SEQUENTIAL LOGIC

SYNCHRONOUS SEQUENTIAL CIRCUITS
ASYNCHRONOUS SEQUENTIAL CIRCUITS
LATCHES(SR AND S'R' LATCH)
CONTROLLED LATCHES
D-LATCH

DIGITAL LOGIC DESIGN

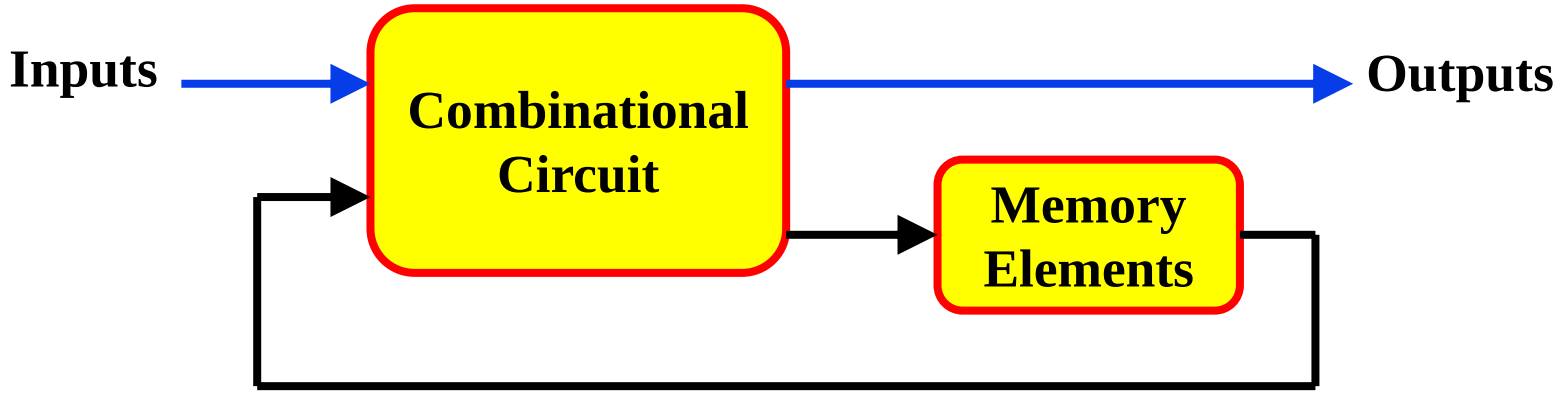
Iqra Chaudhary (Lecturer CS dept. NUML)

The story so far ...

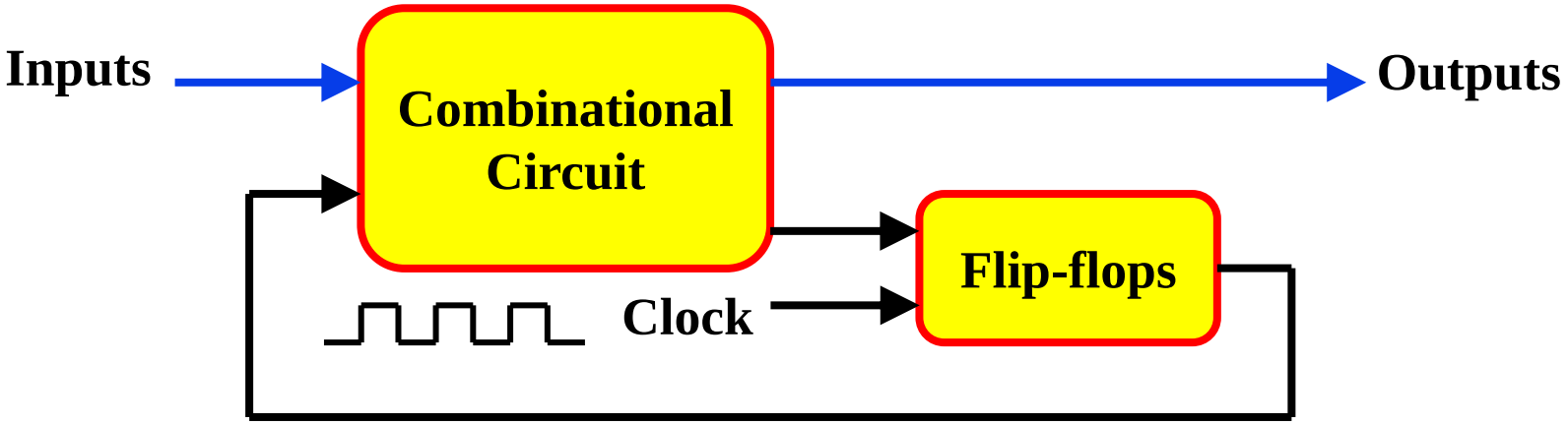
- **Logical operations which respond to *combinations* of inputs to produce an output.**
  - **Call these *combinational logic* circuits.**
- **For example, can add two numbers. But:**
  - **No way of adding two numbers, then adding a third (a sequential operation);**
  - **No way of remembering or storing information after inputs have been removed.**
- **To handle this, we need *sequential logic* capable of storing intermediate (and final) results.**

Sequential Circuits

★ Asynchronous

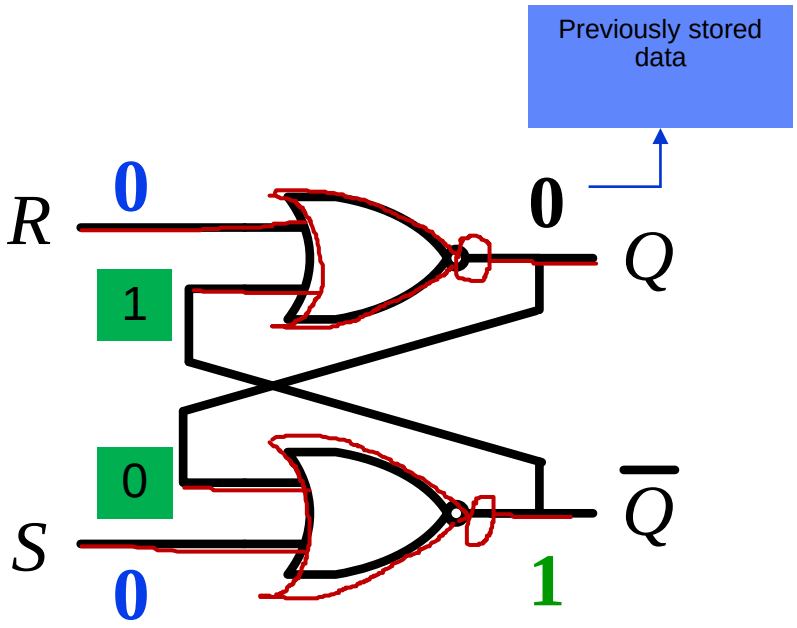


★ Synchronous



Latches (using Nor gate)

★ *SR* Latch



S	R	Q_0	Q	Q'
0	0	0	0	1

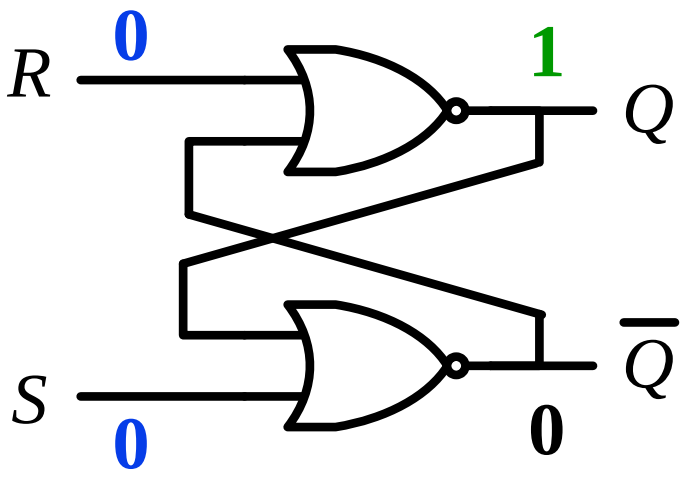
$Q = Q_0$

Initial Value

Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Latches

★ *SR* Latch



Previously stored value

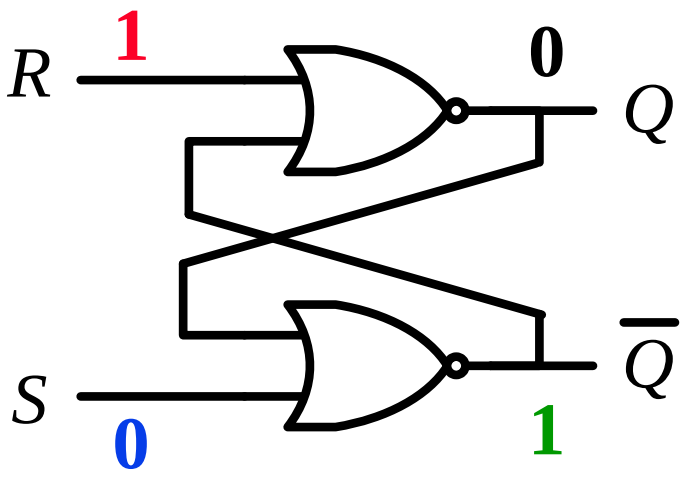
Current value

S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	$Q = Q_0$

Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Latches

★ *SR* Latch



Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

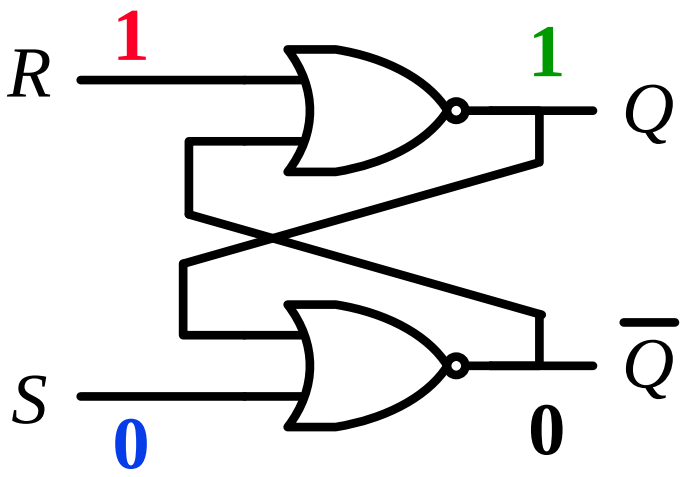
Previously stored value

Current value

S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$

Latches

★ *SR* Latch



Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Previously stored value

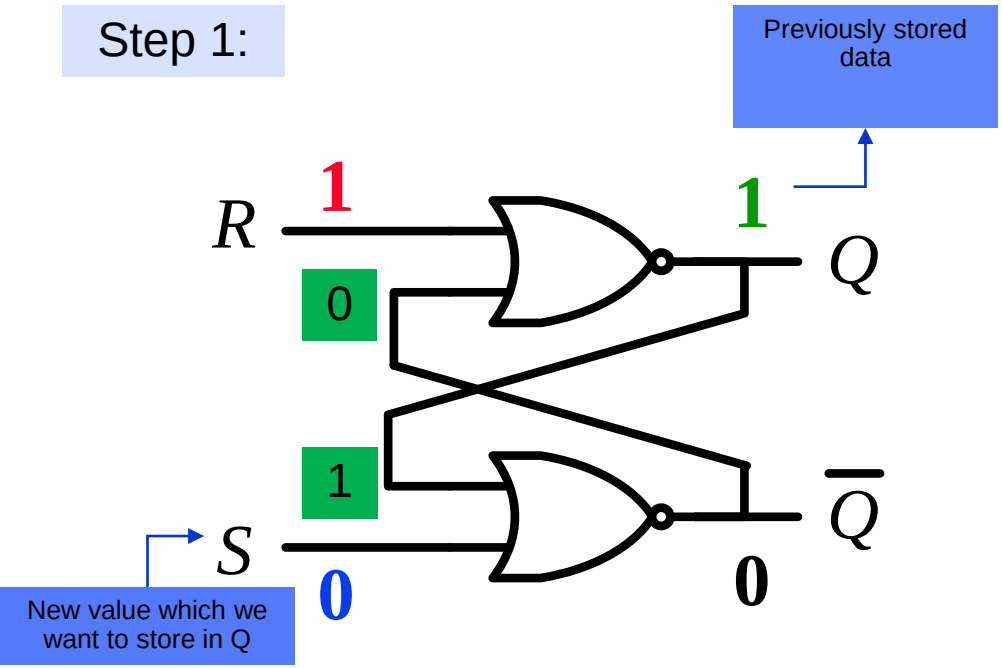
Current value

S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	$Q = 0$

Latches (Repeat)

★ *SR* Latch

Step 1:



Previously stored value			Current value	
S	R	Q_0	Q	Q'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1		

$Q = Q_0$

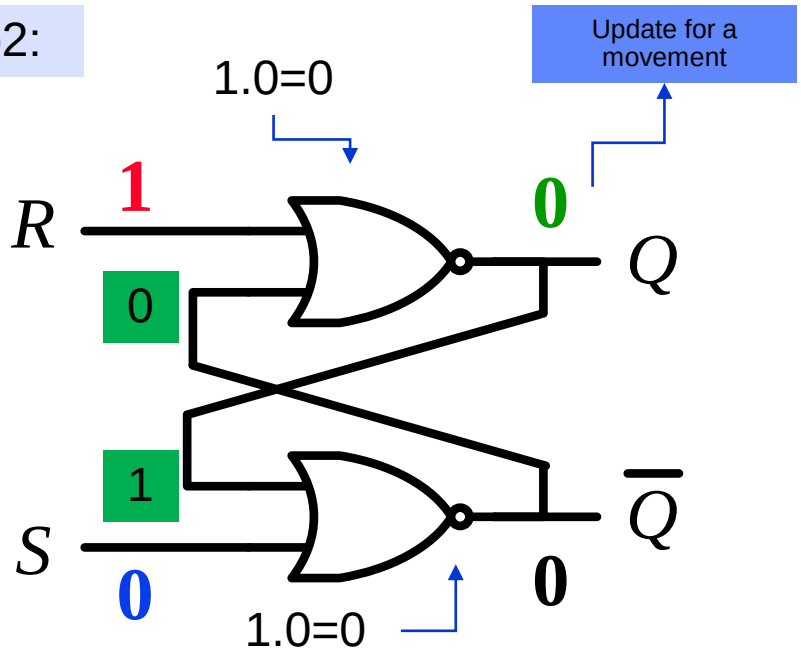
$Q = 0$

Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Latches (Repeat)

★ *SR* Latch

Step2:



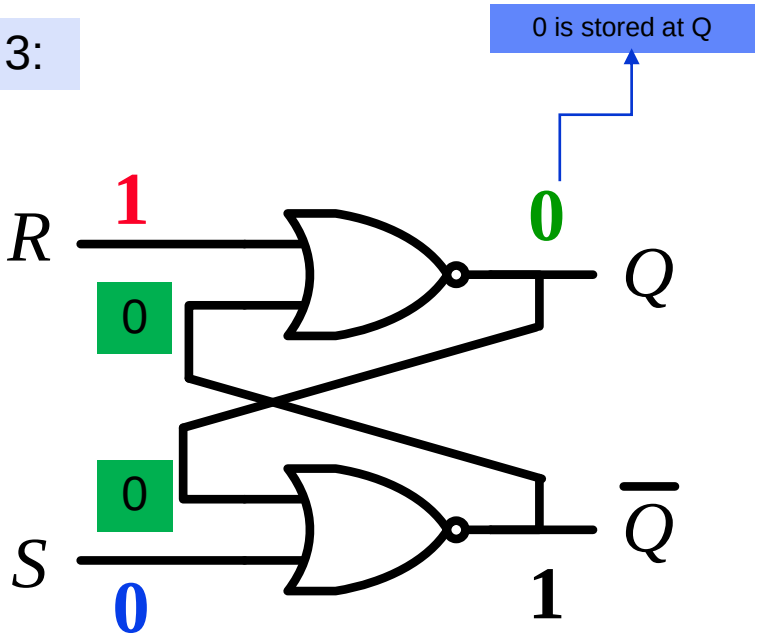
Previously stored value			Current value		
S	*R*	Q_0	*Q*	*Q'*	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1			

Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Latches (Repeat)

★ *SR* Latch

Step 3:



Previously stored value

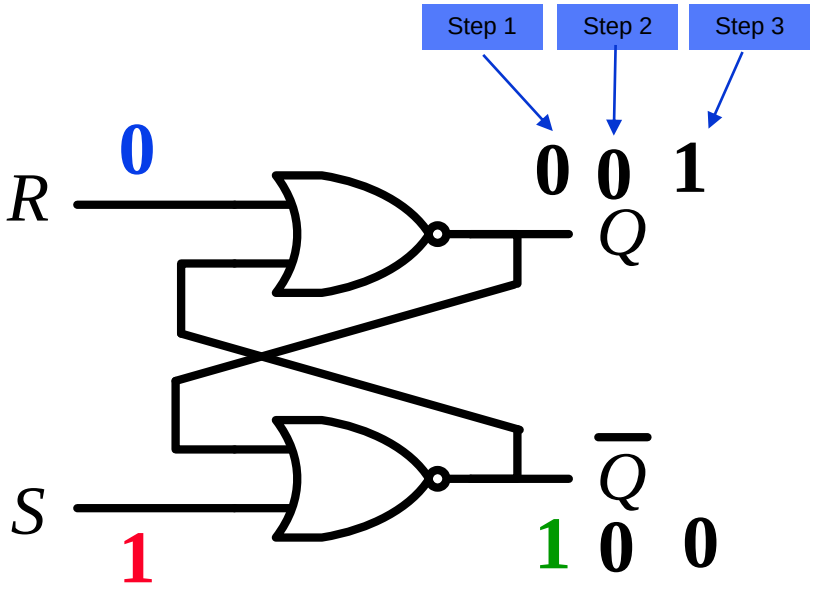
Current value

S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	

Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Latches

★ *SR* Latch

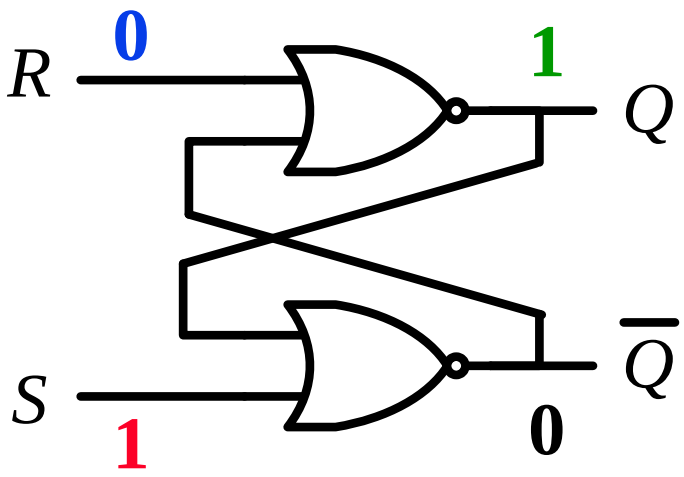


Previously stored value			Current value		
S	*R*	Q_0	*Q*	*Q'*	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	
1	0	0	1	0	$Q = 1$

Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Latches

★ *SR* Latch



Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

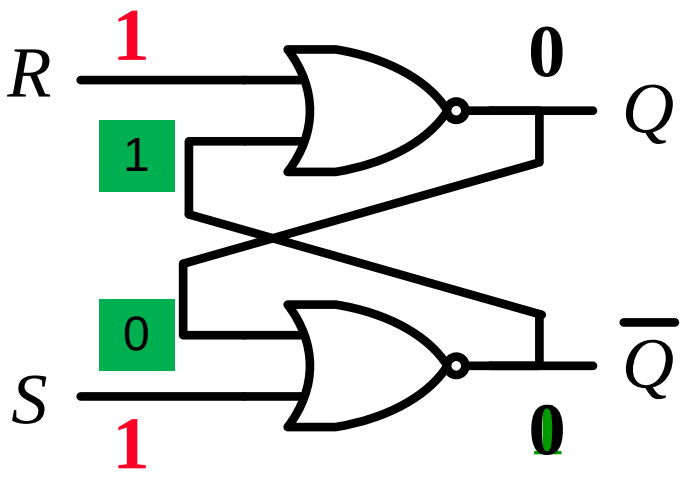
Previously stored value

Current value

S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	
1	0	0	1	0	$Q = 1$
1	0	1	1	0	$Q = 1$

Latches

★ *SR* Latch



Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

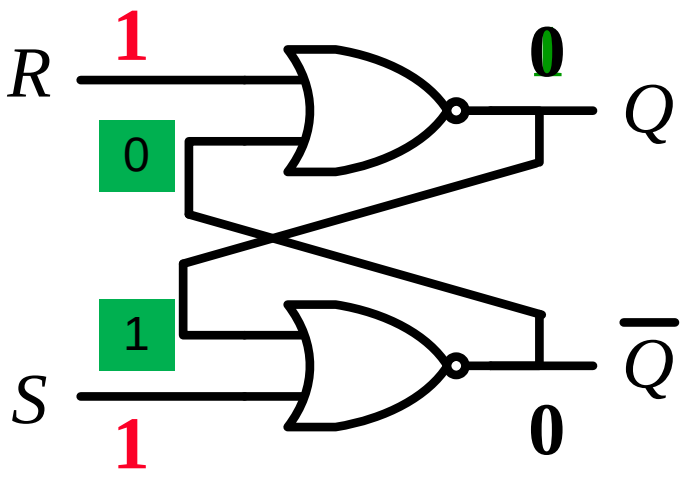
Previously stored value

Current value

S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	
1	0	0	1	0	$Q = 1$
1	0	1	1	0	
1	1	0	0	0	$Q = Q'$

Latches

★ *SR* Latch



Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

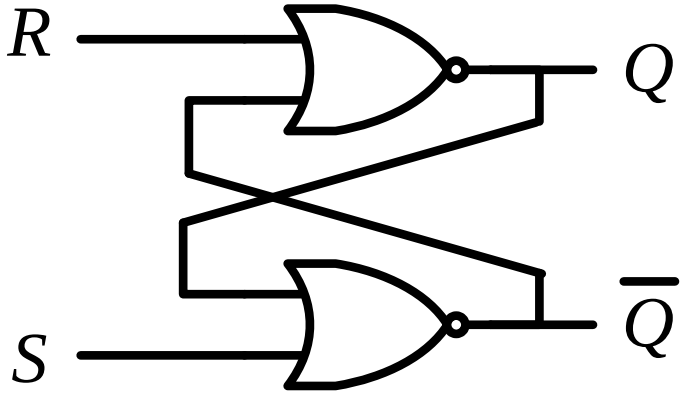
Previously stored value

Current value

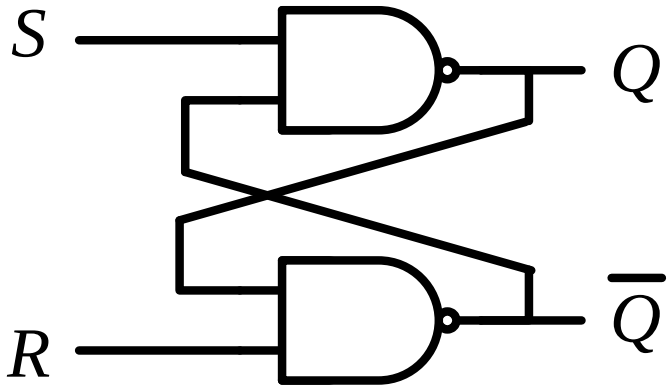
S	R	Q_0	Q	Q'	
0	0	0	0	1	$Q = Q_0$
0	0	1	1	0	
0	1	0	0	1	$Q = 0$
0	1	1	0	1	
1	0	0	1	0	$Q = 1$
1	0	1	1	0	
1	1	0	0	0	$Q = Q'$
1	1	1	0	0	$Q = Q'$

Latches

★ *SR* Latch



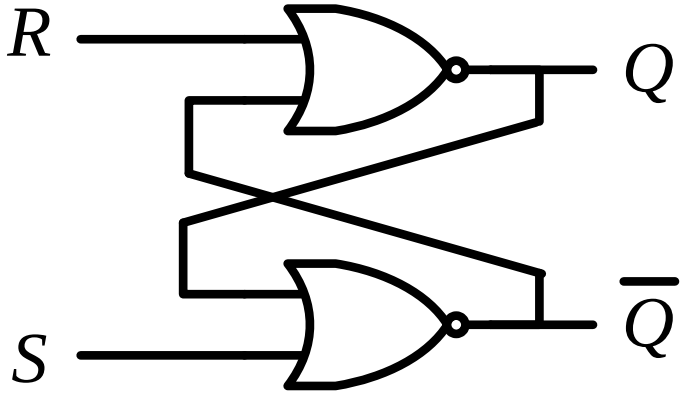
S	R	Q	
0	0	Q_0	No change
0	1	0	Reset
1	0	1	Set
1	1	Q=Q'=0	Invalid/undefine state/forbidden state



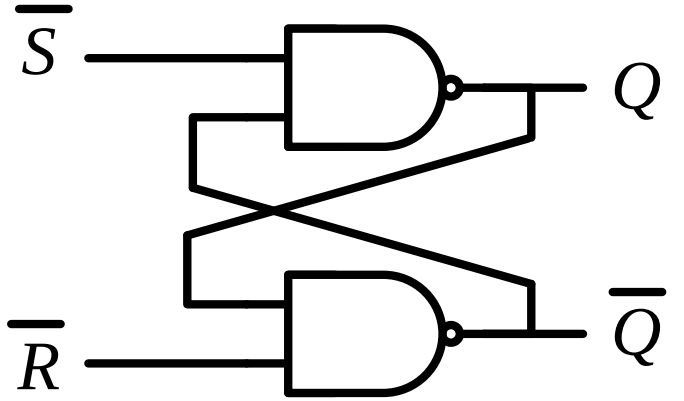
S	R	Q	
0	0	Q=Q'=1	Invalid
0	1	1	Set
1	0	0	Reset
1	1	Q_0	No change

Latches

★ *SR* Latch



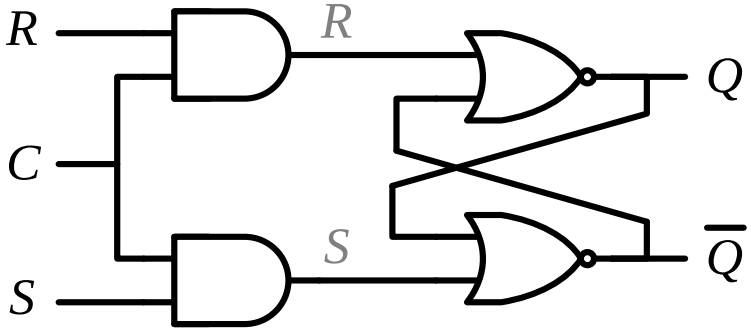
S *R*	*Q*	
0 0	Q_0	No change
0 1	0	Reset
1 0	1	Set
1 1	*Q*=*Q'*=0	Invalid

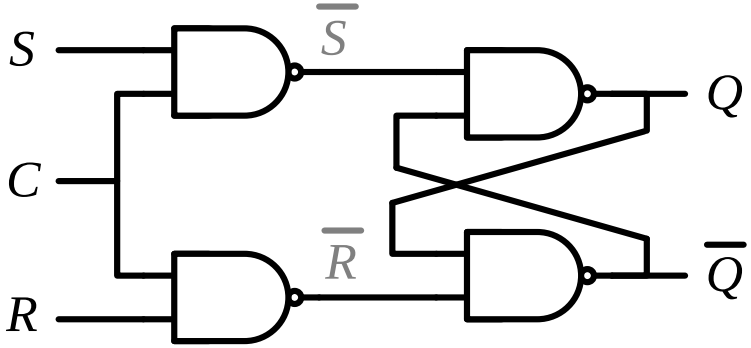


S' *R'*	*Q*	
0 0	*Q*=*Q'*=1	Invalid
0 1	1	Set
1 0	0	Reset
1 1	Q_0	No change

Controlled Latches

★ *SR* Latch with Control Input

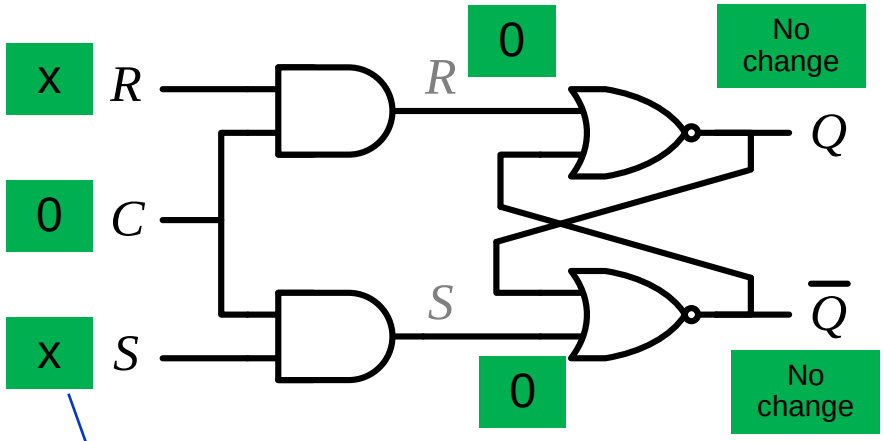


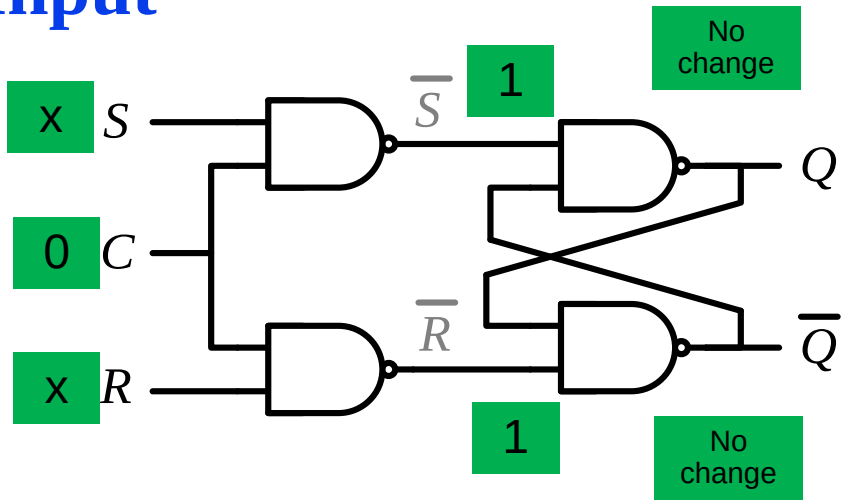


C S R	Q	
0 x x	Q_0	No change
1 0 0	Q_0	No change
1 0 1	0	Reset
1 1 0	1	Set
1 1 1	$Q=Q'$	Invalid

Controlled Latches

★ *SR* Latch with Control Input





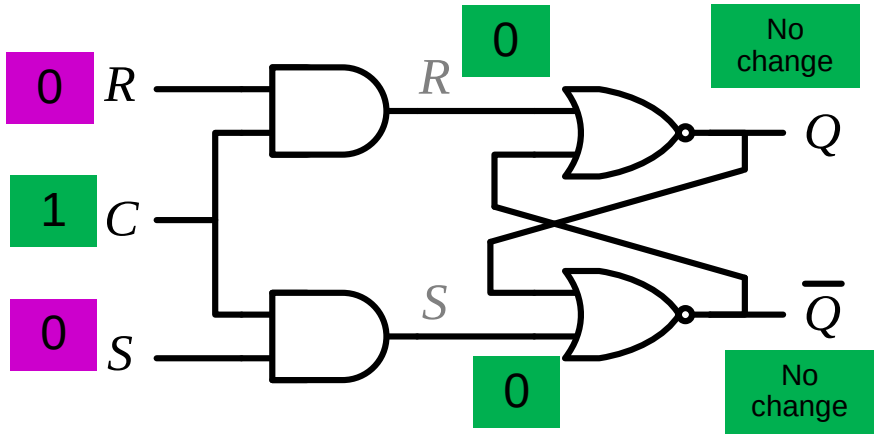
The value of x can be 1 or 0

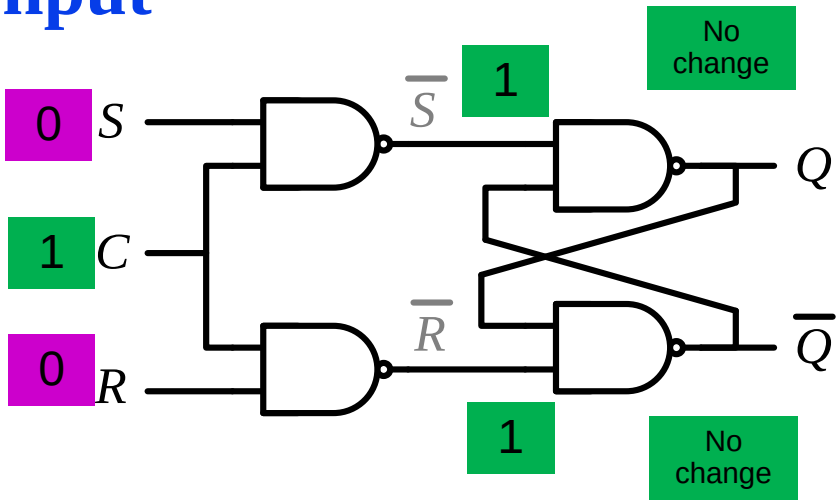
C	S	R	Q
0	x	x	Q_0

No change

Controlled Latches

★ *SR* Latch with Control Input





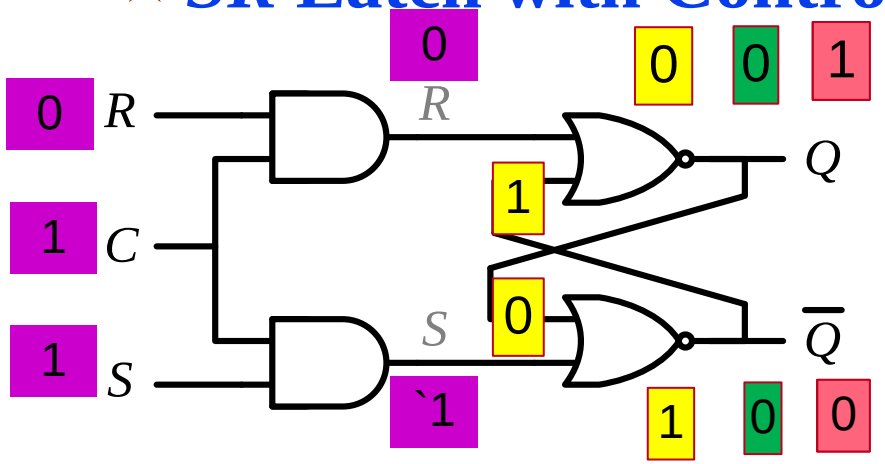
C S R	Q
0 x x	Q_0
1 0 0	Q_0

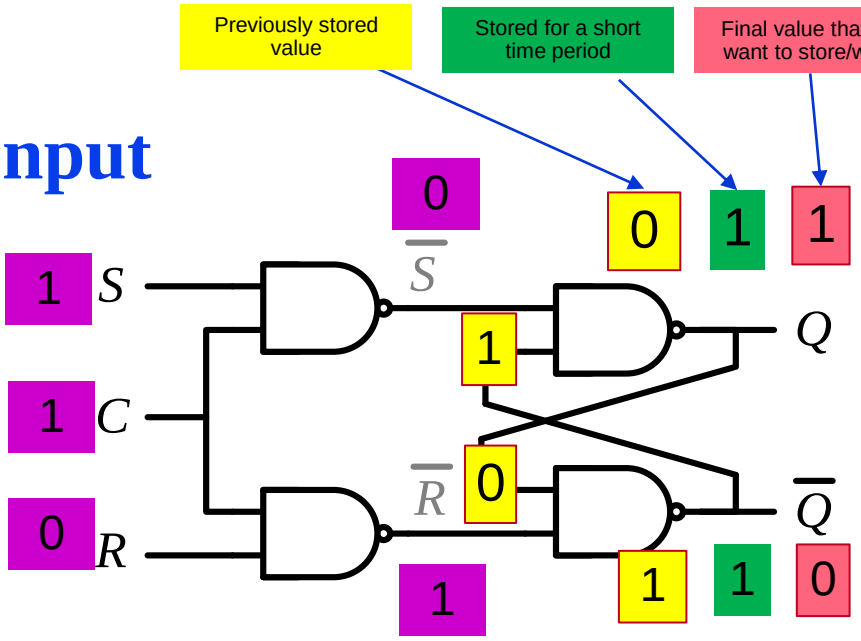
No change

No change

Controlled Latches

★ *SR* Latch with Control Input





C S R	Q
0 x x	Q_0
1 0 0	Q_0
1 1 0	1

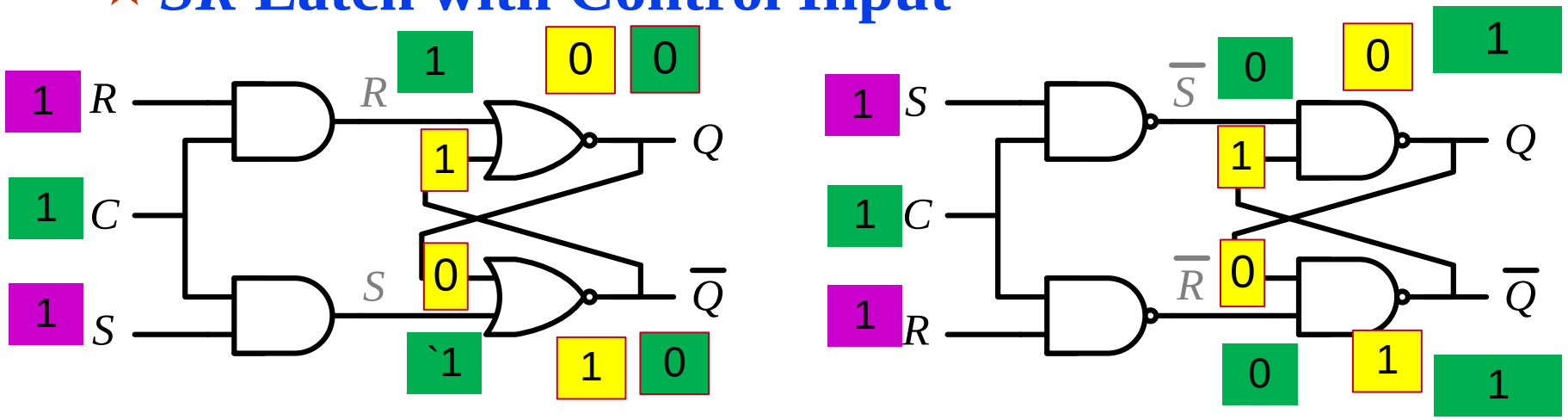
No change

No change

Set

Controlled Latches

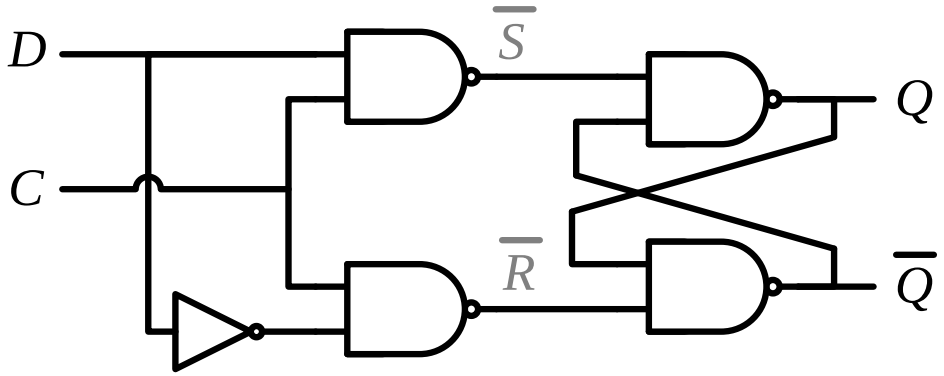
★ *SR* Latch with Control Input



C S R	Q	
0 x x	Q_0	No change
1 0 0	Q_0	No change
1 1 0	1	Set
1 1 1	Q=Q'	Invalid/Forbidden state
1 0 1	0	Reset

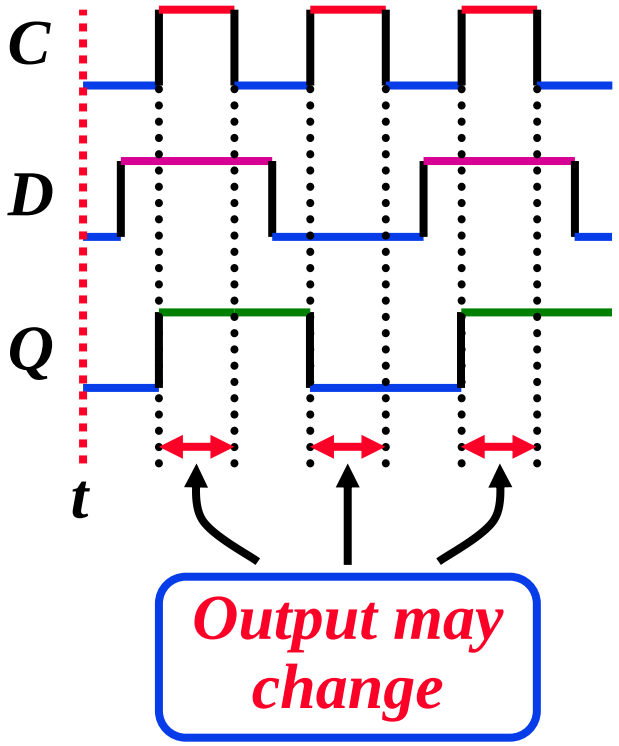
Controlled Latches

★ *D* Latch (*D* = *Data*)



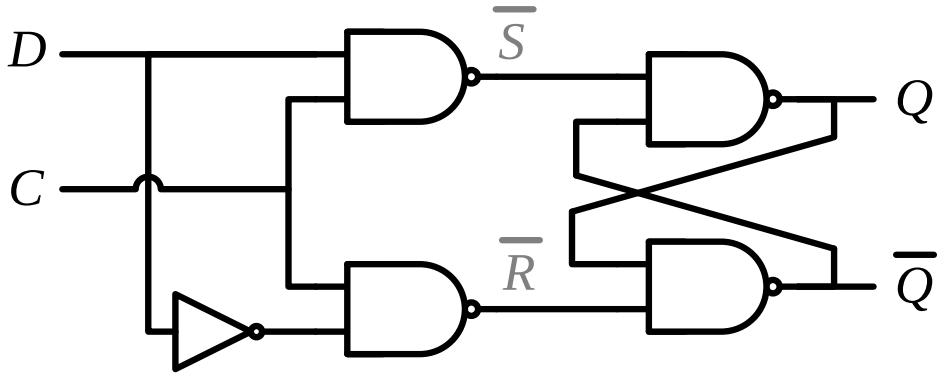
C D	Q	
0 x	Q_0	No change
1 0	0	Reset
1 1	1	Set

Timing Diagram



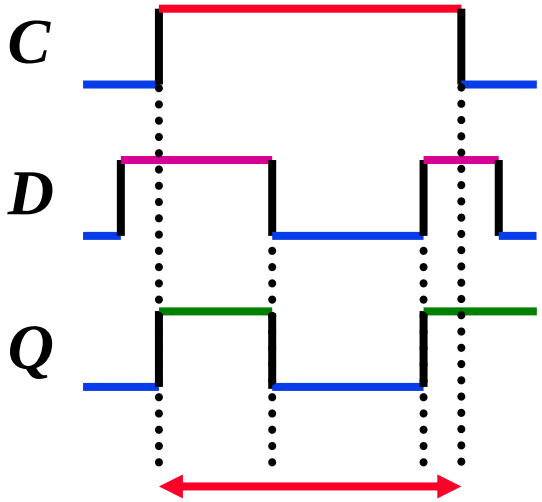
Controlled Latches

★ *D* Latch (*D* = *Data*)



C D	Q	
0 x	Q_0	No change
1 0	0	Reset
1 1	1	Set

Timing Diagram



Output may change

Thanks