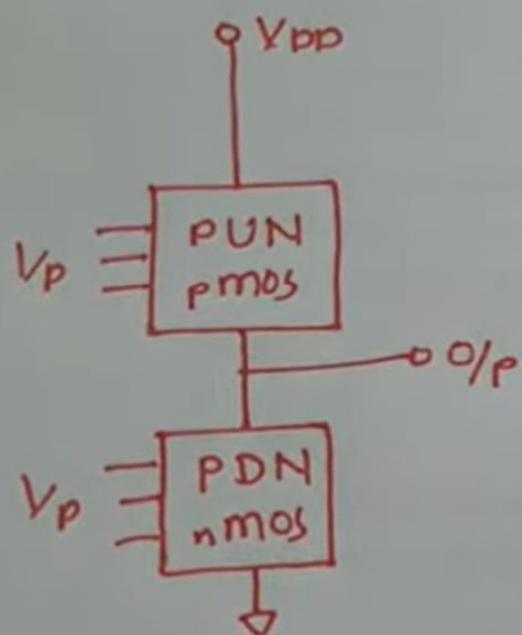
CMOS NAND Gate Working >



→ (.) Operation

pMOS - Parallel

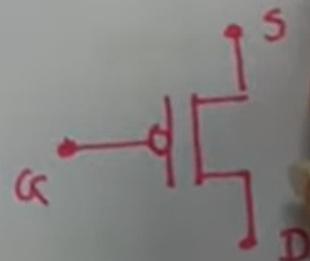
nMOS - Series



Truth table for the CMOS inverter:

Input A B	pMOS		nMOS		Output Y
	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	
0 0	ON	ON	OFF	OFF	1
0 1	ON	OFF	ON	OFF	0
1 0	OFF	ON	OFF	ON	0
1 1	OFF	OFF	ON	ON	1

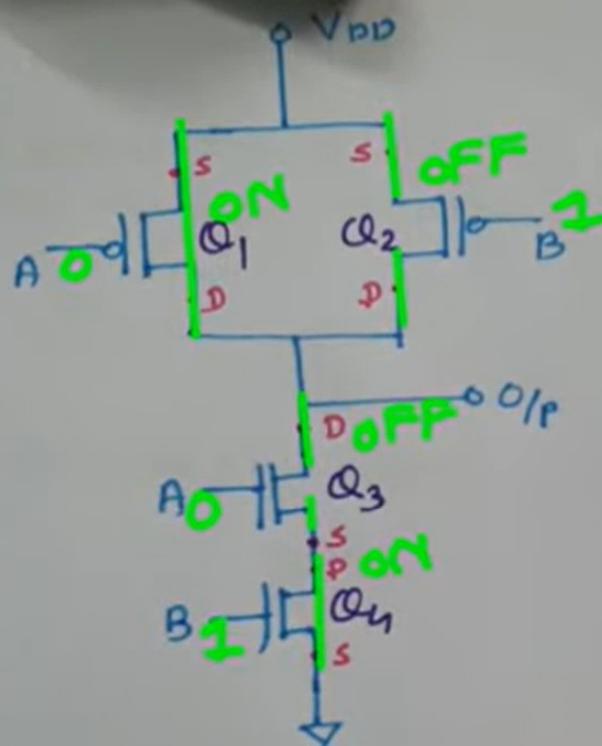
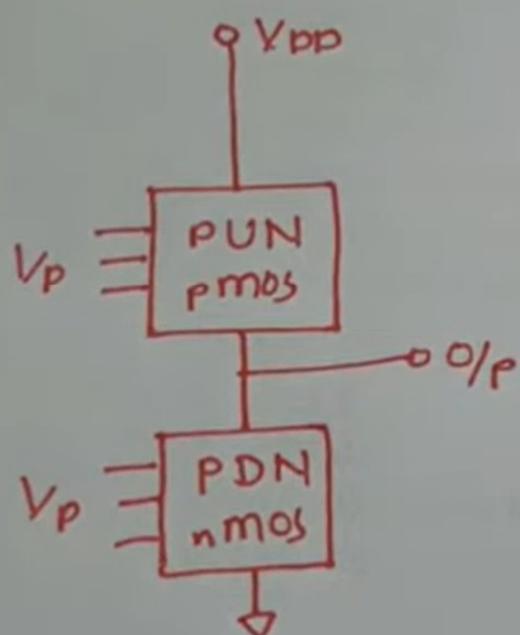
Diagram of a simplified nMOS logic symbol. It shows a single nMOS transistor with its drain terminal labeled D, source terminal labeled S, and gate terminal labeled G. Below the symbol, the text states:  $G=1 \rightarrow \text{ON}$  and  $G=0 \rightarrow \text{OFF}$ .



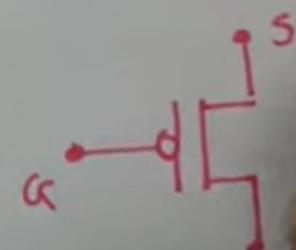
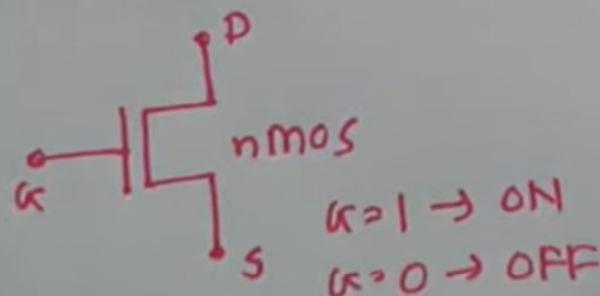
→ (.) Operation

pMOS - Parallel

nMOS - Series



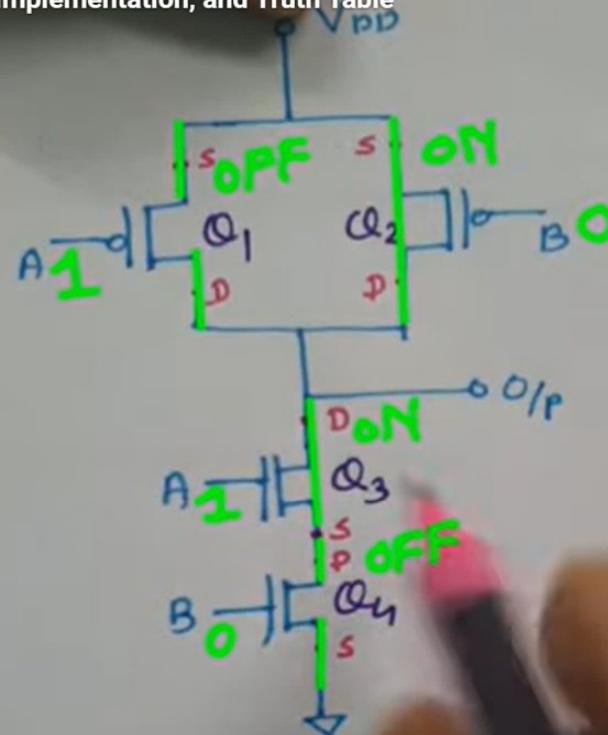
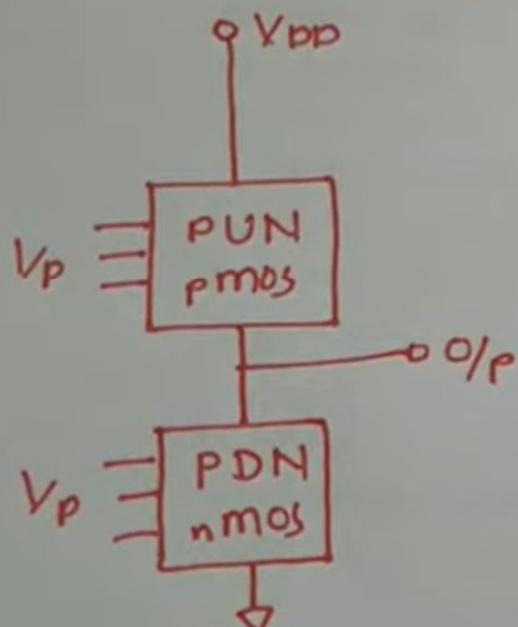
Input	pMOS		nMOS		Output
	A	B	Q <sub>1</sub> , Q <sub>2</sub>	Q <sub>3</sub> , Q <sub>4</sub>	
0 0	ON	ON	OFF	OFF	1
0 1	ON	OFF	OFF	ON	1



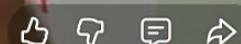
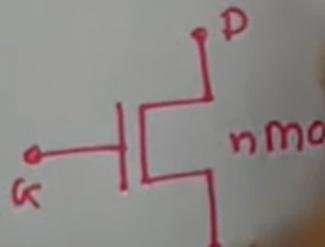
## CMOS NAND Gate Explained: Circuit, Working, Implementation, and Truth Table

pMOS - Parallel

nMOS - Series



Input A B	pMOS		nMOS		Output Y
	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	
0 0	ON	ON	OFF	OFF	1
0 1	ON	OFF	OFF	ON	1
1 0	OFF	ON	ON	OFF	0



11:26 / 12:22

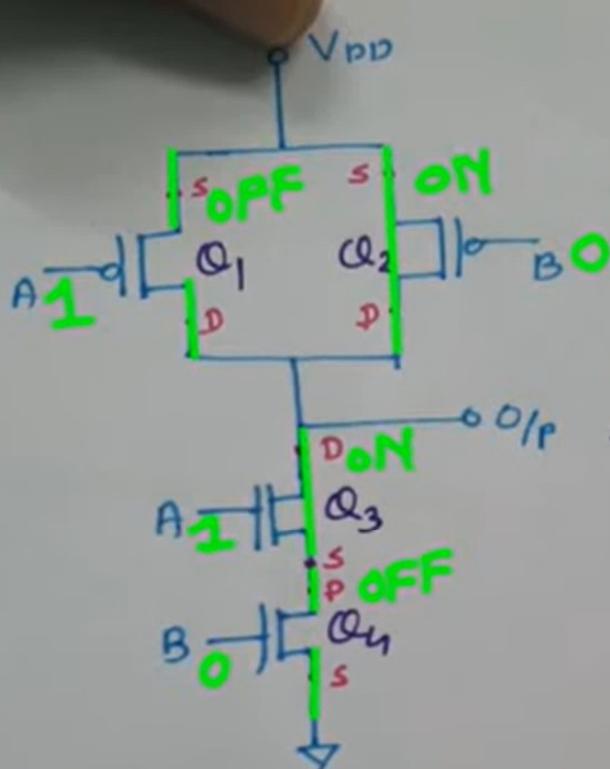
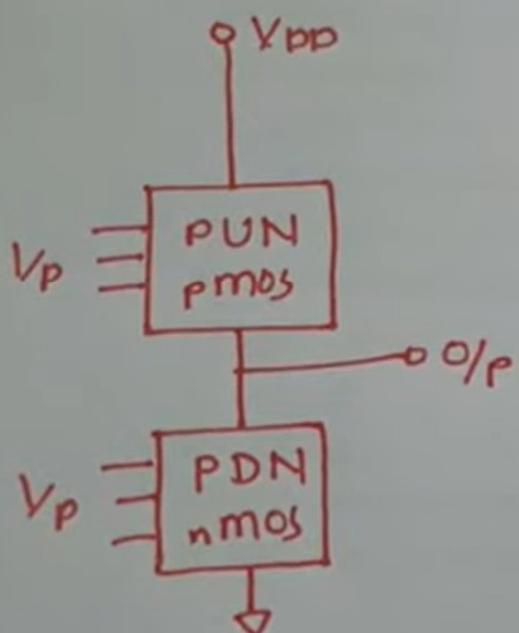
CMOS NAND Gate Working >



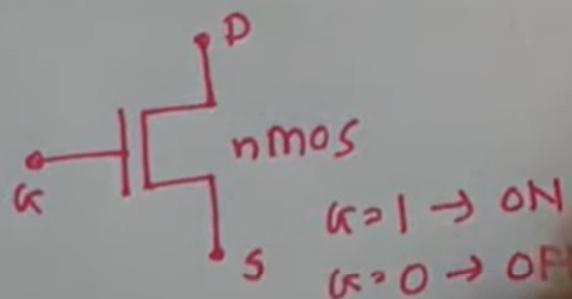
→ (.) Operation

pMOS - Parallel

nMOS - Series



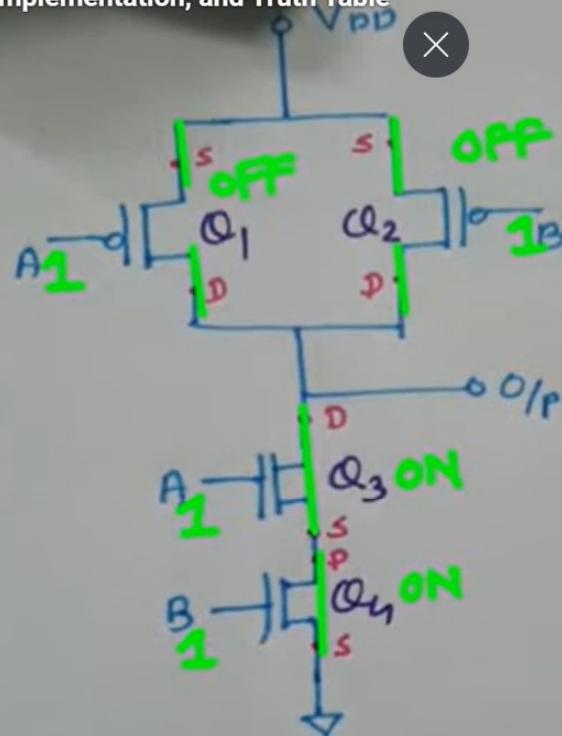
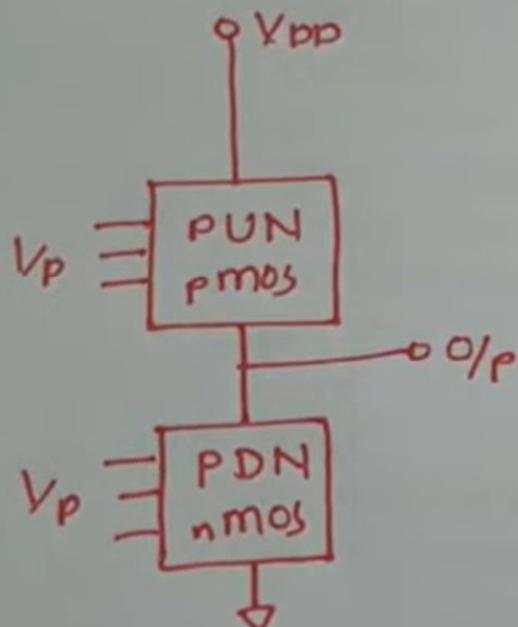
Input		pMOS	nMOS	Output
A	B	$Q_1, Q_2$	$Q_3, Q_4$	$Y$
0	0	ON	ON	OFF
0	1	ON	OFF	ON
1	0	OFF	ON	ON



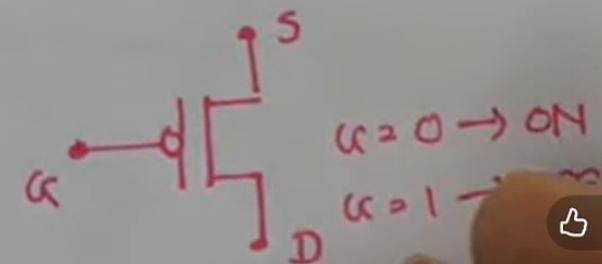
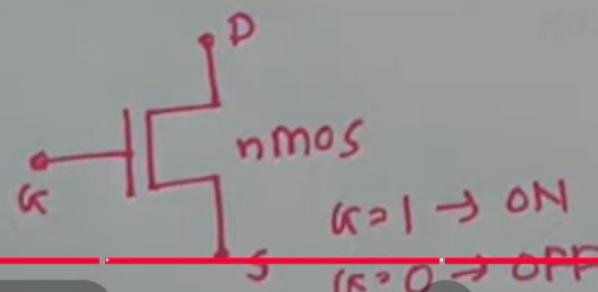
## CMOS NAND Gate Explained: Circuit, Working, Implementation, and Truth Table

pMOS - Parallel

nMOS - Series



Input A B	pmos		nmos		Output Y
	$Q_1$	$Q_2$	$Q_3$	$Q_4$	
0 0	ON	ON	OFF	OFF	1
0 1	ON	OFF	OFF	ON	1
1 0	OFF	ON	ON	OFF	1
1 1	OFF	OFF	OFF	ON	0

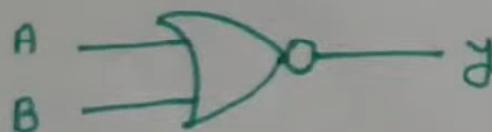


## Design of CMOS NOR gate

→ Boolean eqn.

$$Y = \overline{A + B}$$

→ Symbol



→ Truth Table

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

→ (+) Operation

nMOS - Parallel

pMOS - Series.



2:53 / 11:23

CMOS Circuit Rules &gt;

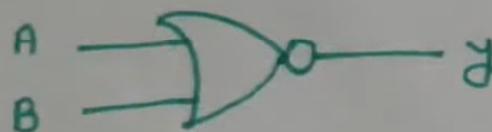


## Design of CMOS NOR gate

→ Boolean eqn.

$$Y = \overline{A + B}$$

→ Symbol



→ Truth Table

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

→ (+) Operation

nMOS - Parallel

pMOS - Series.



2:51 / 11:23

CMOS Circuit Rules &gt;



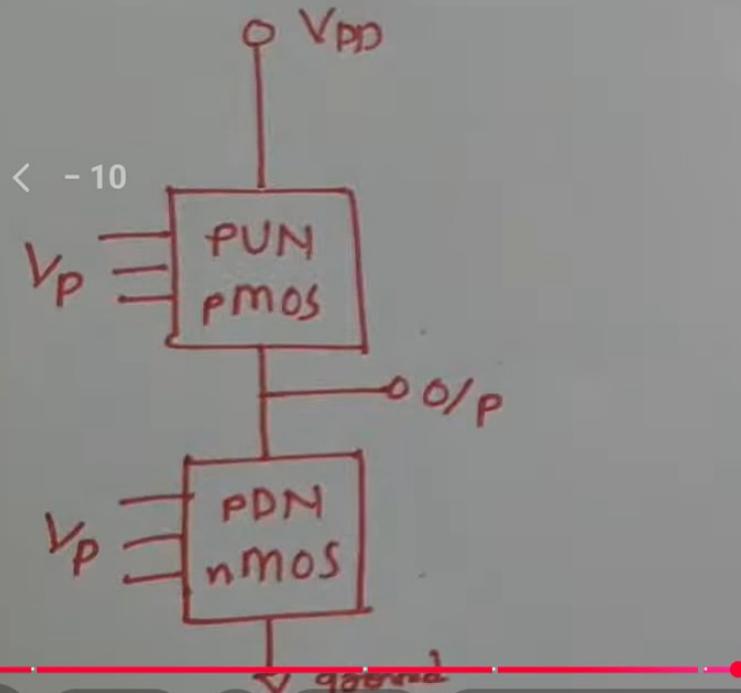
## CMOS NOR Gate Explained: Circuit, Working, Implementation, and Truth Table



→ (+) Operation

nMOS - Parallel

pMOS - Series.



4:13 / 11:23

CMOS NOR Gate implementation >



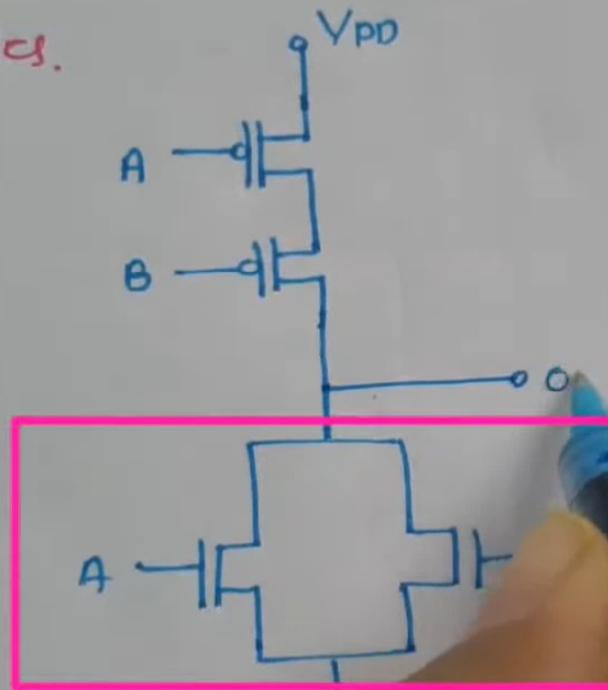
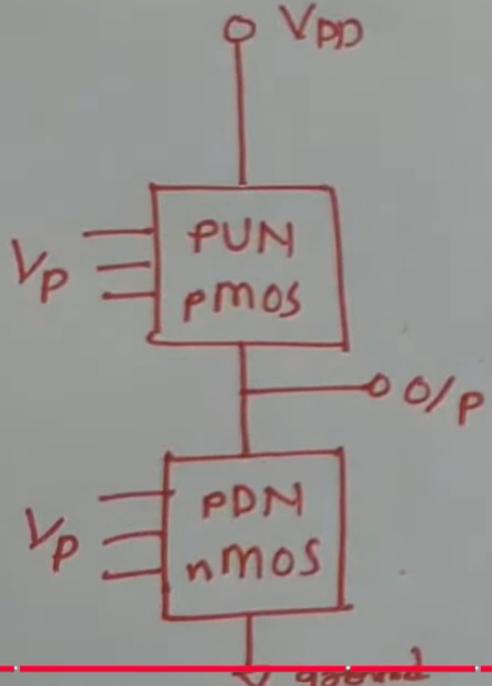
## CMOS NOR Gate Explained: Circuit, Working, Implementation, and Truth Table

1 1 | 0

→ (+) Operation

nMOS - Parallel

pMOS - Series.

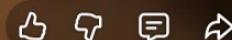
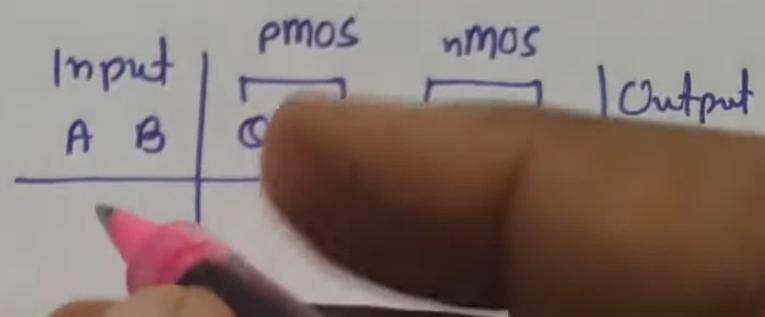
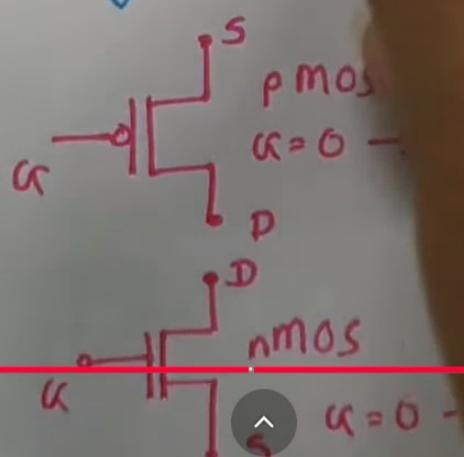
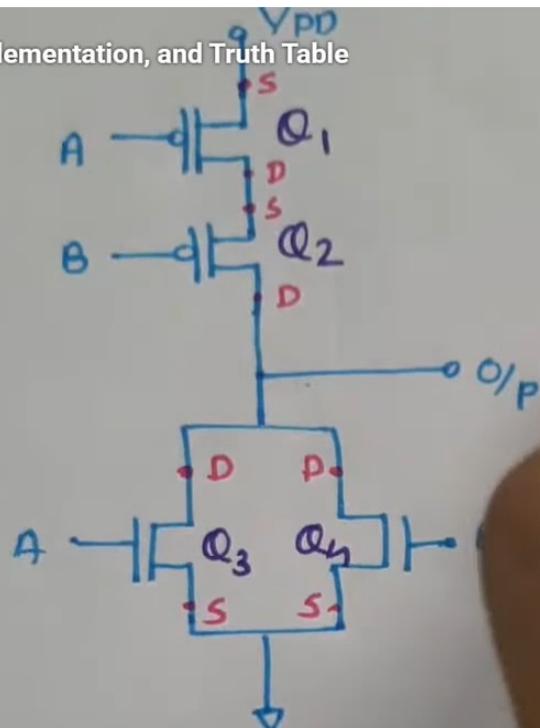
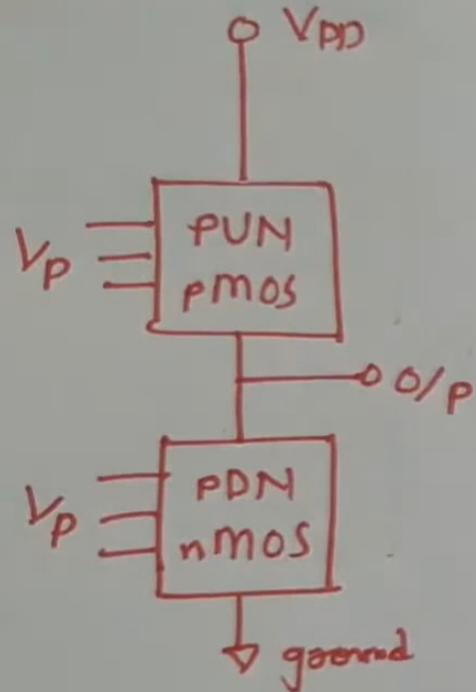


5:29 / 11:23

CMOS NOR Gate implementation >



## PMOS - Series



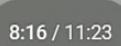
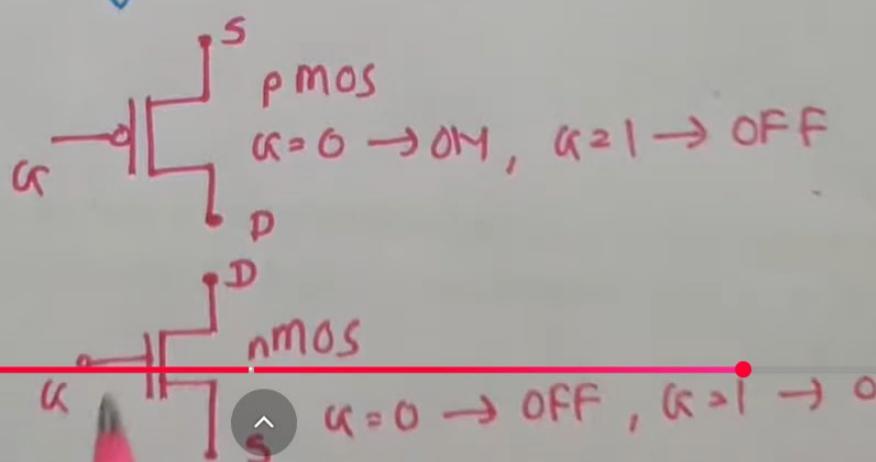
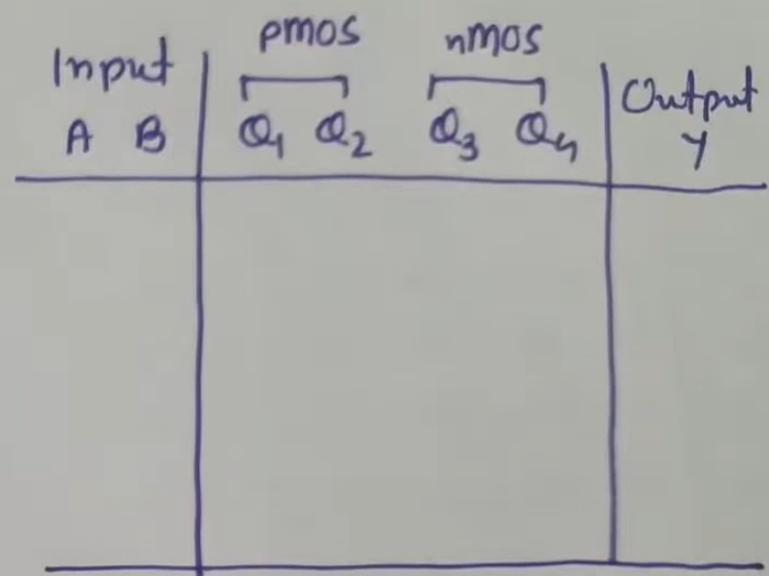
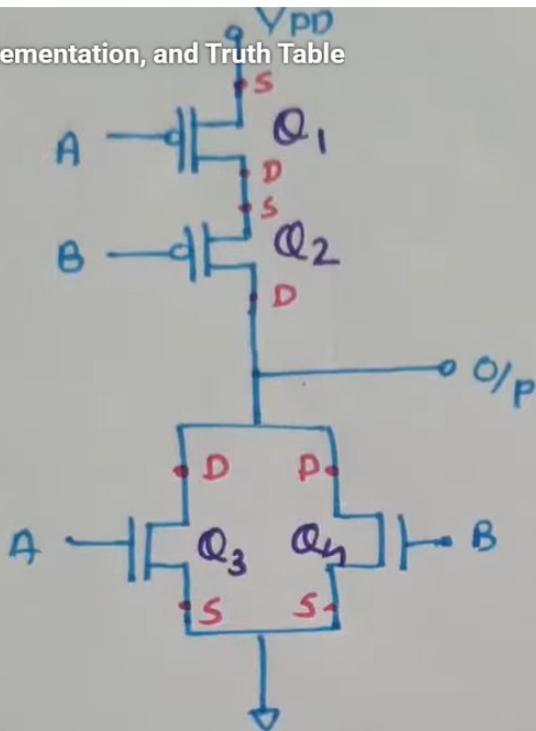
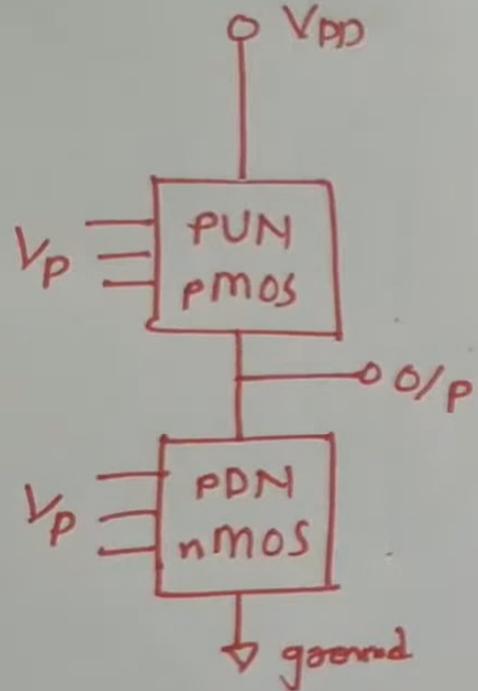
8:24 / 11:23

CMOS NOR Gate Working &gt;



## PMOS - Series

CMOS NOR Gate Explained: Circuit, Working, Implementation, and Truth Table

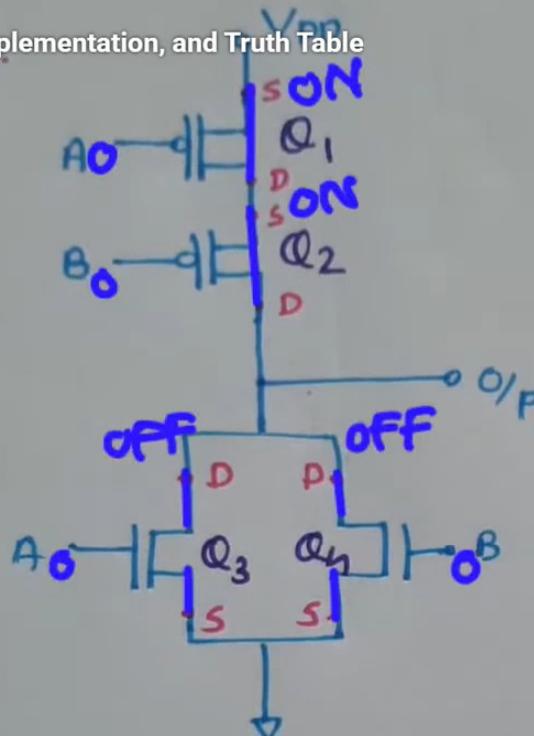
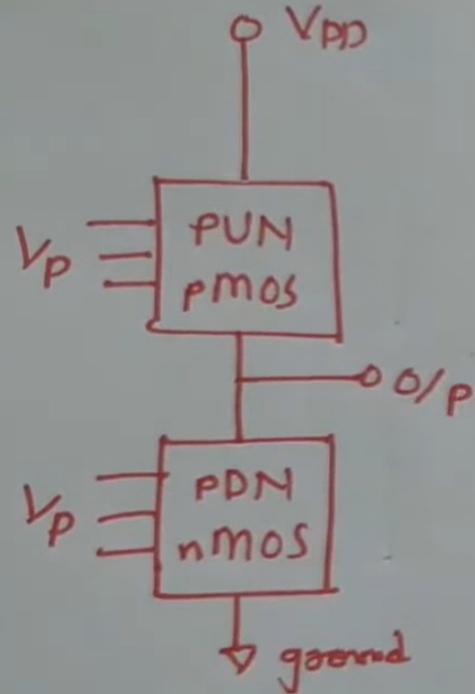


8:16 / 11:23

CMOS NOR Gate Working >



## CMOS NOR Gate Explained: Circuit, Working, Implementation, and Truth Table



Input	pmos		nmos		Output
	$Q_1$	$Q_2$	$Q_3$	$Q_4$	
0 0	ON	ON	OFF	OFF	1
0 1	ON	OFF	ON	OFF	0
1 0	OFF	ON	ON	OFF	0
1 1	OFF	OFF	ON	ON	0

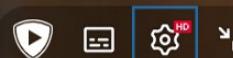
$\alpha$  — p MOS  
 $\alpha = 0 \rightarrow ON, \alpha = 1 \rightarrow OFF$

$\kappa$  — n MOS  
 $\kappa = 0 \rightarrow OFF, \kappa > 1 \rightarrow ON$

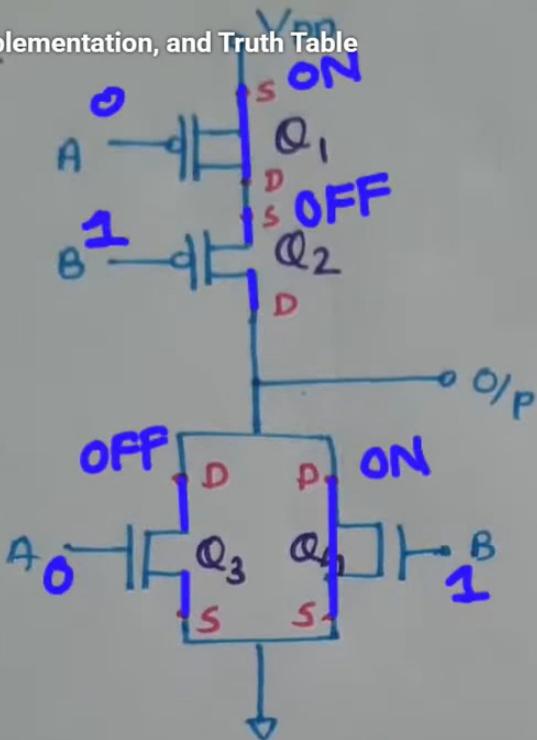
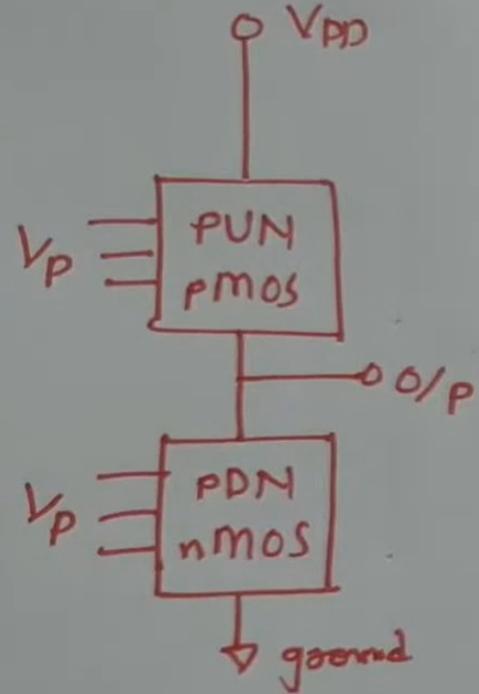


9:11 / 11:23

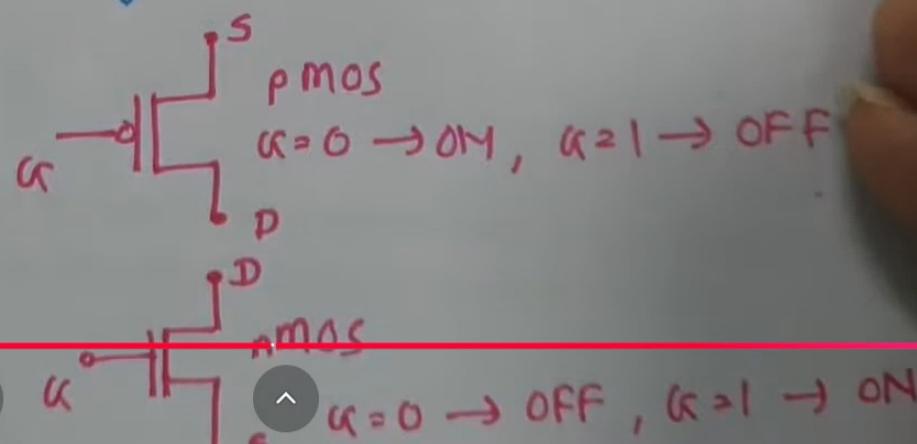
CMOS NOR Gate Working >



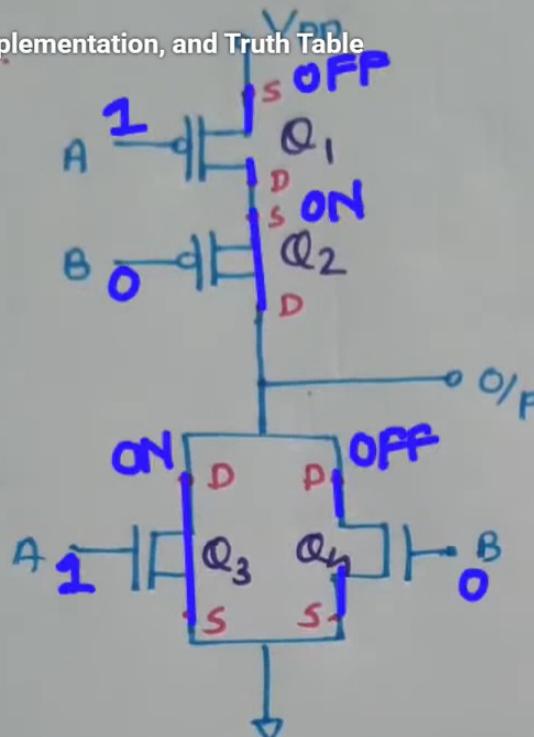
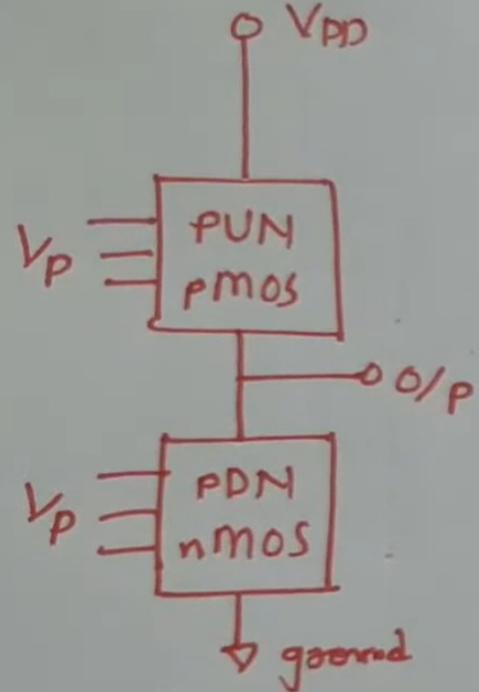
## CMOS NOR Gate Explained: Circuit, Working, Implementation, and Truth Table



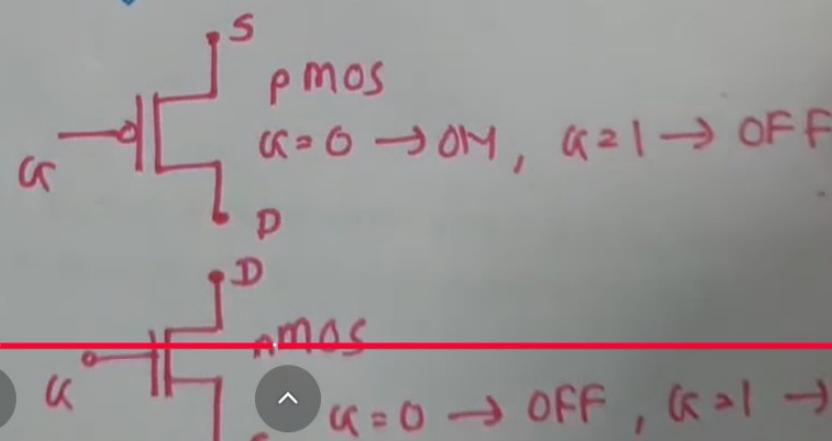
Input	pmos		nmos		Output Y
	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	
0 0	ON	ON	OFF	OFF	1
0 1	ON	OFF	OFF	ON	0
1 0	OFF	ON	ON	OFF	0
1 1	OFF	OFF	ON	ON	1



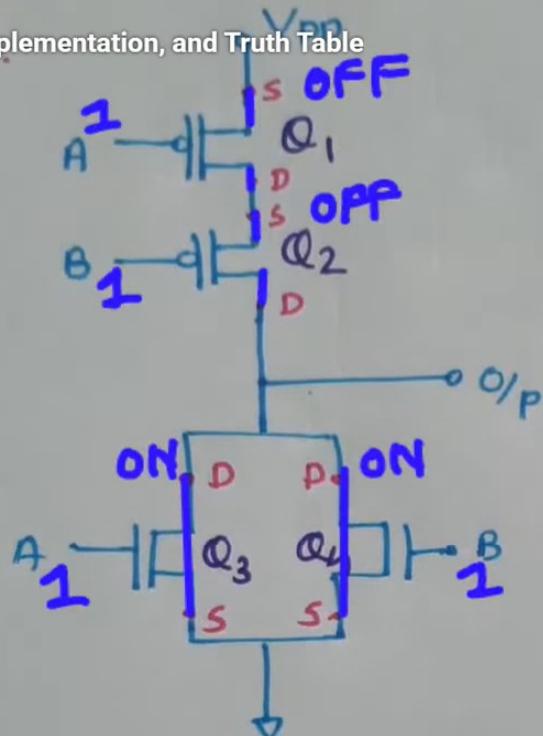
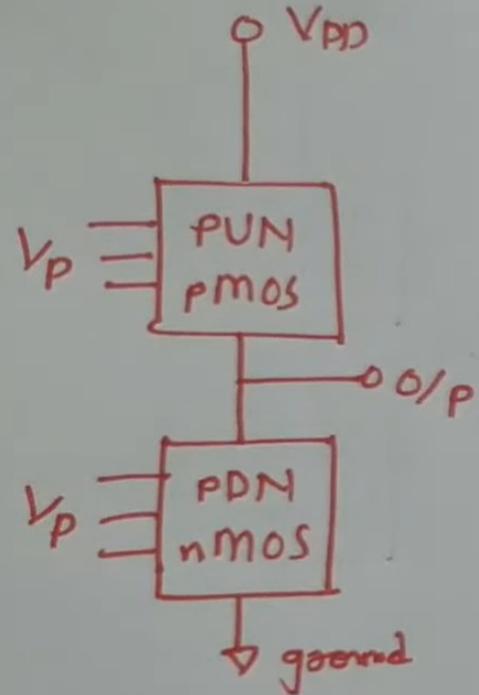
## CMOS NOR Gate Explained: Circuit, Working, Implementation, and Truth Table



Input	pmos	nmos	Output		
A	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$\gamma$
0 0	ON	ON	OFF	OFF	1
0 1	ON	OFF	OFF	ON	0
1 0	OFF	ON	ON	OFF	0
1 1					



## CMOS NOR Gate Explained: Circuit, Working, Implementation, and Truth Table



Input	pmos		nmos		Output
	$Q_1$	$Q_2$	$Q_3$	$Q_4$	
0 0	ON	ON	OFF	OFF	1
0 1	ON	OFF	OFF	ON	0
1 0	OFF	ON	ON	OFF	0
1 1	OFF	OFF	ON	ON	0

$\alpha$  — pMOS  
 $\alpha = 0 \rightarrow \text{ON}, \alpha = 1 \rightarrow \text{OFF}$

$\kappa$  — nMOS  
 $\kappa = 0 \rightarrow \text{OFF}, \kappa = 1 \rightarrow \text{ON}$



11:00 / 11:23

CMOS NOR Gate Working >



Boolean Function using CMOS Transistor

→ Implement  $y = A(B+C) + CD$  using CMOS Transistor

→ For ( $\cdot$ ) Operation

nMOS  $\rightarrow$  Series

pMOS  $\rightarrow$  Parallel

→ ( $+$ ) Operation

nMOS  $\rightarrow$  Parallel

pMOS  $\rightarrow$  Series.

## Boolean function using CMOS Transistor

→ Implement  $y = A \cdot (B + C) + CD$  using CMOS Transistor

→ For ( $\cdot$ ) Operation      → ( $+$ ) Operation

nMOS  $\rightarrow$  Series

nMOS  $\rightarrow$  Parallel

pMOS  $\rightarrow$  Parallel

pMOS  $\rightarrow$  Series.

→ At last we need to invert the f.

< - 5



2:00 / 8:46

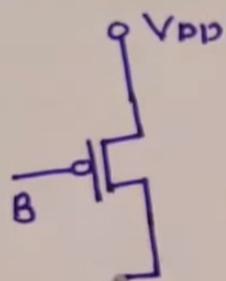
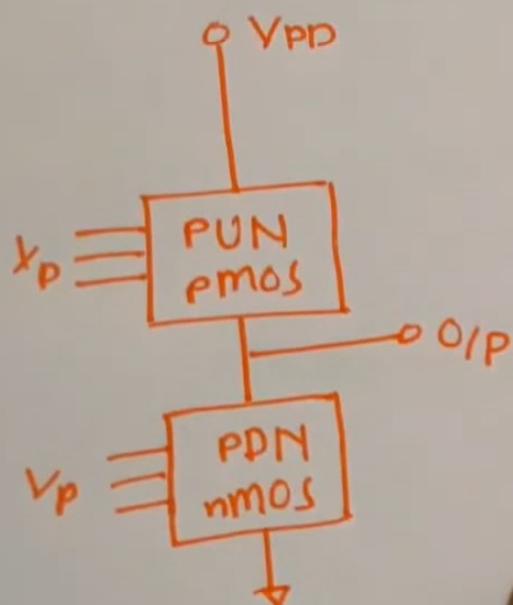
CMOS Circuit Rules &gt;



→ Implement  $y = A \cdot (B+C) + CD$  using CMOS Transistor

→ For (-) Operation      → (+) Operation  
nMOS → Series  
pMOS → Parallel  
pMOS → Series.

→ At Least we need to invert the function

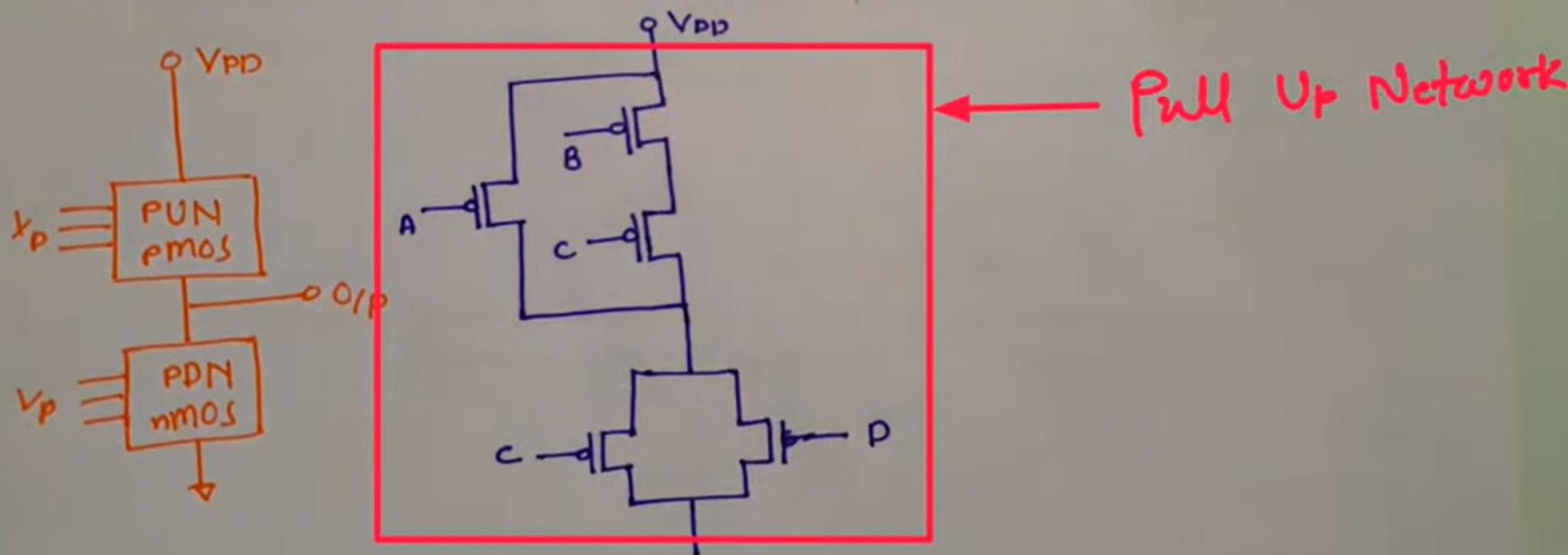


→  $B+C$  means, In Pull Up Network, by using pMOS, we should connect pMOS in series.

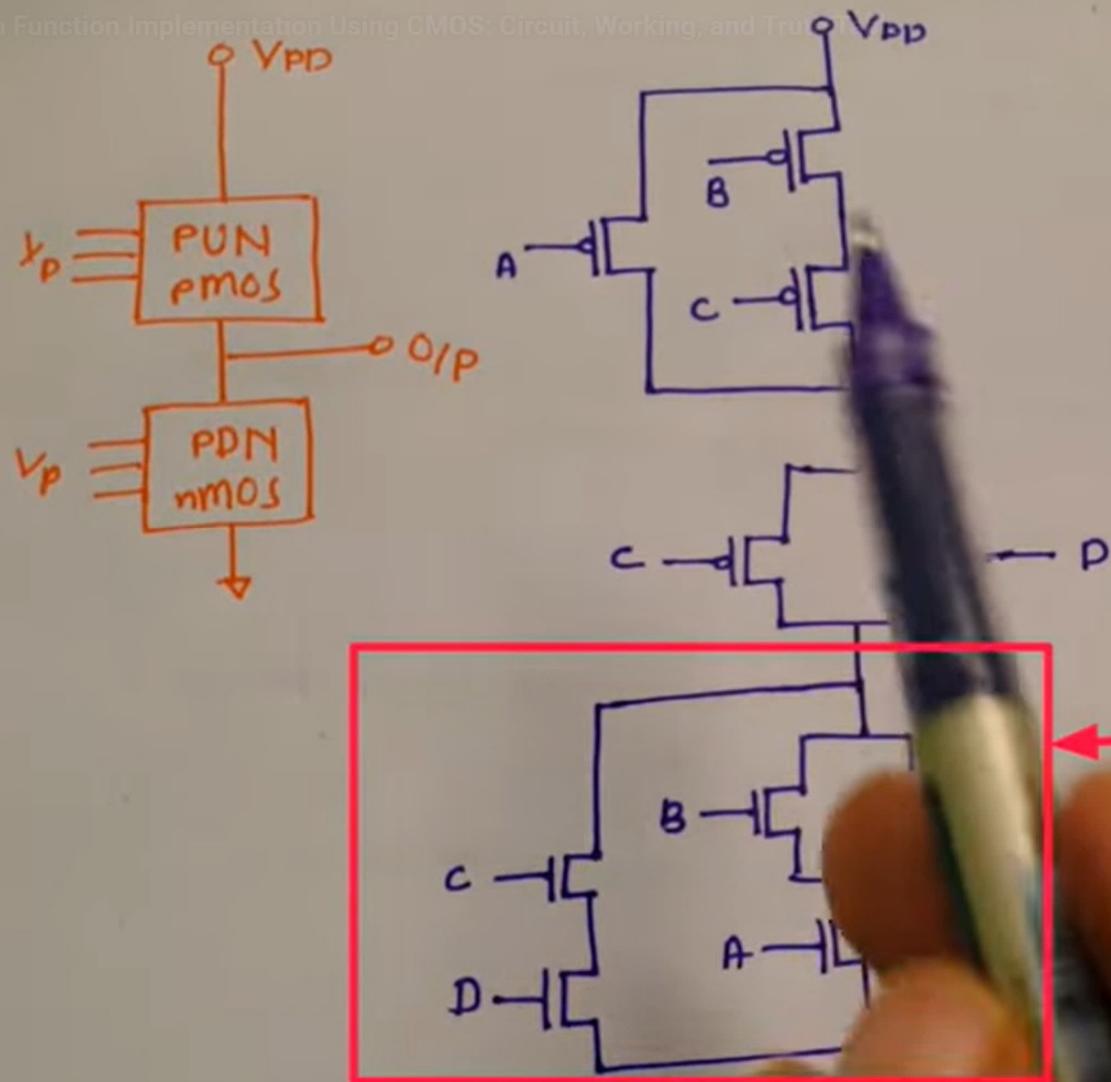
→ Implement  $y = A \cdot (B + C) + CD$  using CMOS Transistor

→ For (-) Operation → (+) Operation  
nMOS → Series  
pMOS → Parallel  
pMOS → Series.

→ At least we need to invert the function



## Boolean Function Implementation Using CMOS: Circuit, Working, and Truth Table

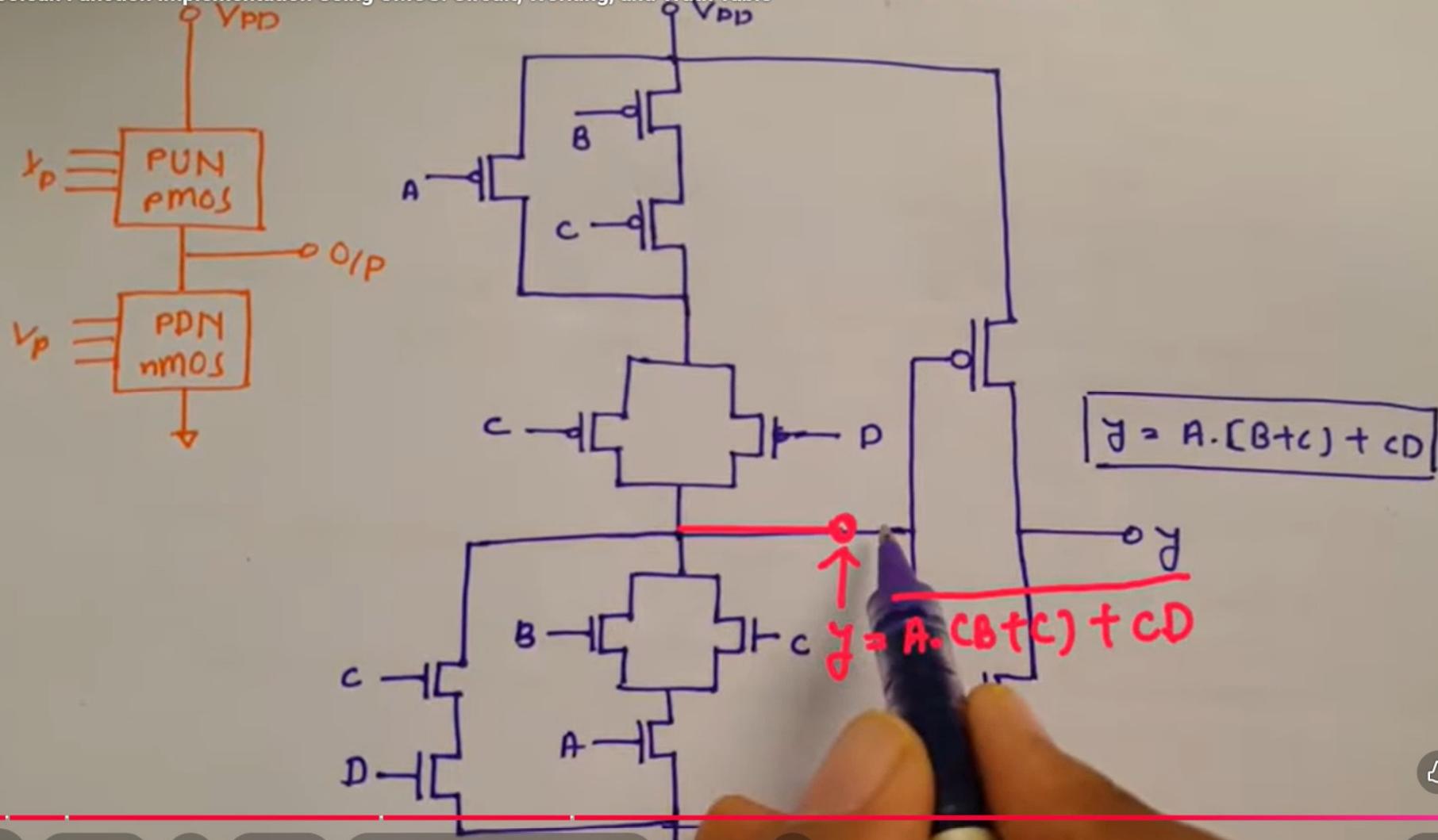


$$Y = A \cdot (B + C) + C \cdot D$$

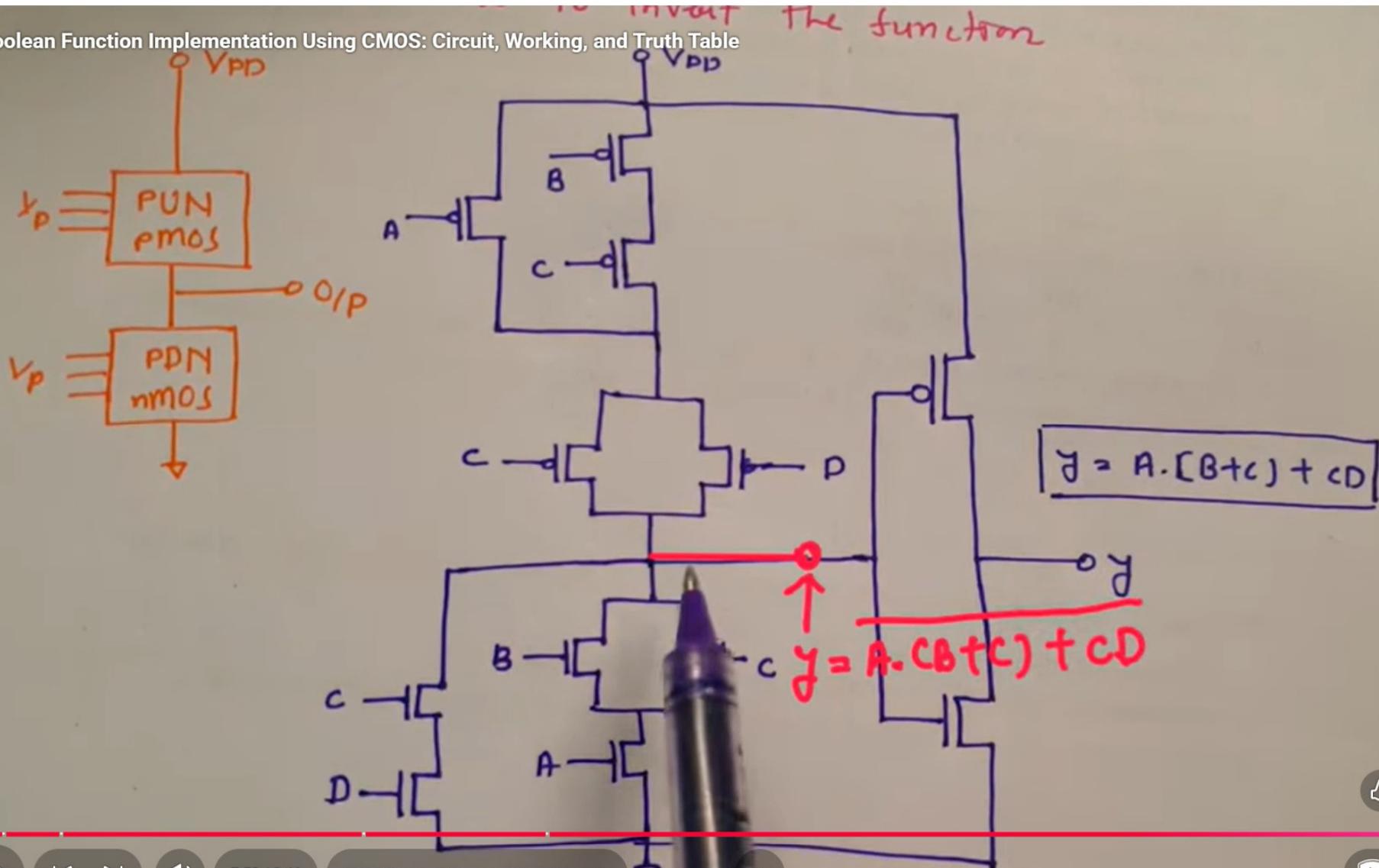
Pull Down Network

Boolean Function Implementation Using CMOS: Circuit, Working, and Truth Table

## The function

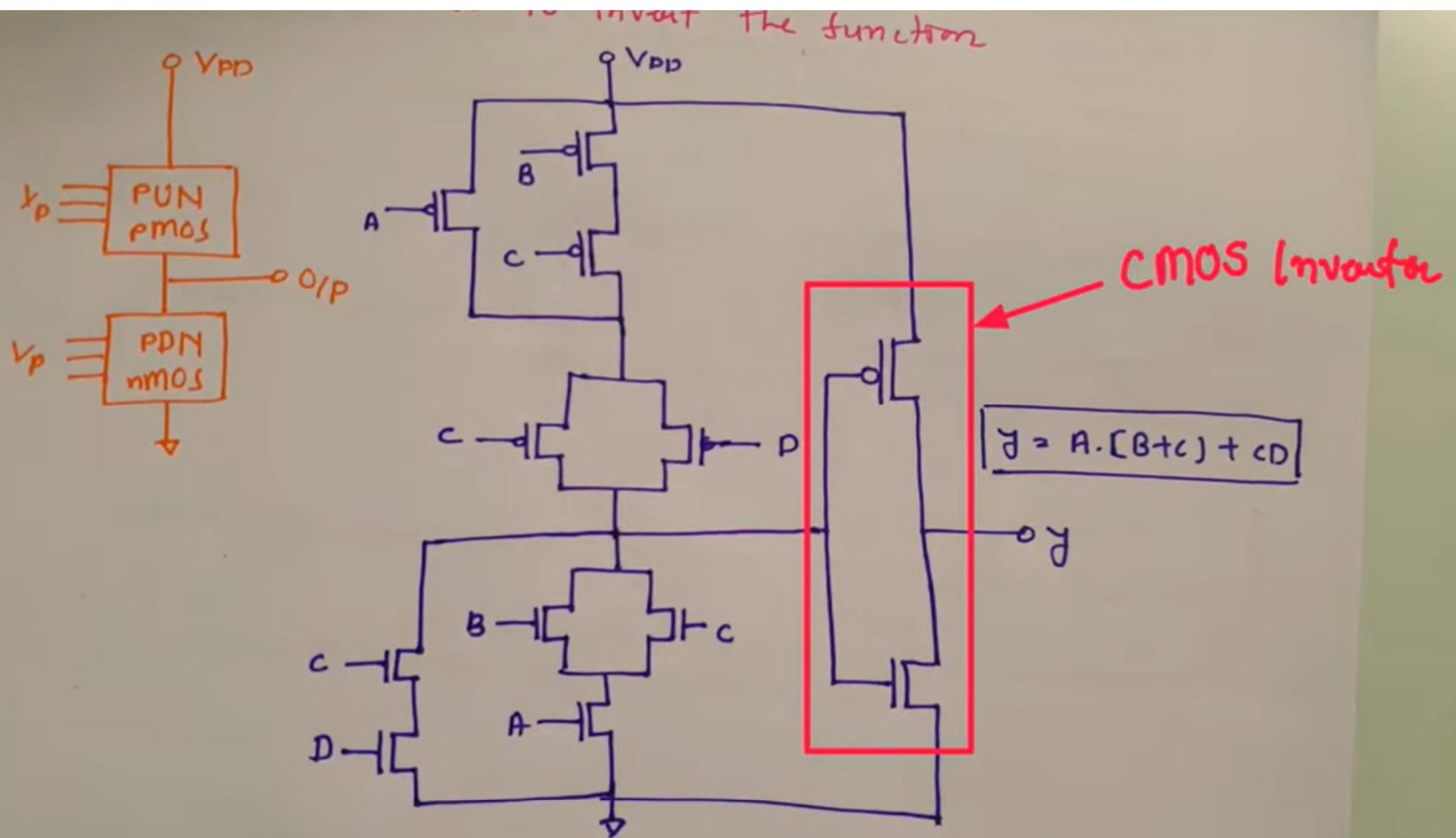


## Boolean Function Implementation Using CMOS: Circuit, Working, and Truth Table



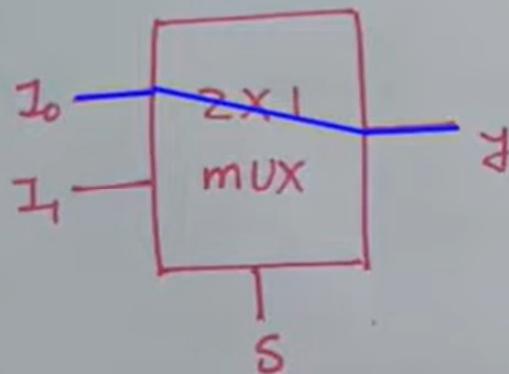
7:58 / 8:46

CMOS Boolean Function implementation &gt;



CMOS Multiplexer

→ I will explain 2x1 MUX



Selection line	Output
1	Y
0	1



◀



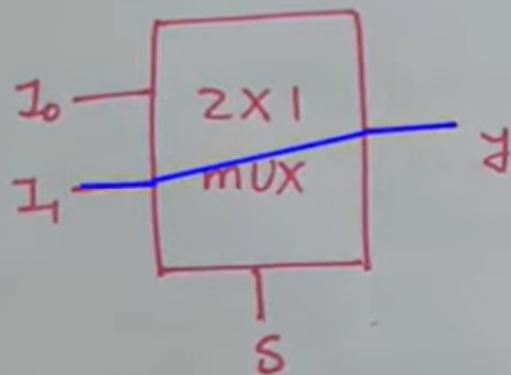
1:10 / 9:11

2 to 1 Multiplexer (Working & Truth Table) >



## CMOS Multiplexer

→ I will explain 2x1 MUX

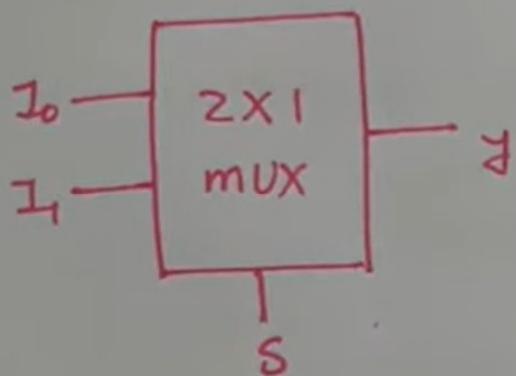


Selection Line	Output
1	$I_0$
0	$I_1$
11c	



## CMOS Multiplexer

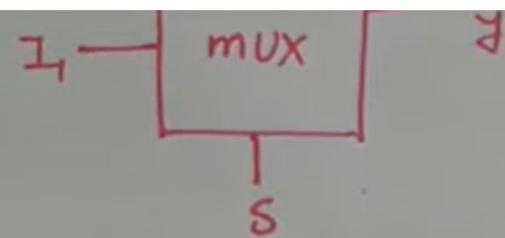
→ I will explain 2x1 MUX



Selection line		Output
$s$	0	$y$
$s$	1	$I_0$
$s$	1	$I_1$

→ O/P

$$y = \bar{s}I_0 + sI_1$$



$s$	$y$
0	$I_0$
1	$I_1$

$$\rightarrow \text{O/P} \\ y = \bar{s}I_0 + sI_1$$

→ For (-) Operation

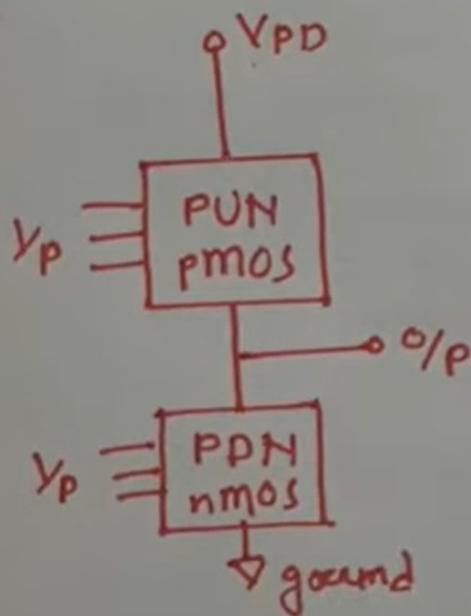
nMOS → Series

pMOS → Parallel

→ (+) Operation

nMOS → Parallel

pMOS → Series.



## CMOS Multiplexer: Basics, Circuit, Rules, Working, Implementation & Truth Table

→ For (-) Operation

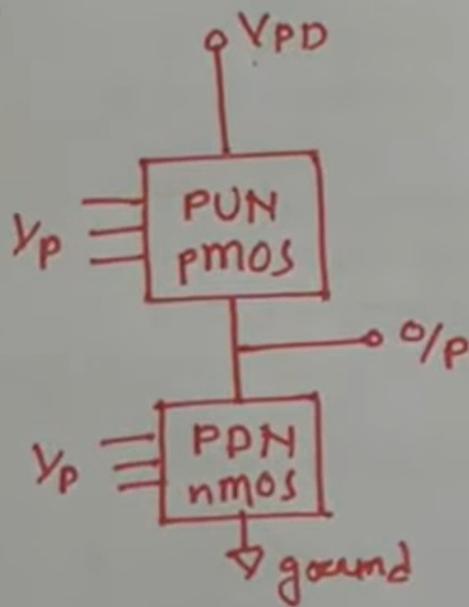
nMOS → Series

pMOS → Parallel

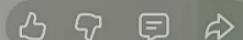
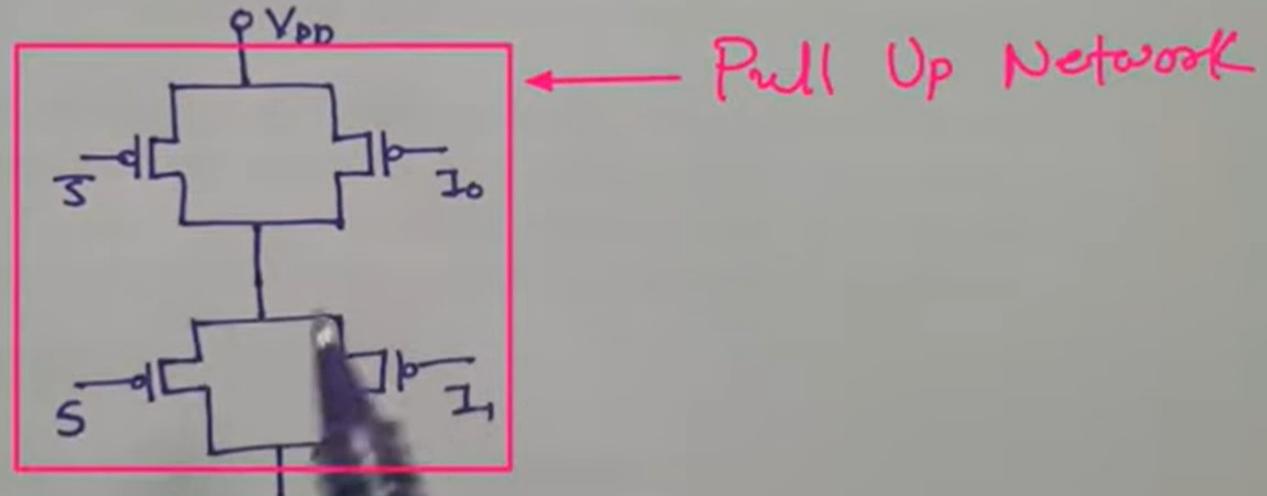
→ (+) Operation

nMOS → Parallel

pMOS → Series



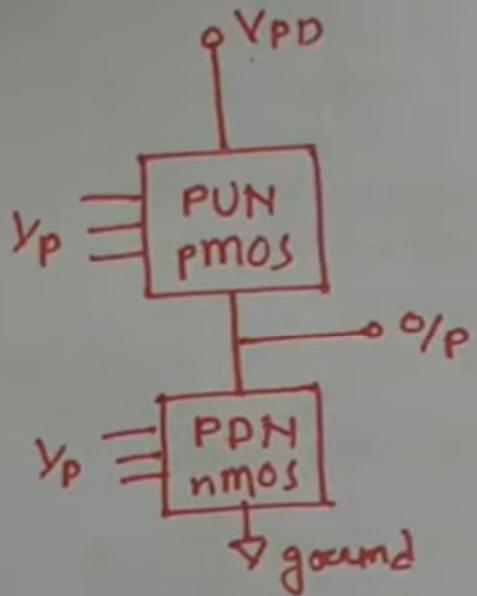
$$I_o = \bar{s}I_0 + sI_1$$



6:26 / 9:11 CMOS Multiplexer implementation >

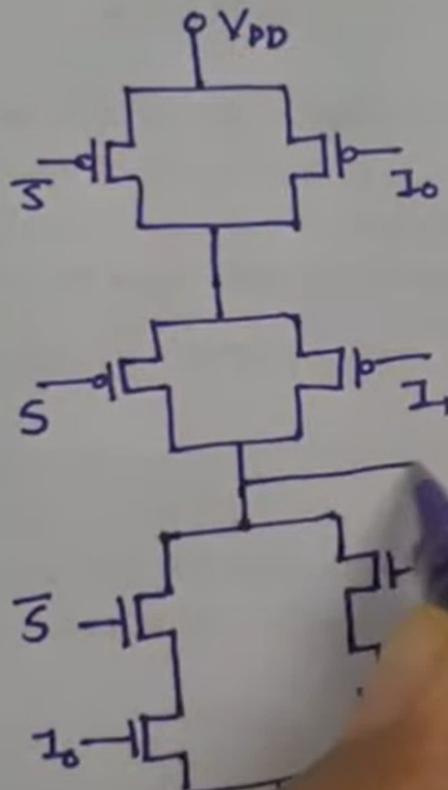


nMOS  $\rightarrow$  Series  
pMOS  $\rightarrow$  Parallel)



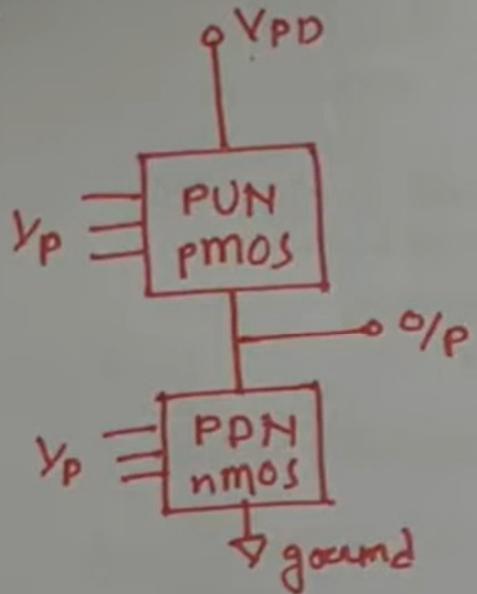
nMOS  $\rightarrow$  Parallel  
pMOS  $\rightarrow$  Series.

$$I = \bar{S} I_0 + S I_1$$

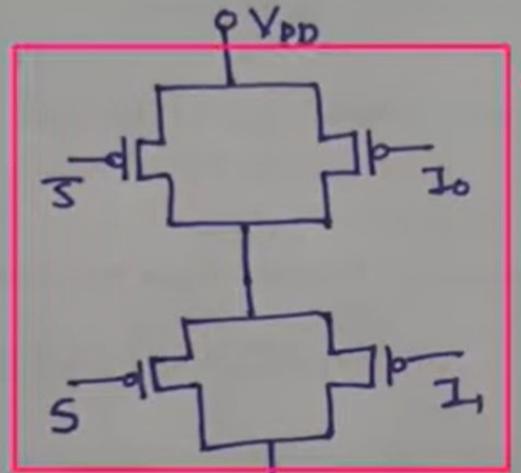


nMOS  $\rightarrow$  Series  
pMOS  $\rightarrow$  Parallel

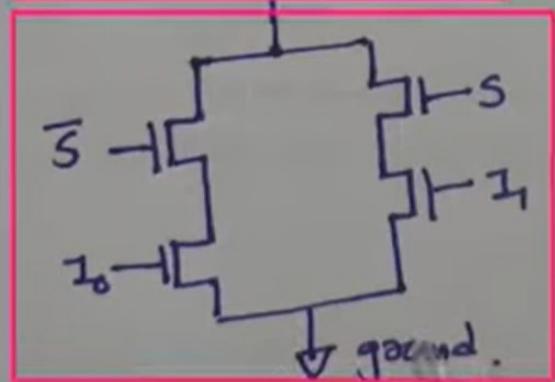
nMOS  $\rightarrow$  Parallel  
pMOS  $\rightarrow$  Series.



$$I = \bar{s}I_0 + sI_1$$

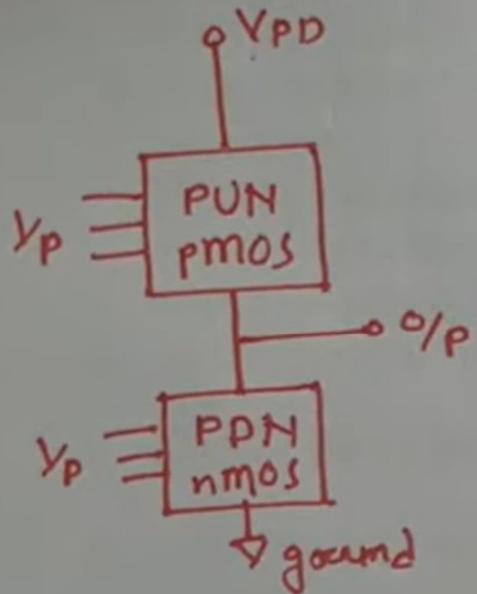


Pull Up Network



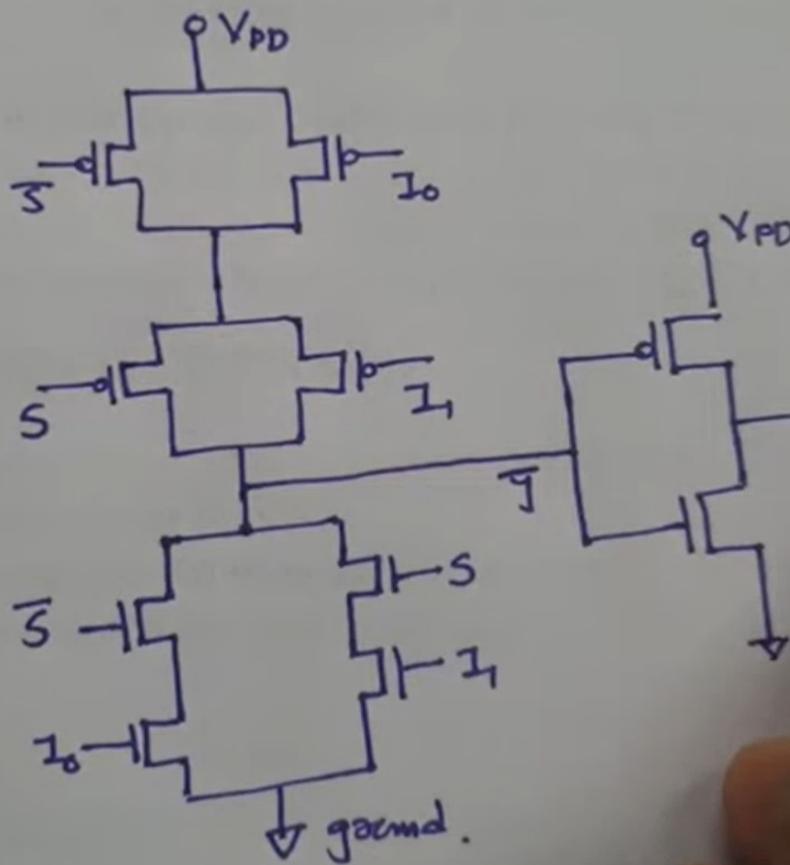
Pull Down Network

nMOS  $\rightarrow$  Series  
pMOS  $\rightarrow$  Parallel



nMOS  $\rightarrow$  Parallel  
pMOS  $\rightarrow$  Series.

$$I = \bar{s}I_0 + sI_1$$



→ For (-) Operation

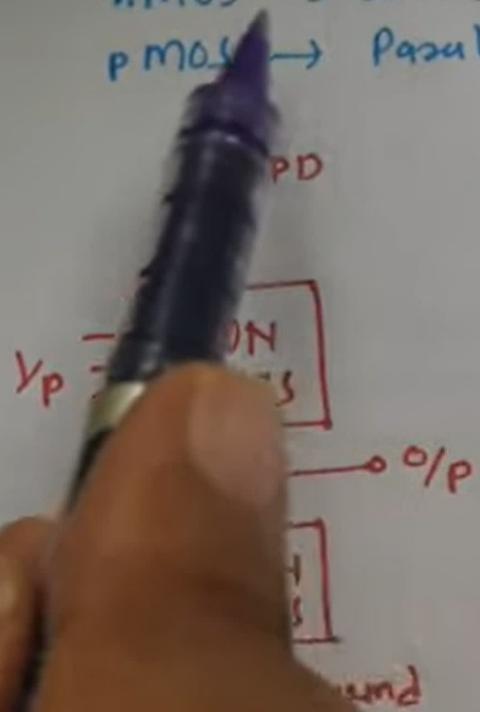
nMOS → Series

pMOS → Parallel

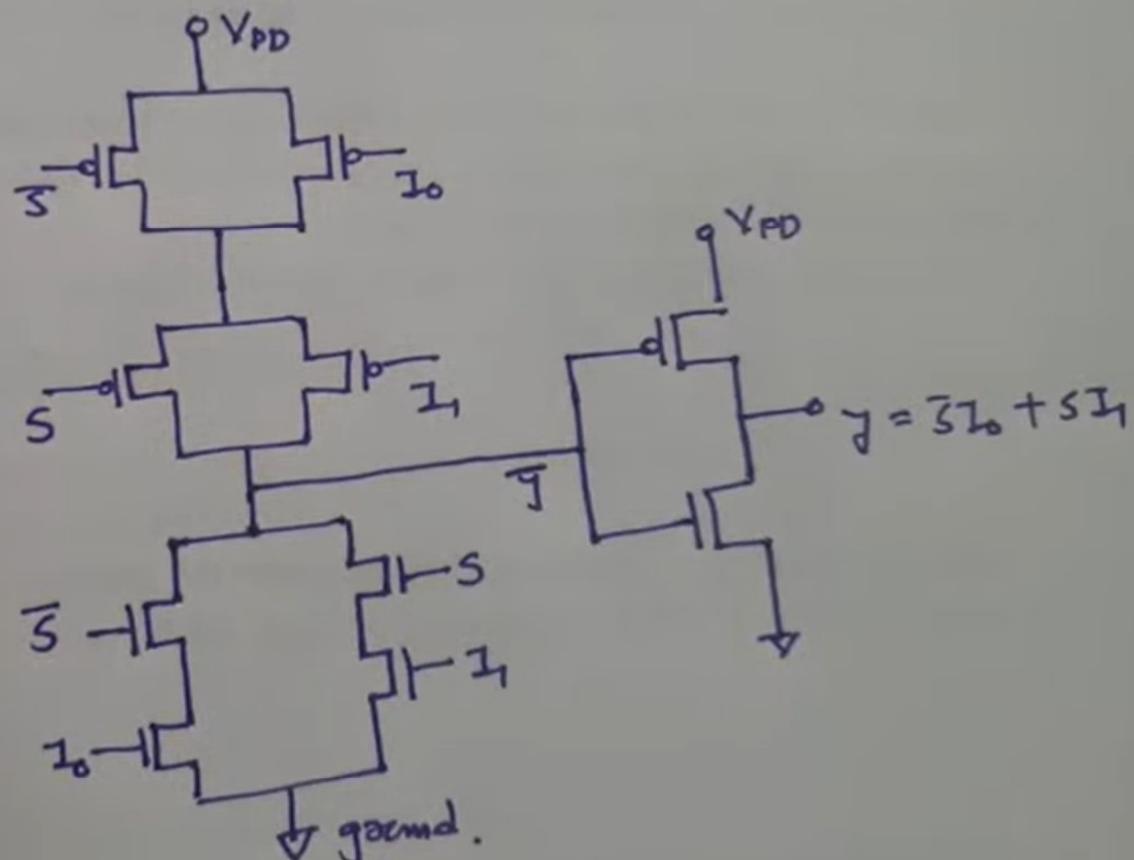
→ (+) Operation

nMOS → Parallel

pMOS → Series.

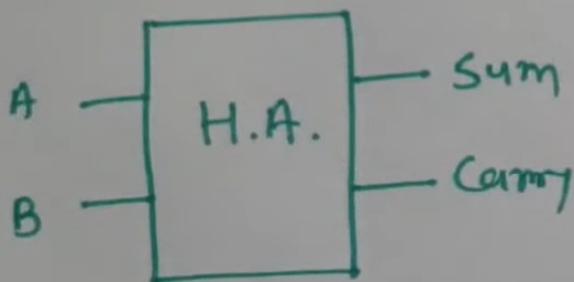


$$I = \bar{s} I_o + s I_1$$



## CMOS Half Adder

→ Half Adder is two bits addition module.

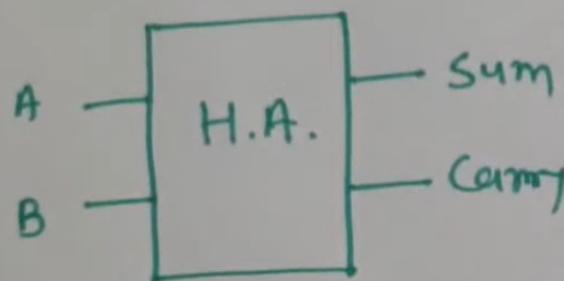


Truth Table

Inputs		Outputs	
A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{aligned}\rightarrow \text{sum} &= A \oplus B \\ &= AB + \bar{A}\bar{B} \\ &= \overline{AB + \bar{A}\bar{B}}\end{aligned}$$

→ Half Adder is two bits addition module.



Truth Table

Inputs		Outputs	
A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{aligned} \text{Sum} &= A \oplus B \\ &= AB + \bar{A}\bar{B} \\ &= \overline{AB + \bar{A}\bar{B}} \end{aligned}$$

$$\begin{aligned} \text{Carry} &= A \cdot B \\ &= \overline{\overline{A} \cdot B} \\ &= \overline{\overline{A} + B} \end{aligned}$$



3:37 / 10:51

CMOS Circuit Rules &gt;



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

CMOS Half Adder: Basics, Circuit, Rules, Working, Implementation & Truth Table

→ For (.) Operation

pMOS → Parallel

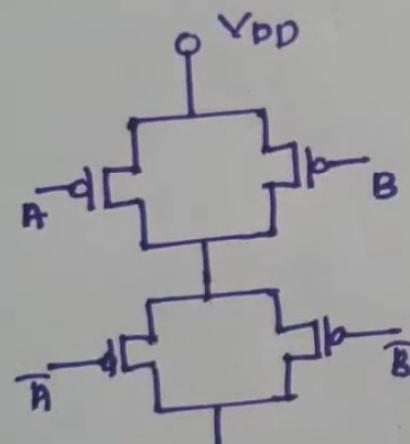
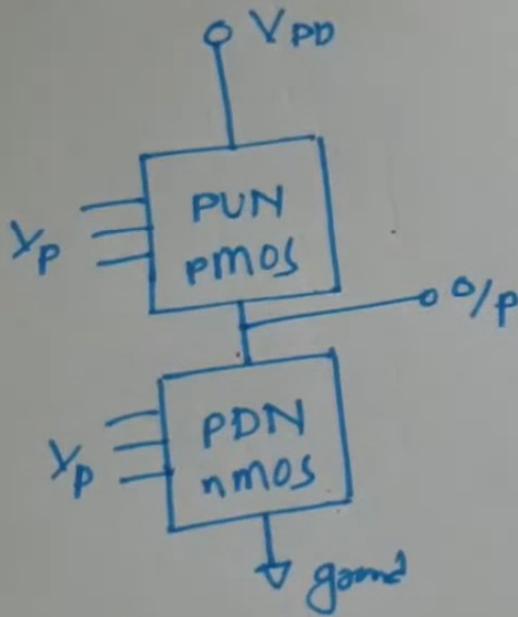
nMOS → Series

→ For (+) Operation

pMOS → Series

nMOS → Parallel

$$\text{Sum} = \overline{A \cdot B} + \overline{A} \cdot \overline{B}$$



+ 5 >



7:22 / 10:51

CMOS Half Adder implementation >



## CMOS Half Adder: Basics, Circuit, Rules, Working, Implementation & Truth Table

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

$\rightarrow$   $f_{00} (.)$  Operation

$\rightarrow$   $f_{00} (+)$  Operation

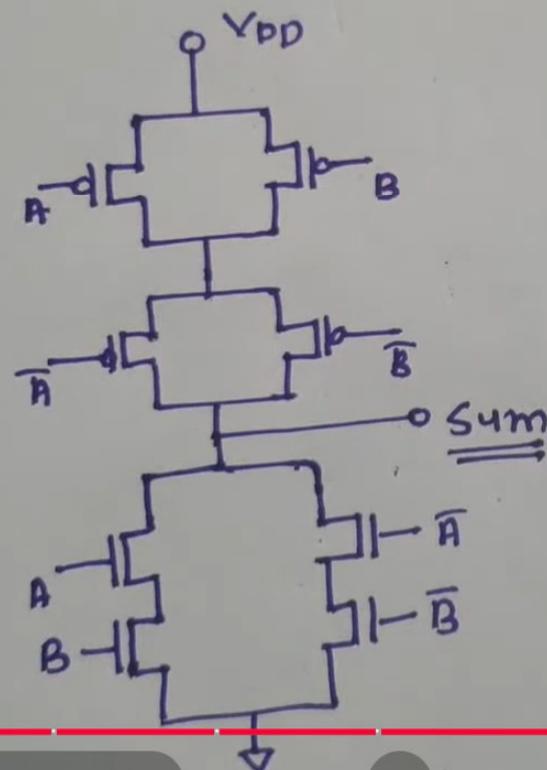
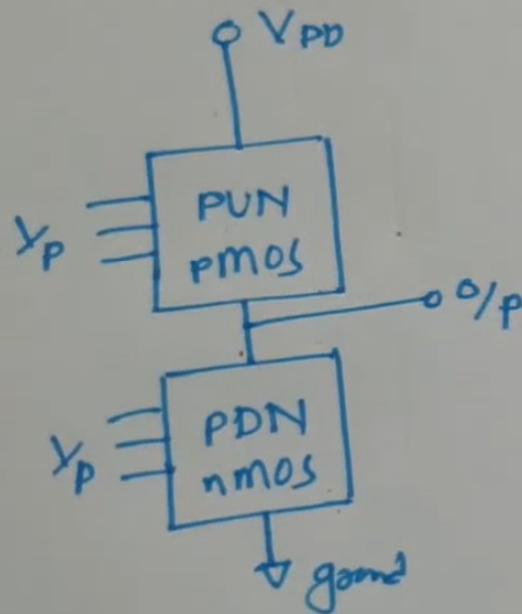
pMOS  $\rightarrow$  Parallel

nMOS  $\rightarrow$  Series

nMOS  $\rightarrow$  Series

nMOS  $\rightarrow$  Parallel

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



8:46 / 10:51

CMOS Half Adder implementation >



→ For (+) Operation → For (-) Operation

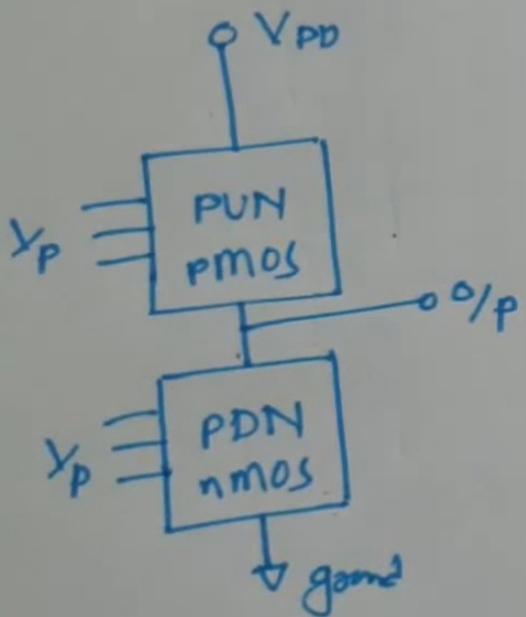
pMOS → Parallel

nMOS → Series

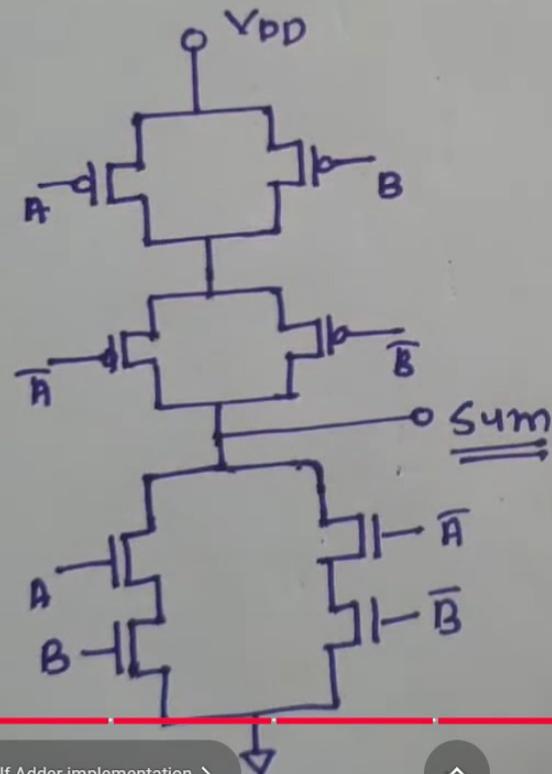
pMOS → Series

nMOS → Parallel

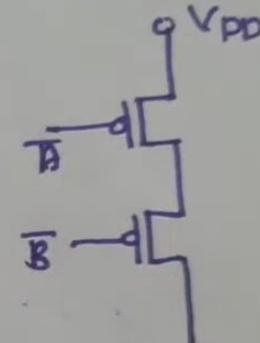
$$= \overline{A} + \overline{B}$$



$$\text{Sum} = \overline{A} \cdot B + A \cdot \overline{B}$$

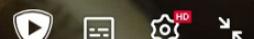
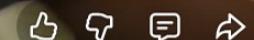


$$\text{Carry} = \overline{A} + \overline{B}$$



9:42 / 10:51

CMOS Half Adder implementation >



→  $f_{\text{OR}}(.)$  Operation →  $f_{\text{OR}}(+)$  Operation

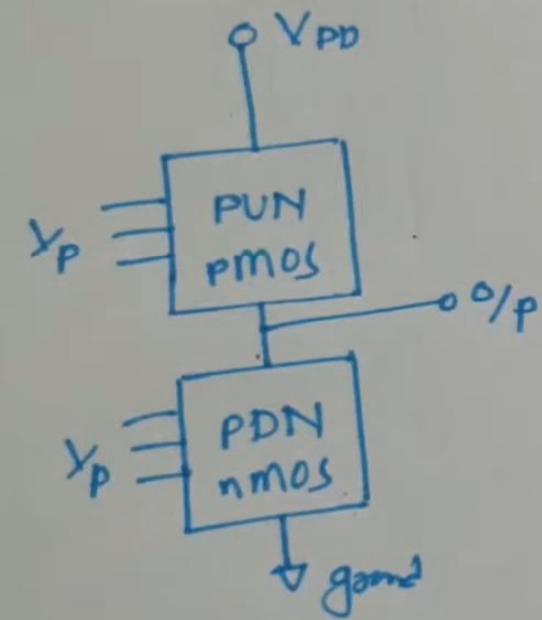
pMOS → Parallel

nMOS → Series

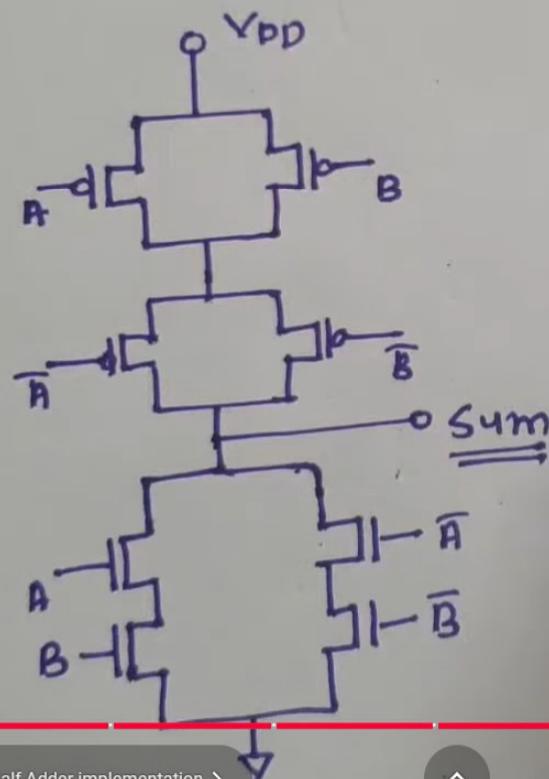
pMOS → Series

nMOS → Parallel

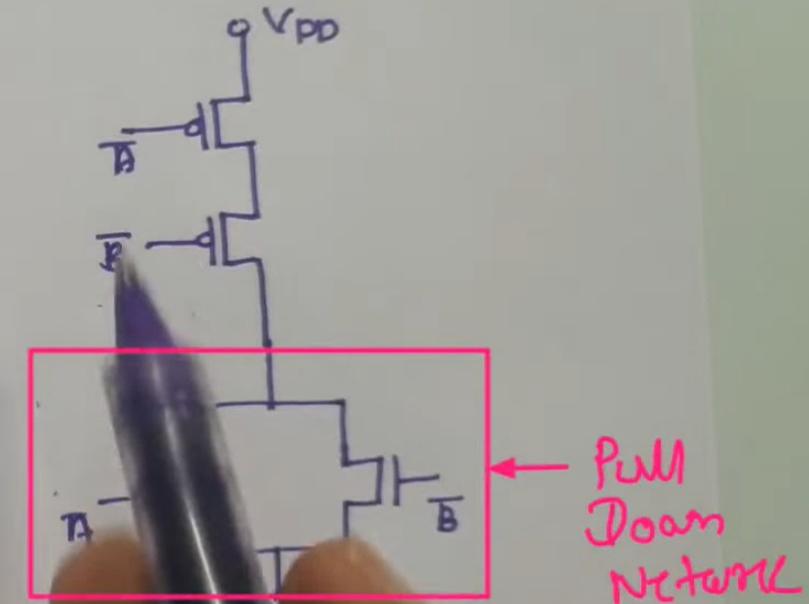
$$\neg \bar{A} + \bar{B}$$



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



$$\text{Carry} = \overline{\bar{A} + \bar{B}}$$

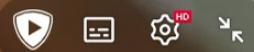


Pull  
Down  
Network



10:13 / 10:51

CMOS Half Adder implementation >



→  $\text{F}_{\text{OR}} (\cdot)$  Operation

pMOS → Parallel

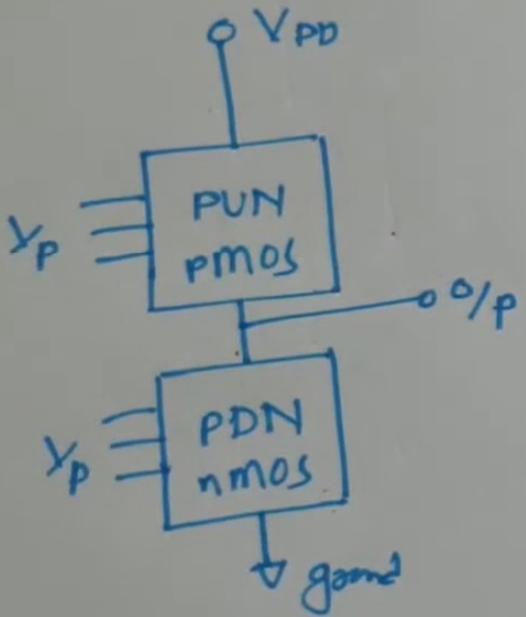
nMOS → Series

→  $\text{F}_{\text{OR}} (+)$  Operation

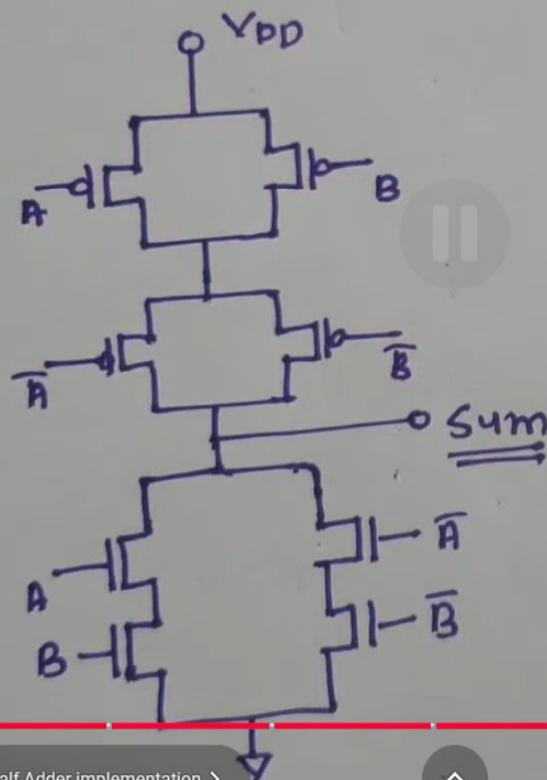
rMOS → Series

nmOS → Parallel

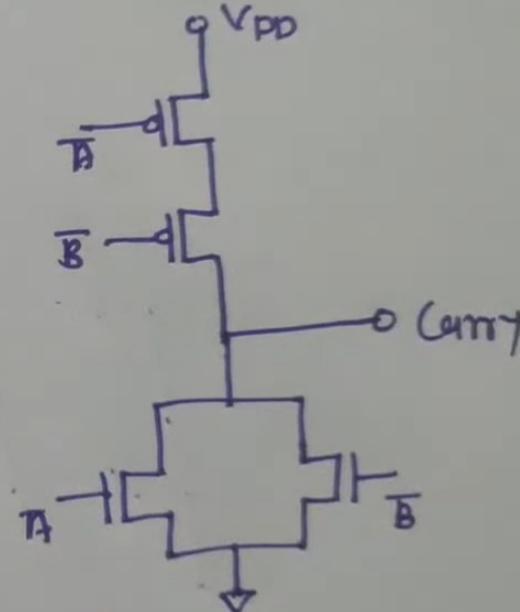
$$\neg \bar{A} + \bar{B}$$



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



$$\text{Carry} = \overline{\bar{A} + \bar{B}}$$



10:27 / 10:51

CMOS Half Adder implementation >



## Stick Diagram of CMOS NOR gate

→ Step 1

Boolean eq.  
 $y = \overline{A+B}$

→ Step 2

CMOS Circuit

→ Step 3

make stick diagram with colour code.

$$\rightarrow y = \overline{A+B}$$

→ For (+) operation

→ (+) operation

pMOS - Parallel

nMOS - S



2:02 / 12:22

Step - 2 - CMOS NOR Gate Circuit >



## CMOS NOR Gate Stick Diagram: Circuit, Design & Working

stick diagram with colour code.

$$\rightarrow Y = \overline{A + B}$$

$\rightarrow$  For (-) operation

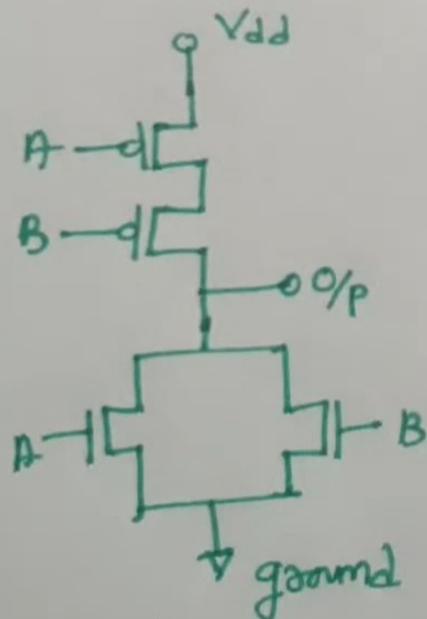
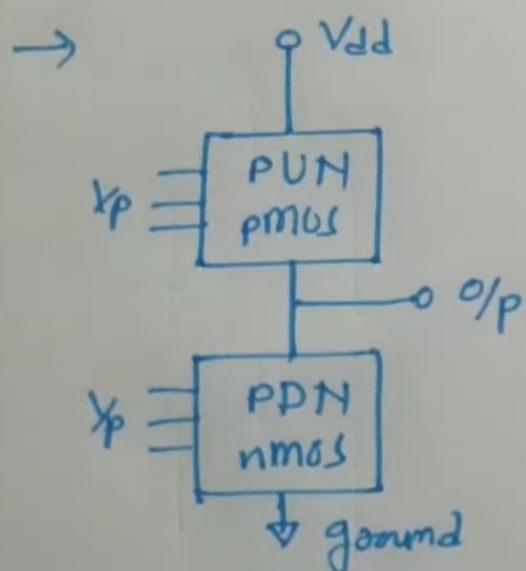
pMOS - Parallel

nMOS - Series

$\rightarrow$  (+) Operation

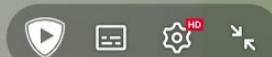
pMOS - Series

nMOS - Parallel



4:58 / 12:22

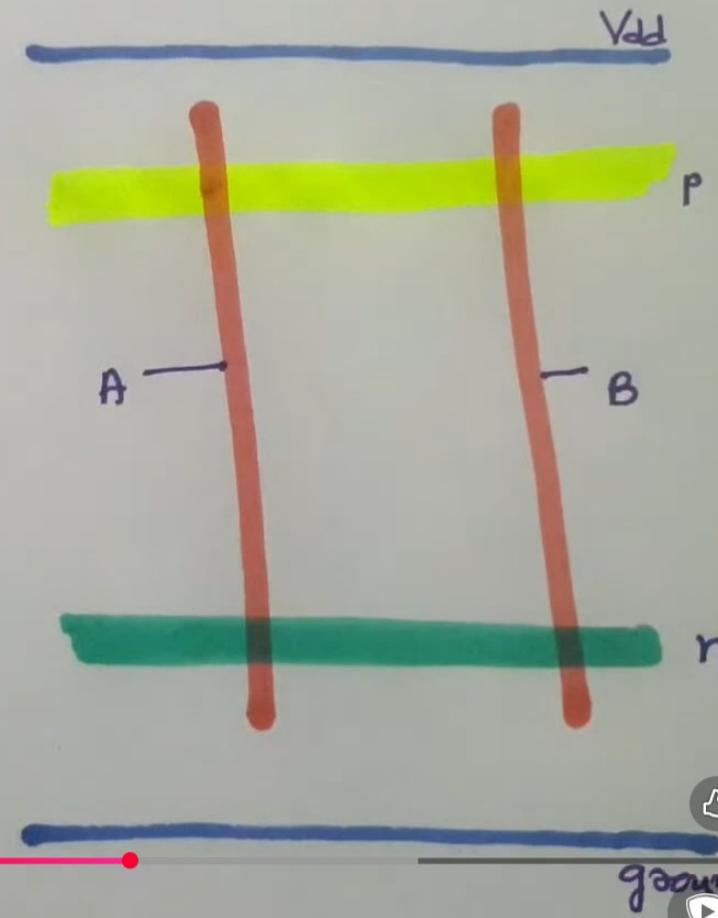
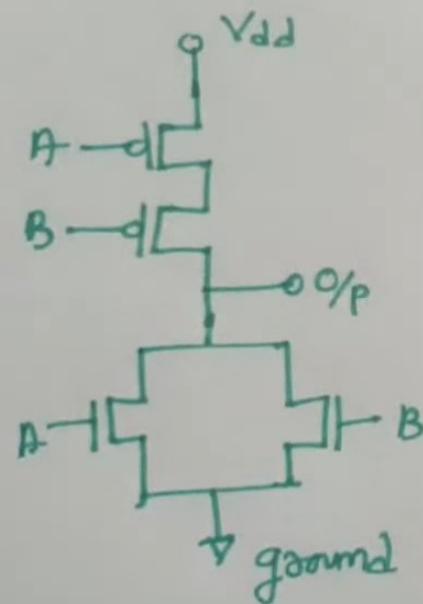
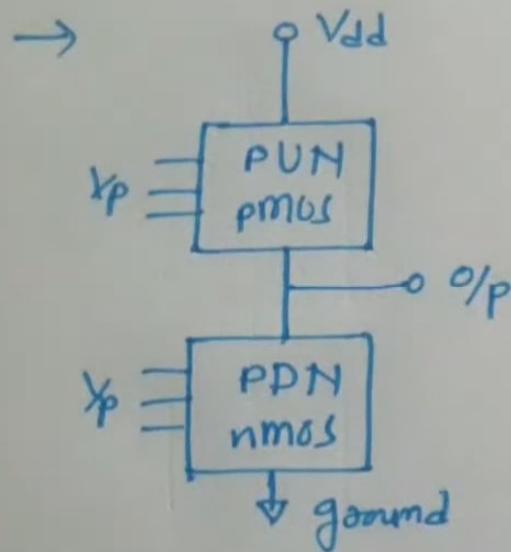
Step - 3 - Stick Diagram of CMOS NOR Gate &gt;



CMOS NOR Gate Stick Diagram: Circuit, Design & Working

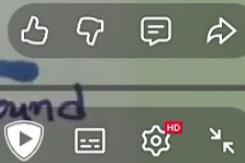
→ For (+) Operation  
pMOS - Parallel  
nMOS - Series

→ (+) Operation  
pMOS - Series  
nMOS - Parallel

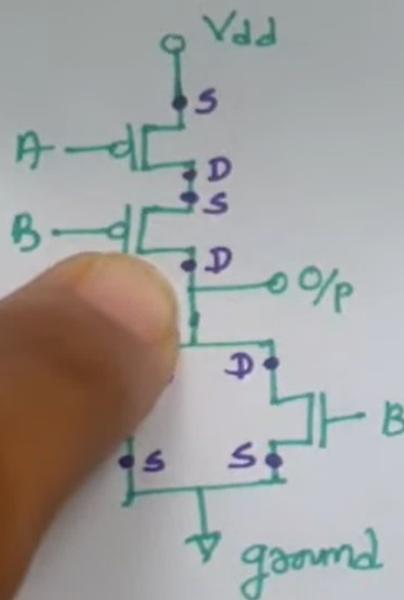
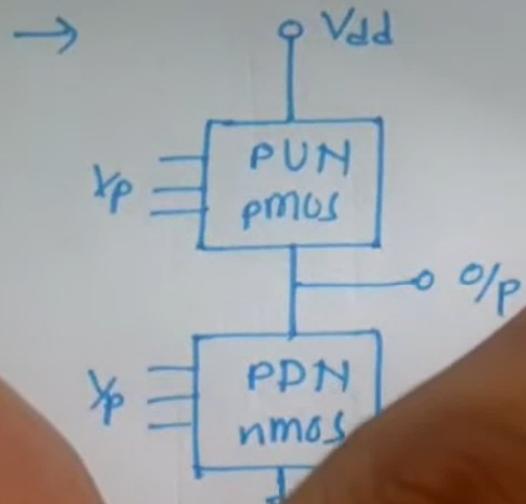


7:28 / 12:22

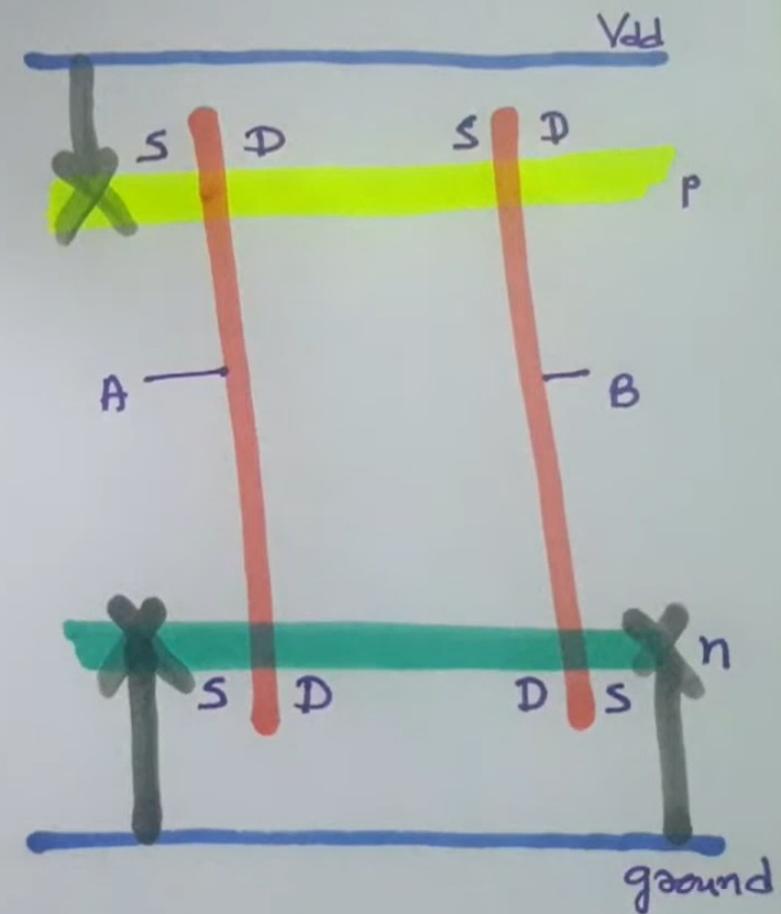
Step - 3 - Stick Diagram of CMOS NOR Gate >



→ For (-) operation  
pMOS - Parallel  
nMOS - Series



→ (+) Operation  
pMOS - Series  
nMOS - Parallel



## CMOS NOR Gate Stick Diagram: Circuit, Design & Working

→ For (+) Operation

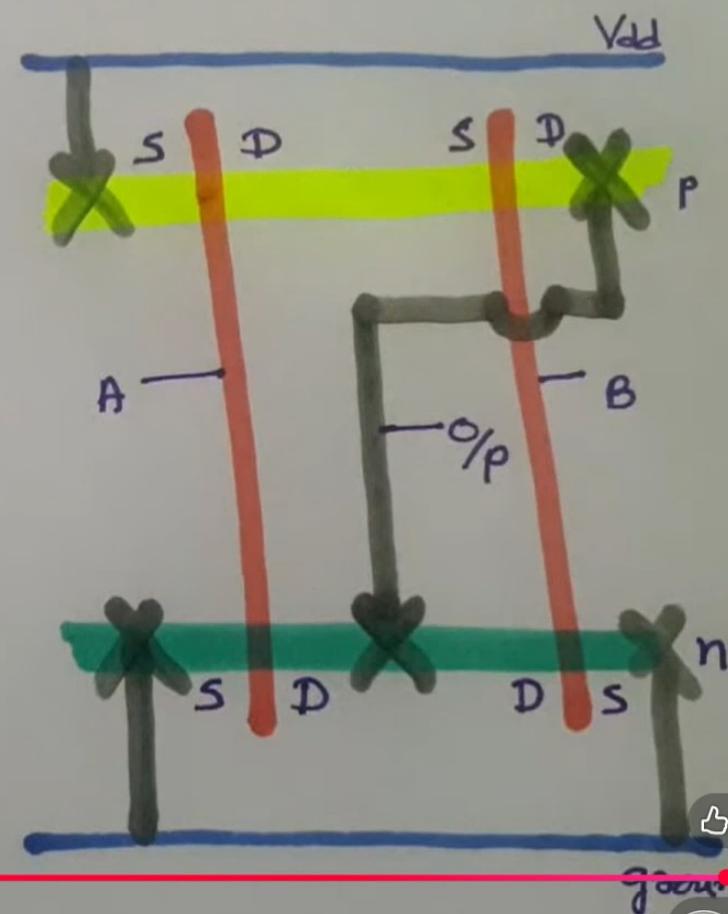
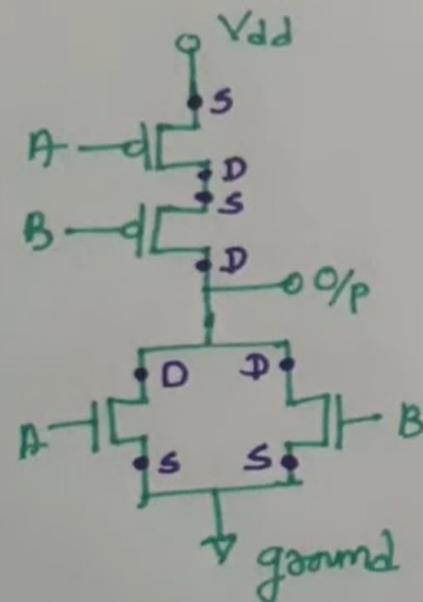
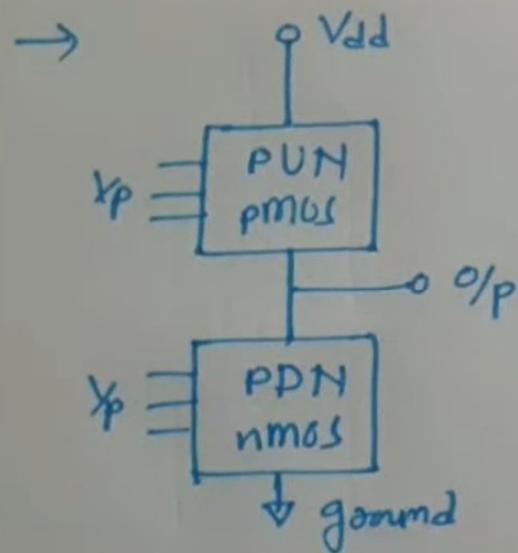
pMOS - Parallel

nMOS - Series

→ (+) Operation

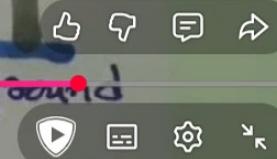
pMOS - Series

nMOS - Parallel



11:04 / 12:22

Step - 3 - Stick Diagram of CMOS NOR Gate >



## Stick Diagram of CMOS NAND gate

→ Step-1 Boolean eqn.  
 $y = \overline{(A \cdot B)}$

→ Step-2 CMOS Circuit

→ Step 3 Stick diagram with colour code.

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

→ (•) Opera

## CMOS NAND Gate Stick Diagram: Circuit, Design & Working

Step 3

stick diagram with colour code.

$$\rightarrow Y = \overline{A \cdot B}$$

$\rightarrow (\cdot)$  Operation

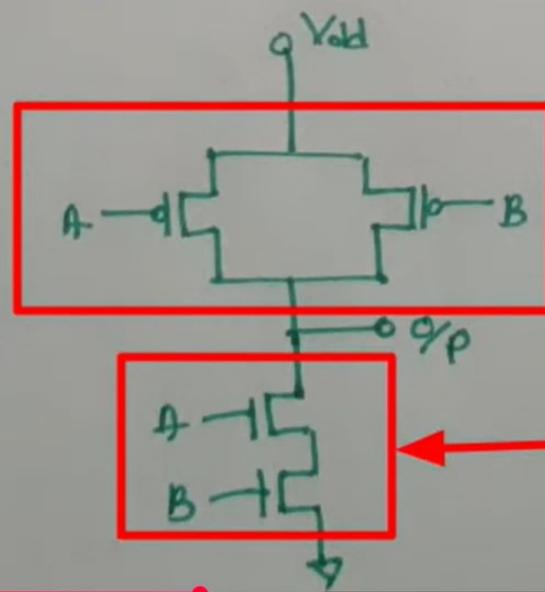
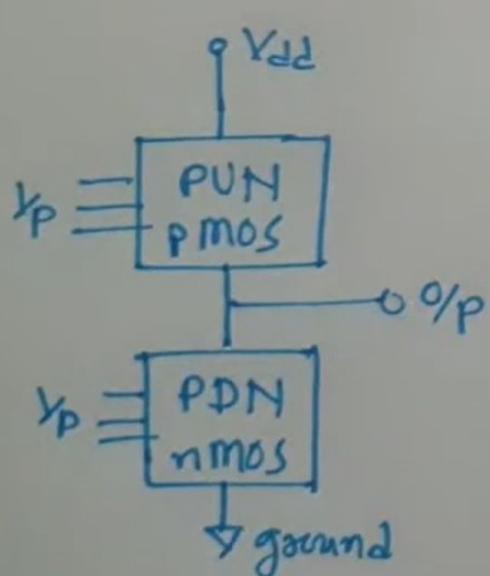
pmos  $\rightarrow$  Parallel

nmos  $\rightarrow$  Series

$\rightarrow (+)$  Operation

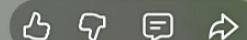
pmos  $\rightarrow$  Series

nmos  $\rightarrow$  Parallel



Pull Up N/w

Pull Down N/w

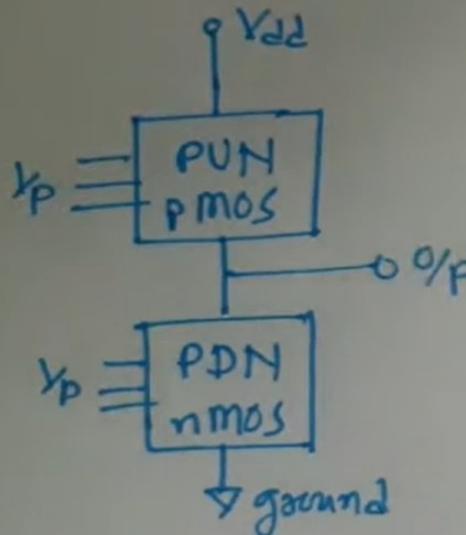


4:18 / 10:35

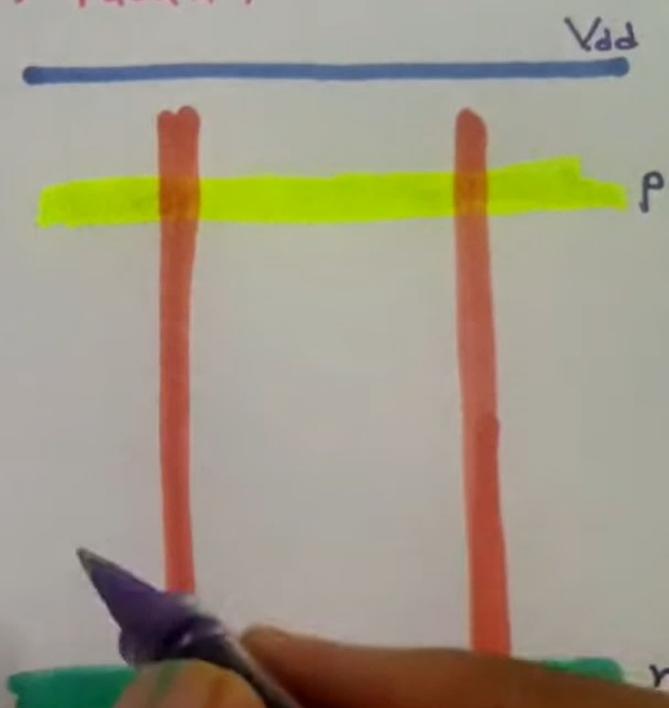
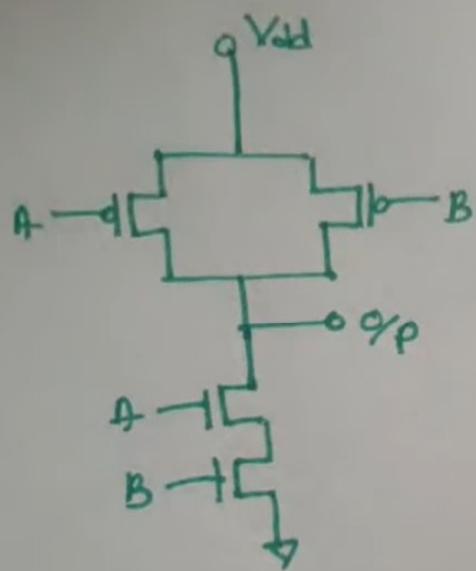
Step - 2 - CMOS NAND Gate Circuit



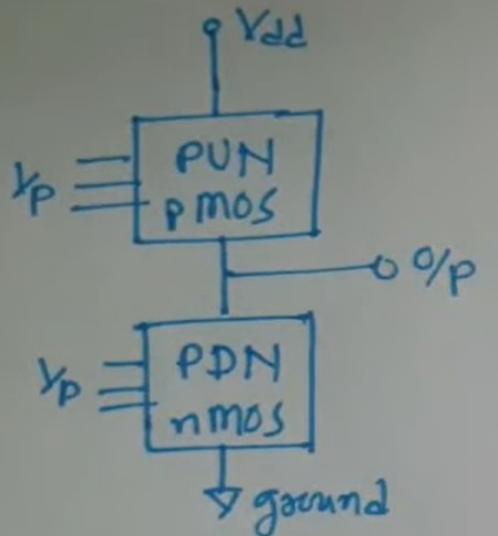
→ (-) Operation  
PMOS → Parallel  
NMOS → Series



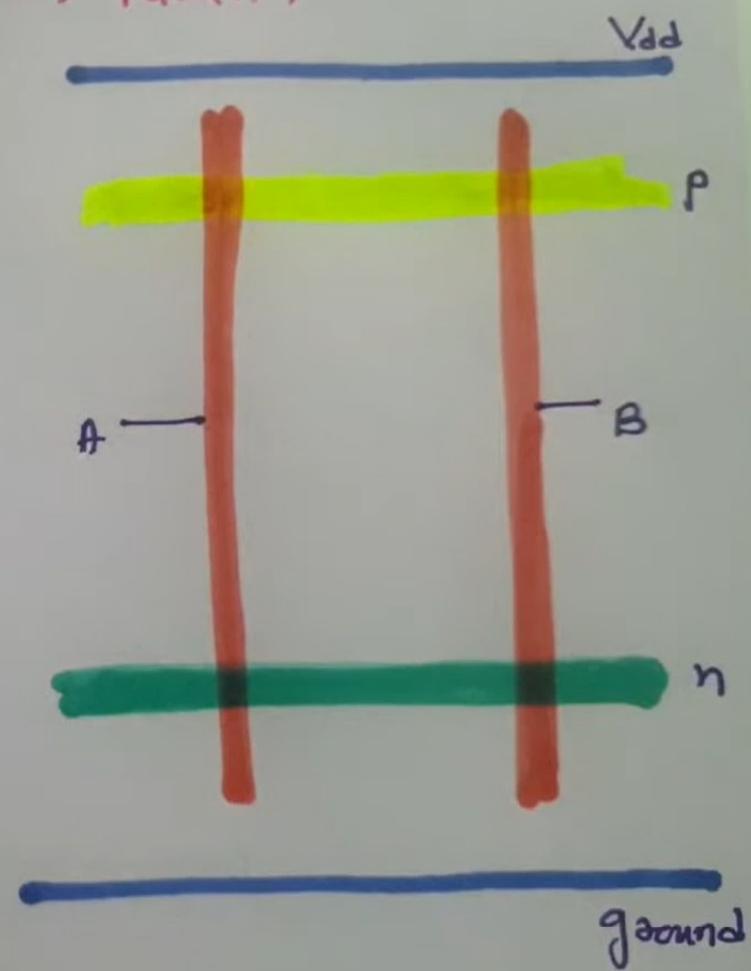
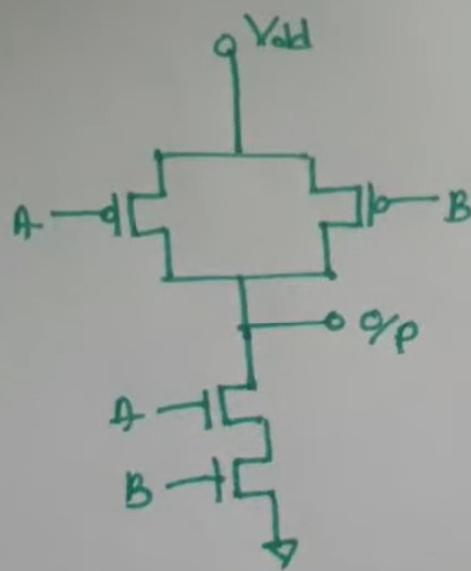
→ (+) Operation  
PMOS → Series  
NMOS → Parallel



→ (-) Operation  
PMOS → Parallel  
NMOS → Series

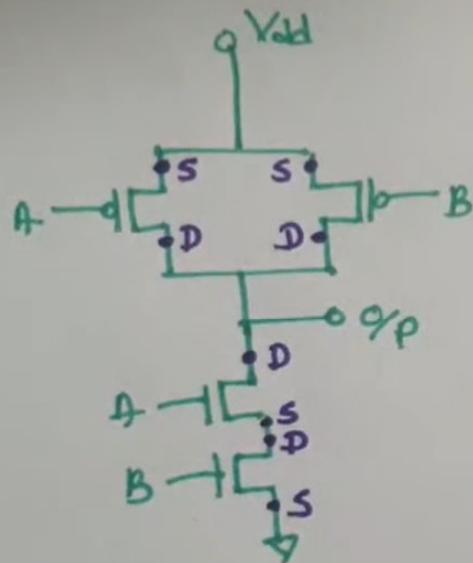
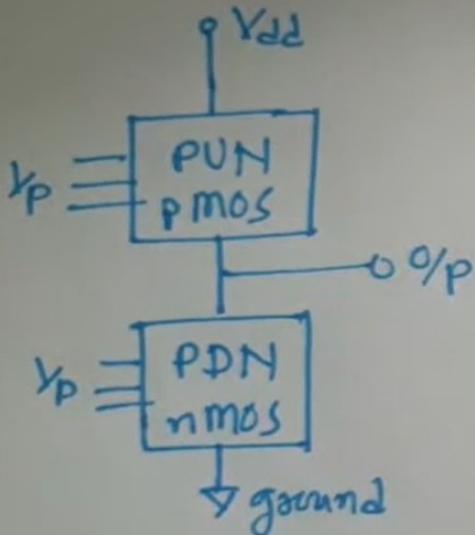


→ (+) Operation  
PMOS → Series  
NMOS → Parallel

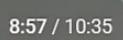
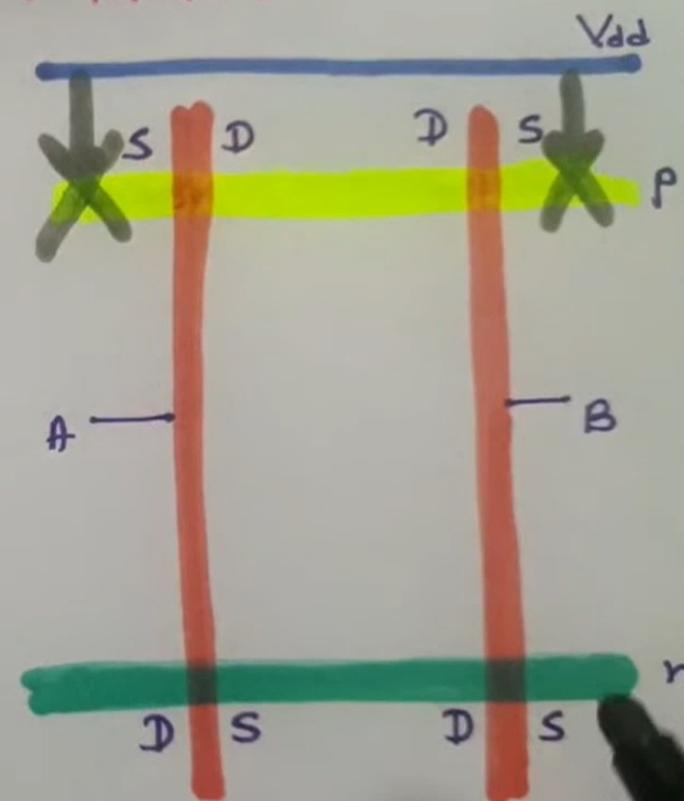


## CMOS NAND Gate Stick Diagram: Circuit, Design & Working

→ (+) Operation  
pmos → Parallel  
nmos → Series

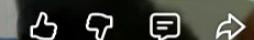


→ (-) Operation  
pmos → Series  
nmos → Parallel



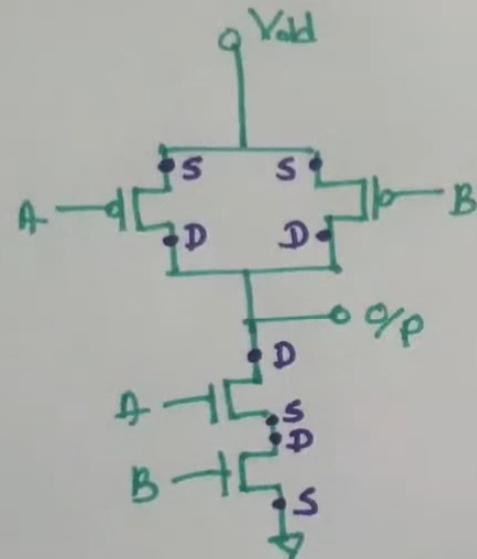
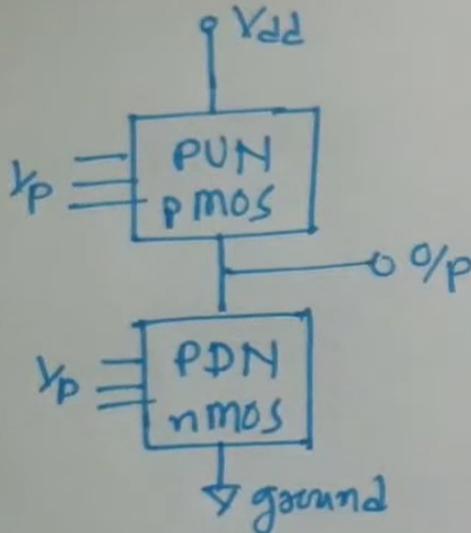
8:57 / 10:35

Step - 3 - Stick Diagram of CMOS NAND Gate &gt;

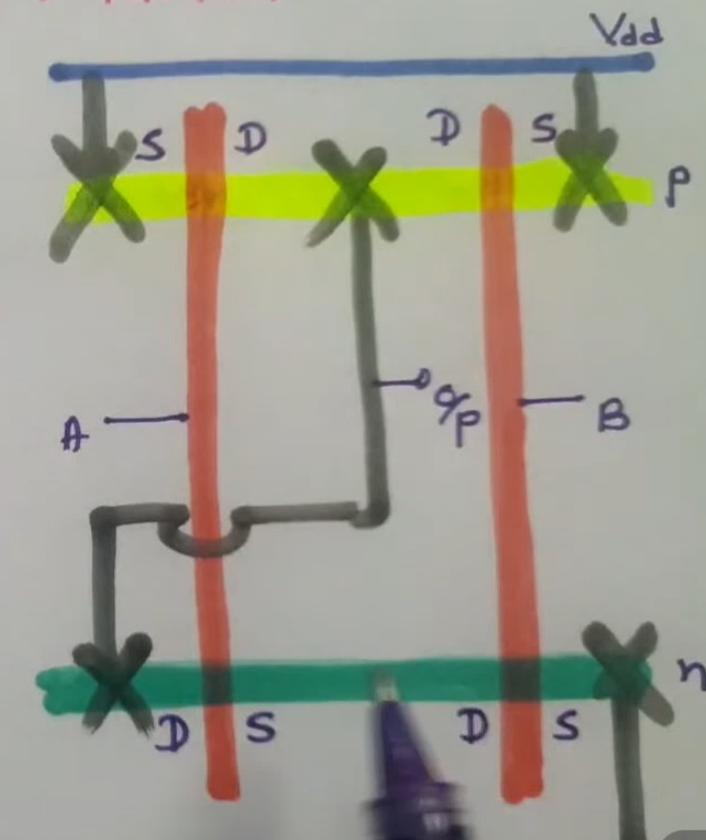


## CMOS NAND Gate Stick Diagram: Circuit, Design & Working

→ (-) Operation  
 pmos → Parallel  
 nmos → Series

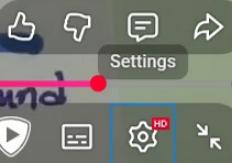


→ (+) Operation  
 pmos → Series  
 nmos → Parallel



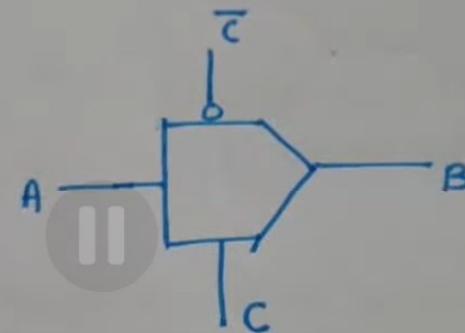
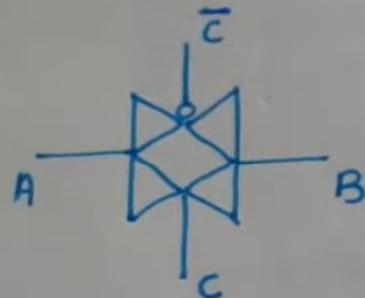
9:49 / 10:35

Step - 3 - Stick Diagram of CMOS NAND Gate >

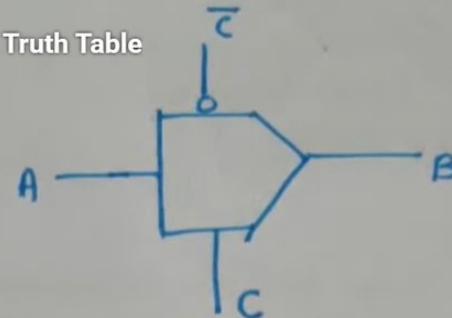
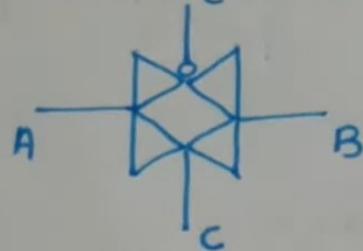


CMOS Transmission gate and CMOS Transmission gate as Tri state Buffer.

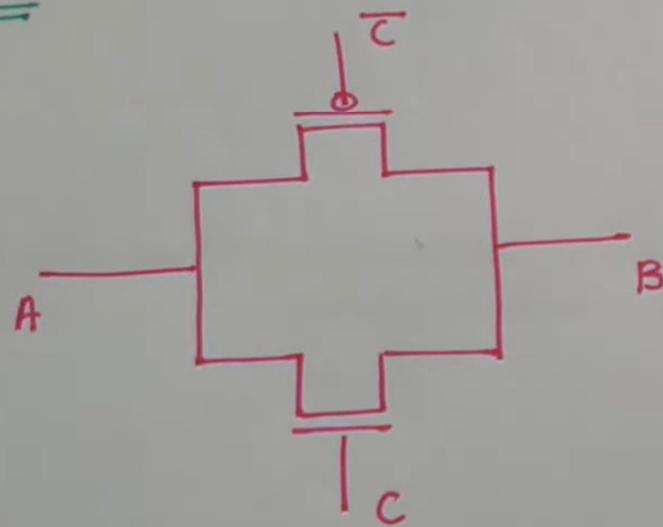
→ Symbols



## CMOS Transmission Gate Explained: Symbols, Circuit Design, Working & Truth Table



→ Circuit



- It has one pMOS and one nMOS.
- Gate is complementary to two transistors.



◀

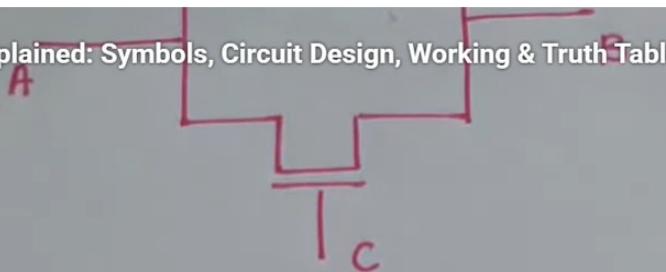
▶

2:39 / 8:06

Circuit of CMOS Transmission Gate >

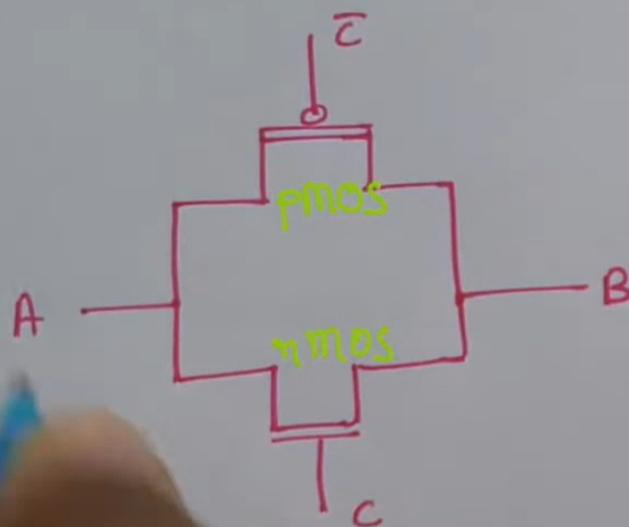


## CMOS Transmission Gate Explained: Symbols, Circuit Design, Working & Truth Table



- Gate is complementary to two transistors.

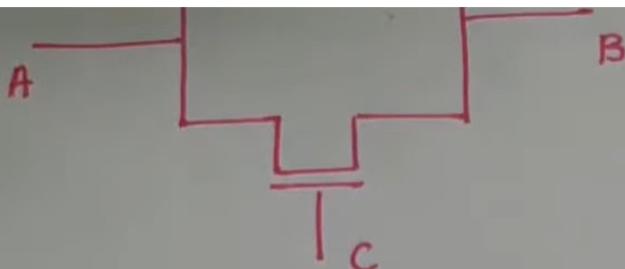
→ Working & Truth Table



3:14 / 8:06

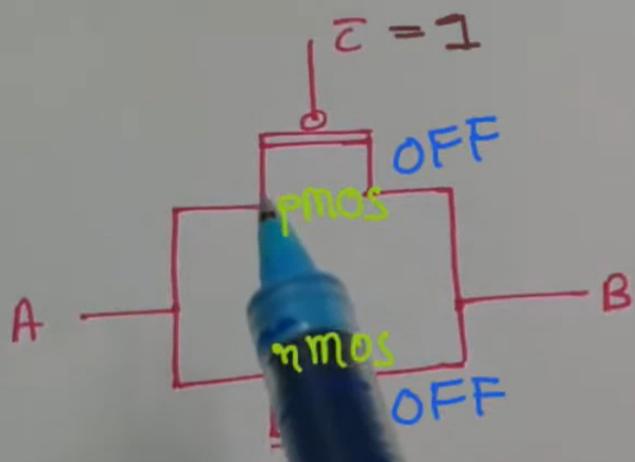
Working &amp; Truth Table of CMOS Transmission Gate &gt;





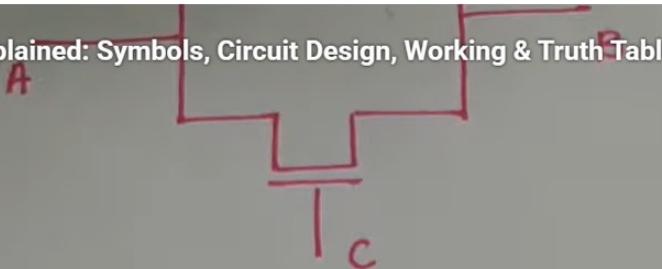
- Gate is complementary to two transistors.

→ Working & Truth Table



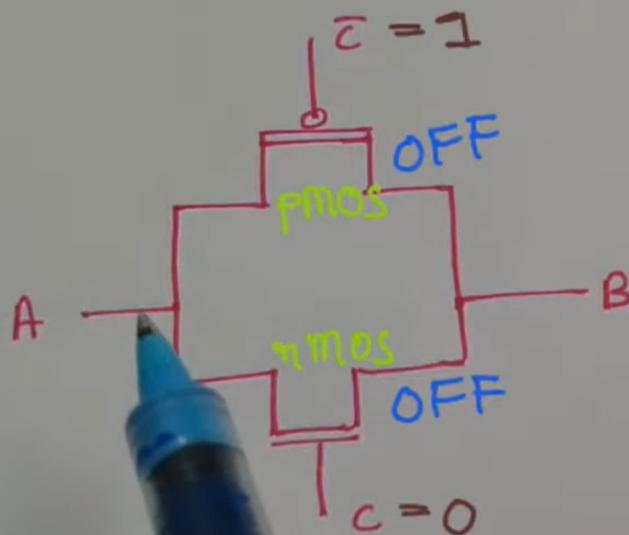
Inputs		Output
C	A	B
0	0	
0	1	
1	0	
1	1	

## CMOS Transmission Gate Explained: Symbols, Circuit Design, Working & Truth Table



- Gate is complementary to two transistors.

→ Working & Truth Table

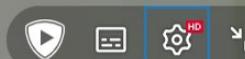


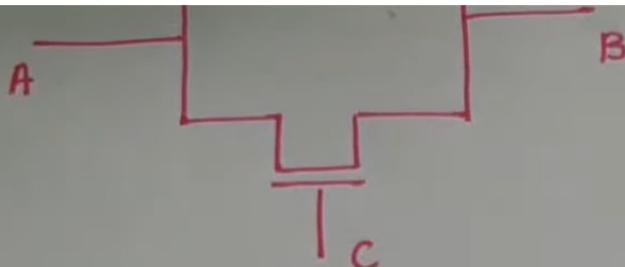
Inputs		Output
C	A	B
0	0	1
0	1	0
1	0	0
1	1	1



4:31 / 8:06

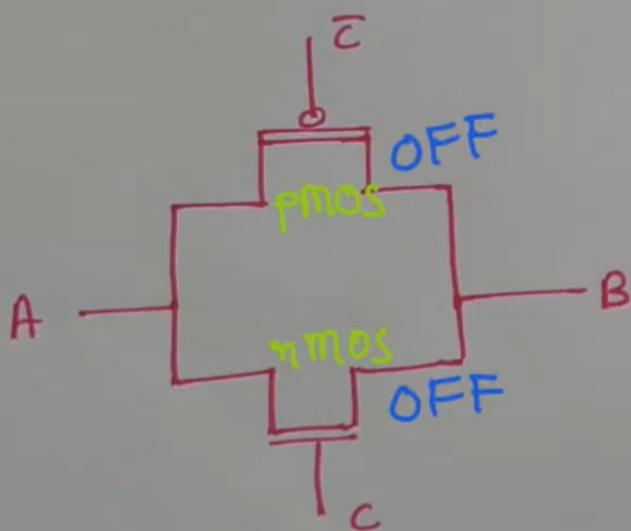
Working & Truth Table of CMOS Transmission Gate >





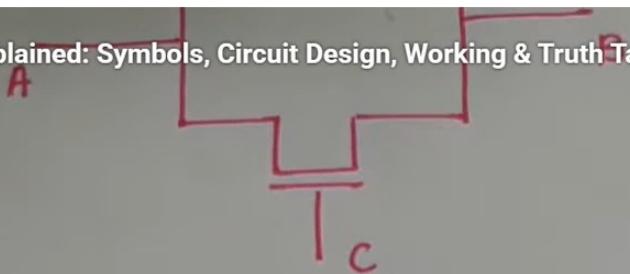
- Gate is complementary to two transistors.

→ Working & Truth Table



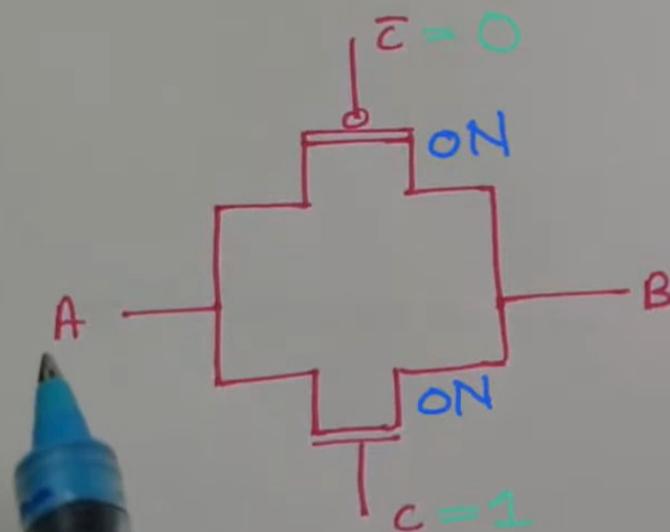
Inputs		Output
C	A	B
0	0	H.I.
0	1	H.I.
1	0	
1	1	

## CMOS Transmission Gate Explained: Symbols, Circuit Design, Working & Truth Table



- Gate is complementary to two transistors.

→ Working & Truth Table



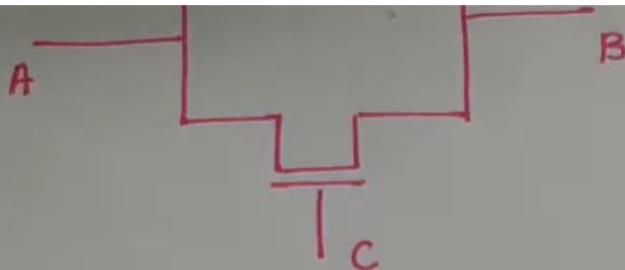
Inputs		Output
C	A	B
0	0	H.I.
0	1	H.I.
1	0	
1	1	



5:14 / 8:06

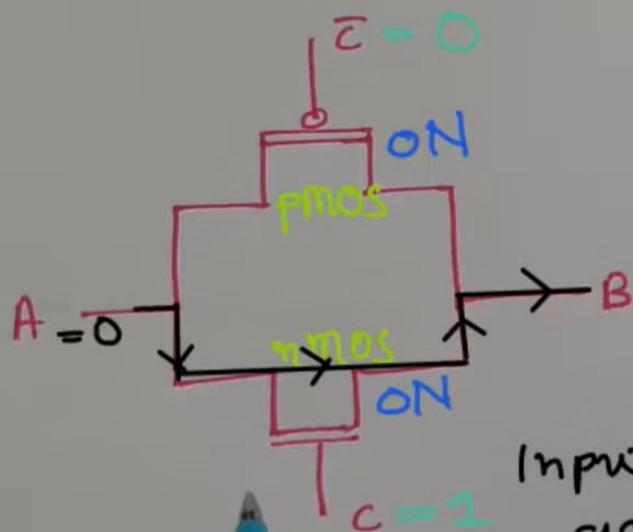
Working & Truth Table of CMOS Transmission Gate >





- Gate is complementary to two transistors.

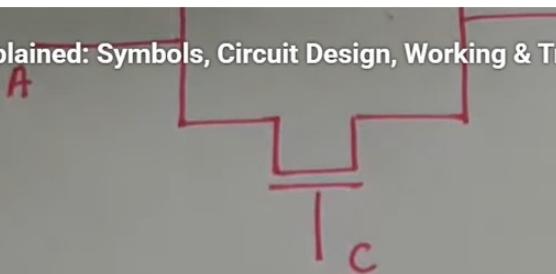
→ Working & Truth Table



Inputs		Output
C	A	B
0	0	H.I.
0	1	H.I.
1	0	0
1	1	

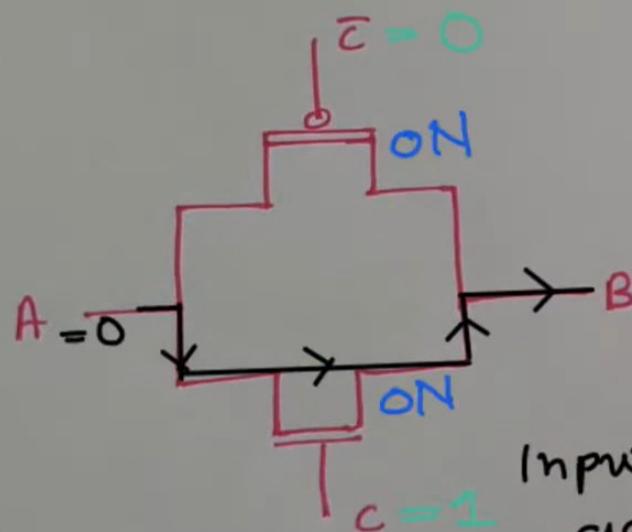
Input  $A = 0$ , will pass through nMOS as nMOS is good to pass logic 0.

## CMOS Transmission Gate Explained: Symbols, Circuit Design, Working & Truth Table



- Gate is complementary to two transistors.

→ Working & Truth Table



Inputs		Output
C	A	B
0	0	H.I.
0	1	H.I.
1	0	0
1	1	

Input  $A = 0$ , will pass through nMOS as nMOS is good to pass logic 0.

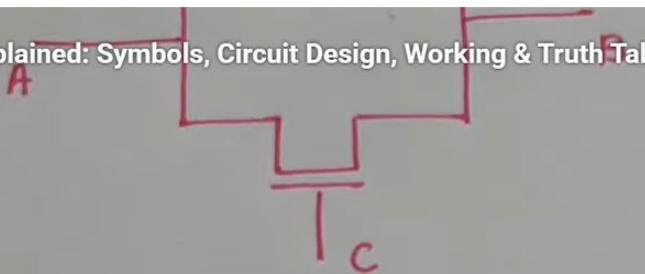


5:46 / 8:06

Working &amp; Truth Table of CMOS Transmission Gate &gt;

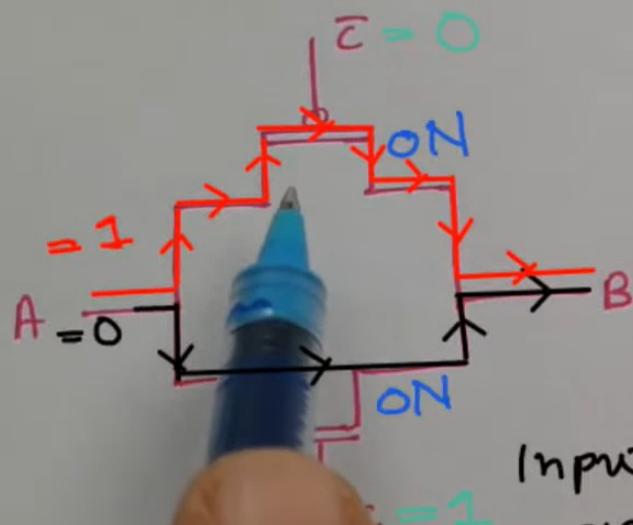


## CMOS Transmission Gate Explained: Symbols, Circuit Design, Working & Truth Table



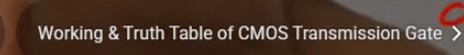
- Gate is complementary to two transistors.

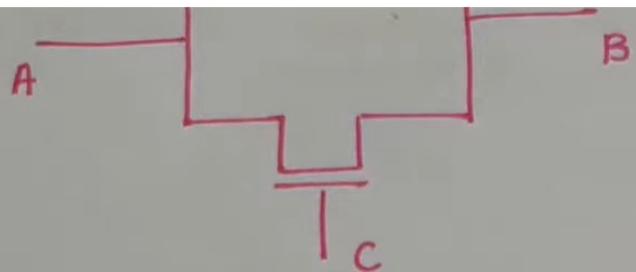
→ Working & Truth Table



Inputs		Output
C	A	B
0	0	H.I.
0	1	H.I.
1	0	0
1	1	

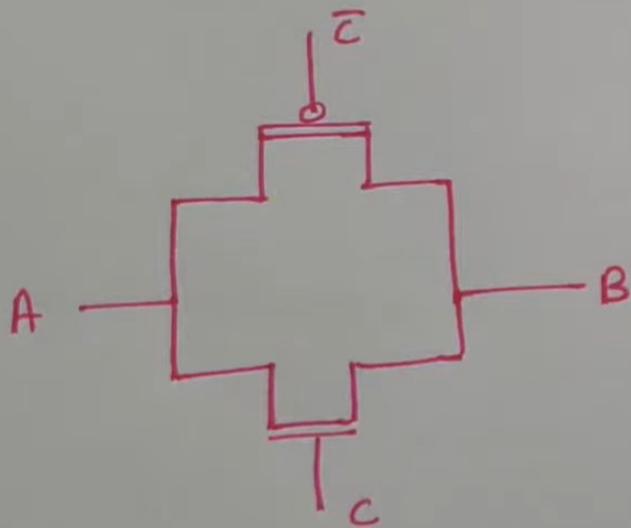
Input  $A = 0$ , will pass through nMOS  
as nMOS is good to pass logic 0.  
Input  $A = 1$ , will pass through pMOS  
as pMOS is good to pass logic 1.





- Gate is complementary to two transistors.

→ Working & Truth Table



Inputs		Output
C	A	B
0	0	H.I.
0	1	H.I.
1	0	0
1	1	1

Multiplexer implementation using Transmission Gate.

→ Here I will explain 4x1 MUX using transmission gate

A	B	f
0	0	I0
0	1	I1
1	0	I2
1	1	I3

→ Boolean function

$$f = \bar{A}\bar{B} \cdot I_0 + \bar{A}B \cdot I_1 + A\bar{B} \cdot I_2 + AB \cdot I_3$$
**CMOS Circuit**

Engineering Funda - 17/30



Explained: Symbols, Circuit...

Engineering Funda



4 to 1 Multiplexer Implementation using...

Engineering Funda



4 to 1 Multiplexer Implementation using...

Engineering Funda



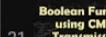
D Latch Implementation using Transmission Gate | CMOS...

Engineering Funda



Boolean Function Implementation using...

Engineering Funda



Boolean Function Implementation using...

Engineering Funda



Dynamic CMOS ( Basics, ...)

Engineering Funda

**4 to 1 Multiplexer Implementation using Transmission Gates | VLSI by Engineering Funda**
**Engineering Funda** 600k subscribers

Join

Subscribe

1.9k



Share

Download

All

From the series

From Engineering Funda



152k views 5 years ago INDIA



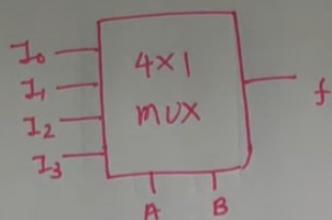
Search



ENG IN

03:54 PM  
24-11-2025

→ Here I will explain 4x1 MUX using transmission gate

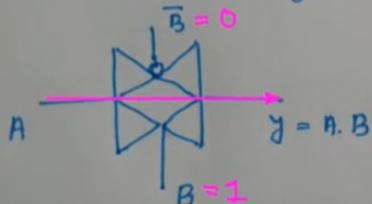


A	B	f
0	0	I0
0	1	I1
1	0	I2
1	1	I3

→ Boolean function

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

→ Transmission gate



A	B	Y
0	0	
0	1	
1	0	
1	1	

## CMOS Circuit

Engineering Funda - 17/30



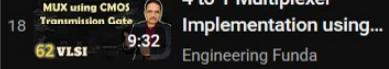
Explained: Symbols, Circuit...

Engineering Funda



4 to 1 Multiplexer Implementation using...

Engineering Funda



4 to 1 Multiplexer Implementation using...

Engineering Funda



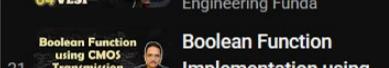
D Latch Implementation using Transmission Gate | CMOS...

Engineering Funda



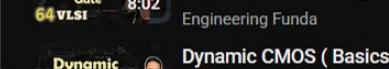
Boolean Function Implementation using...

Engineering Funda



Boolean Function Implementation using...

Engineering Funda



Dynamic CMOS ( Basics, ... )

Engineering Funda

## 4 to 1 Multiplexer Implementation using Transmission Gates | VLSI by Engineering Funda

Engineering Funda 600k subscribers

Join

Subscribe

1.9k



Share

Download

All

From the series

From Engineering Funda

152k views 5 years ago INDIA



Search

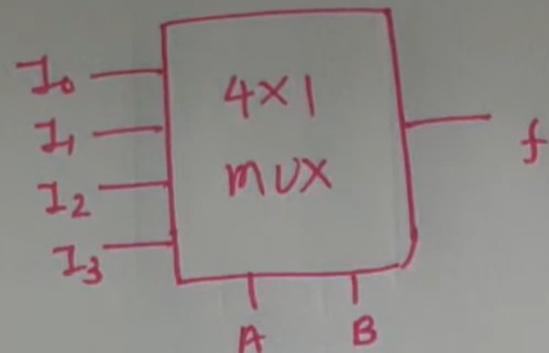


ENG  
IN



04:07 PM  
24-11-2025

## 4 to 1 Multiplexer Implementation using Transmission Gates | VLSI by Engineering Funda

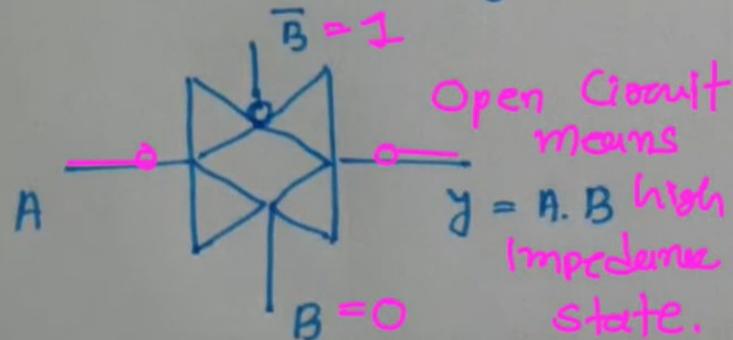


A	B	f
0	0	I <sub>0</sub>
0	1	I <sub>1</sub>
1	0	I <sub>2</sub>
1	1	I <sub>3</sub>

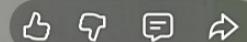
→ Boolean function

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

→ Transmission gate



A	B	Y
0	0	H.I
0	1	0
1	0	H.I
1	1	1

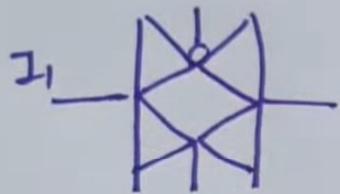
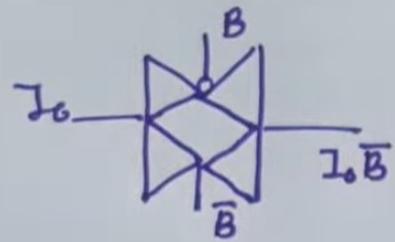


4:43 / 9:31

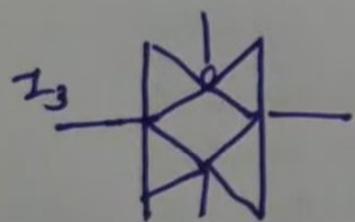
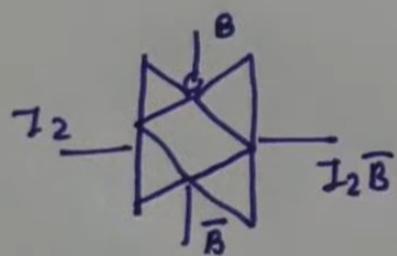
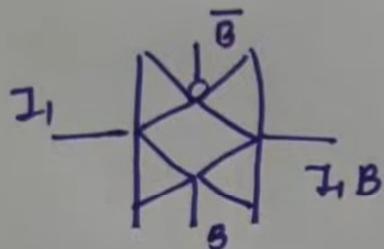
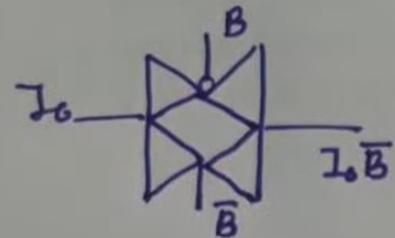
Truth Table & Working of CMOS Transmission Gate >



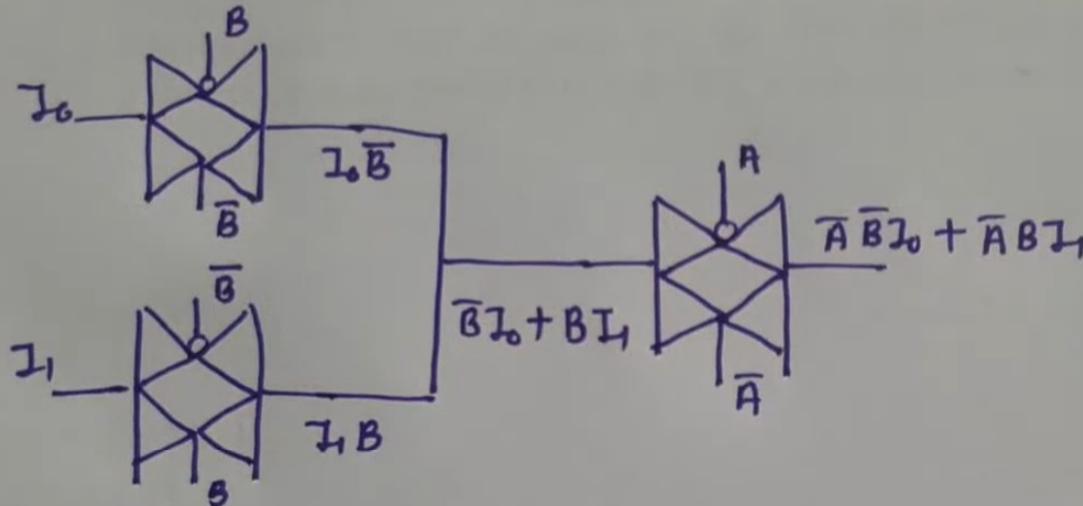
$$\rightarrow f = \bar{A} \cdot \bar{B} \cdot I_0 + \bar{A} \cdot B \cdot I_1 + A \cdot \bar{B} \cdot I_2 + A \cdot B \cdot I_3$$



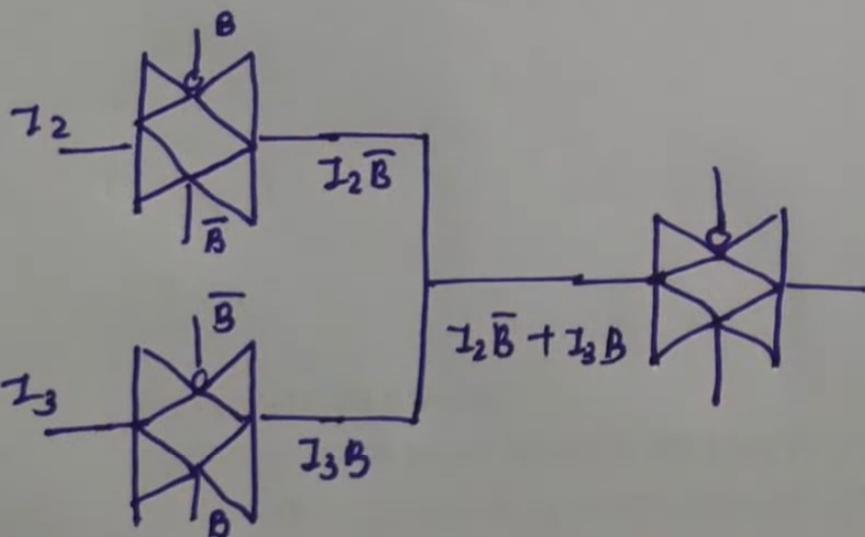
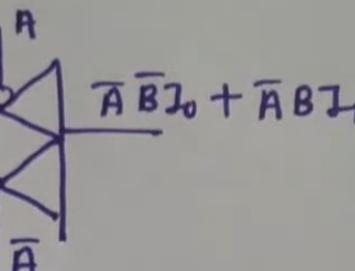
$$\rightarrow f = \bar{A} \cdot \bar{B} \cdot I_0 + \bar{A} \cdot B \cdot I_1 + A \cdot \bar{B} \cdot I_2 + A \cdot B \cdot I_3$$



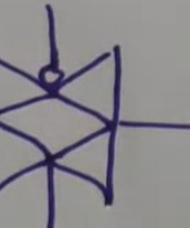
$$\rightarrow f = \bar{A} \cdot \bar{B} \cdot I_0 + \bar{A} \cdot B \cdot I_1 + A \cdot \bar{B} \cdot I_2 + A \cdot B \cdot I_3$$



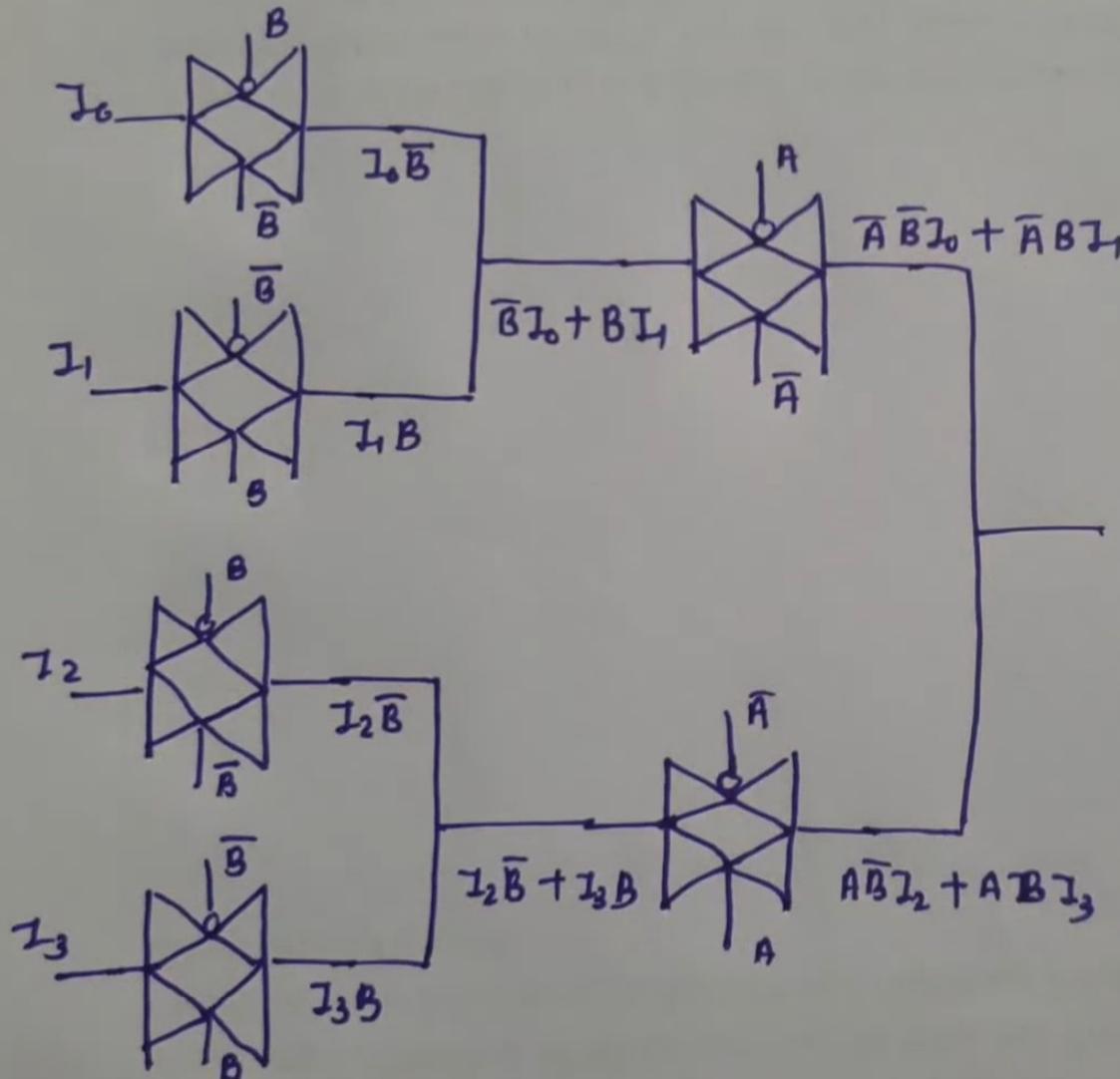
$$\bar{B}I_0 + BI_1$$



$$I_1B + I_2B$$

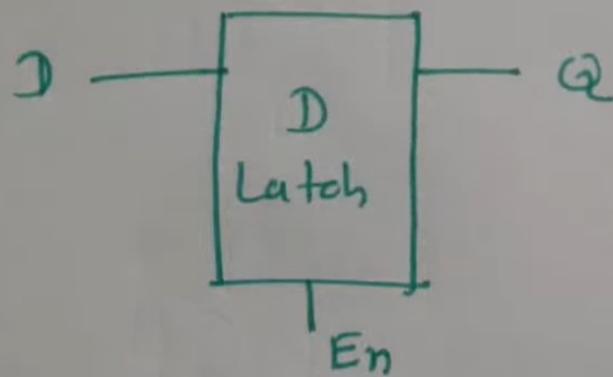


$$\rightarrow f = \bar{A} \cdot \bar{B} \cdot I_0 + \bar{A} \cdot B \cdot I_1 + A \cdot \bar{B} \cdot I_2 + A \cdot B \cdot I_3$$



## D Latch using Transmission Gate

- In D Latch we don't have clock input and with D Flip Flop we give clock input.



$\rightarrow E_n = 1, Q = D$

$\rightarrow E_n = 0, Q = M_{term}$

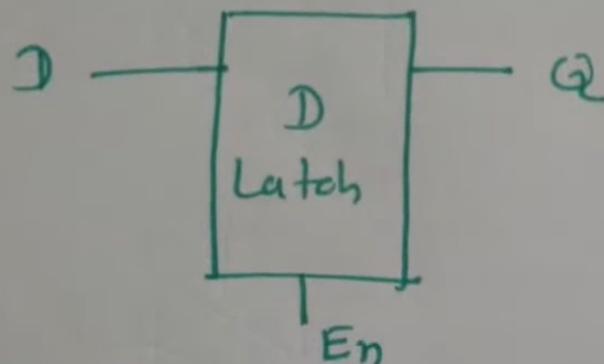


1:31 / 9:02

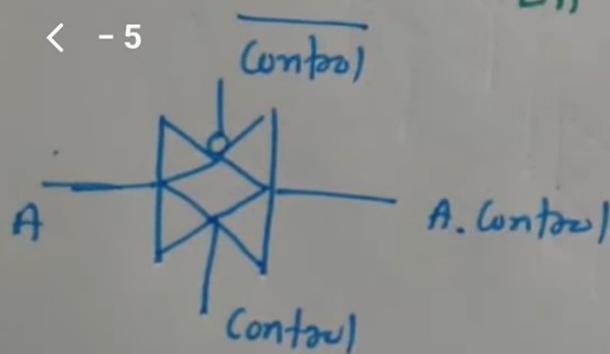
Symbol of CMOS Transmission Gate >



In D Latch we have D Input and with  
D Flip flop we give clock input.



$$\rightarrow En = 1, Q = D$$
$$\rightarrow En = 0, Q = \text{High}$$



Control	A	Y
0	0	H.1.
0	1	H.1.
1	0	0
1	1	1

$$Y = A \cdot \text{Control}$$

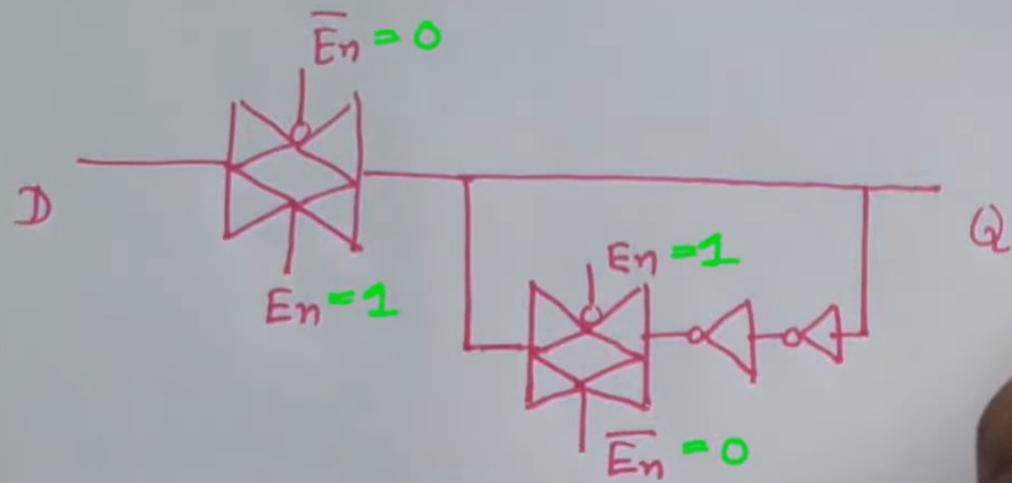


3:40 / 9:02

Truth Table & Working of CMOS Transmission Gate >

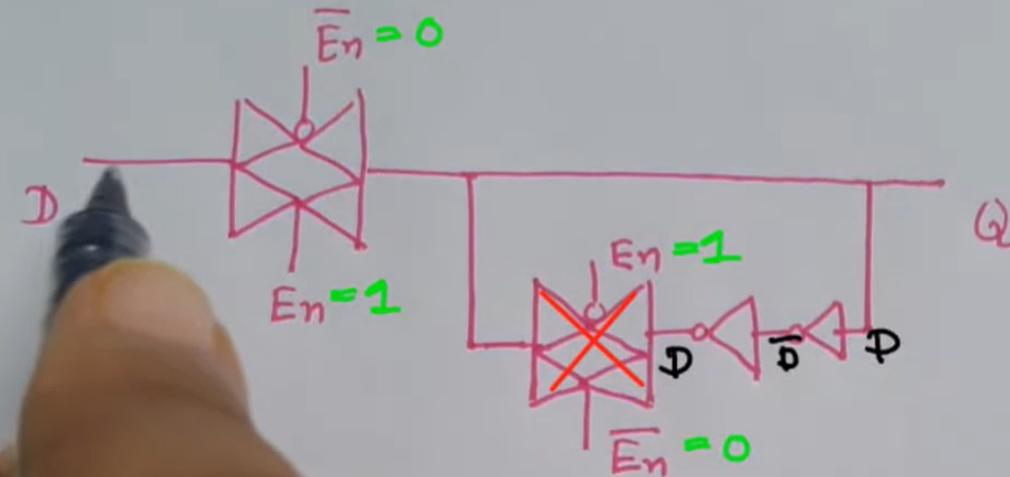


' Control



$\rightarrow E_n = 1, Q = D$

## D Latch Implementation using Transmission Gate | CMOS Transmission Gate | VLSI by Engineer...



$$\rightarrow E_n = 1, Q = D$$

$$\text{Inv } 1 = \bar{D}$$

$$\text{Inv } 2 = D$$

$$\text{T.G.2} = H \cdot I$$

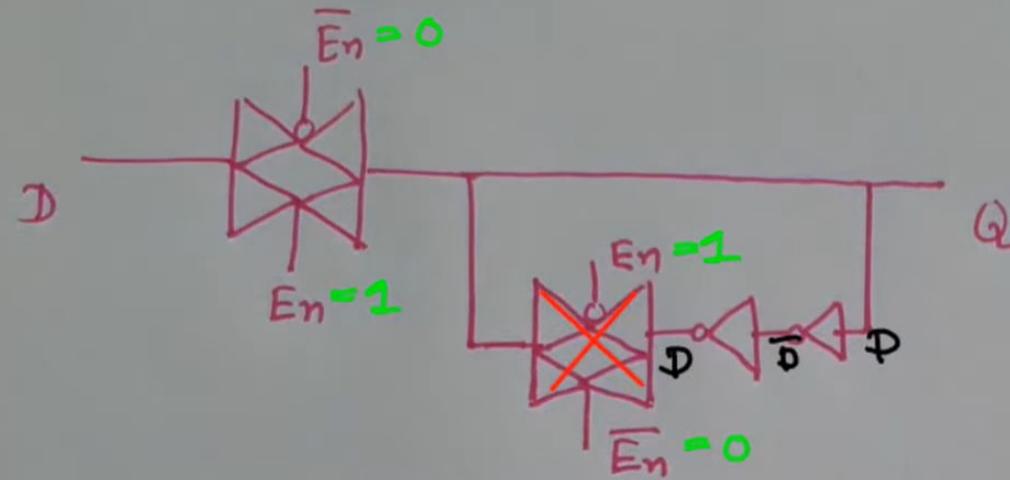


6:47 / 9:02

D Latch Implementation using Transmission Gates &gt;



## D Latch Implementation using Transmission Gate | CMOS Transmission Gate | VLSI by Engineer...

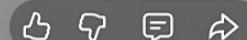


$\rightarrow E_n = 1, [Q = D]$

$$\text{Inv 1} = \overline{D}$$

$$\text{Inv 2} = D$$

$$\text{T.G.2} = H.1$$

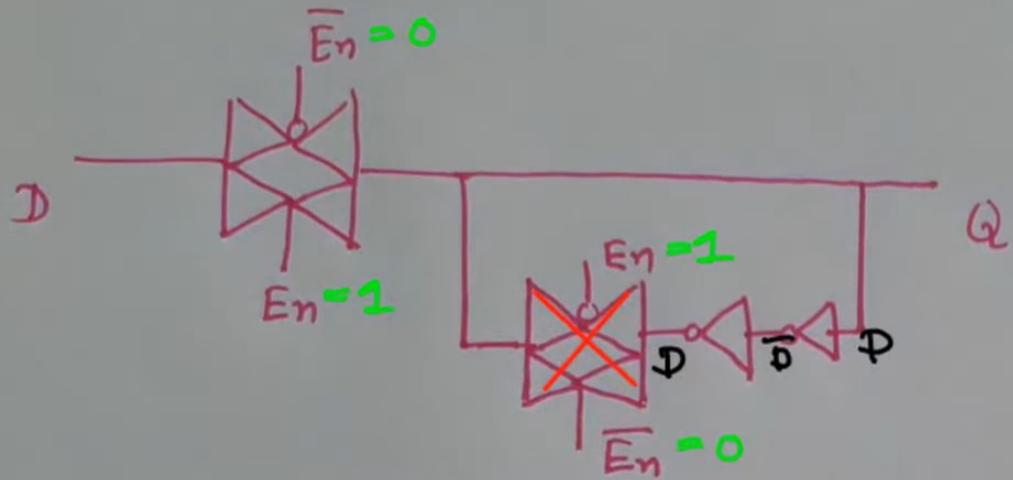


6:53 / 9:02

D Latch Implementation using Transmission Gates >



' Contau1



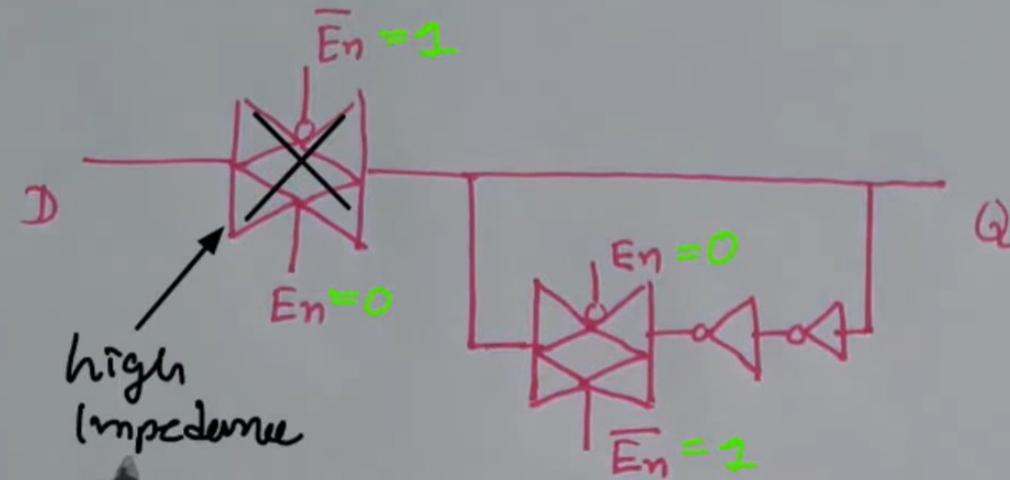
$$\rightarrow E_n = 1, [Q = D]$$

$$\text{Inv 1} = \overline{D}$$

$$\text{Inv 2} = D$$

$$\text{T.G.2} = \text{H.I}$$

## D Latch Implementation using Transmission Gate | CMOS Transmission Gate | VLSI by Engineer...



$$\rightarrow E_n = 1, \boxed{Q = D}$$

$$\text{Inv } 1 = \overline{D}$$

$$\text{Inv } 2 = D$$

$$\text{T.G.2} = H.1$$

$$\rightarrow E_n = 0, \\ \text{T.G.1} = H.1.$$

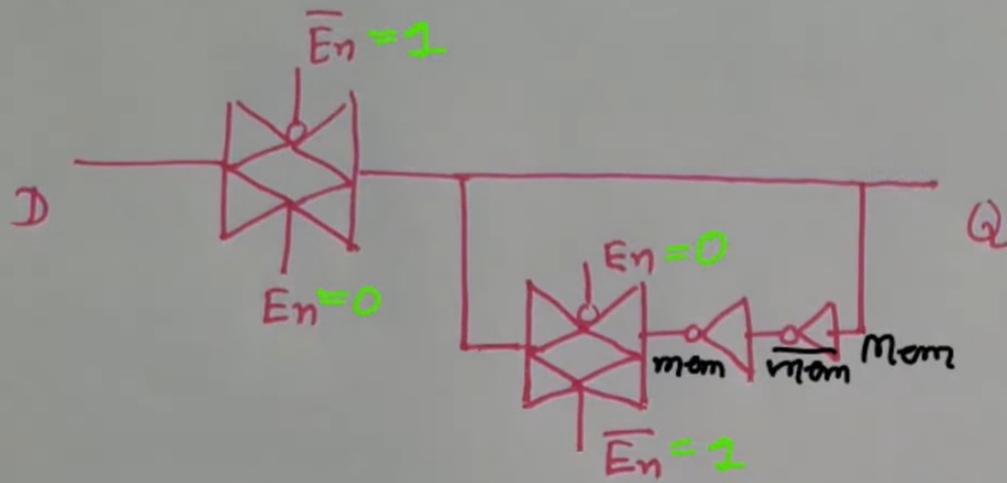


7:25 / 9:02

D Latch Implementation using Transmission Gates >



## D Latch Implementation using Transmission Gate | CMOS Transmission Gate | VLSI by Engineer...



$\rightarrow E_n = 1, [Q = D]$

$$\text{Inv } 1 = \overline{D}$$

$$\text{Inv } 2 = D$$

$$\text{T.G.2} = H.I.$$

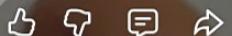
$\rightarrow E_n = 0,$

$$\text{T.G.1} = H.I.$$

$$\text{Inv } 1 = \overline{Mem}$$

$$\text{Inv } 2 = Mem$$

$$\text{T.G.2} = Mem, [Q = Mem]$$



8:30 / 9:02

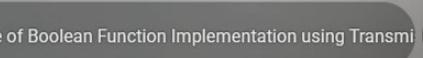
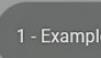
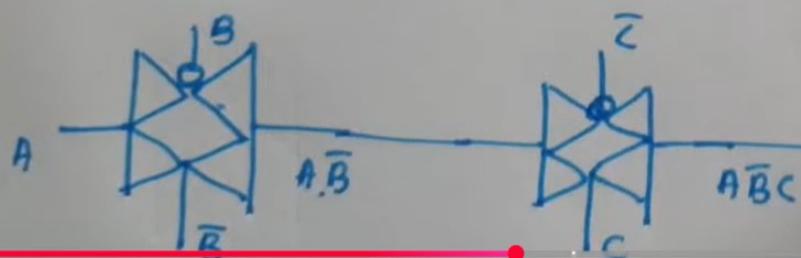
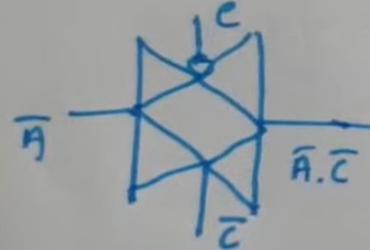
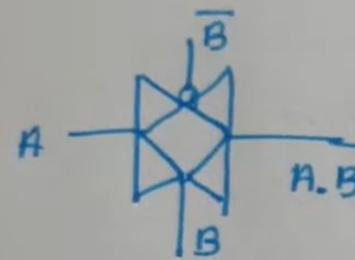
D Latch Implementation using Transmission Gates >





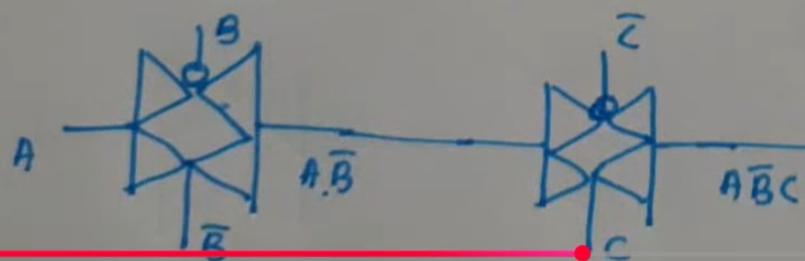
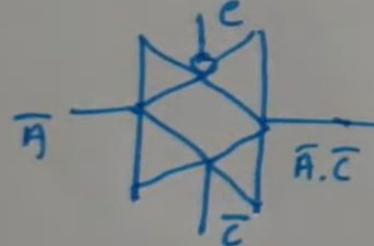
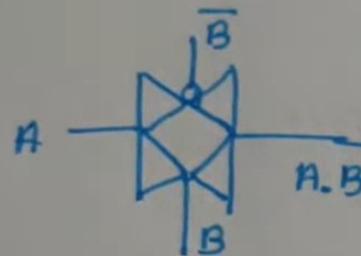
Examples of Boolean function using Transmission gate.

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



Examples of Boolean function using Transmission gate.

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



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

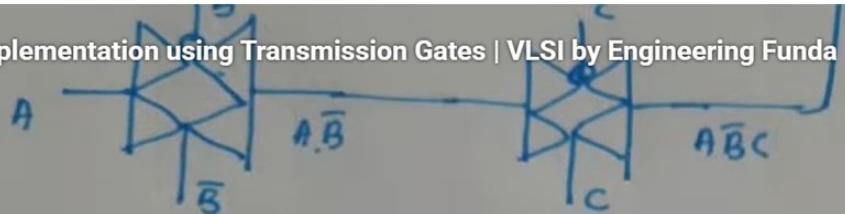


3:08 / 8:02

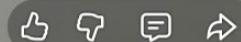
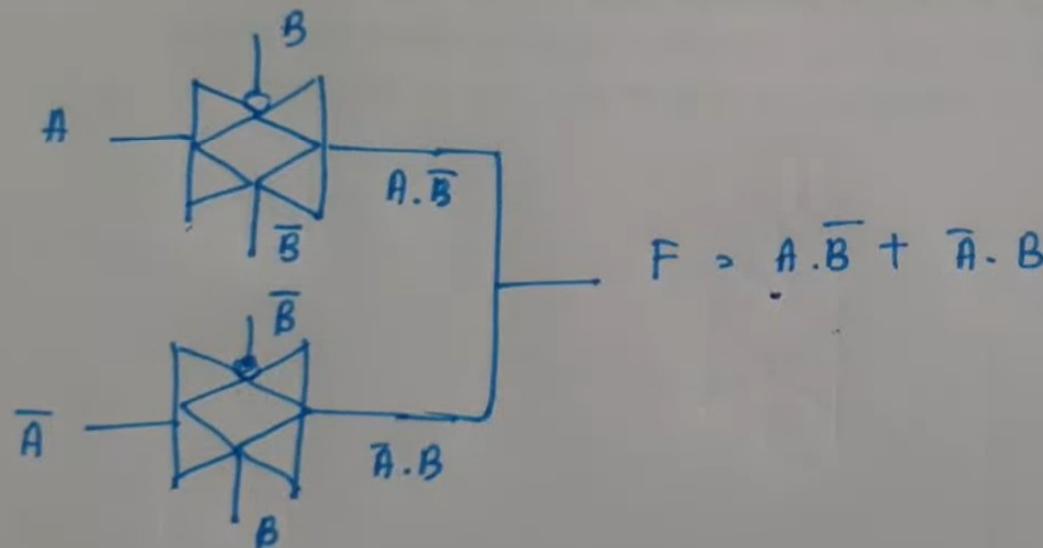
2 - Example of Boolean Function Implementation using Transmi



## Boolean Function Implementation using Transmission Gates | VLSI by Engineering Funda



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



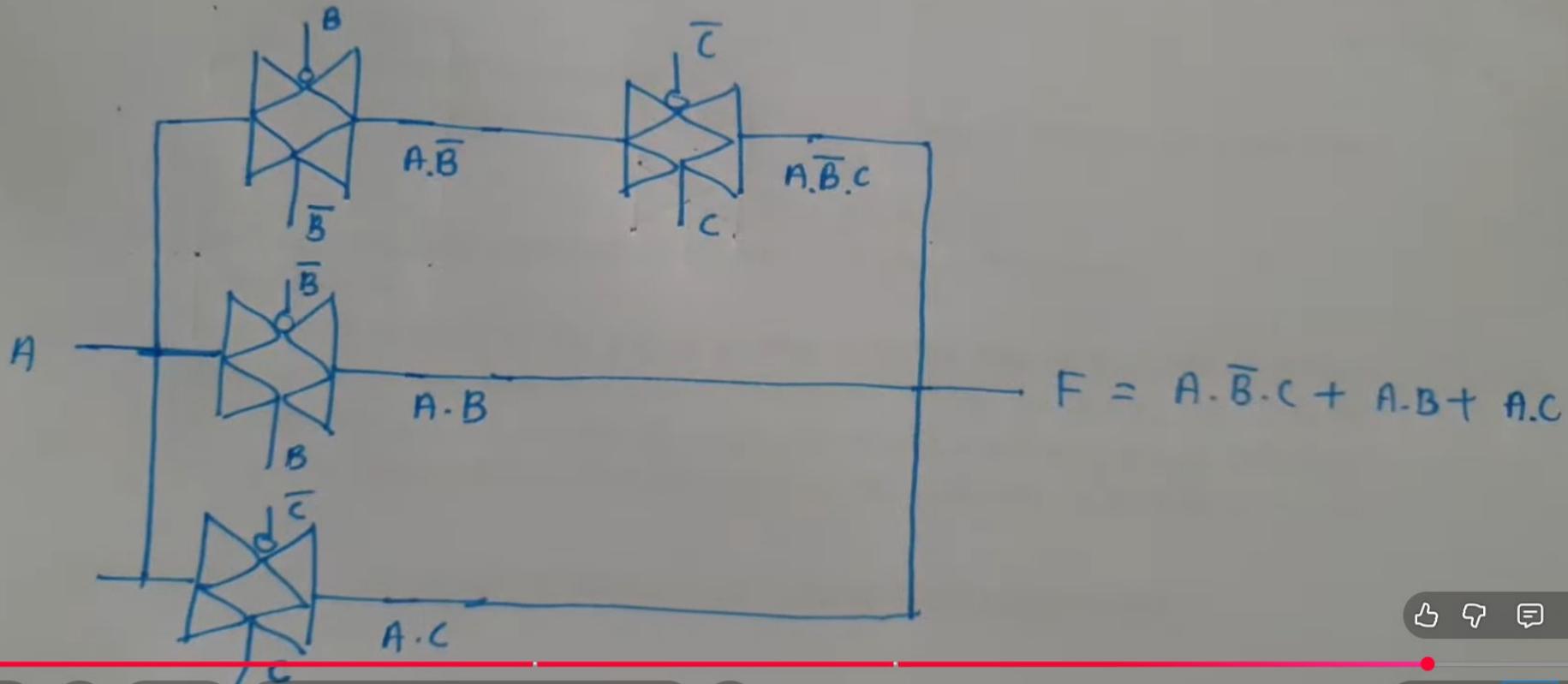
4:42 / 8:02

2 - Example of Boolean Function Implementation using Transm



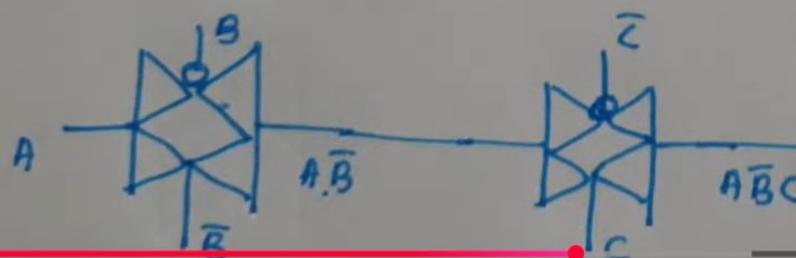
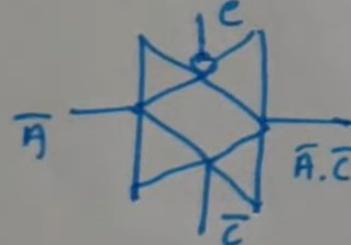
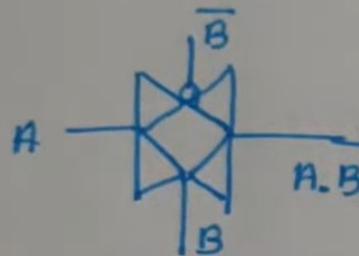


$$F = A \cdot \bar{B} \cdot C + A \cdot B + A \cdot C$$



Examples of Boolean function using Transmission gate.

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



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



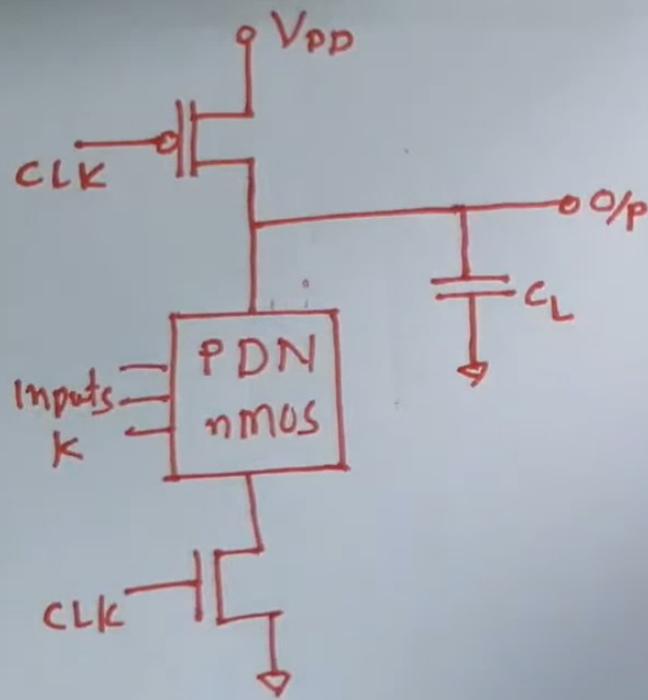
3:07 / 8:02

2 - Example of Boolean Function Implementation using Transmi



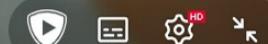
## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained

### Dynamic CMOS



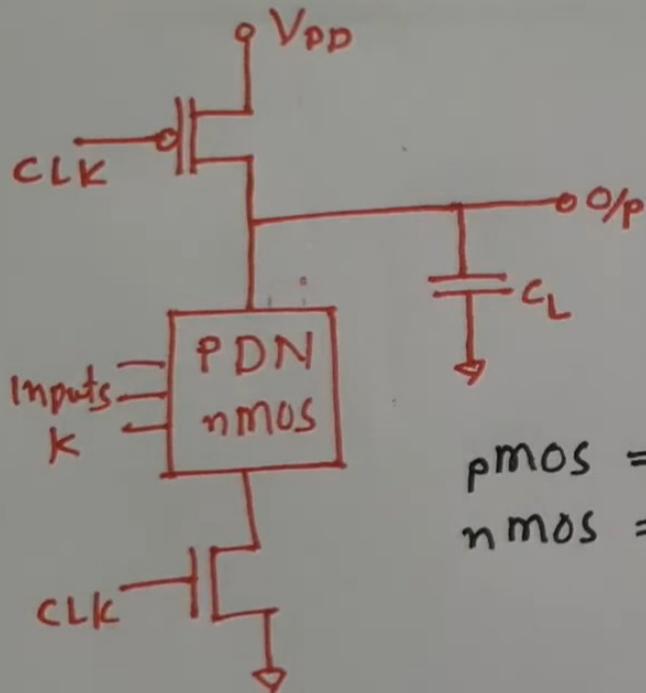
1:29 / 15:33

How Dynamic CMOS is better compared to Static CMOS &gt;

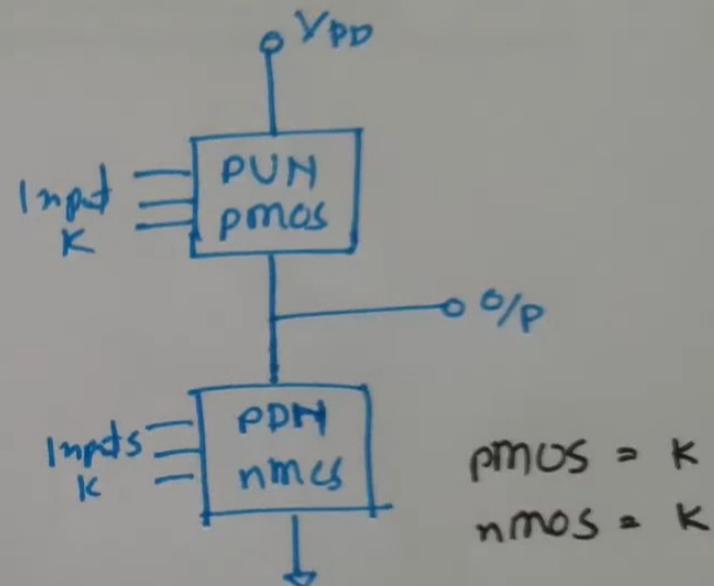


## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained

### Dynamic CMOS



$$pMOS = 1$$
$$nMOS = K+1$$



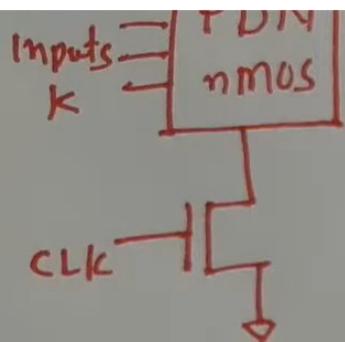
$$pMOS = K$$
$$nMOS = K$$



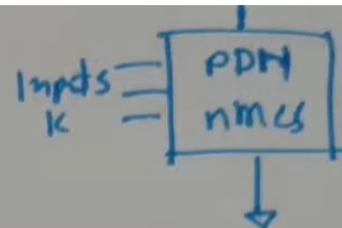
3:47 / 15:33

Dynamic CMOS Inverter >

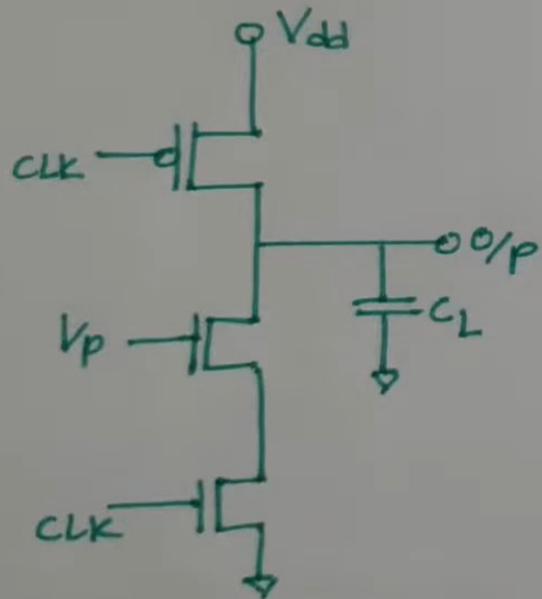




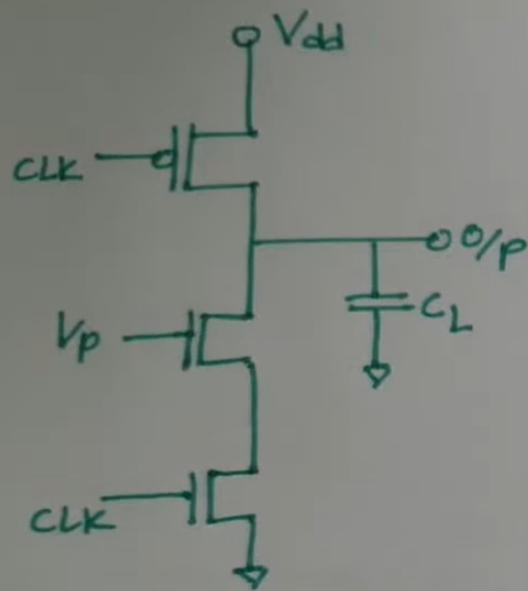
$$\begin{aligned} pMOS &= 1 \\ nMOS &= K+1 \end{aligned}$$



$$\begin{aligned} pMOS &= K \\ nMOS &= K \end{aligned}$$

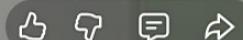


## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained



→ Circuit works in two modes

- 1] Pre-charging
- 2] Evaluation

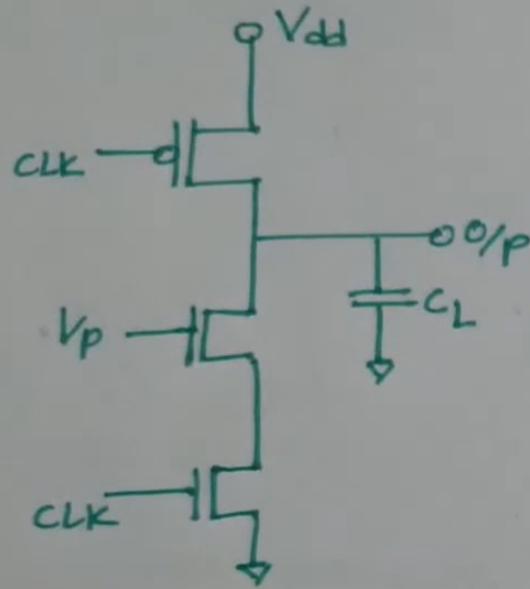


5:08 / 15:33

Dynamic CMOS Inverter Working &gt;

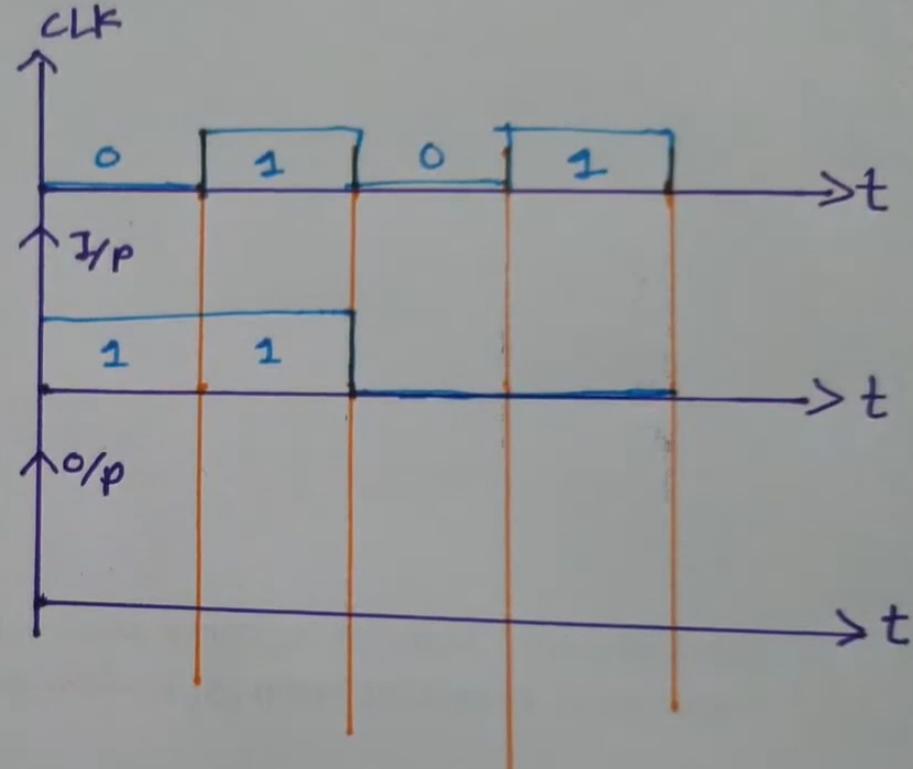


## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained

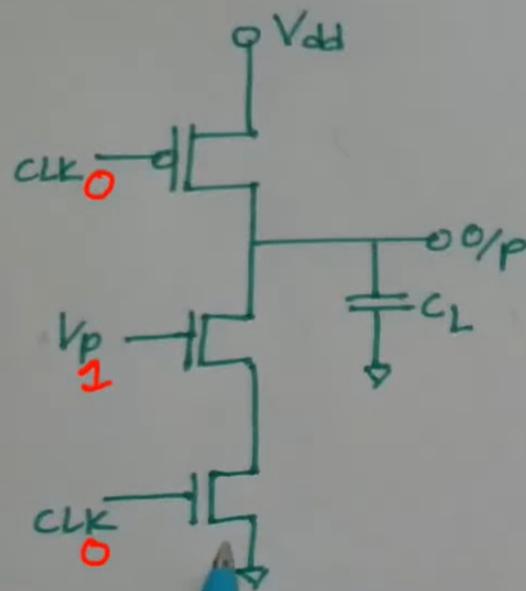


→ Circuit works in two modes

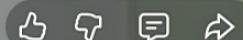
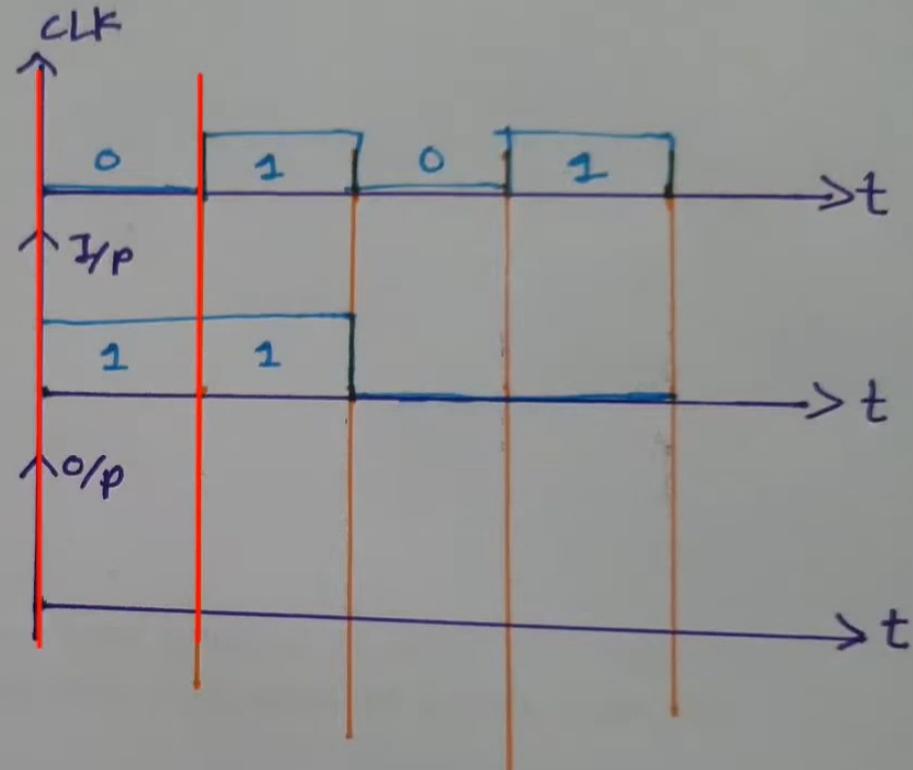
- 1] Pre-charging
- 2] Evaluation



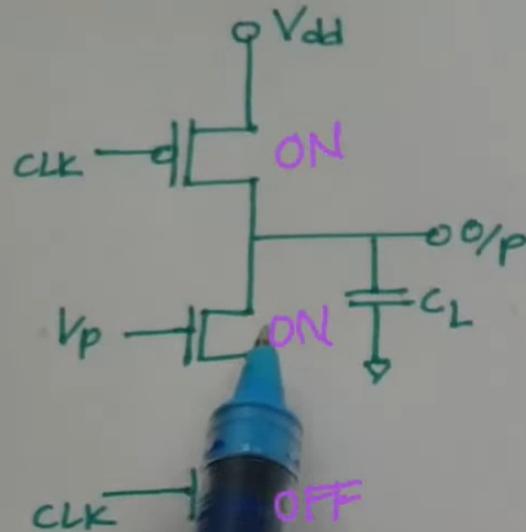
## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained



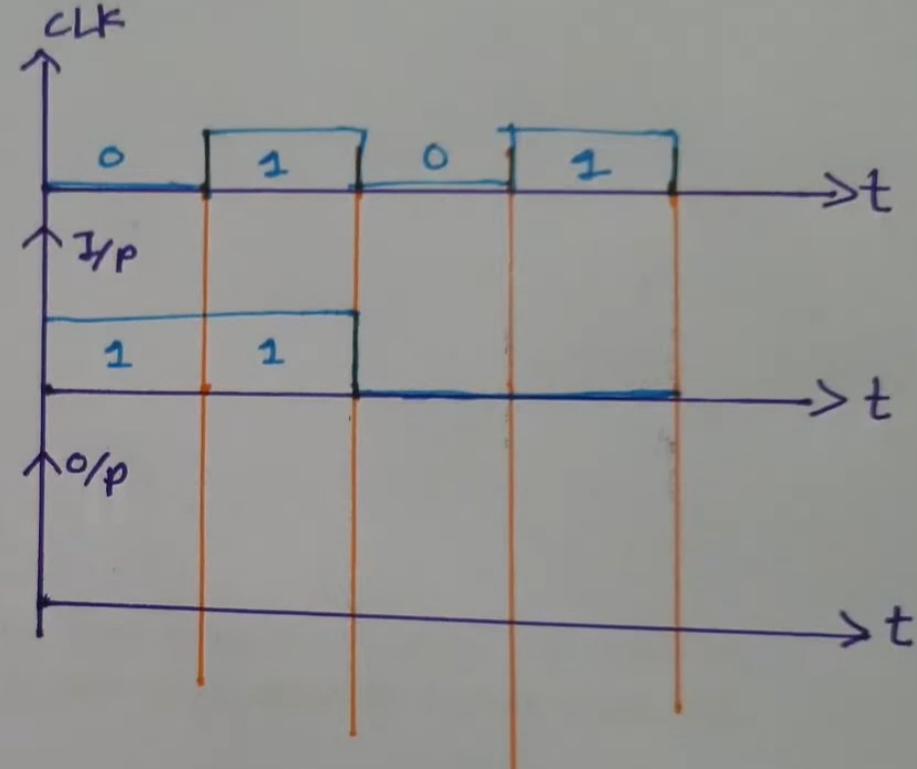
→ Circuit works in two modes  
1) charging  
2) aluation



## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained



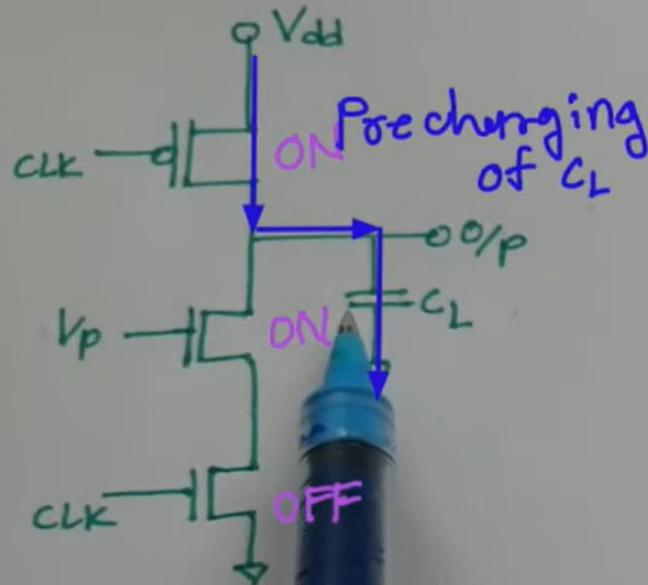
→ Circuit works in two modes  
1) Clamping  
2) Inverting



6:44 / 15:33 Dynamic CMOS Inverter Working >

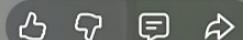
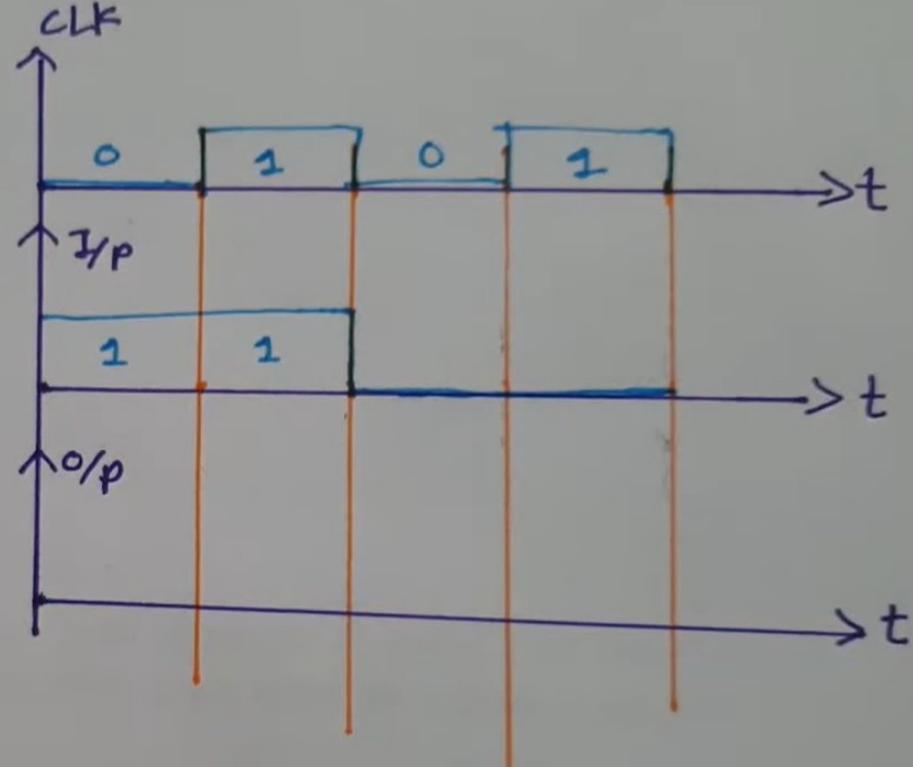


## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained



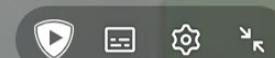
→ Circuit works in  
modes

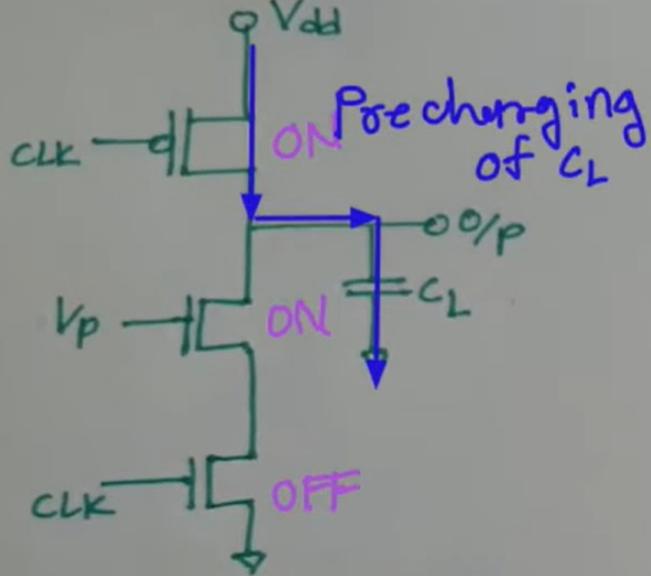
- 1] Precharging
- 2] Output



6:58 / 15:33

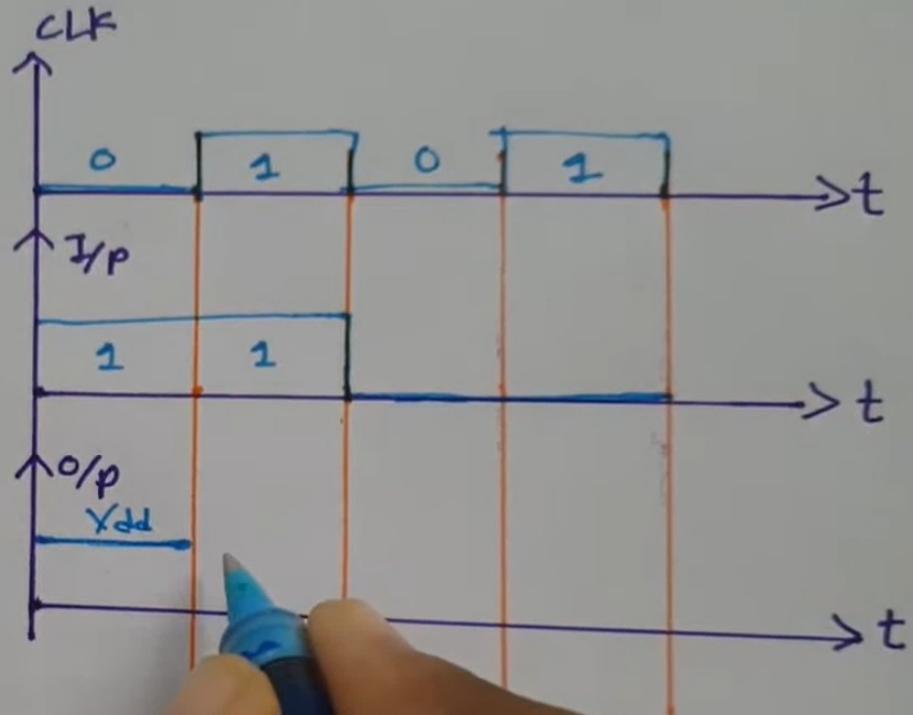
Dynamic CMOS Inverter Working >

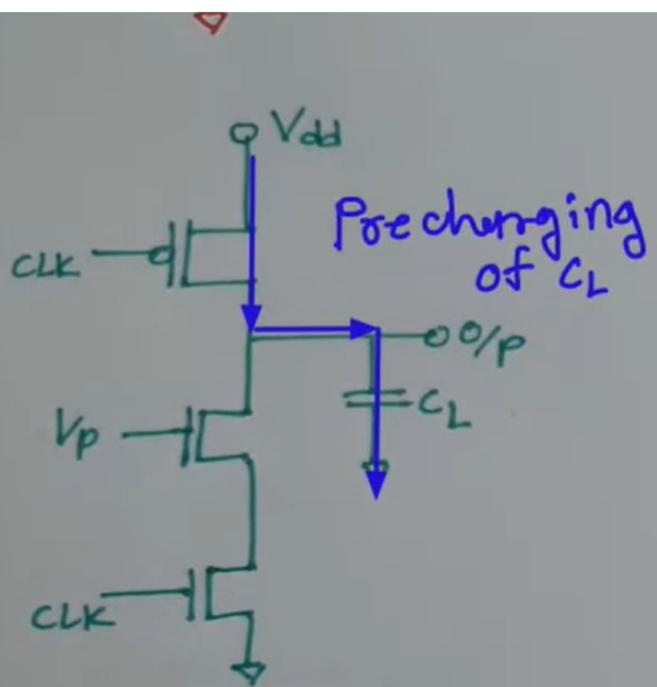




→ Circuit works in two modes

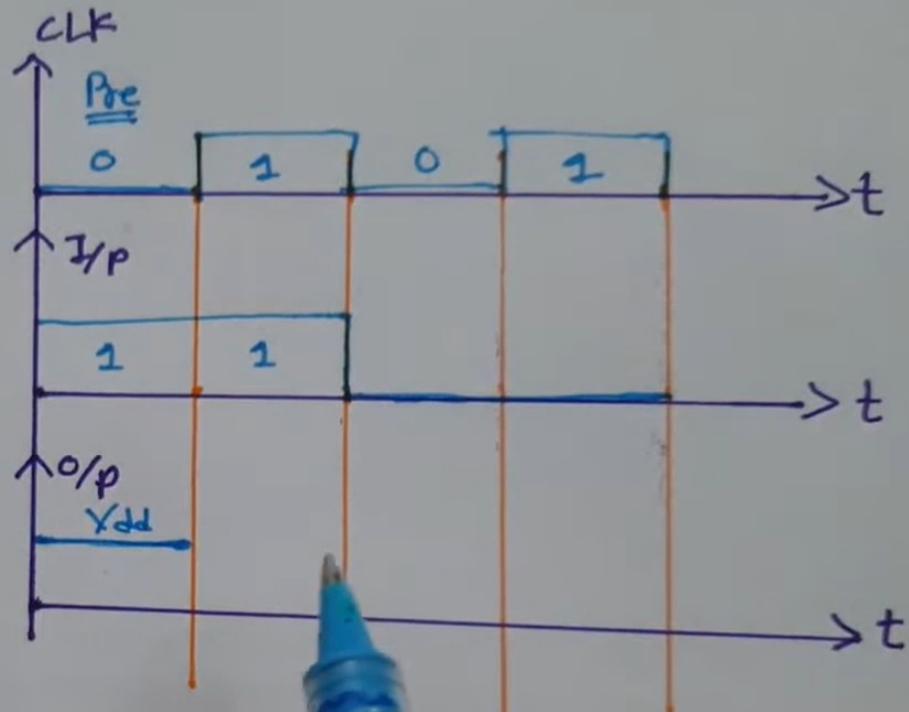
- 1] Pre-charging
- 2] Evaluation



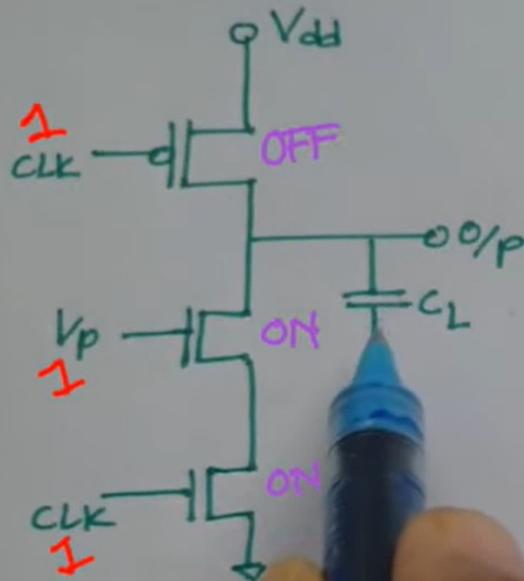


→ Circuit works in two modes

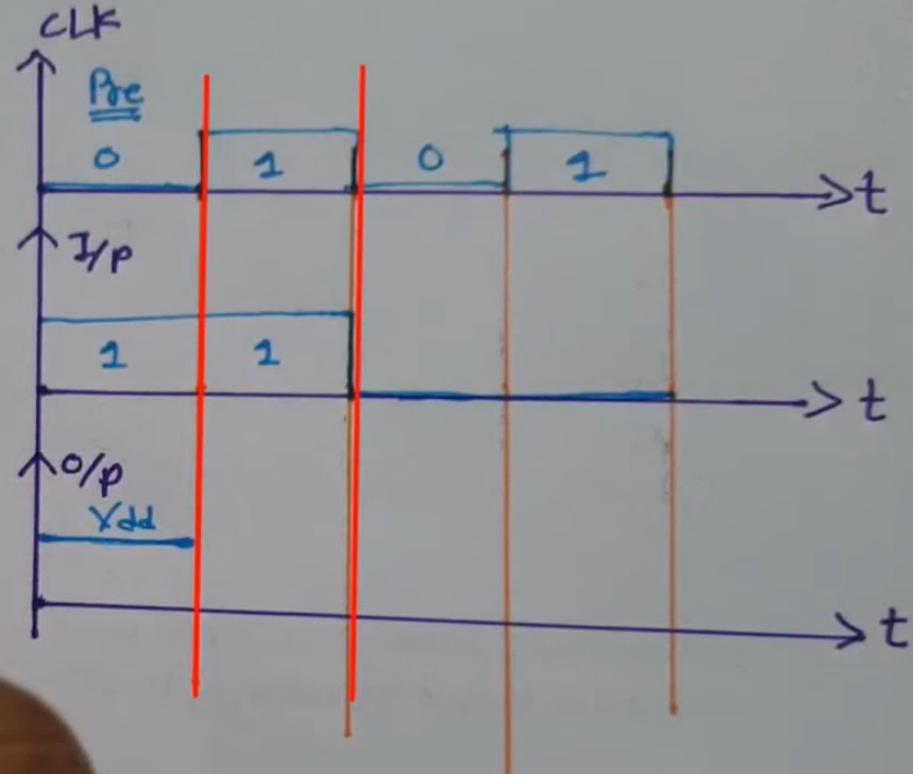
- 1] Pre-charging
- 2] Evaluation



## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained



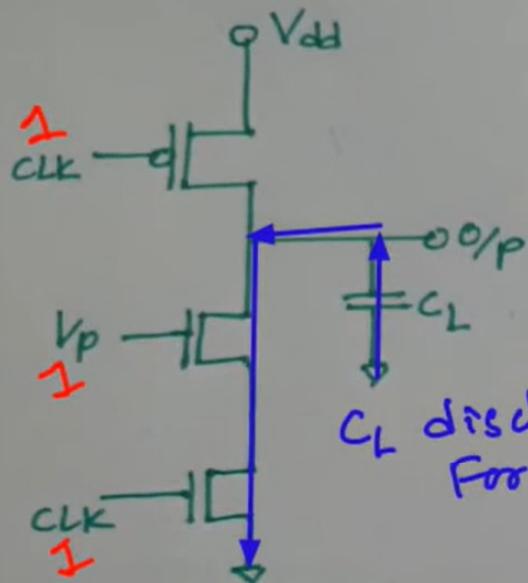
→ Circuit



8:05 / 15:33

Dynamic CMOS Inverter Working >

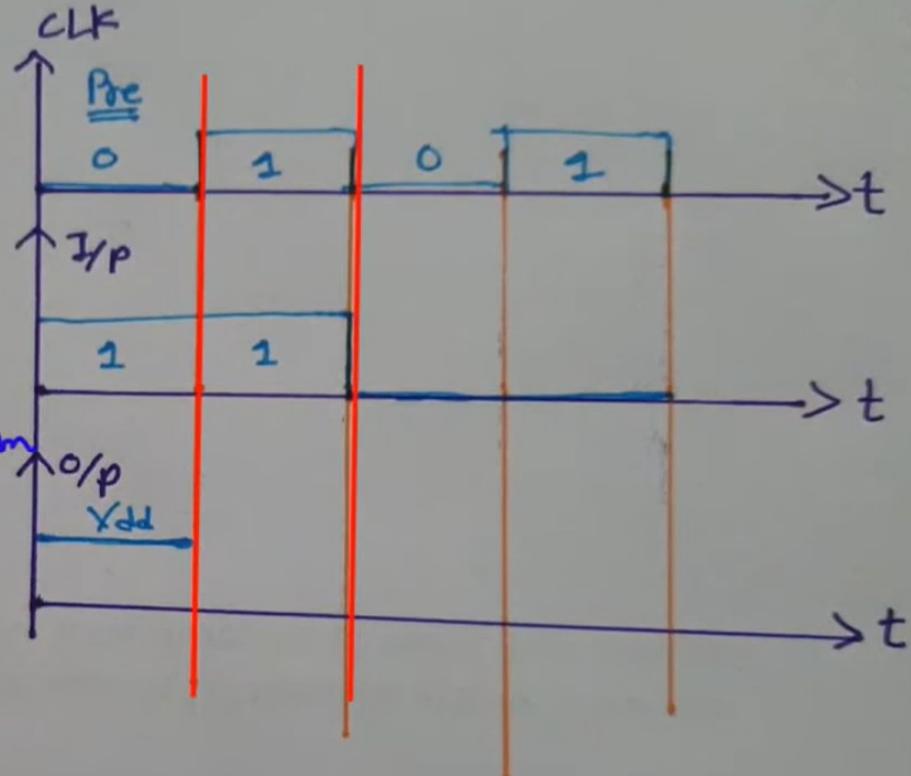




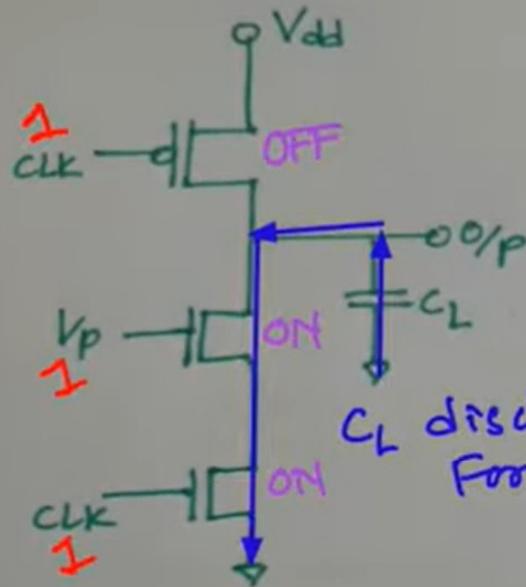
$C_L$  discharging  
For evaluation

→ Circuit operates in two modes

- 1] Pre charging
- 2] Inverting

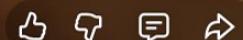
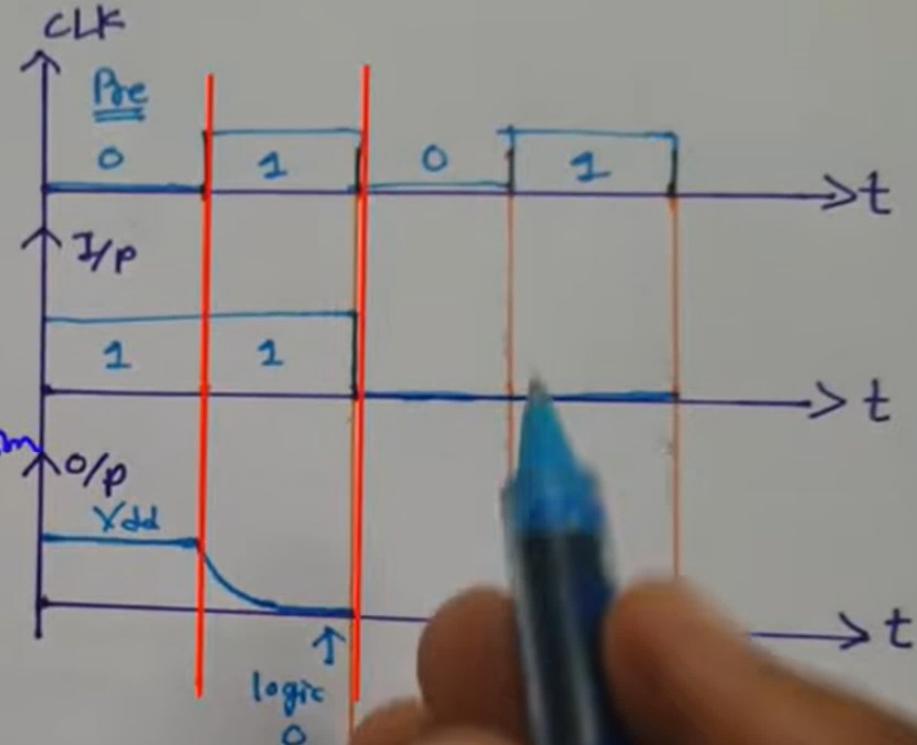


## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained



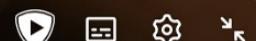
→ Circuit works in two modes

- 1] Pre-charging
- 2] Evaluation



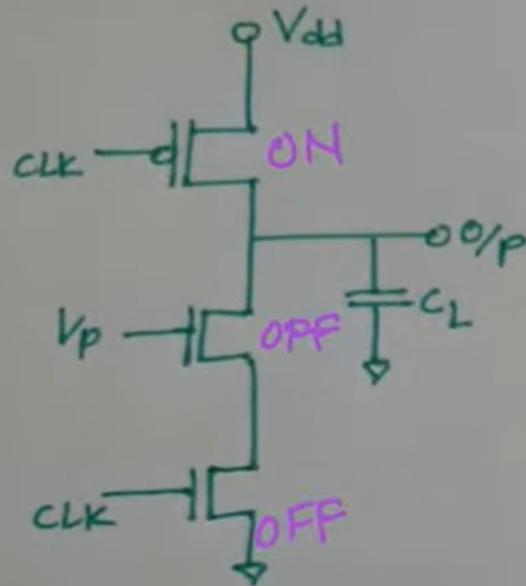
8:36 / 15:33

Dynamic CMOS Inverter Working >



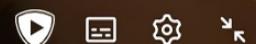
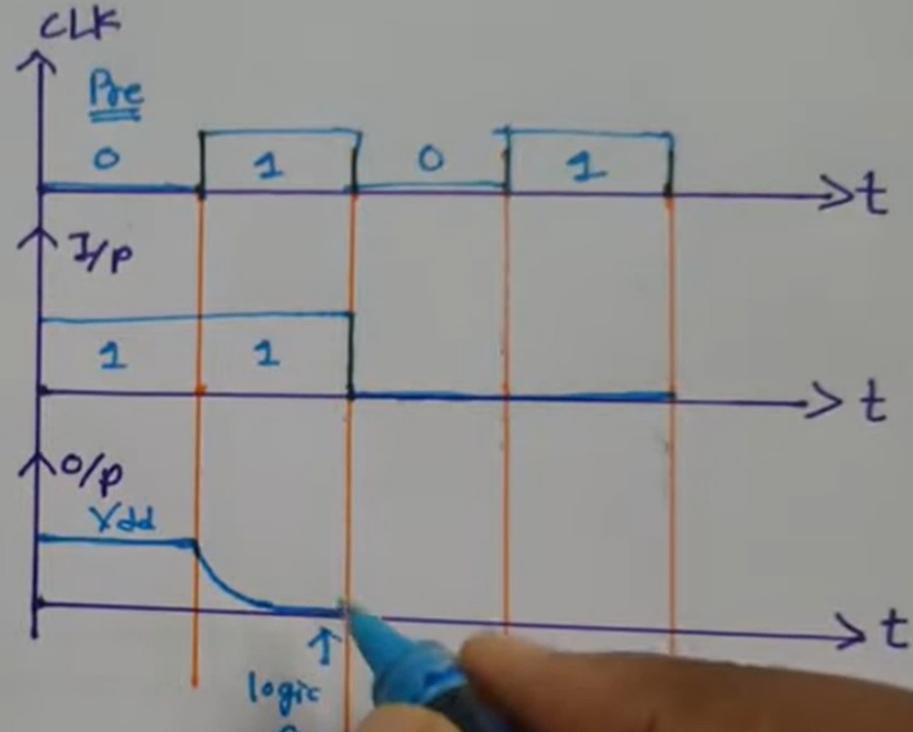


## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained



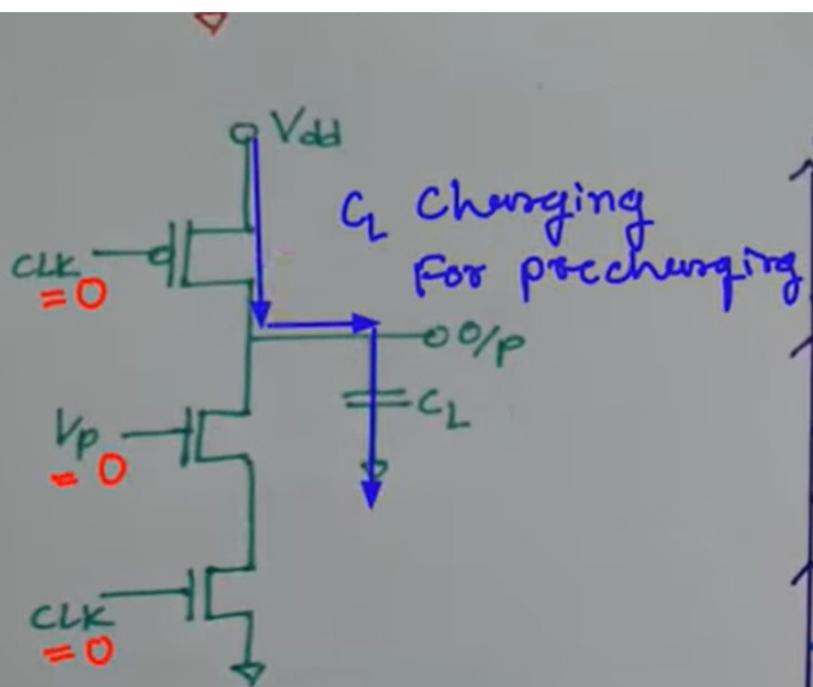
→ Circuit works in two modes

- 1] Pre-charging
- 2] Evaluation



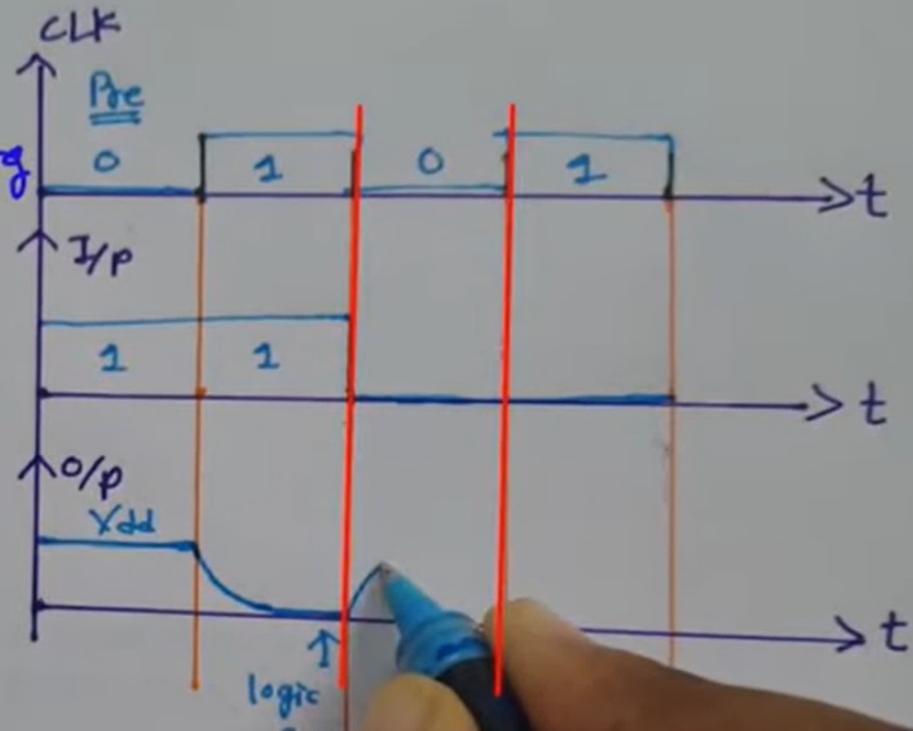
8:59 / 15:33

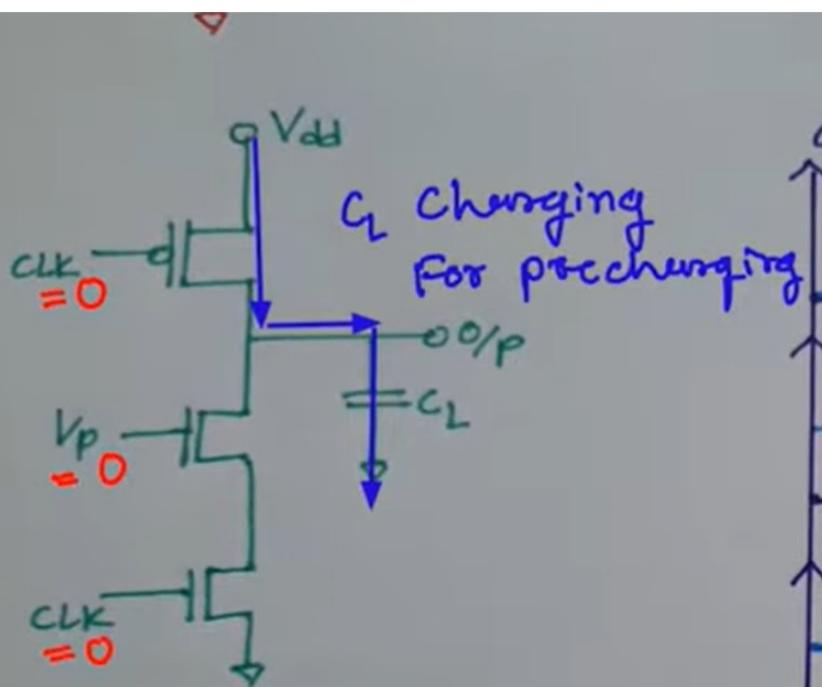
Dynamic CMOS Inverter Working >



→ Circuit works in two modes

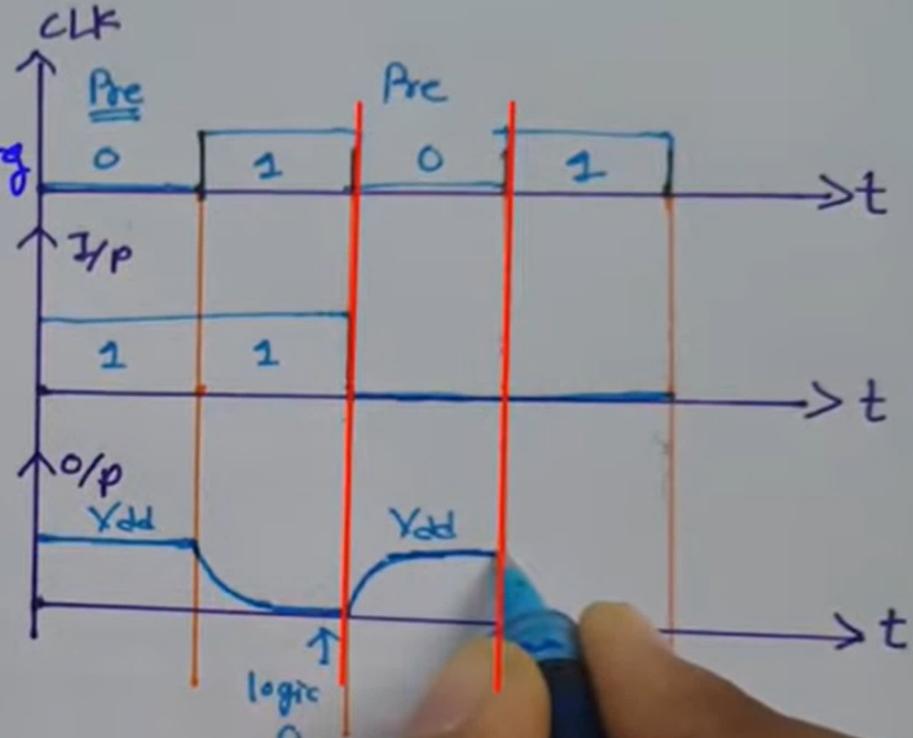
- 1] Pre-charging
- 2] Evaluation



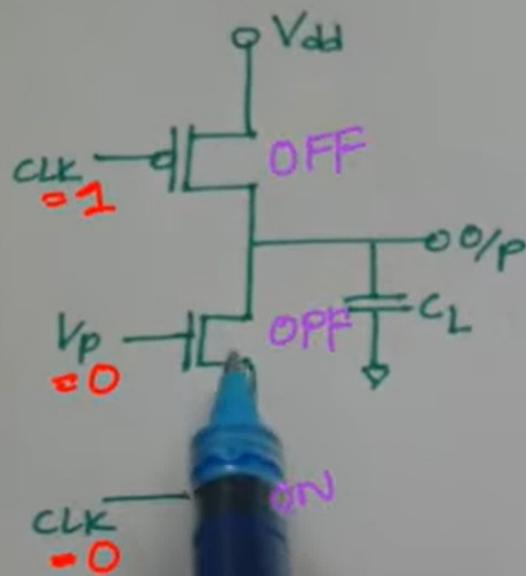


→ Circuit works in two modes

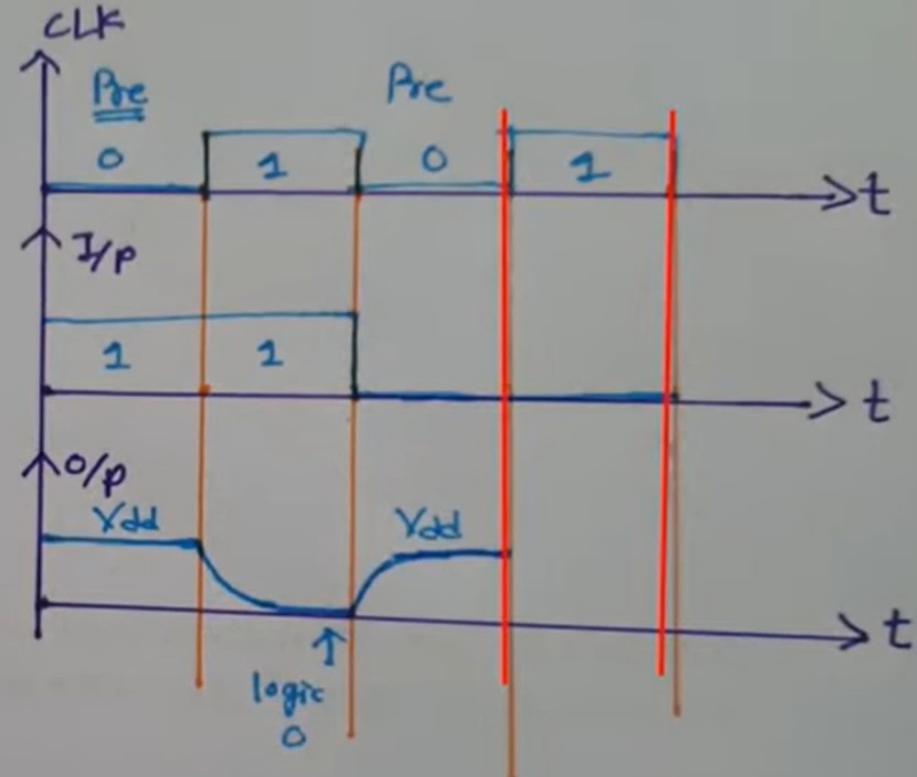
- 1] Pre-charging
- 2] Evaluation

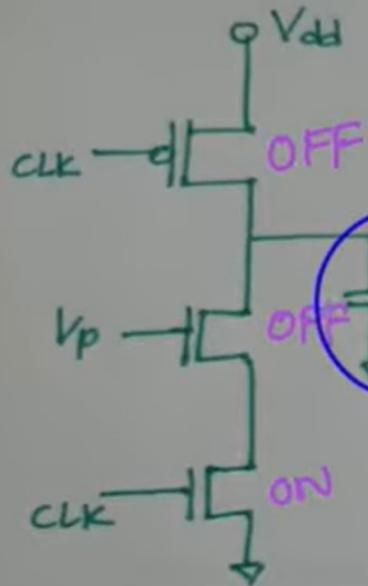


## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained



→ Circuit  
two modes

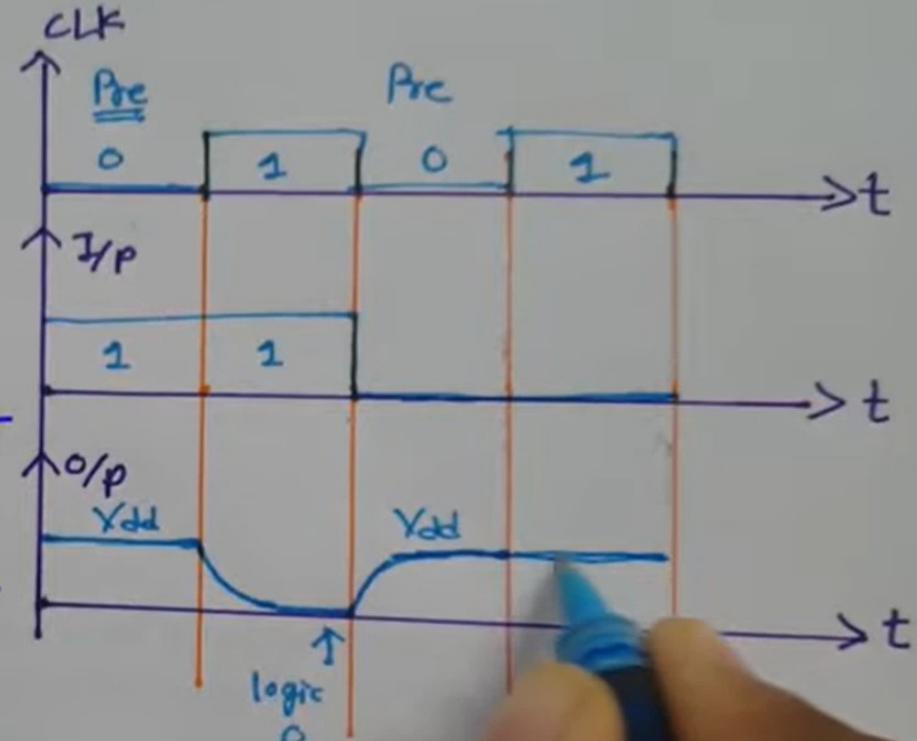


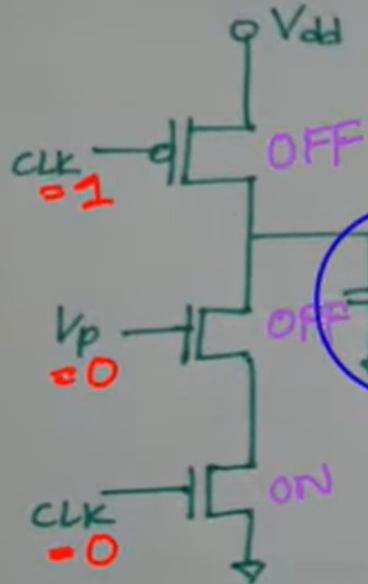


Load  
Capacitance  
does not get  
Path to  
discharge.

→ Circuit works in two modes

- 1] Pre-charging
- 2] Evaluation

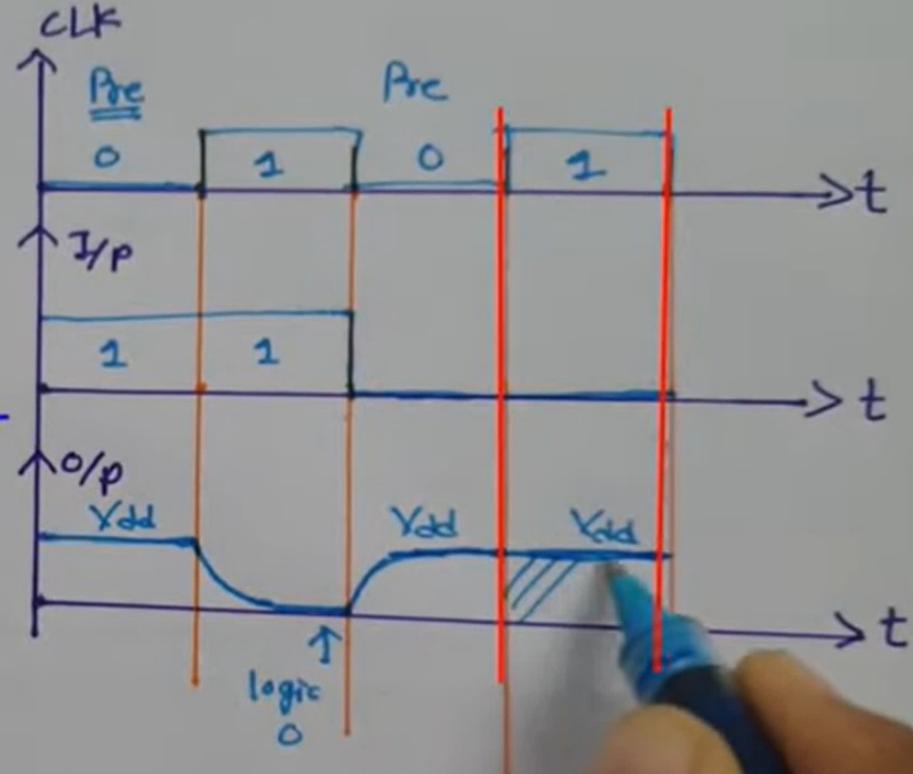




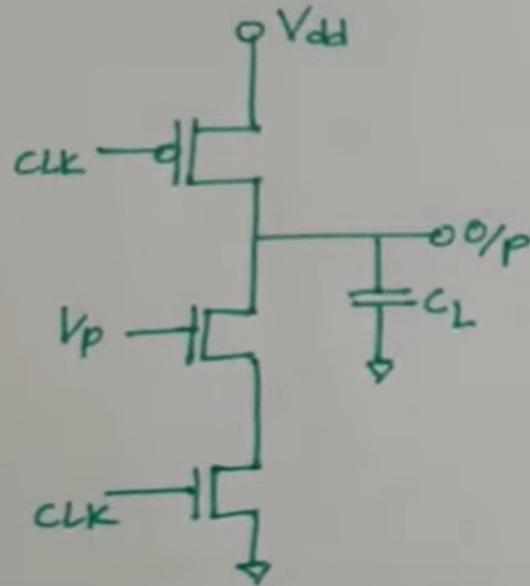
Load  
Capacitance  
does not get  
Path to  
discharge.

→ Circuit works in two modes

- 1] Pre-charging
- 2] Evaluation

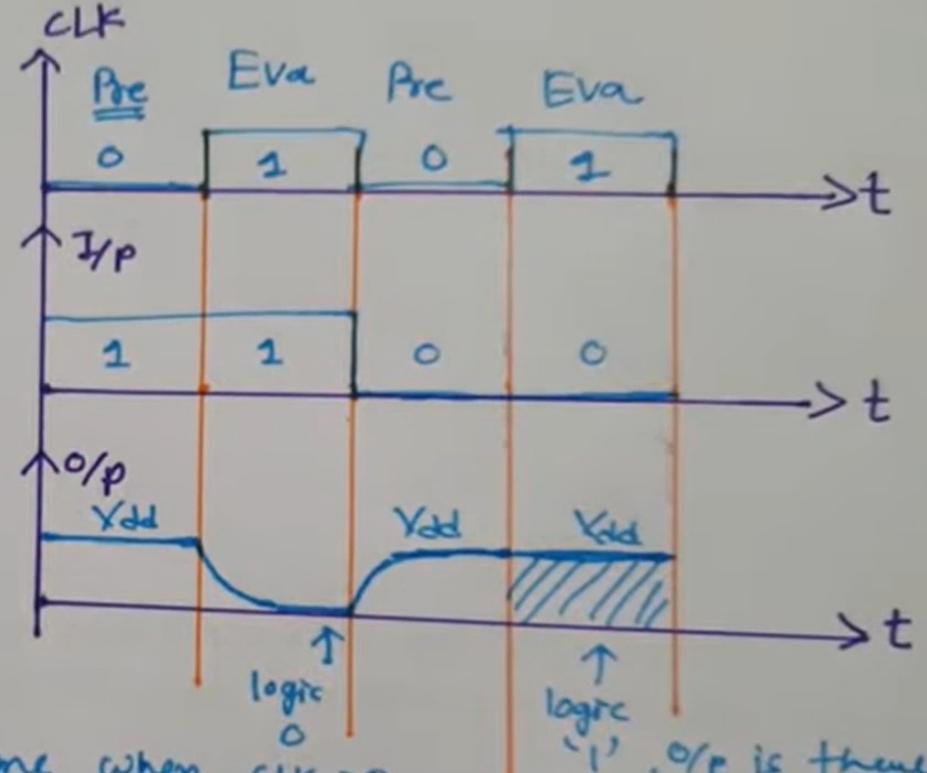


## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained



→ Circuit works in two modes

- 1] Pre-charging - It is done when CLK = 0.
- 2] Evaluation - It is done when CLK = 1.



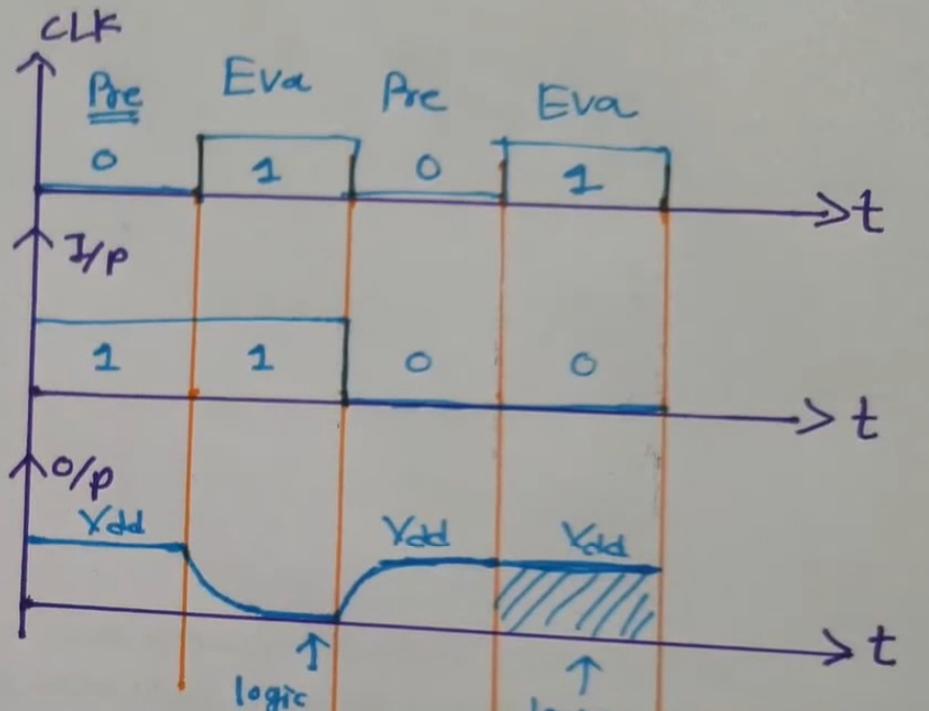
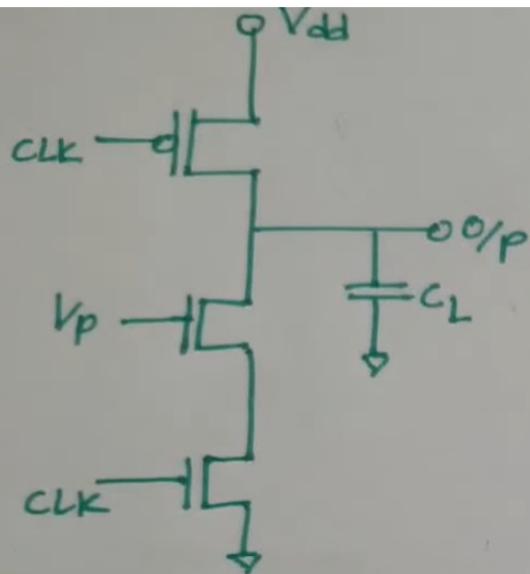
, O/p is there  
with high  
impedance



11:22 / 15:33

Dynamic CMOS Inverter Working >





→ Circuit works in two modes

1] Pre-charging - It is done when  $CLK = 0$

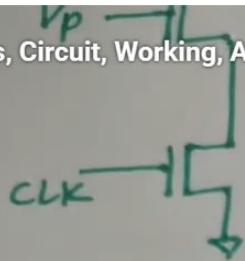
2] Evaluation - It is done when  $CLK = 1$ .

$O/p$  is there with high impedance.

### Advantages

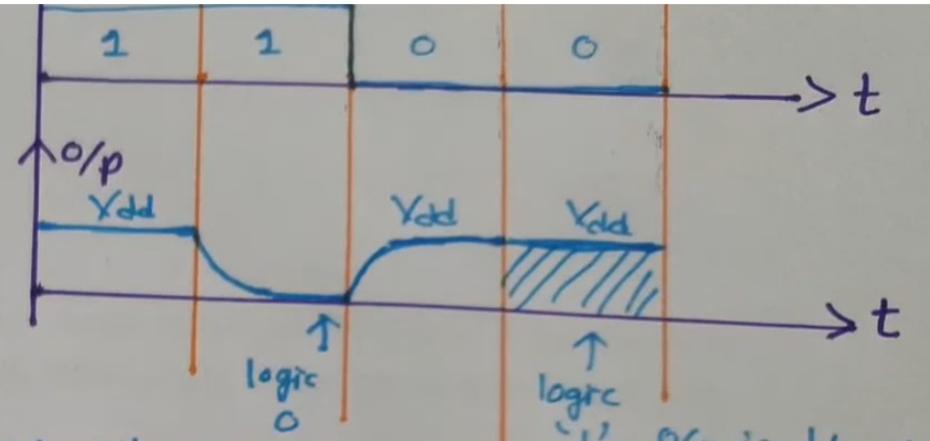
- $V_{OH} = V_{DD}$ ,  $V_{OL} = 0$ , It has max voltage swing at o/p.
- $I_{SS} = 0$

## Dynamic CMOS ( Basics, Circuit, Working, Advantages & Disadvantages) Explained



→ Circuit works in two modes

- 1] Pre-charging - It is done when  $CLK = 0$
- 2] Evaluation - It is done when  $CLK = 1$ .



, O/p is there with high impedance.

### Advantages

- $V_{OH} = V_{DD}$ ,  $V_{OL} = 0$ , It has max voltage swing at o/p.
- $I_{SS} = 0$
- It has only 1 pmos.
- It has faster switching Speed.
- Circuit size is small.



13:44 / 15:33

Advantages of Dynamic CMOS >



2) Evaluation - It is done when  $CLK = 1$ .

Impedance.

### Advantages

- $V_{OH} = V_{DD}$ ,  $V_{OL} = 0$ , It has max voltage swing at o/p.
- $I_{SS} = 0$
- It has only 1 pmos.
- It has faster switching speed.
- Circuit size is small.
- It has less capacitance loading compared to static CMOS.

### Disadvantages

- When o/p is logic '1', it stays in high impedance mode which is more effected by noise.

→ Circuit works :

- 1) Pre-charging - It is done when  $\text{CLK} = 0$
- 2) Evaluation - It is done when  $\text{CLK} = 1$ .

$V_{DD}$  |  
0 |

logic  
'1'

, O/p is there  
with high  
impedance.

### Advantages

- $V_{OH} = V_{DD}$ ,  $V_{OL} = 0$ , It has max voltage swing at o/p.
- $I_{SS} = 0$
- It has only 1 pmos.
- It has faster switching Speed.
- Circuit size is small.
- It has less Capacitive Loading compared to static CMOS.

### Disadvantages

- When o/p is logic '1', it stays in high impedance mode which is more effected by noise.
- It requires precharging.

→ Circuit works in two phases:

- 1) Pre-charging - It is done when  $\text{CLK} = 0$
- 2) Evaluation - It is done when  $\text{CLK} = 1$ .

0 |  
1 |

logic  
0 |

1 |, O/p is there  
with high  
impedance.

### Advantages

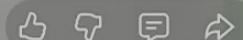
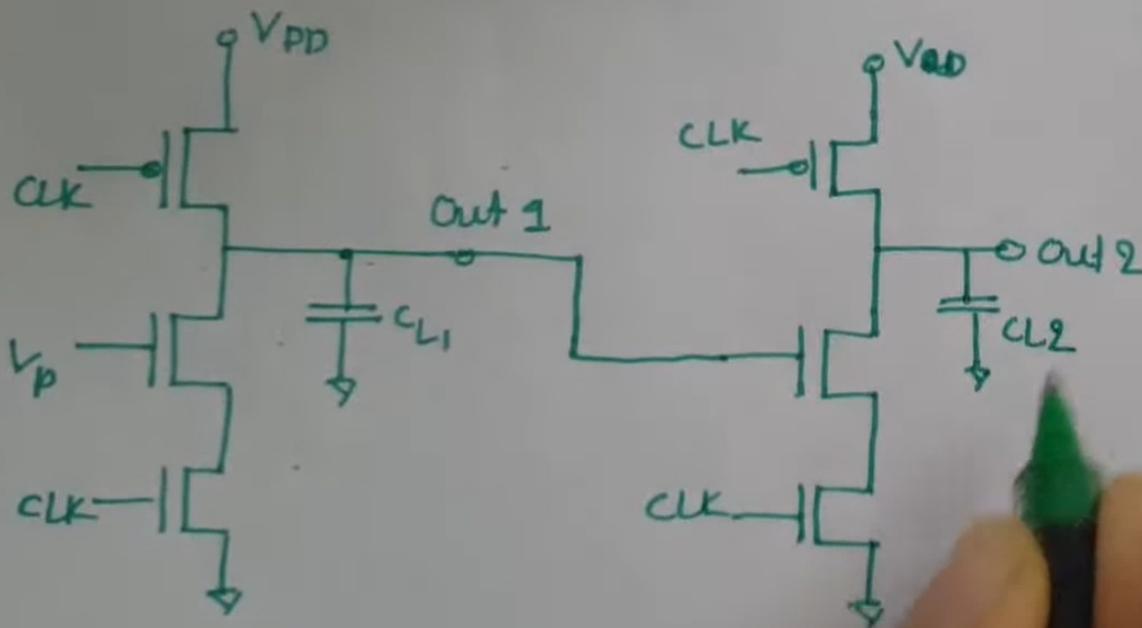
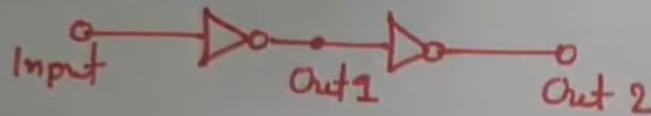
- $V_{OH} = V_{DD}$ ,  $V_{OL} = 0$ , It has max voltage swing at o/p.
- $I_{SS} = 0$
- It has only 1 pmos.
- It has faster switching Speed.
- Circuit size is small.
- It has less Capacitive Loading compared to static CMOS.

### Disadvantages

- When o/p is logic '1', it stays in high impedance mode which is more effected by noise.
- It requires precharging.
- Cascading issue, — Race problem.

## Cascading in Dynamic CMOS

→ Cause



2:16 / 12:57

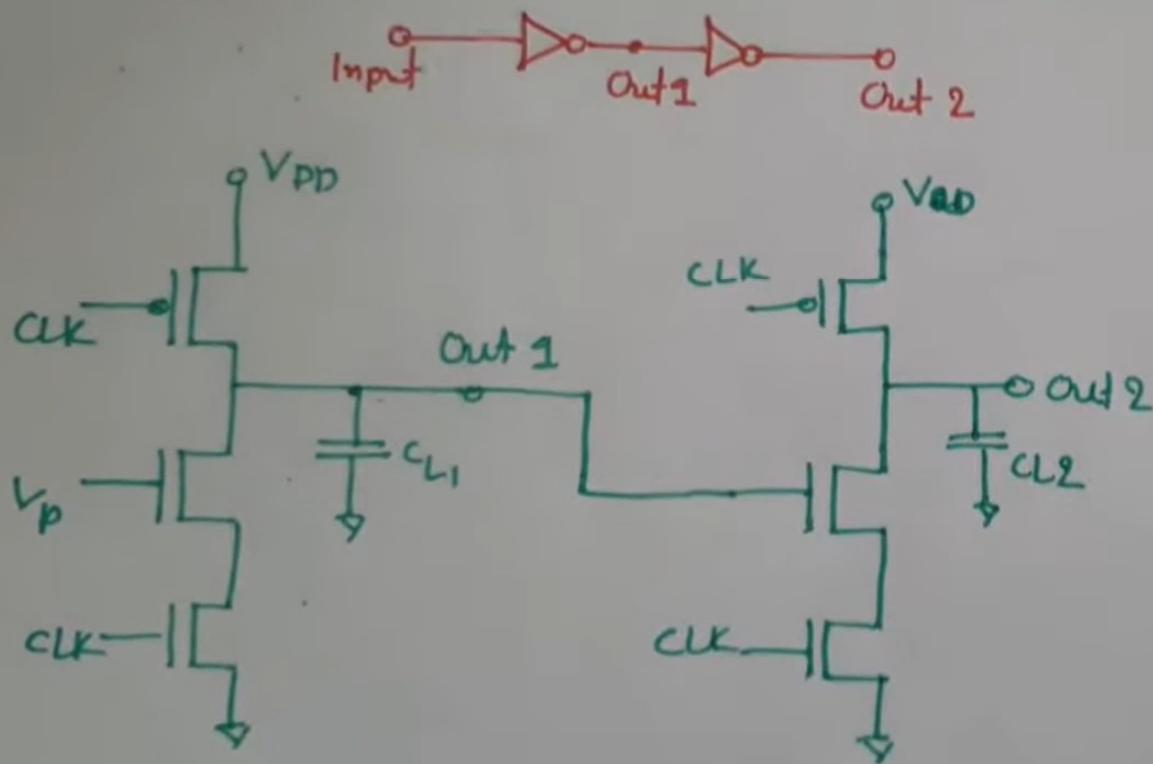
Circuit of NOT Gate Cascading in Dynamic CMOS >

^



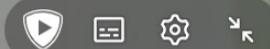
## Cascading in Dynamic CMOS

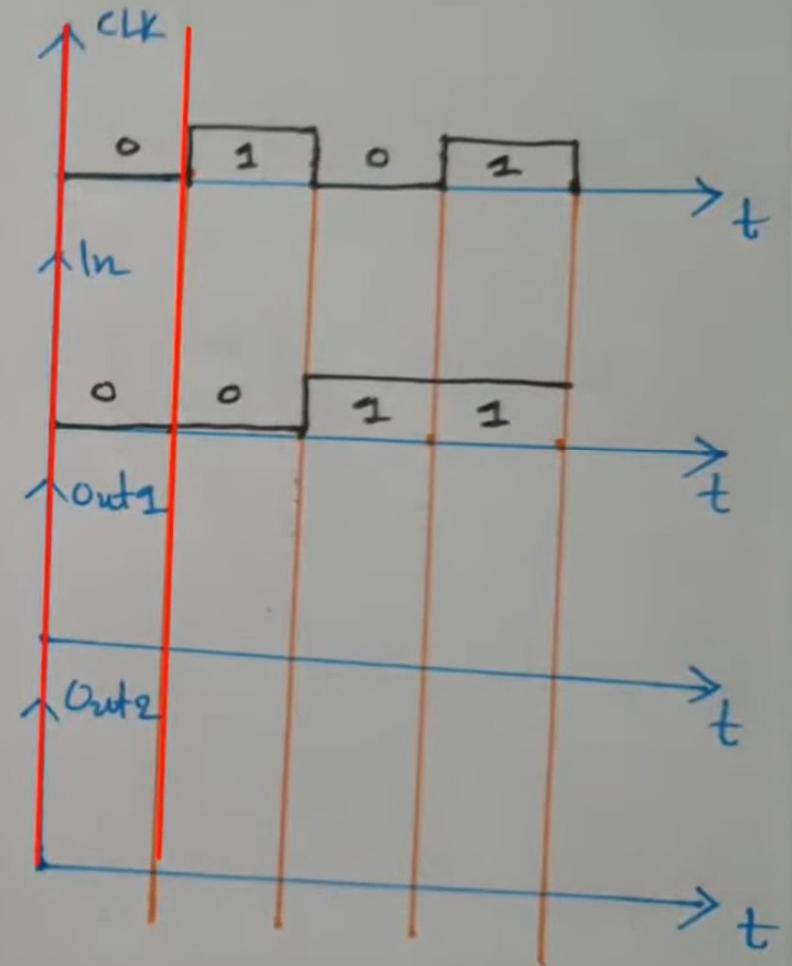
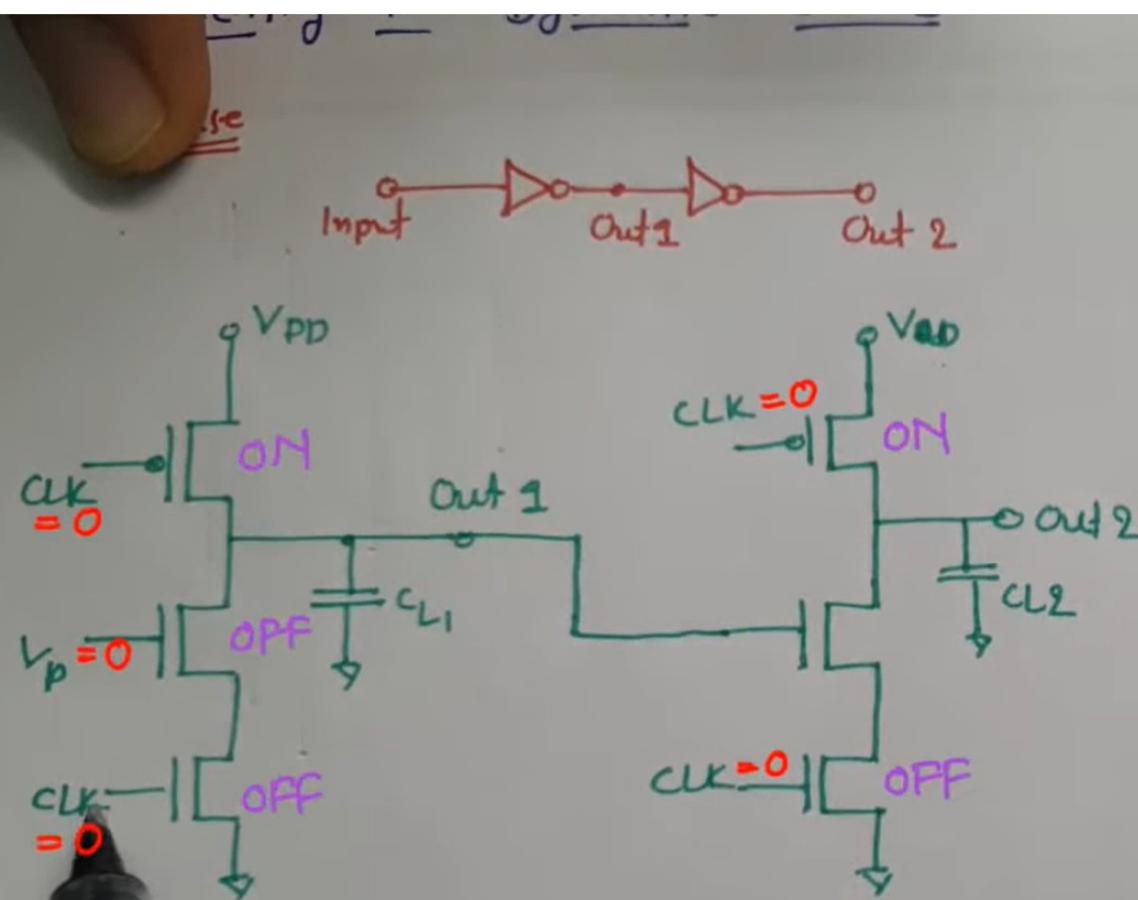
→ Case

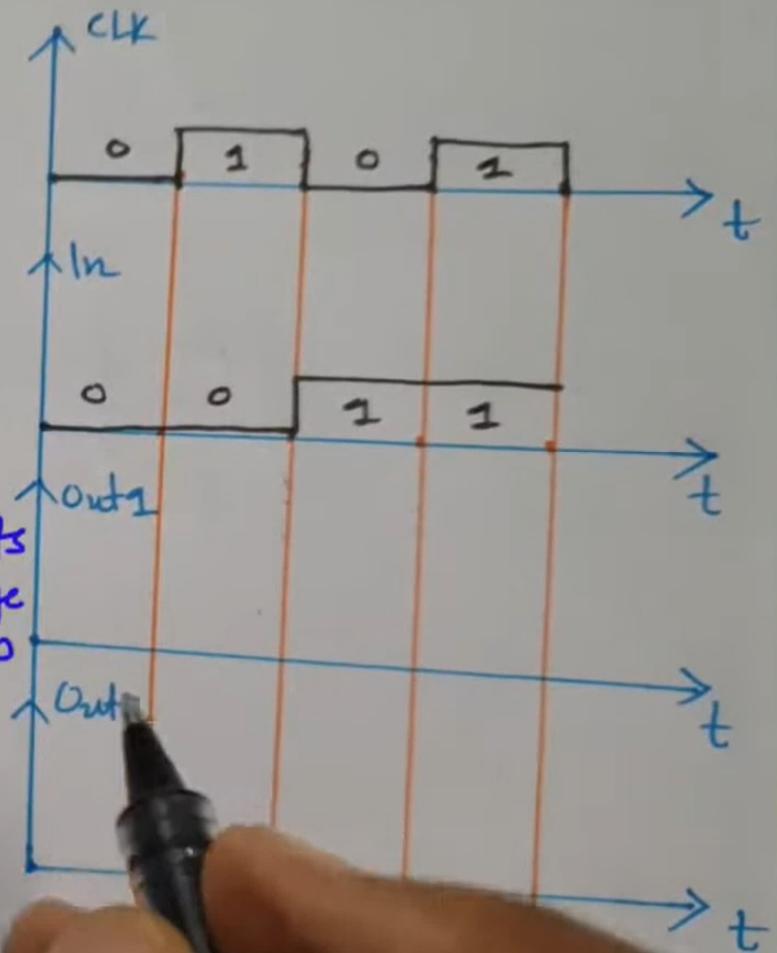
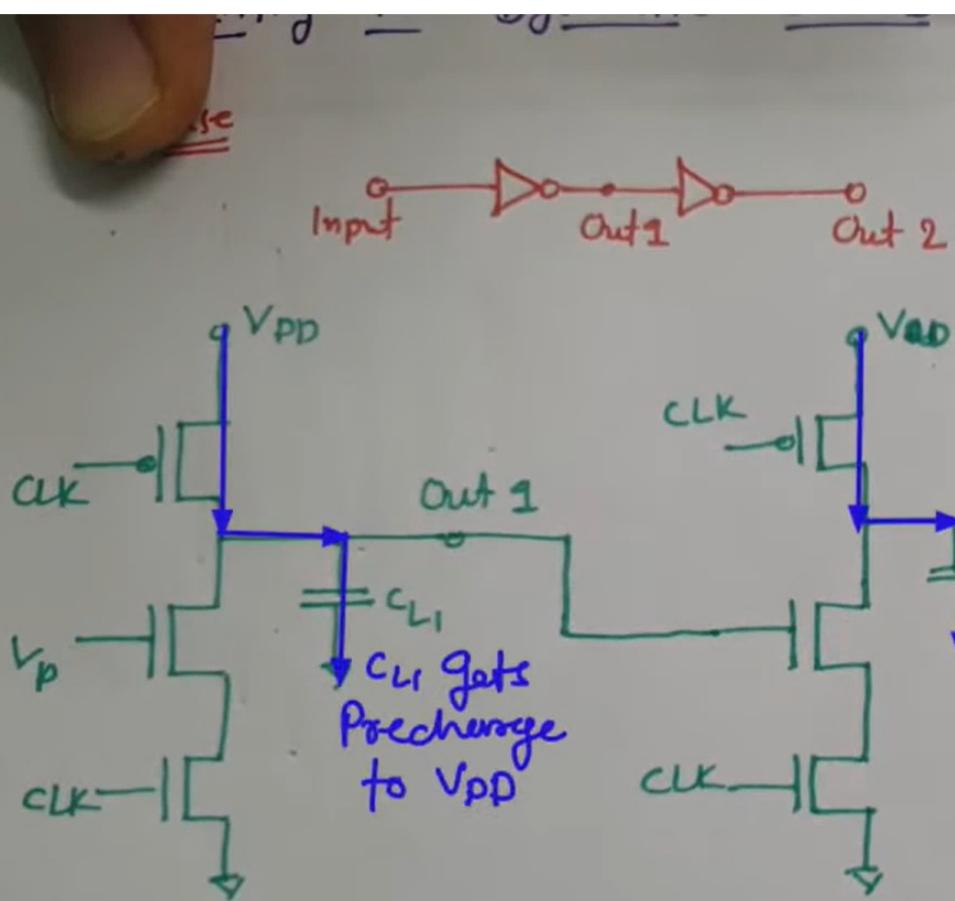


2:17 / 12:57

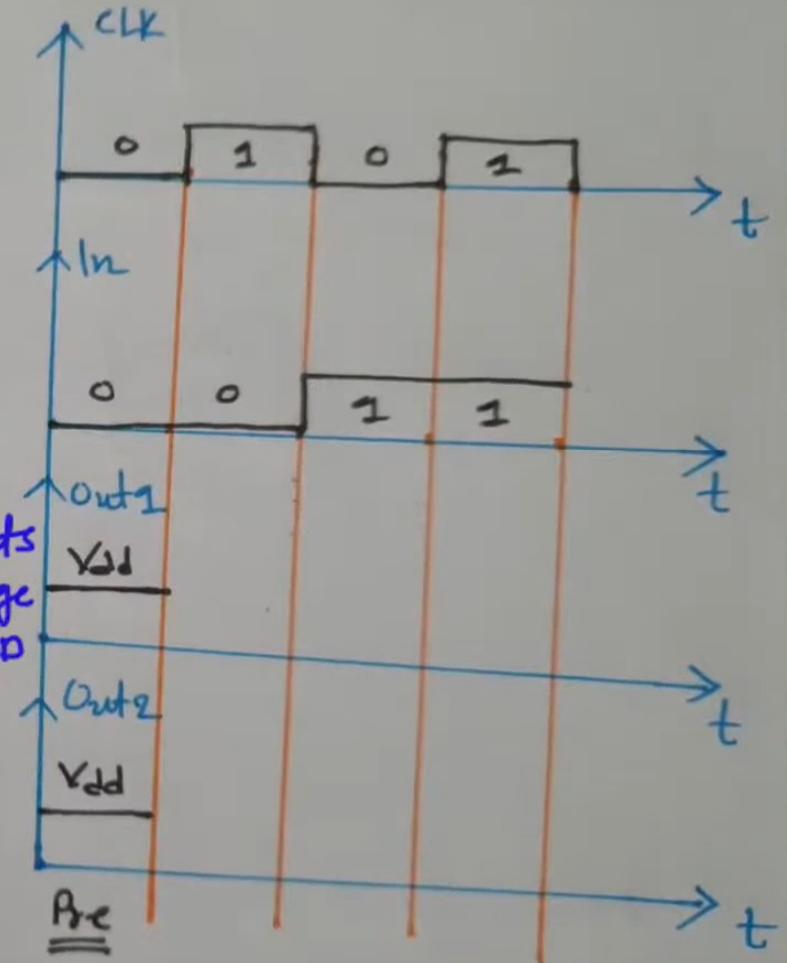
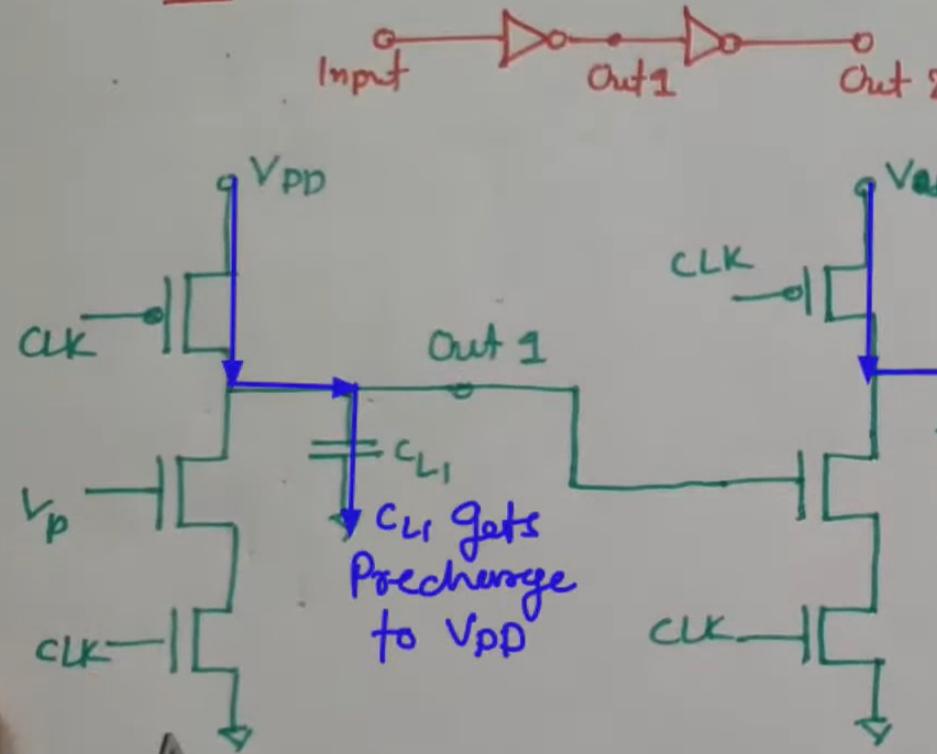
Circuit of NOT Gate Cascading in Dynamic CMOS >



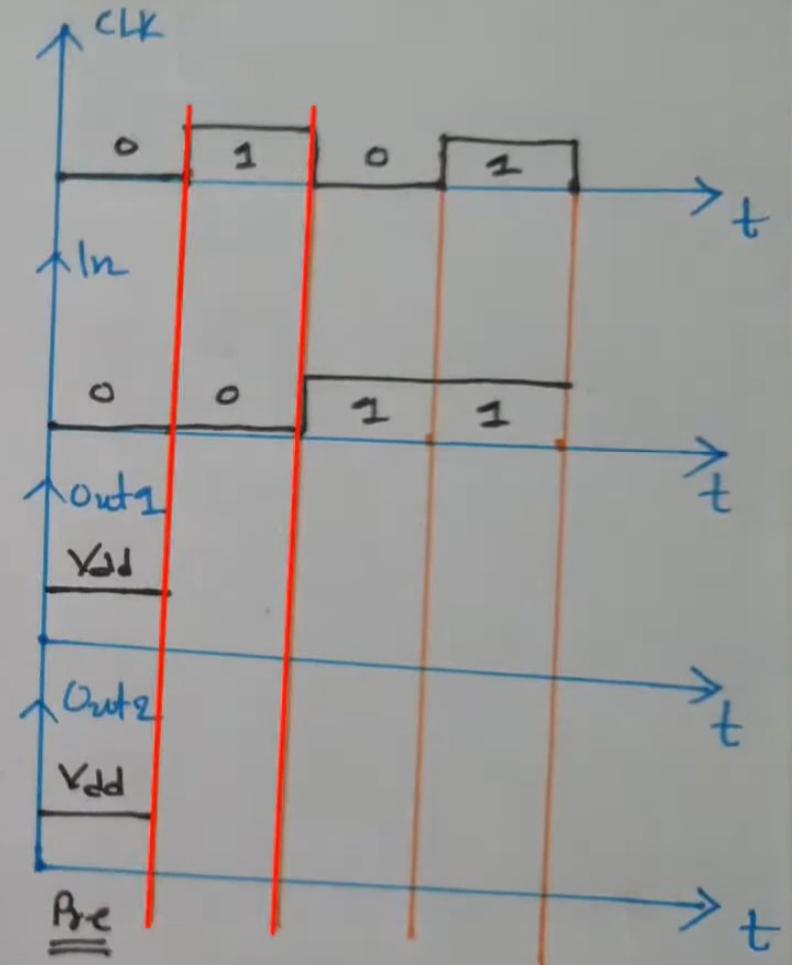
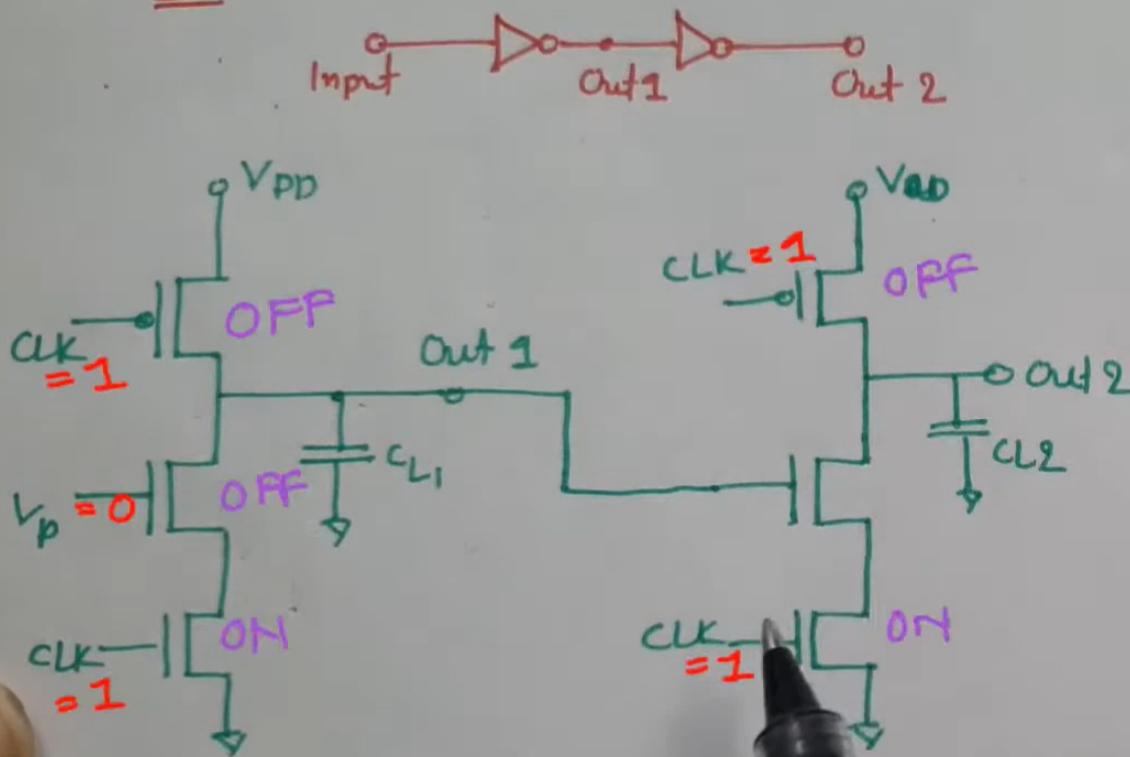




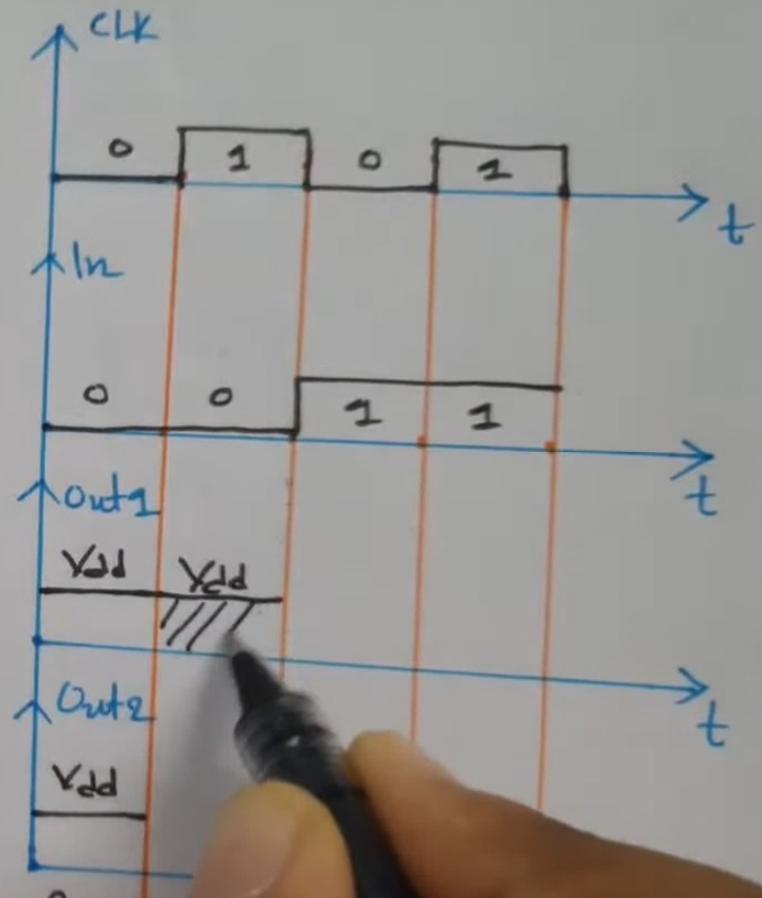
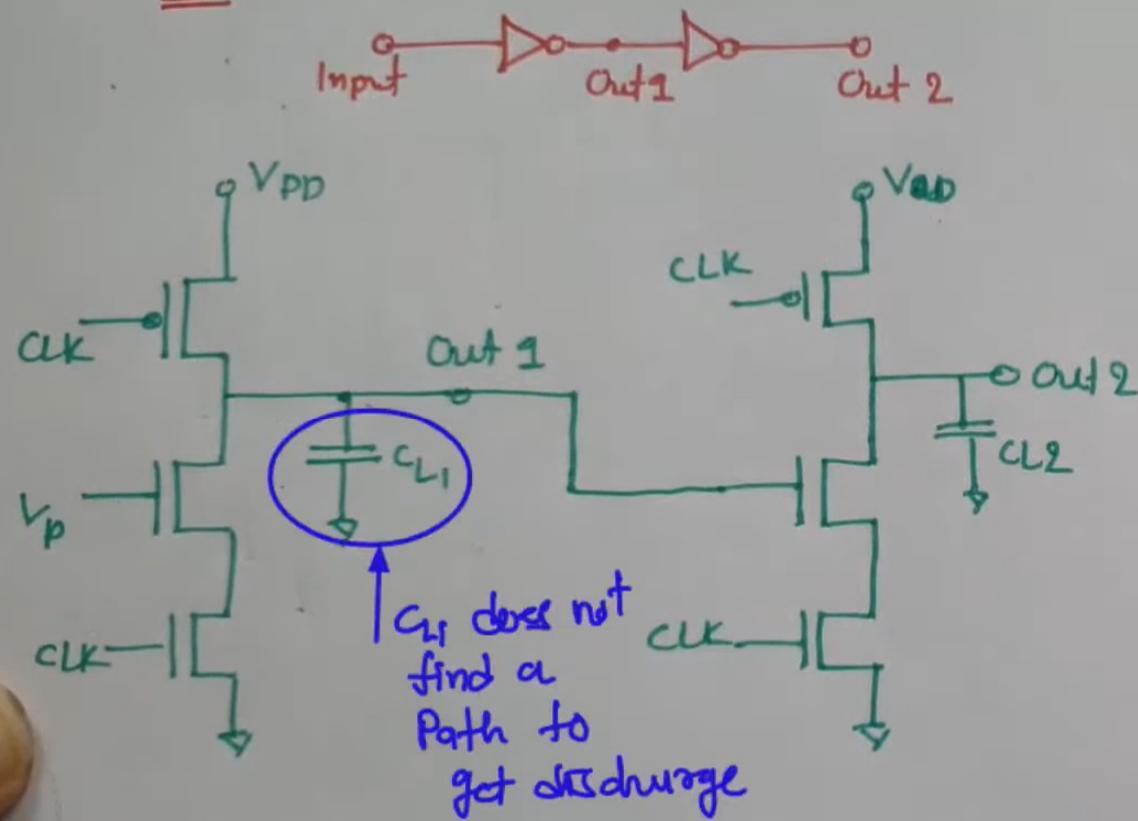
$\rightarrow$  Case



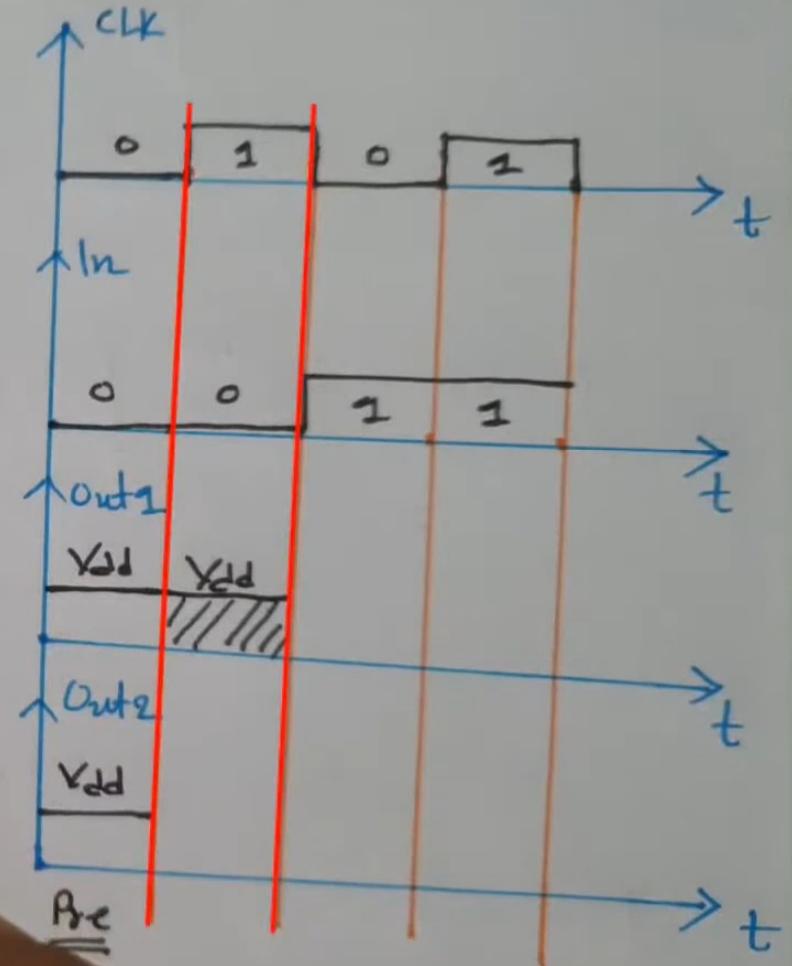
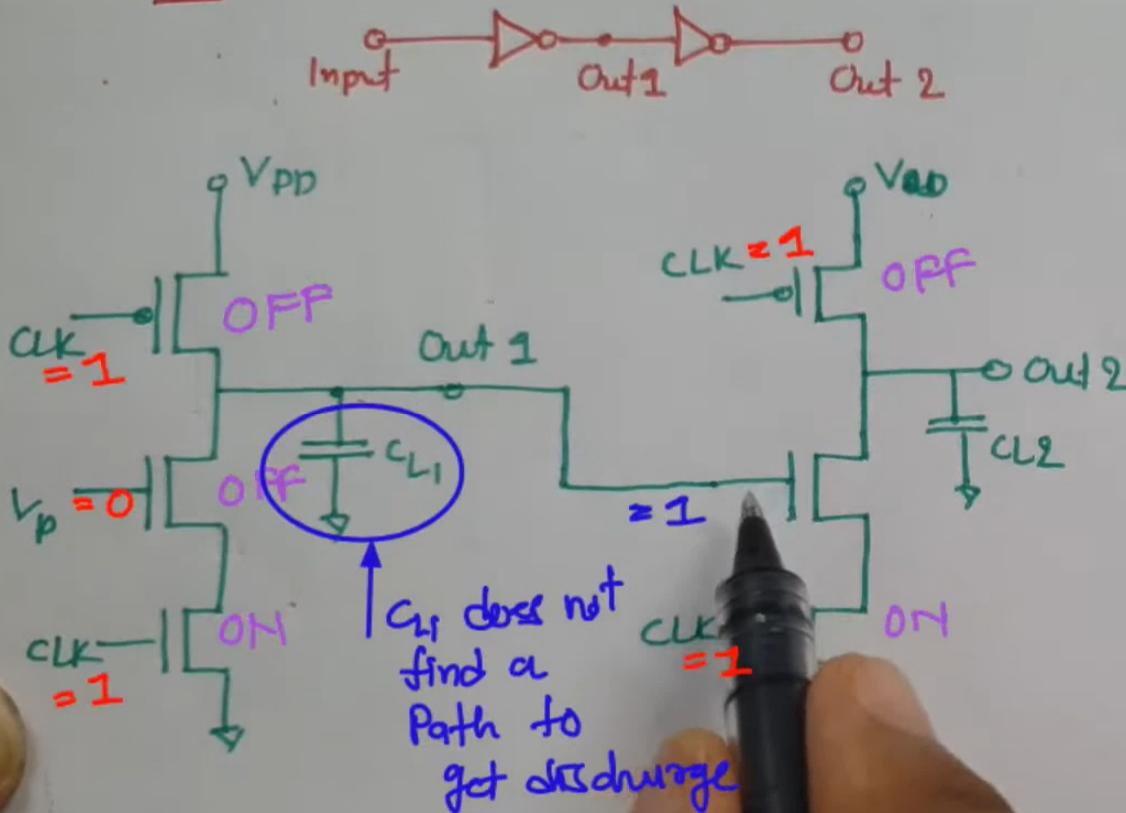
→ Case



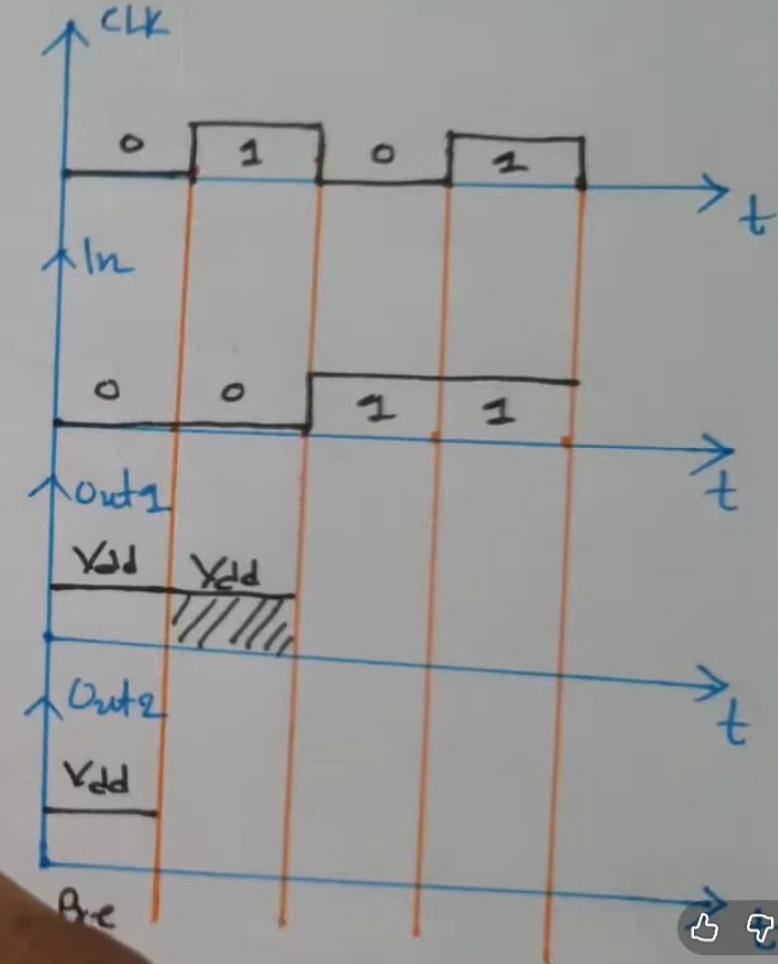
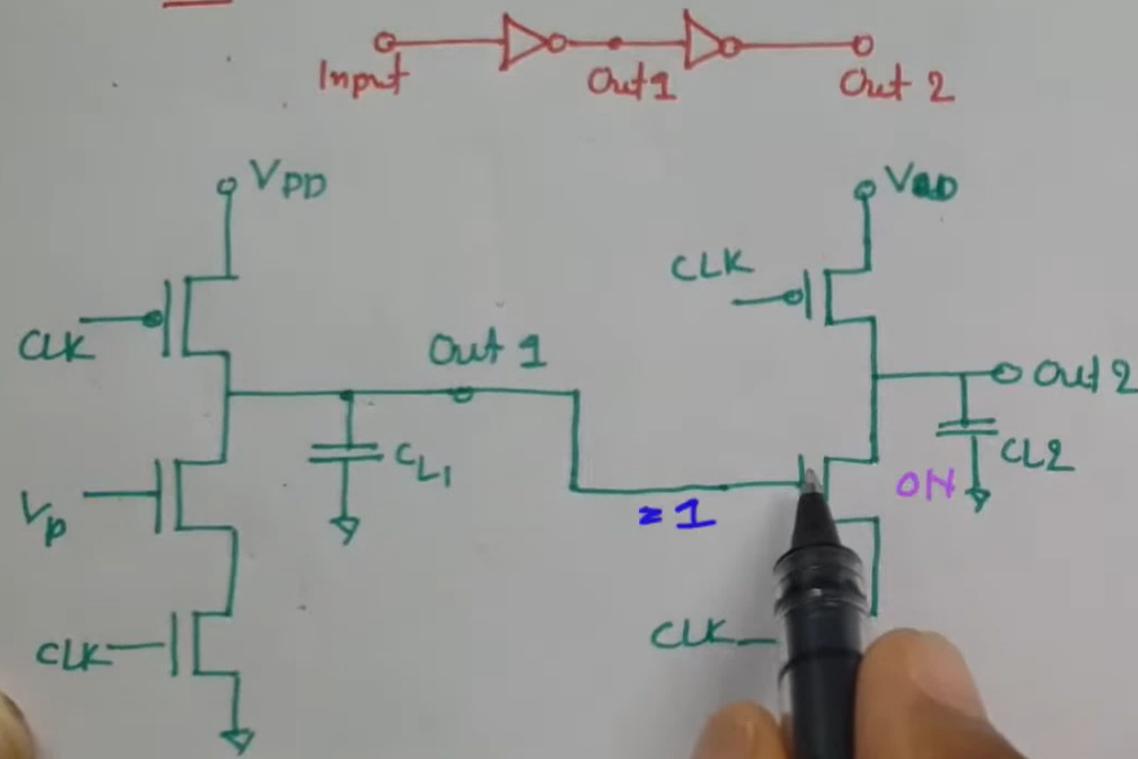
→ Cause



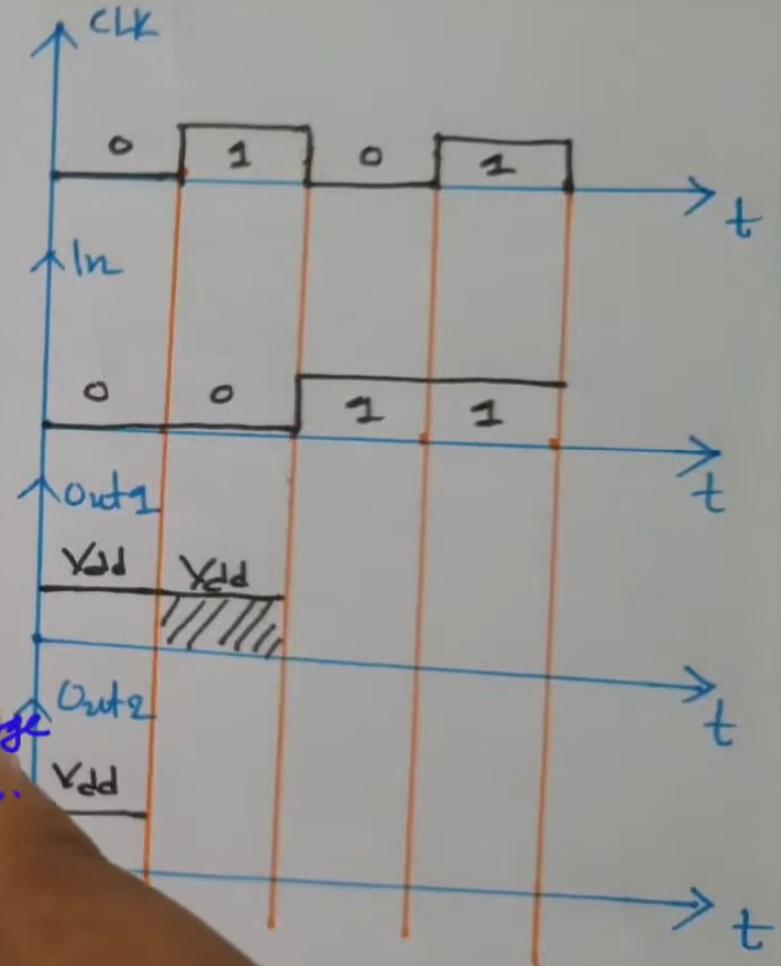
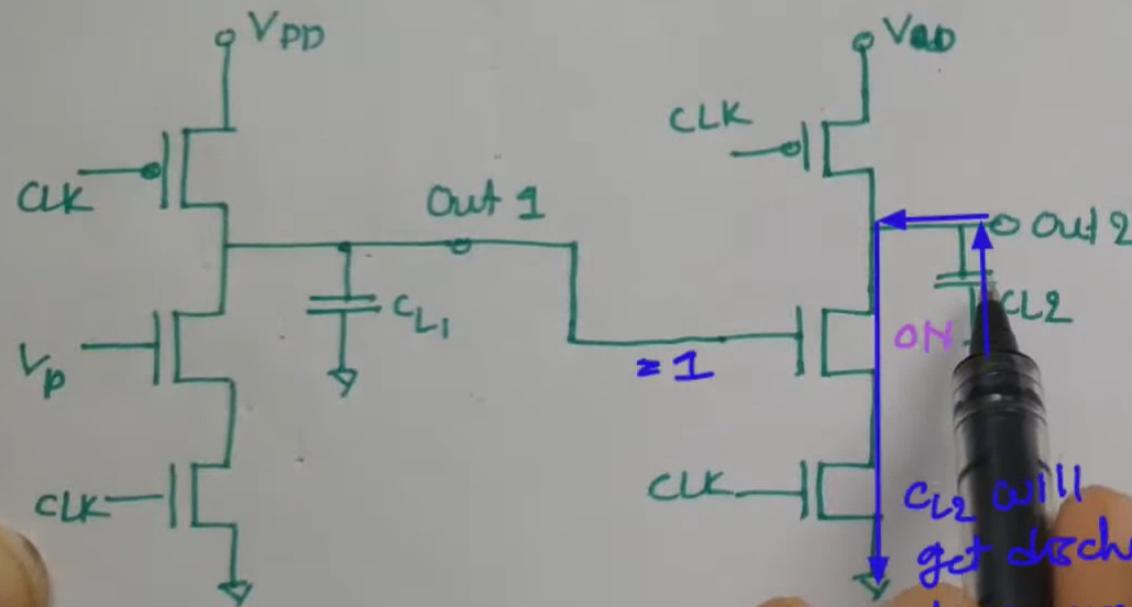
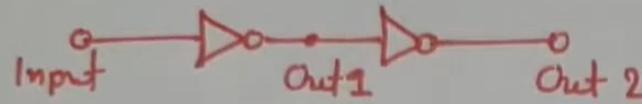
→ Case



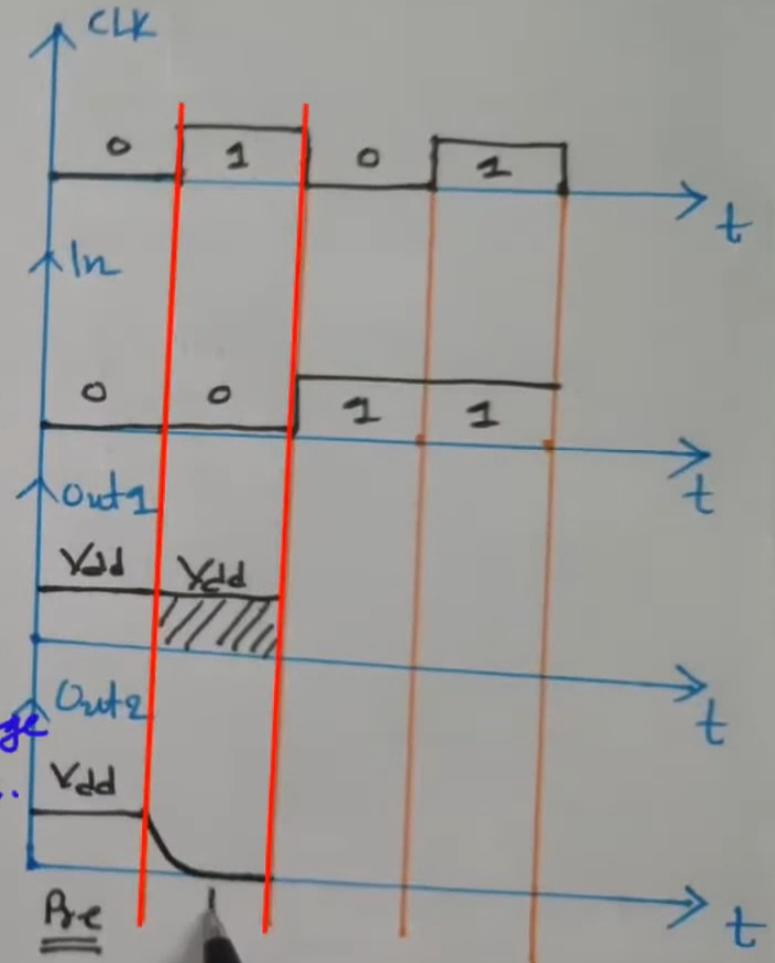
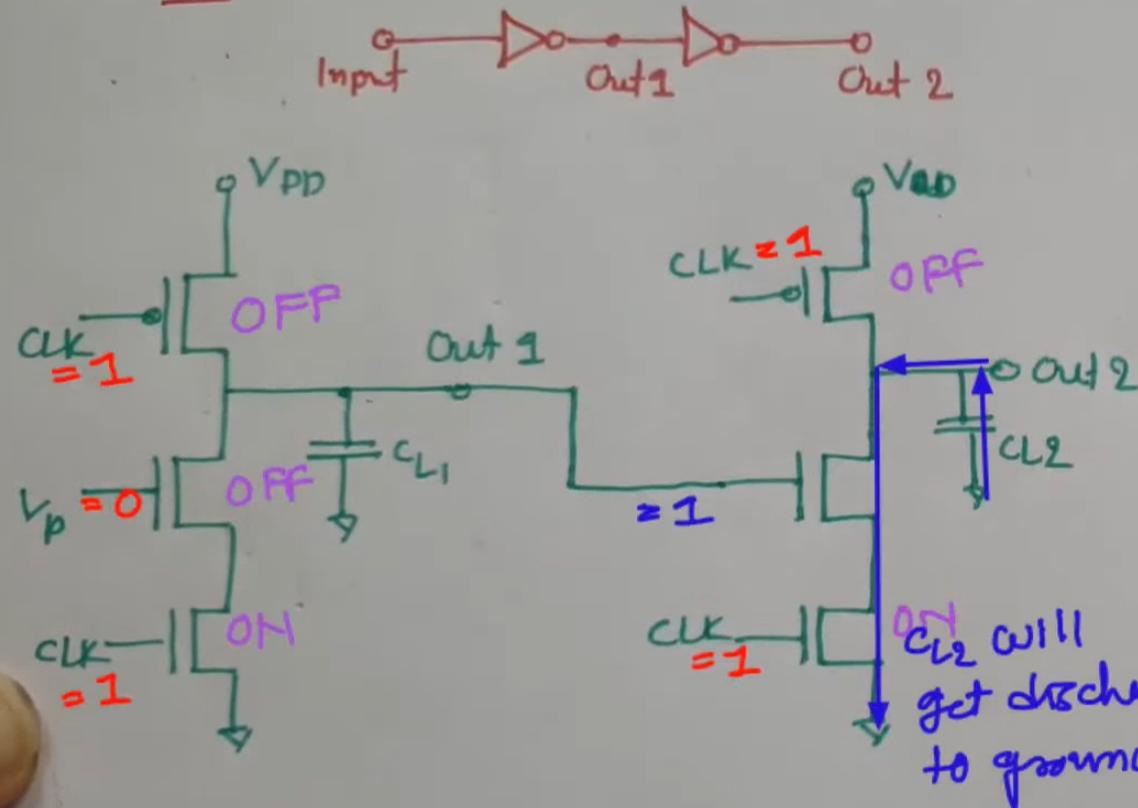
→ Case



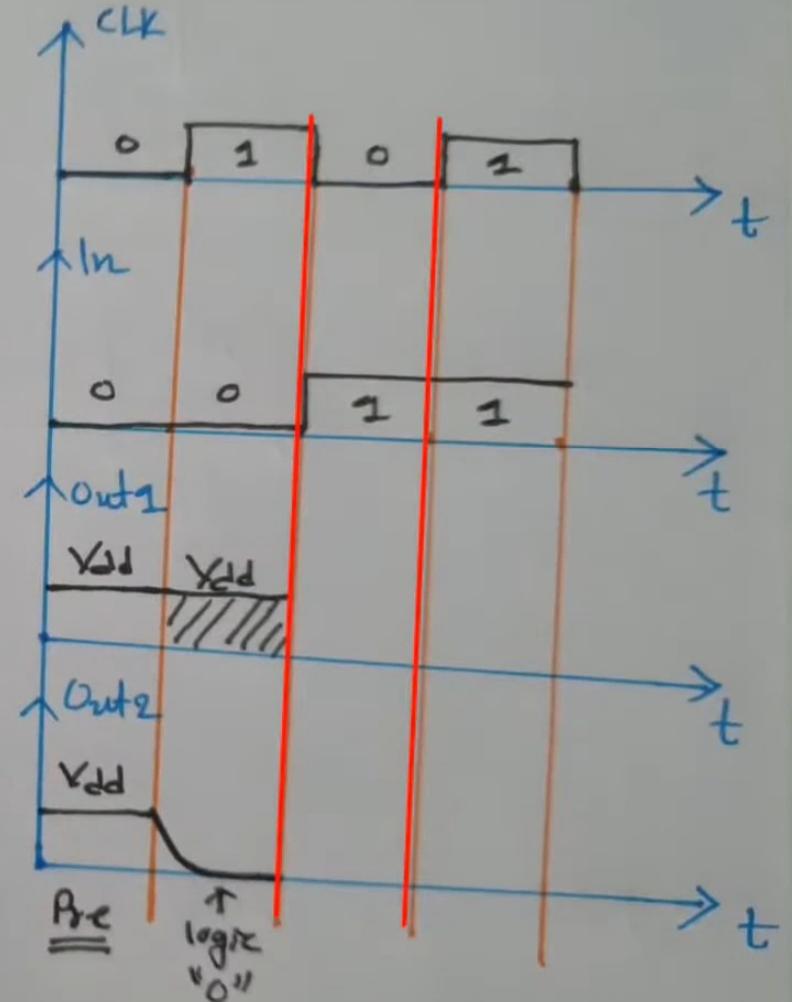
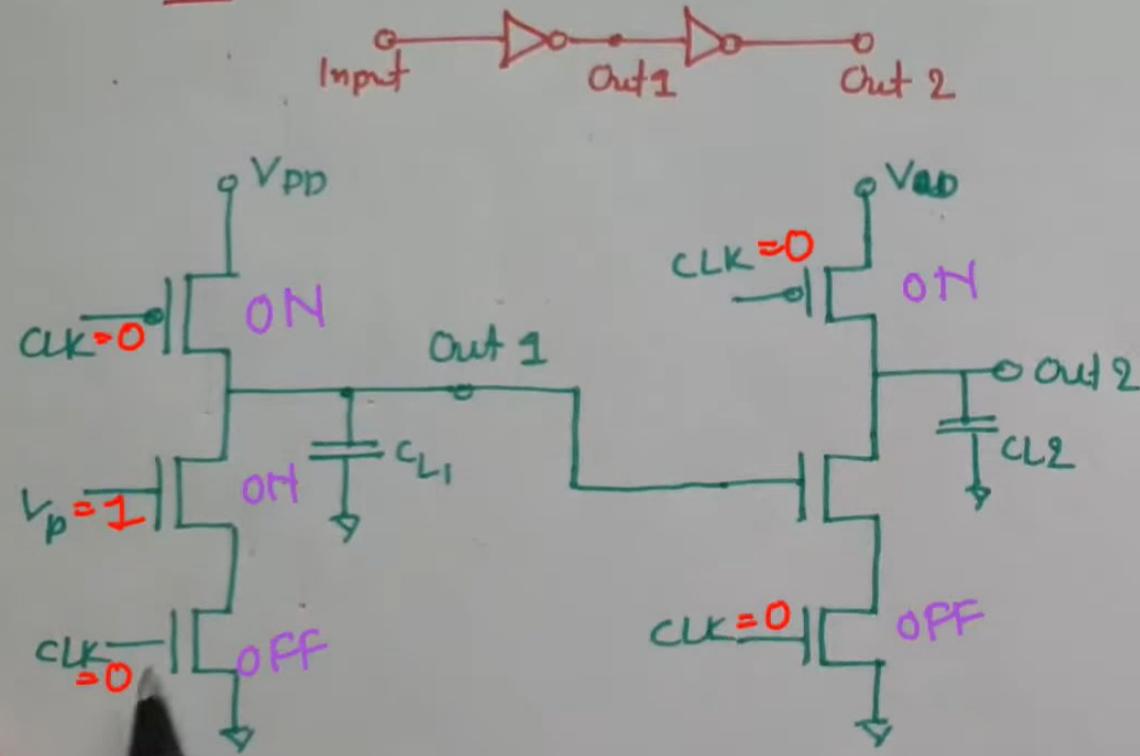
$\rightarrow$  Case



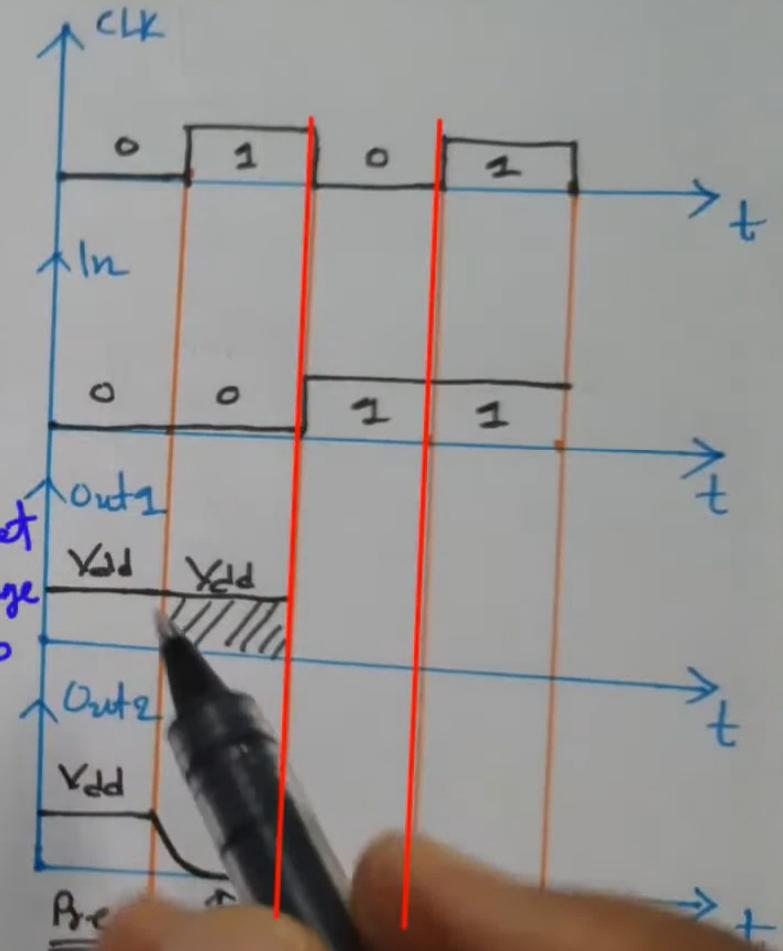
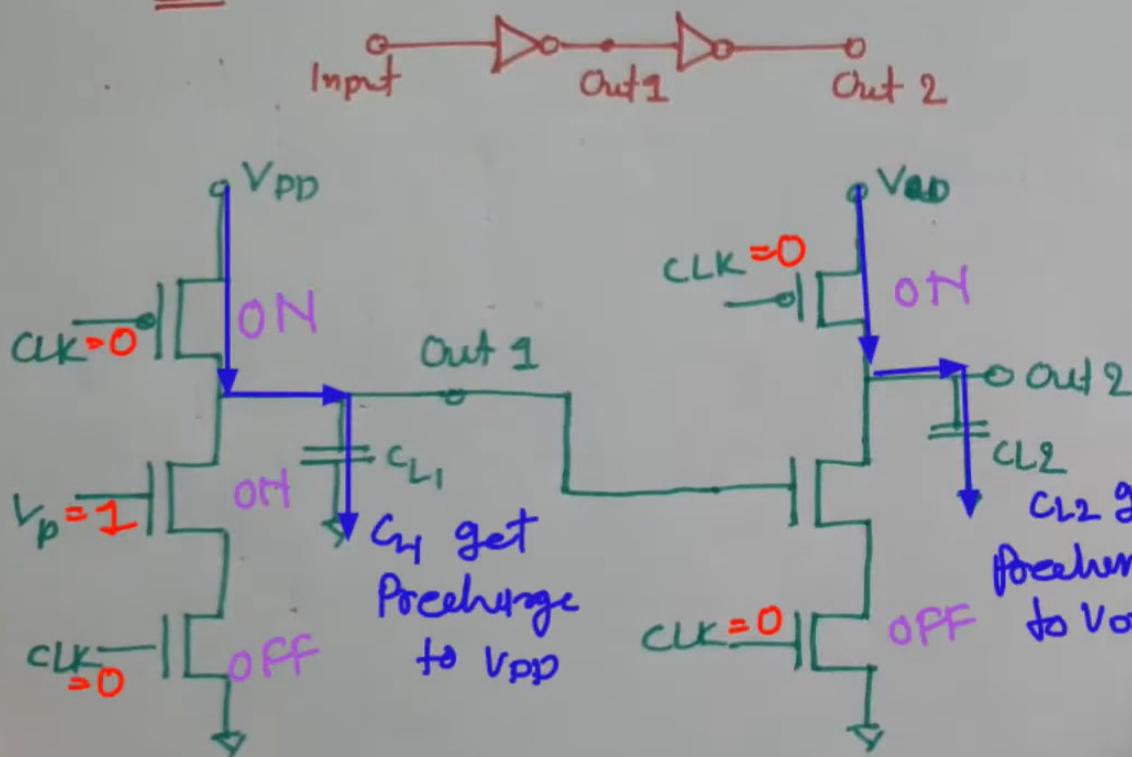
$\rightarrow$  Case



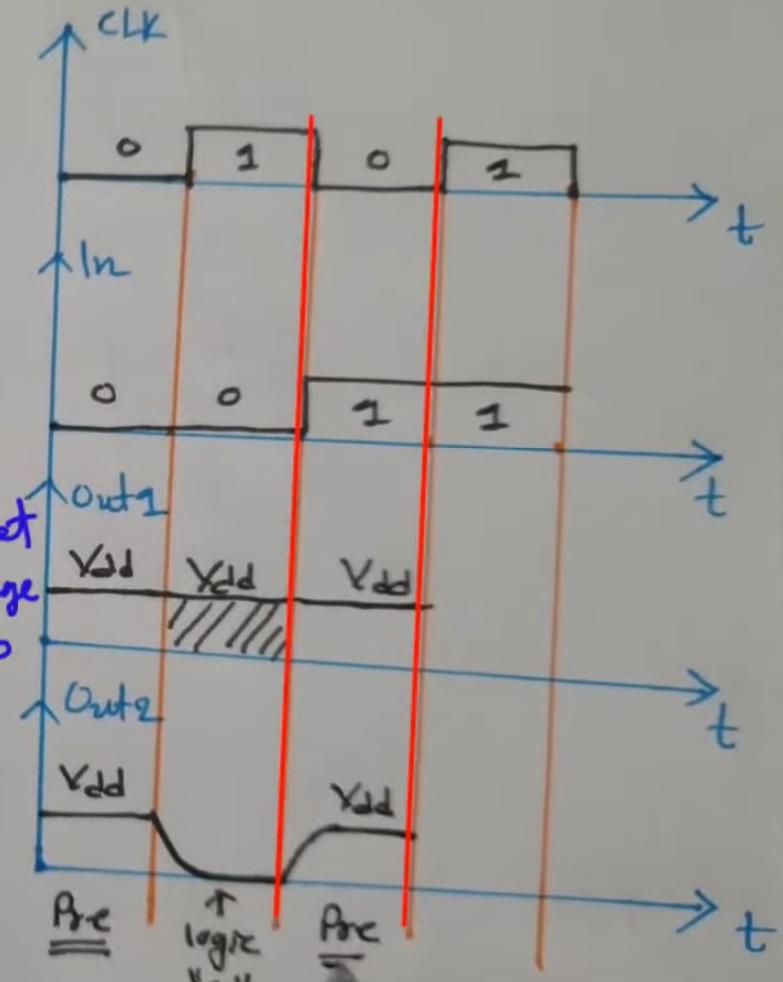
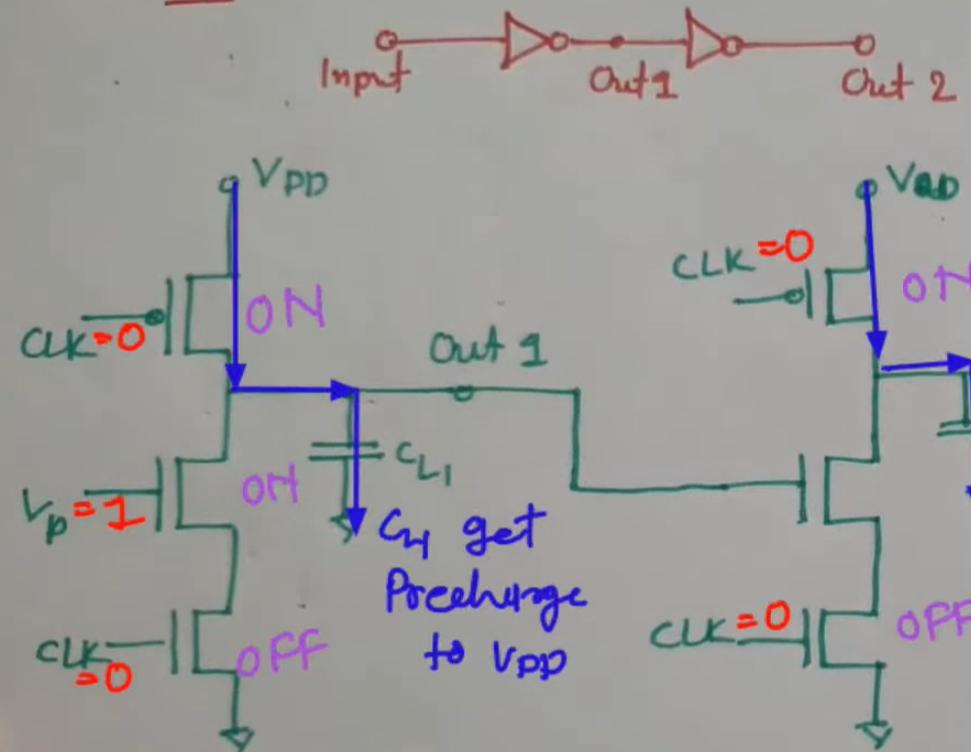
$\rightarrow$  Case



→ Case

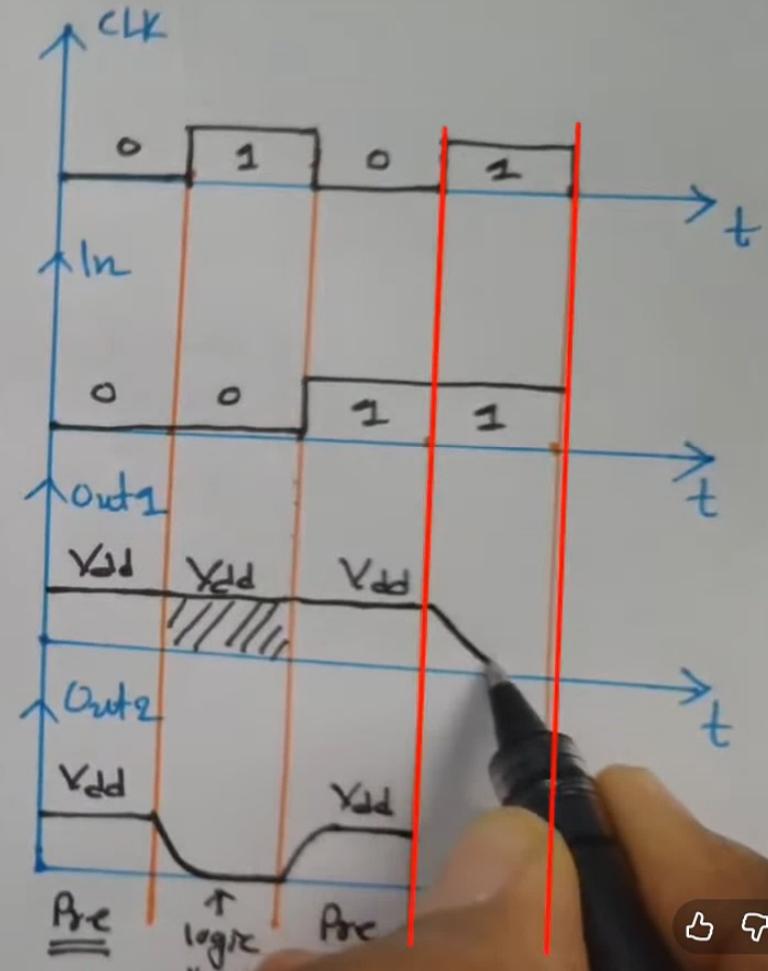
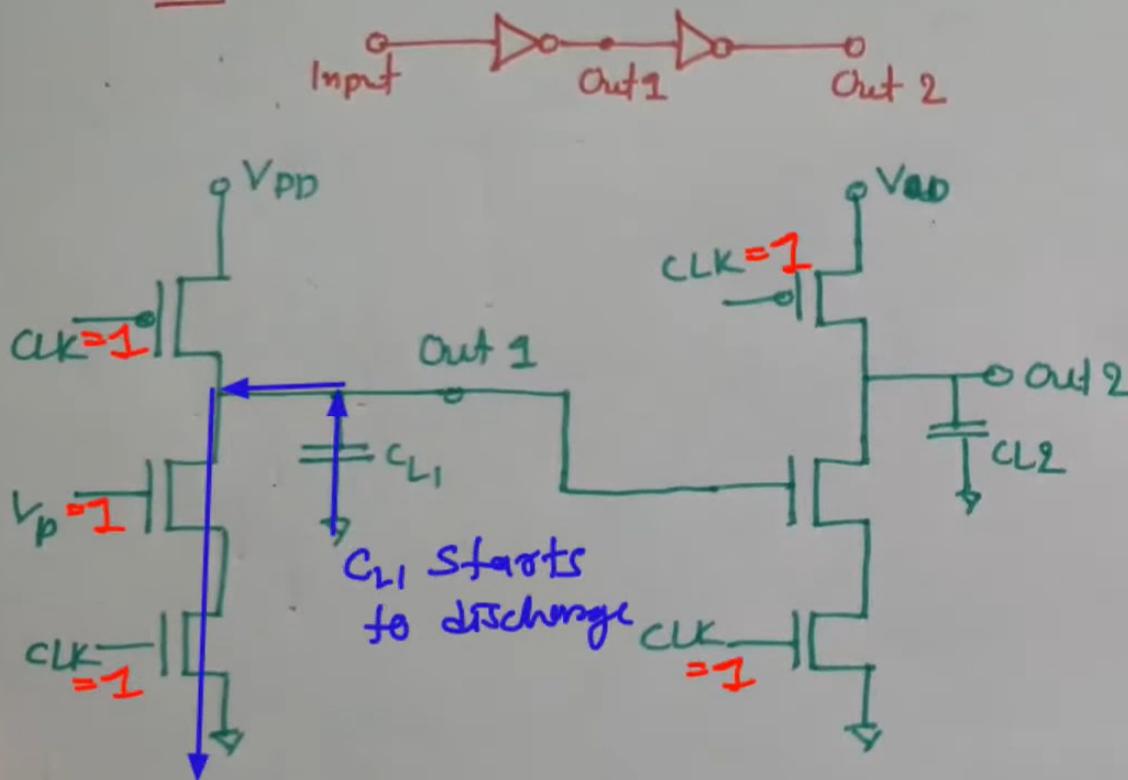


$\rightarrow$  Case



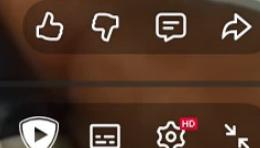
## Cascading Issues of Dynamic CMOS | Solution of Cascading Issues in Dynamic CMOS

$\rightarrow$  Cause

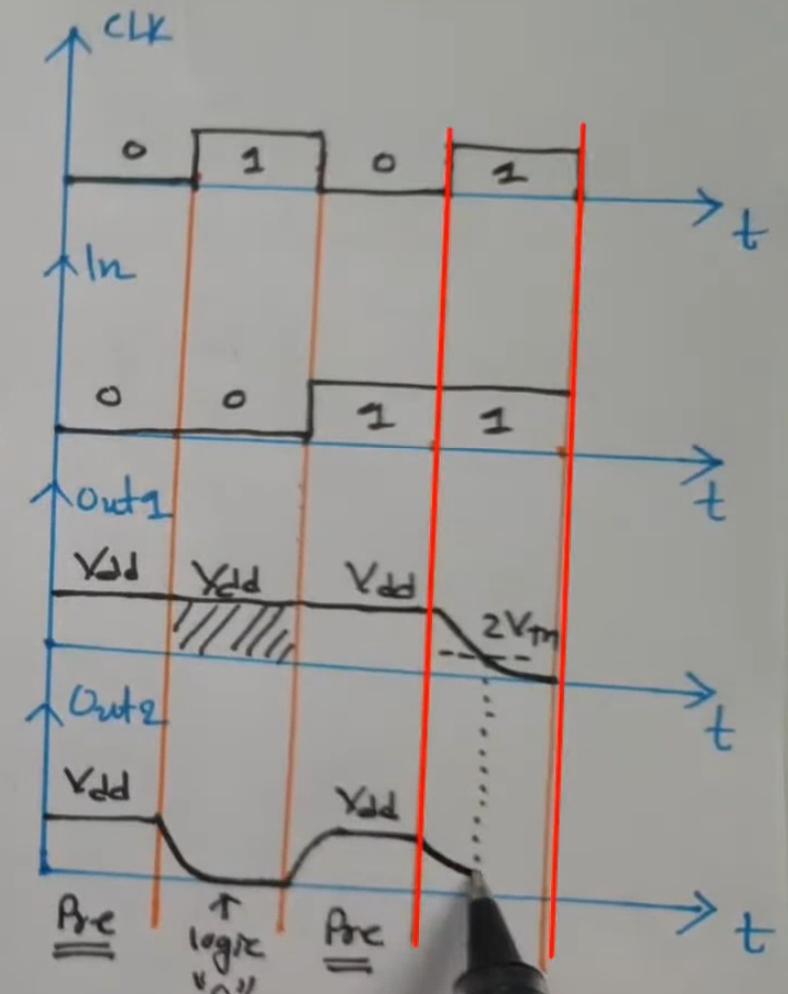
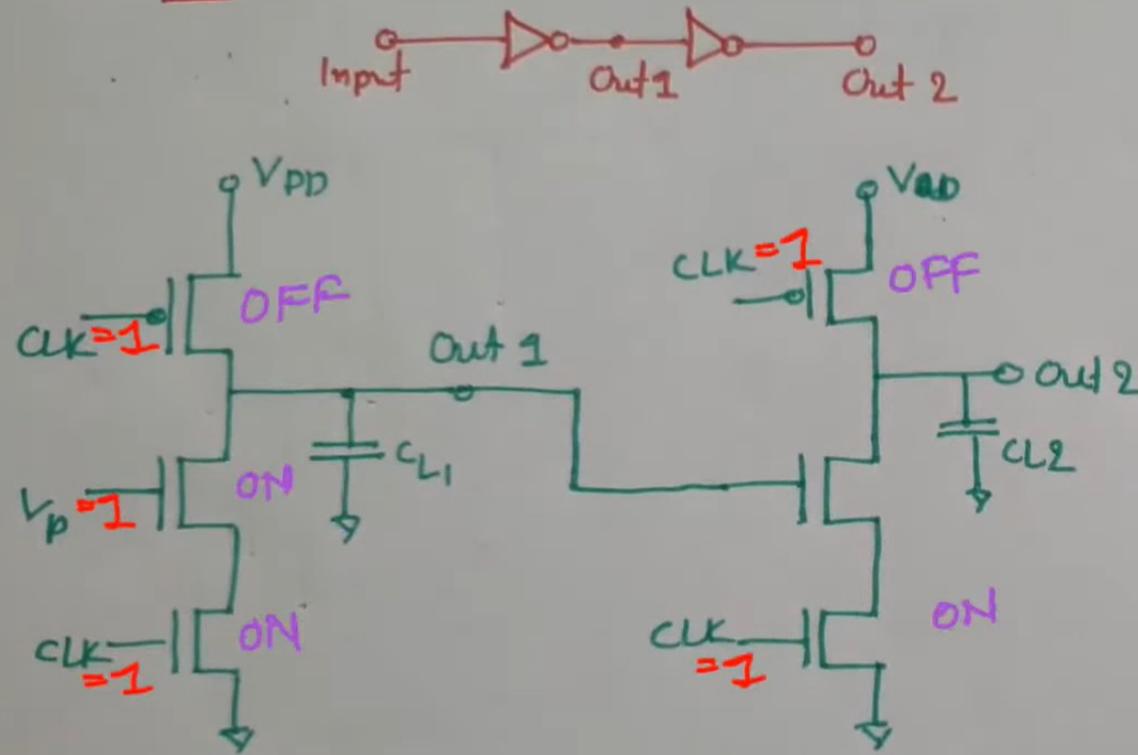


7:42 / 12:57

Timing Diagram of Cascading in Dynamic CMOS >

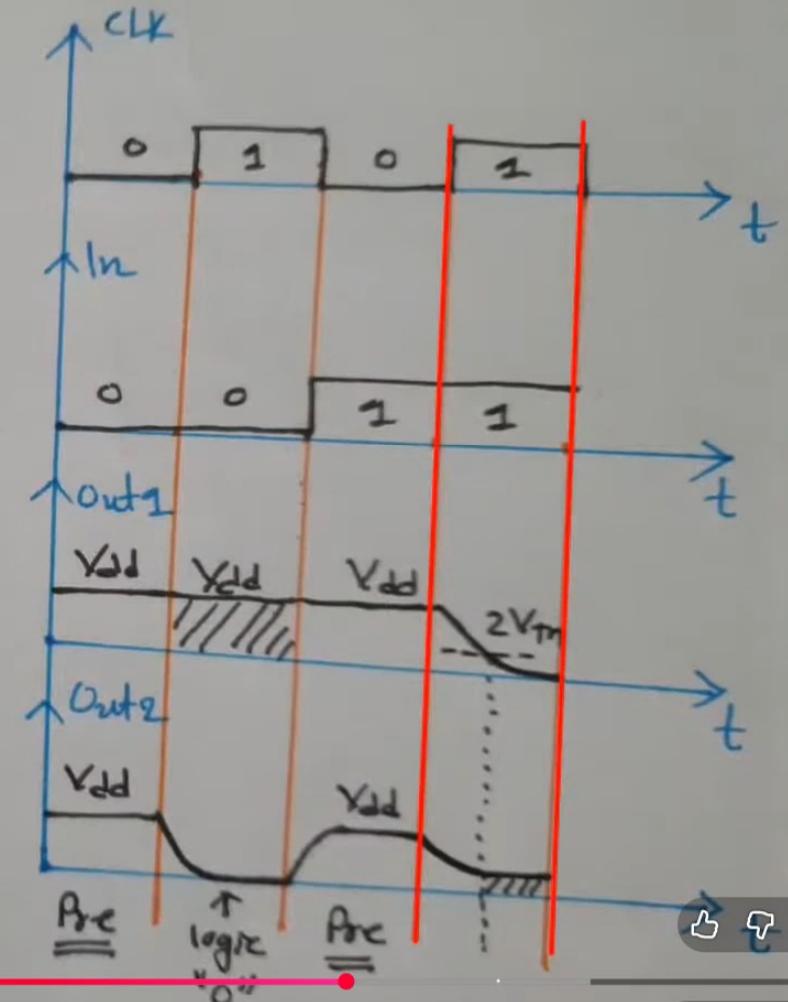
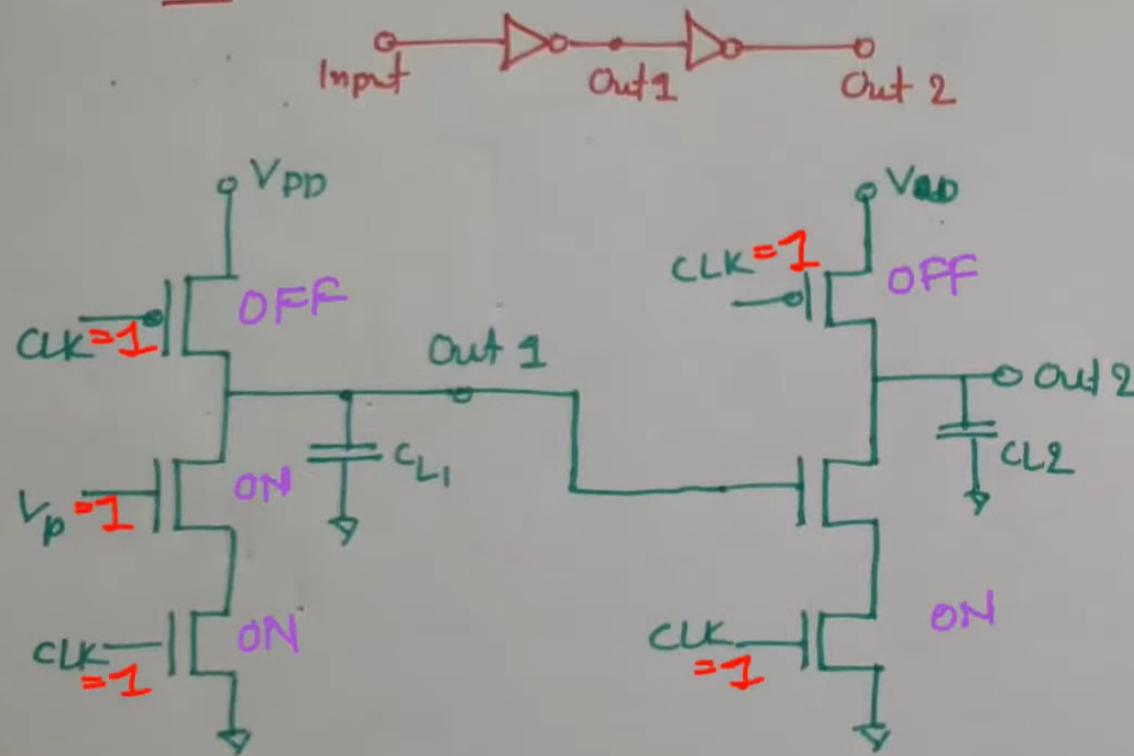


$\rightarrow$  Cose



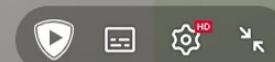
## Cascading Issues of Dynamic CMOS | Solution of Cascading Issues in Dynamic CMOS

$\rightarrow$  Case

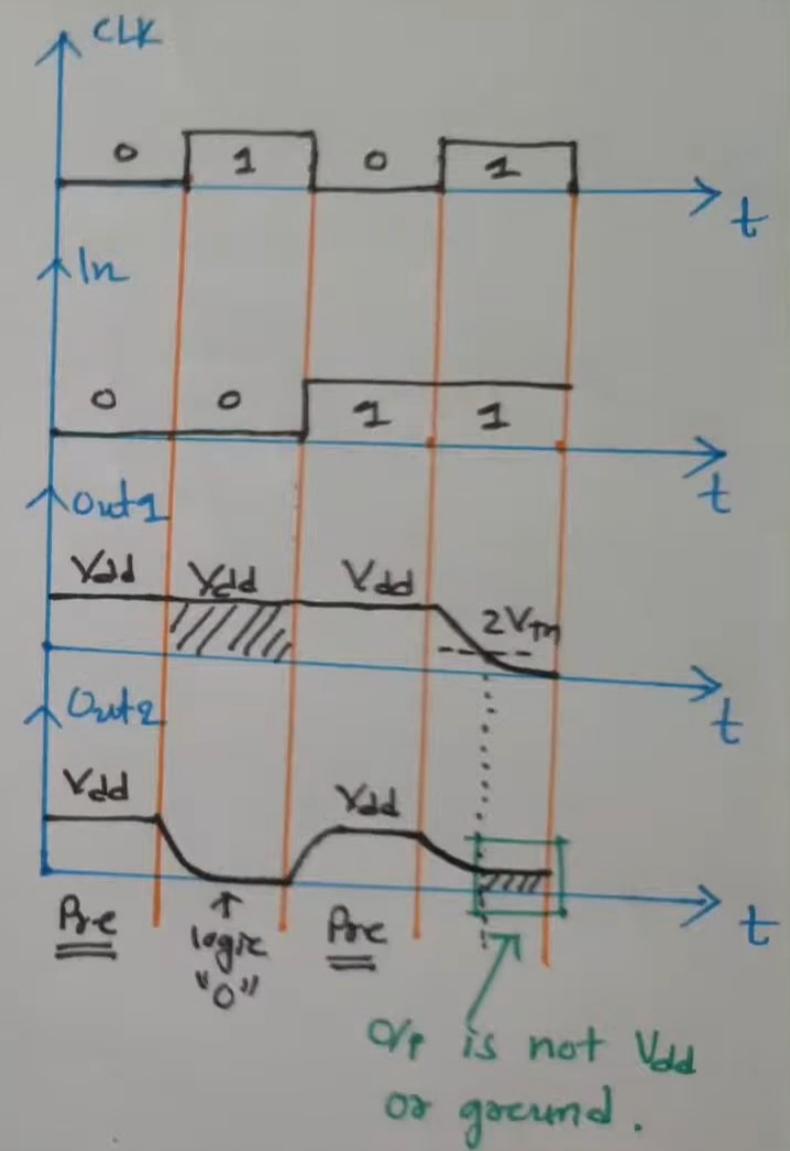
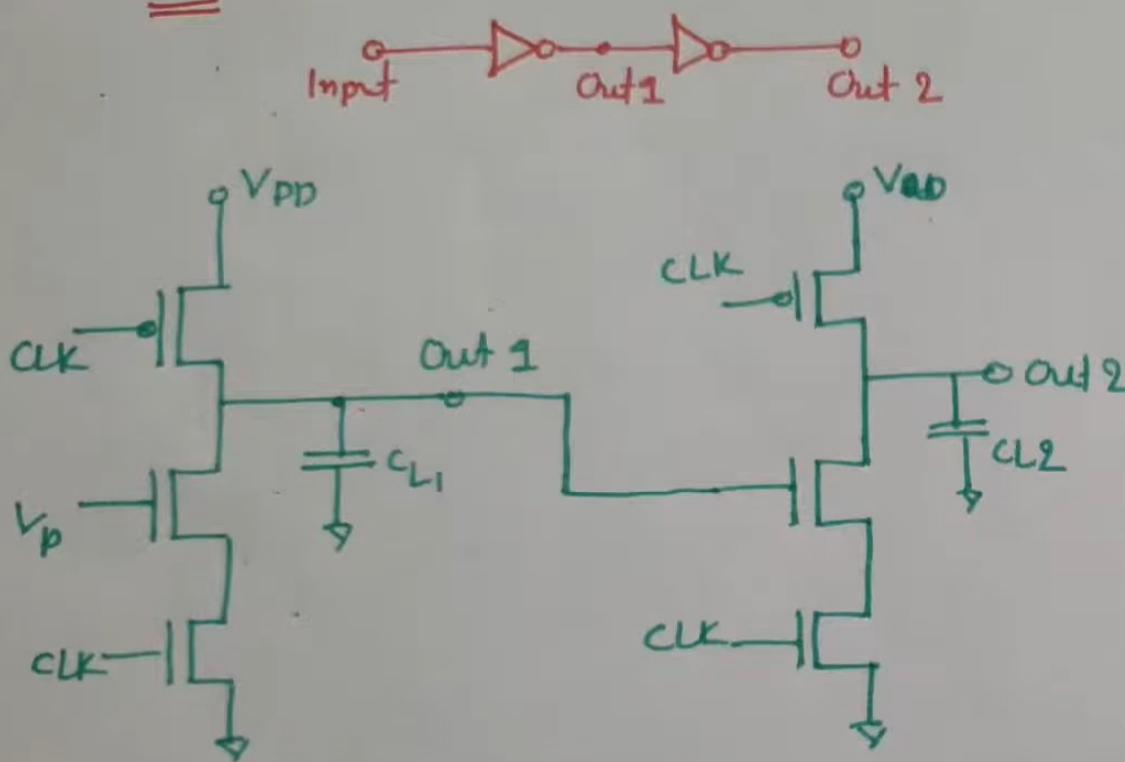


9:18 / 12:57

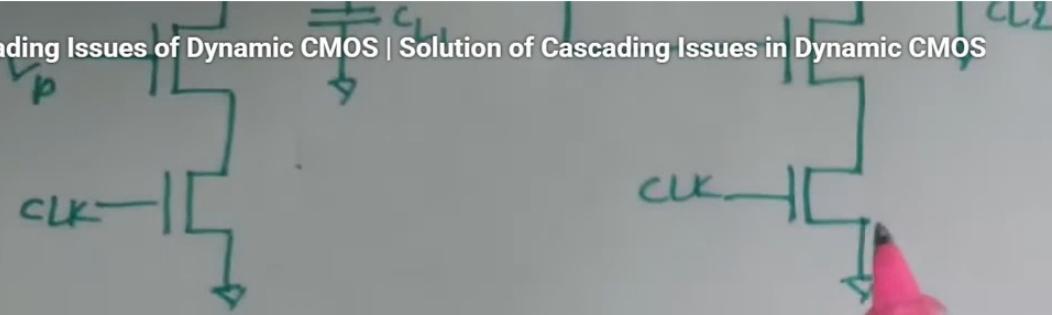
Timing Diagram of Cascading in Dynamic CMOS >



→ Cause

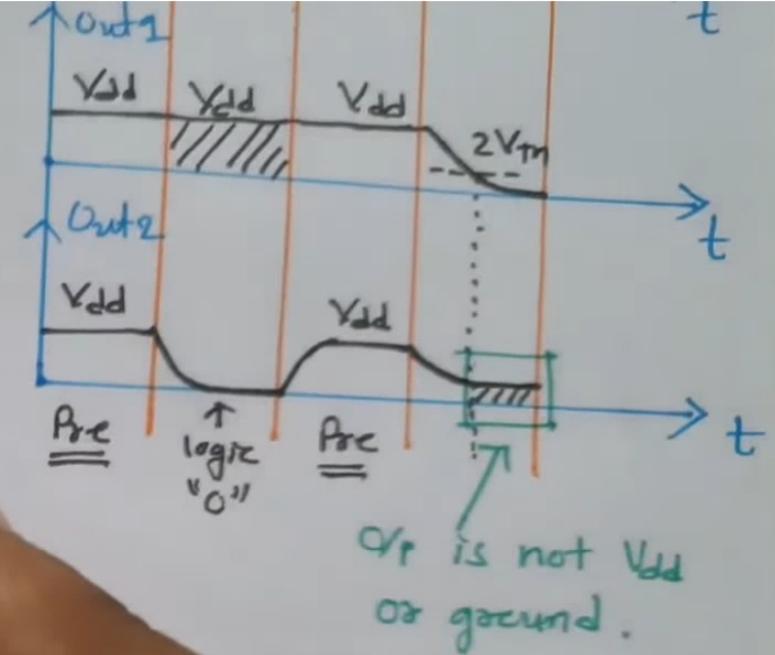


## Cascading Issues of Dynamic CMOS | Solution of Cascading Issues in Dynamic CMOS



→ Sol.

- Domino logic CMOS
- NORCA Logic CMOS
- Retiming of clock.



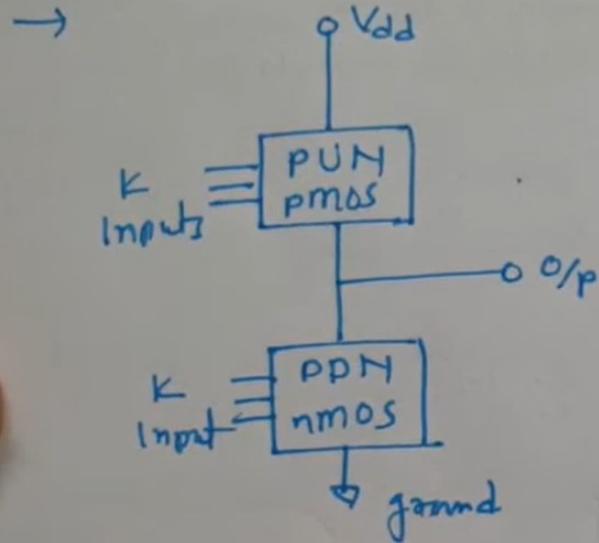
11:26 / 12:57

Solution of Cascading Issues in Dynamic CMOS >

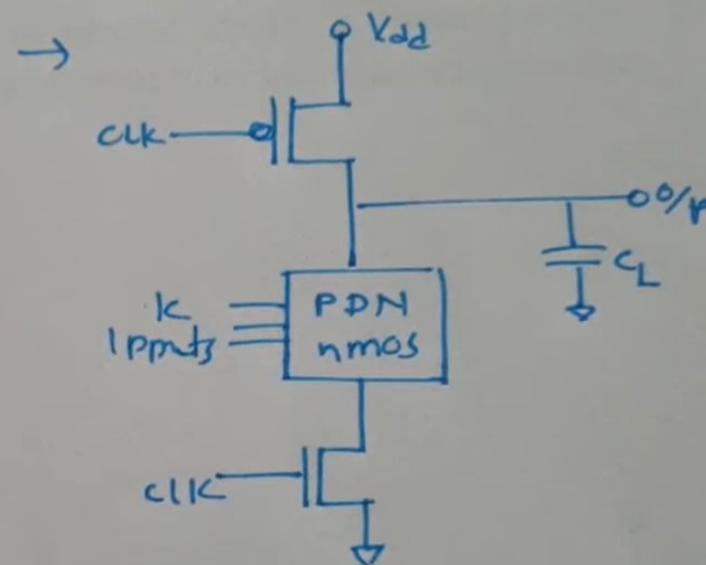


## Comparison of static and Dynamic CMOS

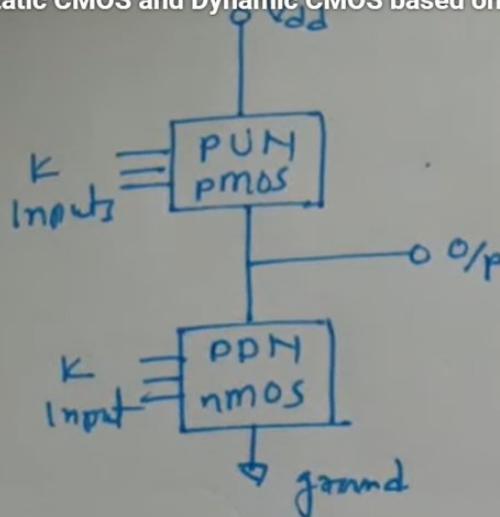
### Static CMOS



### Dynamic CMOS



→ nMOS =

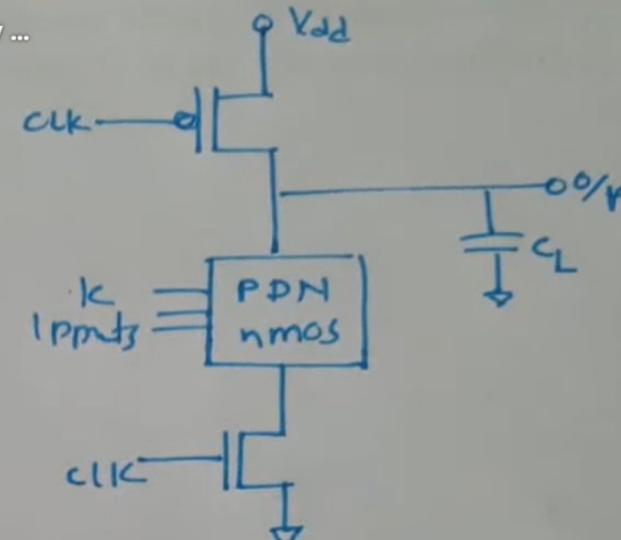


$$\rightarrow n\text{MOS} = K$$

$$p\text{MOS} = K$$

$\rightarrow$  Due to pmos overall size is more.

$\rightarrow$  Due to more pmos, capacitive loading is more.

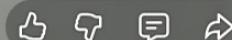


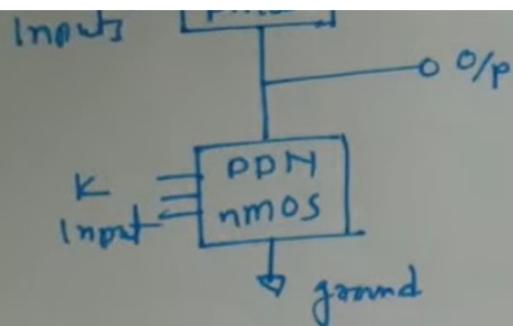
$$n\text{MOS} = K + 1$$

$$p\text{MOS} = 1$$

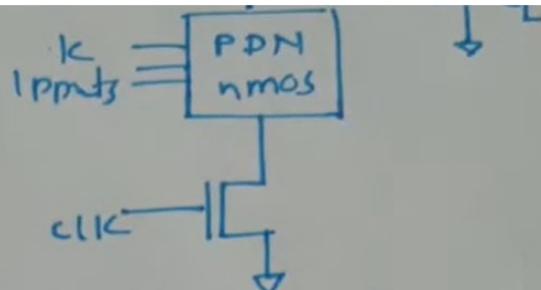
$\rightarrow$  Due to only one pmos overall size less.

$\rightarrow$  Due to less pmos, capacitive loading is less.





- $nMOS = K$
- $pMOS = K$
- Due to pmos overall size is more.
- Due to more pmos, capacitive loading is more.
- Slower switching chro.
- No cascading issues.
- No need of precharge for operation.



- $nMOS = K + 1$
- $pMOS = 1$
- Due to only one pmos overall size less.
- Due to less pmos, capacitive loading is less.
- faster switching chro.
- In Cascading, some issues.
- Pre-charge of load  $C_L$  is req'd for dynamic CMOS operation.

- Due to more pMOS, capacitive loading is more.
  - Slower switching char.
  - No cascading issues.
  - No need of precharge for operation.
  - It has no such issues.
  - No charge leakage.
- Due to less pMOS, capacitive loading is less.
- faster switching char.
  - In cascading, some issues.
  - Pre-charge of load  $C_L$  is req'd for dynamic CMOS operation.
  - If o/p = logic '1', o/p stage in high impedance, so noise immunity is less.
  - Charge leakage creates problems



5:58 / 7:05

Leakage of charge in Static and Dynamic CMOS >



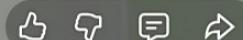
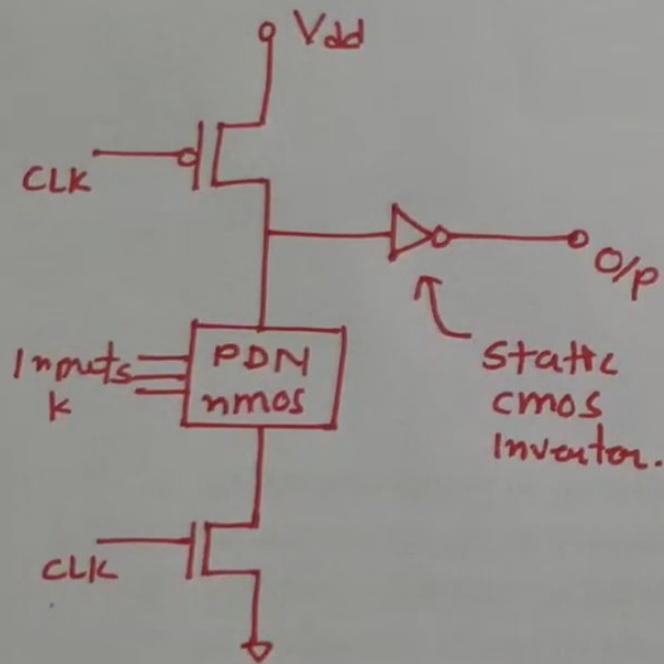
## Domino logic cmos

- In static CMOS, we use PDN made up of pMOS. As pMOS has more size, higher load capacitance & slower switching speed, we use Dynamic CMOS.
- But, in Dynamic CMOS, there are issues of cascading.
- Sol'n to Cascading problem can be done by domino CMOS.



Domino Logic CMOS (Basics, Circuit, Number of Transistors & Working) Explained.

- But we use Dynamic CMOS, There are issues of cascading.
- Sol'n to Cascading problem can be done by domino CMOS.



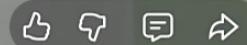
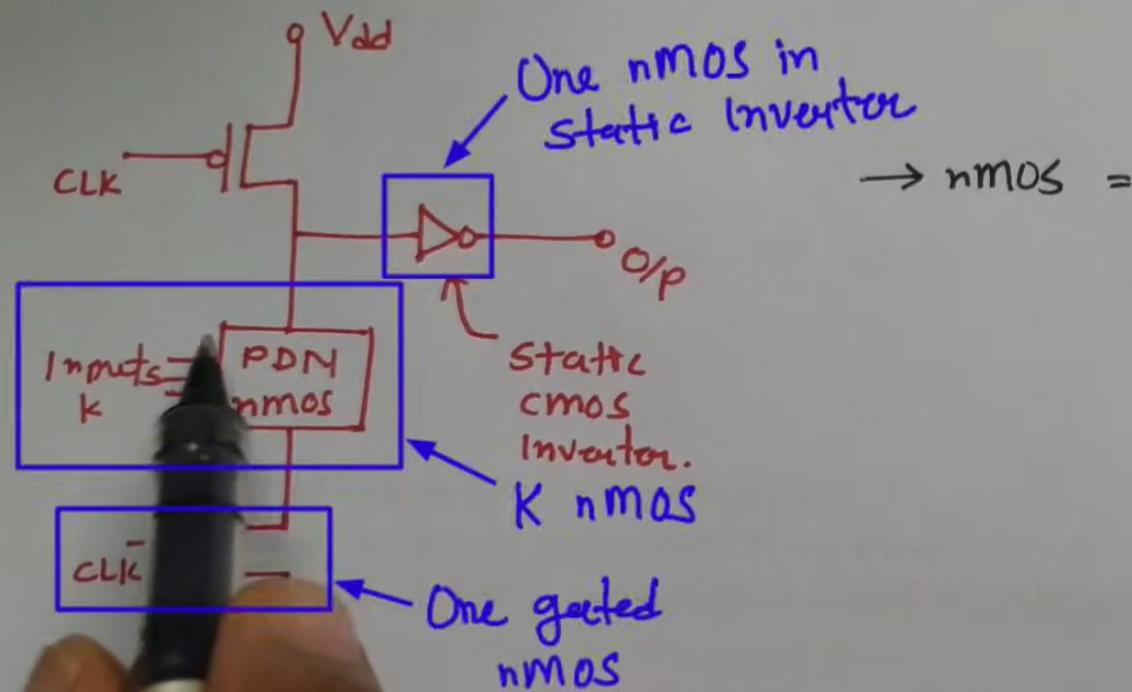
2:23 / 12:19

Number of Transistors in Domino Logic CMOS &gt;



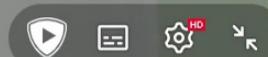
## Domino Logic CMOS (Basics, Circuit, Number of Transistors & Working) Explained

- But, we have Dynamic CMOS, There are issues of cascading.
- Sol'n to Cascading problem can be done by domino CMOS.



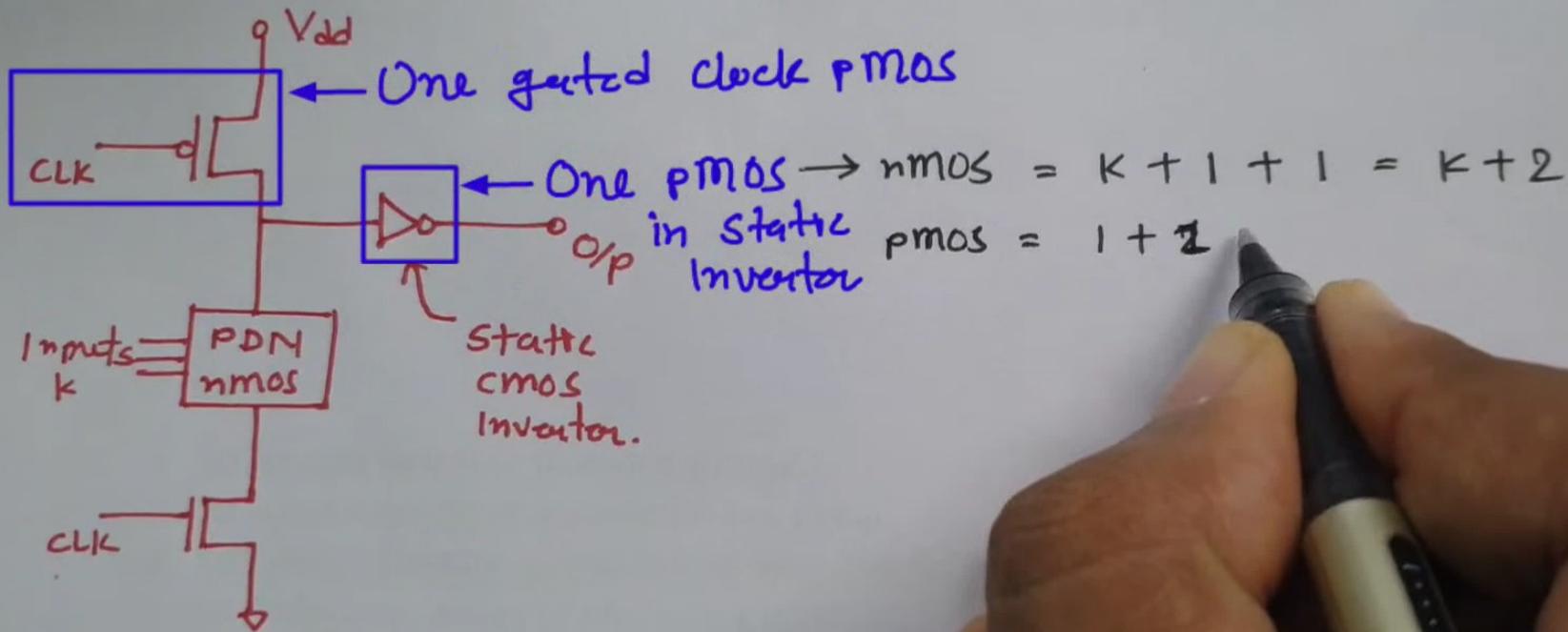
2:35 / 12:19

Number of Transistors in Domino Logic CMOS &gt;



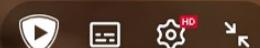
## Domino Logic CMOS (Basics, Circuit, Number of Transistors & Working) Explained

- But we have Dynamic CMOS, There are issues of cascading.
- Sol'n to Cascading problem can be done by domino CMOS.



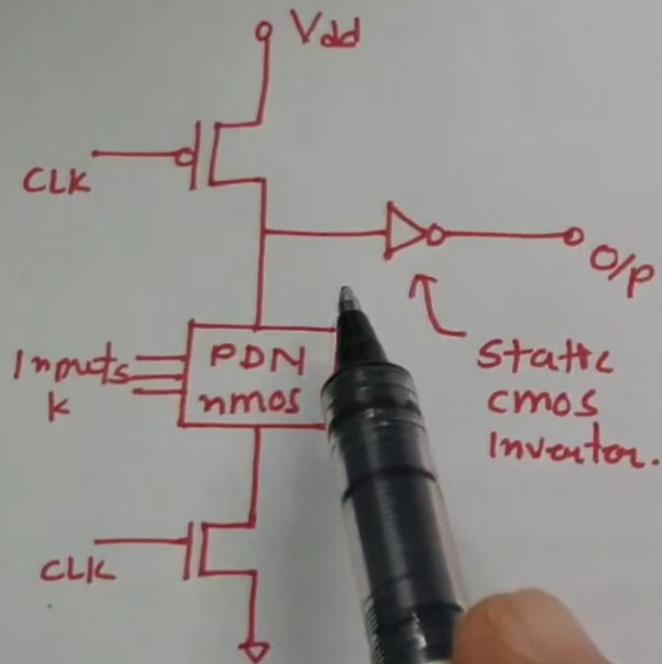
3:13 / 12:19

Number of Transistors in Domino Logic CMOS &gt;



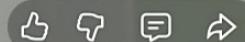
## Domino Logic CMOS (Basics, Circuit, Number of Transistors & Working) Explained

- But we have Dynamic CMOS, There are issues of cascading.
- Sol'n to Cascading problem can be done by domino CMOS.



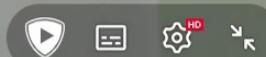
$$\rightarrow nMOS = k + 1 + 1 = k + 2$$

$$pMOS = 1 + 1 = 2$$

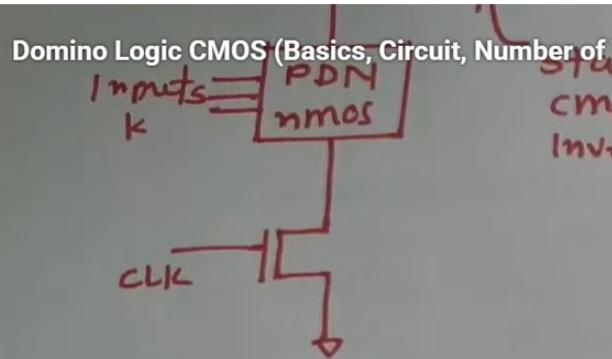


3:23 / 12:19

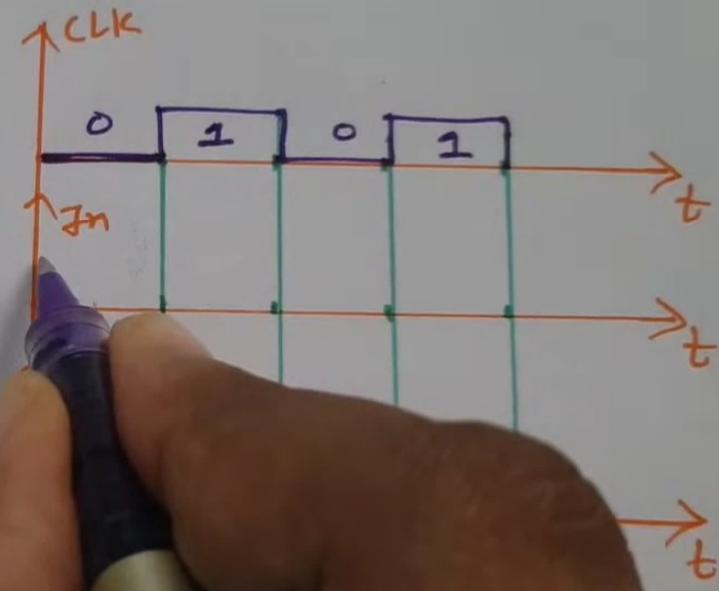
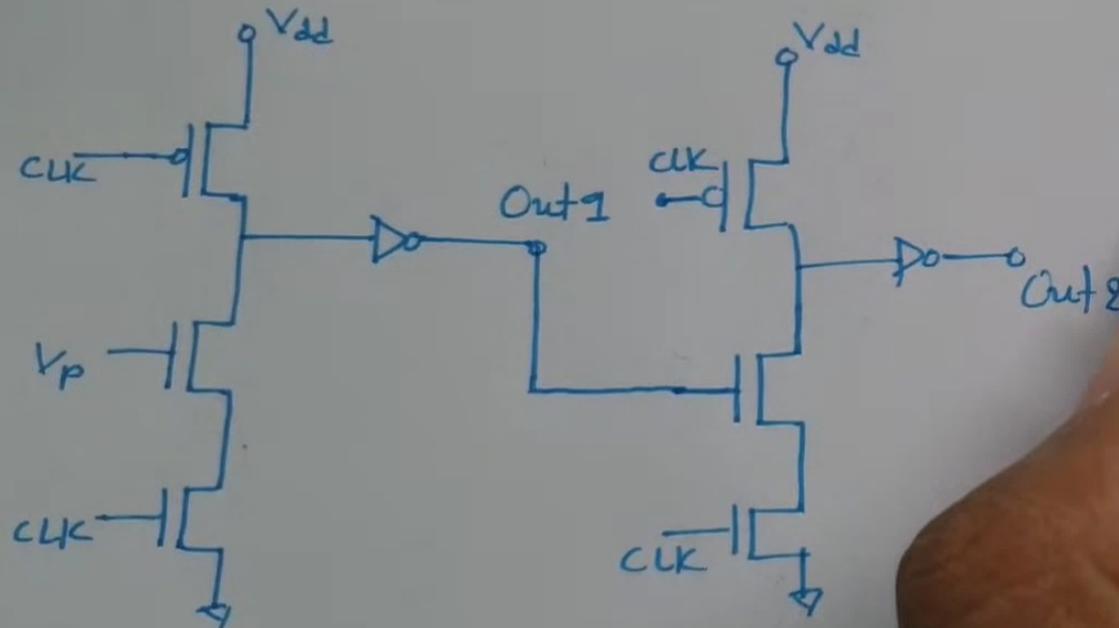
Number of Transistors in Domino Logic CMOS >



## Domino Logic CMOS (Basics, Circuit, Number of Transistors & Working) Explained



static  
cmos  
inverter.

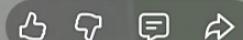
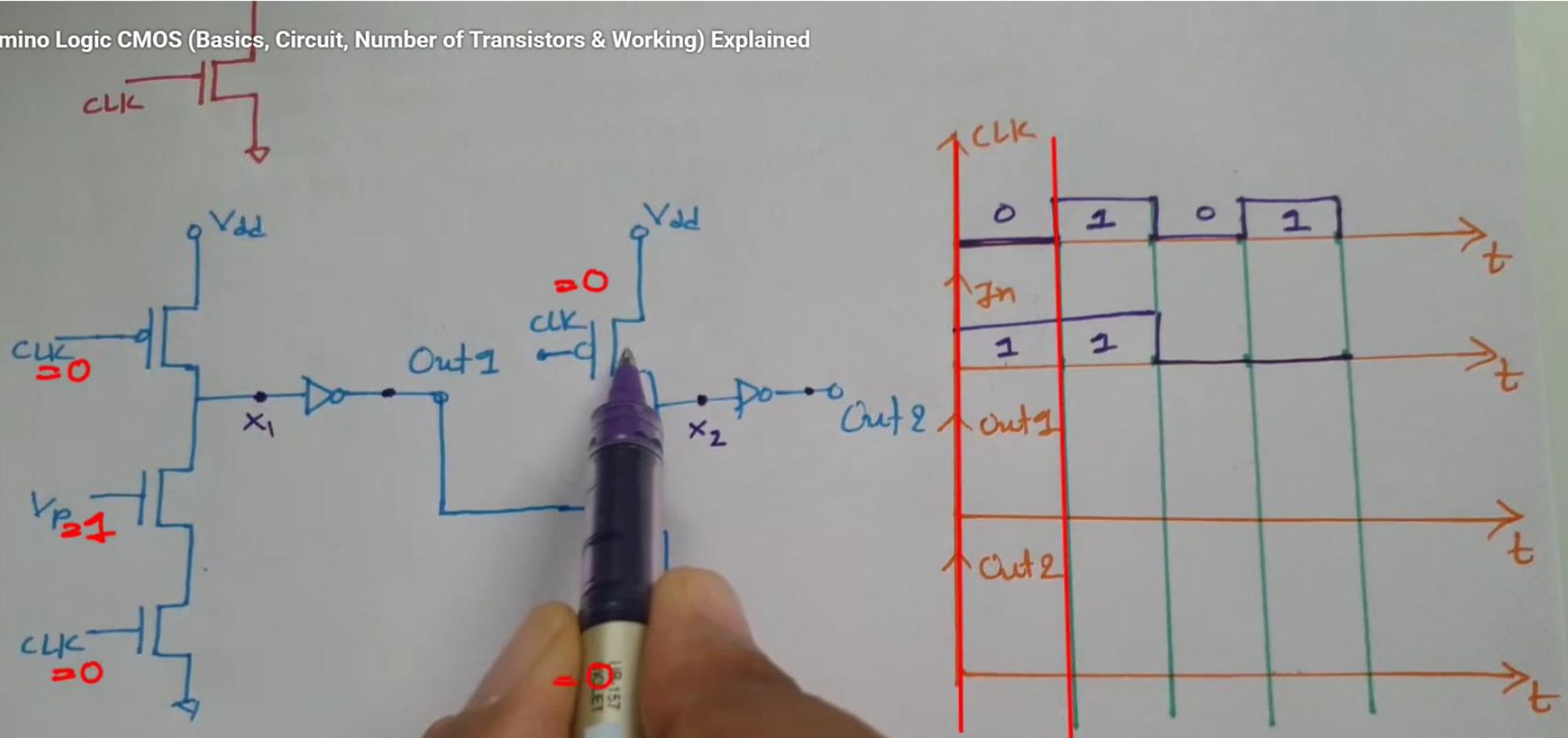


5:42 / 12:19

Working of Domino Logic CMOS >



## Domino Logic CMOS (Basics, Circuit, Number of Transistors & Working) Explained

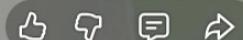
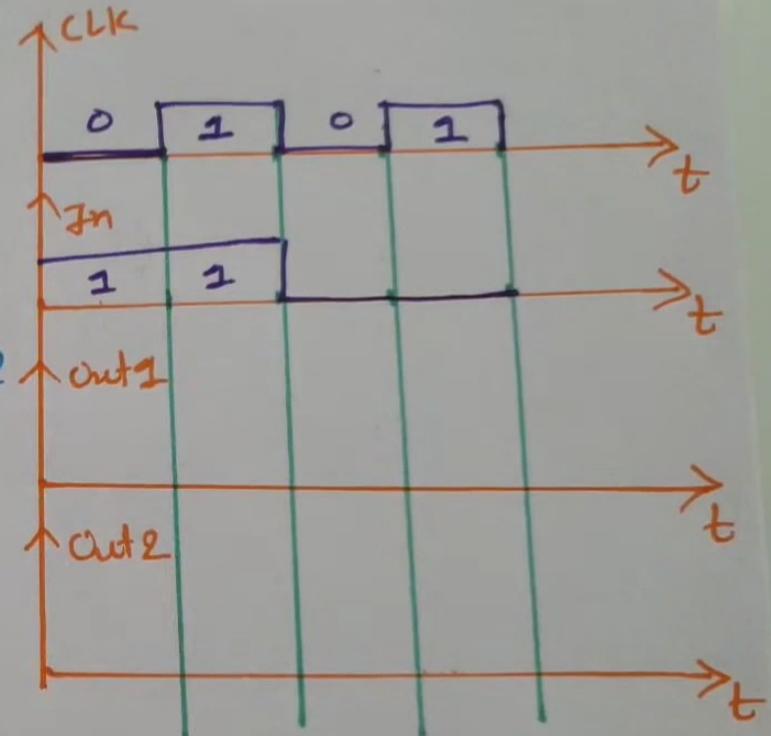
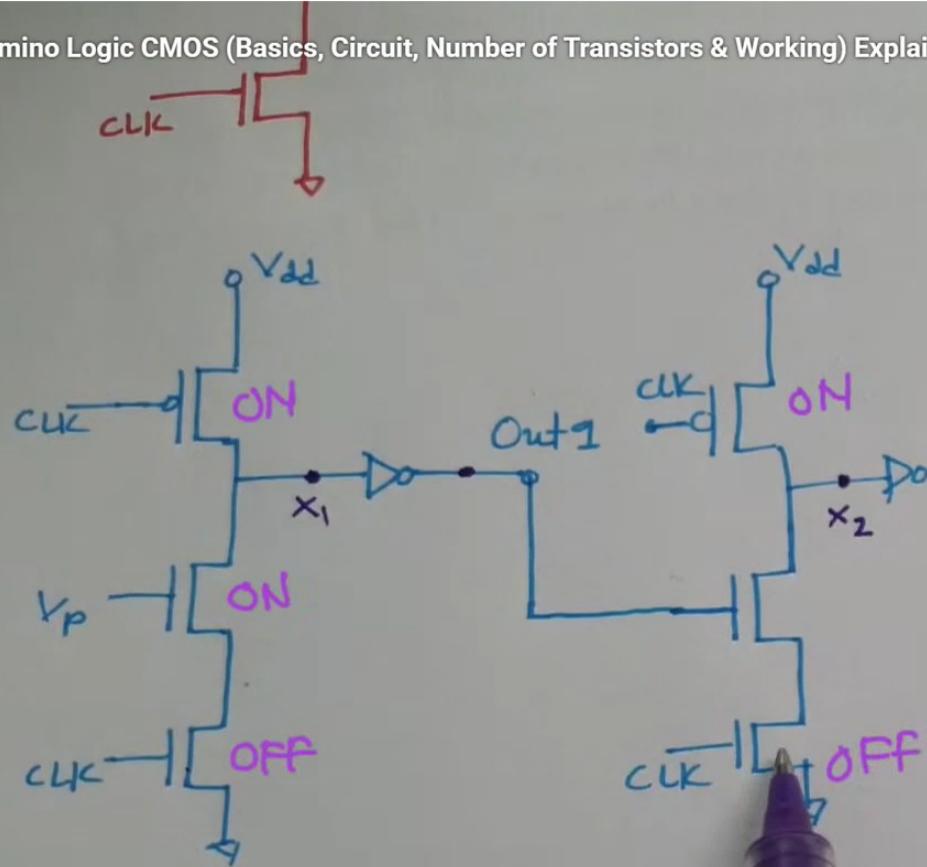


6:01 / 12:19

Working of Domino Logic CMOS &gt;



## Domino Logic CMOS (Basics, Circuit, Number of Transistors & Working) Explained



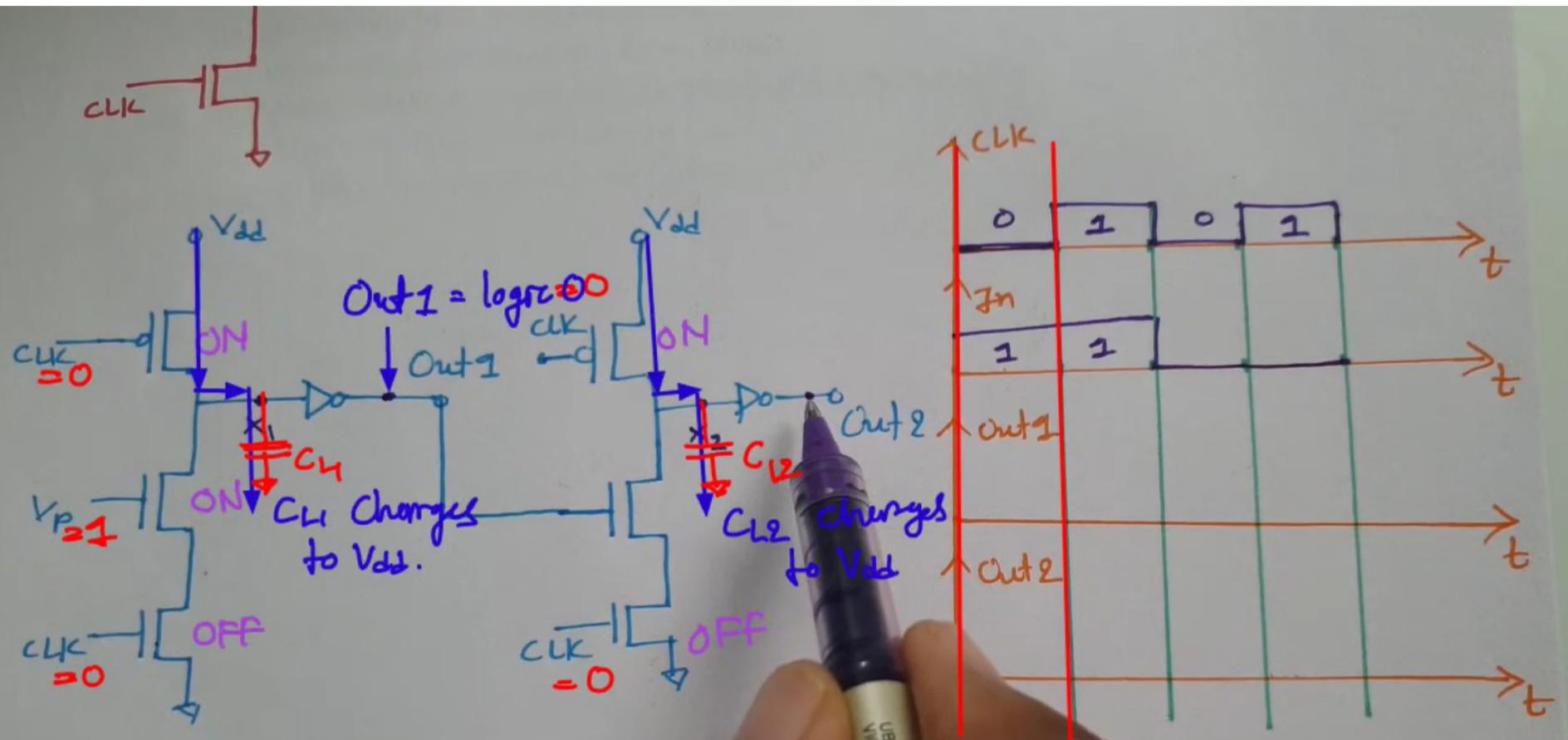
◀ ▶

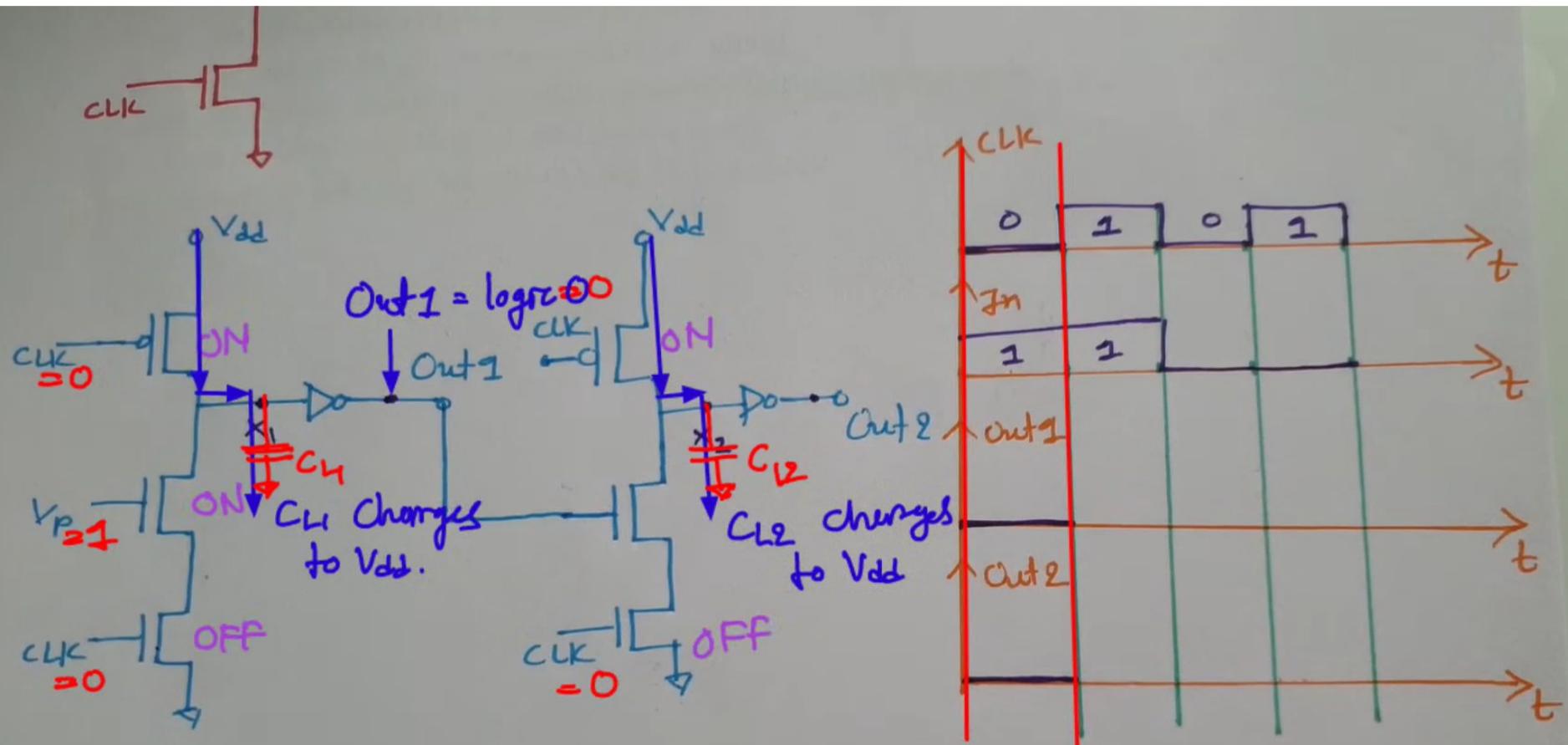


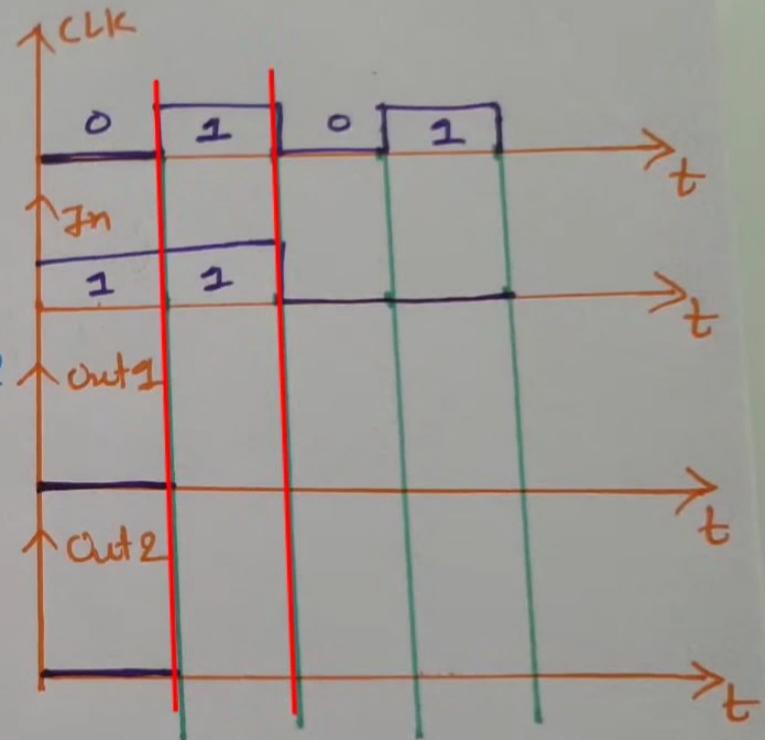
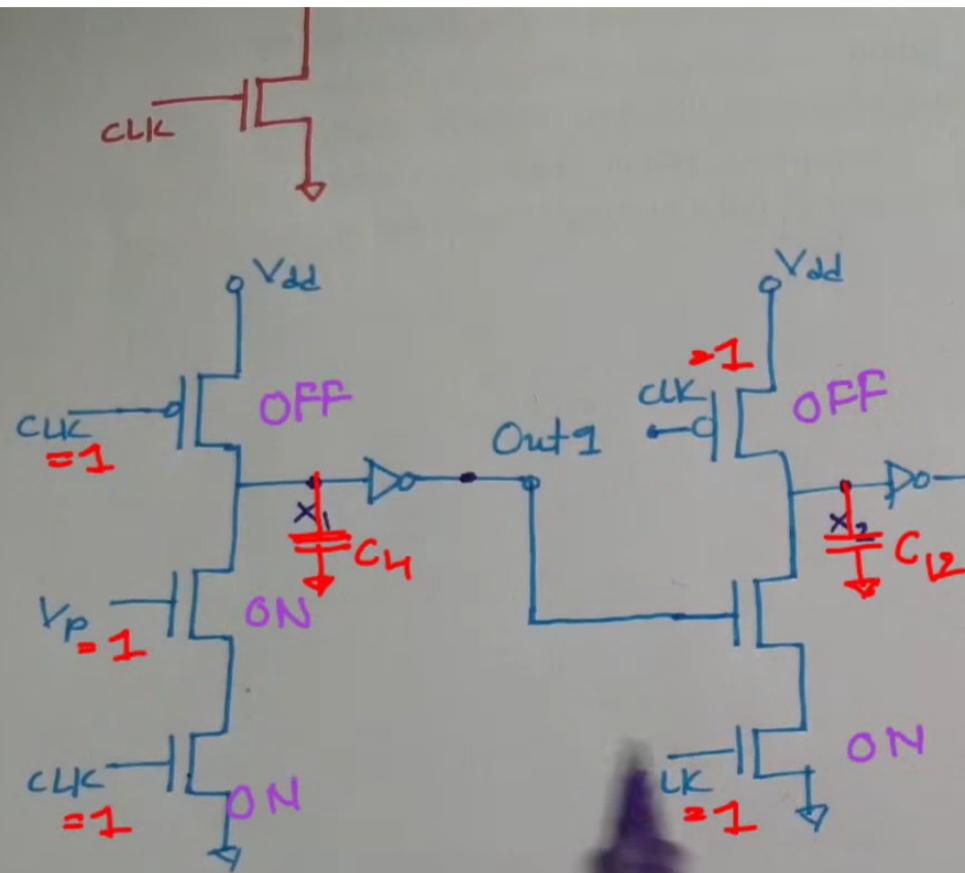
6:04 / 12:19

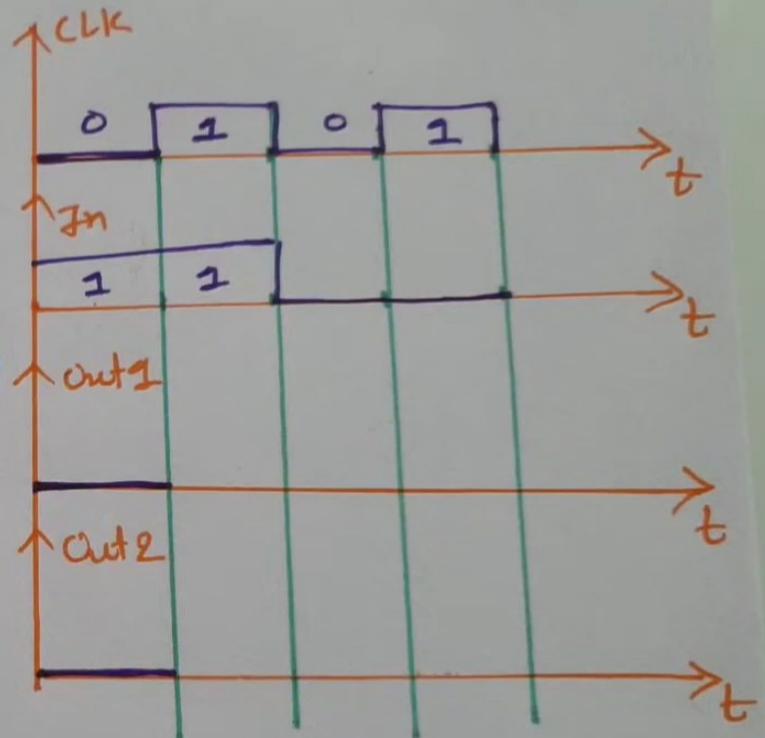
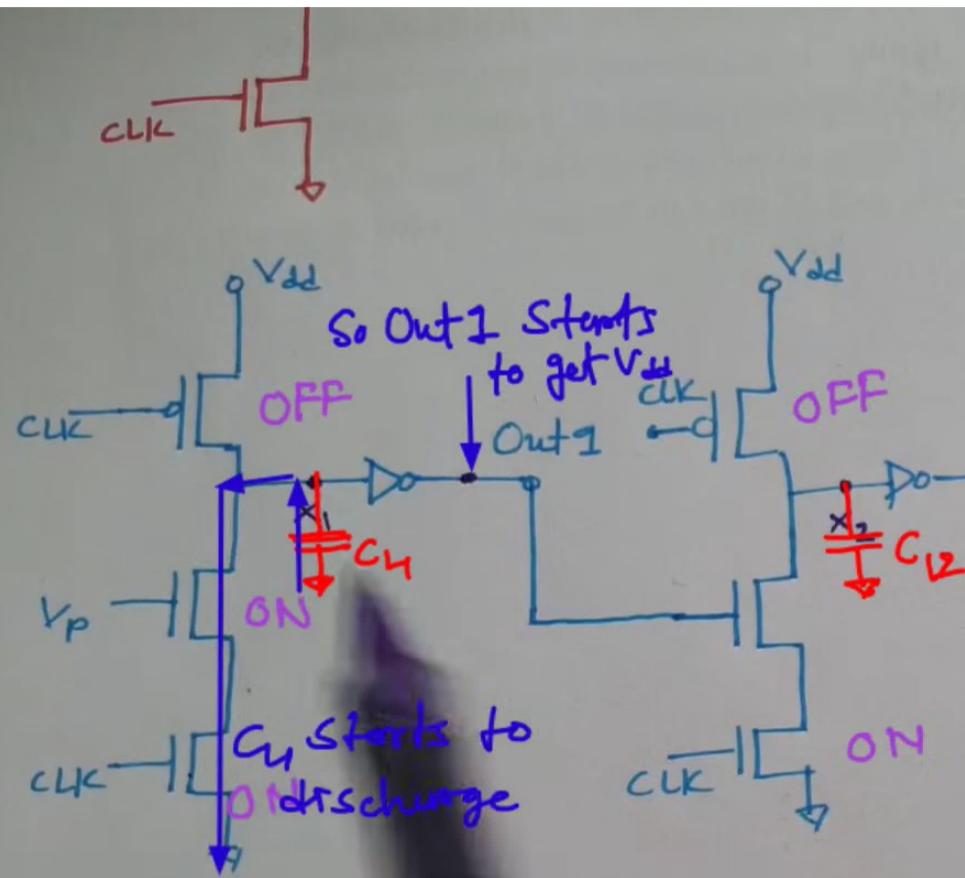
Working of Domino Logic CMOS >

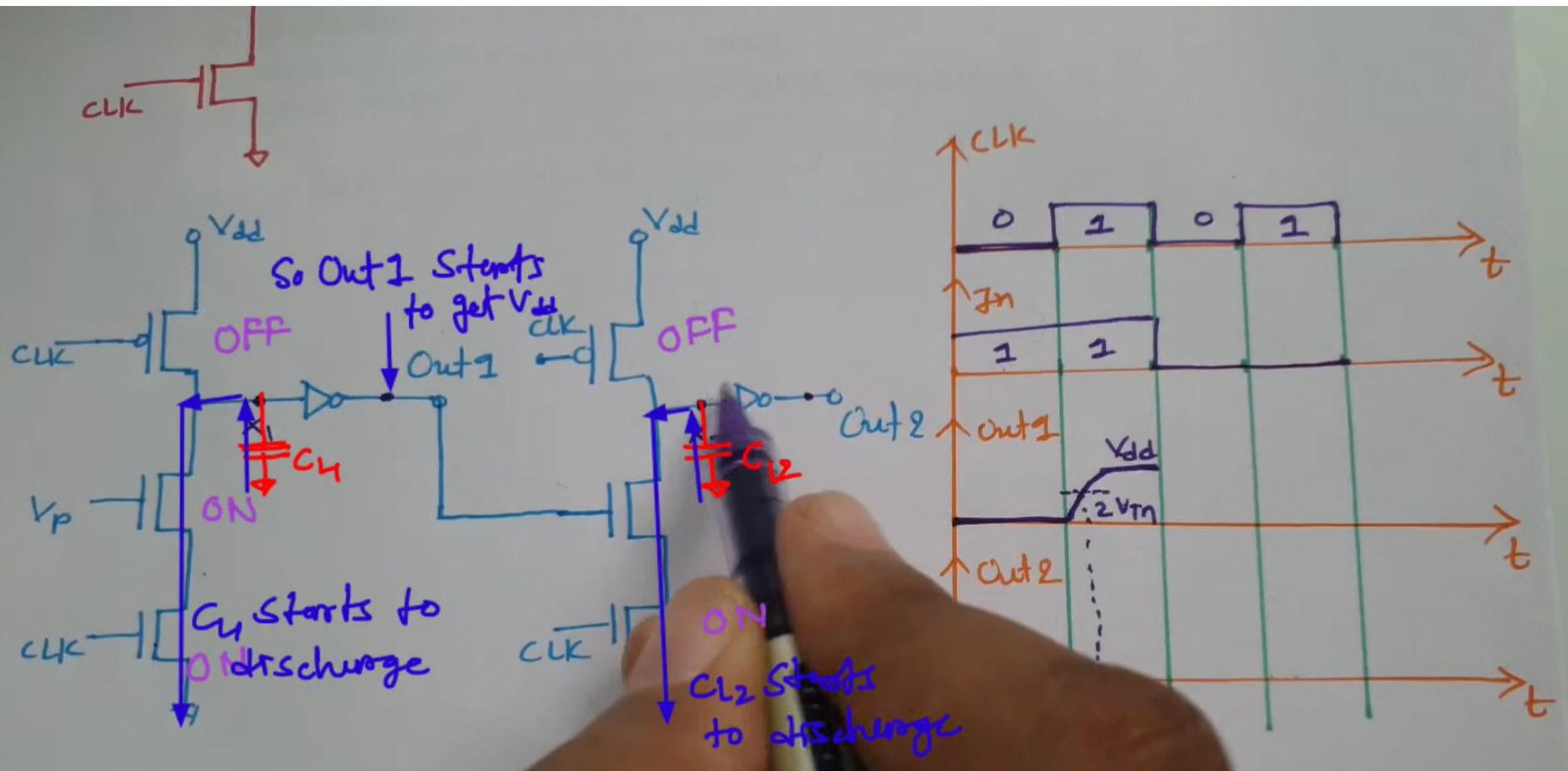


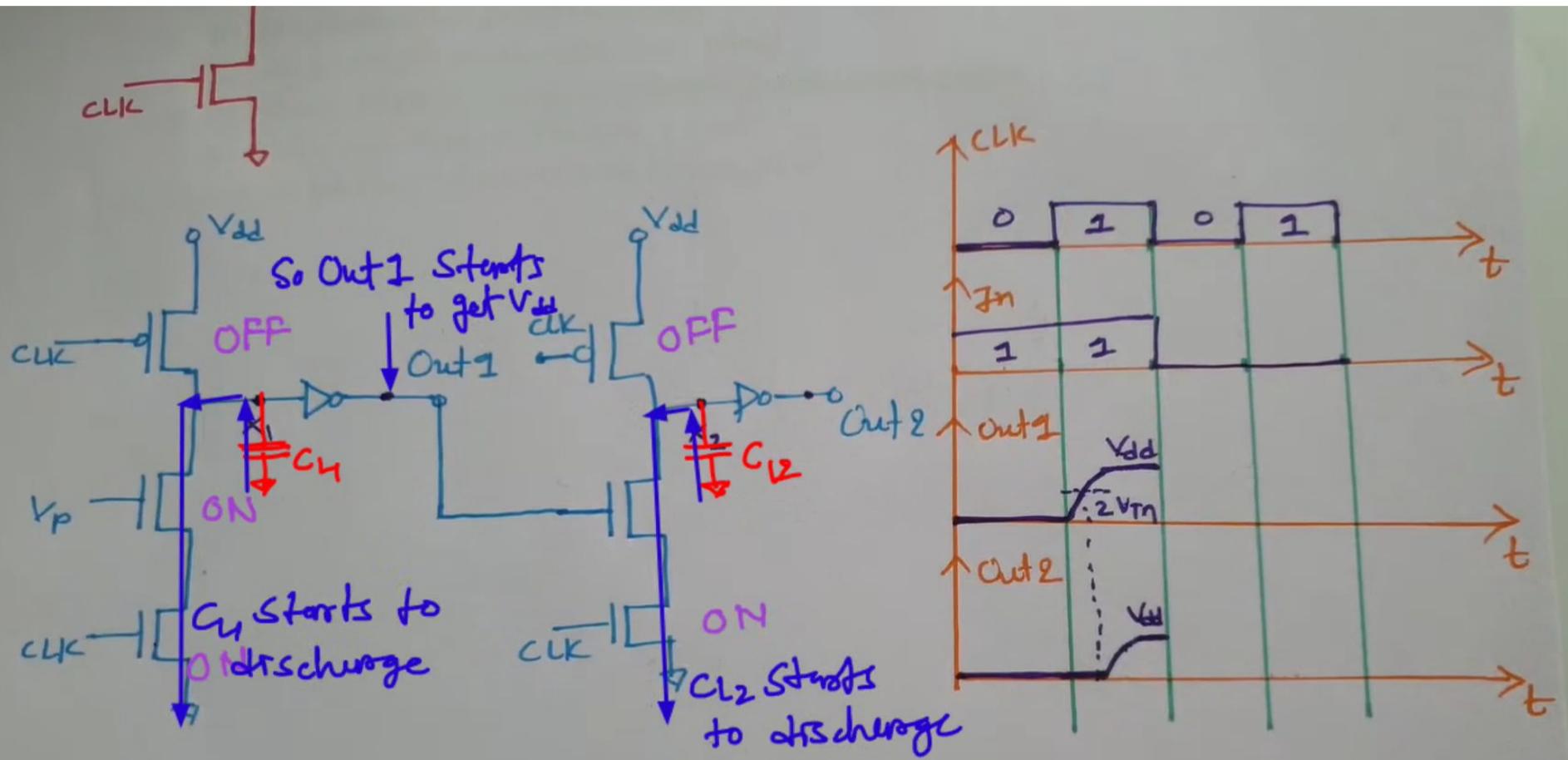


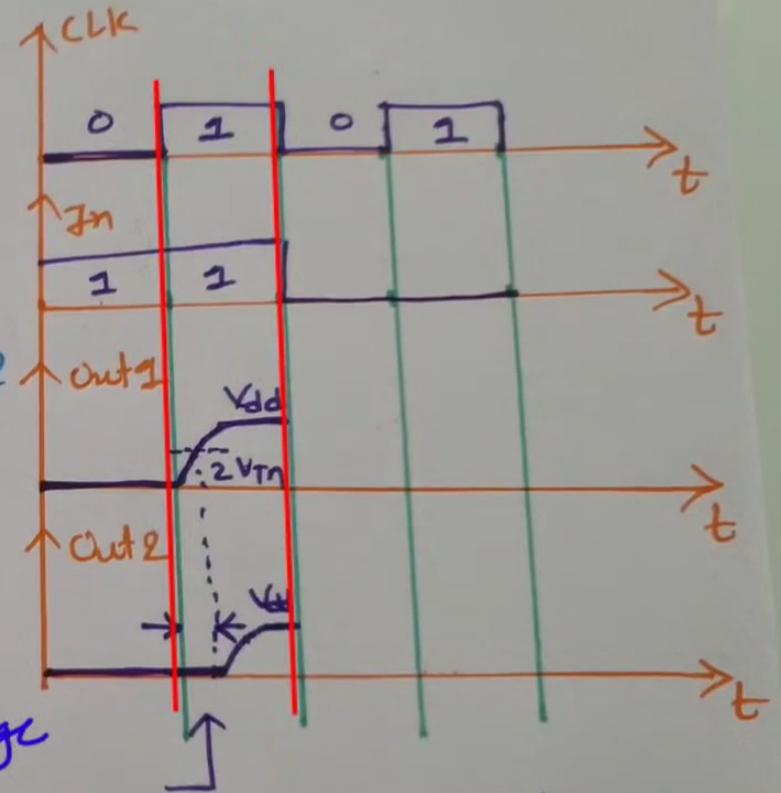
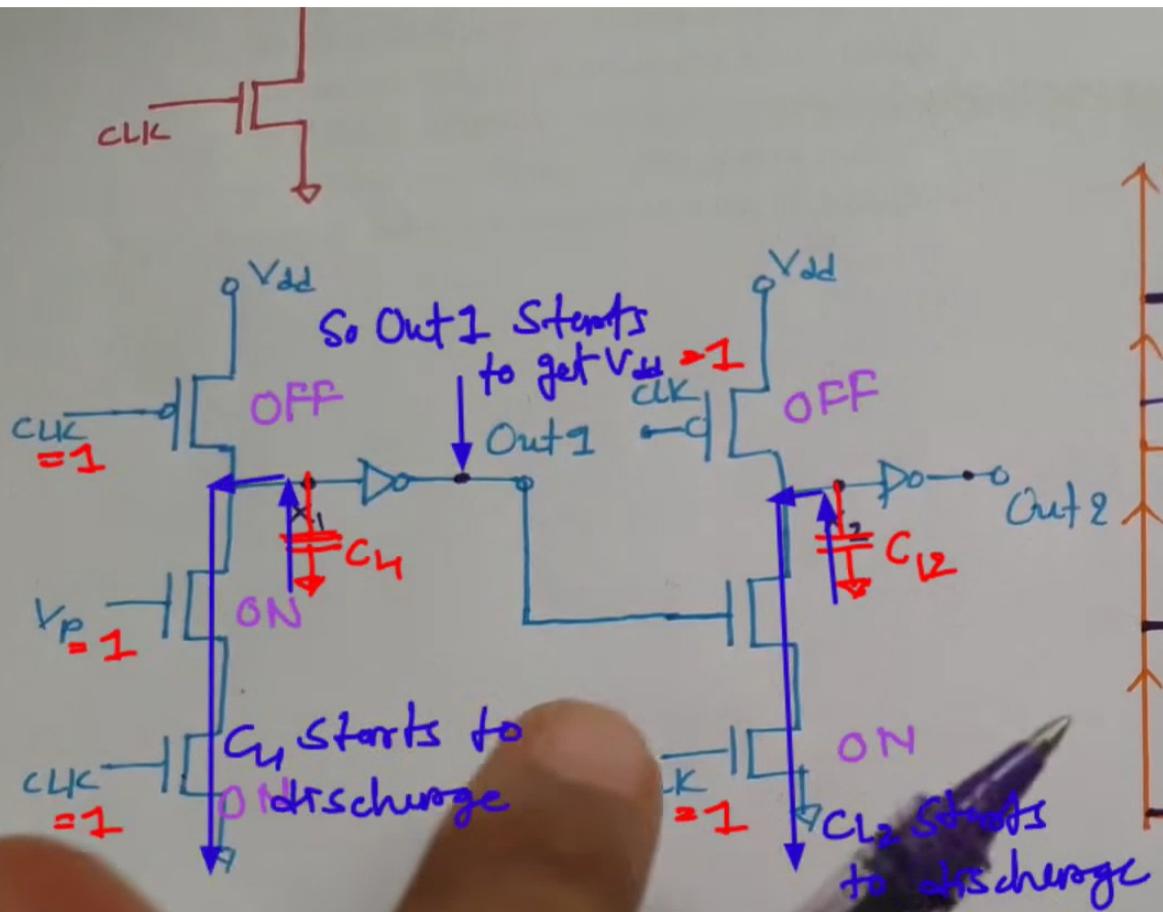




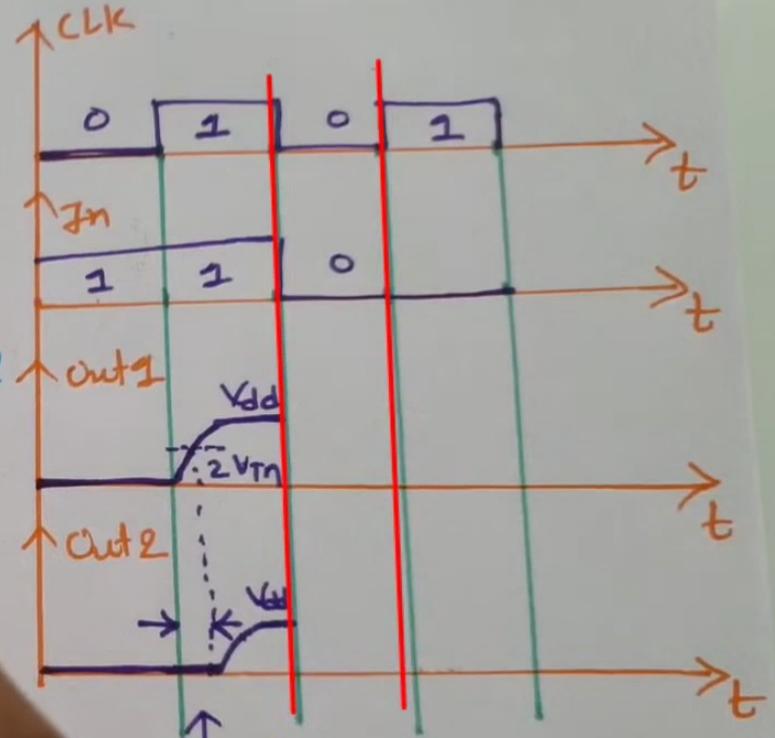
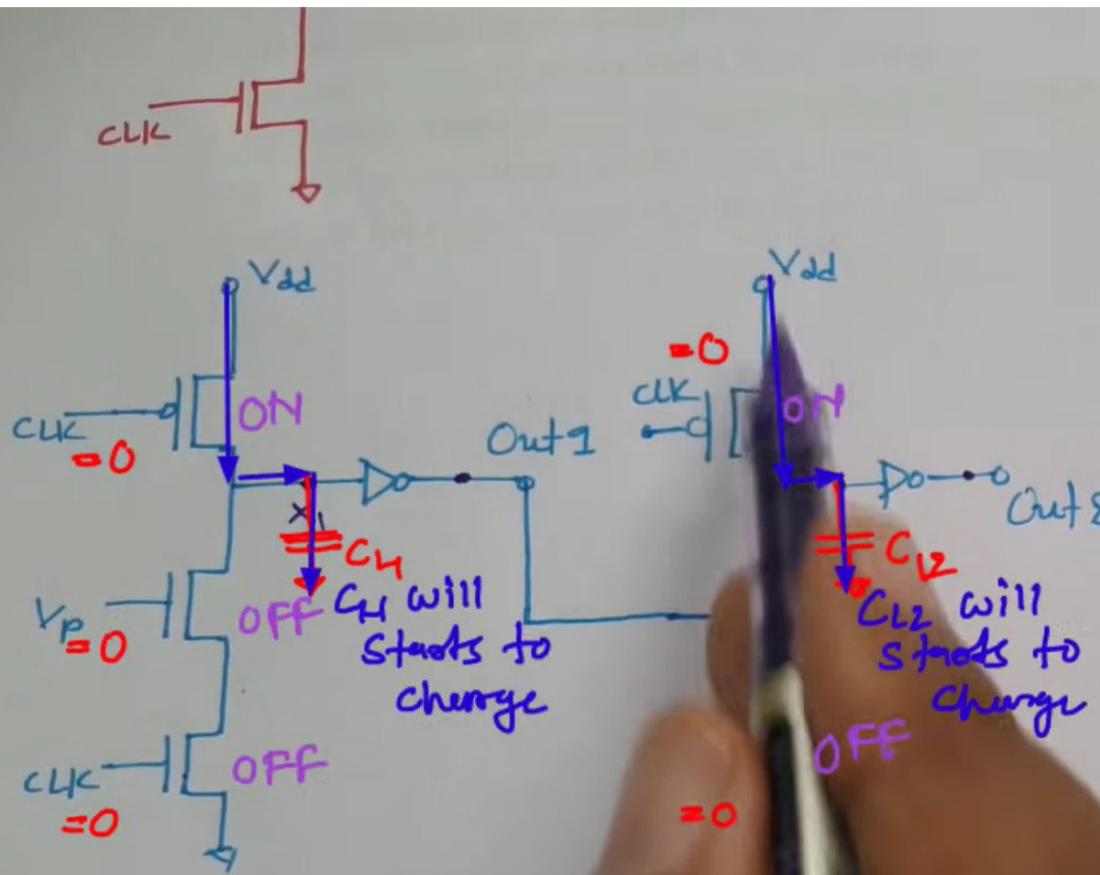






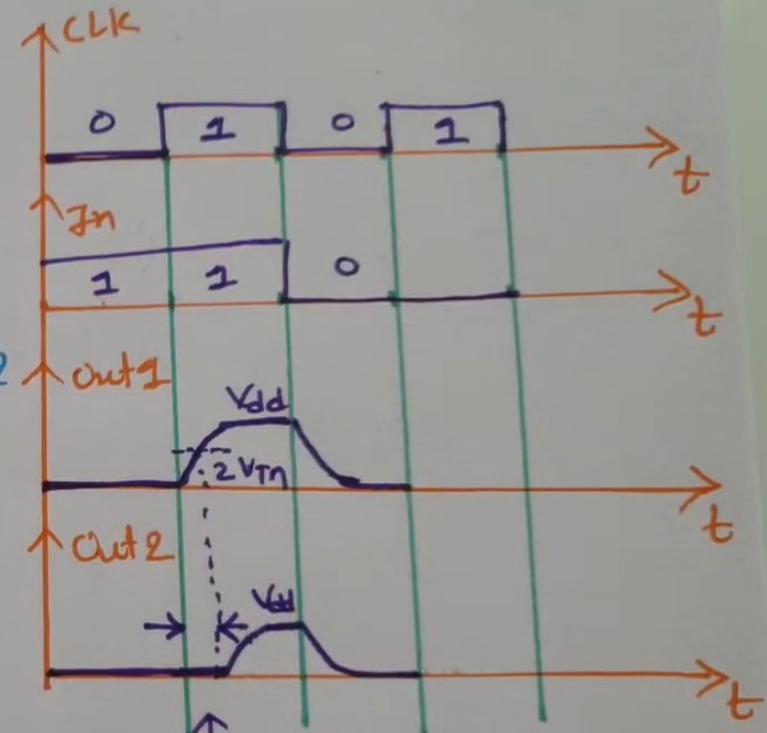
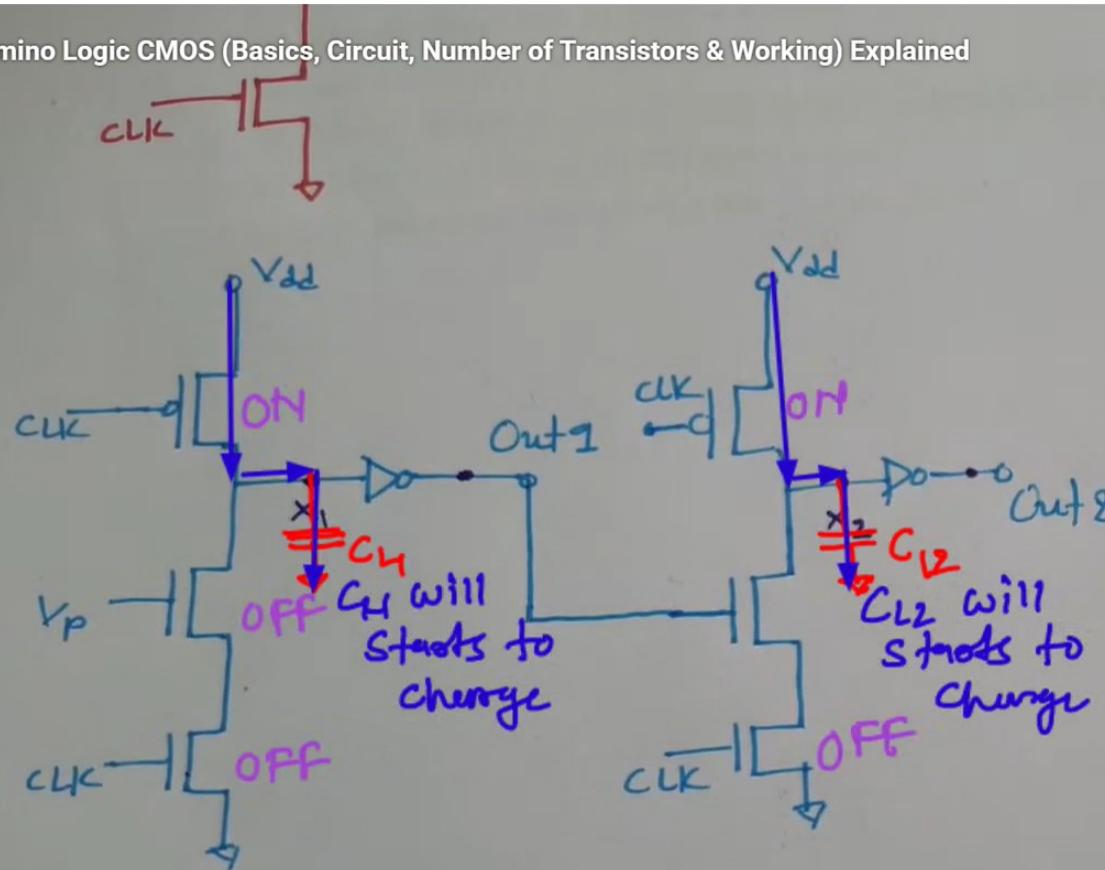


Domino logic is solving issue of Cascading but it falls in a problem RACE.

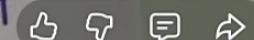


mino logic is solving  
the problem of Cascading but  
falls in a problem

## Domino Logic CMOS (Basics, Circuit, Number of Transistors & Working) Explained



Domino logic is solving issue of Cascading but it falls in a problem RACE.

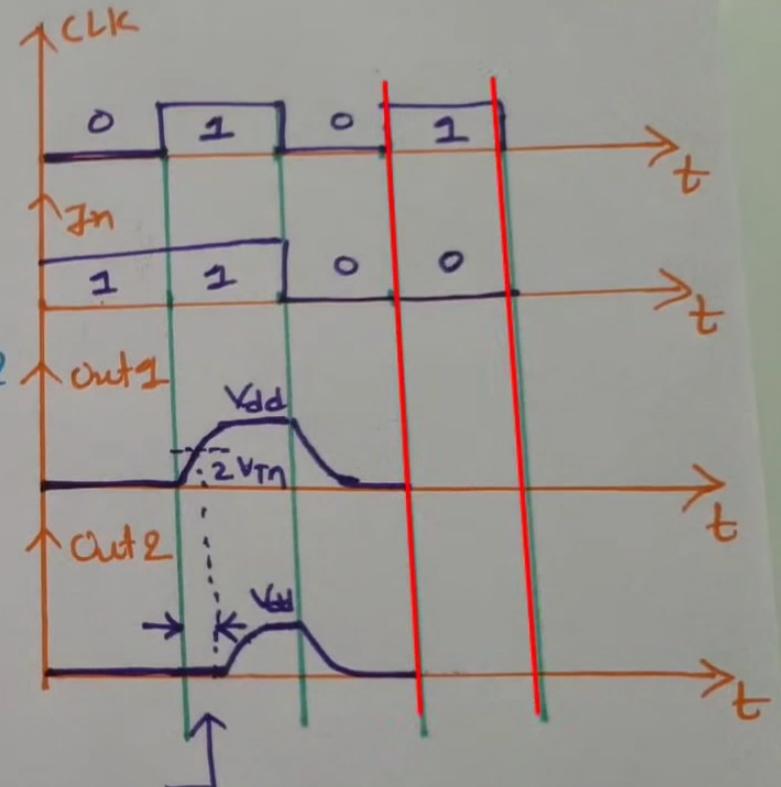
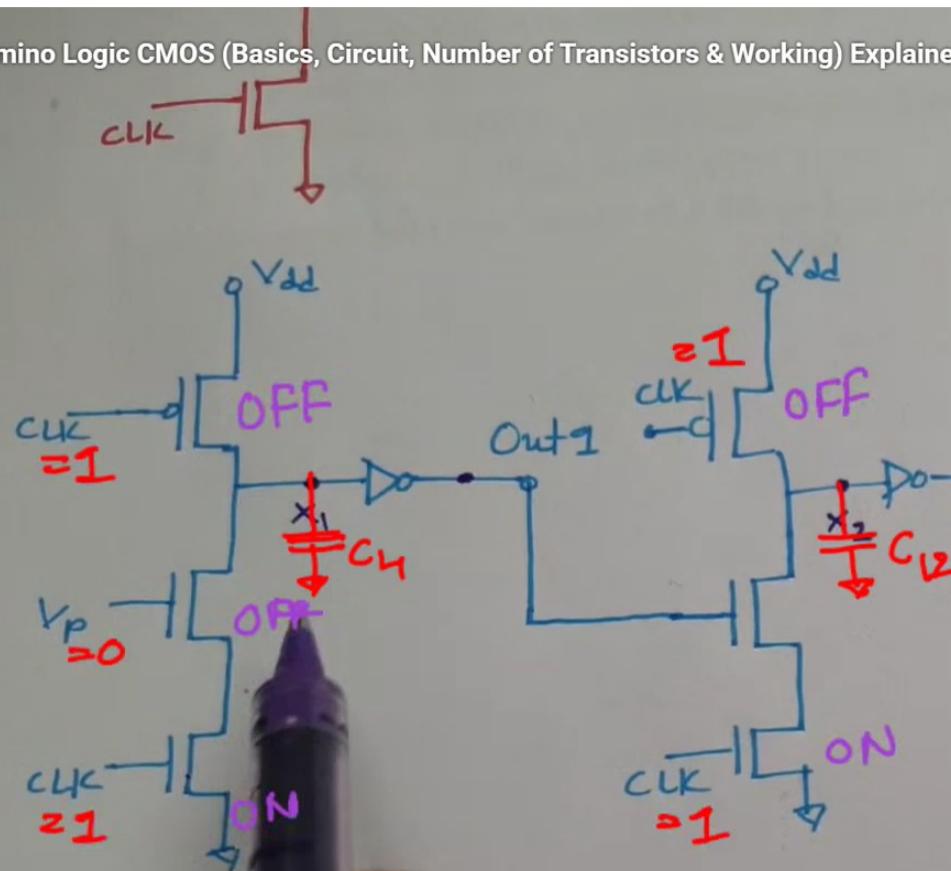


10:33 / 12:19

Working of Domino Logic CMOS >



## Domino Logic CMOS (Basics, Circuit, Number of Transistors & Working) Explained



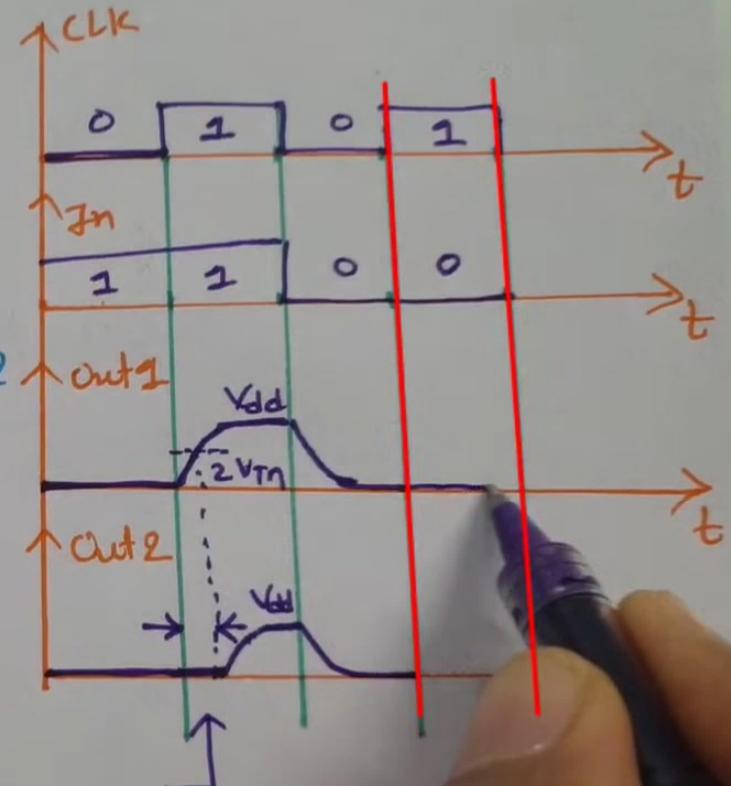
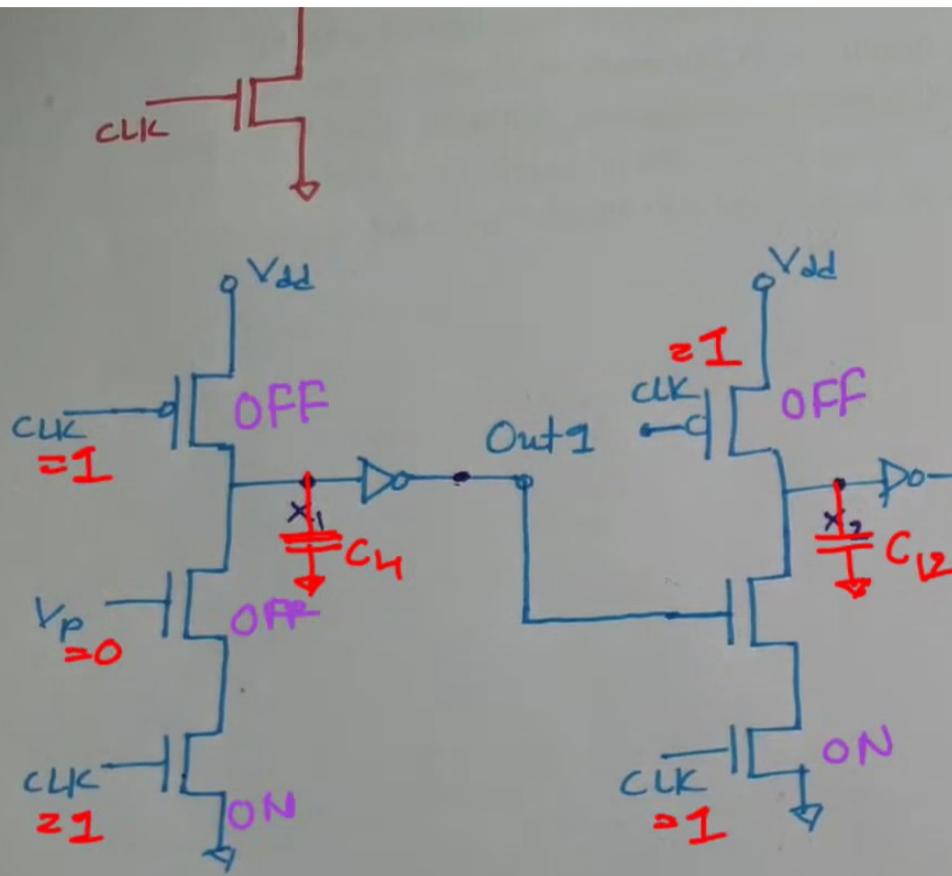
Domino logic is solving issue of Cascading but it falls in a problem RACE.



10:52 / 12:19

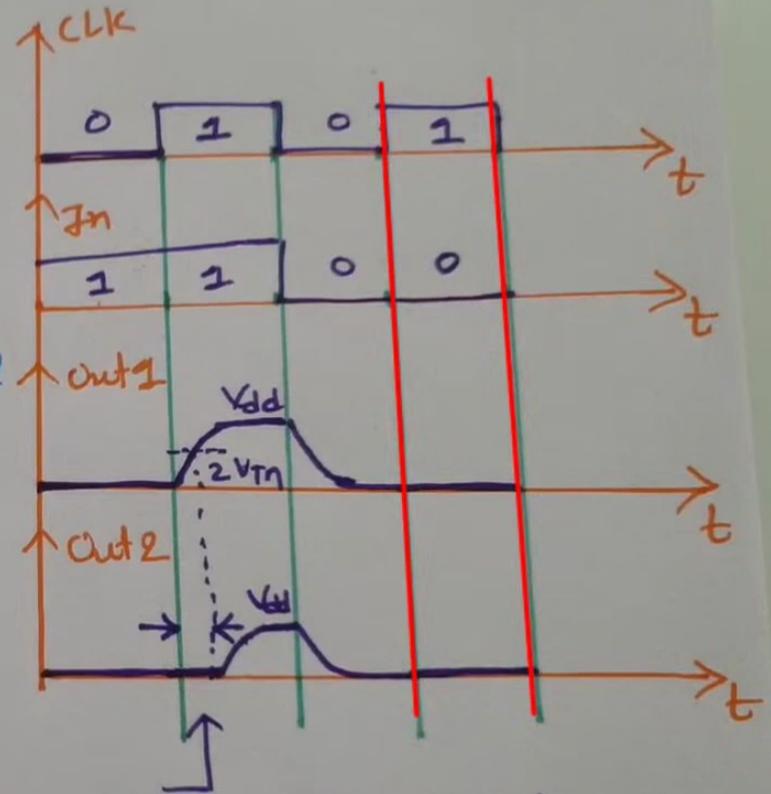
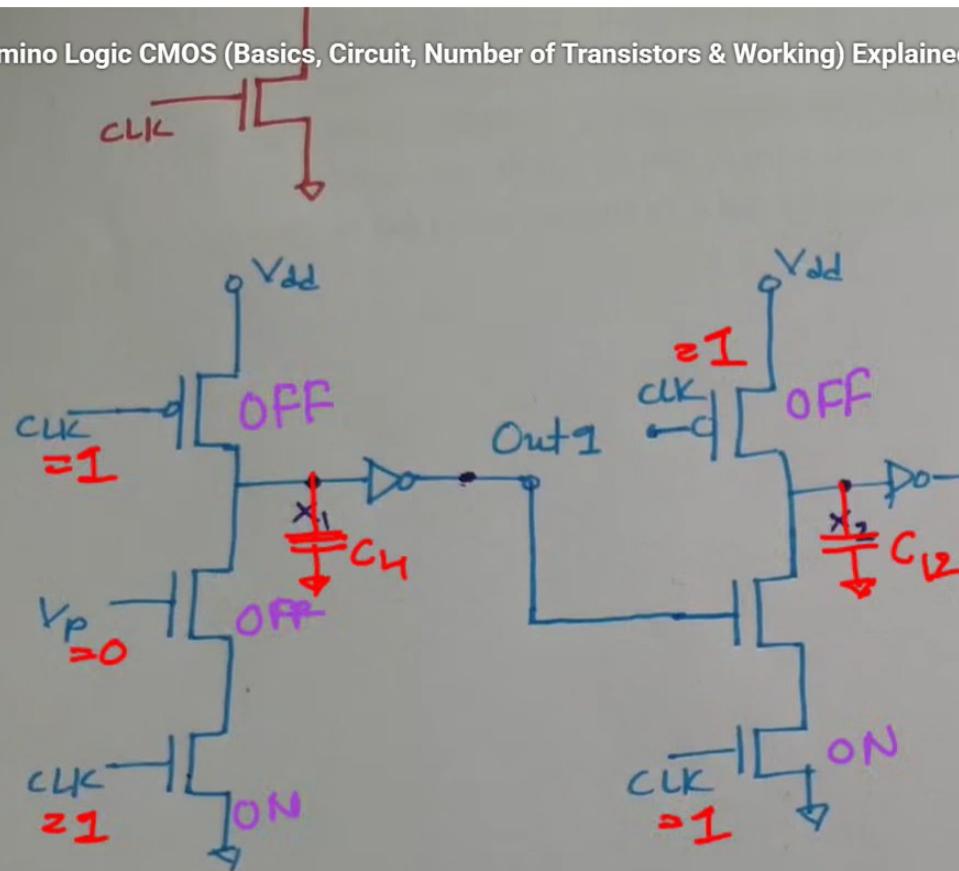
Working of Domino Logic CMOS >





Domino 1  
issue of  
1+ fan  
RACE

## Domino Logic CMOS (Basics, Circuit, Number of Transistors & Working) Explained



Domino logic is solving issue of Cascading but it falls in a problem RACE.



11:24 / 12:19

Working of Domino Logic CMOS >



Boolean Function using CMOS and Equivalent size of CMOS

Find a equivalent CMOS Inverter circuit for simultaneous switching of all inputs, assume that  $(\frac{W}{L})_P = 15$  for all PMOS transistors and  $(\frac{W}{L})_n = 10$  for all NMOS transistors for the following Boolean eq.

$$F = [(C + D + E) \cdot (A + B)]^n$$

→ (.) Operation

PMOS → Parallel

NMOS



1:29 / 13:10

CMOS Circuit Rules &gt;



Boolean Function Implementation Using CMOS and Equivalent Size of CMOS Circuit  
for the following Boolean eqn.

$$F = [(C + D + E) \cdot (A + B)]$$

→ (·) Operation

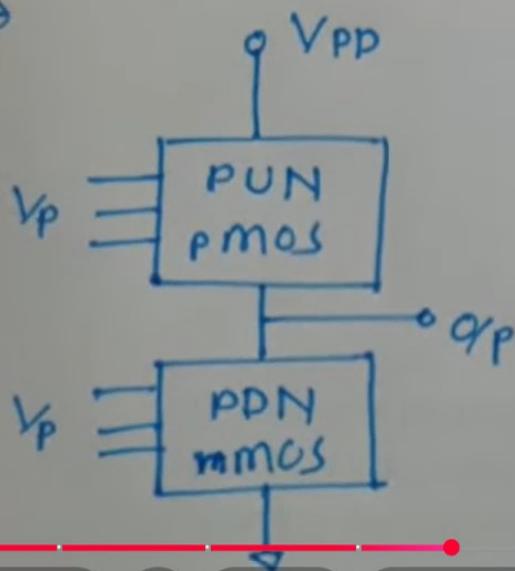
pMOS → Parallel

nMOS → Series

→ (+) Operation

pMOS → Series

nMOS → Parallel



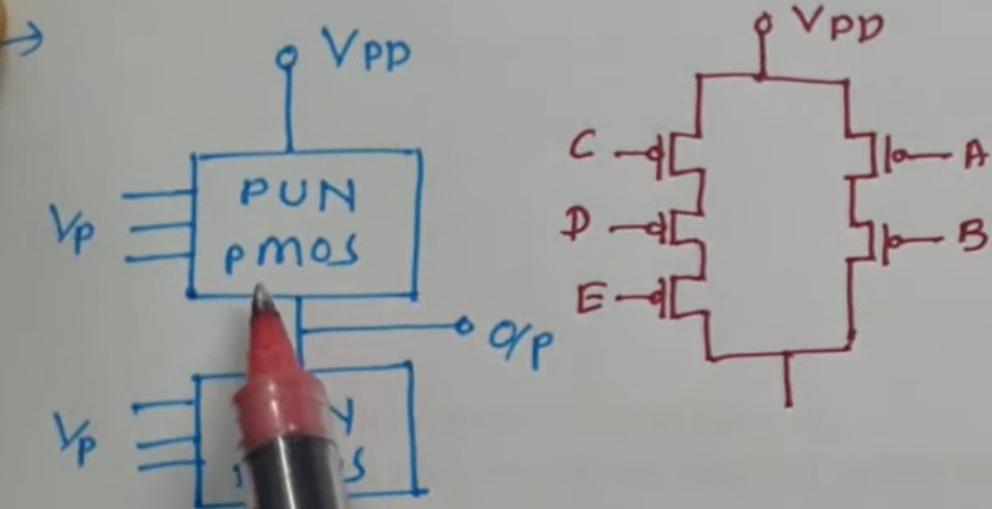
3:32 / 13:10

CMOS Boolean Function implementation >

$$F = [(C + D + E), (A + B)]'$$

→ (·) Operation  
PMOS → Parallel  
NMOS → Series

→ (+) Operation  
PMOS → Series  
NMOS → Parallel



Boolean Function Implementation Using CMOS and Equivalent Size of CMOS Circuit

$$F = (\bar{C} \cdot \bar{D} \cdot \bar{E}) + (\bar{A} + B)$$

→ (-) Operation

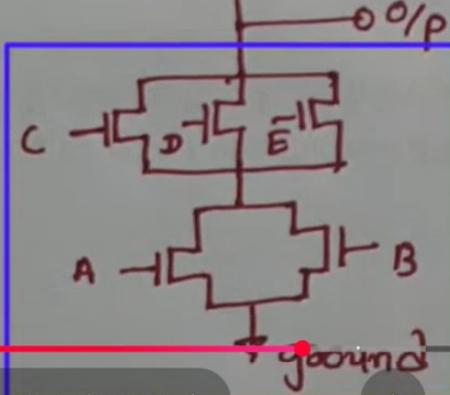
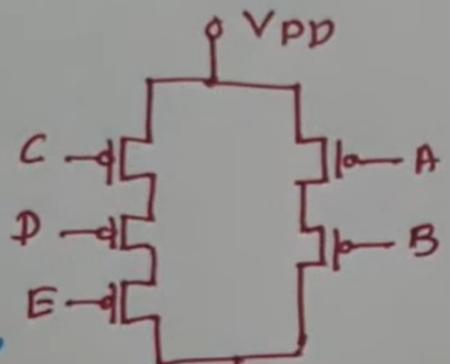
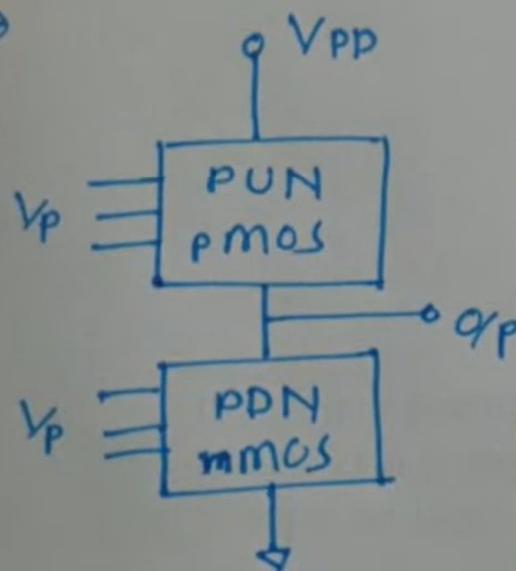
pMOS → Parallel

nMOS → Series

→ (+) Operation

pMOS → Series

nMOS → Parallel



Pull Down Network

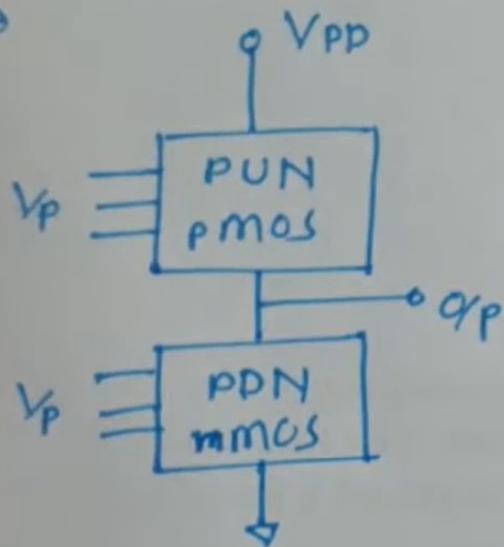


5:57 / 13:10

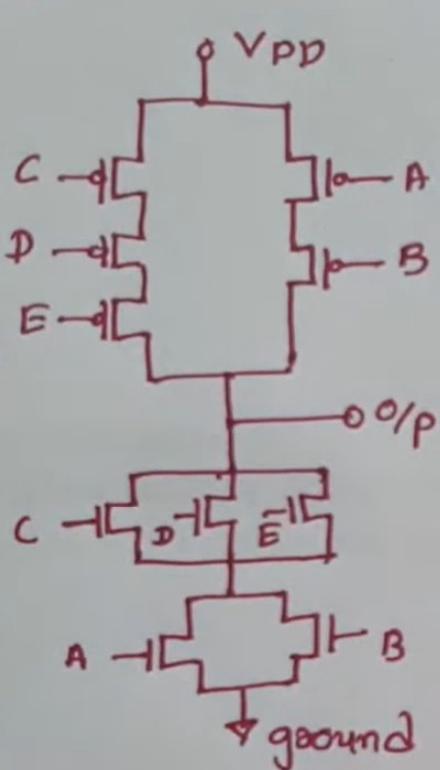
CMOS Boolean Function implementation



$nMOS \rightarrow Series$



$nMOS \rightarrow Parallel$



$$\left(\frac{W}{L}\right)_P = 15, \quad \left(\frac{W}{L}\right)_n = 10$$

$\rightarrow$  for Series (pmos or nmos)

$$\left(\frac{W}{L}\right)_{eq} = \frac{1}{\frac{1}{(W/L)_1} + \frac{1}{(W/L)_2} + \dots}$$

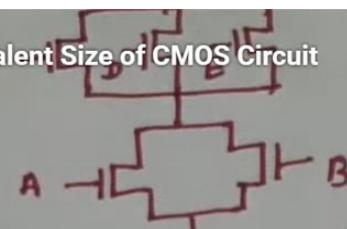
$\rightarrow$  for Parallel (pmos or nmos)

$$\left(\frac{W}{L}\right)_{eq} = (W/L)_1 + (W/L)_2 + \dots$$

$$\rightarrow \left(\frac{W}{L}\right)_{((DE))_P} = \frac{1}{\frac{1}{15} + \frac{1}{15} + \frac{1}{15}} = 5$$

## Boolean Function Implementation Using CMOS and Equivalent Size of CMOS Circuit

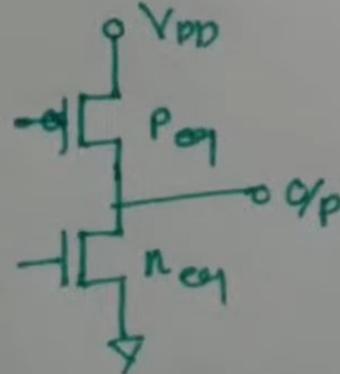
$\rightarrow \frac{W}{L} \text{ mmos}$



$$\rightarrow \left(\frac{W}{L}\right)_{(DE)_n} = 10 + 10 + 10 = 30 \rightarrow \text{ground}$$

$$\rightarrow \left(\frac{W}{L}\right)_{(AB)_n} = 10 + 10 = 20$$

$$\rightarrow \left(\frac{W}{L}\right)_n = \frac{1}{\frac{1}{30} + \frac{1}{20}} = 12$$



$$\rightarrow \left(\frac{W}{L}\right)_q = \frac{1}{\frac{1}{15} + \frac{1}{15} + \frac{1}{15}} = 5$$

$$\rightarrow \left(\frac{W}{L}\right)_{(AB)_p} = \frac{1}{\frac{1}{15} + \frac{1}{15}} = 7.5$$

$$\rightarrow \left(\frac{W}{L}\right)_p = 5 + 7.5 = 12.5$$

$$\left(\frac{W}{L}\right)_q = \left(\frac{W}{L}\right)_p + \left(\frac{W}{L}\right)_n$$

$$= \frac{1}{\frac{1}{12.5} + \frac{1}{12}}$$

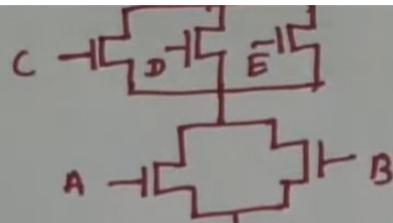
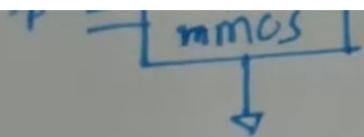
=



12:26 / 13:10

Equivalent size of CMOS Circuit >

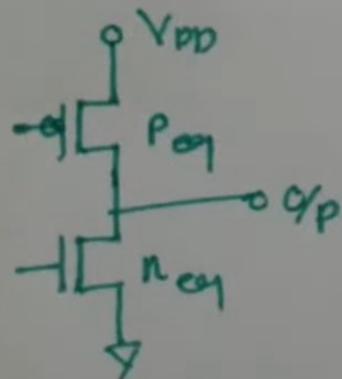




$$\rightarrow \left(\frac{\omega}{L}\right)_{(CDE)_n} = 10 + 10 + 10 = 30 \rightarrow \text{ground}$$

$$\rightarrow \left(\frac{\omega}{L}\right)_{(AB)_n} = 10 + 10 = 20$$

$$\rightarrow \left(\frac{\omega}{L}\right)_n = \frac{1}{\frac{1}{30} + \frac{1}{20}} = 12$$



$$\left(\frac{\omega}{L}\right)_q = \left(\frac{\omega}{L}\right)_p + \left(\frac{\omega}{L}\right)_n + \dots$$

$$\rightarrow \left(\frac{\omega}{L}\right)_{(CDE)_p} = \frac{1}{\frac{1}{15} + \frac{1}{15} + \frac{1}{15}} = 5$$

$$\rightarrow \left(\frac{\omega}{L}\right)_{(AB)_p} = \frac{1}{\frac{1}{15} + \frac{1}{15}} = 7.5$$

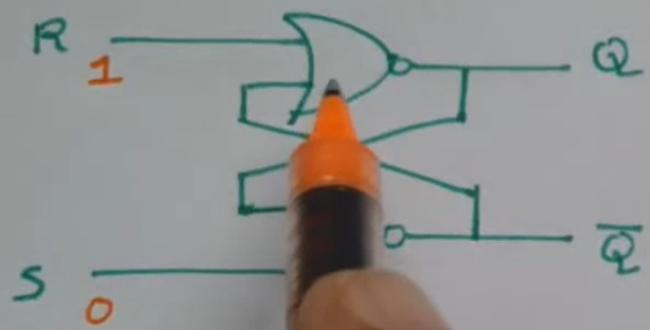
$$\rightarrow \left(\frac{\omega}{L}\right)_p = 5 + 7.5 = 12.5$$

$$\left(\frac{\omega}{L}\right)_q = \frac{1}{\left(\frac{\omega}{L}\right)_p + \left(\frac{\omega}{L}\right)_n}$$

$$= \frac{1}{\frac{1}{12.5} + \frac{1}{12}}$$

$$= \boxed{6.12}$$

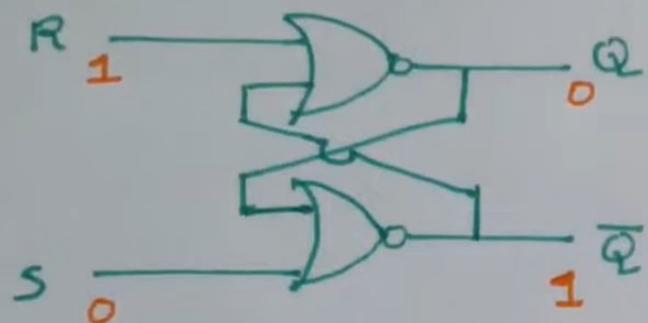
CMOS SR Latch using NOR gates.



S	R	Q	$\bar{Q}$	Operation
0	0	-	-	
0	1	1	0	
1	0	0	1	
1	1	-	-	

For NOR gate, if any input is logic '1',  
then output is logic '0'.

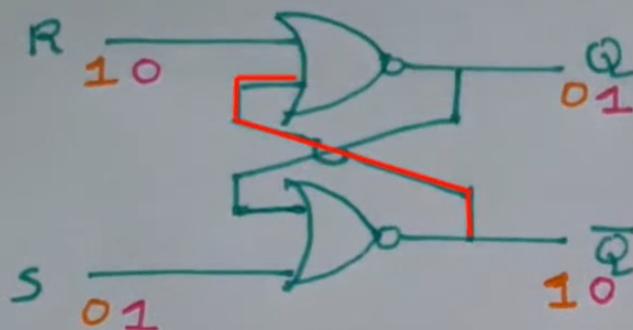
CMOS SR Latch using NOR gates.



S	R	Q	$\bar{Q}$	Operation
0	0	-	-	-
0	1	0	1	-
1	0	-	-	-
1	1	-	-	-

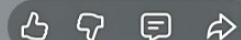
For NOR gate, if any input is logic '1',  
then output is logic '0'.

CMOS SR latch using NOR gates.



S	R	$Q$	$\bar{Q}$	Operation
0	0			
0	1	0	1	Reset
1	0			
1	1			

For NOR gate, if any input is logic '1', then output is logic '0'.

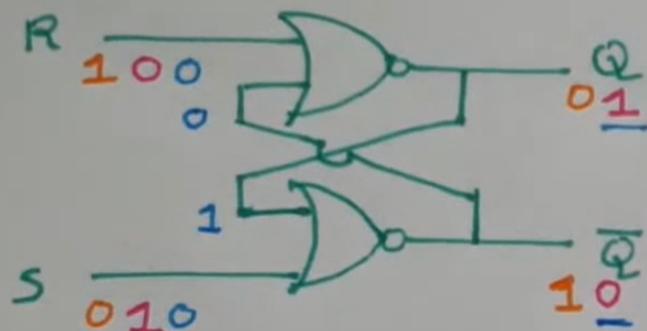


4:01 / 11:13

SR Latch using NOR Gates (Basics, Working &amp; Truth Table)



# CMOS SR Latch using NOR gates.



S	R	$Q$	$\bar{Q}$	Operation
0	0	<u>0</u>	<u>1</u>	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1			

For NOR gate, if any input is logic '1', then output is logic '0'.

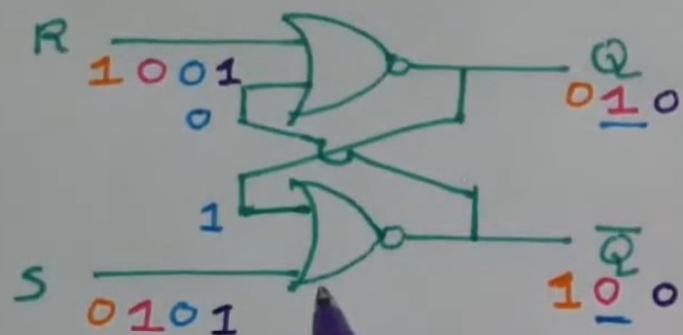


5:26 / 11:13

SR Latch using NOR Gates (Basics, Working &amp; Truth Table)



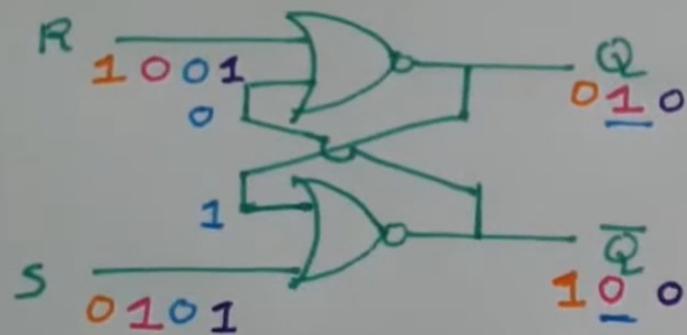
CMOS SR Latch using NOR gates.



S	R	Q	$\bar{Q}$	Operation
0	0	<u>0</u>	<u>1</u>	<u>Hold</u>
0	1	0	1	<u>Reset</u>
1	0	1	0	<u>Set</u>
1	1			

For NOR gate, if any input is logic '1',  
then output is logic '0'.

CMOS SR Latch using NOR gates.



S	R	Q	$\bar{Q}$	Operation
0	0	1	0	<u>Hold</u>
0	1	0	1	<u>Reset</u>
1	0	1	0	<u>Set</u>
1	1	0	0	Not allowed

For NOR gate, if any input is logic '1', then output is logic '0'.

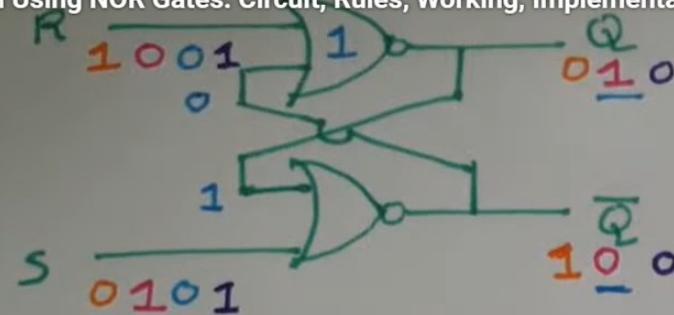


6:06 / 11:13

SR Latch using NOR Gates (Basics, Working &amp; Truth Table)

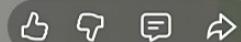
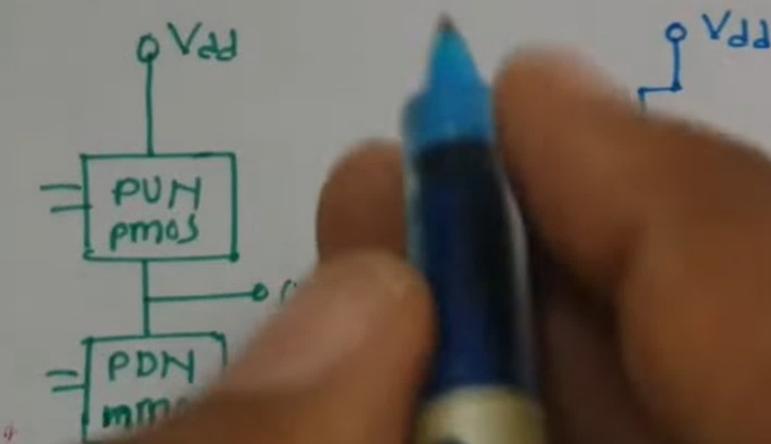


## CMOS SR Latch Using NOR Gates: Circuit, Rules, Working, Implementation & Truth Table



S	R	Q	$\bar{Q}$	Operation
0	0	<u>1</u>	0	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Not allowed

→ (+) Operation → PMOS - Series → NMOS - Parallel

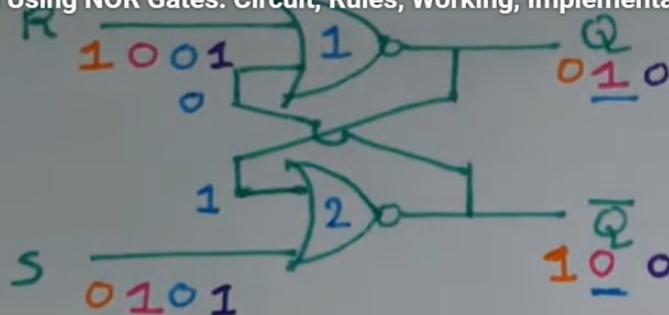


9:15 / 11:13

CMOS SR Latch using NOR Gates implementation >

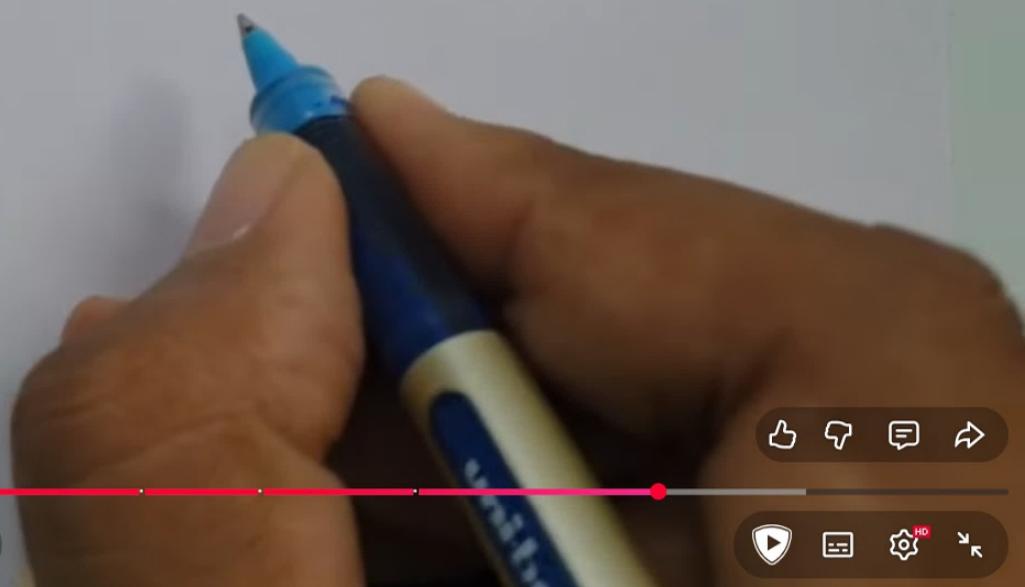
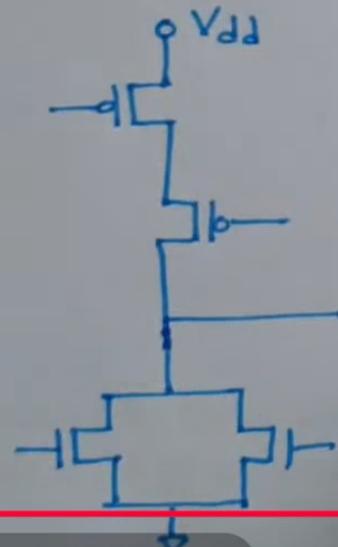
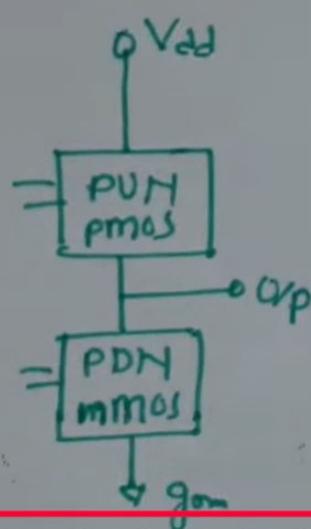


## CMOS SR Latch Using NOR Gates: Circuit, Rules, Working, Implementation & Truth Table

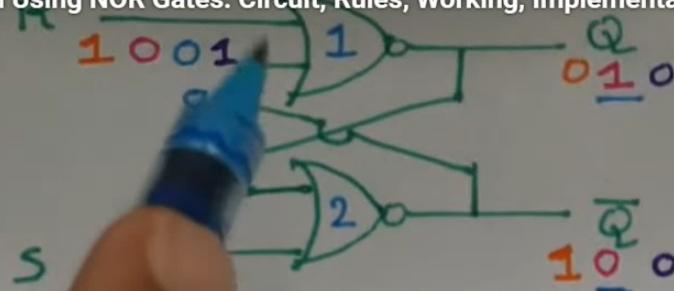


S	R	$Q$	$\bar{Q}$	Operation
0	0	<u>0</u>	1	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Not allowed

→ (+) Operation → PMOS - Series → NMOS - Parallel

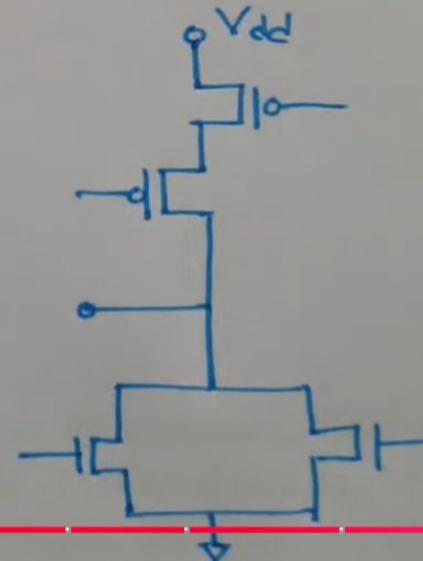
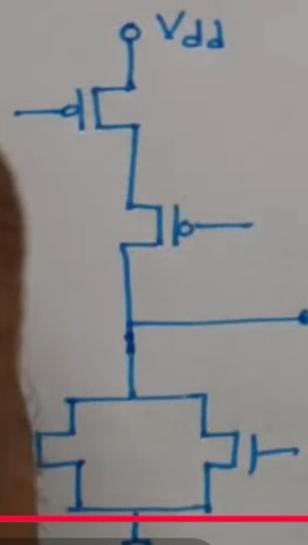


## CMOS SR Latch Using NOR Gates: Circuit, Rules, Working, Implementation & Truth Table



S	R	$Q$	$\bar{Q}$	Operation
0	0	<u>0</u>	1	<u>Hold</u>
0	1	0	1	<u>Reset</u>
1	0	1	0	<u>Set</u>
1	1	0	0	Not allowed

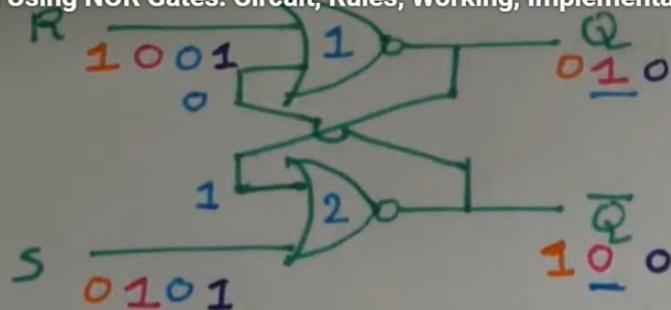
on  $\rightarrow$  PMOS - Series  $\rightarrow$  NMOS - Parallel



9:33 / 11:13

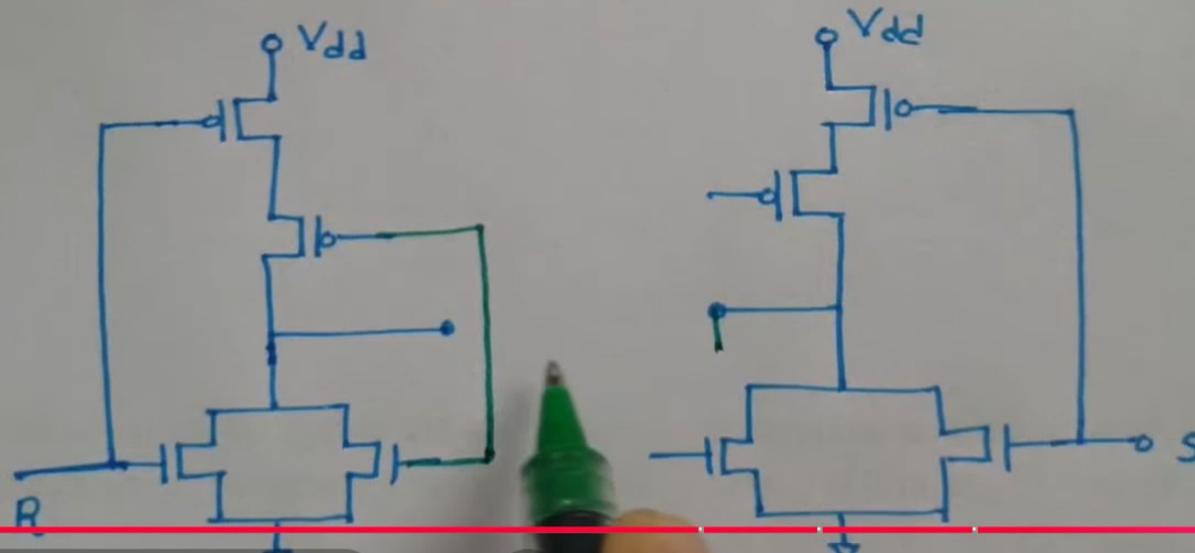
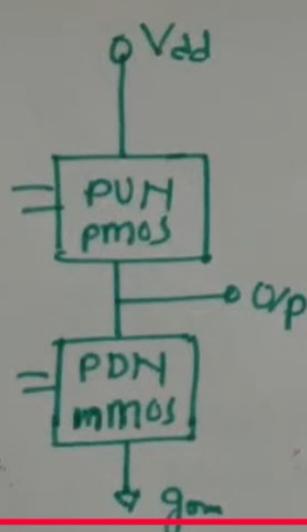
CMOS SR Latch using NOR Gates implementation >

## CMOS SR Latch Using NOR Gates: Circuit, Rules, Working, Implementation & Truth Table



S	R	Q	$\bar{Q}$	Operation
0	0	<u>Q</u>	<u><math>\bar{Q}</math></u>	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Not allowed

→ (+) Operation → PMOS - Series → NMOS - Parallel

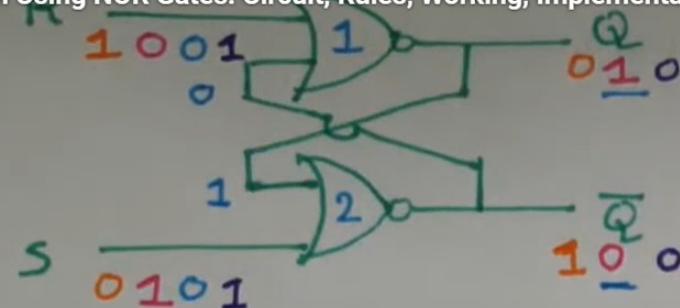


10:25 / 11:13

CMOS SR Latch using NOR Gates implementation >

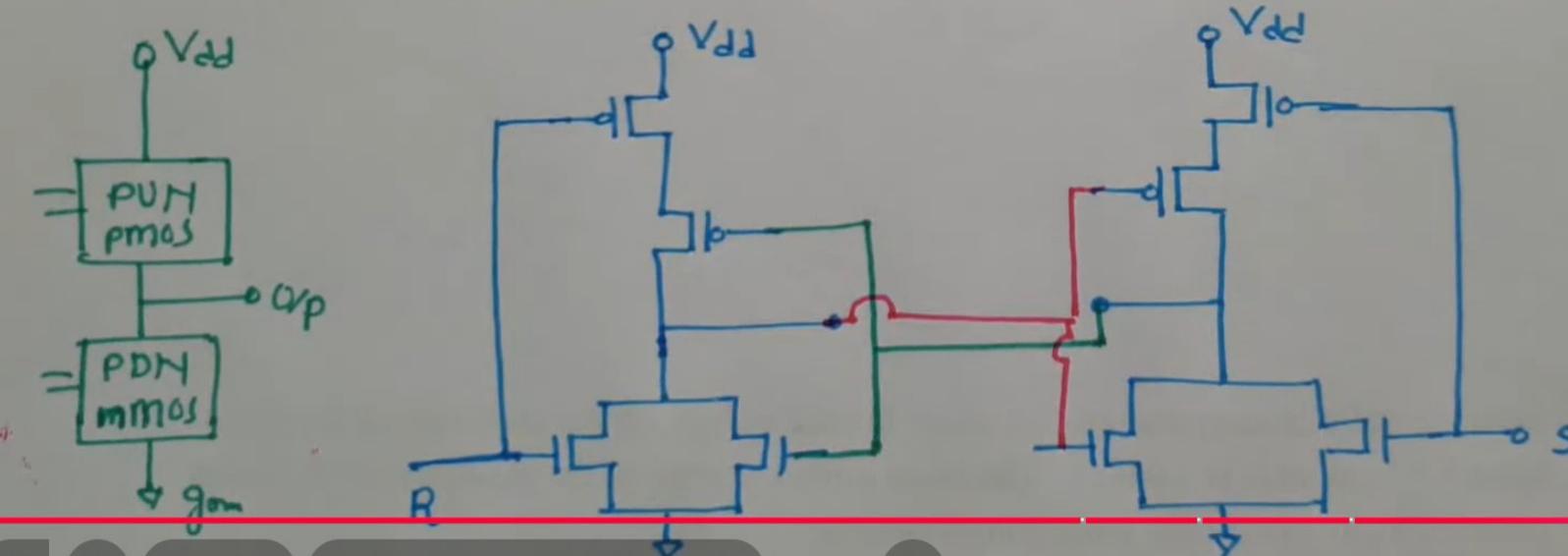


## CMOS SR Latch Using NOR Gates: Circuit, Rules, Working, Implementation & Truth Table



S	R	Q	$\bar{Q}$	Operation
0	0	0	1	<u>Hold</u>
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Not allowed

→ (+) Operation → PMOS - Series → NMOS - Parallel

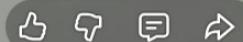
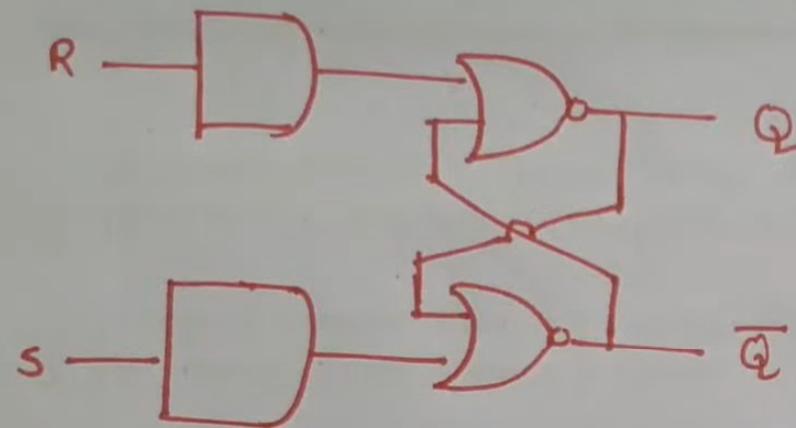


10:54 / 11:13

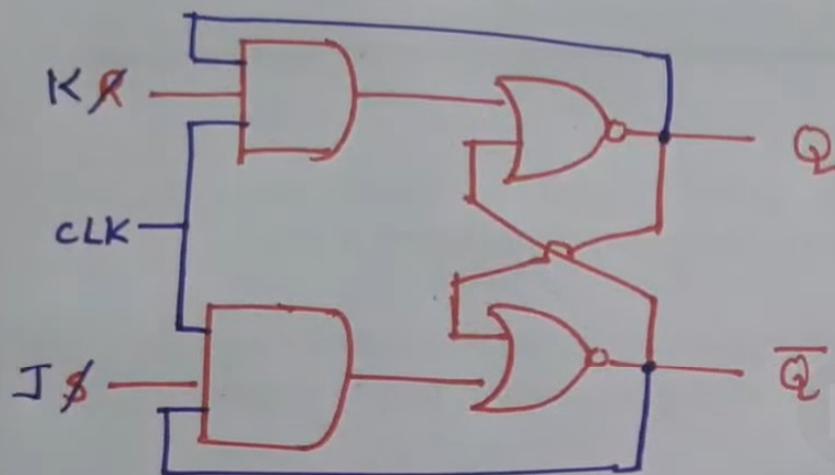
CMOS SR Latch using NOR Gates implementation >



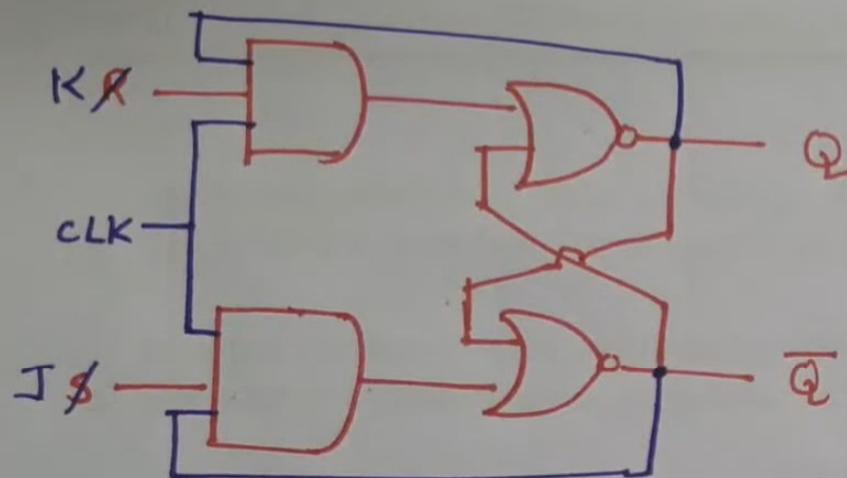
# CMOS JK Flip Flop using NOR gates



## CMOS JK Flip Flop using NOR gates

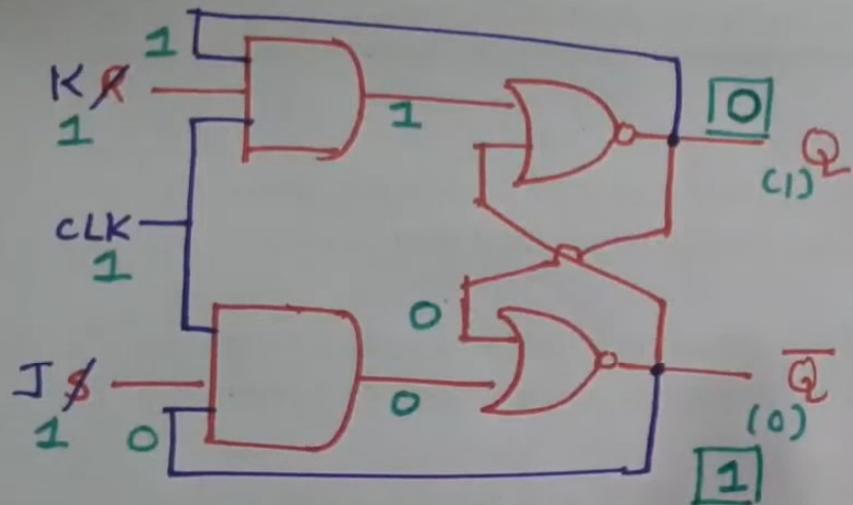


CMOS JK Flip Flop using NOR gates



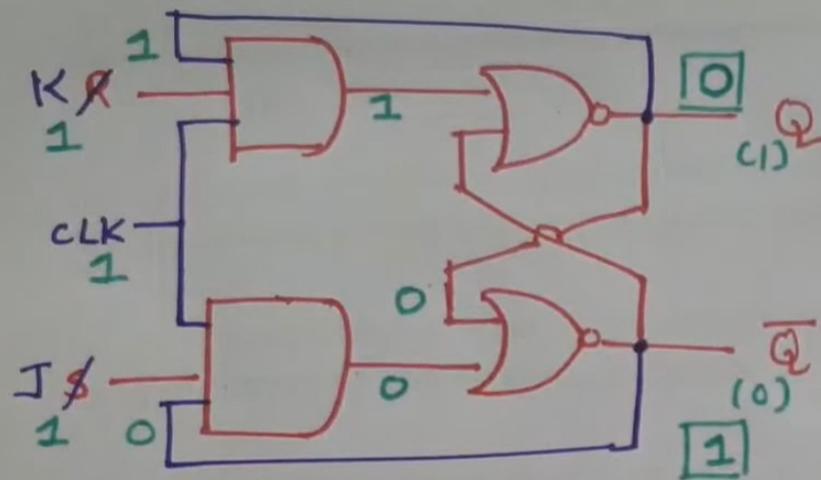
CLK	J	K	$Q$	$\bar{Q}$
0	x	x	...	...
1	0	0	...	...
1	0	1	0	1
1	1	0	1	0
1	1	1	...	...

CMOS JK Flip Flop using NOR gates



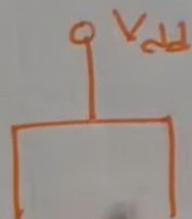
CLK	J	K	$Q$	$\bar{Q}$
0	x	x	...	...
1	0	0	...	...
1	0	1	0	1
1	1	0	1	0
1	1	1	1	0

For NOR gate, if any input is logic '1', then Output will be logic '0'.

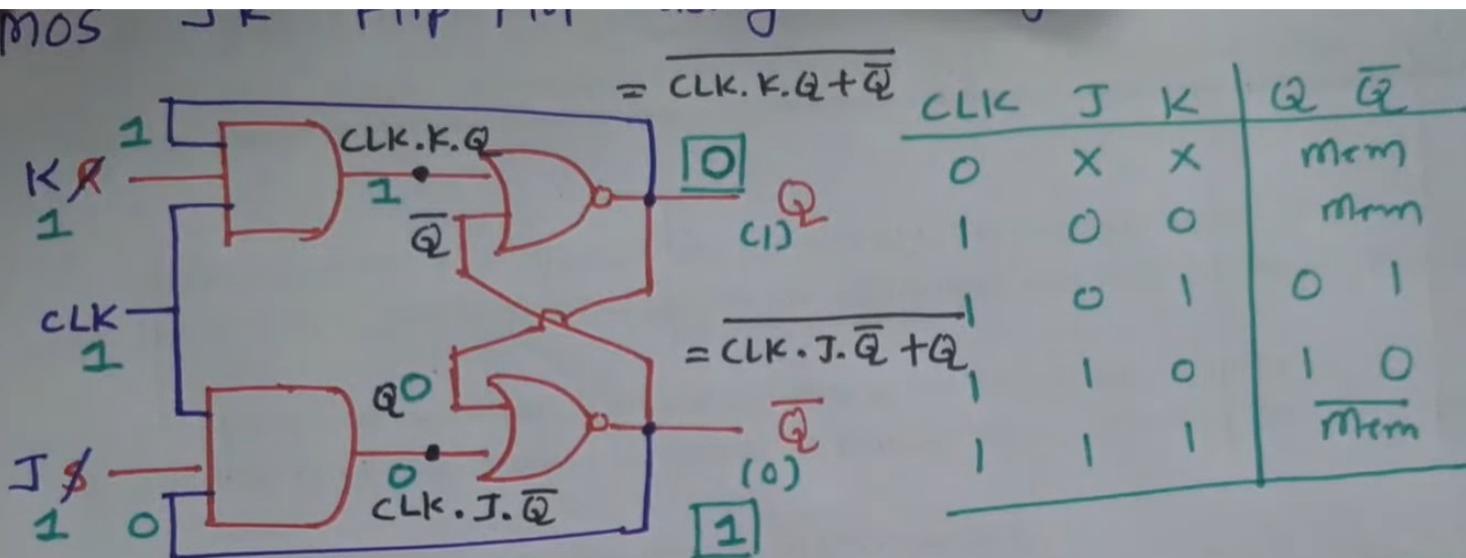


CLK	J	K	Q	$\bar{Q}$
0	X	X	Mem	
1	0	0	Mem	
1	0	1	0	1
1	1	0	1	0
1	1	1	Mem	

$\rightarrow$  (-) Operation  $\rightarrow$  pmos (Parallel)  $\rightarrow$  nmos (Series)  
 (+) Operation  $\rightarrow$  pmos (Series)  $\rightarrow$  nmos (Parallel)

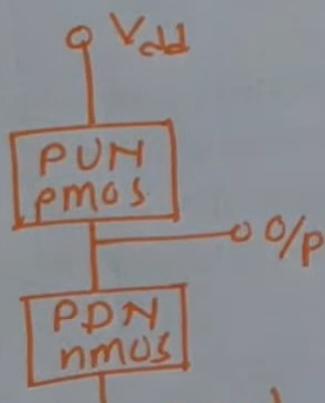


CMOS

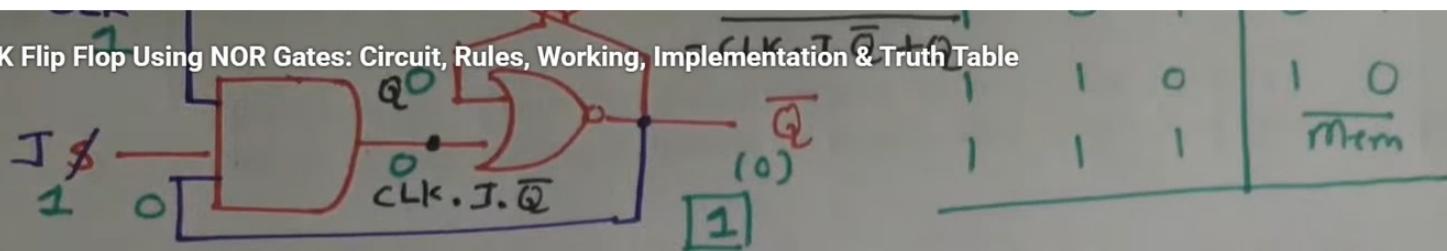


CLK	J	K	Q	$\bar{Q}$
			0	X
0	X	X	Mem	
1	0	0	Mem	
0	0	1	0	1
1	0	0	1	0
1	1	1	Mem	

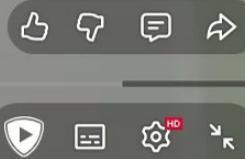
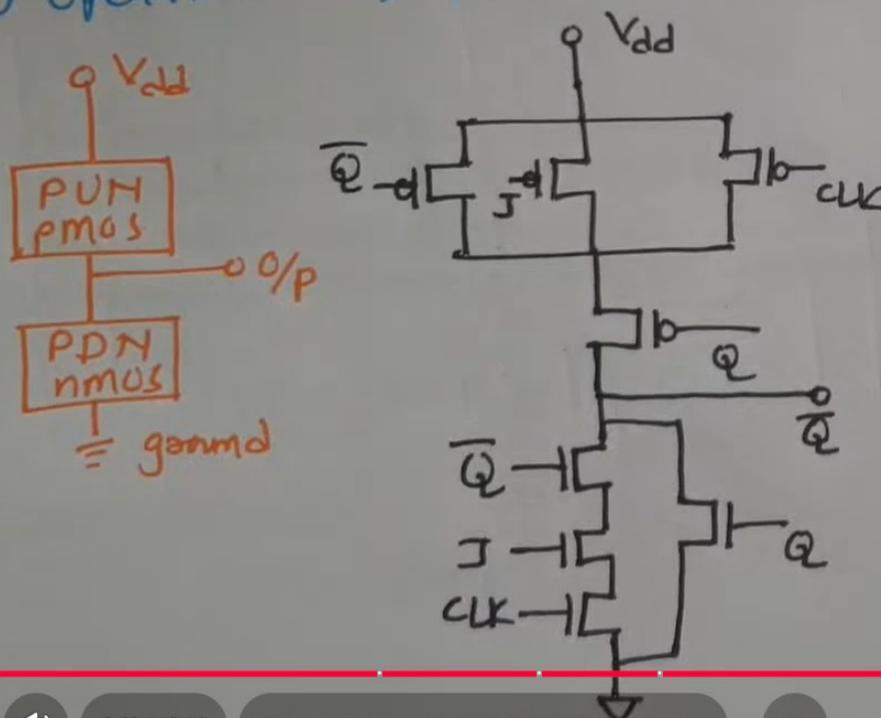
$\rightarrow$  (-) Operation  $\rightarrow$  pmos (Parallel)  $\rightarrow$  nmos (Series)  
 $\rightarrow$  (+) Operation  $\rightarrow$  pmos (Series)  $\rightarrow$  nmos (Parallel)



## CMOS JK Flip Flop Using NOR Gates: Circuit, Rules, Working, Implementation & Truth Table

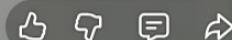
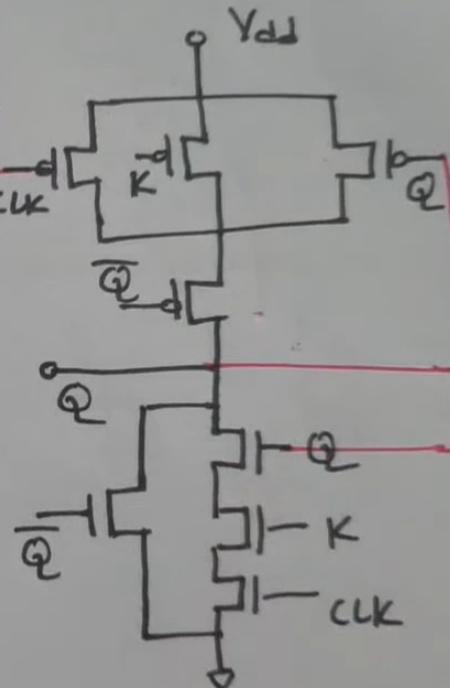
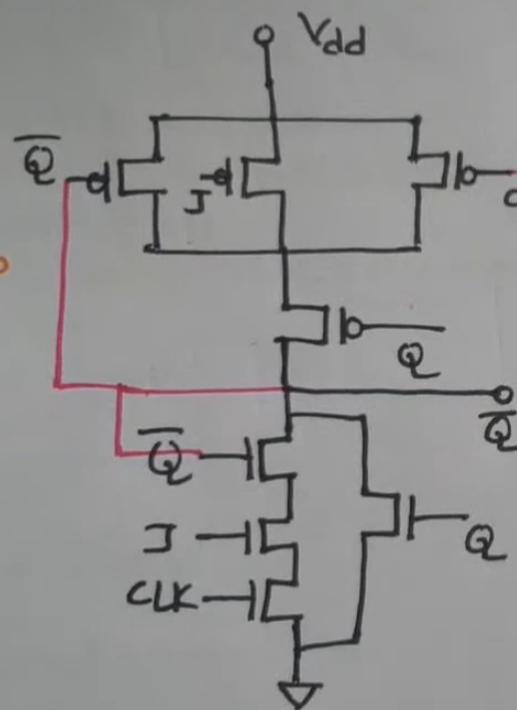
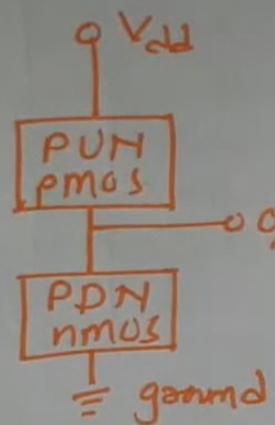


→ (.) Operation  $\rightarrow$  pmos (Parallel)  $\rightarrow$  nmos (Series)  
 (+) Operation  $\rightarrow$  rmos (Series)  $\rightarrow$  nmos (Parallel)



## CMOS JK Flip Flop Using NOR Gates: Circuit, Rules, Working, Implementation & Truth Table

→ (-) Operation → PMOS (Parallel) → nmos (Series)  
(+/-) Operation → PMOS (Series) → nmos (Parallel)



11:26 / 12:14

CMOS JK Flip Flop using NOR Gates implementation &gt;

