S-R Latch with NORs

R (reset)

Q

S (set)

$\overline{Q}$

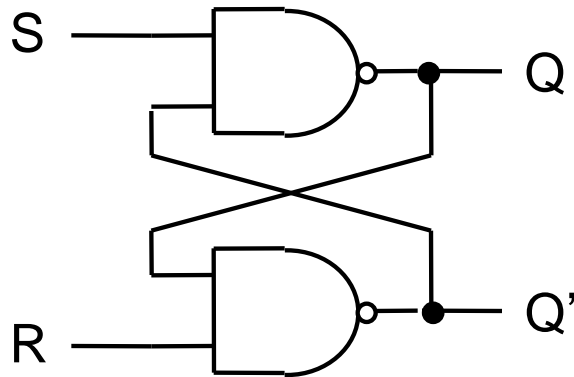| S	R	Q	Q'	
1	1	0	0	Undefined
1	0	1	0	Set
0	1	0	1	Reset
0	0			
		0	1	Stable
		1	0	

S-R latch made from cross-coupled NORs

If Q = 1, set state

If Q = 0, reset state

Usually S=0 and R=0

S=1 and R=1 generates unpredictable results

S-R Latch with NANDs

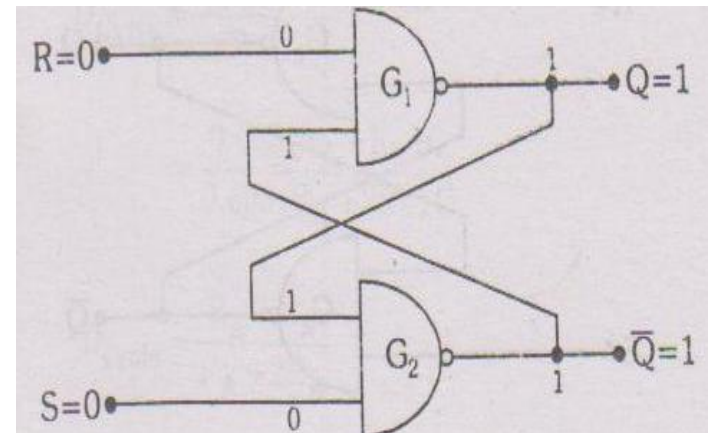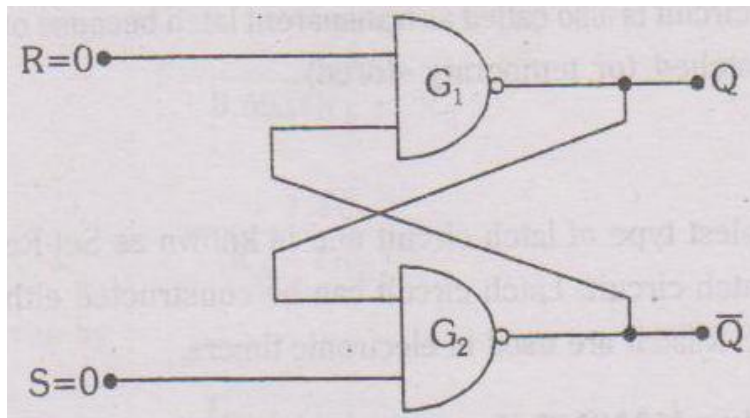| S	R	Q	Q'	
0	0	1	1	Disallowed
0	1	1	0	Set
1	0	0	1	Reset
1	1			
		0	1	Store
		1	0	

Latch made from cross-coupled NANDs
Sometimes called S'-R' latch
Usually S=1 and R=1
S=0 and R=0 generates unpredictable results
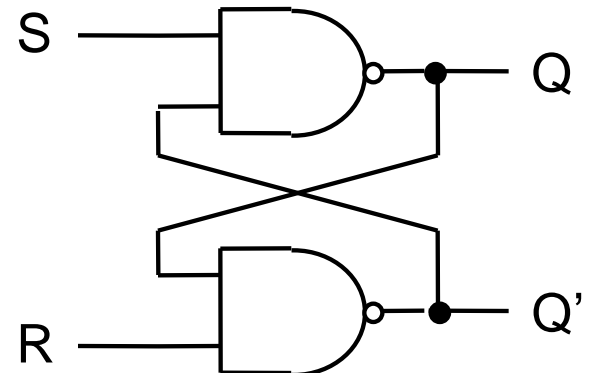
Case I: R
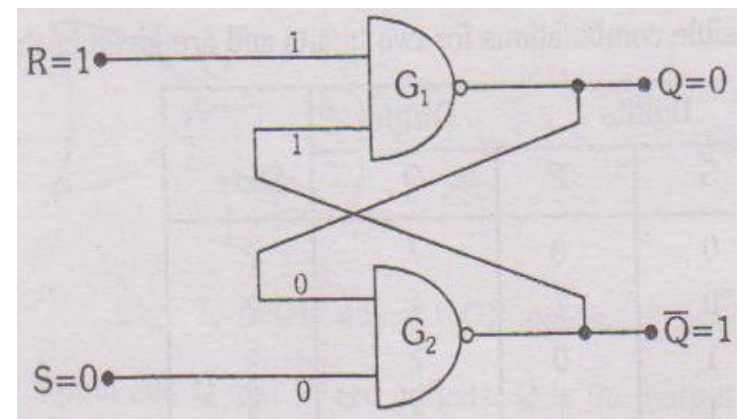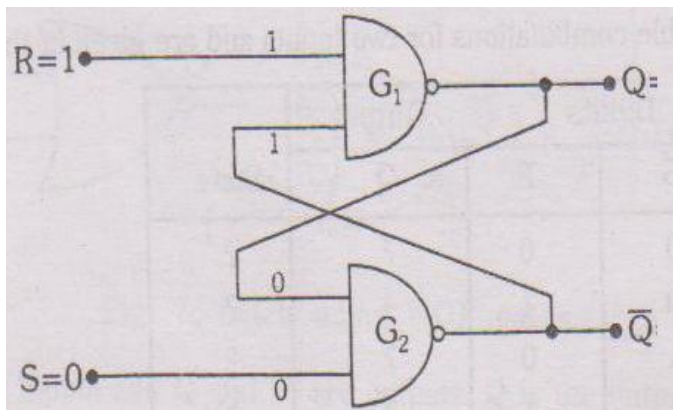
Inputs		Output	
S	R	Q	State
0	0	X	Invalid
0	1	0	Reset
1	0	1	Set
1	1	Q	No Change

Case II: S = 0 , R = 1

Inputs		Output	
S	R	Q	State
0	0	X	Invalid
0	1	0	Reset
1	0	1	Set
1	1	Q	No Change

Case III: S = 1 , R = 0

Inputs		Output	
S	R	Q	State
0	0	X	Invalid
0	1	0	Reset
1	0	1	Set
1	1	Q	No Change
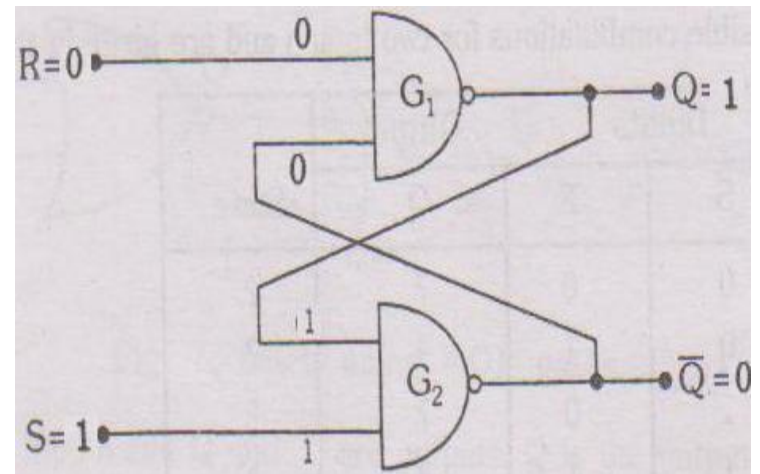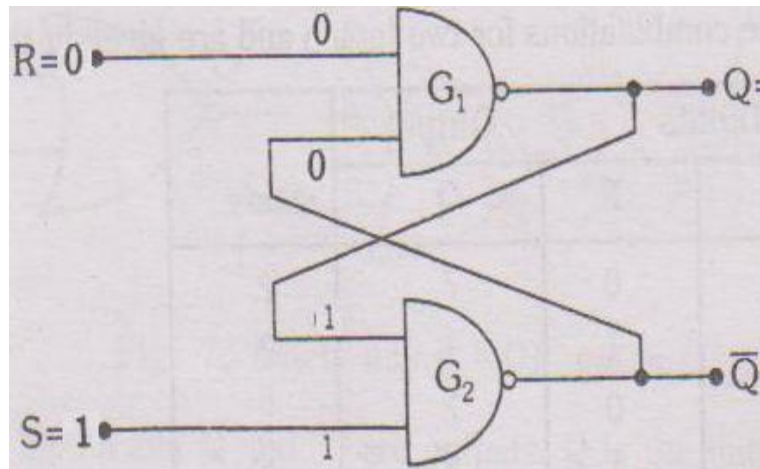
Case III: S = 1 , R = 0

Inputs		Output	
S	R	Q	State
0	0	X	Invalid
0	1	0	Reset
1	0	1	Set
1	1	Q	No Change

S-R Latches
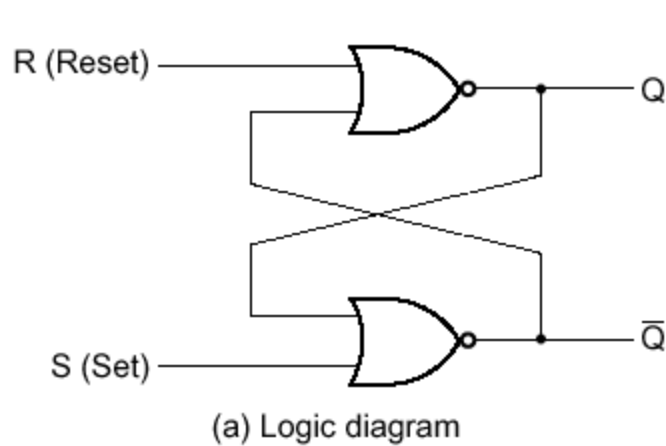
(a) Logic diagram

S	R	Q	Q'	
1	1	0	0	Undefined
1	0	1	0	Set
0	1	0	1	Reset
0	0			
		0	1	Stable
		1	0	

(a) Logic diagram
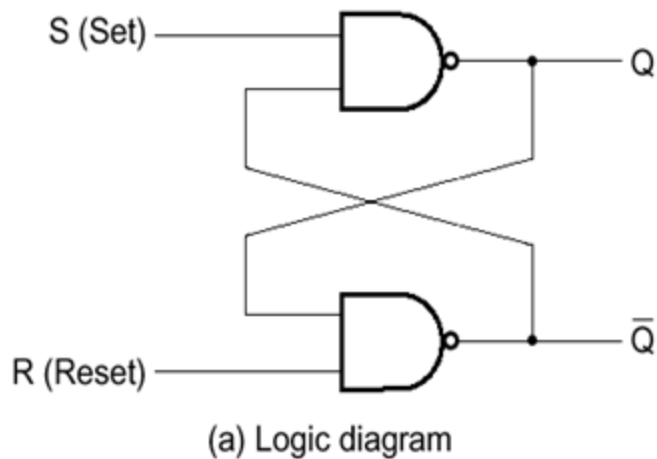
| S	R	Q	Q'	
0	0	1	1	Disallowed
0	1	1	0	Set
1	0	0	1	Reset
1	1			
		0	1	Store
		1	0	

Summary

- Latches are based on combinational gates (e.g. NAND, NOR)
- Latches store data even after data input has been removed
- S-R latches operate like cross-coupled inverters with control inputs (S = set, R = reset)
- With additional gates, an S-R latch can be converted to a D latch (D stands for data)
- D latch is simple to understand conceptually
  - When C = 1, data input D stored in latch and output as Q
  - When C = 0, data input D ignored and previous latch value output at Q
- Next time: more storage elements!

Gated latches(Clocked Flip Flop)

S-R Latch with control input

Gated
S-R
Latch

Logic Symbol

$C = 0$ disables all latch state changes
Control signal enables data change when $C = 1$
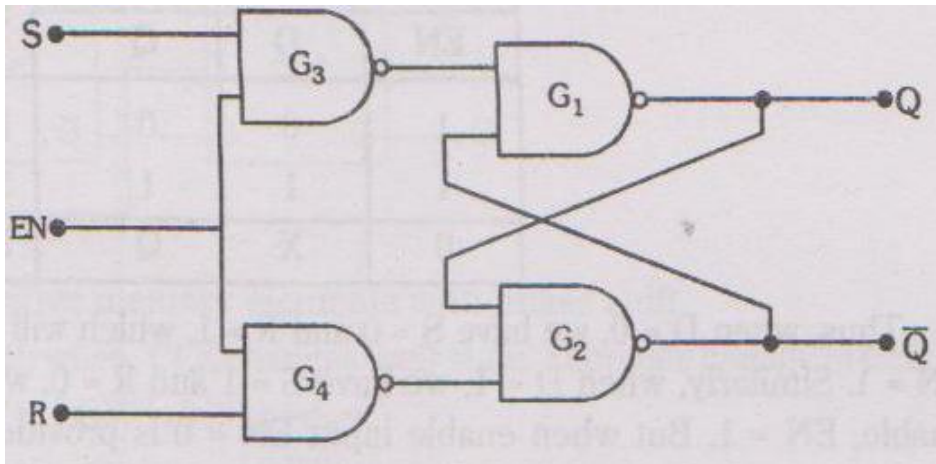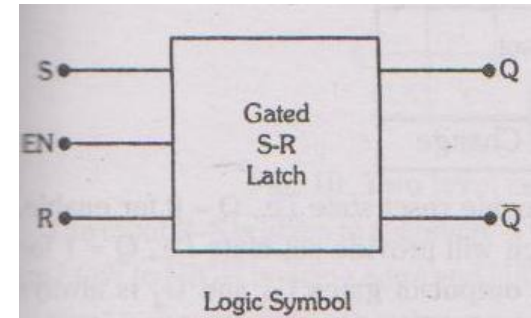
C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; Reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table