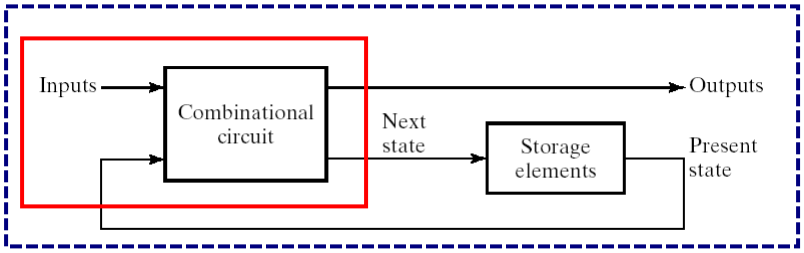
Computer Architecture
(CSC-3501)
Lecture 7
(07 Feb 2008)

Seung-Jong Park (Jay)
http://www.csc.lsu.edu/~sjpark

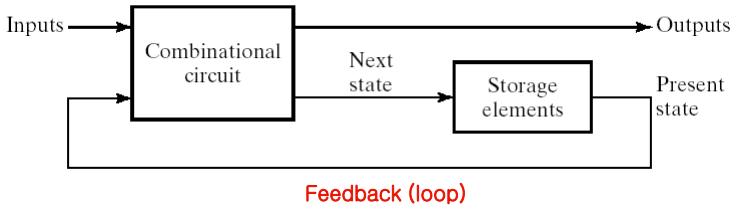
Announcement

Combinational vs. Sequential Logic

- Combinational Logic
  - Memoryless
  - Outputs determined by current values of inputs
- Sequential Logic
  - Has memory
  - Outputs determined by previous and current values of inputs



Sequential Logic



Feedback (loop)

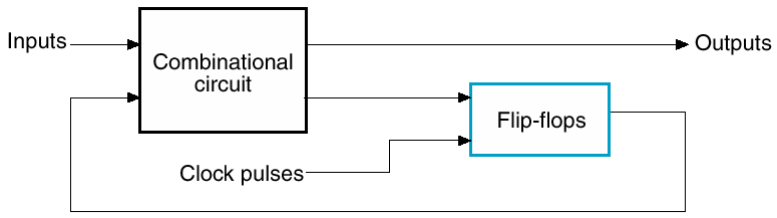
- *State* of system is stored
- States and inputs determine outputs

Sequential Circuits

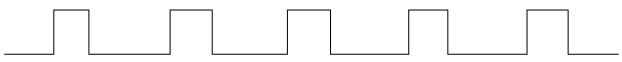
- Types
  - Synchronous: State changes synchronized by one or more clocks
  - Asynchronous: Changes occur independently
- Kinds
  - S-R latch
  - D latch
  - SR Flip-flop
  - D Flip-flop
- Combinational Logics consist of
  - Gates
    - Can be built from Transistors
- Sequential Logics consist of
  - Latch
    - Can be built from Gates
  - Flip-flop
    - Can be built from latch

Clocking of Synchronous

- Changes enabled by clock



(a) Block diagram



(b) Timing diagram of clock pulses

Comparison

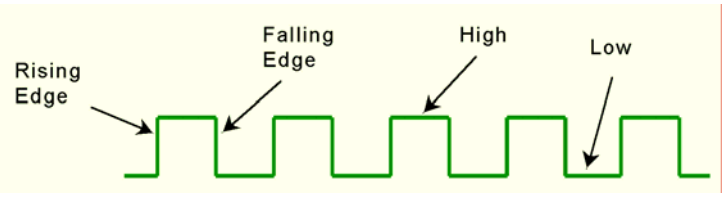
- Synchronous
  - Easier to analyze because can factor out gate delays
  - Set clock so changes allowed to occur before next clock pulse
- Asynchronous
  - Asynchronous (latch) means that output changes very soon after inputs change
  - Potentially faster
  - Harder to analyze
- Will look mostly at synchronous

Sequential Circuits

- As the name implies, sequential logic circuits require a means by which events can be sequenced.
- State changes are controlled by clocks.
  - A “clock” is a special circuit that sends electrical pulses through a circuit.
- Clocks produce electrical waveforms such as the one shown below.

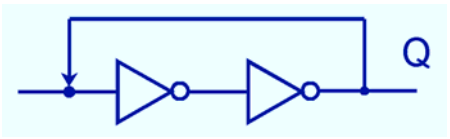
Edge- vs. Level-Triggered

- State changes occur in sequential circuits only when the clock ticks.
- Circuits can change state on the rising edge, falling edge, or when the clock pulse reaches its highest voltage.
- Circuits that change state on the rising edge, or falling edge of the clock pulse are called *edge-triggered.*
- *Level-triggered circuits* change state when the clock voltage reaches its highest or lowest level.



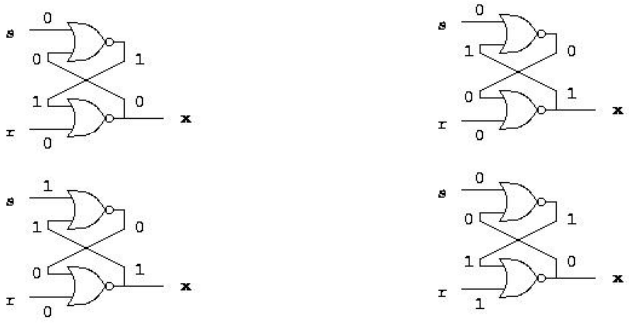
Feedback of Sequential Circuits

- To retain their state values, sequential circuits rely on *feedback.*
- Feedback in digital circuits occurs when an output is looped back to the input.
- A simple example of this concept is shown below.
  - If Q is 0 it will always be 0, if it is 1, it will always be 1. Why?



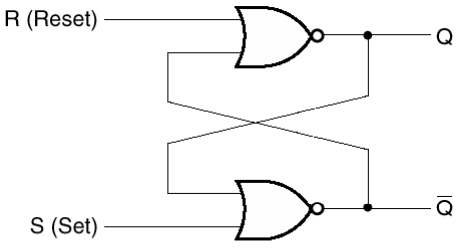
SR (set-reset) Latch (asynchronous)

- You can see how feedback works by examining the most basic sequential logic components, the SR latch.
  - The "SR" stands for set/reset.
- The internals of an SR latch are shown below, along with its block diagram.



SR (set-reset) Latches

- Basic storage made from gates



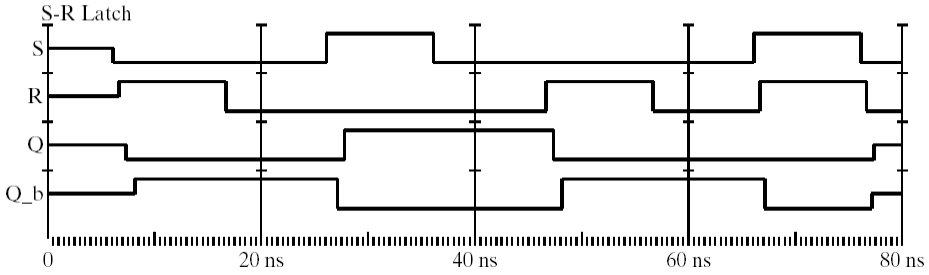
(a) Logic diagram

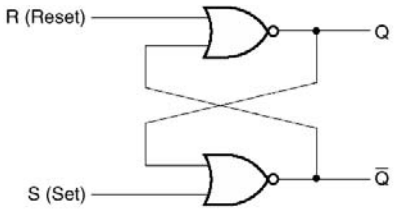
S	R	Q	$\overline{Q}$	
1	0	1	0	Set state
0	0	1	0	
0	1	0	1	Reset state
0	0	0	1	
1	1	0	0	Undefined

(b) Function table

•S & R both 0 in "resting" state
•Have to keep both from 1 at same time

Operation

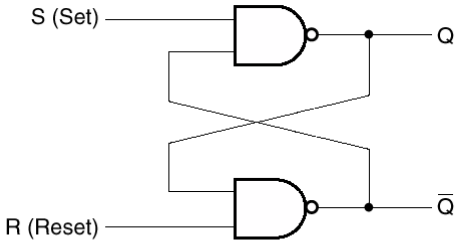




S	R	Q	$\overline{Q}$	
1	0	1	0	Set state
0	0	1	0	
0	1	0	1	Reset state
0	0	0	1	
1	1	0	0	Undefined

$\overline{S}\overline{R}$ Latch

- Similar – made from NANDs



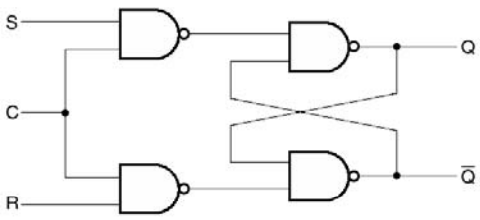
(a) Logic diagram

S	R	Q	$\overline{Q}$	
0	1	1	0	Set state
1	1	1	0	
1	0	0	1	Reset state
1	1	0	1	
0	0	1	1	Undefined

(b) Function table

Add Control Input

- Gates when state can change
- Gate is off when inputs are invalid or unstable



(a) Logic diagram

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	Undefined

(b) Function table

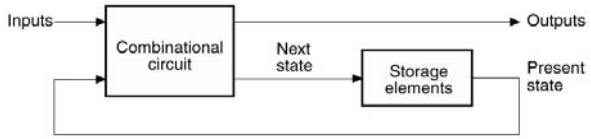
- Is there latch w/ no illegal state?

Transparency

- As long as C (the *trigger*) is high, state can change soon after input change
- This is called *transparency*
- What's problem with that?
  - Unstable results for unstable inputs -> not accurate results

Effects of Transparency

- Output of one latch may feedback
  - So more state changes may happen
  - Depends on gate delays



Block Diagram of a Sequential Circuit

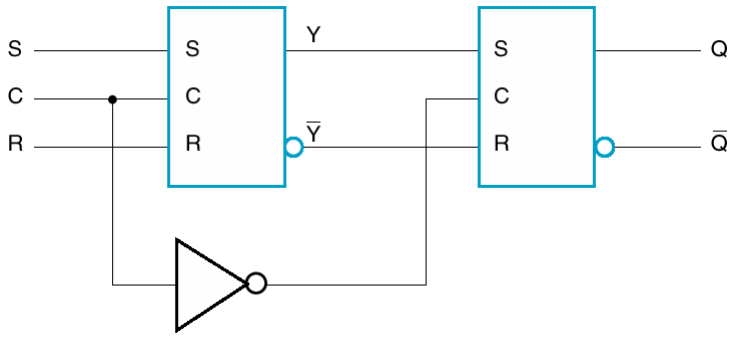
- Want to change latch state *once*
  - Depending on inputs at time of clock

Flip-Flops

- Synchronous version of latches
  - *Be careful this book use latch and flip-flop interchangeably*
- Ensure only one transition
- Edge trigger
- Pulse trigger
  - Master-slave FF

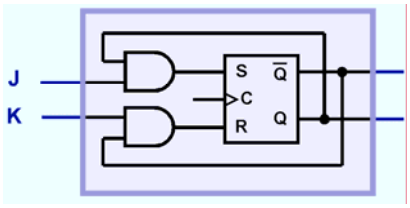
Master-Slave Flip-Flop

- Either R (master) or L (slave) is enabled, not both



J-K Flip-Flop

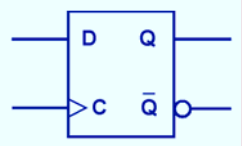
- At the right, we see how an SR flip-flop can be modified to create a JK flip-flop.
- The characteristic table indicates that the flip-flop is stable for all inputs.

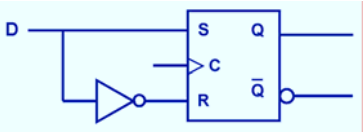


J	K	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\overline{Q}$(t)

D Flip-Flop

- Another modification of the SR flip-flop is the D flip-flop, shown below with its characteristic table.
- You will notice that the output of the flip-flop remains the same during subsequent clock pulses. The output changes only when the value of D changes.
- The D flip-flop is the fundamental circuit of computer memory.
  - D flip-flops are usually illustrated using the block diagram shown below.
- The characteristic table for the D flip-flop is shown at the right.





D	Q(t+1)
0	0
1	1