Dhirubhai Ambani Institute of Information and Communication Technology

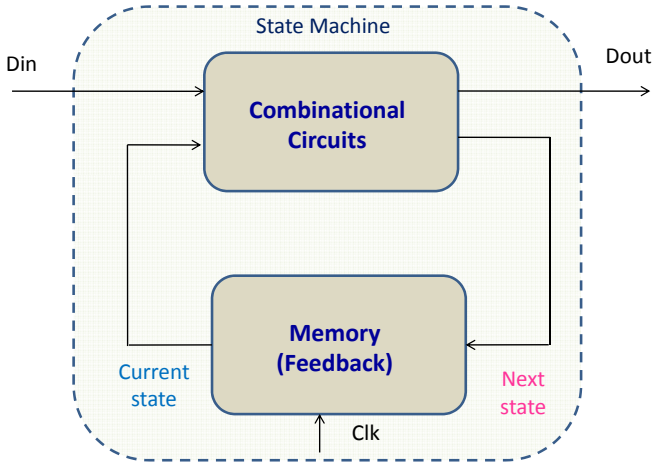
EL114

Digital Logic Design

Dr. Yash Agrawal
Assistant Professor,
DA-IICT, Gandhinagar

Introduction

Sequential Circuit



Introduction

How to realize Storage (Memory) Element

Basic Building Block for Storage/Memory

Latch

- Latch is level triggered single-bit memory element.
- Latch responds immediately to change in input(s)and (possibly in the presence of level enables control signal).
- Output can change multiple times during active enable (clock) signal.

Flip-flop

- Flip-flop is edge triggered single-bit memory element.
- Output can change only one time during clock cycle.

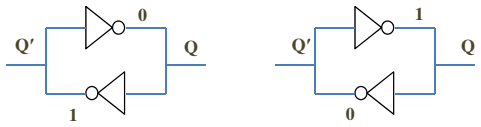
Basic Block for Storage/Memory

- Bi-stable Element

Bi-stable is the element/circuit having two stable states and is used to design latches and flip-flops.

Basic Bi-stable element

Cross-coupled inverters hold value Q and Q′



Cross-coupled inverters hold/store value Q and Q′

Basic Hardware Designs for Storage/Memory Element

Storage using Cross-Coupled Inverters

Cross-coupled inverters hold value Q and Q′

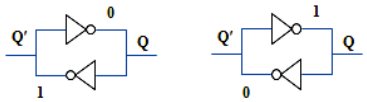
Read: Get value from either Q or Q′

Storage: Maintains its state as long as power is applied.

1. How to hold/store even when power is removed…?
2. How to feed-in input or change state…?

Basic Hardware Designs for Storage/Memory Element

How to feed-in input or change state…?

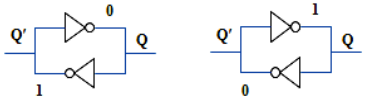


Option1:

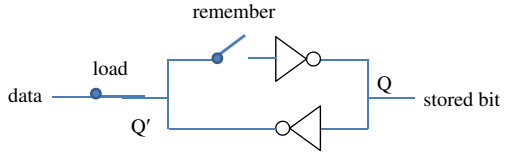
Write/Put opposite values on Q and Q′
This requires Analog Overdriving.

Basic Hardware Designs for Storage/Memory Element

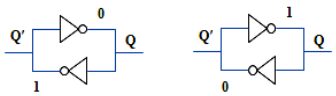
How to feed-in input or change state…?



Option2:
Using Digital Convention
Temporary break the feedback path.



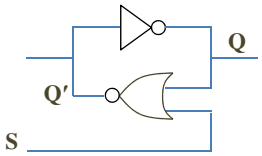
How to feed-in input or change state…?



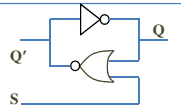
Option3:

Using Digital Convention

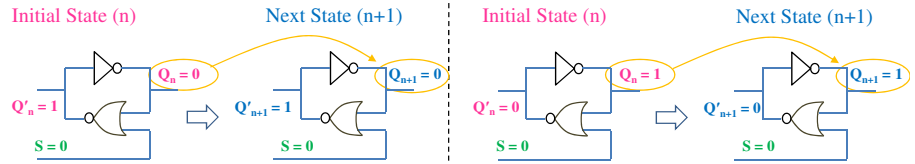
Storage and Setting output (O/P) using **NOR Gate** Logic



Storage and Setting output (O/P) using **NOR Gate** Logic

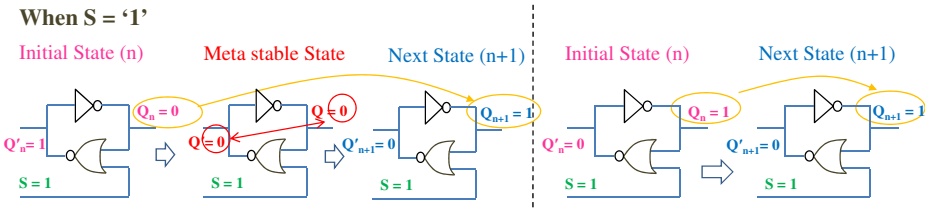


When S = '0'



Inference: For **S = '0'**, Q retains same logic in Next Stage. Hence acts as **storage** of O/P bit.

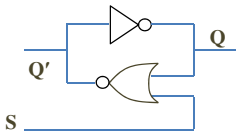
When S = '1'



Inference: For **S = '1'**, Q is always at logic high '1' in Next Stage. Hence acts as **setting** of O/P bit.

Using Digital Convention

Storage and Setting output (O/P) using **NOR Gate** Logic

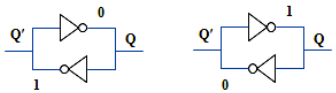


So using SET (S) bit, output bit (Q) can either

1. Retain (store) the previous bit when S = '0' or
2. Set to logic high (1) when S = '1'

Now, how to Reset output bit (Q = '0')….?

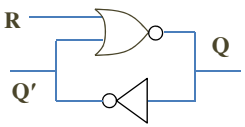
How to feed-in input or change state…?



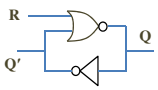
Option3:

Using Digital Convention

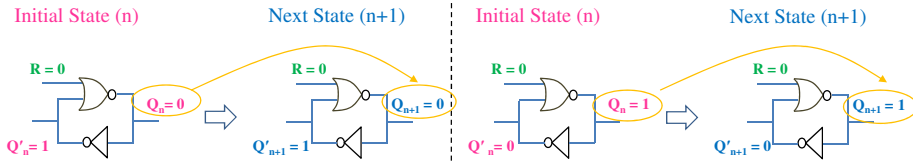
Storage and Re-setting output (O/P) using **NOR Gate** Logic



Storage and Re-setting output (O/P) using **NOR Gate** Logic

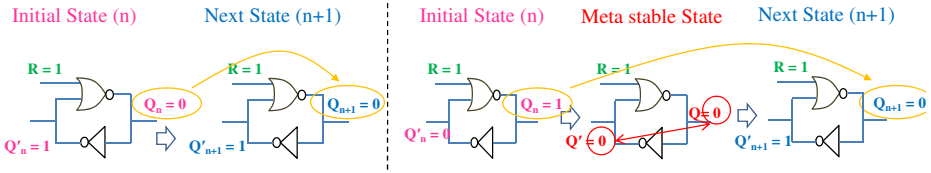


When R = '0'



Inference: For **R = '0'**, Q retains same logic in Next Stage. Hence acts as **storage** of O/P bit.

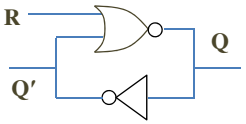
When R = '1'



Inference: For **R = '1'**, Q is always at logic low '0' in Next Stage. Hence acts as **re-setting** of O/P bit.

Using Digital Convention

Storage and Re-setting output (O/P) using **NOR Gate** Logic



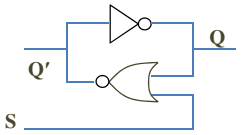
So using RESET (R) bit, output bit (Q) can either

1. Retain (store) the previous bit when R = '0' or
2. Re-set to logic low (0) when R = '1'

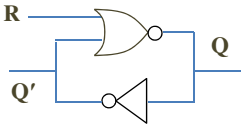
Can you combine both 'S' and 'R' bits…?

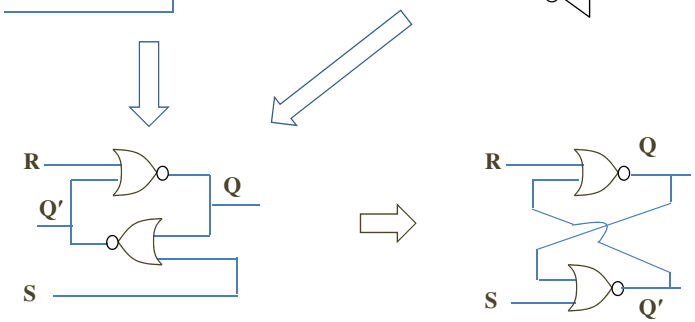
Cross-coupled NORs (S-R Latch)

Circuit for **S**etting Q (i.e. for storing '1')

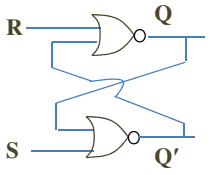


Circuit for **R**esetting Q (i.e. for storing '0')

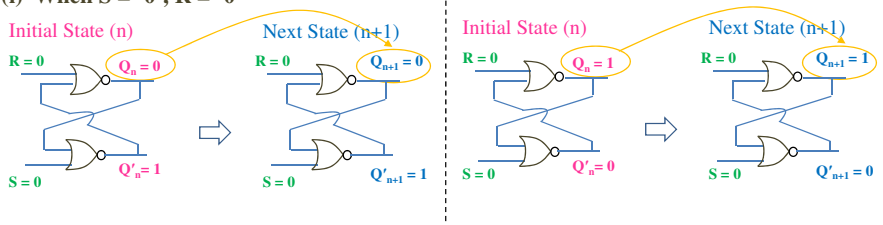




Operation: Cross-coupled NORs (S-R Latch)

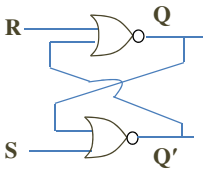


(i) When S = '0', R = '0'

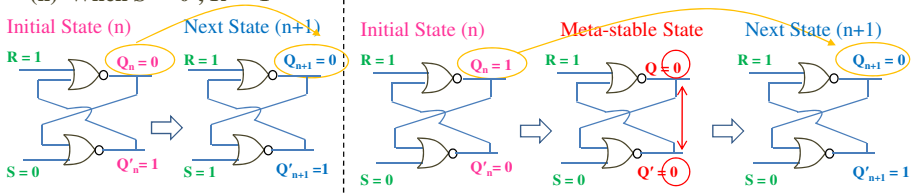


Inference: For **S = '0' and R = '0'**, Q retains same logic in Next Stage. Hence acts as **storage** of O/P bit.

Operation: Cross-coupled NORs (S-R Latch)

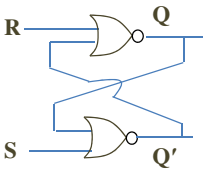


(ii) When S = '0', R = '1'

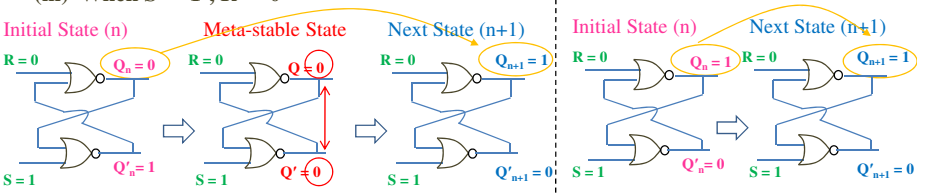


Inference: For **S = '0' and R = '1'**, Q is always low (logic '0') in Next Stage. Hence acts as **re-setting** of O/P .

Operation: Cross-coupled NORs (S-R Latch)

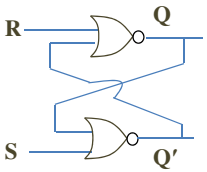


(iii) When S = '1', R = '0'

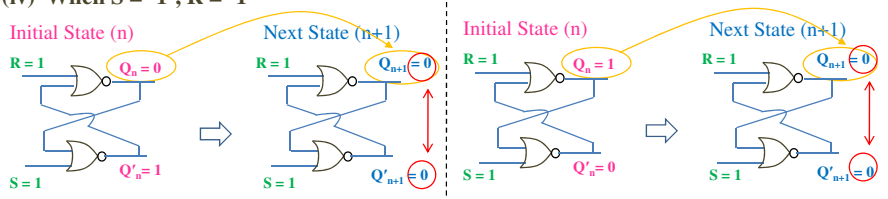


Inference: For **S = '1' and R = '0'**, Q is always high (logic '1') Next Stage. Hence acts as **setting** of O/P bit.

Operation: Cross-coupled NORs (S-R Latch)

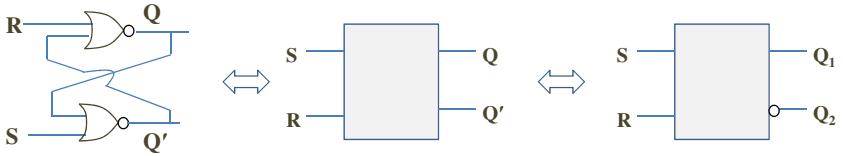


(iv) When S = '1', R = '1'



Inference: For **S = '1' and R = '1'**, Q goes to meta-stable (forbidden) in Next Stage. Hence this stage is invalid and not used.

Cross-coupled NORs (S-R Latch)

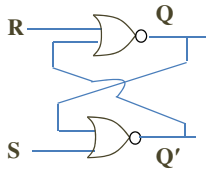


Truth Table

Input of Latch		Output of Latch	
S	**R**	**Q_{n+1}**	**State**
0	0	Previous State	No Change
0	1	0	Reset
1	0	1	Set
1	1	Undefined	Forbidden

* Q'_{n+1} is complementary to Q_{n+1}.

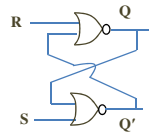
Cross-coupled NORs (S-R Latch)

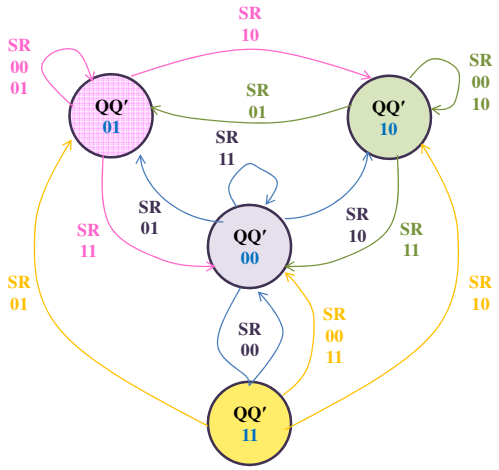


For **Combinational logic, truth table** is an important visualization and analyzing the function.

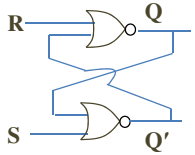
For **Sequential logic**, along with truth table, **state diagram** is an important visualization and analyzing the function.

State diagram: Cross-coupled NORs (S-R Latch)





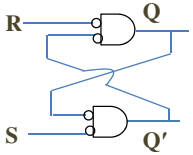
Cross-coupled NORs (S-R Latch)



It responds to **active-High inputs**

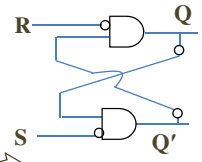
Demorgan's Law

Cross-coupled ANDs (S-R Latch)



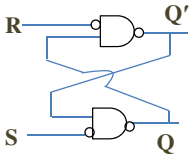
It responds to active-High inputs

Cross-coupled ANDs (S-R Latch)



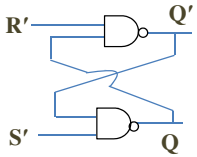
It responds to active-High inputs

Cross-coupled NANDs (S-R Latch)



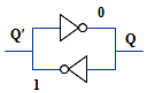
It responds to active-High inputs

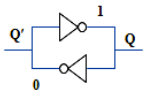
Cross-coupled NANDs (S′-R′ Latch)



It responds to **active-Low inputs**

How to feed-in input or change state…?

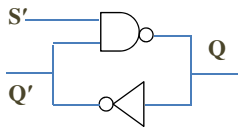




Option4:

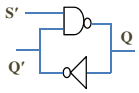
Using Digital Convention

Storage and Setting output (O/P) using **NAND Gate** Logic

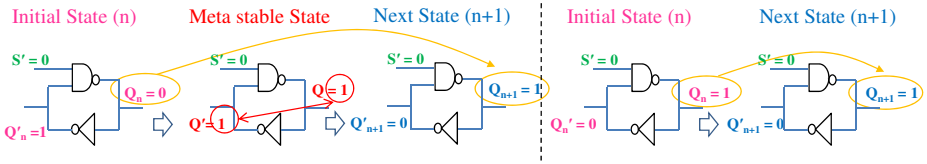


Basic Hardware Designs for Storage/Memory Element

Storage and Setting output (O/P) using **NAND Gate** Logic

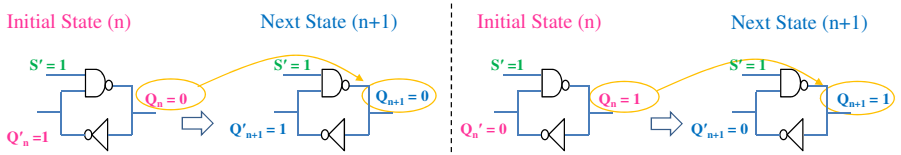


When S′ = '0'



Inference: For **S′ = '0'**, Q is always at logic high '1' in Next Stage. Hence acts as **setting** of O/P bit.

When S′ = '1'

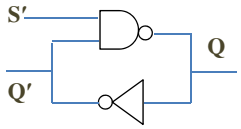


Inference: For **S′ = '1'**, Q retains same logic in Next Stage. Hence acts as **storage** of O/P bit.

Basic Hardware Designs for Storage/Memory Element

Using Digital Convention

Storage and Setting output (O/P) using **NAND Gate** Logic



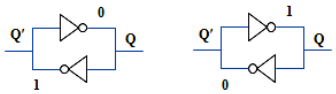
So using SET (S′) bit, output bit (Q) can either

1. Retain (store) the previous bit when S′ = '1' or
2. Set to logic high (1) when S′ = '0'

Now, how to Reset output bit (Q = '0')….?

Basic Hardware Designs for Storage/Memory Element

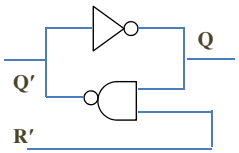
How to feed-in input or change state…?



Option4:

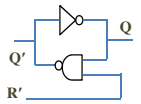
Using Digital Convention

Storage and Re-setting output (O/P) using **NAND Gate** Logic

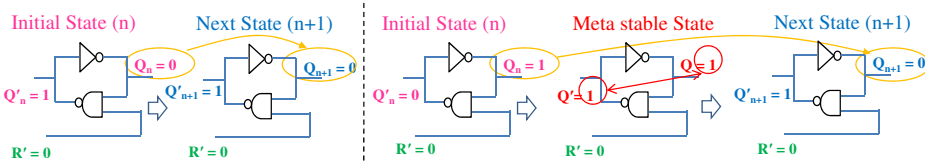


Basic Hardware Designs for Storage/Memory Element

Storage and Re-setting output (O/P) using **NAND Gate** Logic

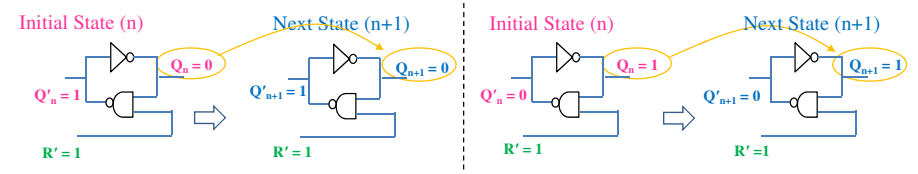


When R′ = '0'



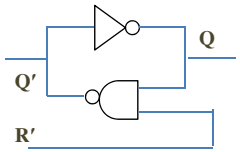
Inference: For **R′ = '0'**, Q is always at logic low '0' in Next Stage. Hence acts as **re-setting** of O/P bit.

When R′ = '1'



Inference: For **R′ = '1'**, Q retains same logic in Next Stage. Hence acts as **storage** of O/P bit.

Using Digital Convention
Storage and Re-setting output (O/P) using **NAND Gate** Logic

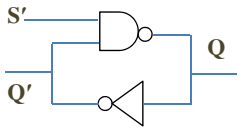


So using RESET (R′) bit, output bit (Q) can either
1. Retain (store) the previous bit when R′ = ‘1’ or
2. Re-set to logic low (0) when R′ = ‘0’

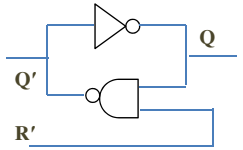
Can you combine both S′ and R′ bits…?

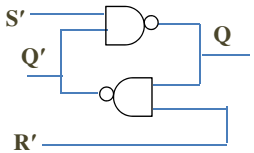
Cross-coupled NANDs (S′-R′ Latch)

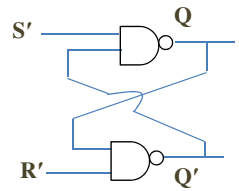
Circuit for **S**etting Q (i.e. for storing ‘1’)



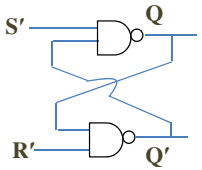
Circuit for **R**esetting Q (i.e. for storing ‘0’)



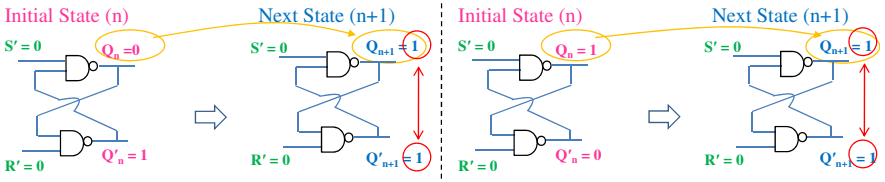




Operation: Cross-coupled NANDs (S′-R′ Latch)

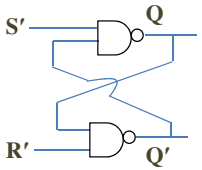


(i) When S′ = ‘0’, R′ = ‘0’

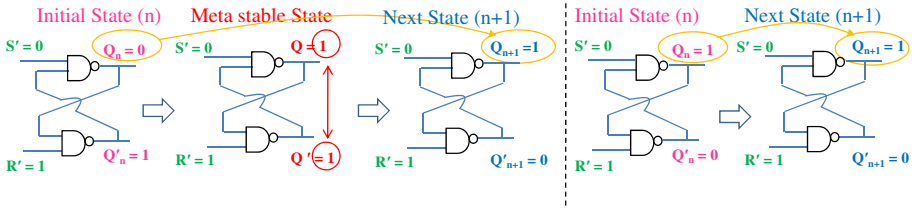


Inference: For **S′ = ‘0’ and R′ = ‘0’**, Q goes to meta-stable (forbidden) in Next Stage. Hence this stage is invalid and not used.

Operation: Cross-coupled NANDs (S′-R′ Latch)

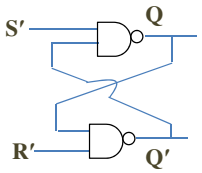


(ii) When S′ = ‘0’, R′ = ‘1’

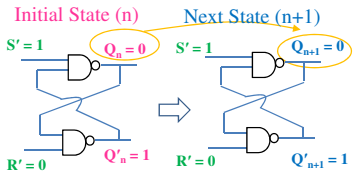


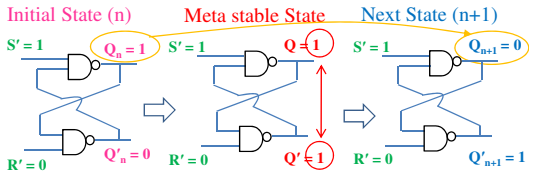
Inference: For **S′ = ‘0’ and R′ = ‘1’**, Q is always high (logic ‘1’) in Next Stage. Hence acts as **setting** of O/P bit.

Operation: Cross-coupled NANDs (S′-R′ Latch)



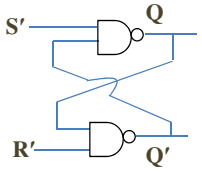
(iii) When S′ = '1', R′ = '0'



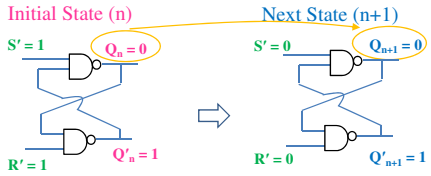


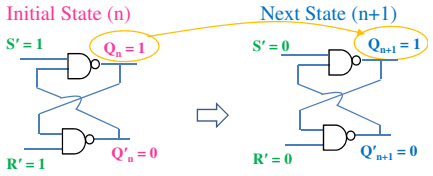
Inference: For **S′ = '1' and R′ = '0'**, Q is always low (logic '0') in Next Stage. Hence acts as **re-setting** of O/P .

Operation: Cross-coupled NANDs (S′-R′ Latch)



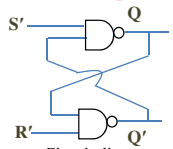
(iv) When S′ = '1', R′ = '1'



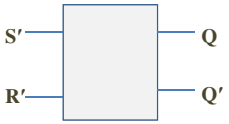


Inference: For **S′ = '1' and R′ = '1'**, Q retains same logic in Next Stage. Hence acts as **storage** of O/P bit.

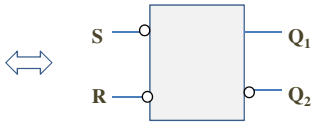
Cross-coupled NANDs (S′-R′ Latch)



Circuit diagram



Block diagram



Block diagram

Truth Table

Input of Latch		Output of Latch	
S′	R′	Q_{n+1}	State
0	0	Undefined	Forbidden
0	1	1	Set
1	0	0	Reset
1	1	Previous State	No Change

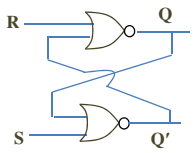
* Q'_{n+1} is complementary to Q_{n+1}.

S′-R′ hence acts as Negative logic latch.

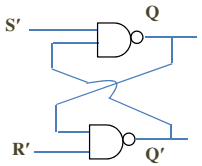
Assignment-7

1. Draw state diagram if cross-coupled NANDs (S′-R′ latch).

Cross-coupled NORs (S-R Latch)

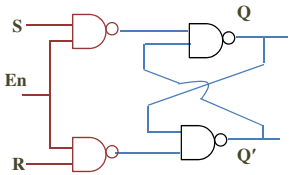


Cross-coupled NANDs (S′-R′ Latch)

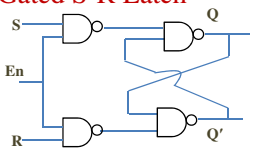


How to add controllability in it…?

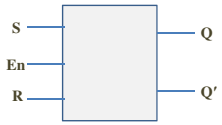
Gated S-R Latch



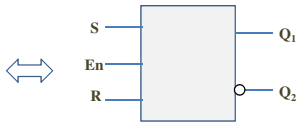
Gated S-R Latch



Circuit diagram



Block diagram



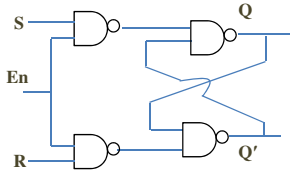
Block diagram

Truth Table

	Input of Latch		Output of Latch	
En	**S**	**R**	**Q_{n+1}**	**State**
1	0	0	Previous State	No Change
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	Undefined	Forbidden
0	x	x	Previous State	No Change

* Q'_{n+1} is complementary to Q_{n+1}.

Gated S-R Latch

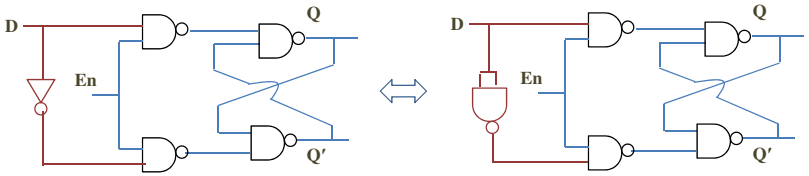


	Input of Latch		Output of Latch	
En	**S**	**R**	**Q_{n+1}**	**State**
1	0	0	Previous State	No Change
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	Undefined	Forbidden
0	x	x	Previous State	No Change

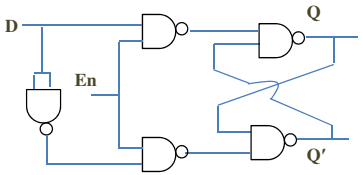
This condition is undesirable

How to avoid it…?

Gated D Latch



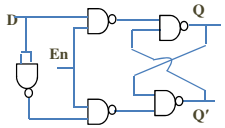
Gated D Latch



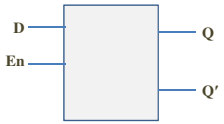
- D-Latch also known as transparent or delay latch.
- There is no ’00’ or ’11’ condition in this latch.

Basic Hardware Designs for Storage/Memory Element

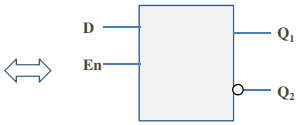
Gated D Latch



Circuit diagram



Block diagram



Block diagram

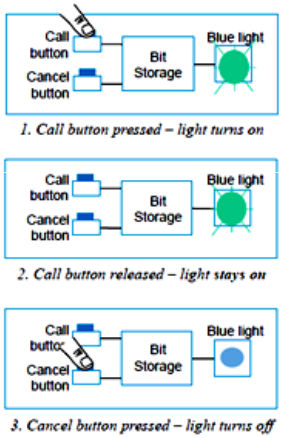
Truth Table

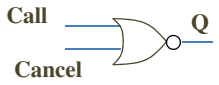
Input of Latch		Output of Latch	
En	**D**	**Q_{n+1}**	**State**
1	0	0	Reset
1	1	1	Set
0	x	Previous State	No Change

* Q'_{n+1} is complementary to Q_{n+1} .

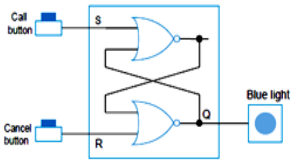
Exemplary Systems

1. Flight Attendant Call Button



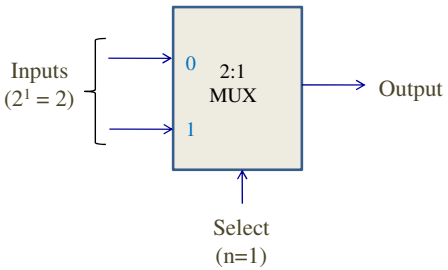


Does this logic work..?



Numericals

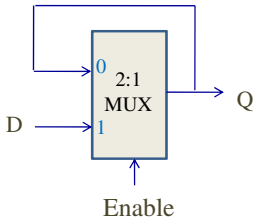
Implement Positive and Negative logic based D latch using 2:1 Multiplexer.



Numericals

Implement Positive and Negative logic based D latch using 2:1 Multiplexer.

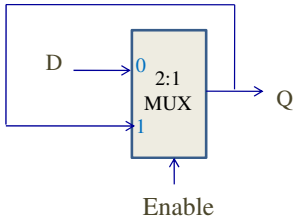
Positive logic based D-latch



Numericals

Implement Positive and Negative logic based D latch using 2:1 Multiplexer.

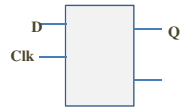
Negative logic based D-latch

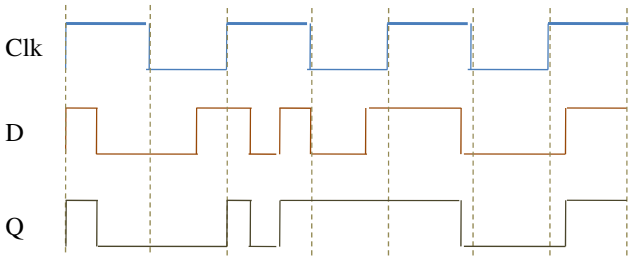


Issues/Limitations with Latches

Latches are level sensitive circuits. Output changes for entire period of logic high/low signal.

Consider Positive level triggered D-Latch





Issues/Limitations with Latches

Latches are level sensitive circuits. Output changes for entire period of logic high/low signal.

Many times this is not desirables like in memories, processors.

How to resolve this timing issues..?

Flip-flops (edge triggered circuits)

This we will discuss Next….

Assignment-7

1. Draw state diagram if cross-coupled NANDs (S′-R′ latch).