Digital Logic Circuits
'Memory Elements'
ELEC2200
Summer 2009

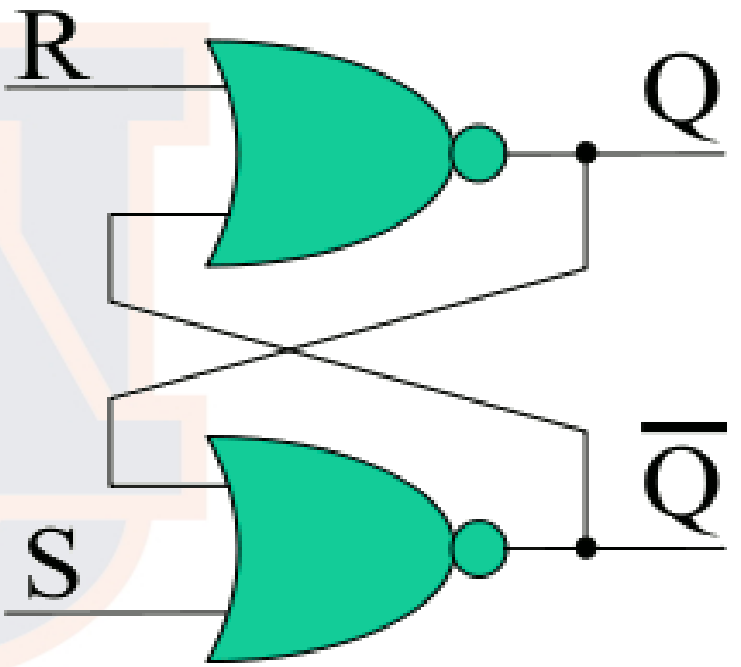
David J. Broderick
brodedj@auburn.edu
http://www.auburn.edu/~brodedj
Office: Broun 360

Memory Elements

- Combinational Logic
    - Output is function of inputs only
- Sequential Logic
    - Output is function of inputs **and** previous outputs
- We need a way to `remember' a previous logical value
- 2 basic types of memory elements
    - Latches – Level-sensitive to inputs
    - Flip-flops – Edge triggered on active edge of clock

Set-Reset Latch (NOR)

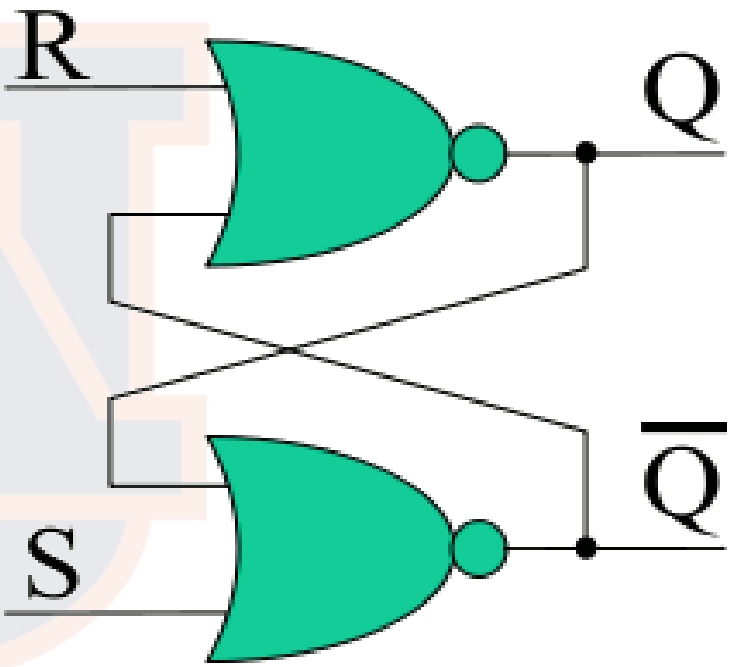
- Simplest Memory Element
- 2 NOR gates with feedback
  - Level sensitive
  - Active High
    - Input must be high to perform command
  - Only one input can be active
- Outputs: current state and its inverse



Set-Reset Latch (NOR)

- Recall the NOR gate:

X	Y	(X+Y)'
0	0	1
0	1	0
1	0	0
1	1	0

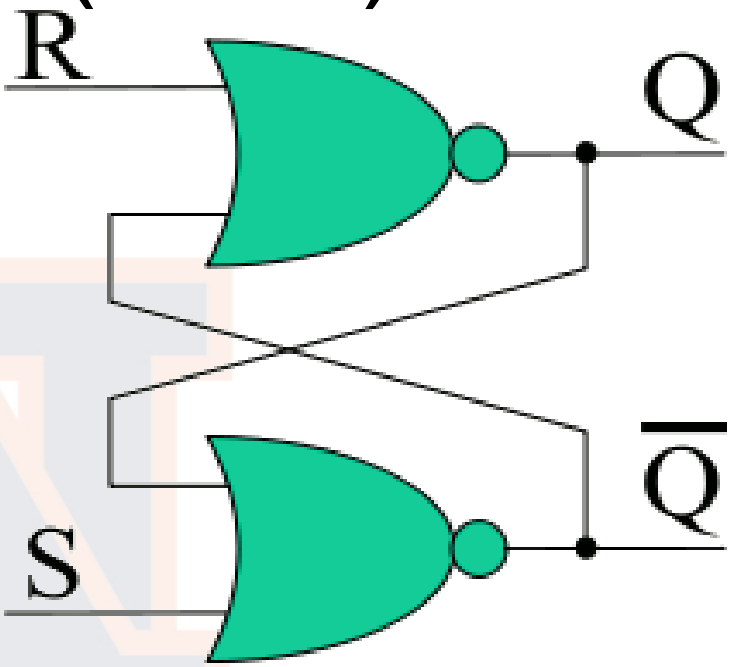


- This leads to:

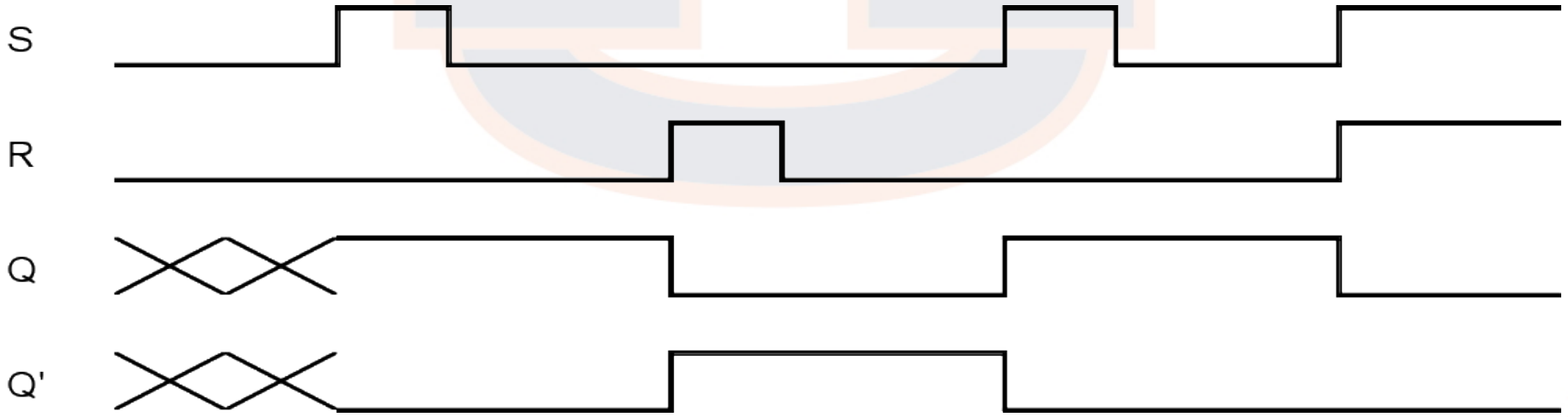
R	S	Q*	Q'	Command
0	0	Q	Q'	Remember
0	1	1	0	Set
1	0	0	1	Reset
1	1	?	?	Invalid

Set-Reset Latch (NOR)

R	S	Q*	Q'	Command
0	0	Q	Q'	Remember
0	1	1	0	Set
1	0	0	1	Reset
1	1	?	?	Invalid

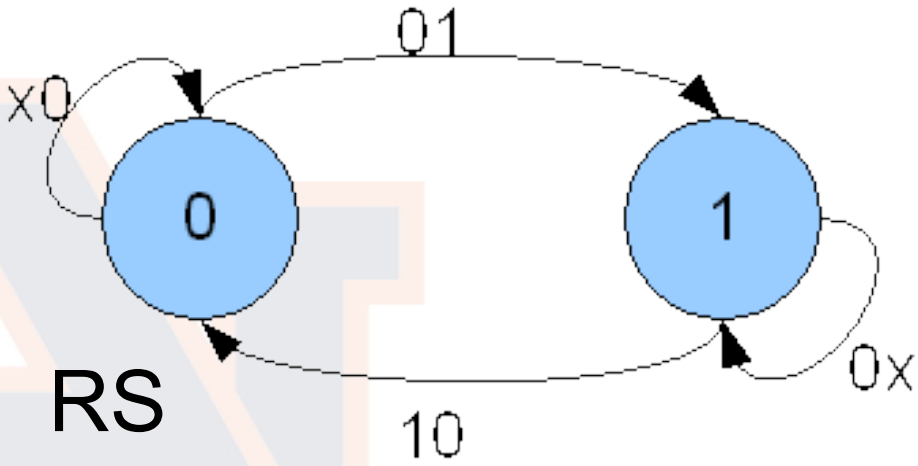


- Timing Diagram



Set-Reset Latch (NOR)

- State Diagram
- In this case:
  - State is output(Q)
  - Each transition has a stimulus, (SR)
- Don't cares still used for some stimuli



R	S	Q*	Q'*	Command
0	0	Q	Q'	Remember
0	1	1	0	Set
1	0	0	1	Reset
1	1	?	?	Invalid

$$Q^* = S + \overline{R}Q$$

Characteristic Equation

- Output of sequential logic is not only dependent on inputs but also the previous states
- Q*=f(S,R,Q)
- For an SR Latch:

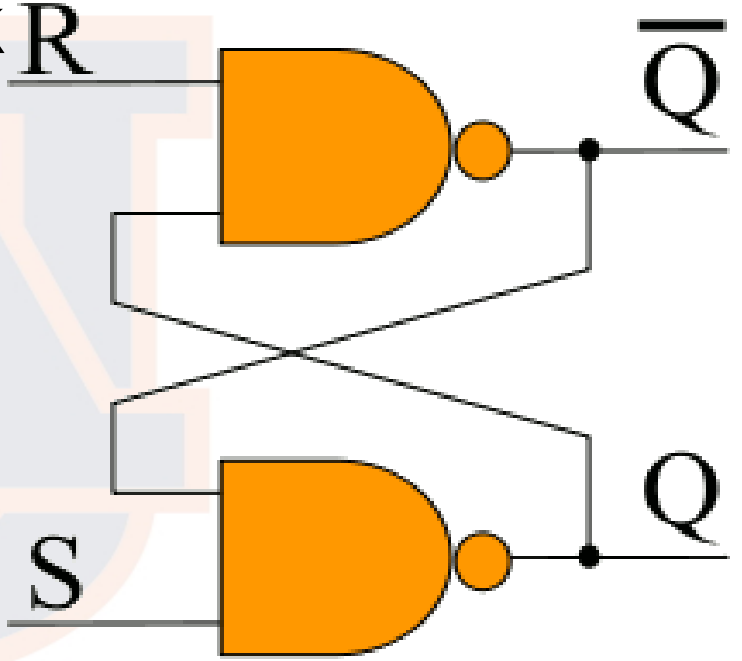
Q\RS	00	01	11	10
0	0	1	x	0
1	1	1	x	0

$$Q^* = S + \overline{R}Q$$

R	S	Q*	Q'*	Command
0	0	Q	Q'	Remember
0	1	1	0	Set
1	0	0	1	Reset
1	1	?	?	Invalid

Set-Reset Latch (NAND)

- Dual circuit
- 2 NAND gates with feedback
  - Level sensitive
  - Active **Low**
    - Input must be **low** to perform command
  - Only one input can be active

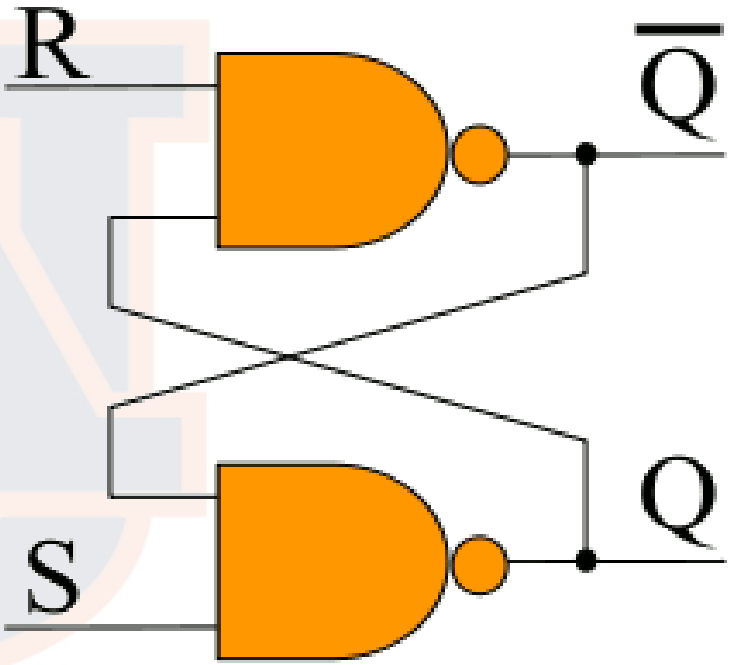


- Outputs: current state and its inverse
- Outputs reversed from NOR SR latch

Set-Reset Latch (NAND)

- Recall the NAND gate:

X	Y	(X·Y)'
0	0	1
0	1	1
1	0	1
1	1	0

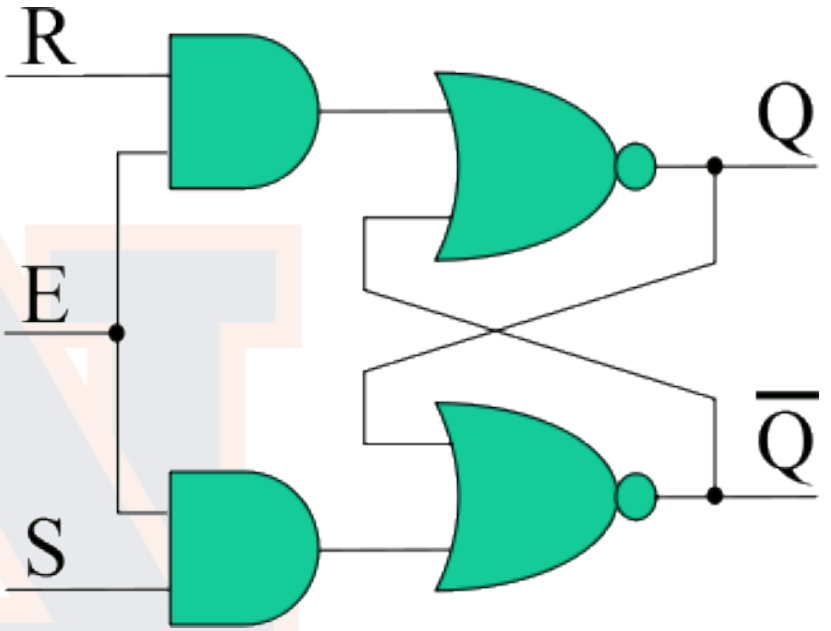


- This leads to:

R	S	Q*	Q'*	Command
0	0	Q	Q'	Invalid
0	1	1	0	Reset
1	0	0	1	Set
1	1	?	?	Storage

Gated SR Latch

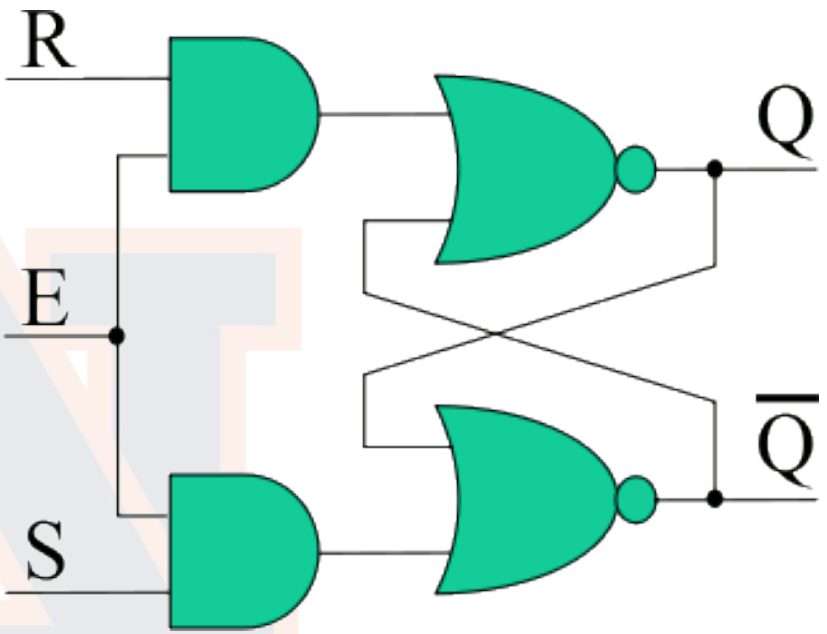
- If new input, E (enable), is inactive (low in this case) latch will be in 'remember' state
- Still level sensitive, so still a latch
- Active high inputs
  - E (enable)
  - R (Reset)
  - S (Set)
- Invalid command still possible
- Dual circuit also possible (NAND)



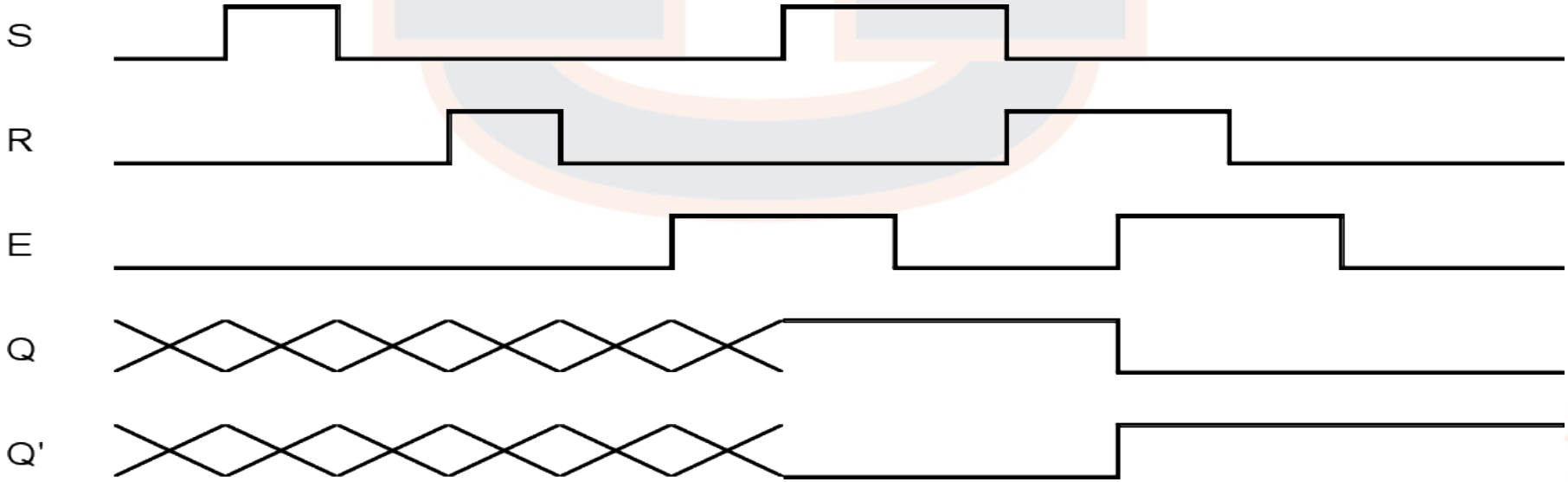
E	R	S	Q*	Q'*	Command
0	x	x	Q	Q'	Remember
1	0	0	Q	Q'	Remember
1	0	1	1	0	Set
1	1	0	0	1	Reset
1	1	1	?	?	Invalid

Gated SR Latch

E	R	S	Q*	Q'*	Command
0	x	x	Q	Q'	Remember
1	0	0	Q	Q'	Remember
1	0	1	1	0	Set
1	1	0	0	1	Reset
1	1	1	?	?	Invalid

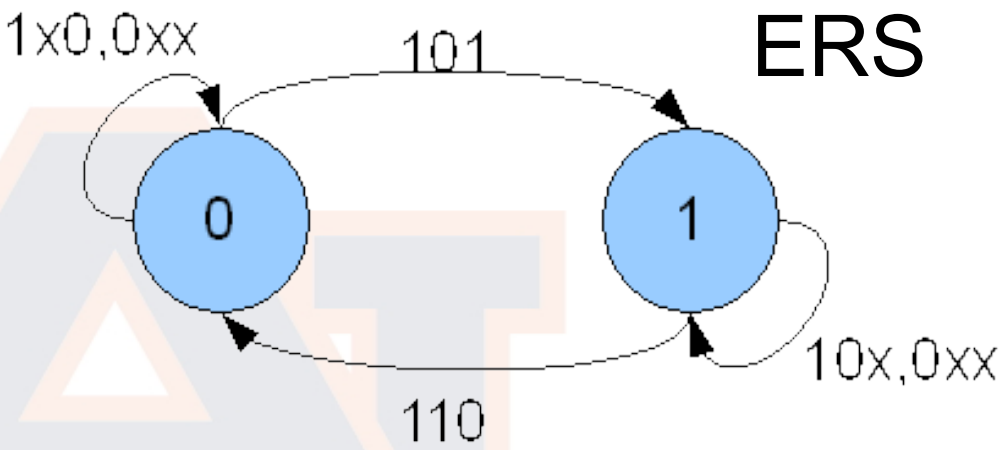


- Timing Diagram



Gated SR Latch

- State Diagram
- Don't cares on inputs can be 0 or 1
- If E=0, other inputs don't matter
- When Q=0, both reset and remember will remain in the same state
- When Q=1, both set and remember will remain in the same state



E	R	S	Q*	Q'*	Command
0	x	x	Q	Q'	Remember
1	0	0	Q	Q'	Remember
1	0	1	1	0	Set
1	1	0	0	1	Reset
1	1	1	?	?	Invalid

Characteristic Equation

- Gated SR Latch

QE\RS	00	01	11	10
00	0	0	0	0
01	0	1	x	0
11	1	1	x	0
10	1	1	1	1

E	R	S	Q*	Q'*	Command
0	x	x	Q	Q'	Remember
1	0	0	Q	Q'	Remember
1	0	1	1	0	Set
1	1	0	0	1	Reset
1	1	1	?	?	Invalid

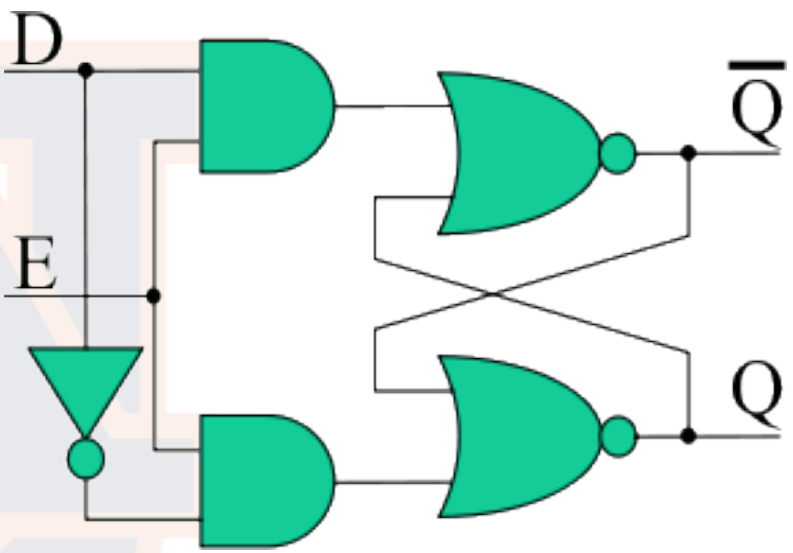
$$Q^* = SE + Q\overline{R} + \overline{E}Q$$

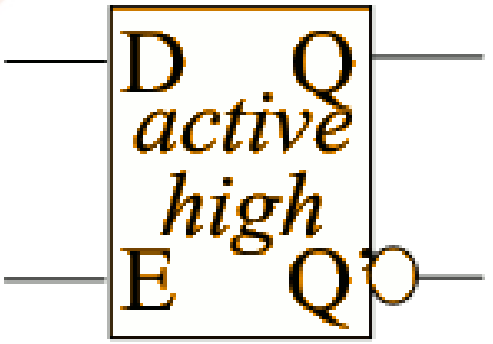
Enabled Data Latch

- AKA: Delay Latch, D Latch, transparent D Latch
- No more undefined state!

E	D	Q*	Command
0	x	Q	Remember
1	0	0	Transparent
1	1	1	Transparent

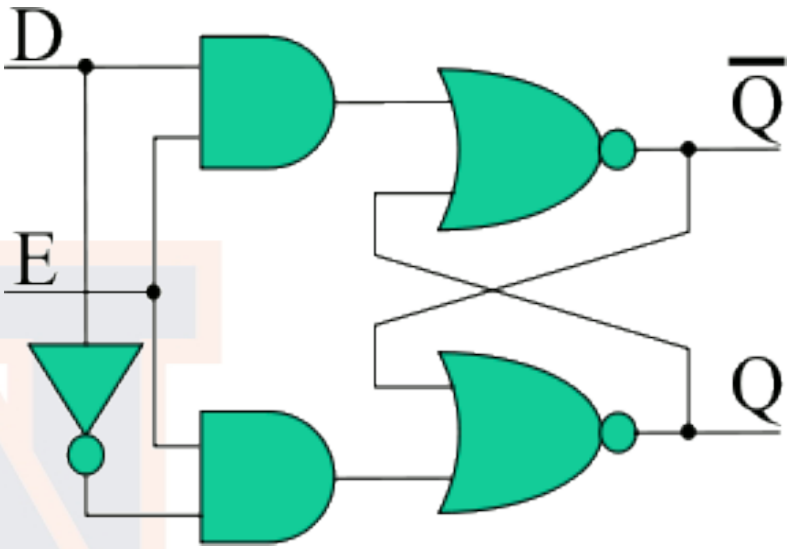
- When enabled (E=1) the latch is transparent (Q=D)



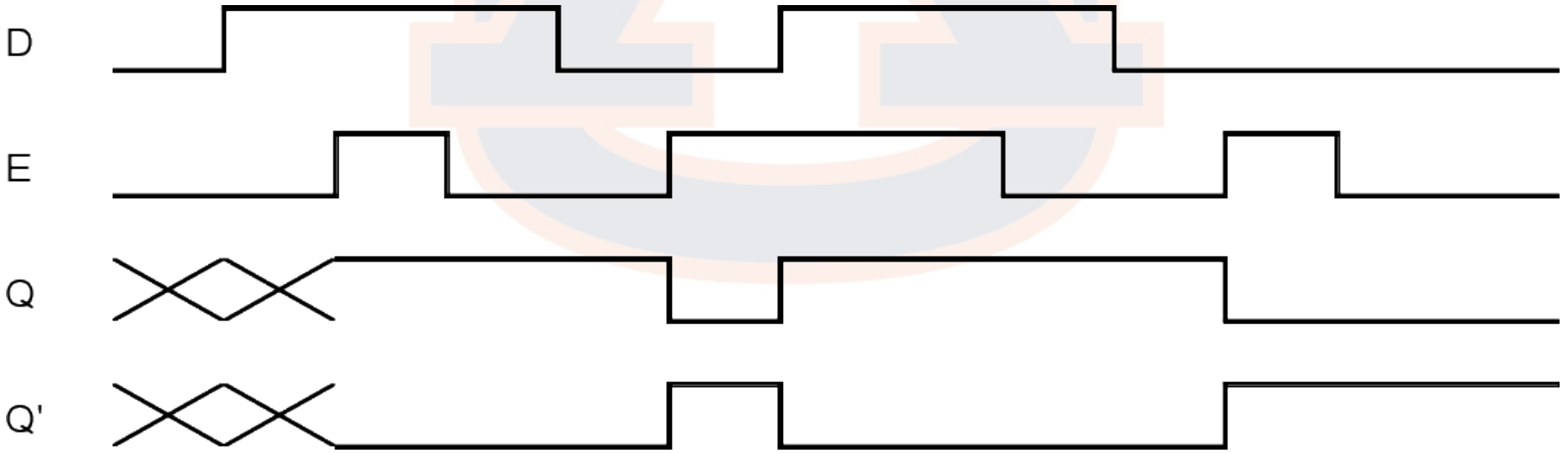


Enabled Data Latch

E	D	Q*	Command
0	x	Q	Remember
1	0	0	Transparent
1	1	1	Transparent

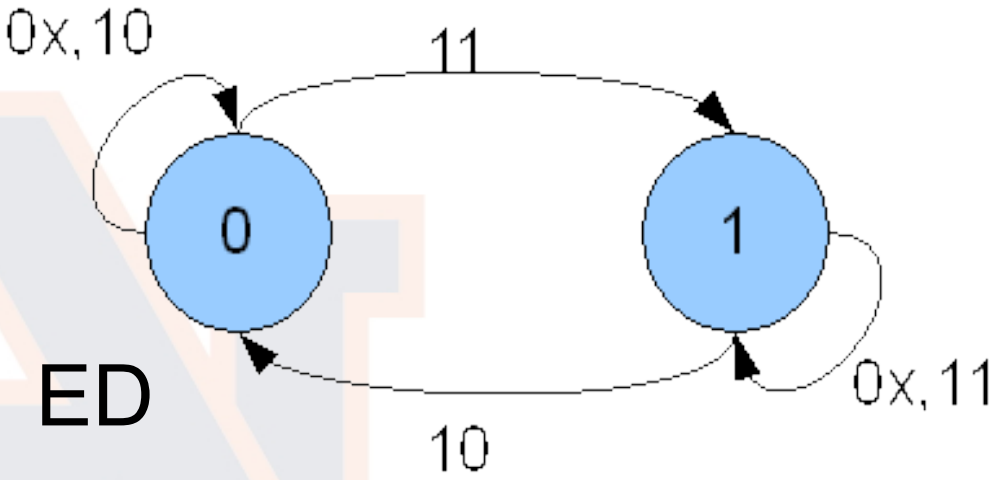


Timing Diagram



Enabled Data Latch

- D Latch State Diagram
  - Still only two states,
  - Stimulus changed
- When Q=0, both reset and remember will remain in the same state
- When Q=1, both set and remember will remain in the same state



E	D	Q*	Command
0	x	Q	Remember
1	0	0	Transparent
1	1	1	Transparent

Characteristic Equation

- Enabled D Latch

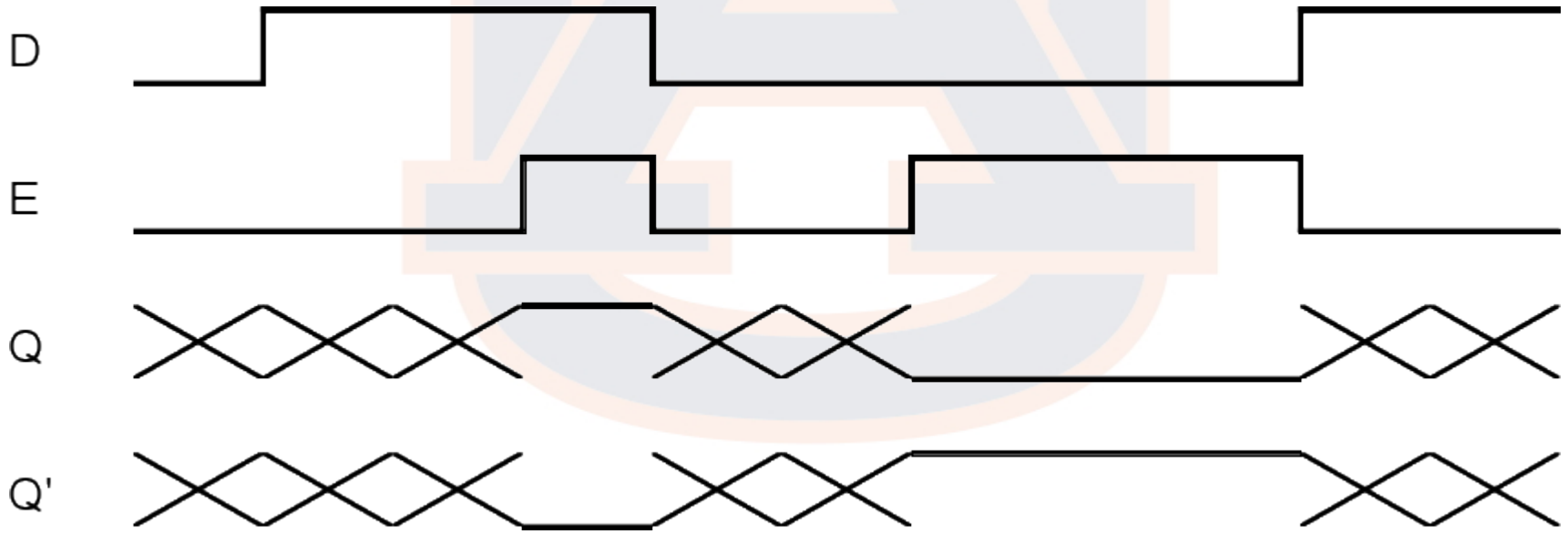
Q\ED	00	01	11	10
0	0	0	1	0
1	1	1	1	0

E	D	Q*	Command
0	x	Q	Remember
1	0	0	Transparent
1	1	1	Transparent

$$Q^* = DE + \overline{E}Q$$

Something to avoid

- Transitions on D and E at the same time
- Inputs don't change instantaneously, can lead to unknown state

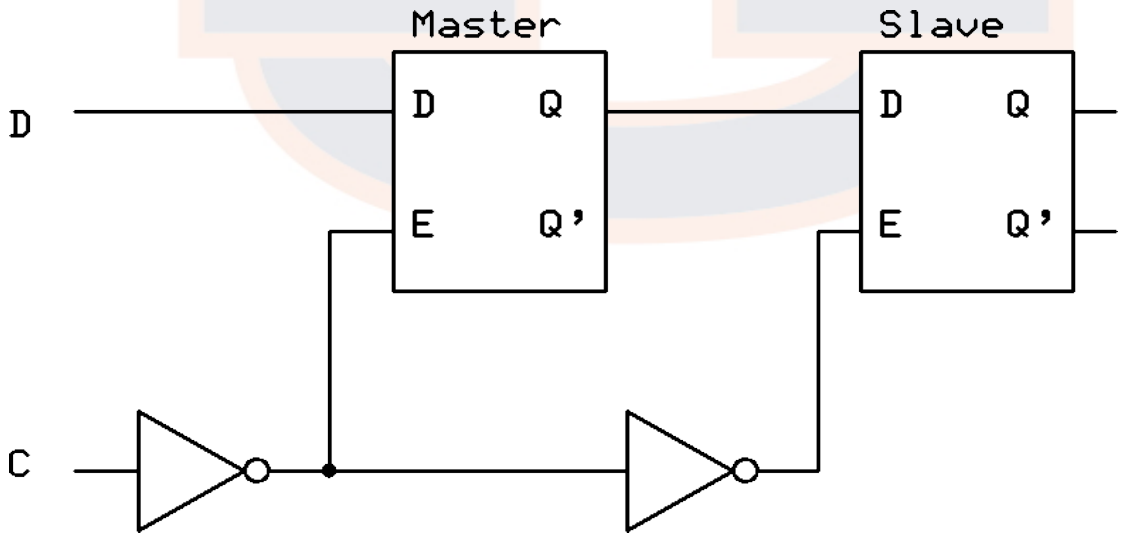


Flip-Flops

- Latches are level sensitive, transition when a level (0 or 1) is present on the inputs
- Flip-flops transition on a clock signal
    - Positive pulse
    - Negative pulse
    - Positive edge
    - Negative edge
- Can be built from basic gates or from latches

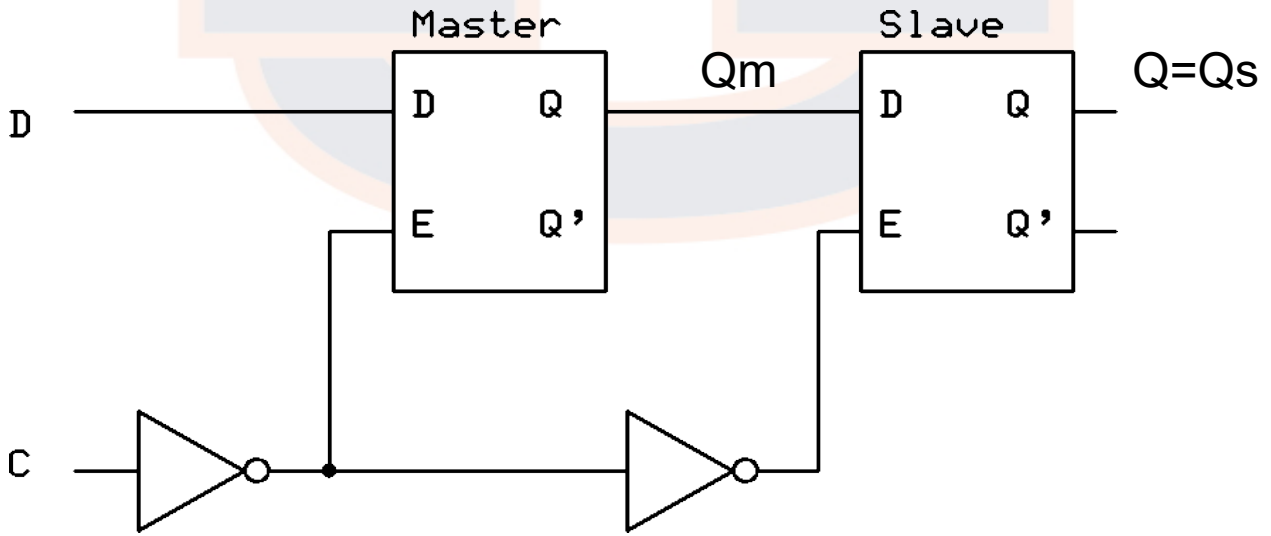
Master-Slave Flip-Flops

- When clock is low
    - Master D latch is transparent
    - Slave D latch is remembering
- When clock is high
    - Master D latch is remembering
    - Slave D latch is transparent

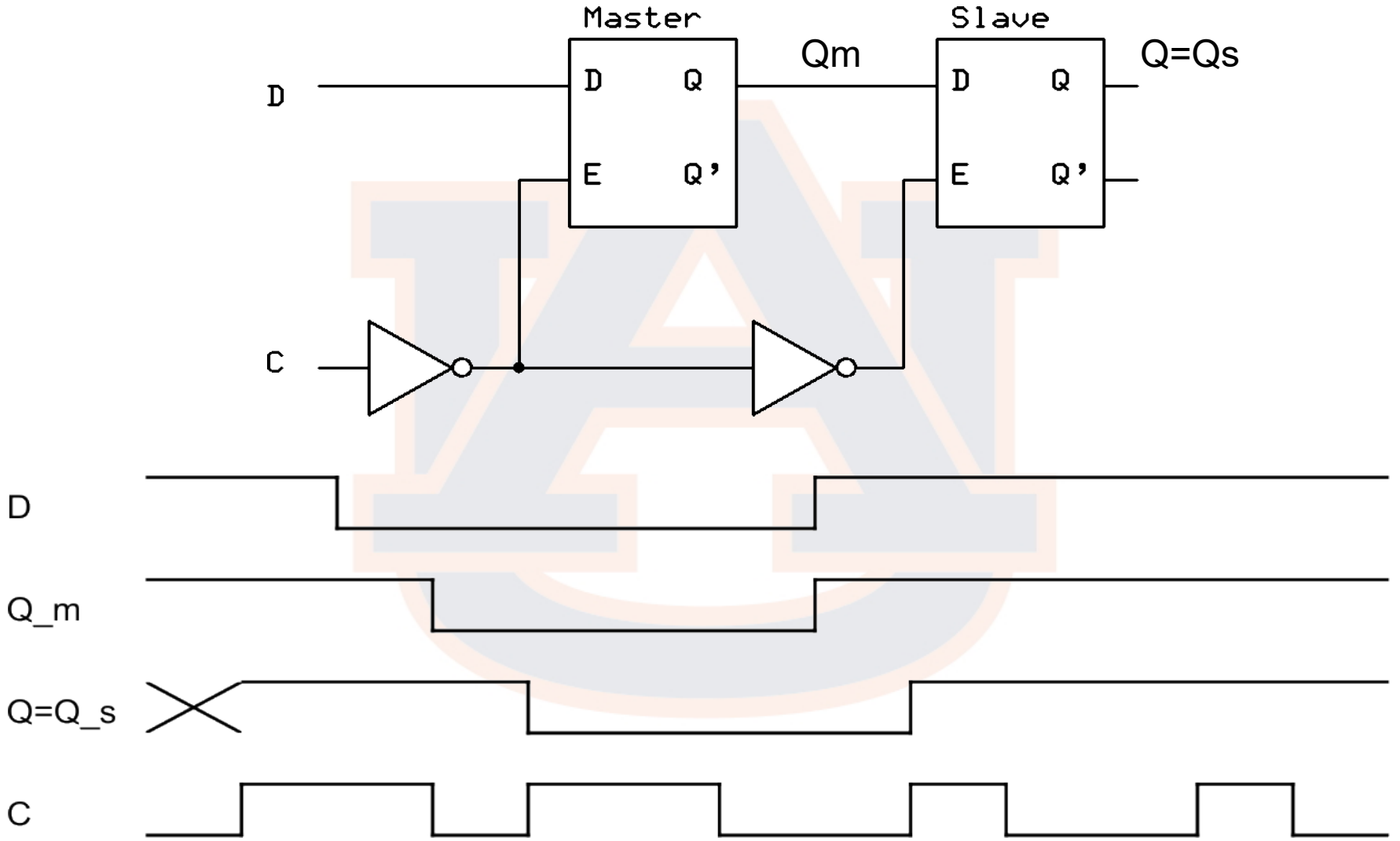


Master-Slave Flip-Flops

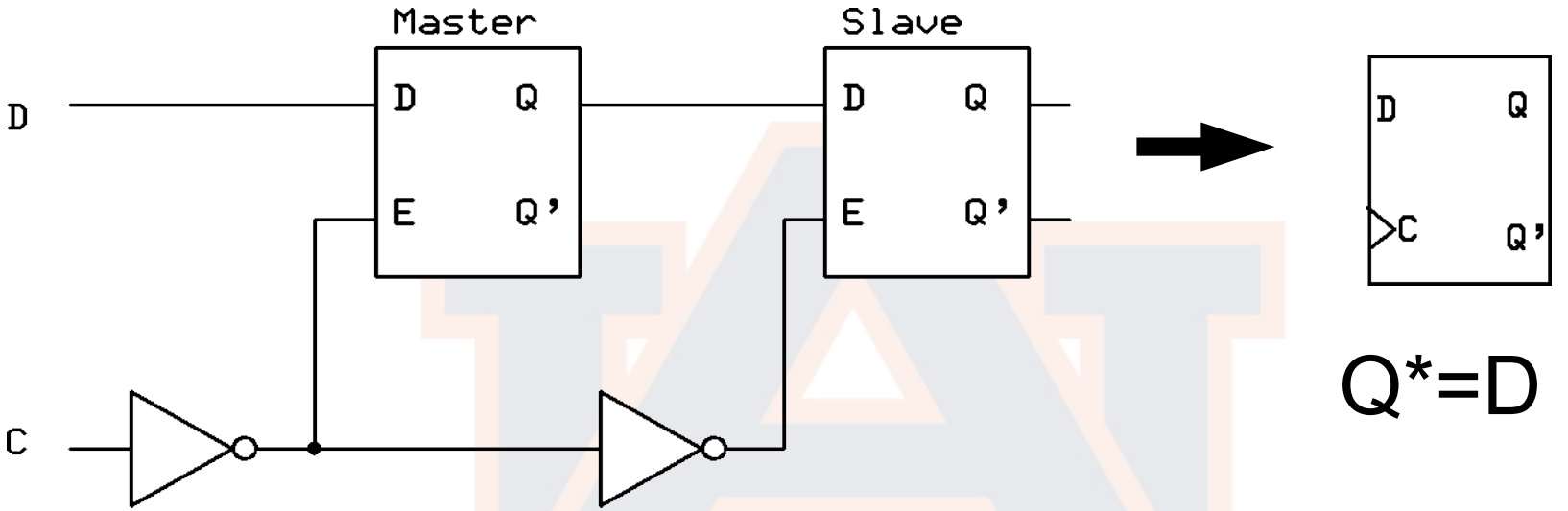
- Qm can transition whenever C is low
- Q (Qs) transitions on the positive edge of a pulse
- Whatever value was present on D when the positive edge of the clock (C) occurred is allowed to pass to Q at that time
- Ideally all transitions occur simultaneously



Master-Slave Flip-Flops

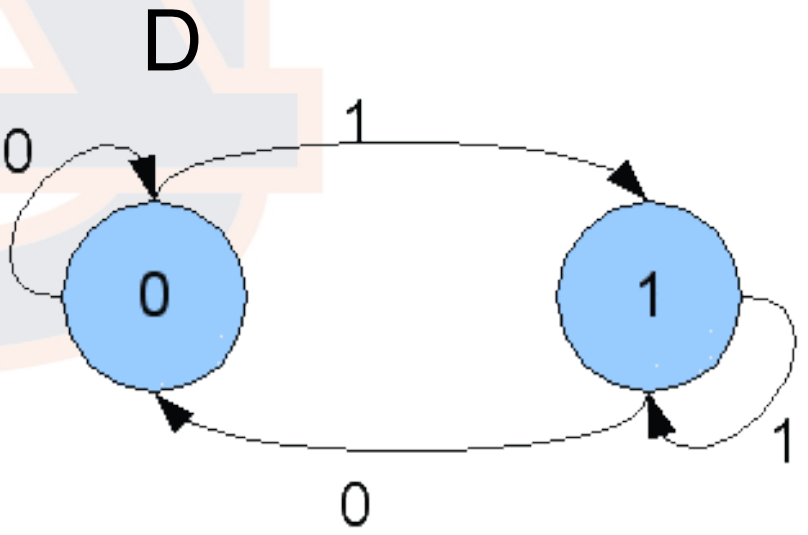


Master-Slave Flip-Flops



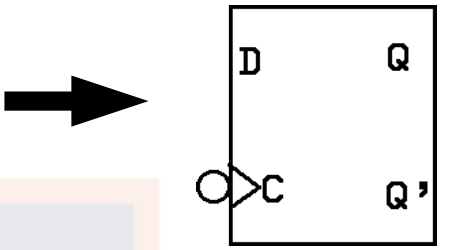
Q*=D

D	C	Q*	Command
0	↑	0	Reset
1	↑	1	Set

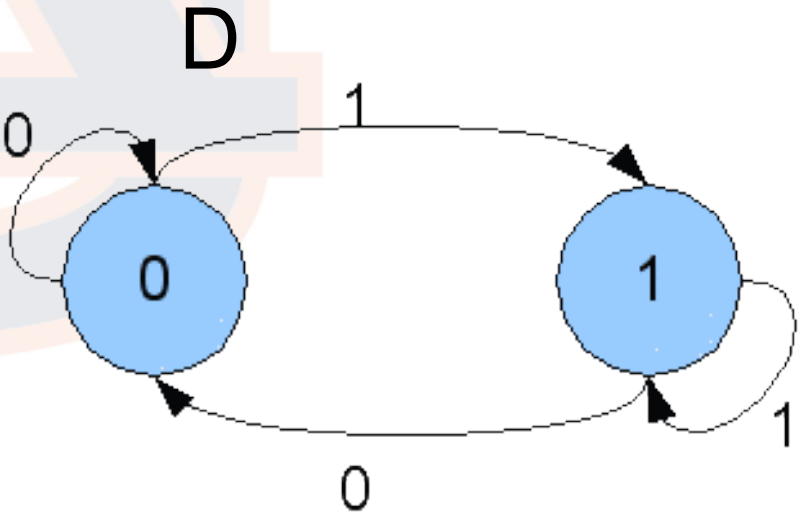


Master-Slave Flip-Flops

- Negative edge of clock can be active too
- Q transitions on negative going edge of clock
- Negative edge is now the active edge



D	C	Q*	Command
0	↓	0	Reset
1	↓	1	Set



Characteristic Equation

- Master-Slave D Flip-flop

Q\D	0	1
0	0	1
1	0	1

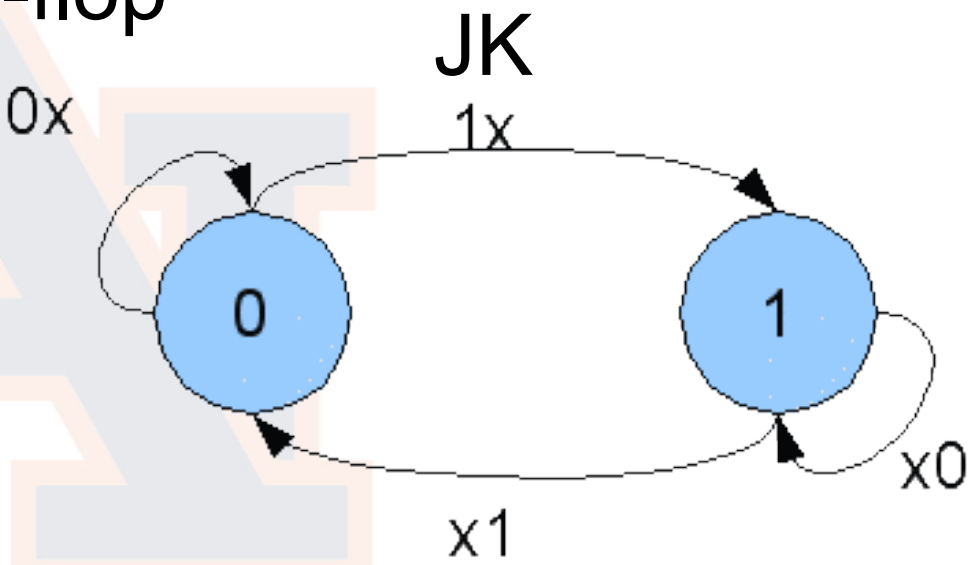
Q*=D

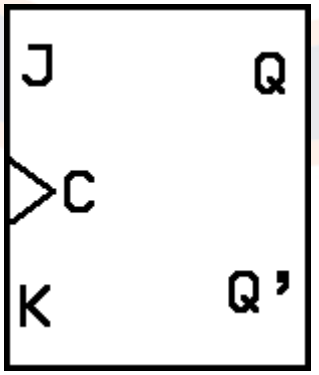
D	C	Q*	Command
0	↓	0	Reset
1	↓	1	Set

J-K Flip-Flop

- Another method of removing the invalid input present in a SR latch/flip-flop

J	K	C	Q*	Command
0	0	↑	Q	Remember
0	1	↑	0	Reset
1	0	↑	1	Set
1	1	↑	Q'	Toggle





Characteristic Equation

- JK Flip-flop

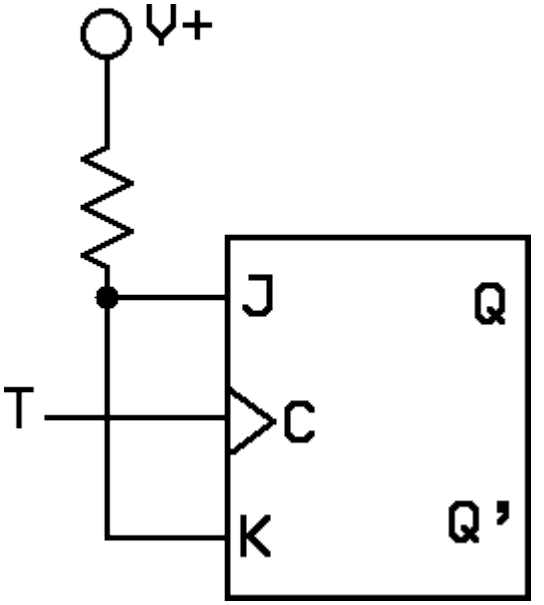
Q\JK	00	01	11	10
0	0	0	1	1
1	1	0	0	1

J	K	C	Q*	Command
0	0	↑	Q	Remember
0	1	↑	0	Reset
1	0	↑	1	Set
1	1	↑	Q'	Toggle

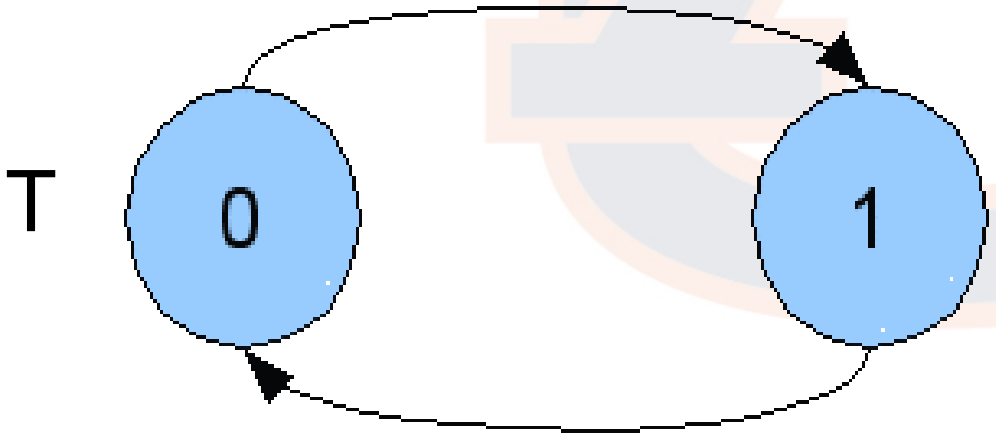
$$Q^* = \overline{K}Q + J\overline{Q}$$

Toggle (T) Flip-Flop

- Built from J-K Flip-flops
- Edge-triggered (1st type)



T	Q*	Command
↓	Q'	Toggle



Characteristic Equation

- T Flip-flop (edge triggered)

Q	
0	1
1	0

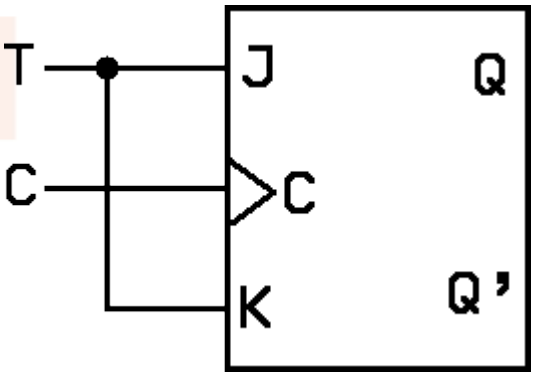
T	Q*	Command
↓	Q'	Toggle

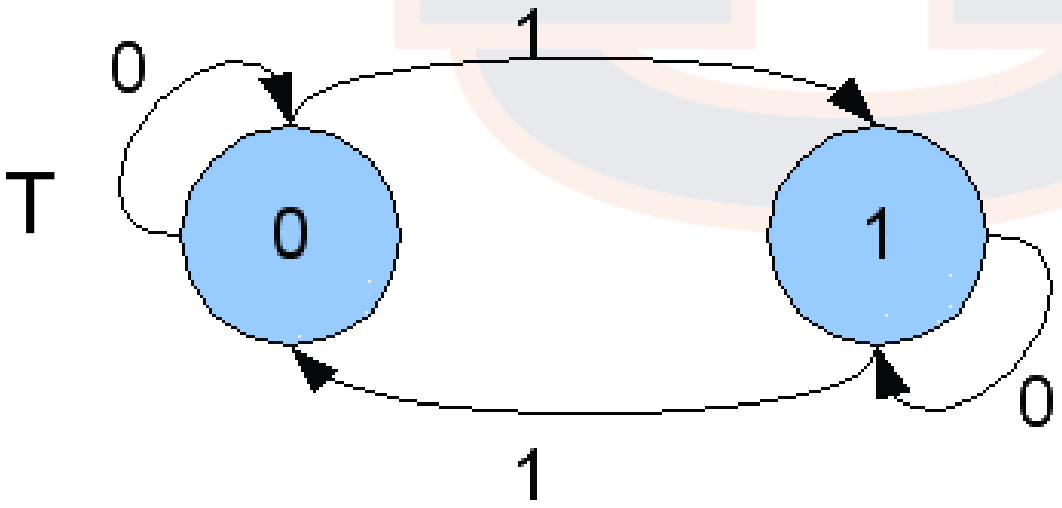
$$Q^* = \overline{Q}$$

Toggle (T) Flip-Flop

- Built from J-K Flip-flops
- Clocked (2nd type)

T	C	Q*	Command
0	↓	Q	Remember
1	↓	Q	Toggle





Characteristic Equation

- T Flip-flop (clocked)

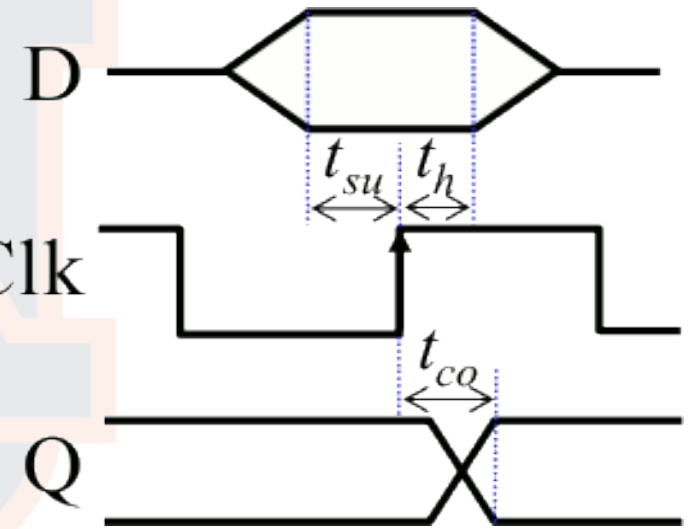
Q\T	0	1
0	0	**1**
1	**1**	0

T	C	Q*	Command
0	↓	Q	Remember
1	↓	Q	Toggle

$$Q^* = \overline{T}Q + T\overline{Q}$$

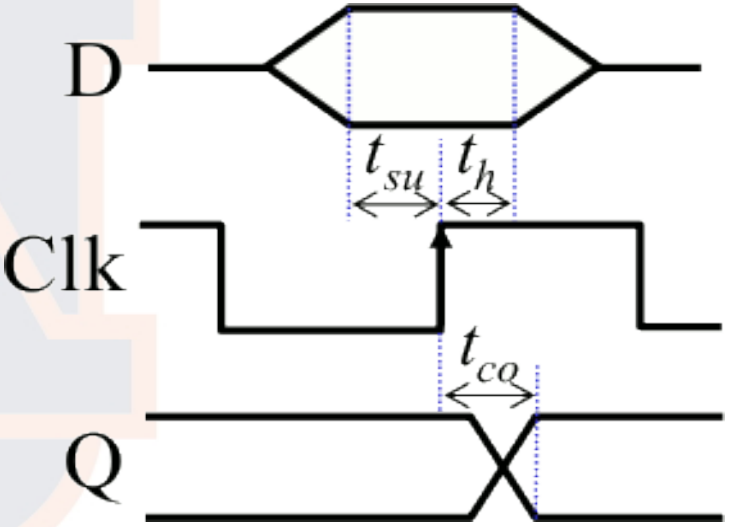
Timing Considerations

- Transitions don't actually happen simultaneously
- Minimum setup time (t_{su}) – minimum time D must be valid at input before active edge of clock
- Minimum hold time (t_h) – minimum time D must remain valid after active edge of clock
- Clock-to-output delay (t_{co}) – maximum time before Q is valid after the active edge of clock



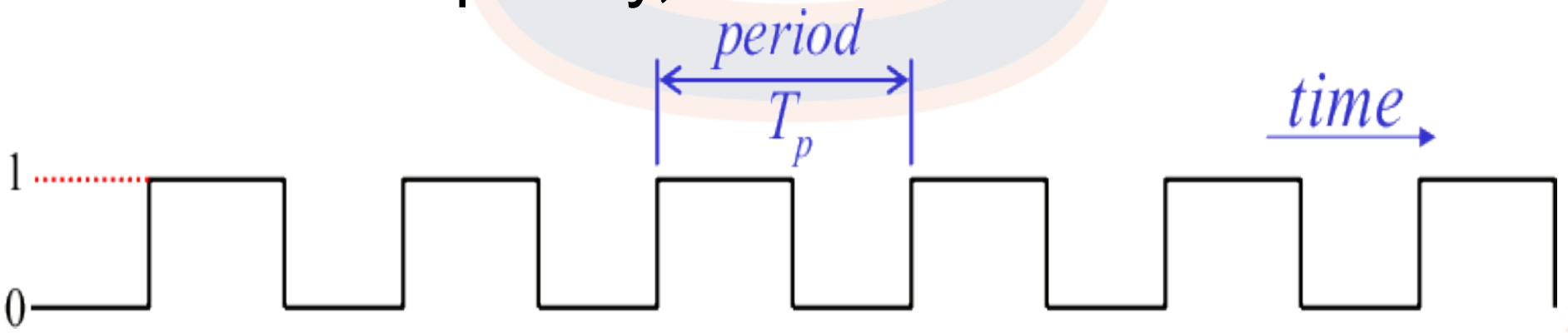
Timing Considerations

- Violations of these times must be avoided in design process
- If violation occurs:
  - Real circuit will resolve to an unknown state (Q=Q')
  - Simulation will not be valid and can lead to invalid simulation and manufacturing problems (clock-data-races)



Clock Signals

- Typically a periodic signal (a sequence of pulses) used to:
    - Sample data
    - Store sampled data in memory elements
- T= period of periodic signal, the time it take the signal to repeat itself
- f= clock frequency, 1/T



Flip-Flop Initialization

- Some flip-flops have another set of inputs for initialization
- These input can be
    - Synchronous – Change output on active clock edge
    - Asynchronous – Change inputs regardless of active clock edge
- Initialization important:
    - To remove undefined logic value in simulation
    - Start a system in a known state

Flip-Flop Initialization

- Initialization inputs called:
    - Preset (pre) sets output (Q) to logic 1 (high)
    - Clear (clr) sets output (Q) to logic 0 (low)
- Initialization inputs can also be:
    - Active high – affect output when logic 1
    - Active low - affect output when logic 0

Flip-Flop Initialization

- For this example:
  - Preset (Pre) is active high and synchronous
  - Clear (Clr) is active low and asynchronous