CSE 260M / ESE 260
Intro. To Digital Logic & Computer Design

Bill Siever
&
Michael Hall

Reminder

- Studios can (and should) be done collaboratively!
- Homework should be done *independently*
  - Ok to discuss *ideas* and related work from *studio*
  - Should not discuss / share details of specific solutions

This week & Misc.

- Hw #3B Posted / Due Sunday
- Calendar updates: Spring break dates were incorrect; Updated now!

Last Lecture

- Sequential Circuits: Have a loop and outputs impact the inputs
- Bistable: Will hold one of two different states
  - May have some “metastable” condition
- SR-Latch: Cross-coupled NOR version
  - JLS version & time
  - Racing and unpredictable behavior

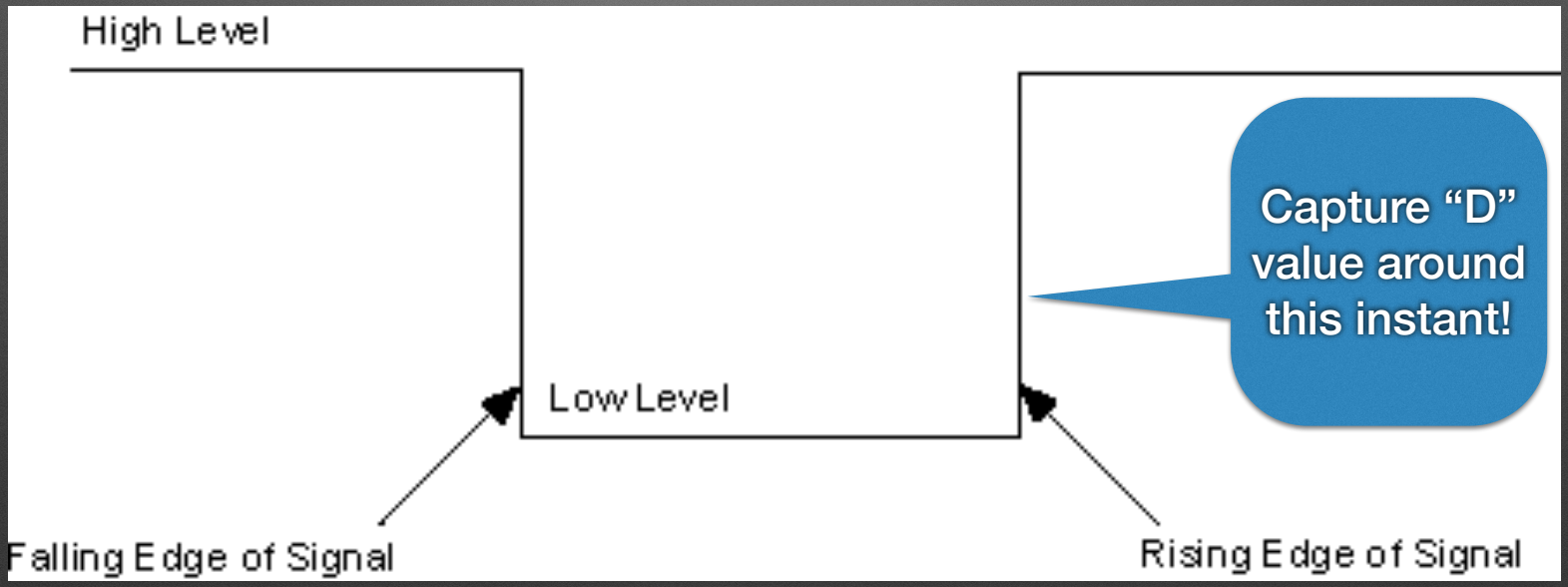
SR Latch Issues: Hazardous Conditions!

- Short pulse
- R & S dropping at the same time

Last Lecture

- D Latch: Transparent on high clock
- D Flip-Flop: Two D-latches that are transparent at opposite clock levels
  - Provide precise timing of data *acquisition* / storage
  - General focus: positive/rising edge triggers

D-Flip-Flop Clock

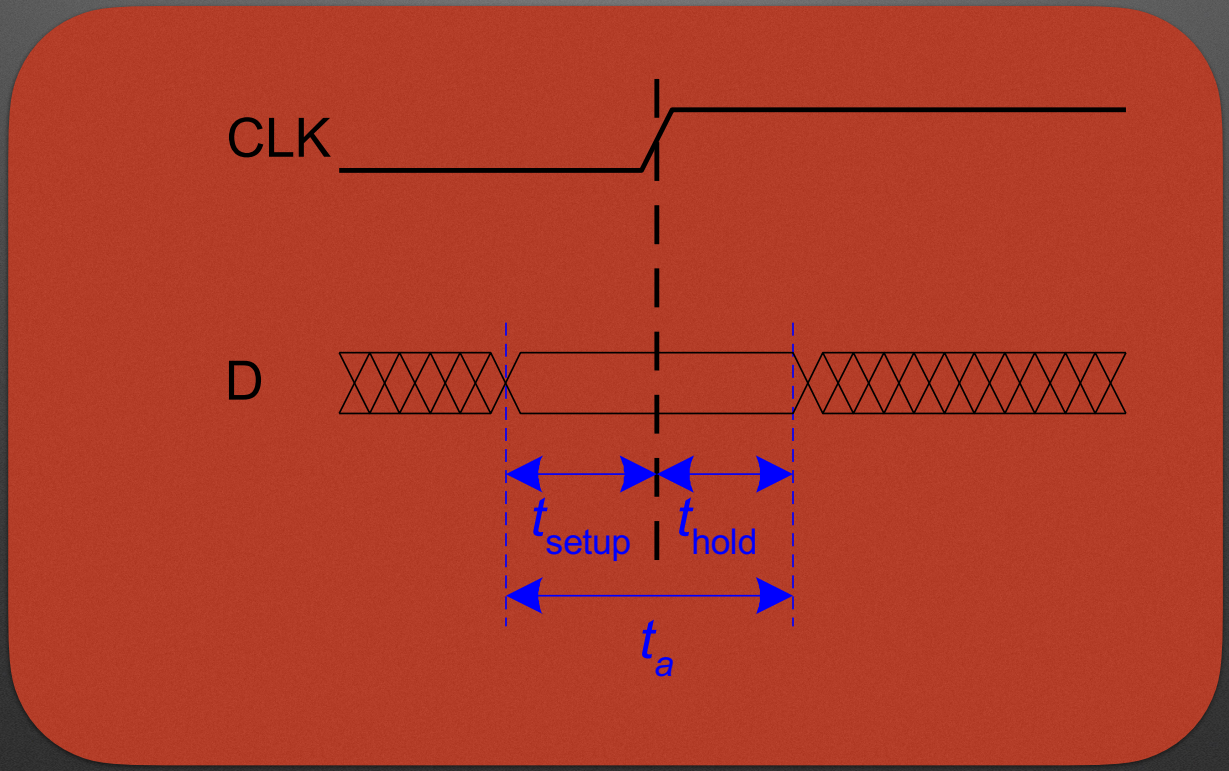


https://www.ni.com/docs/en-US/bundle/ni-hsdio/page/hsdio/fedge_trigger.html

D Flip Flop is built from SR Latches

- Internal latch
  - Fails if pulses too short
  - Unstable if R/S drop at same time / too close
- Setup Time: Time needed before clock to ensure stable capture
- Hold Time: Time After clock edge value must be “held”

Dff: Setup & Hold Time

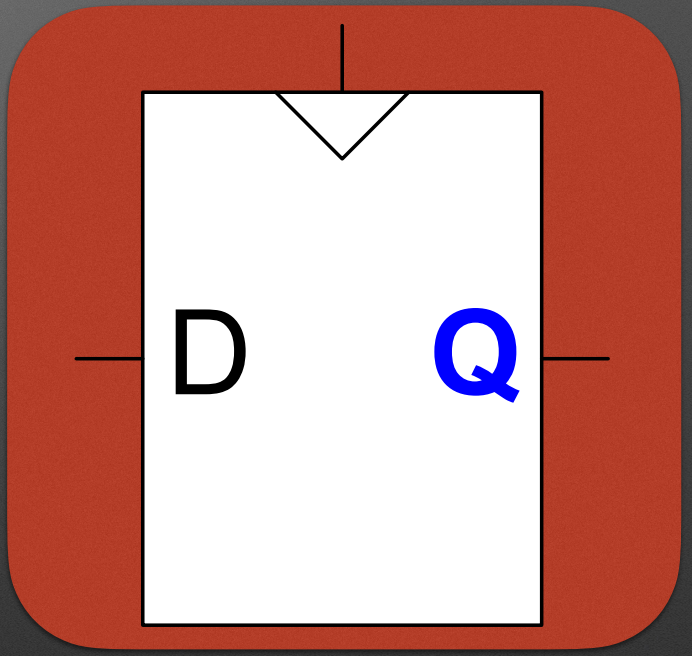


D Flip Flop Time Parameters

- t_{setup}: Time D must be stable before clock
- t_{hold}: Time D must be stable after clock
- t_a: Aperture time ($t_{setup} + t_{hold}$) –
  total window of time D needs to be stable around clock

Dff Time Parameters *Relative to Clock*

- t_{pcq}: Propagation delay from Clock to Q (pcq)
- t_{ccq}: Contamination delay from C to Q (ccq)



Sequential & Synchronous Logic

- *Sequential* circuits
  - Can't be represented with a *simple* table of just inputs and outputs (Possibly a complex table of history of inputs and outputs)
  - Output depends on sequence of inputs and timing
  - *Synchronous* Sequential Circuits
    - Sequential circuits with additional restrictions on form to improve predictability

Synchronous Sequential Circuits

- Are *synchronized* by a common clock
- Uses registers (D Flip-Flops)
- Mix of registers and combinational logic
- All cycles include *at least one register*
- Goal: Impose predictable behavior!

FSM Applications

- Things with modes or sequences of steps. Examples:
  - Washing Machine (fill, agitate, rinse, spin)
  - Stop lights & Traffic control: Green, Yellow, Red
  - Locks: Locked & unlocked
  - Computer programs: Playing game vs. on menu
  - Elevator controls (state = floor)
  - …

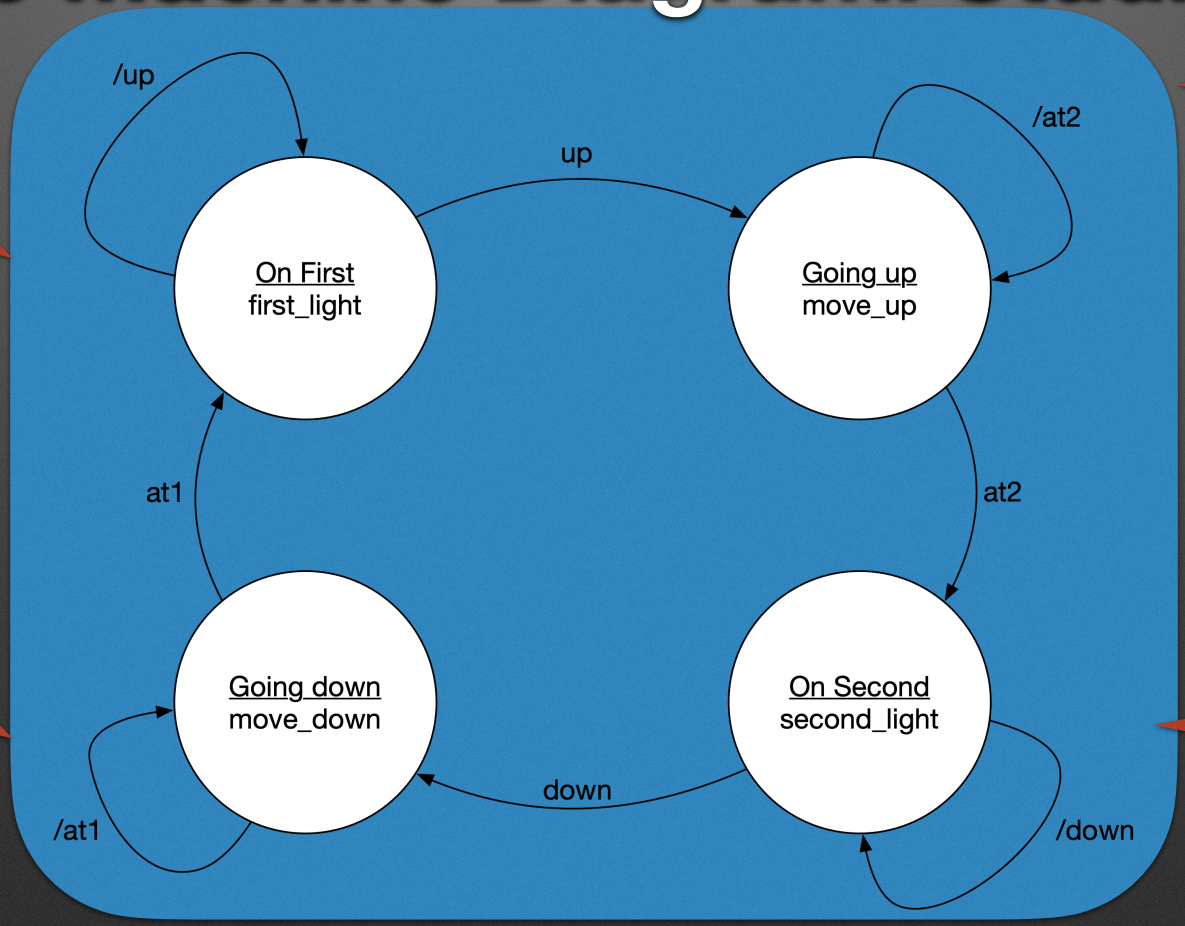
e Machine Diagram: Studic

Assume clock cycle of 10s?

Vs 240s?

What does “clock” do?

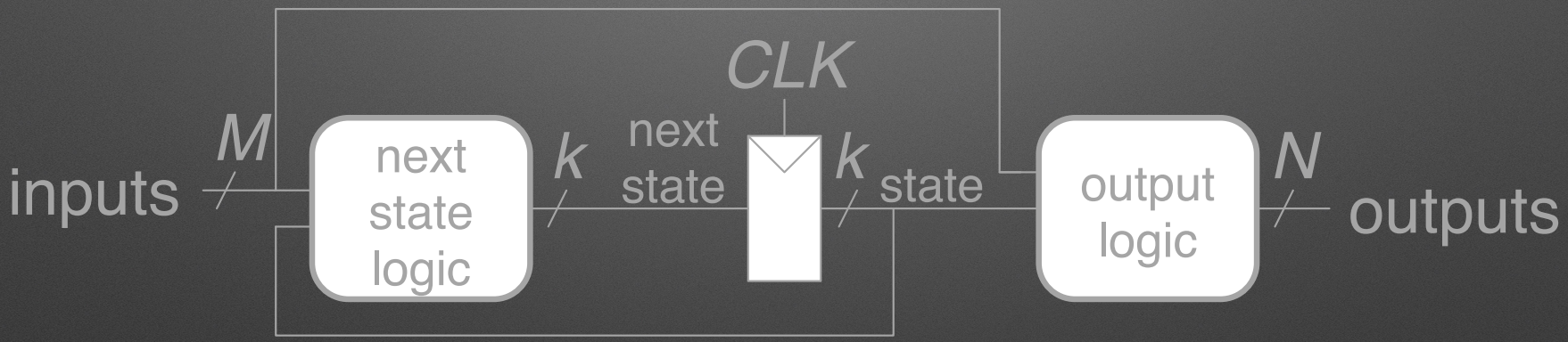
Can the clock be too short?



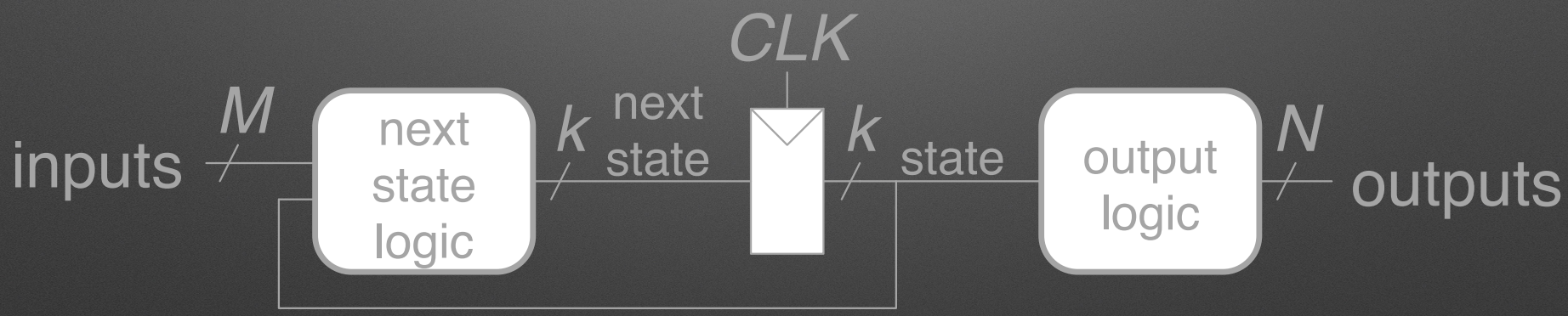
Finite State Machines

- Can be realized with *synchronous* sequential circuits
  - Memory (registers/flip-flops) hold *encoded* state (represents the location or current bubble in diagram)
  - Combinational logic for:
    - Output control (the outputs in the bubbles)
    - Determine next state (the arrows)

FSM: Mealy Machine



FSM: Moore Machine Structure



State Encodings

- “Art”
  - Encoding has impact on equations used
- Major flavors
  - Counting
  - One Hot

Elevator: Examples

STATE NAME	BINARY COUNTING	ONE HOT
ON FIRST		
GOING UP		
ON SECOND		
GOING DOWN		

Elevator: Examples Part 2

STATE NAME	BINARY COUNTING	ONE HOT
ON FIRST		
GOING UP		
ON SECOND		
GOING DOWN		

Decoder

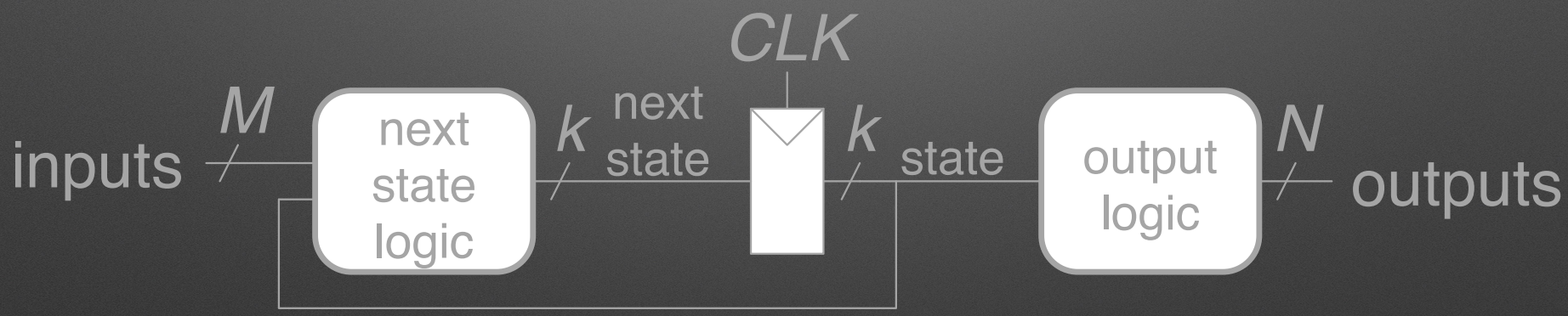
- Decoders…decode
  - n bits of input to 2^n distinct, mutually exclusive outputs
  - It's just ANDs/NOTs to select each distinct binary value

Ex: 2-to-4 Decoder

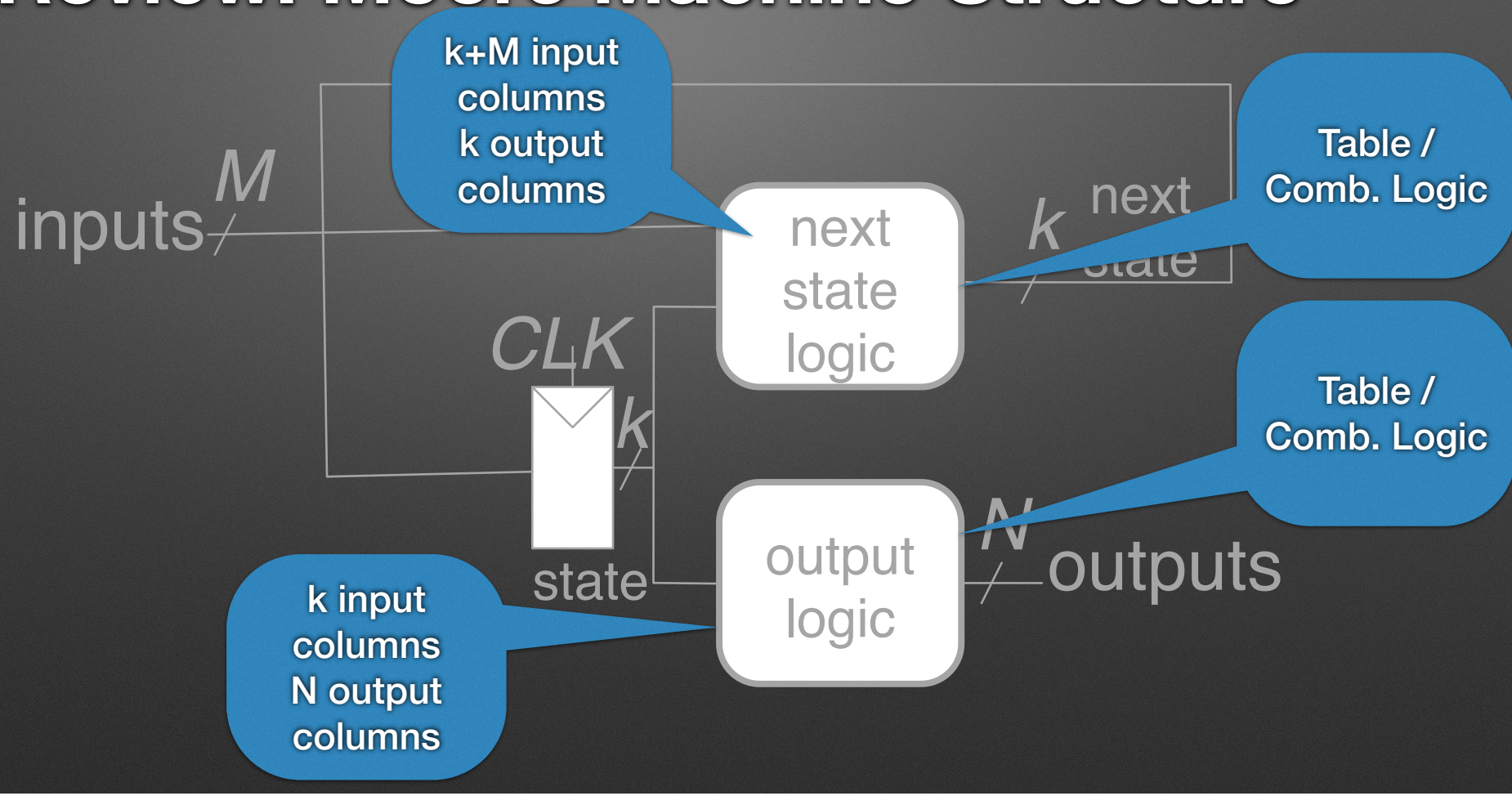
Working Elevator 1

Elevator Rides Again: One Hot

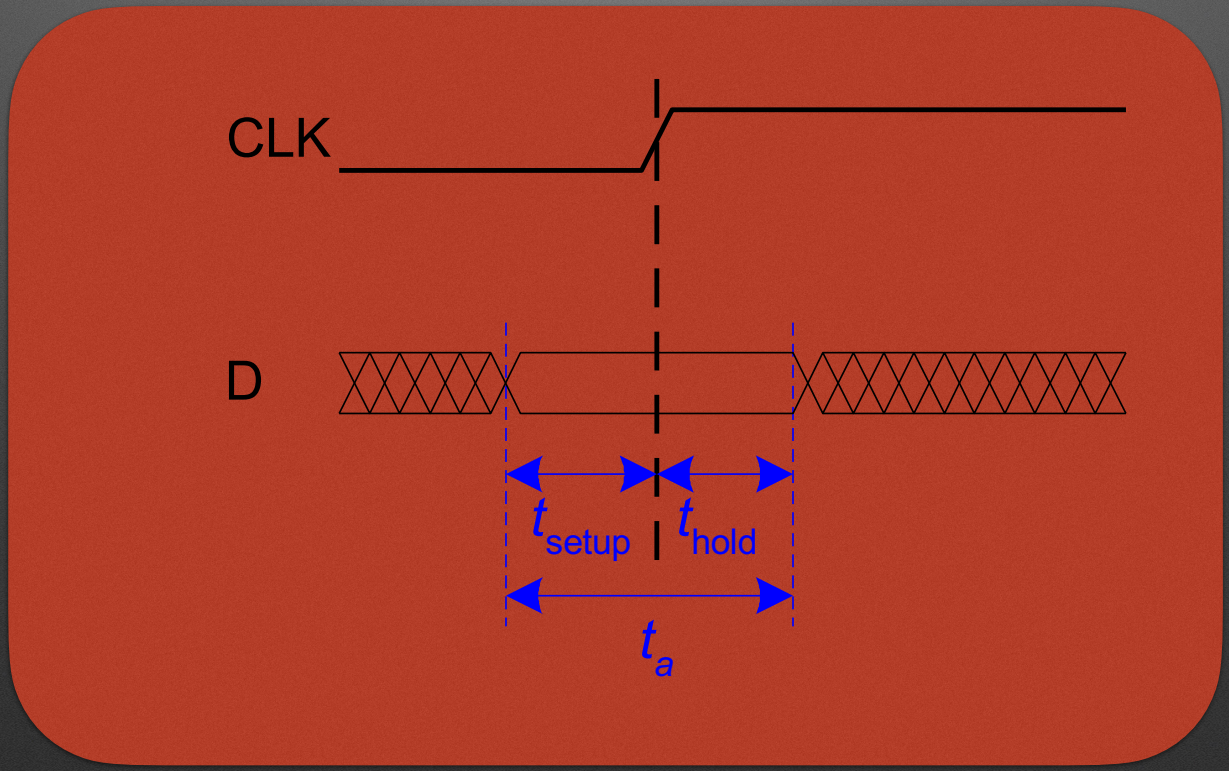
Review: Moore Machine Structure



Review: Moore Machine Structure

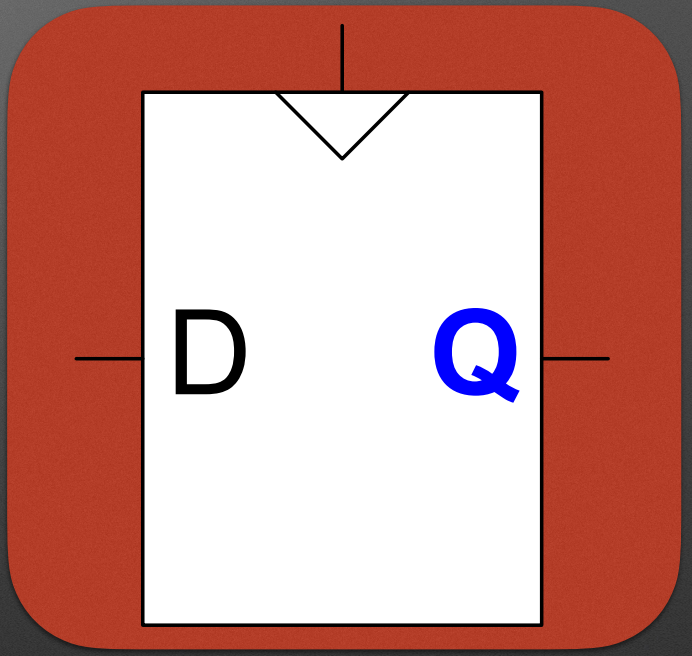


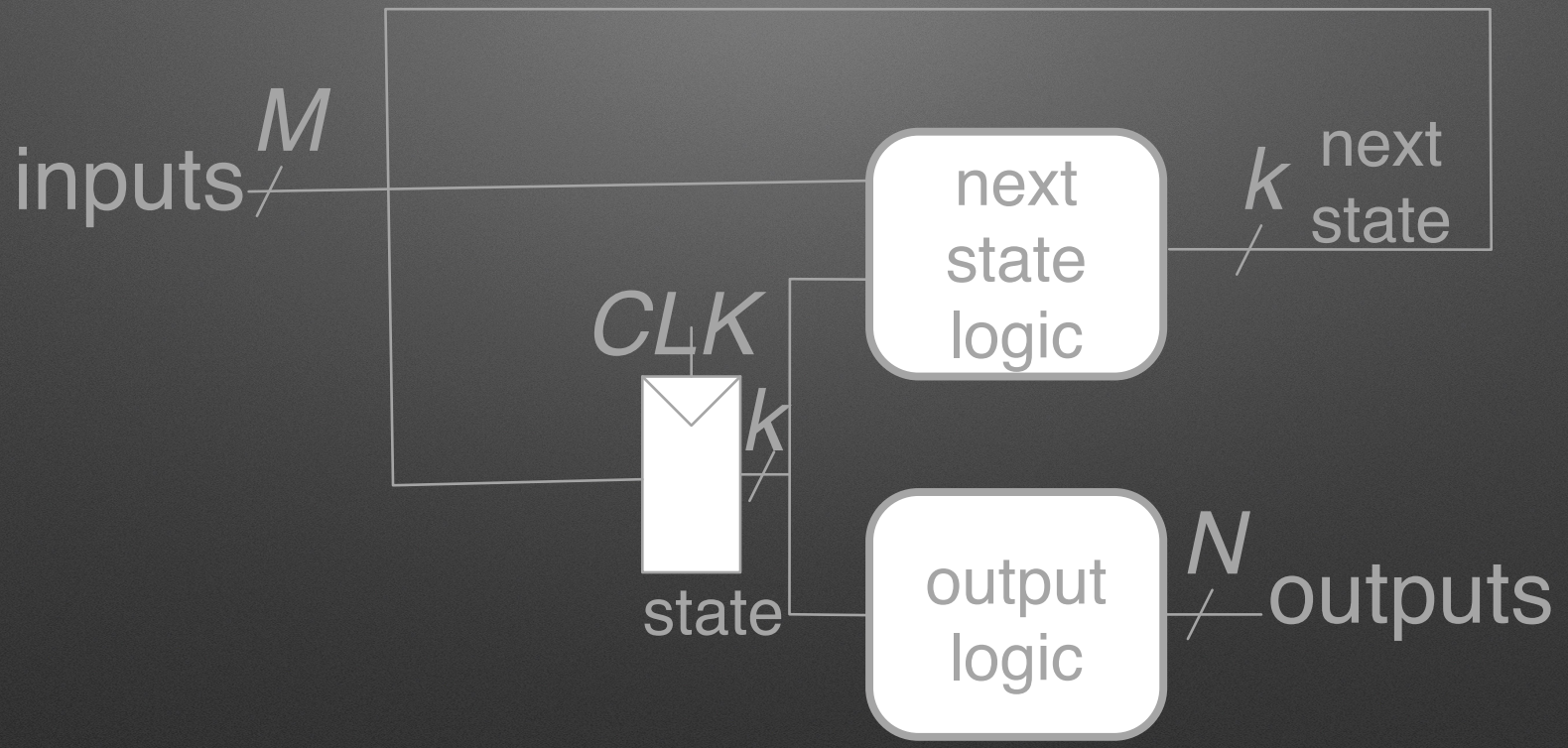
Review: Dff Setup & Hold Time



Dff Time Parameters

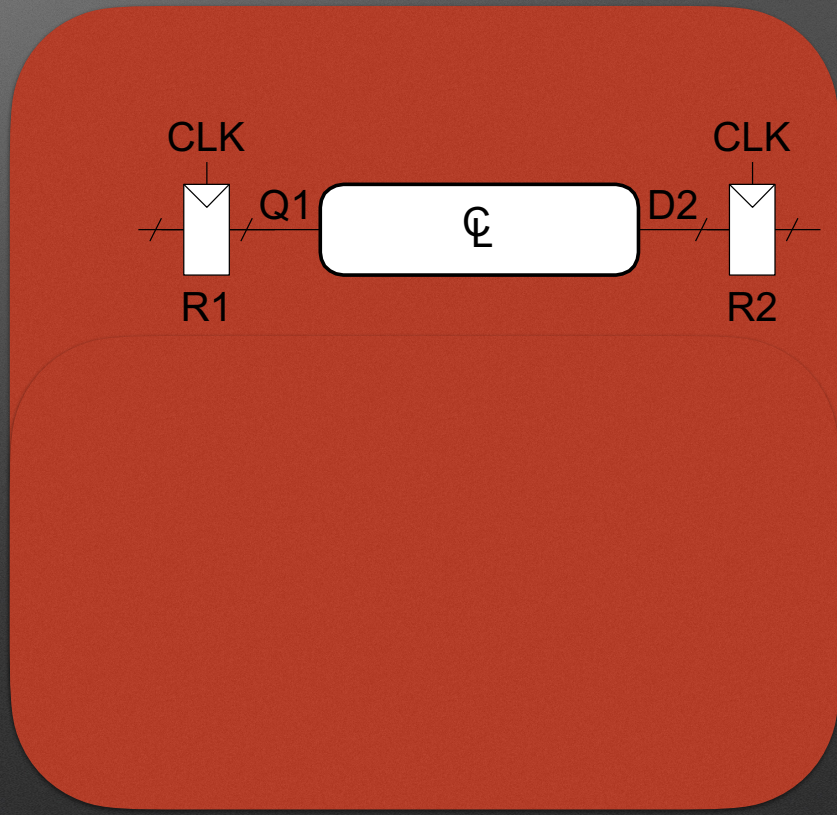
- t_{pcq}: Propagation delay from Clock to Q (pcq)
- t_{ccq}: Contamination delay from C to Q (ccq)




inputs
M
next
state
logic
k
next
state
CLK
k
state
output
logic
N
outputs

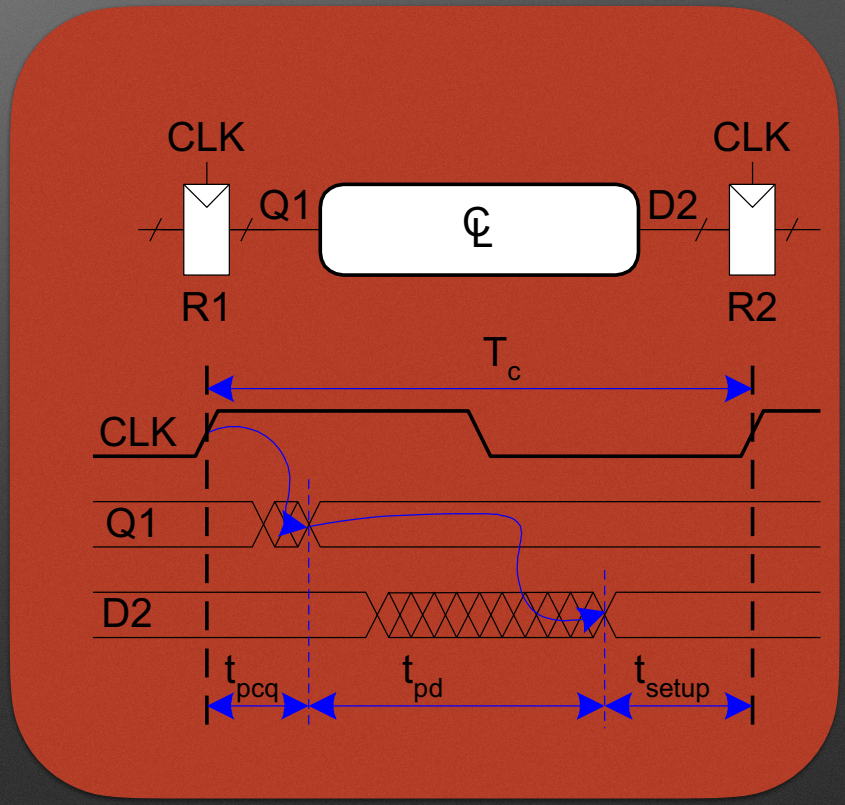
Setup Time Constraint

- Max time from R1 through CL
  - R2's input needs to be stable t_{setup} before clock



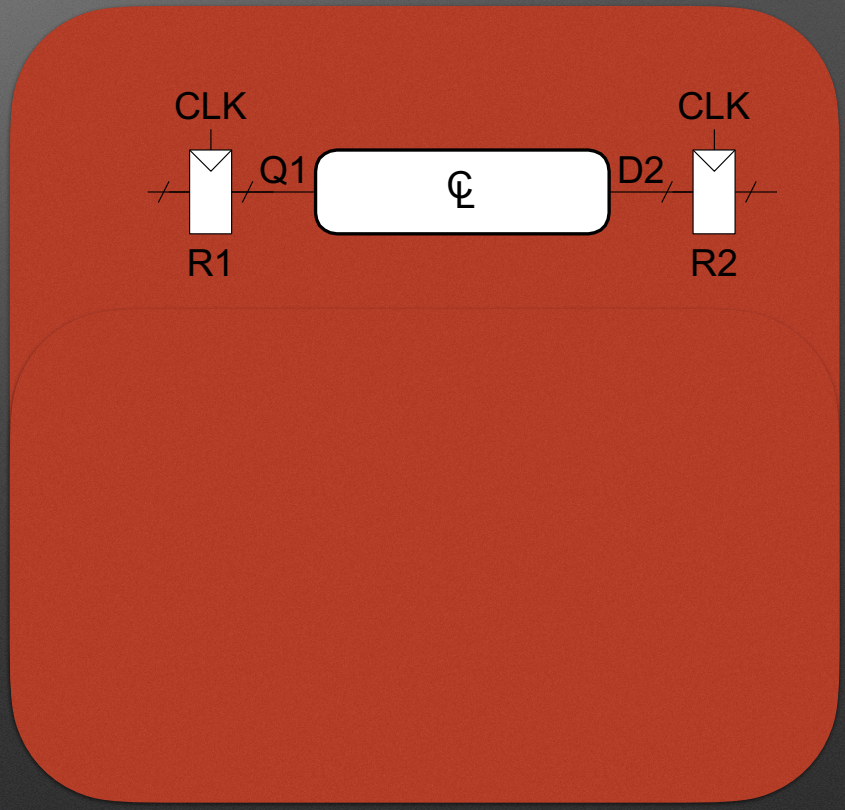
Setup Time Constraint

- Max time from R1 through CL
  - R2's input needs to be stable t_{setup} before clock



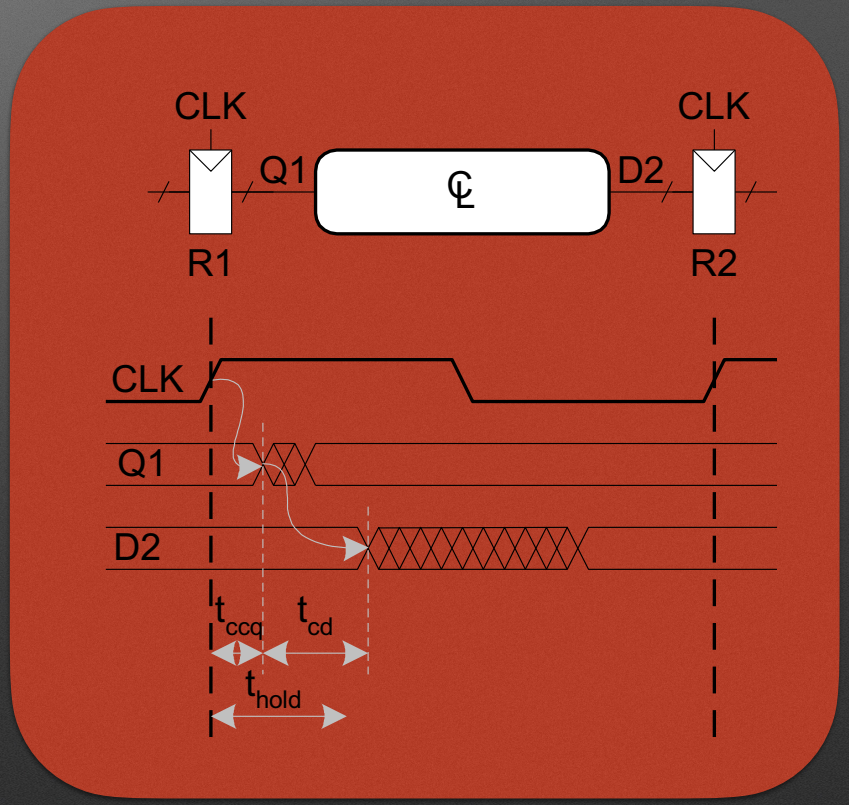
Hold Time Constraint

- Min time from R1 through CL
  - R2's input must be stable t_{hold} after the clock



Hold Time Constraint

- Min time from R1 through CL
  - R2's input must be stable t_{hold} after the clock



Synchronous Timing

- Must meet both
  - Setup Time Constraint
  - Hold Time Constraint

Next Time

- Studio