CHAPTER 7: SEQUENTIAL CIRCUITS – FLIP-FLOPS, REGISTERS, AND COUNTERS

What will we learn?

- Logic circuits that can store information
  - Latches, which store *a single bit*
  - Flip-Flops, which store *a single bit*
  - Registers, which store *multiple bits*
- Shift registers
- Counters
- Design Examples

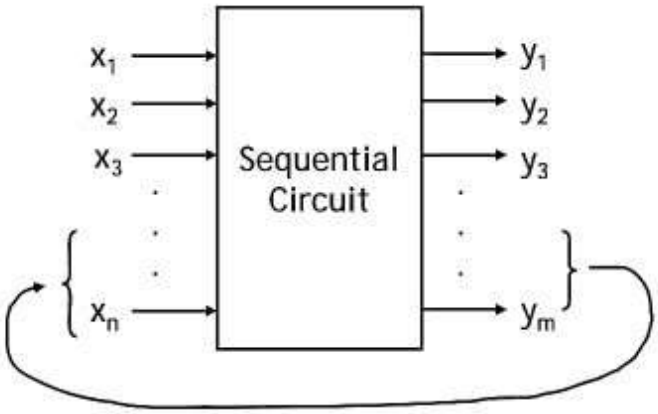
Sequential Circuits

- Combinational Circuits
  - circuits without feedback
  - output = f (current inputs)

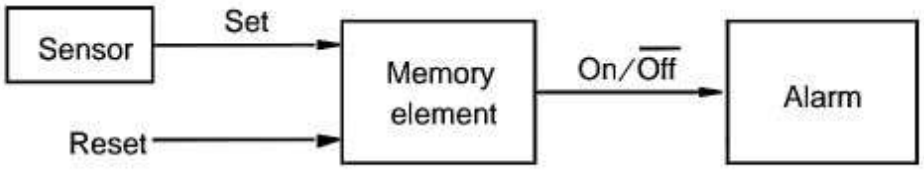
- Sequential Circuits
  - circuits with feedback
  - output = f (current inputs, past inputs, past outputs)
  - how can we feed the past inputs and outputs into the circuits?
    - basis for building "memory" into logic circuits

Circuits with feedback

- How to control feedback?
  - what stops values from cycling around endlessly



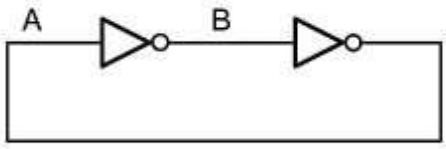
Control of an alarm system



- the simplest case of a sequential circuit
  - Alarm is on when the sensor generates the "Set" signal in response to some undesirable events
  - Once the alarm is on, it can only be turned off manually through a reset button
- Memory is needed to remember that the alarm has to be active until the reset signal arrives

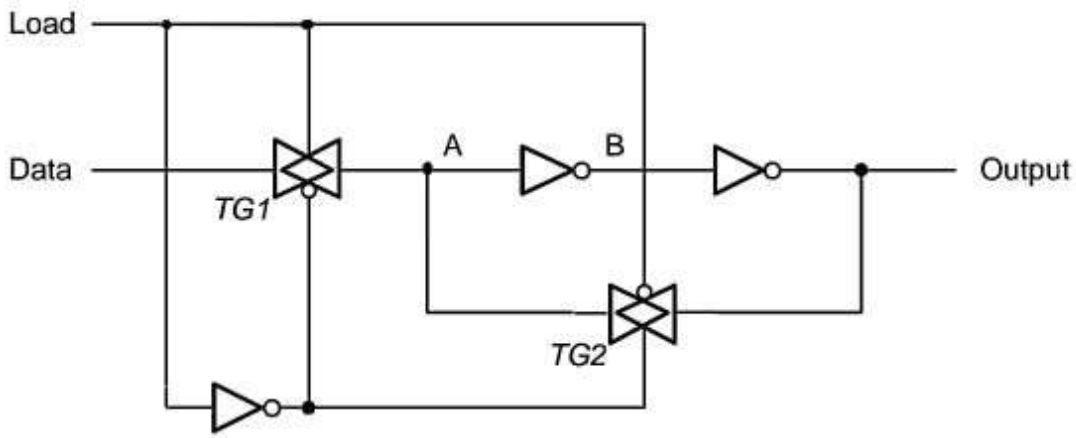
A simple memory element

- The most rudimentary memory element
  - Two inverters form a static memory cell
    - Assume A=0 and B=1, then the below circuit will maintain these values indefinitely (as long as it has power applied)
    - The state is defined by the value of the memory cell
      - Two states



- How to get a new value into the memory cell?
  - selectively break feedback path
  - load new value into cell

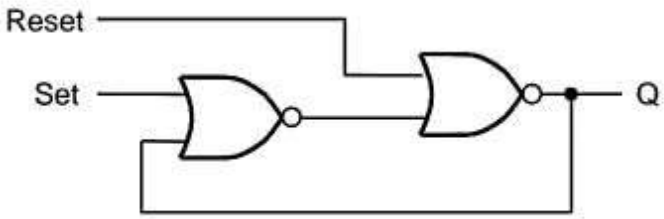
A controlled memory element



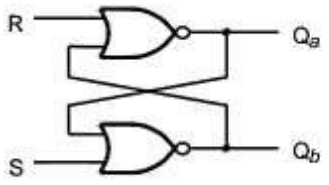
A memory element with NOR gates

- Construct a memory cell using ordinary logic gates
  - Two NOR gates are connected in cross-coupled style
  - Basic Latch

- Two inputs
  - Set
  - Reset



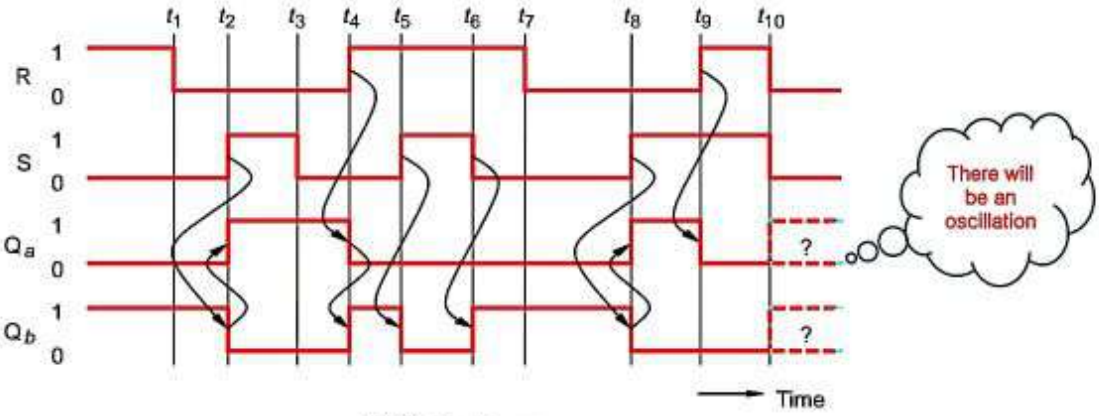
A basic latch built with NOR gates



(a) Circuit

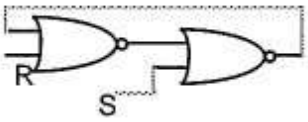
S	R	Q_a	Q_b	
0	0	0/1	1/0	(no change)
0	1	0	1	
1	0	1	0	
1	1	0	0	

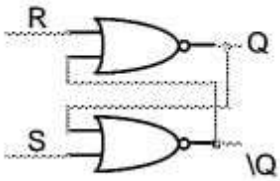
(b) Truth table or *characteristic table*



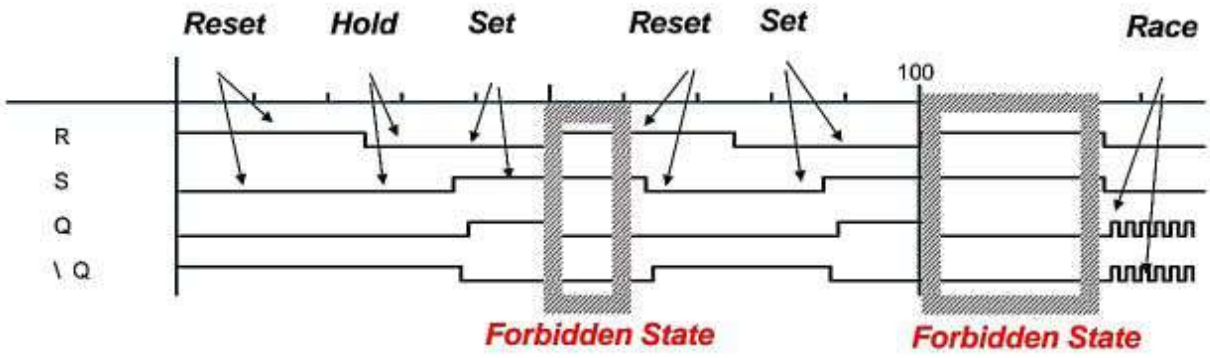
(c) Timing diagram

Timing Waveform





Timing Waveform



State Behavior of R–S Latch

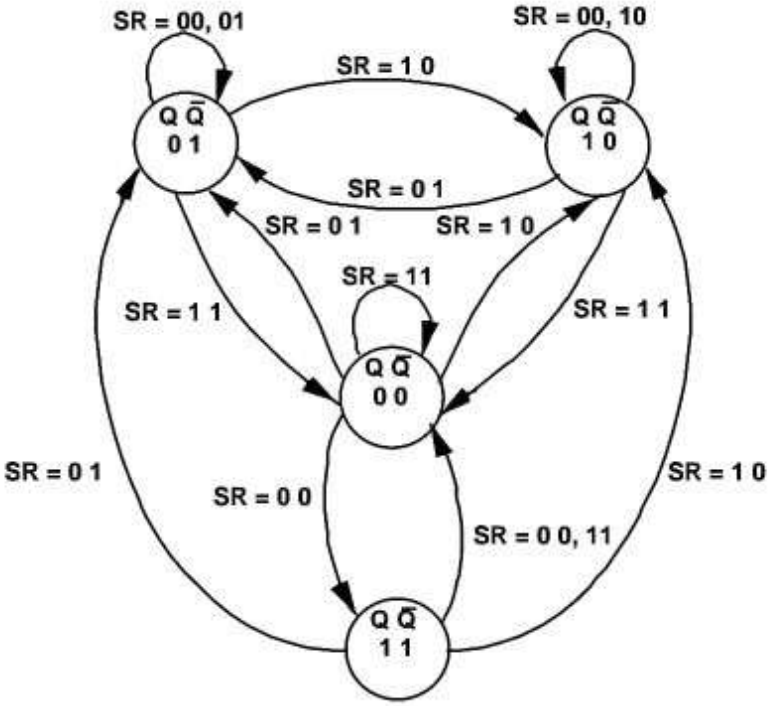
S	R	Q
0	0	hold
0	1	0
1	0	1
1	1	unstable

Truth Table Summary of R-S Latch Behavior

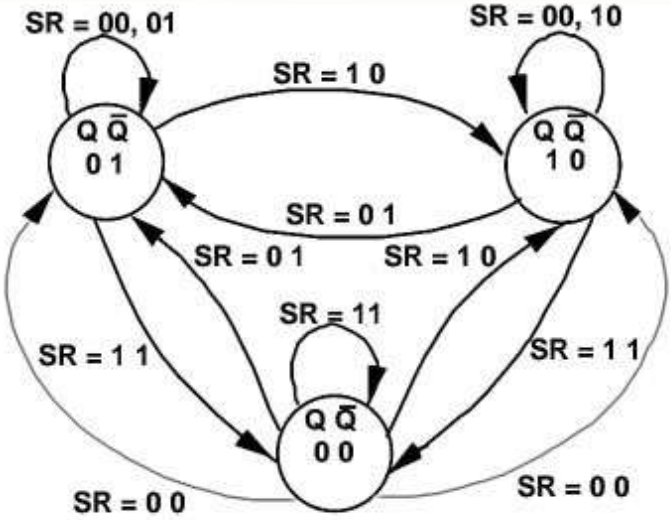
Theoretical R-S Latch State Diagram

12

- State Diagram
  - state: possible values
  - transitions: changes based on inputs



Observed R–S Latch Behavior



Very difficult to observe R-S Latch in the 1-1 state

Ambiguously returns to state 0-1 or 1-0

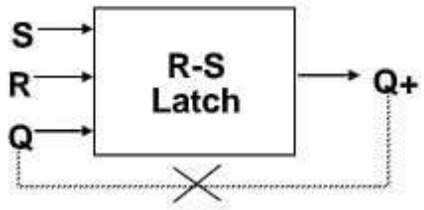
A so-called "race condition"

R–S Latch Analysis

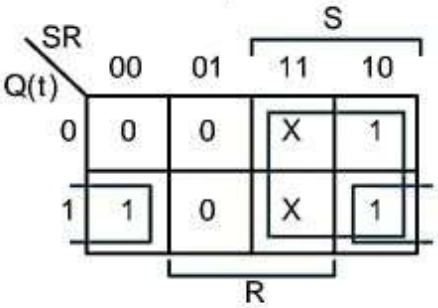
Truth Table:
Next State = F(S, R, Current State)

R-S Latch Revisited

S	R	Q_t	Q+	
0	0	0	0	hold
0	0	1	1	
0	1	0	0	reset 0
0	1	1	0	
1	0	0	1	set 1
1	0	1	1	
1	1	0	x	not allowed
1	1	1	x	



Derived K-Map:



Characteristic Equation:

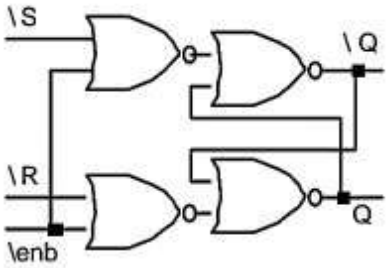
$$Q+ = S + \overline{R}\, Q_t$$

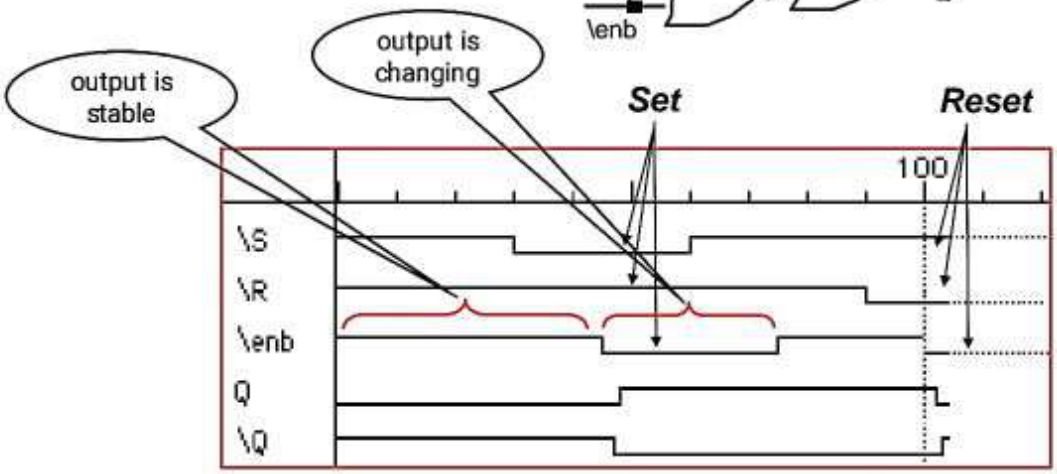
Problems of R-S Latch

- *The slightest glitch on R or S* could cause change in value stored
  - R-S Latch has transparent outputs
    - Transparent outputs : when the memory element's outputs immediately change in response to input changes
- Want to control *when R and S inputs have effect on value stored*
  - Enable Signal (or clock signal)
    - R and S inputs are active *only when Enable = 1*
  - Gated Latches or Level sensitive latches

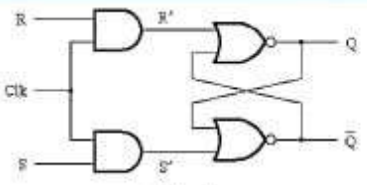
Gated SR latch

- *Control when R and S inputs matters*
  - the latch can be modified to respond to the input signal S and R only when Enable =1





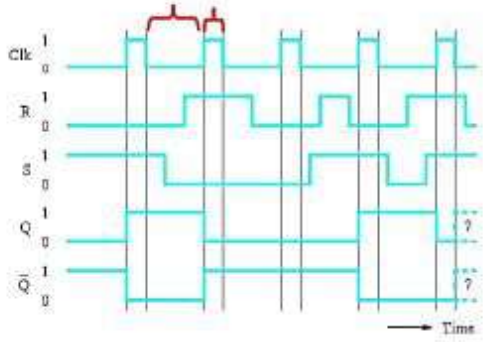
Gated SR Latch



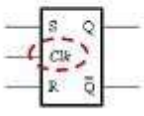
(a) Circuit

Clk	S	R	Q(t + 1)
0	x	x	Q(t) (no change)
1	0	0	Q(t) (no change)
1	0	1	0
1	1	0	1
1	1	1	x

(b) Characteristic table

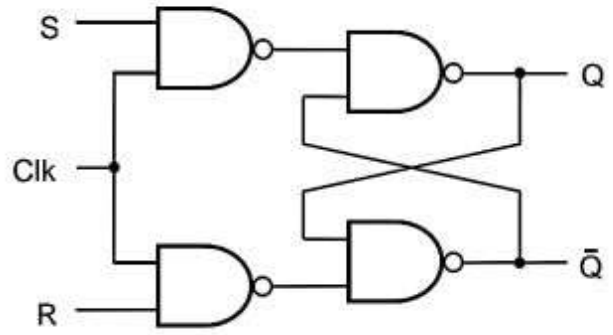


(c) Timing diagram



(d) Graphical symbol

Gated SR latch with NAND gates

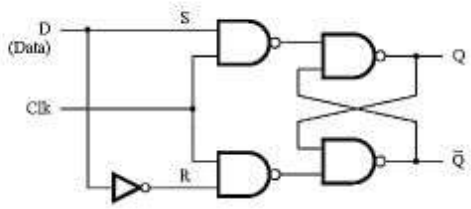


Problems of the Gated S/R Latches

- 1. Forbidden State and Race condition
  - How to eliminate the forbidden state and race condition
  - When S=R=1, $Q=\overline{Q}=0$ (forbidden state)
  - Oscillation (Race condition)
    - D-type Latch
    - JK-Latch (toggling)
      - The output toggles forever when J=K=1
- 2. When cascading level-sensitive Latches
  - Master/Slave F/F's
  - Edge-triggered F/F's

1. How to eliminate the forbidden state?

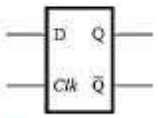
- Gated D-latch
  - eliminate the troublesome situation where *S=R=1*



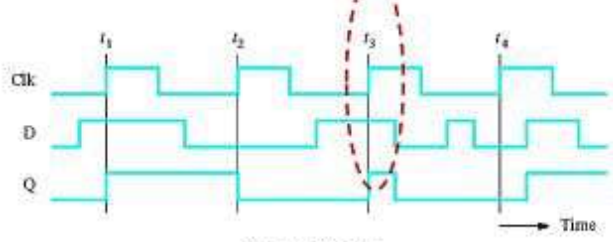
(a) Circuit

Clk	D	Q(t + 1)
0	x	Q(t)
1	0	0
1	1	1

(b) Characteristic table



(c) Graphical symbol

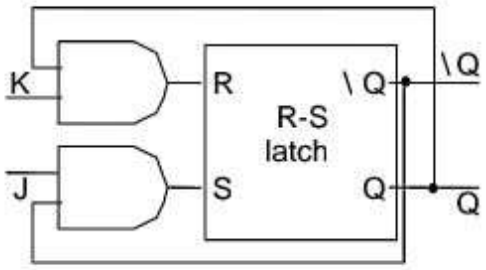


(d) Timing diagram

How to eliminate the forbidden state? cont'd

Idea: use output feedback to guarantee that R and S are never both one

J, K both one yields toggle



J-K Latch

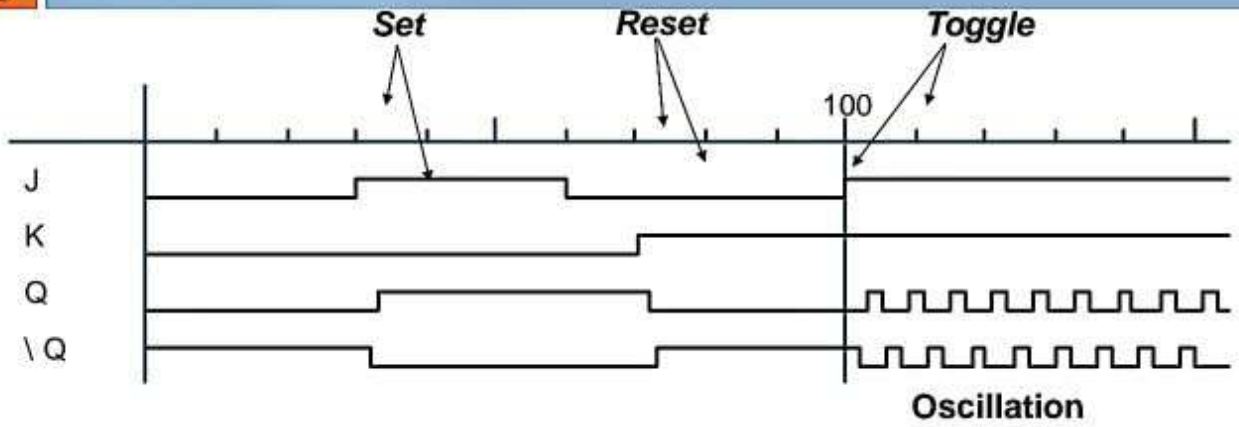
J(t)	K(t)	Q(t)	Q+		
0	0	0	0	hold	
0	0	1	1		
0	1	0	0	reset	0
0	1	1	0		
1	0	0	1	set	1
1	0	1	1		
1	1	0	1	toggle	
1	1	1	0		

Characteristic Equation:

$$Q+ = Q\,\overline{K} + \overline{Q}\,J$$

J-K Latch: Toggles forever in the toggle mode

22



Toggle Correctness: Single State change per clocking event

Solution: Master/Slave Flipflop