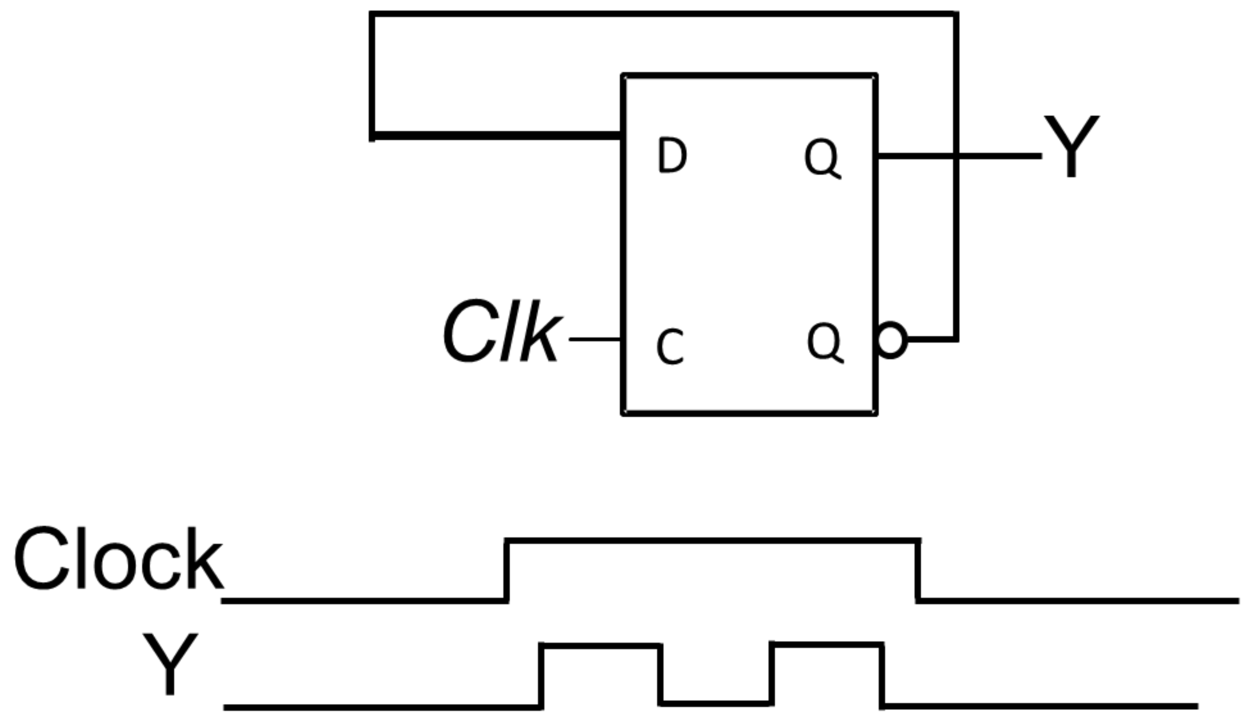
THE UNIVERSITY OF TEXAS AT DALLAS

CS4341 Digital Logic & Computer Design

Lecture Notes 13

Omar Hamdy
Assistant Professor
Department of Computer Science

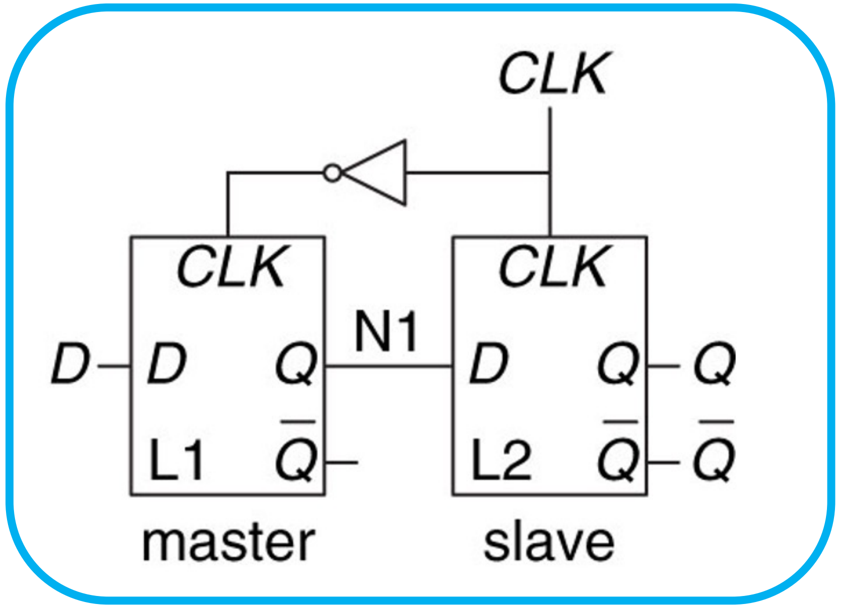
Review: D Latch Timing Problem

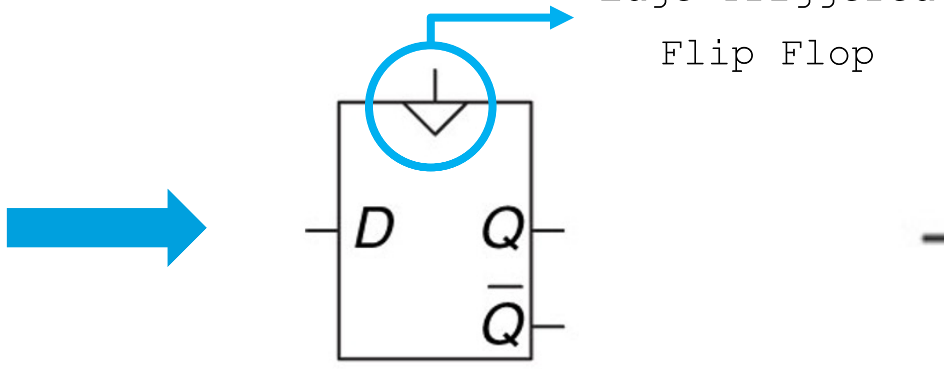


Undesirable behavior

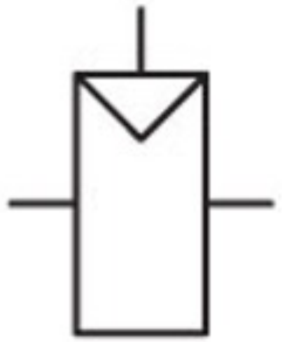
Review: D Flip-Flop

- The solution of the D-Latch undesired behavior is to break the closed path from Y to itself.
- This is done using D flip-flop, which is built from two back-to-back D latches controlled by complementary clocks



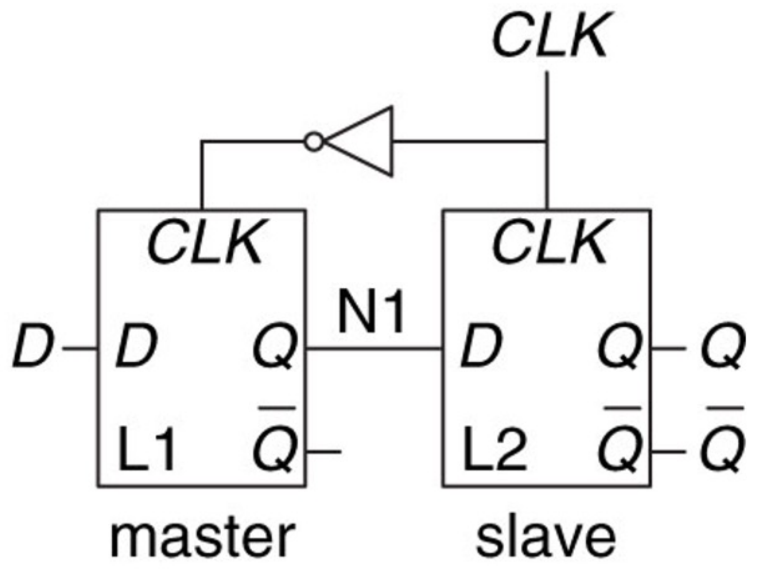


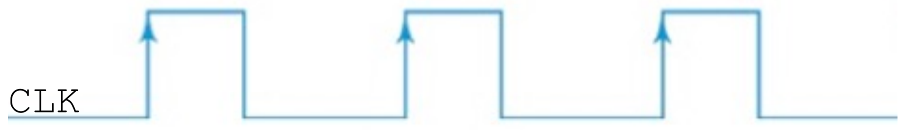
D Flip-Flop



Condensed Symbol

D Flip-Flop Analysis

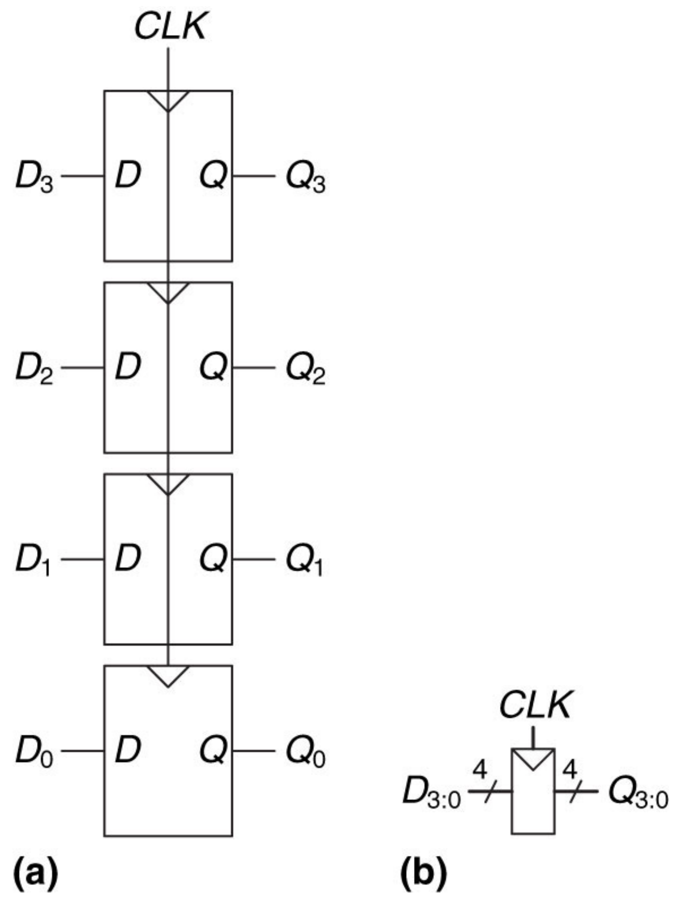




- When CLK is 0, the master latch is transparent, and the slave is opaque. Hence, D value is propagated to N1
- When CLK is 1, the slave latch is transparent, and the master is opaque. Hence, N1 value is propagated to Q.
- From timing perspective: Whatever value was at N immediately before the *CLK* “rises” is copied to Q immediately after the *CLK* “rises”.
- Therefore, a D flip-flop, copies D to Q on the *rising edge* of the *CLK*, and remembers (preserves) its state at all other times.

Registers

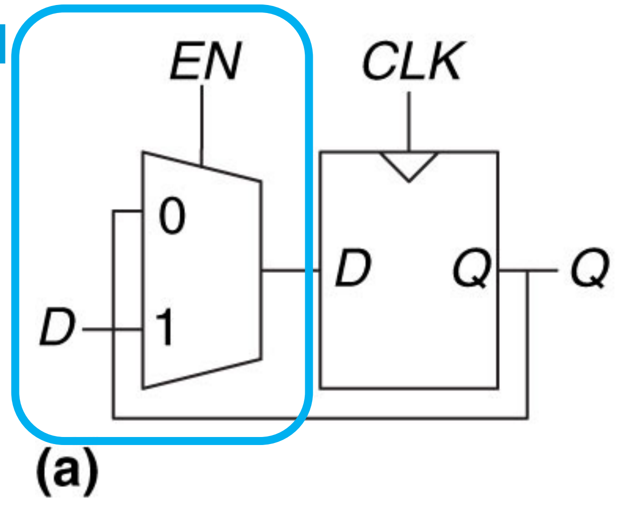
- An N-bit register is a bank of N flip-flops that share a common *CLK* input, so that all bits of the register are updated at the same time
- Registers are the key building block of most sequential circuits
- In 4-bit register, inputs D3:0 and outputs Q3:0 are both 4-bit data busses

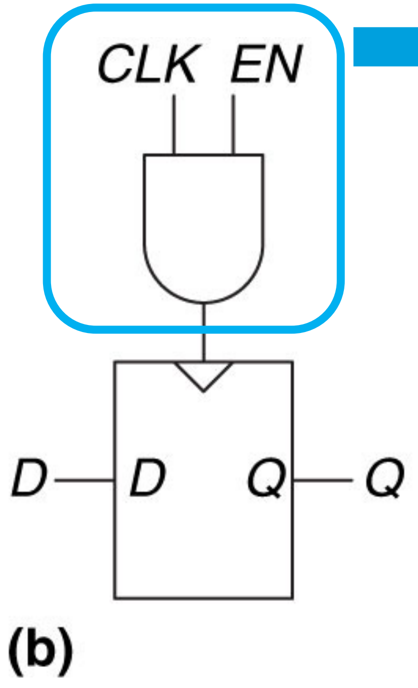


Enabled Flip-Flops

- An Enabled flip-flops adds another control EN to the flip-flop:
  - When EN is TRUE, the enabled flip-flop behaves normally
  - When EN is FALSE, the enabled flip-flop ignores the clock and retains the state.
- How can that be implemented?

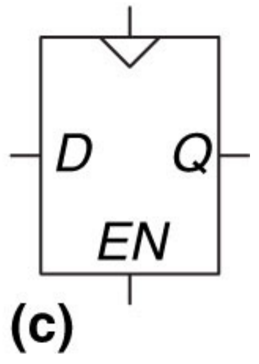
Good approach





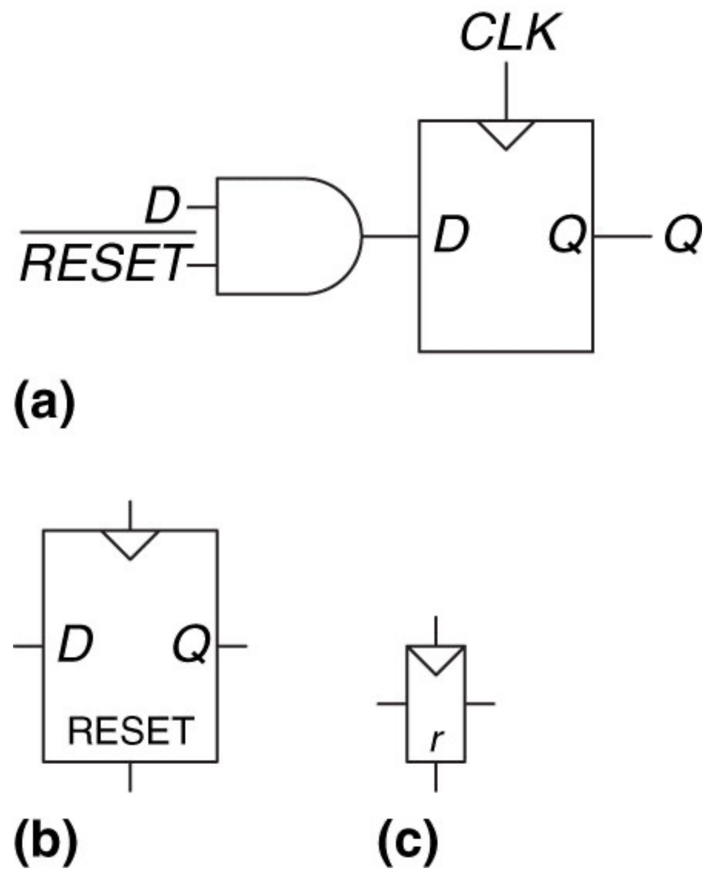
Bad approach:

- EN must not change when CLK is 1, which can cause clock glitch
- AND gate will delay the clock signal and can cause timing error



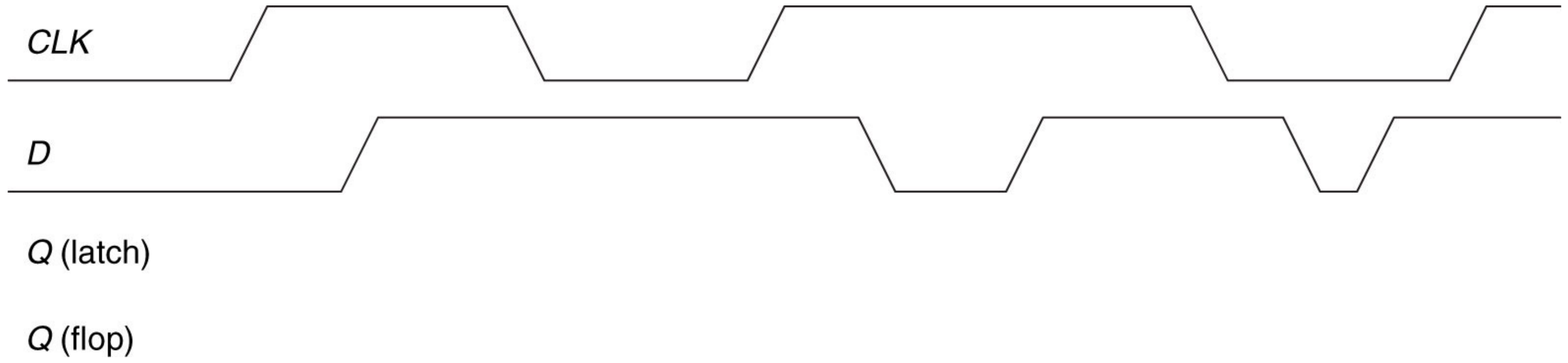
Resettable Flip-Flop

- A resettable flip-flops adds another input $RESET$ to the flip-flop:
  - When $RESET$ is FALSE, the resettable flip-flop behaves normally
  - When $RESET$ is TRUE, the resettable flip-flop ignores D and resets output to 0
  - Can be synchronously or asynchronously reset
- Useful when trying to force all flip-flops into a state (i.e., 0). Example: when system is first turned on.

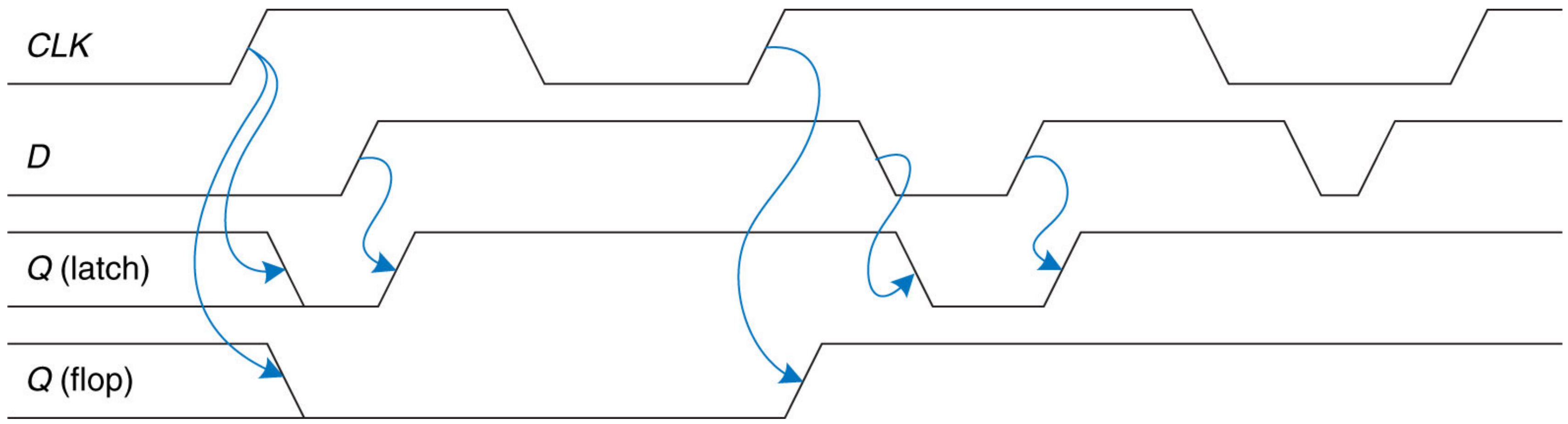


Class Exercise

- Show the output waveforms for the following CLK and D Inputs



Solution

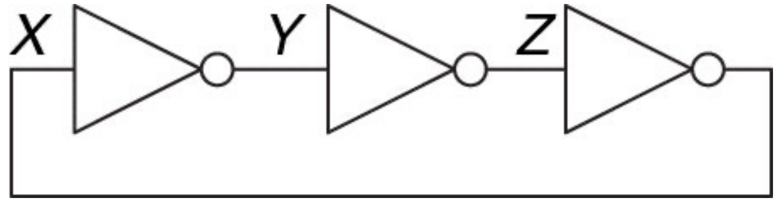


Asynchronous Sequential Circuits Problems

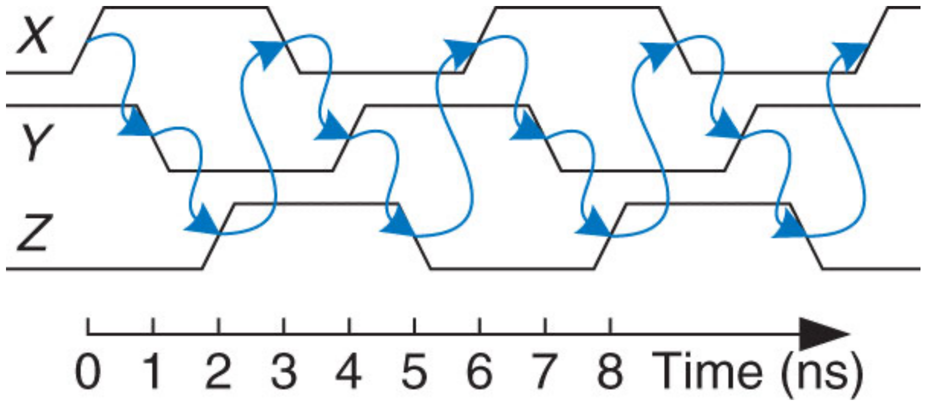
- Sequential circuits are different from combinational circuits that output depends on input as well as current state of the circuit.
- This notion of current state means that the output is fed back into the circuit along with other inputs, known as cyclic path.
- Feeding output into the input directly (as they happen) without control can cause problems and can be challenging to diagnose or control.

Example: Astable Circuits

- Study the behavior of the following circuit assuming an inverter propagation delay of 1ns. Suppose X was 0 the time you started



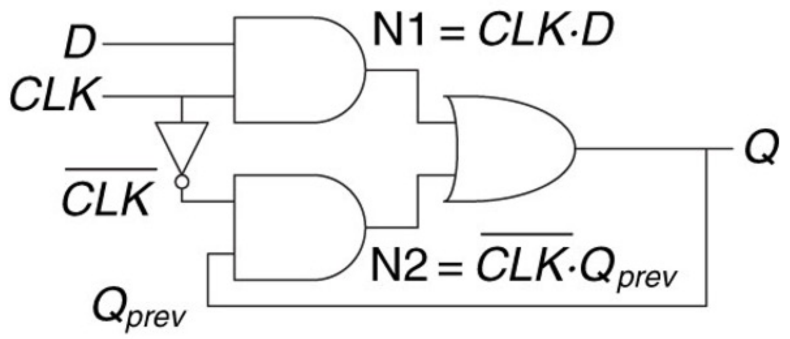
X: 0 (o ns) ➔ 1 (3 ns) ➔ 0 (6 ns) ➔ 1 (9 ns) ➔ 0 (12 ns)

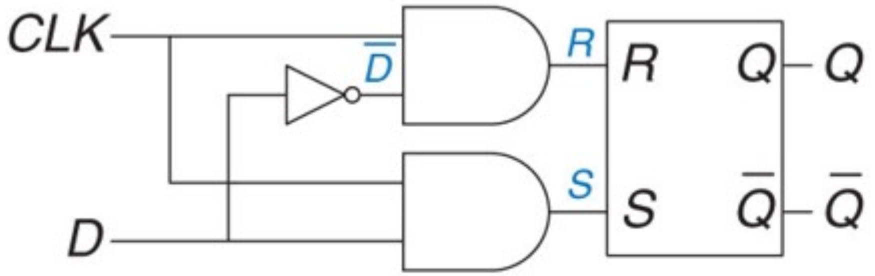


- The circuit has no stable condition .. "astable"
- This particular circuit is called ring oscillator

Example: Race Conditions

- Compare the following design to the D Latch original design, and determine which is better



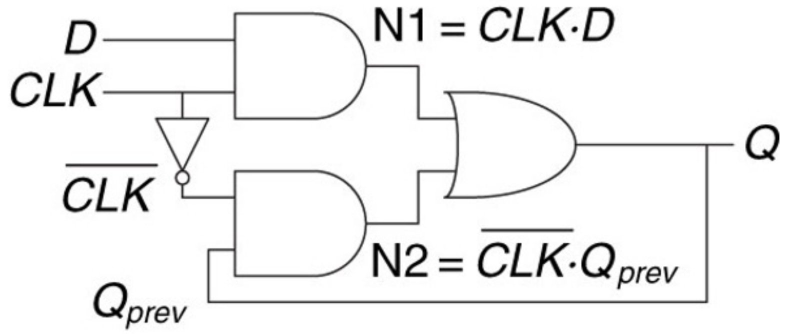


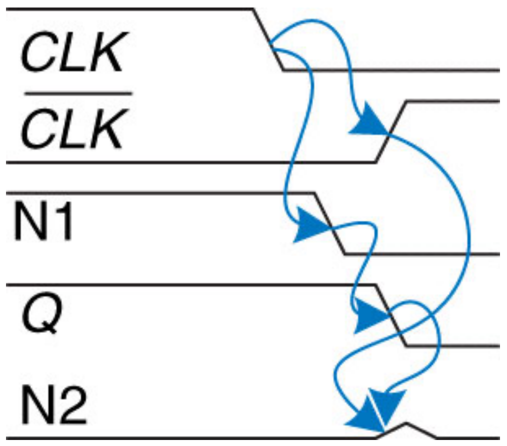
CLK	D	Q_{prev}	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$Q = CLK \cdot D + \overline{CLK} \cdot Q_{prev}$$

Example: Race Conditions

- Let us study the waveform diagram when CLK = D = 1, then CLK changes to 0.





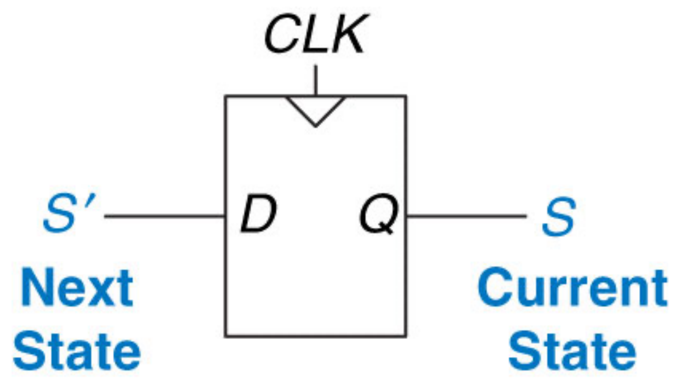
- Race condition is one of the key challenges with asynchronous circuits when output is fed directly into the input of the circuit. Delays can also be impacted by temperature, voltage and other factors.

Synchronous Sequential Circuits

- To avoid the problems of race or astable behaviors, registers are inserted in the feedback return path of the sequential circuit.
- The circuit can then be seen as collection of combinational logic and registers.
- Registers are all “synchronized” by the same clock.
- Only at the rising edge of the clock, registers values along with other inputs can change “the sate” of the circuit (values stored in registers).
- There are finite number of states the circuit can be in $\{S_0, S_1, \ldots, S_{k-1}\}$
- We use `current state` indicating the state at the current time, and `next state` to indicate the new state waiting to enter at the new clock edge.

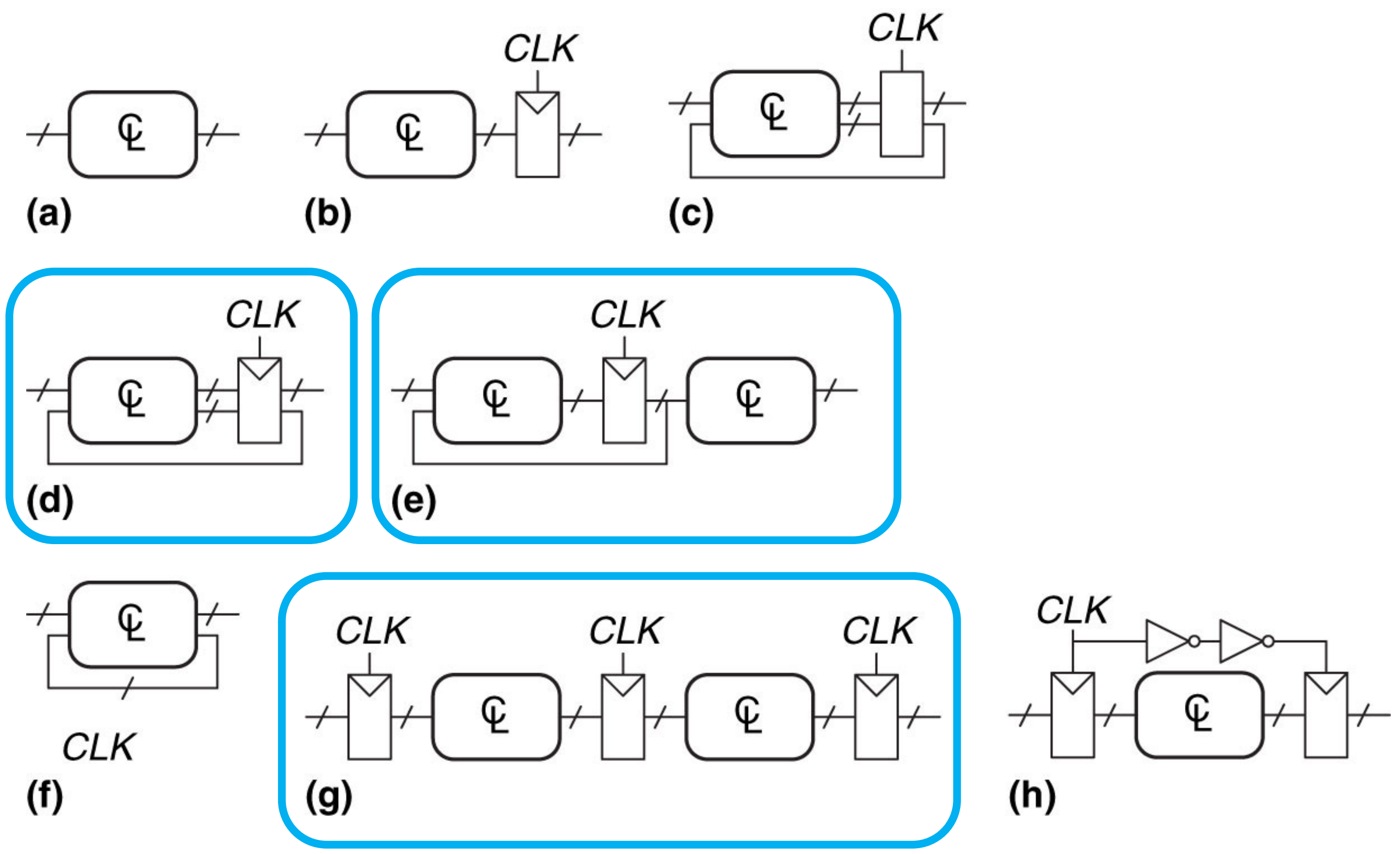
Synchronous Sequential Circuits

- Synchronous sequential circuits mean:
  - Every circuit element is either a register or a combinational circuit
  - At least one circuit element is a register
  - All registers receive the same clock signal
  - Every cyclic path contains at least one register
- Flip-Flop is the simplest form of a synchronous sequential circuit:
  - One input D, one CLK, one output Q and two states {0, 1}
  - `Next state` (S') is D, and `current state` (S) is Q
- Other synchronous sequential circuits include finite sate machines and pipelines



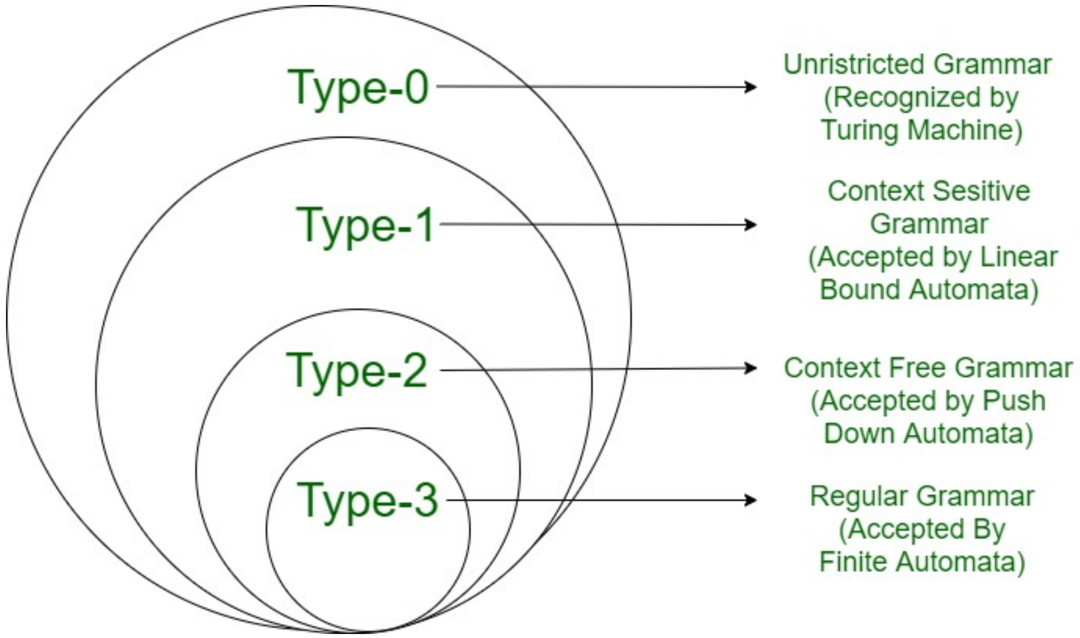
Test Your Knowledge

- Determine the synchronous sequential circuits



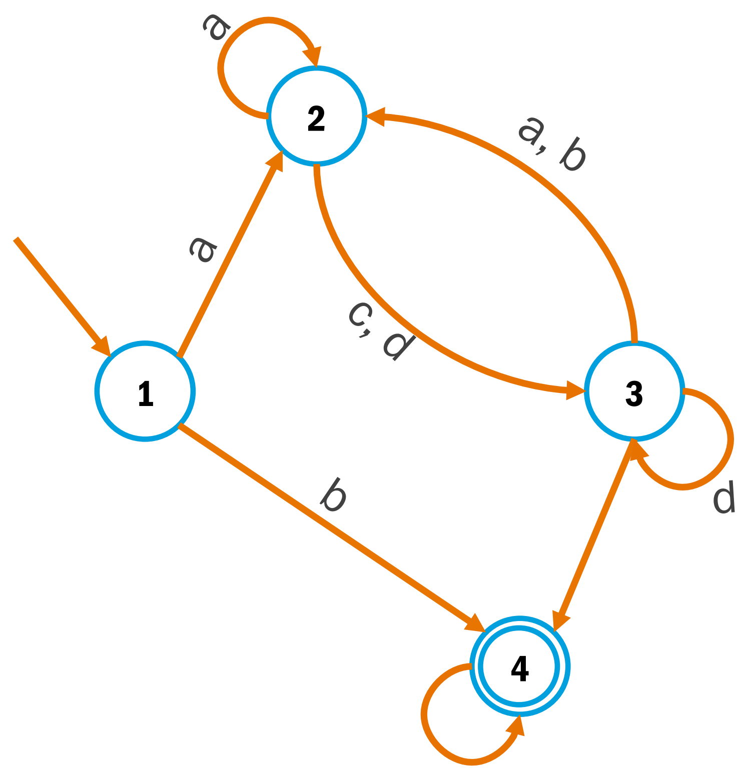
Finite State Machines

- Finite state machine is a mathematical model of computation to describe the system different states.
- FSM can be in exactly one state out of “finite” number of states at any given time.
- FSM is part of the automata theory, which deals with the logic of computation with respect to simple machines.


Type-0
Unristricted Grammar (Recognized by Turing Machine)
Type-1
Context Sesitive Grammar (Accepted by Linear Bound Automata)
Type-2
Context Free Grammar (Accepted by Push Down Automata)
Type-3
Regular Grammar (Accepted By Finite Automata)

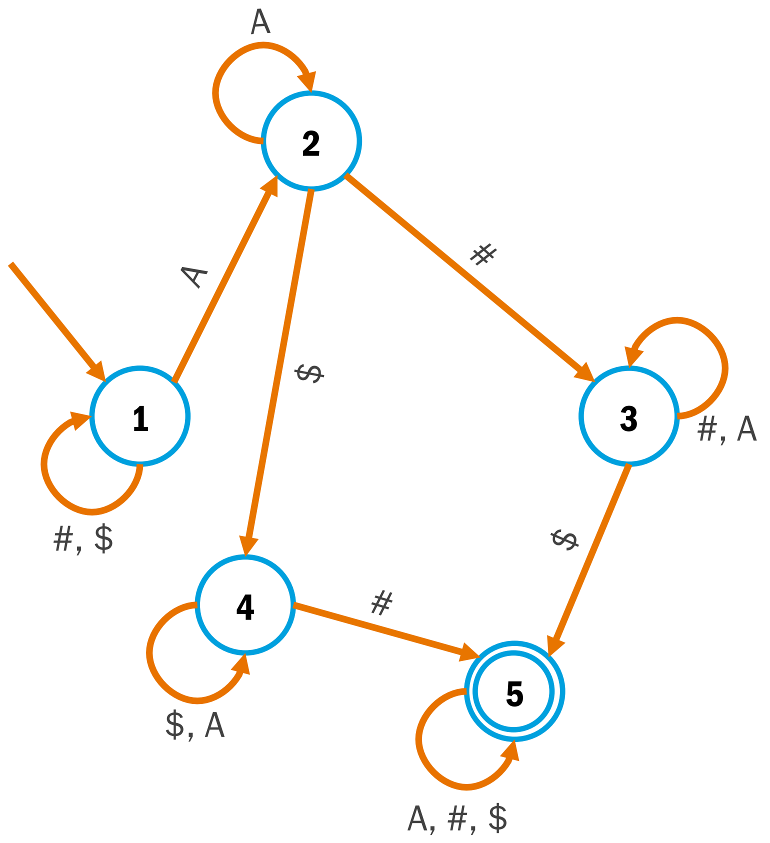
Finite State Machines

- Finite state machine is composed of a known (finite) number of states
- One of them is special and known as the start state
- It has set of transitions which determines how to move from one state to the next
- Conditions (inputs) that tells the machine how to move from one state to the next
- Accepting state (in some types of FSM)



FSM Example

- Design an FSM that validates that a password meets the following criteria:
  - The password must have a sequence in it which:
    - Starts with an alphabet (A)
    - Has at least one number (#)
    - Has at least one symbol ($)



To Do List

- ➢Review lecture notes
- ➢Study chapter 3