Computer Systems and Networks

ECPE 170 – Jeff Shafer – University of the Pacific

Digital Logic
Sequential Circuits

Quiz 1

- Return Quiz 1 and review

Upcoming Events

- **Quiz 2 – Friday Feb 3rd**
  - Topics *may or may not* include:
    - Simplifying Boolean expressions with identities
    - Converting between a truth table and a circuit diagram (with logic gates)
    - Common combinational circuits: adders, decoders, multiplexers, shifters
      - Basic operation of these devices, i.e. inputs and outputs
    - Sequential circuits: SR, JK, D flip-flops
      - Basic operation of these devices, i.e. inputs and outputs

Recap from Last Class

- **Why do real hardware devices use NAND / NOR gates instead of AND / OR / NOT gates?**
  - These are "universal" gates – any function can be made using only NAND or only NOR gates
  - Simplifies manufacturing to use the same gate type

Recap from Last Class

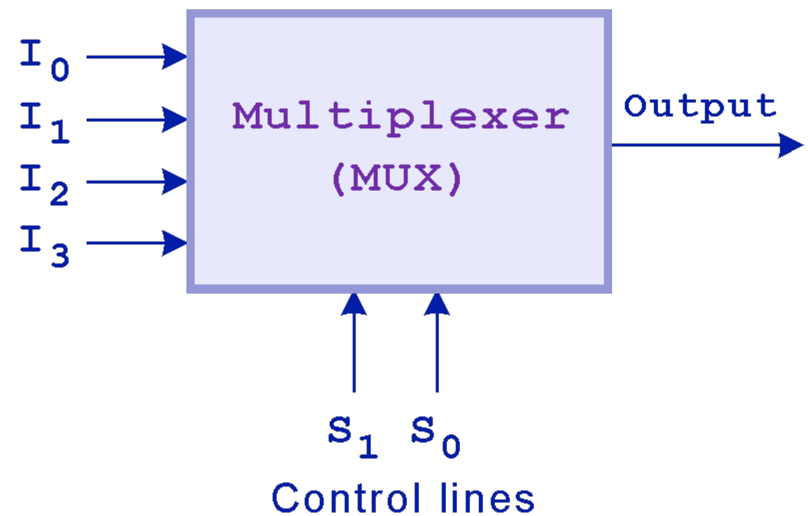
- **What is a combinational circuit?**
  - Circuit where output is based on input only

Recap from Last Class

- **What is the difference between a half-adder and a full-adder?**
  - **Half adder** adds two inputs (x, y) and produces sum and carry-out
  - **Full adder** adds three inputs (x, y, carry-in) and produces sum and carry-out
    - We build it out of two half-adders!

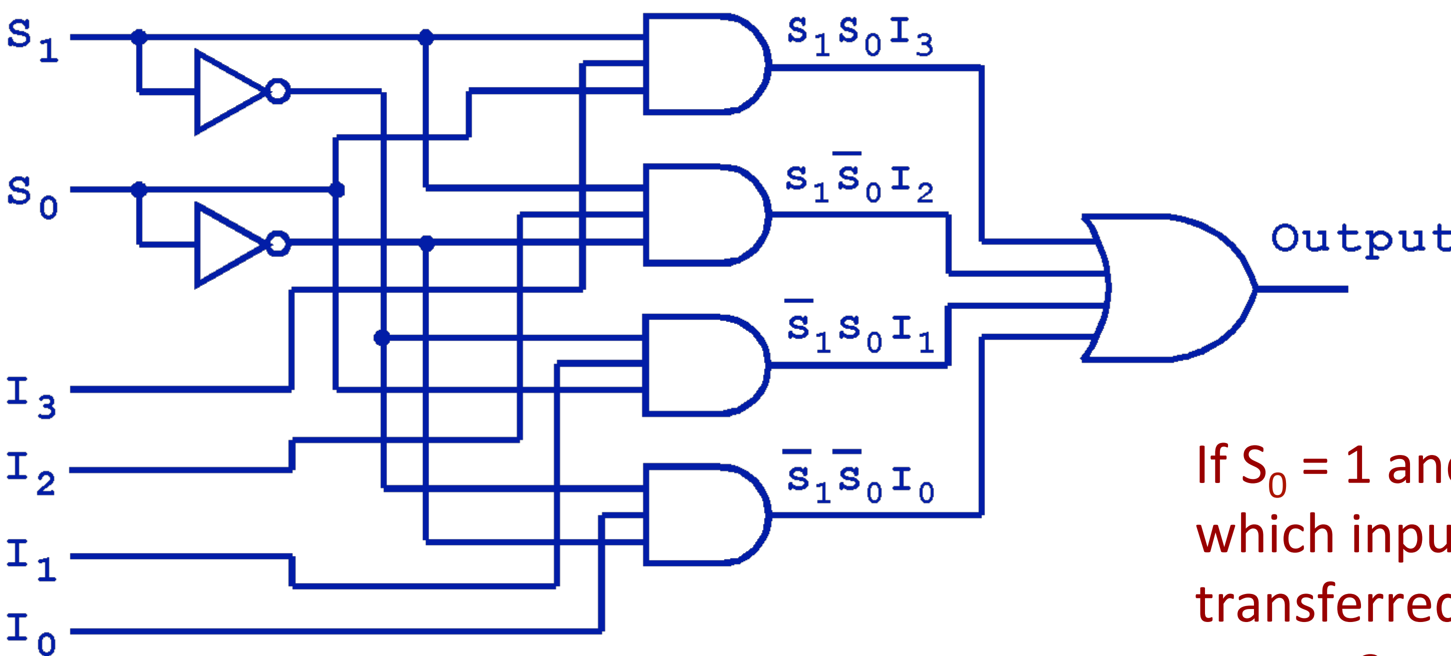
Combinational Circuit – Multiplexer

- A **multiplexer** selects a single output from several inputs
- Which input is chosen?
  - Selected by the value on the multiplexer's control lines
- To select from *n* inputs, $\log_2 n$ control lines are needed.



Combinational Circuit – Multiplexer

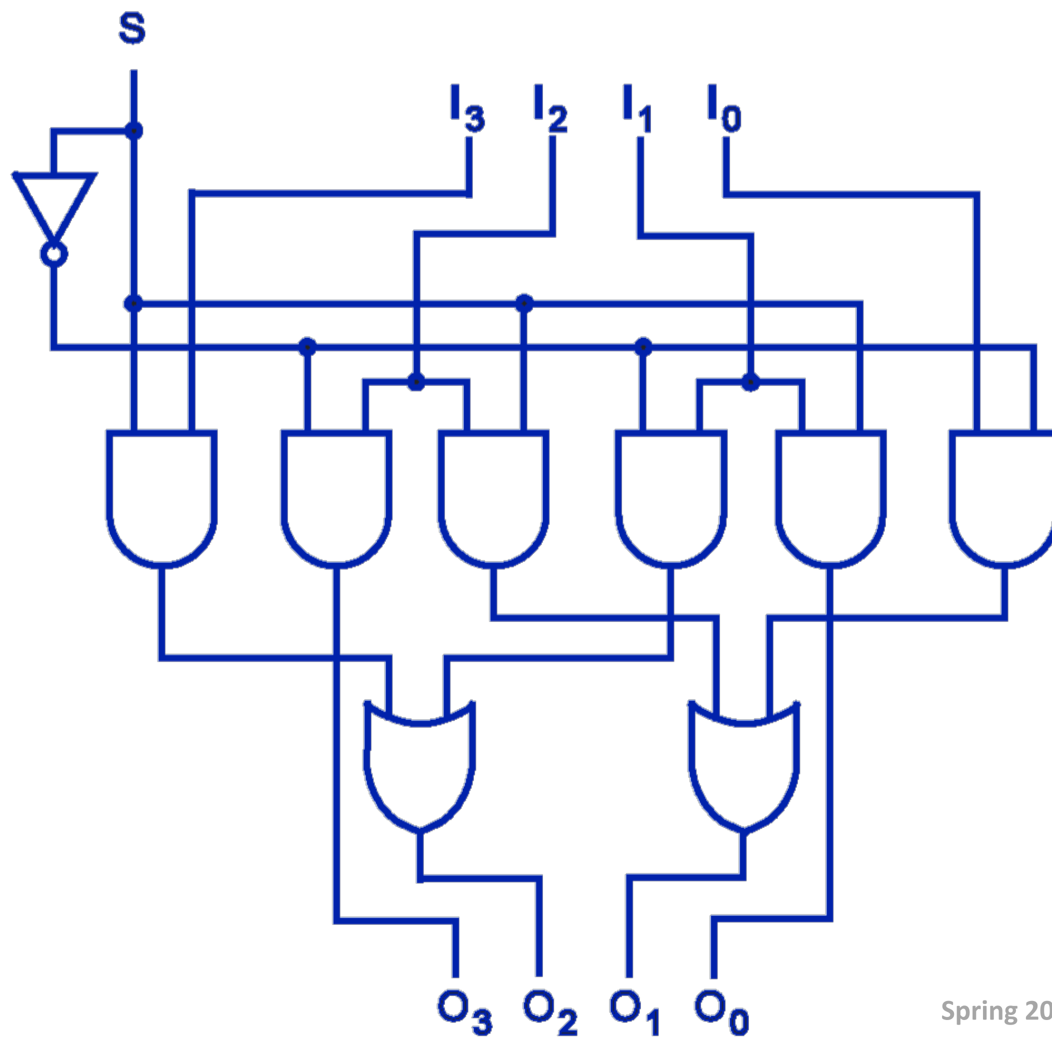
- Implementation of a 4-to-1 multiplexer



If $S_0 = 1$ and $S_1 = 0$, which input is transferred to the output?

Combinational Circuit – Shifter

- This **shifter** moves the bits of a 4-bit input one position to the left or right
- **If S = 0, in which direction do the input bits shift?**
  - Left!



Combinational Circuits

- **Does the output of a combinational circuit change instantly when the input changes?**
  - No – takes a tiny (but measurable) length of time
  - Electrical signals in a wire have a finite speed
  - A transistor takes a finite time to change state

Sequential Circuits

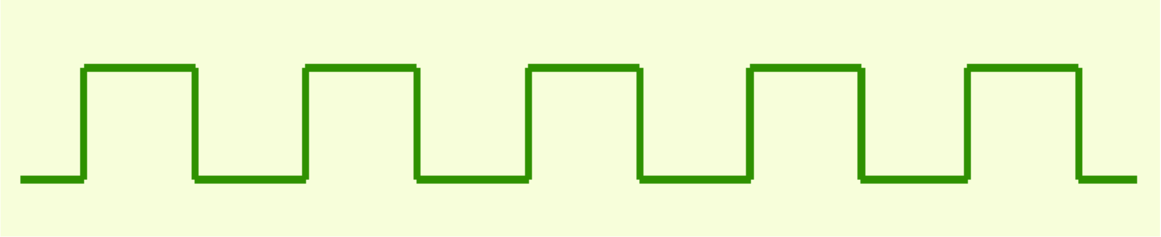
Sequential Circuits

- Combinational logic circuits
  - Immediately apply Boolean function to set of inputs
  - This does not work for all problems!
- What if we want a circuit that changes its value based on (a) its **inputs** and (b) its **current state**?
  - These circuits have to "remember" their current state
  - This is a **sequential logic circuit**

Sequential Circuits

- Sequential logic circuits require a means by which events can be sequenced
  - The **clock**!

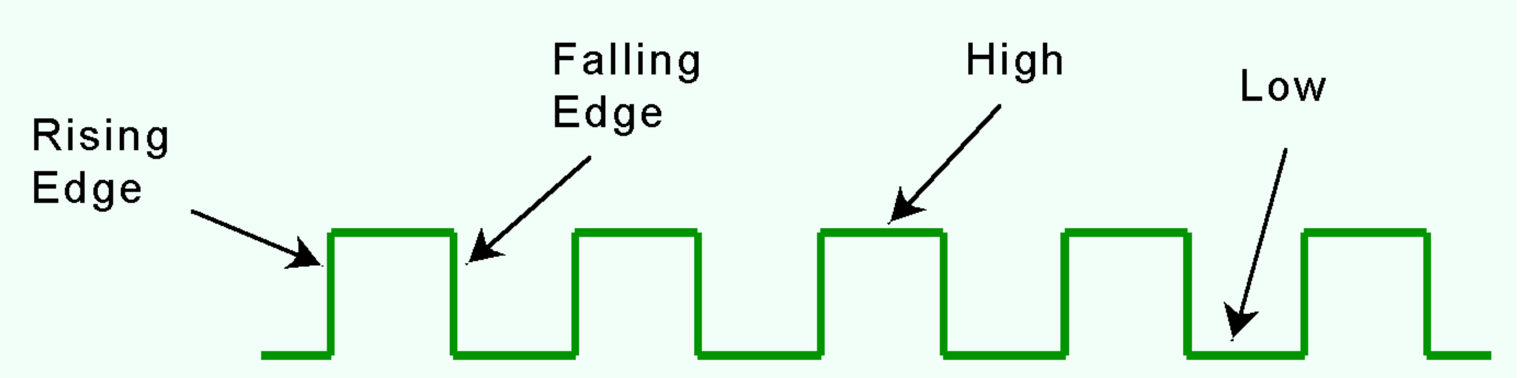
- What is a clock?
  - Not a "wall clock"
  - Circuit that sends electrical pulses through a system



Sequential Circuits

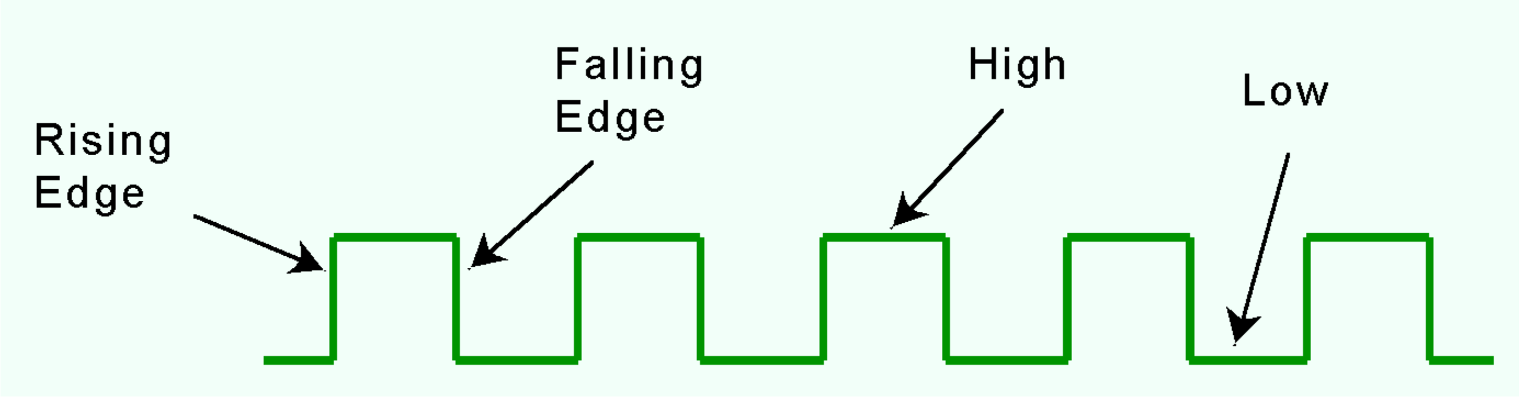
- State changes occur in sequential circuits only when the clock "ticks"
- Circuits can change state on the:
  - Rising edge, or
  - Falling edge, or
  - When the clock pulse reaches its highest voltage

Pick 1 option, not all 3!



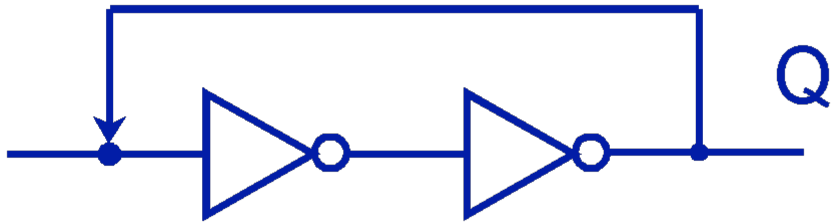
Sequential Circuits

- **Edge-triggered** circuits
  - Change state on the rising edge or falling edge of the clock pulse
- **Level-triggered** circuits
  - Change state when the clock voltage reaches its highest or lowest level



Sequential Circuits

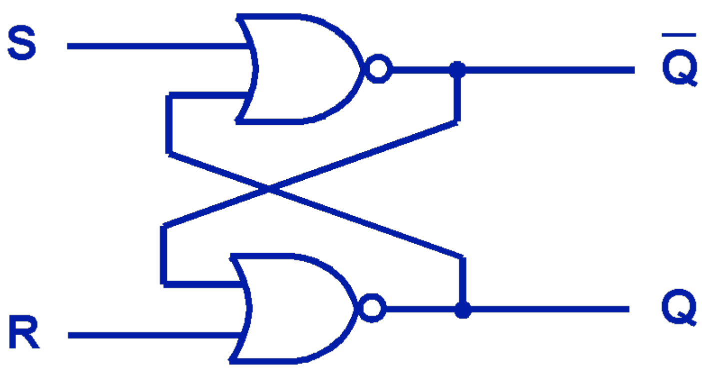
- **How can we make a circuit that uses its current output in deciding its next output?**
  - Feedback – loop an output back to the input
- Example:
  - If Q is 0 it will always be 0
  - If Q is 1, it will always be 1



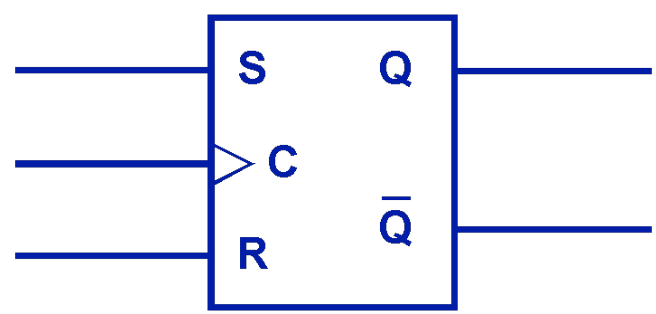
Sequential Circuits – SR Flip-flop

- SR Flip-flop employs feedback
  - The "SR" stands for set/reset
  - Basic storage element

Internal design (clock not shown):

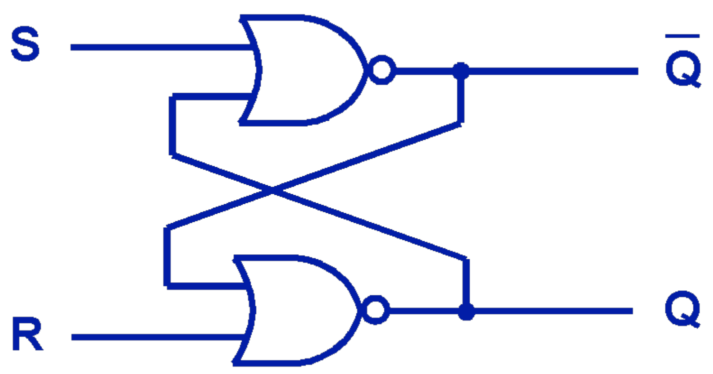


Block diagram (with clock):



Sequential Circuits – SR Flip-flop

- What does the truth table of an SR flip-flop look like?
  - Q(t) is the value of the output Q at time t
  - Q(t+1) is the value of Q after the next clock pulse



S	R	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined

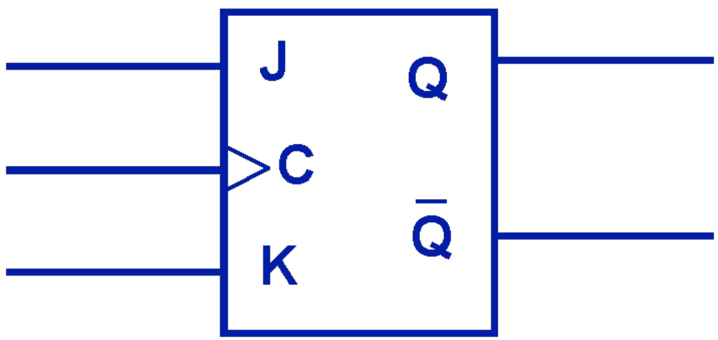
Sequential Circuits – SR Flip-flop

- The SR flip-flop actually has three inputs: S, R, and its current output, Q
- More complete truth table
  - Two undefined values!
  - SR flip-flop unstable when "set" and "reset" are both active

S	R	Present State Q(t)	Next State Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	undefined
1	1	1	undefined

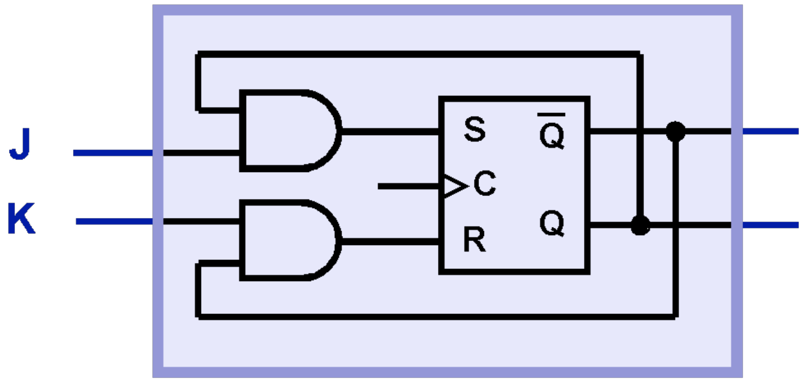
Sequential Circuits – JK Flip-flop

- **JK flip-flop** removes this risk
  - Ensures that both "set" and "reset" inputs to an SR flip-flop will never both be 1
  - "JK" named after Jack Kilby
    - 2000 Nobel Prize winner for invention of the integrated circuit while at Texas Instruments



Sequential Circuits – JK Flip-flop

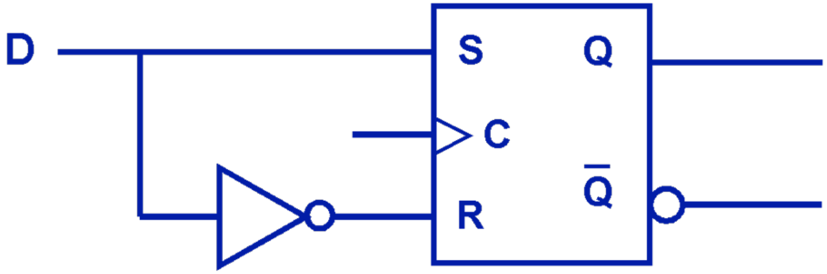
- JK flip-flop is really just a wrapper around a basic SR flip-flop
- JK is stable for all inputs
  - J=K=1: Toggle output



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)

Sequential Circuits – D Flip-flop

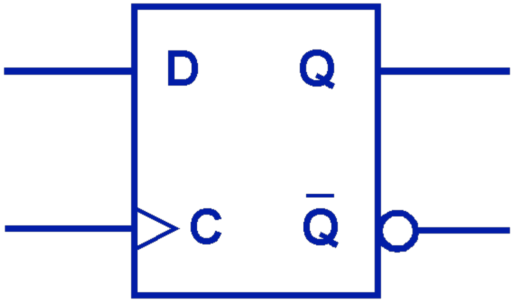
- D Flip-Flop
  - Another modification of the SR flip-flop
  - D=Data *(but I remember D=Delay…)*
- Output of the flip-flop remains the same during subsequent clock pulses
  - Output changes only when D changes



D	Q(t+1)
0	0
1	1

Sequential Circuits – D Flip-flop

- D flip-flop is the fundamental circuit of computer memory
  - Usually illustrated using the block diagram shown below



D	Q(t+1)
0	0
1	1

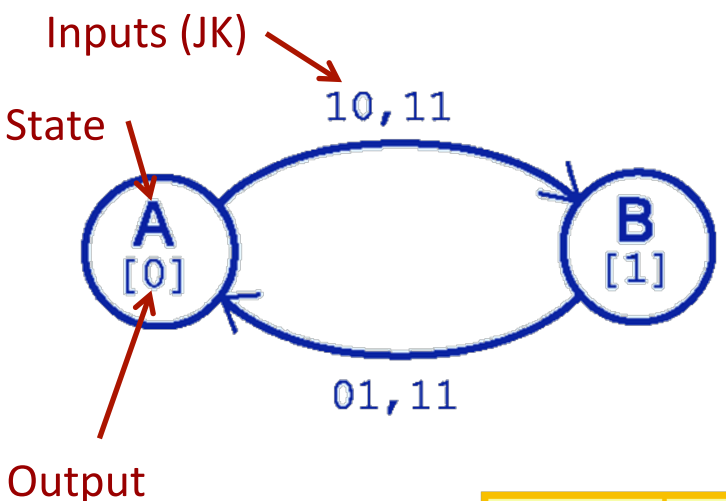
State Machines

State Machines

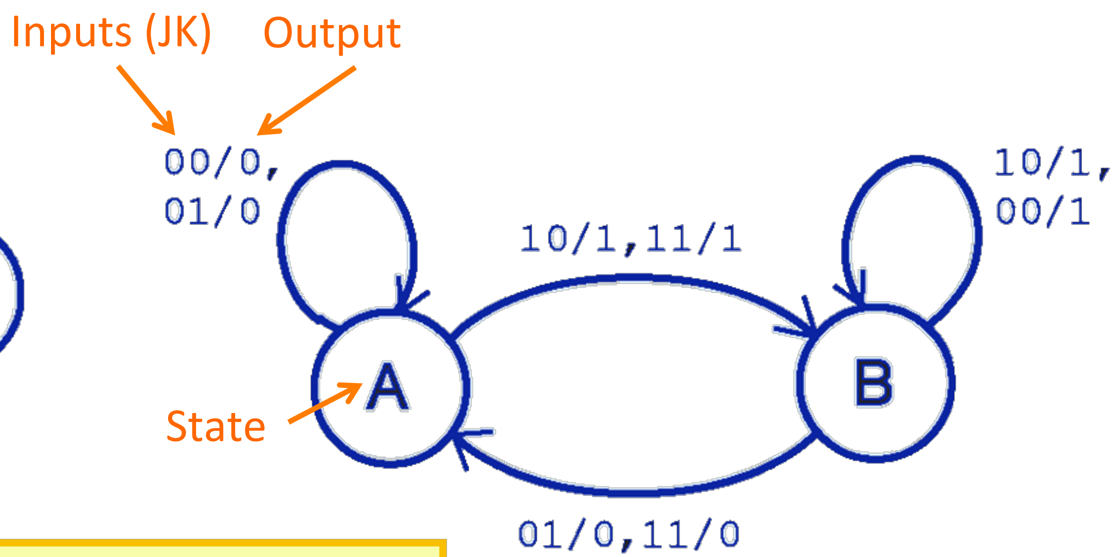
- How do we design complicated sequential systems?
  - **Finite State Machine** (FSM)
  - In visual form
    - A set of nodes that hold the states of the machine
    - A set of arcs that connect the states
- Two different types of state machines: **Moore** and **Mealy**
  - Both produce systems that produce the same output
  - Differ only in how the output of the machines are expressed
    - Moore: place outputs on each node
    - Mealy: present outputs on the transitions

JK Flip-Flop in State Machine Form

Moore FSM



Mealy FSM

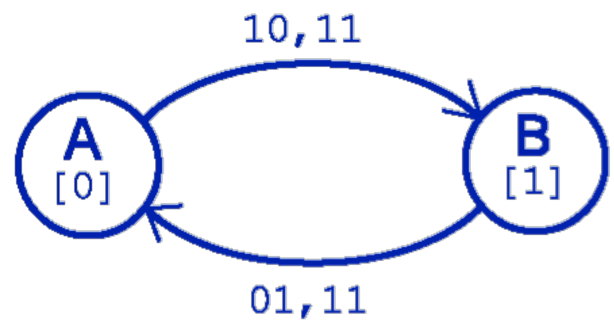


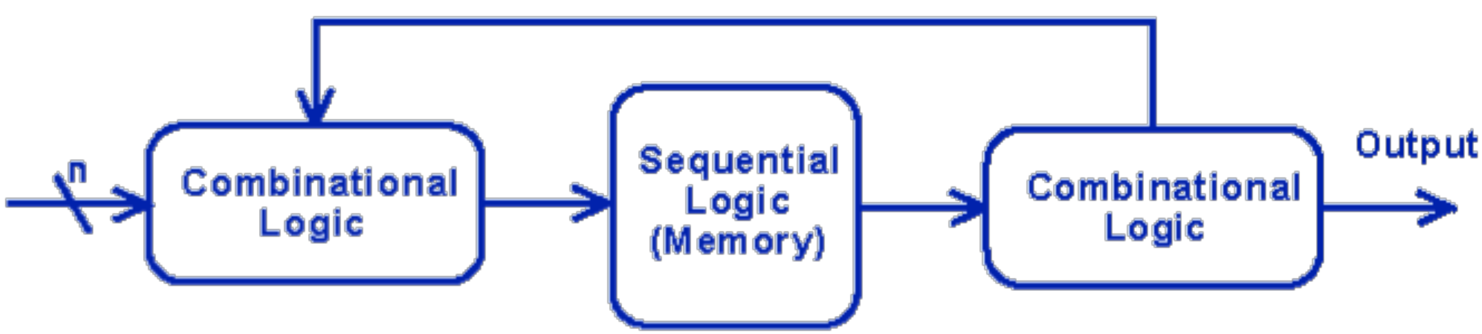
J	K	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\bar{Q}$(t)

Different Implementations

- Although the behavior of Moore and Mealy machines is identical, their implementations differ:

Moore machine:

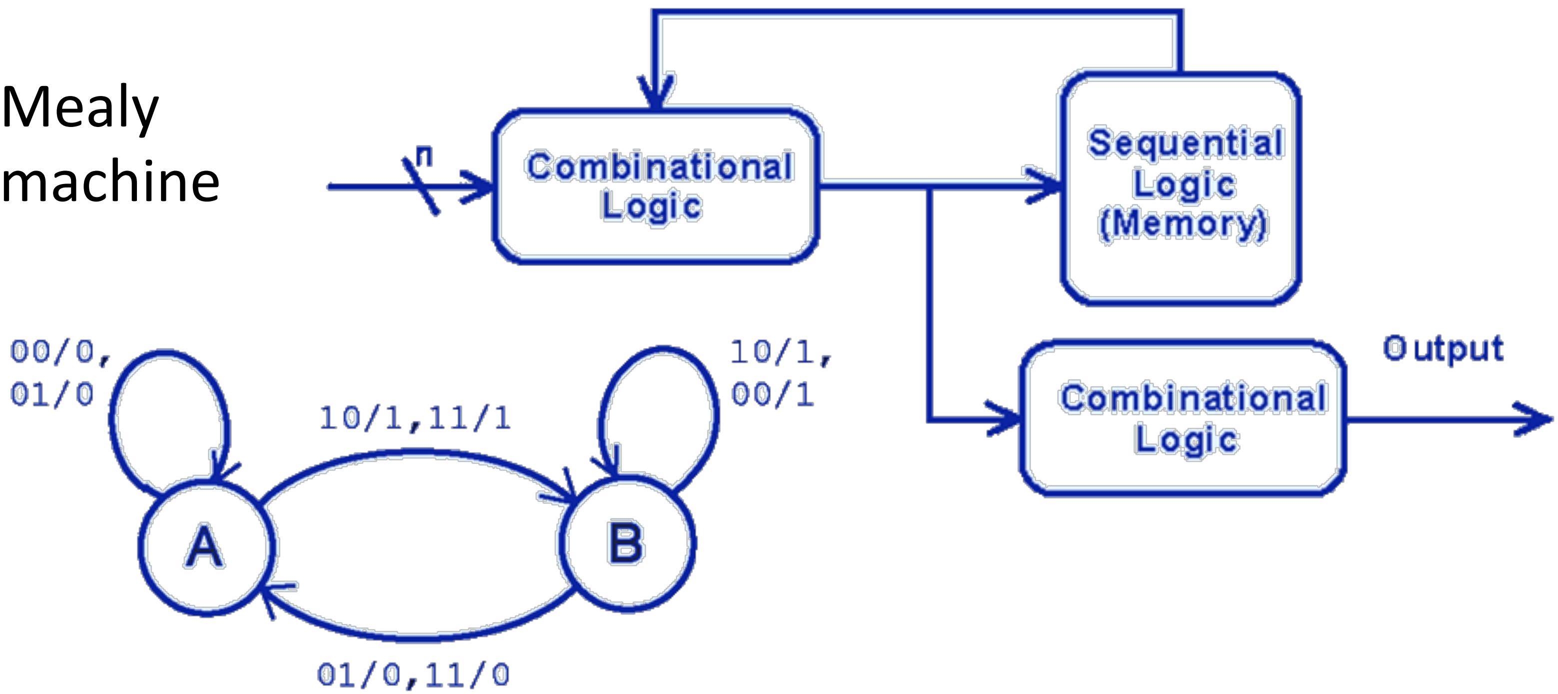




Different Implementations

- Although the behavior of Moore and Mealy machines is identical, their implementations differ:

Mealy machine

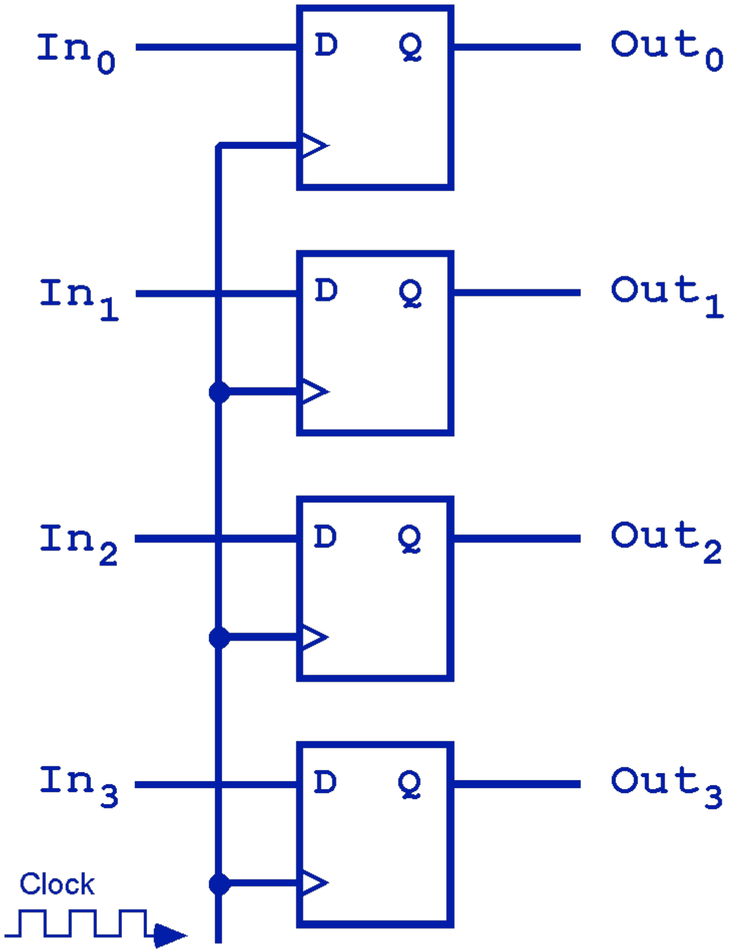


Sequential Circuit Applications

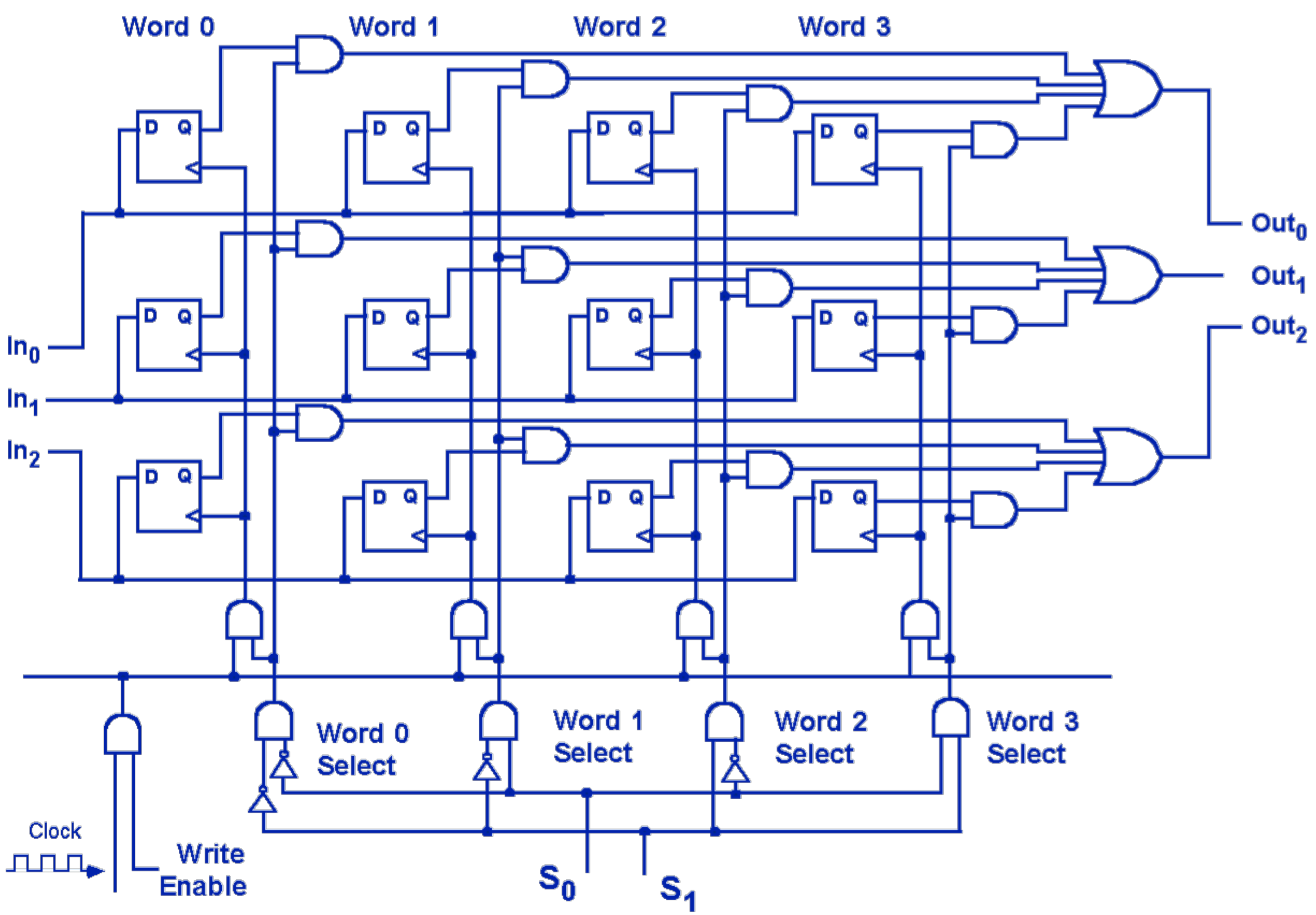
- When do I use sequential circuits?
  - Whenever the application is "**stateful**"
  - The next state of the machine depends on the **current state** of the machine and the input
- Stateful applications requires both combinational and sequential logic
- Examples: Register, Memory, Counters, …

Sequential Circuits – Register

- This illustration shows a 4-bit register consisting of D flip-flops. You will usually see its block diagram (below) instead.

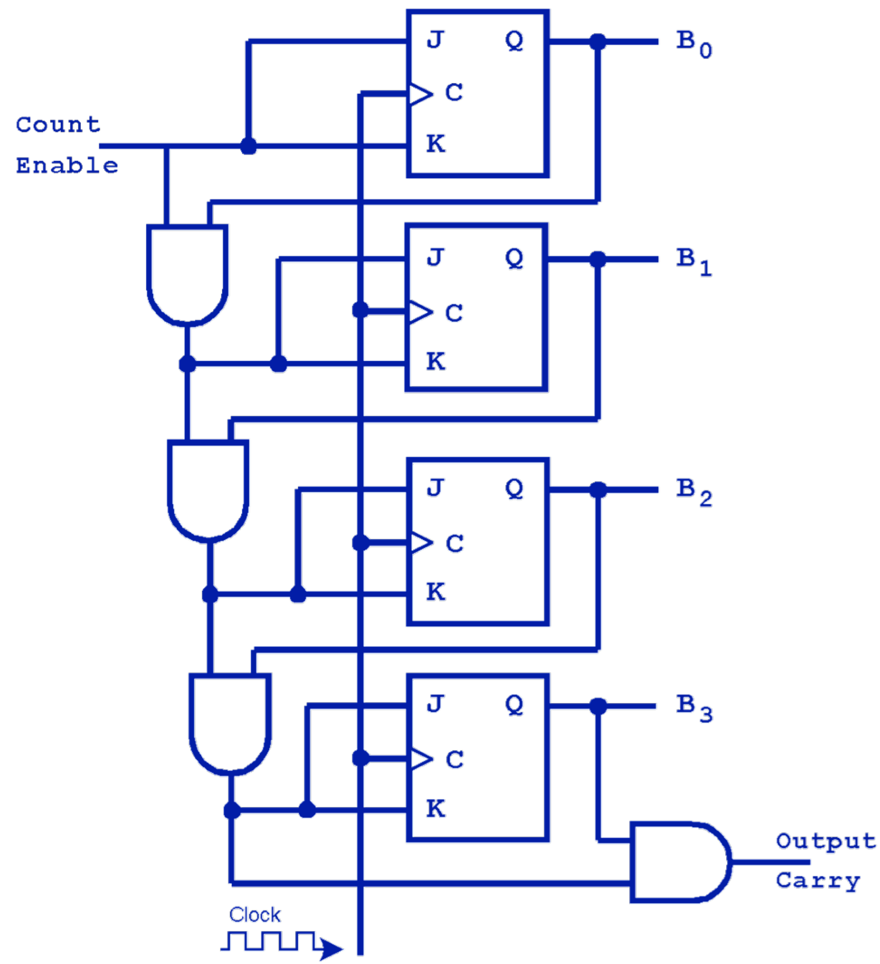


Sequential Circuits – Group of Registers



Sequential Circuits – Binary Counter

- Binary counter operation
  - JK flip-flops toggle when J=K=1
  - Low-order bit is complemented at each clock pulse
  - Whenever low order bit changes from 0 to 1, the next bit is complemented, and so on through the other flip-flops



Designing Circuits

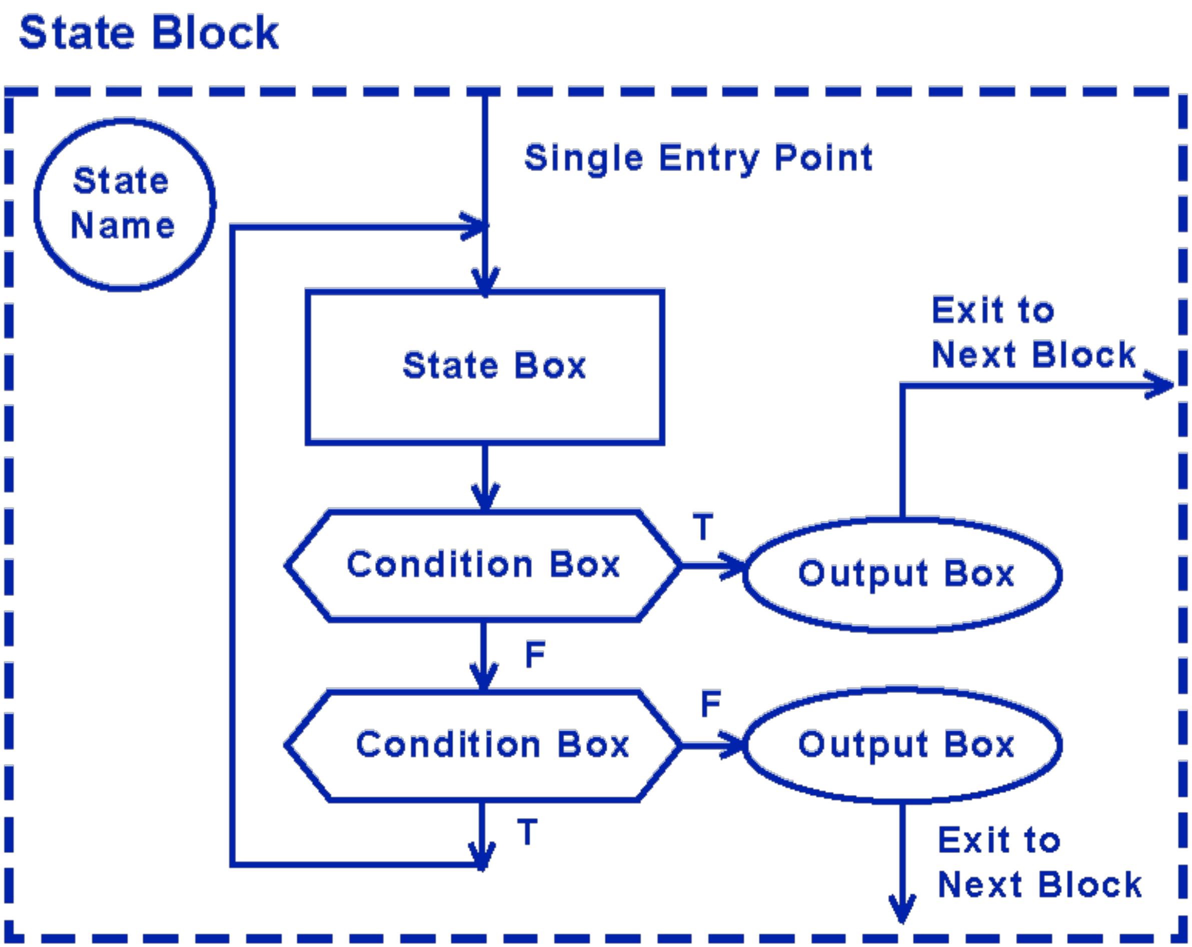
- **Do designers usually lay out circuits by hand?**
  - No – designers today rely on specialized software to create efficient circuits
  - Software is an enabler for the construction of better hardware!
- Many challenges in modern hardware designs
  - Sheer number of gates to implement!
    - Create "building blocks" (modules) that can be quickly assembled
  - Timing constraints – Result is **correct**, but **<u>when</u>** is it correct?
    - Propagation delays occur between the time when a circuit's inputs are energized and when the output is accurate and stable

"Bonus" Slides – Algorithmic State Machines

Algorithmic State Machine

- Moore and Mealy machines are challenging to draw for complex designs
  - An interaction of numerous signals is required to advance a machine from one state to the next
- Alternate approach: **Algorithmic State Machine**
  - A block diagram approach to describing digital systems

Algorithmic State Machine



Algorithmic State Machine – Microwave Oven