HLSM & Time Constraints on Sequential Circuits

Prof. Usagi

RTL(Register Transfer Level) Design

RTL Design Process

- Step 1: Capture a high-level state machine
  - Describe the system's desired behavior as a high-level state machine. The state machine consists of states and transitions. The state machine is high level because the transition conditions and the state actions are more than just Boolean operations on single-bit input and outputs
  - Recommendations:
    - Always list all inputs, outputs and local registers on top of your HLSM diagram
    - Clearly specify the size in bits of each of them
    - On states: update the value of registers, update of outputs
    - On transitions: express conditions in terms of the HLSM inputs or state of the internal values and arithmetic operations between them.

RTL Design Process

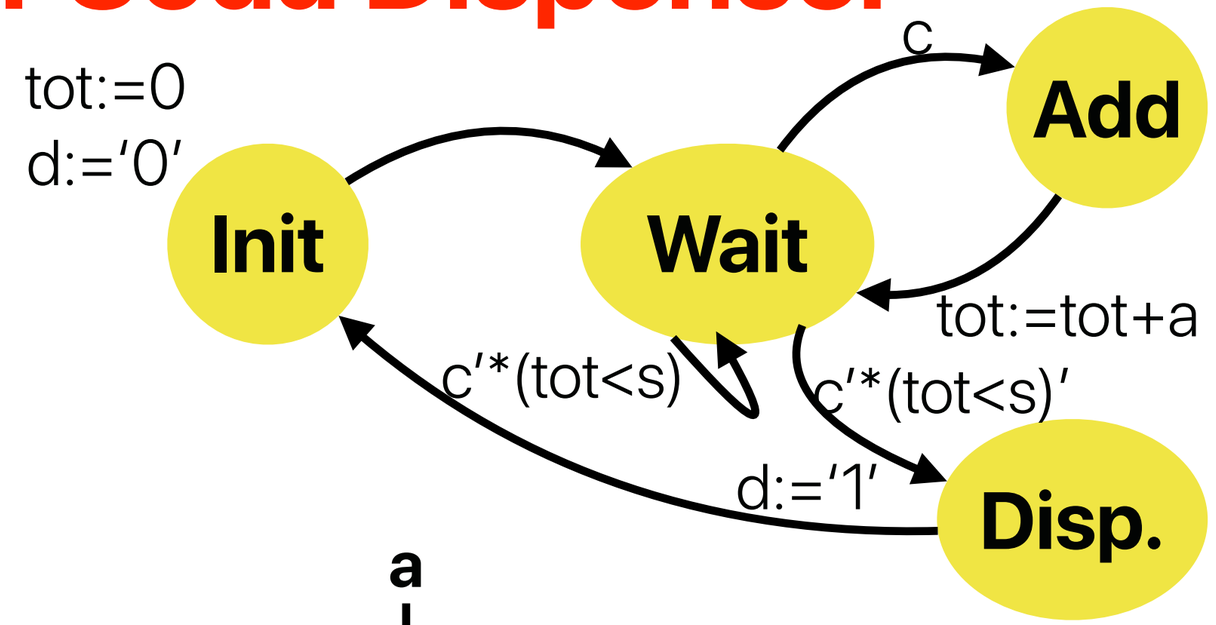
- Step 2: Convert it to a circuit
  - Create a datapath
    - Create a datapath to carry out the data operations of the high level state machine
    - Elements of your datapaths can be registers, adders, comparators, multipliers, dividers, etc.
  - Connect the datapath to a controller
    - Connect the datapath to a controller block.
    - Connect the external control inputs and outputs to the controller block.
    - Clearly label all control signals that are exchanged between the datapath and the controller
  - Derive the controller's FSM
    - Convert the high-level state machine to a finite state machine (FSM) for the controller, by replacing data operations with setting and reading of control signals to and from the datapath
- Final Step Implement the FSM as a state register and logic

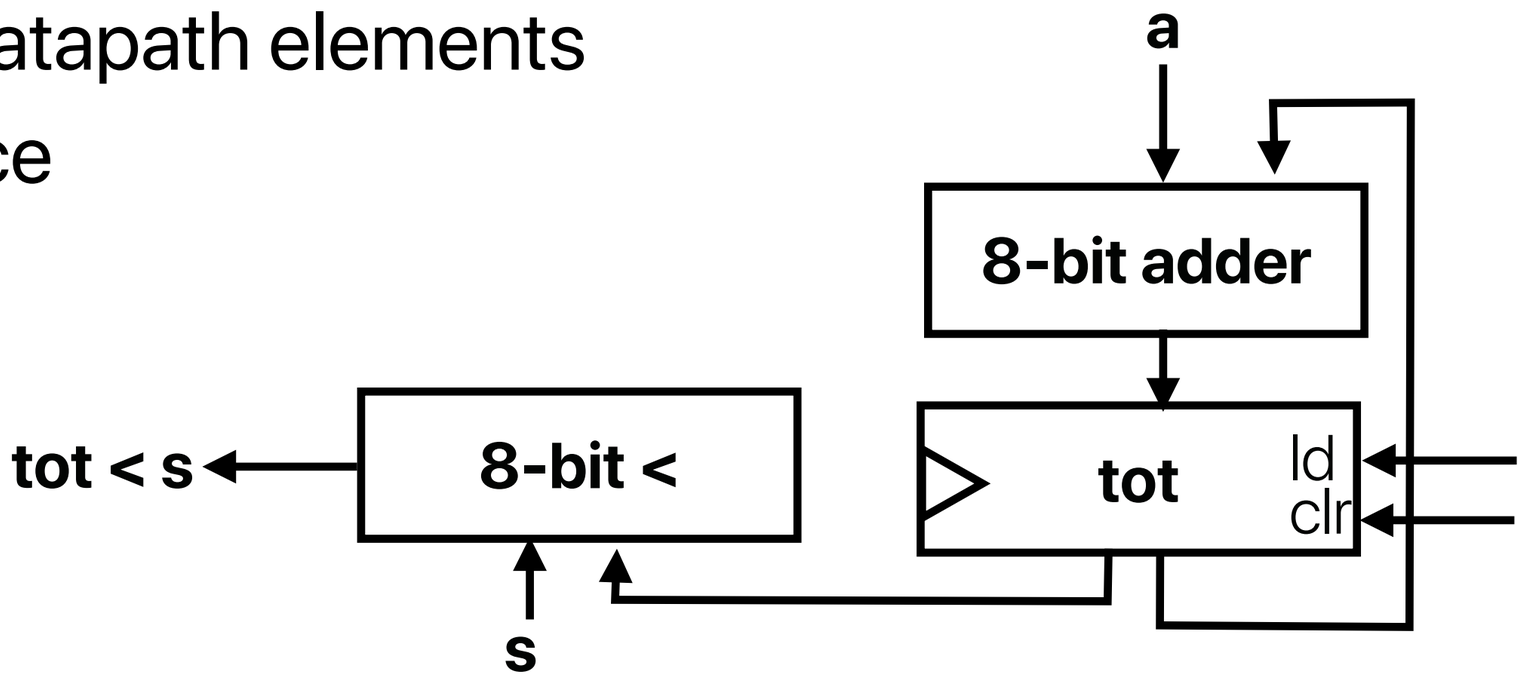
RTL Design Summary

- Capture the behavior with HLSM
- Convertit to a circuit
  - High-level architecture (datapath and control path)
  - Datapath capable of HLSM's data operations
  - Design controller to control the datapath

Create Datapath for Soda Dispenser

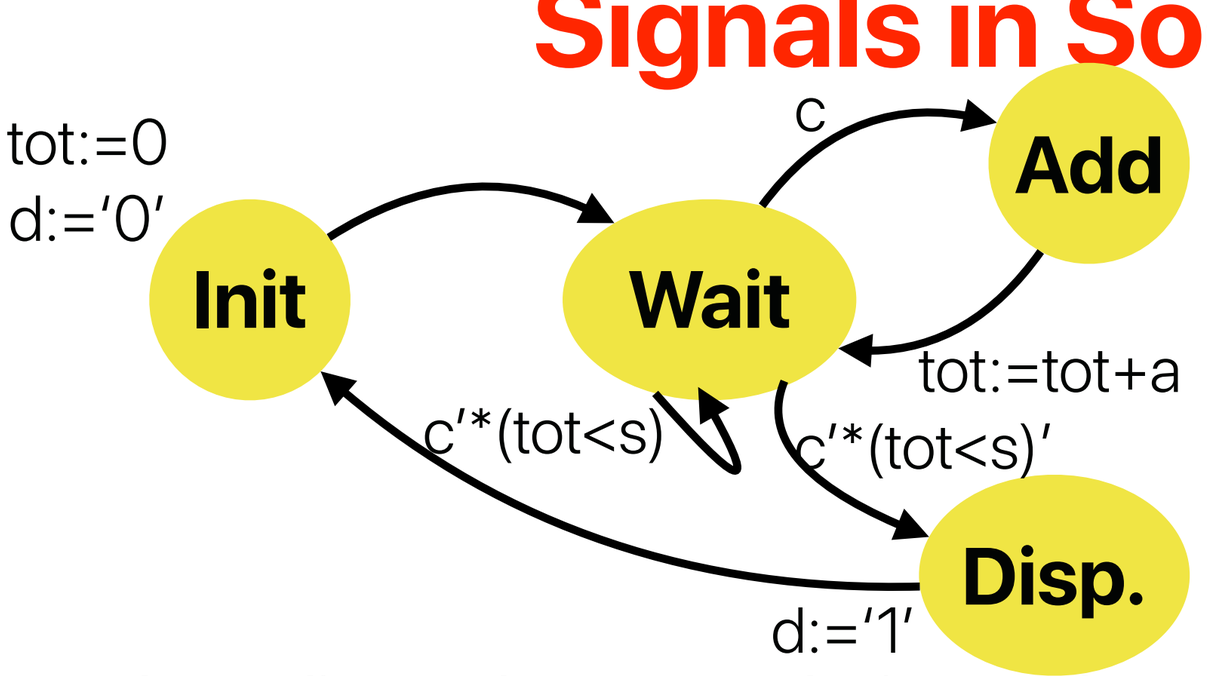
- Register: tot
- Comparator: to compare tot and s
- Adder: to update tot = tot + a
- Connect datapath elements
- I/O interface

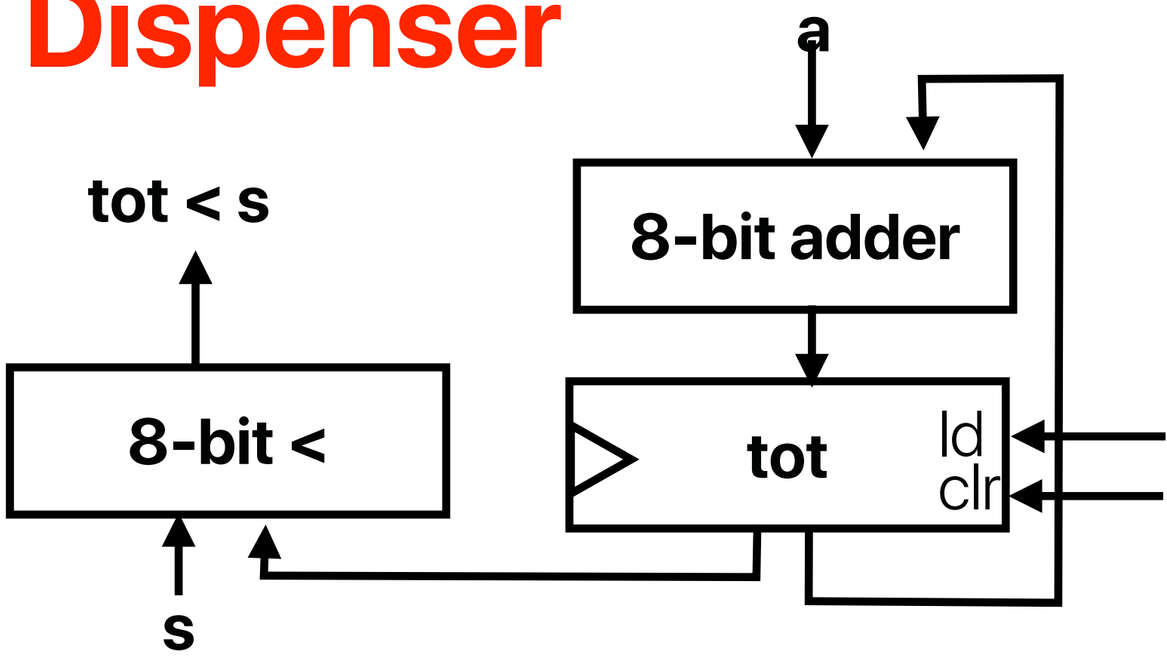




Poll close in 1:30

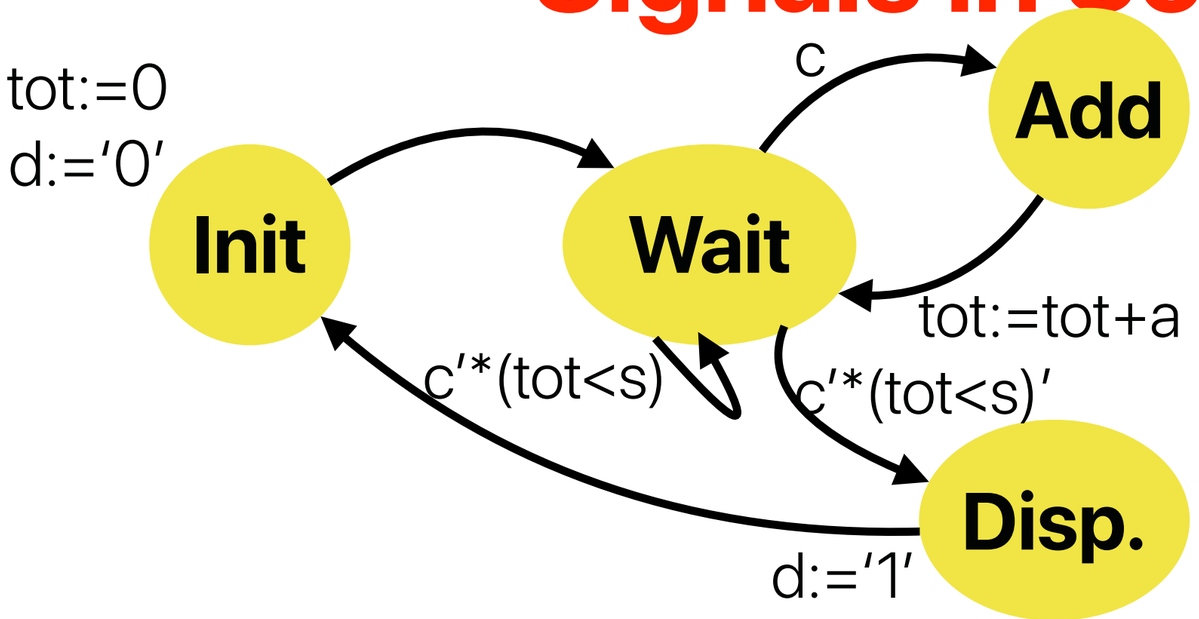
Signals in Soda Dispenser

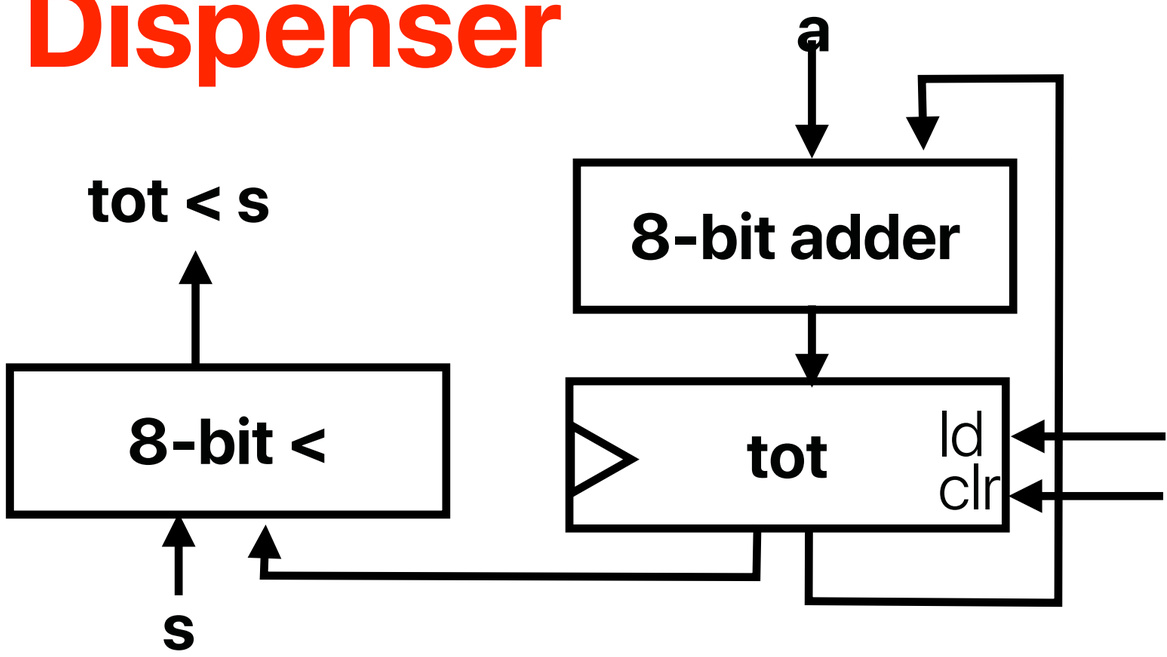




- According to the current design, under which of the following conditions does the register output 'tot' change at the rising clock edge?
  - A. Whenever the value of the coin inserted ('a') changes
  - B. Whenever the cost of the soda ('s') changes
  - C. When the signal tot_ld becomes high
  - D. When the signal tot_clr becomes high
  - E. Both C. & D.

Signals in Soda Dispenser

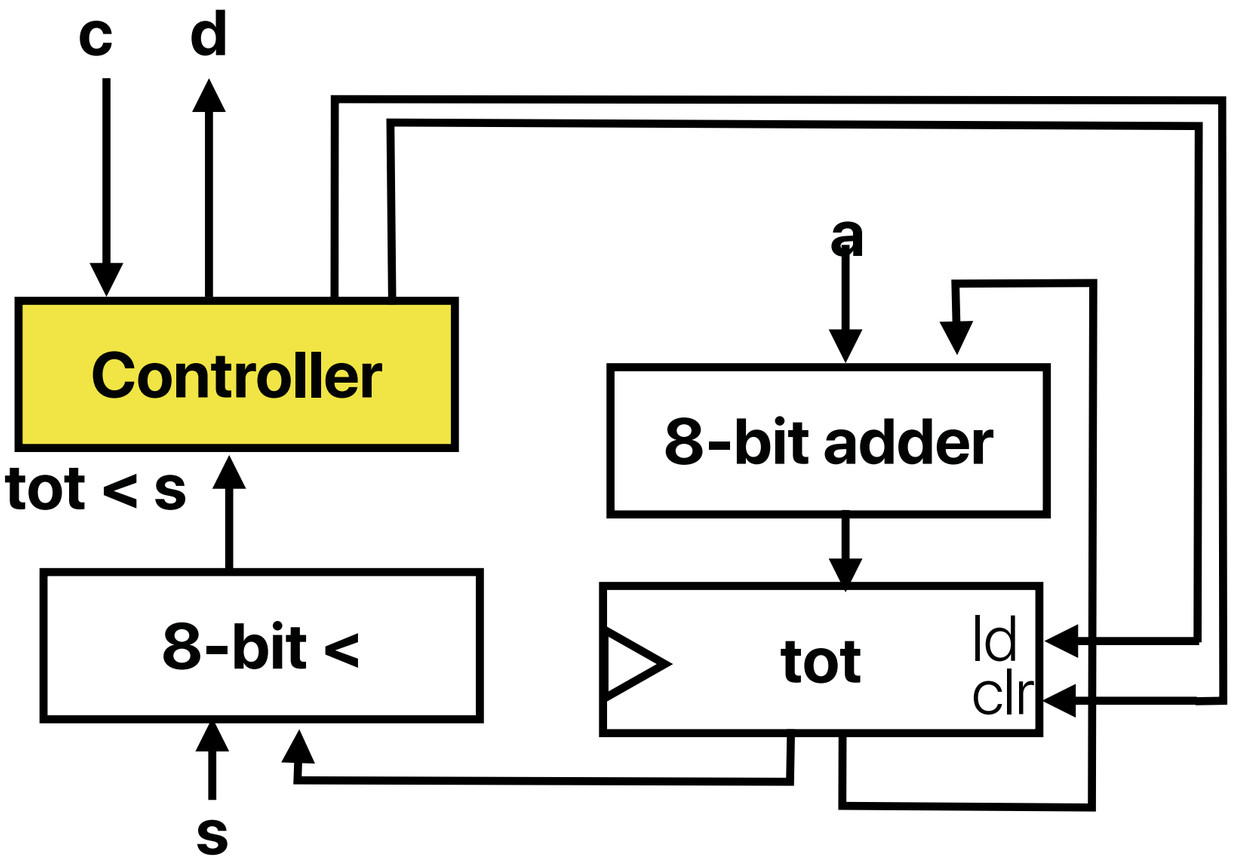




- According to the current design, under which of the following conditions does the register output 'tot' change at the rising clock edge?
  - A. Whenever the value of the coin inserted ('a') changes
  - B. Whenever the cost of the soda ('s') changes
  - C. When the signal tot_ld becomes high
  - D. When the signal tot_clr becomes high
  - E. Both C. & D.

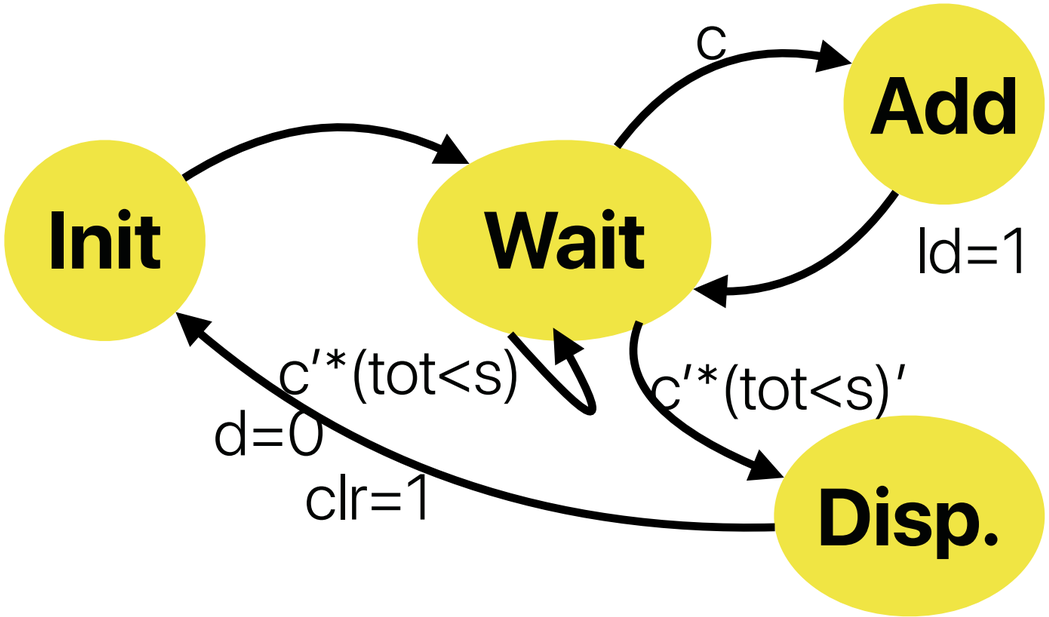
Connect Datapath to a Controller

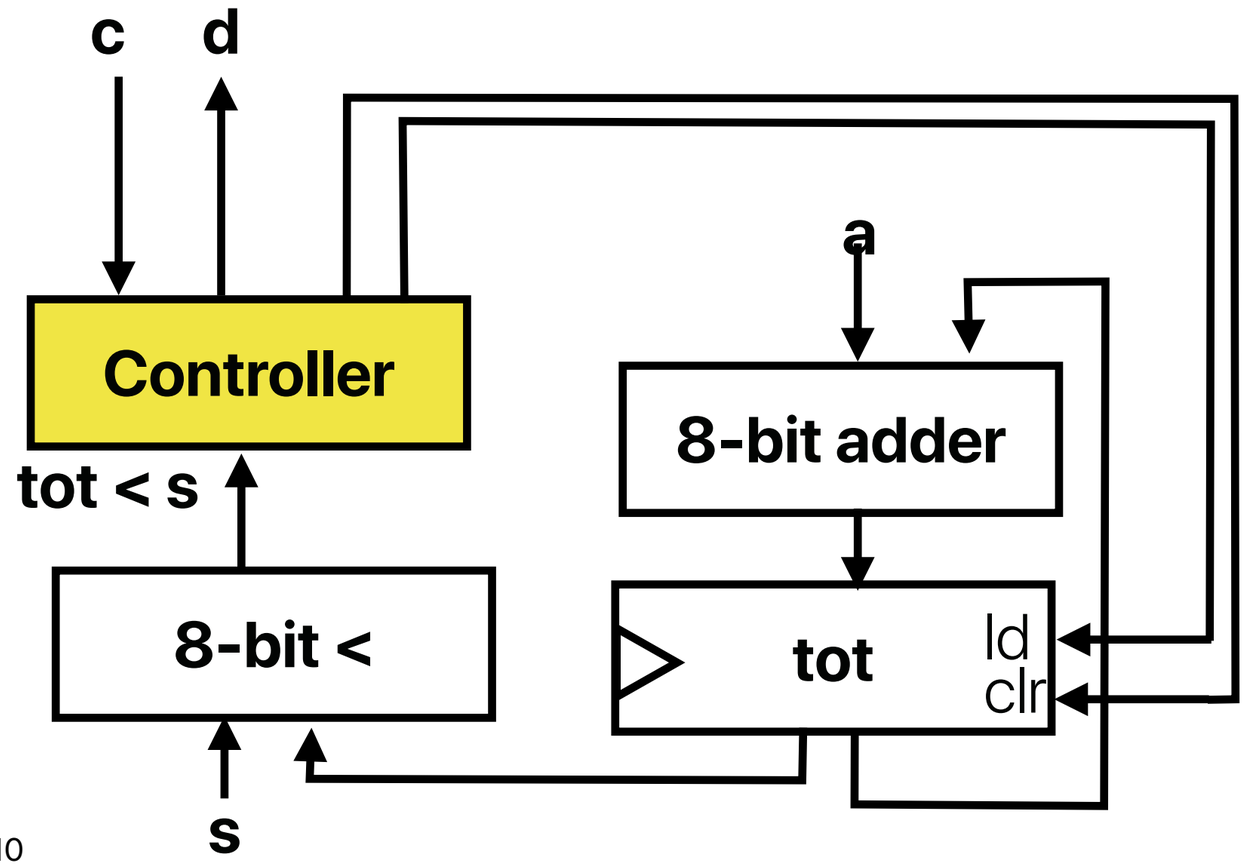
- Controller's inputs
  - External input c (coin detected)
  - Input from datapath comparator's output, which we named tot<s
- Controller's outputs
  - External output d (dispense soda)
  - Outputs to datapathto load and clear the tot register



Derive the Controller's FSM

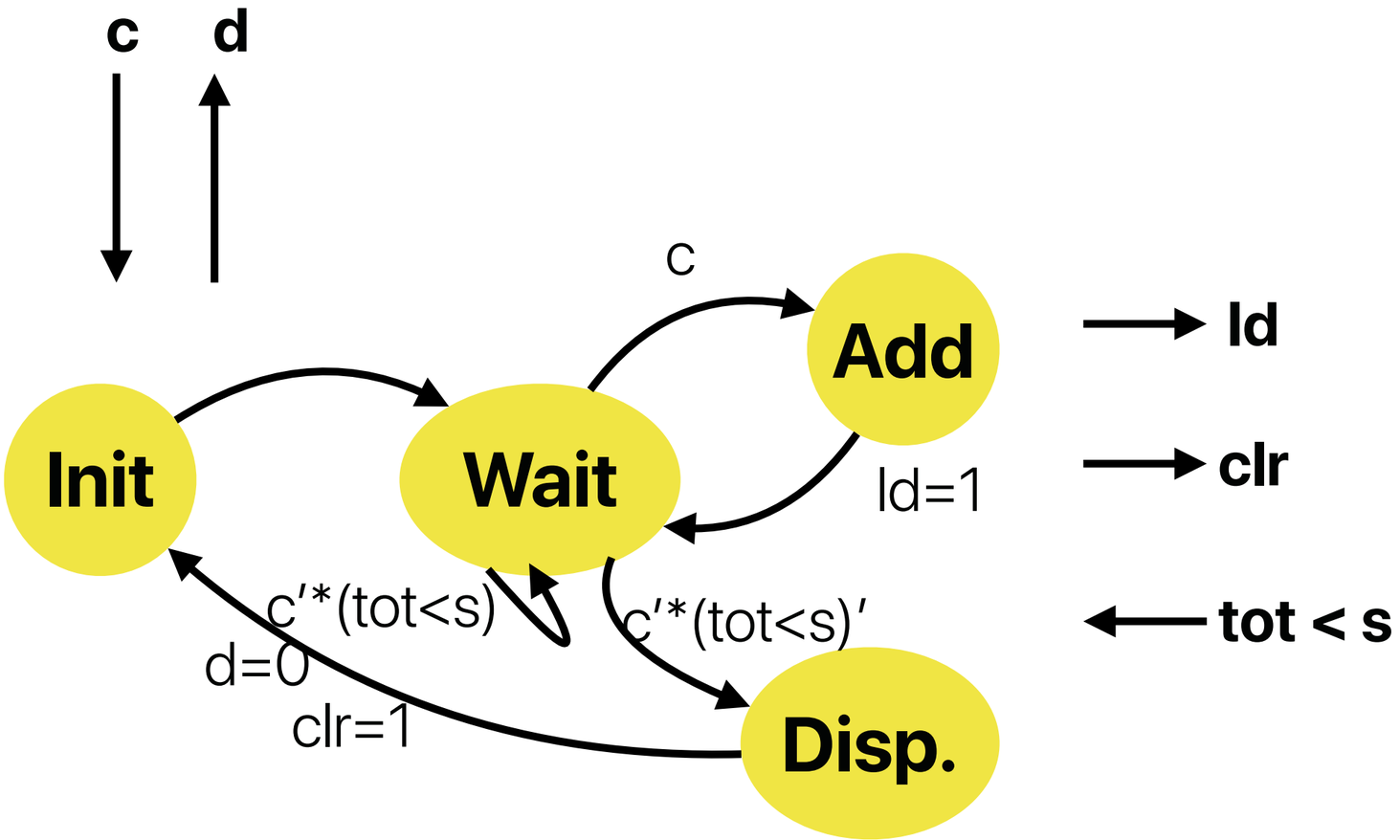
- FSM has the same states and arcs as HLSM
- Replace all references to the data elements in the HLSM with appropriate control signals & values





Final Step: Implement the controller FSM

- Implement the FSM as a state register and logic



Control path of a microprocessor

Let’s put all things together!

- We have learned all datapath components for an ALU!
  - Register
  - Shifter
  - Adders
  - Multiplier
- Processor has only one clock generator
  - Each datapath component has a different latency

Poll close in 1:30

The clock rate of the processor

- If we are designing a processor that supports shift, add, mul operations with the following datapath components, what's the expected processor clock rate?
  - Register — 0.1 ns
  - Shifter — 0.2 ns
  - 32-bit CLA — 1.6 ns
  - 32-bit hierarchical multiplier — 16 ns

  A. ~ 10 GHz

  B. ~ 5 GHz

  C. ~ 500 MHz

  D. ~ 50 MHz

  E. ~ 5 MHz

The clock rate of the processor

- If we are designing a processor that supports shift, add, mul operations with the following datapath components, what's the expected processor clock rate?
  - Register — 0.1 ns
  - Shifter — 0.2 ns
  - 32-bit CLA — 1.6 ns
  - 32-bit hierarchical multiplier — 16 ns

A. ~ 10 GHz

B. ~ 5 GHz

C. ~ 500 MHz

D. ~ 50 MHz

E. ~ 5 MHz

— The slowest component bottlenecks the processor performance

Let’s put all things together!

- We have learned all datapath components for an ALU!
  - Register
  - Shifter
  - Adders
  - Multiplier
- Processor has only one clock generator
  - Each datapath component has a different latency
  - We have make some of the above “serial”

Poll close in 1:30

The clock rate of the processor (2)

- If we are designing a processor that supports shift, add, mul operations with the following datapath components, what's the expected processor clock rate?
  - Register — 0.1 ns
  - Shifter — 0.2 ns
  - Serial 4-bit CLA — 0.3 ns
  - Serial 32-bit shift-and-add multiplier — 0.9 ns

A. ~ 10 GHz
B. ~ 5 GHz
C. ~ 1 GHz
D. ~ 500 MHz
E. ~ 50 MHz

The clock rate of the processor (2)

- If we are designing a processor that supports shift, add, mul operations with the following datapath components, what's the expected processor clock rate?
  - Register — 0.1 ns
  - Shifter — 0.2 ns
  - Serial 4-bit CLA — 0.3 ns
  - Serial 32-bit shift-and-add multiplier — 0.9 ns

A. ~ 10 GHz

B. ~ 5 GHz

C. ~ 1 GHz

D. ~ 500 MHz

E. ~ 50 MHz

Poll close in 1:30

The clock rate of the processor

- If we are designing a processor that supports shift, add, mul operations with the following datapath components, what's the expected processor clock rate?
  - Register — 0.1 ns
  - Shifter — 0.2 ns
  - Serial 8-bit CLA — 0.4 ns
  - Serial 32-bit shift-and-add multiplier — 0.9 ns

  A. ~ 10 GHz
  B. ~ 5 GHz
  C. ~ 1 GHz
  D. ~ 500 MHz
  E. ~ 50 MHz

The clock rate of the processor

- If we are designing a processor that supports shift, add, mul operations with the following datapath components, what's the expected processor clock rate?
  - Register — 0.1 ns
  - Shifter — 0.2 ns
  - Serial 8-bit CLA — 0.4 ns
  - Serial 32-bit shift-and-add multiplier — 0.9 ns

A. ~ 10 GHz

B. ~ 5 GHz

C. ~ 1 GHz

D. ~ 500 MHz

E. ~ 50 MHz

Poll close in 1:30

How many cycles for an add?

- If we are designing a processor that supports shift, add, mul operations with the following datapath components and the clock rate is set to 1GHz, what's the expected number of cycles we need for a 32-bit add operation?
  - Register — 0.1 ns
  - Shifter — 0.2 ns
  - Serial 8-bit CLA — 0.4 ns
  - Serial 32-bit shift-and-add multiplier — 0.9 ns

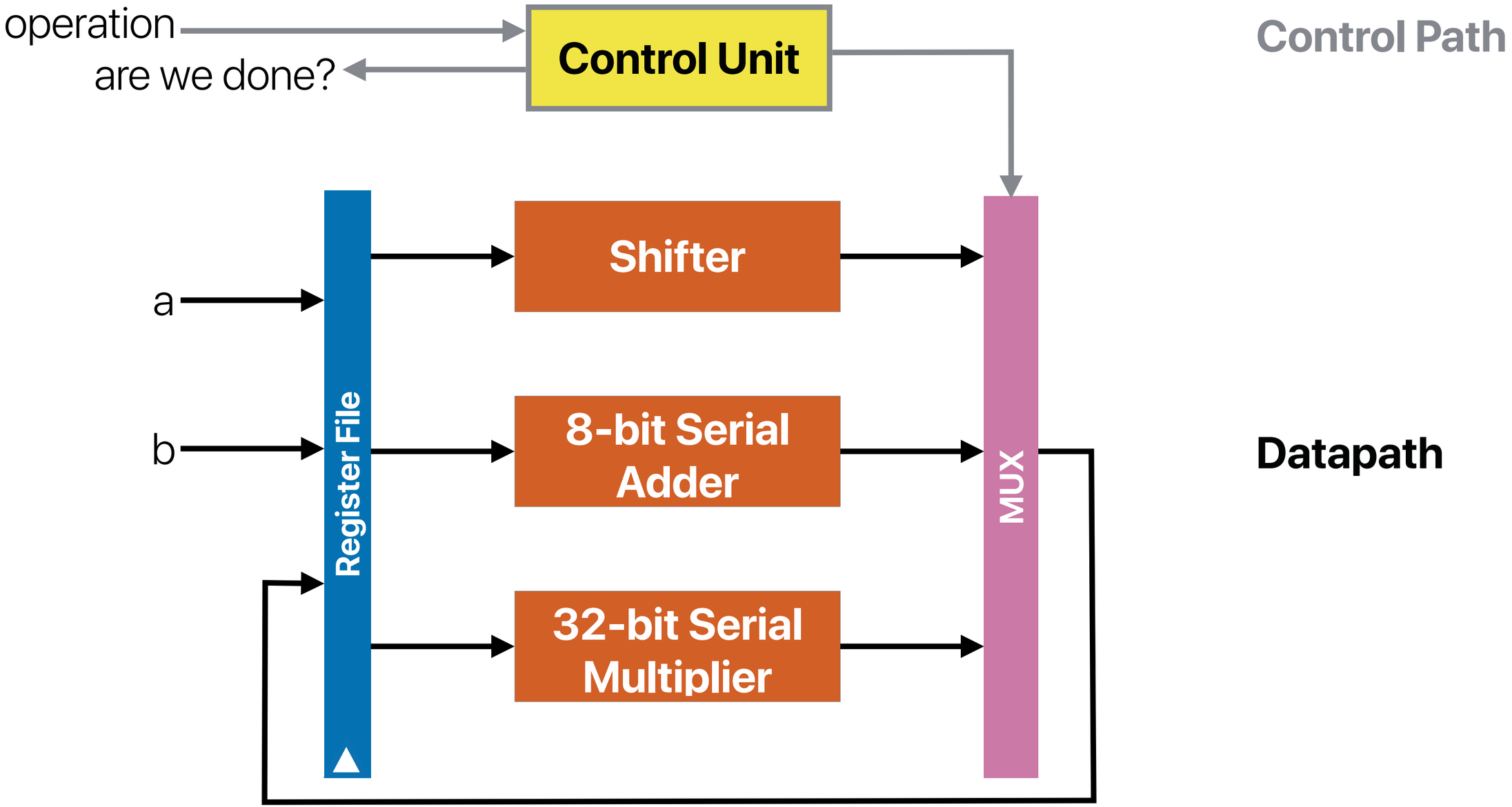
  A. 1
  B. 2
  C. 4
  D. 8
  E. 16

How many cycles for an add?

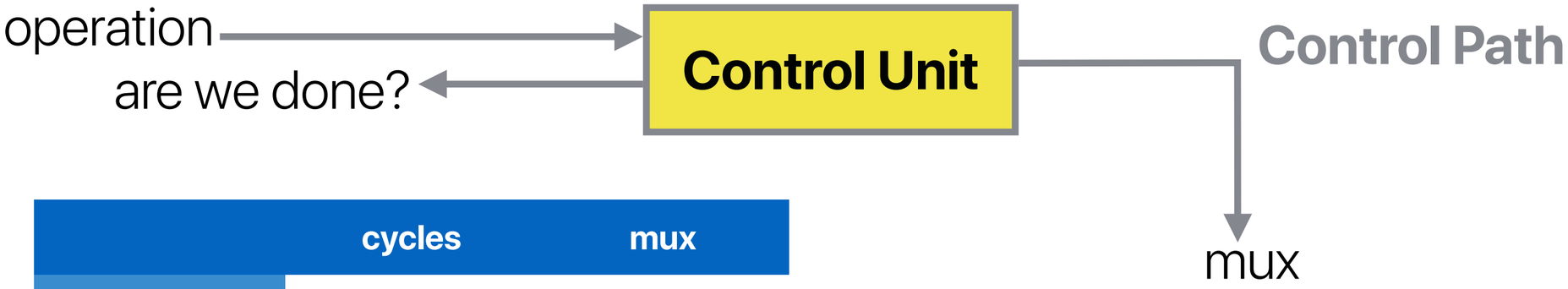
- If we are designing a processor that supports shift, add, mul operations with the following datapath components and the clock rate is set to 1GHz, what's the expected number of cycles we need for a 32-bit add operation?
  - Register — 0.1 ns
  - Shifter — 0.2 ns
  - Serial 8-bit CLA — 0.4 ns
  - Serial 32-bit shift-and-add multiplier — 0.9 ns

A. 1
B. 2
C. 4
D. 8
E. 16

The HLSM for the processor's control

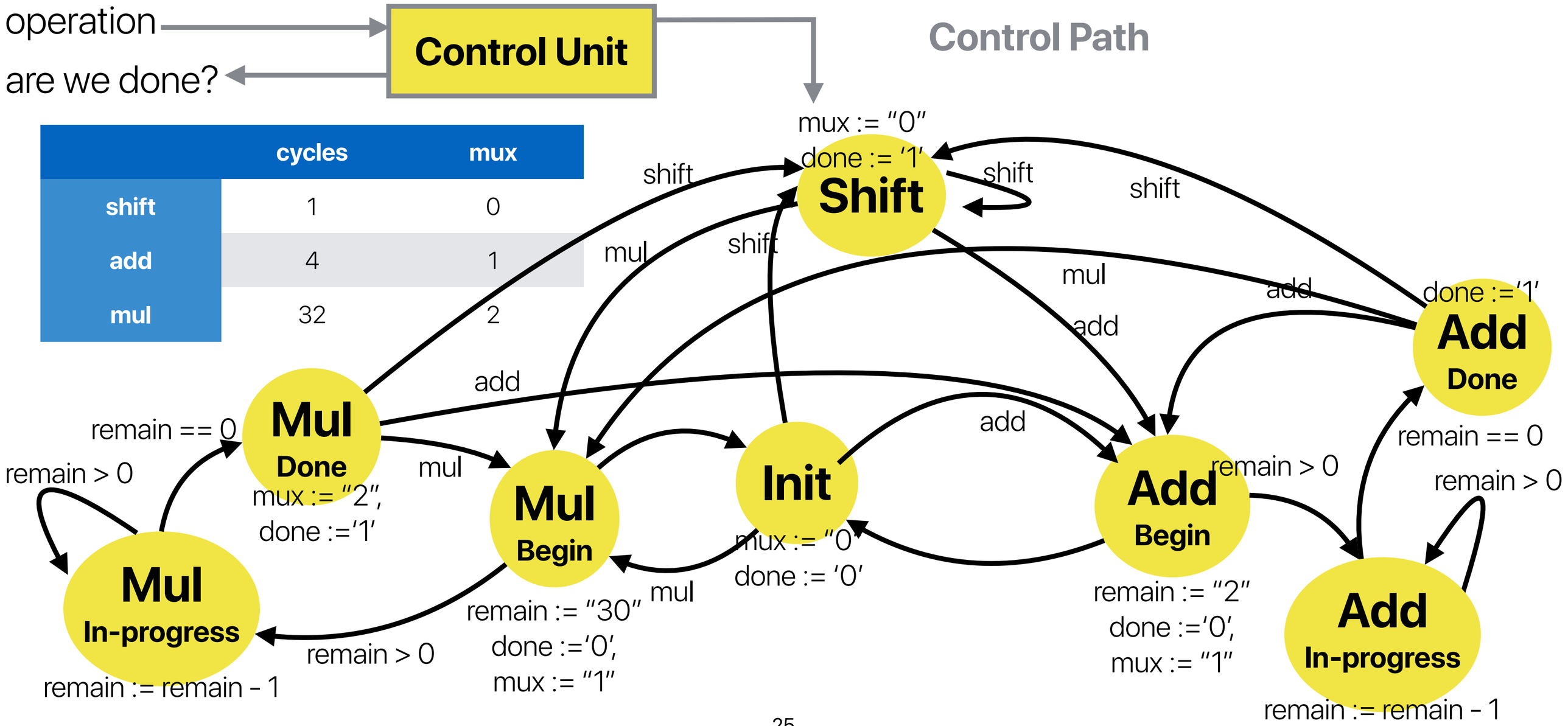


The HLSM for the processor's control

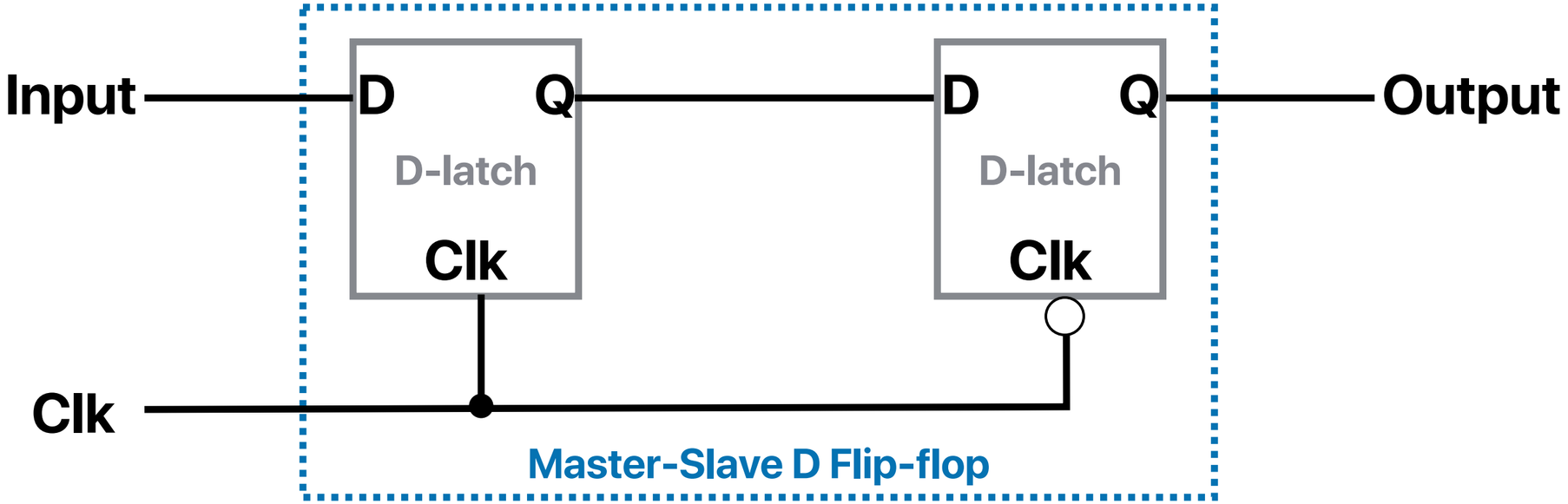


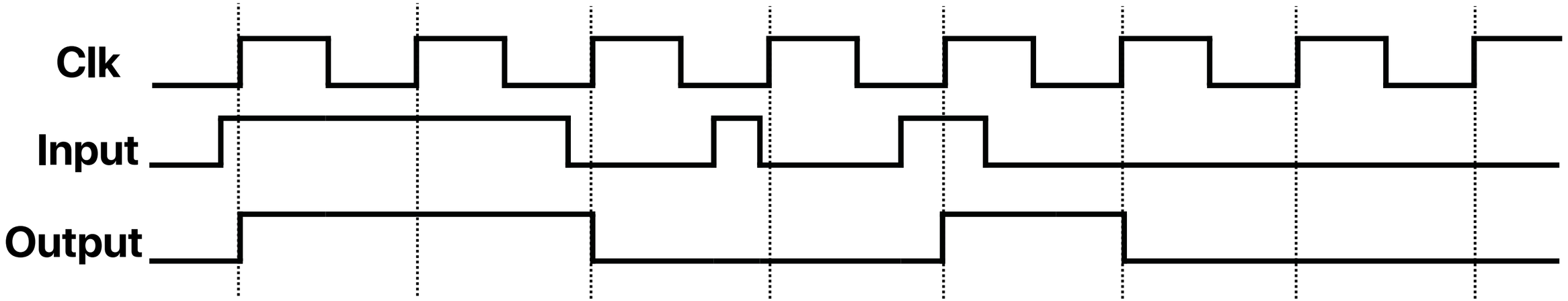
	cycles	mux
shift	1	0
add	4	1
mul	32	2

The HLSM for the processor's control



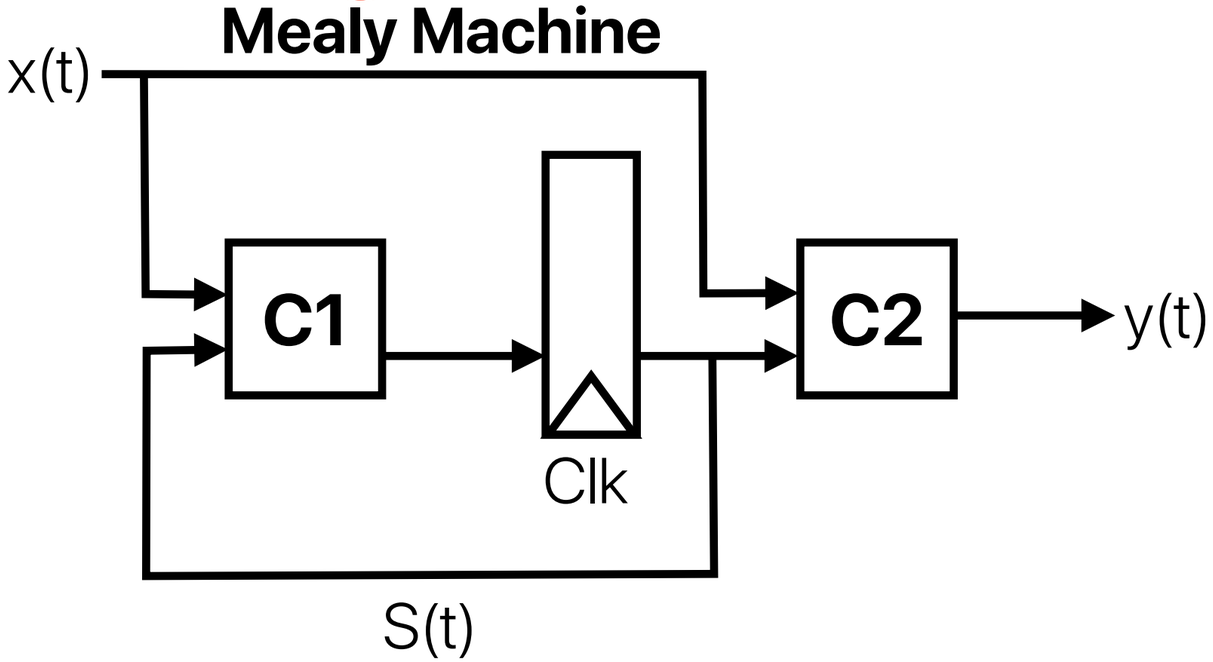
Recap: D flip-flop

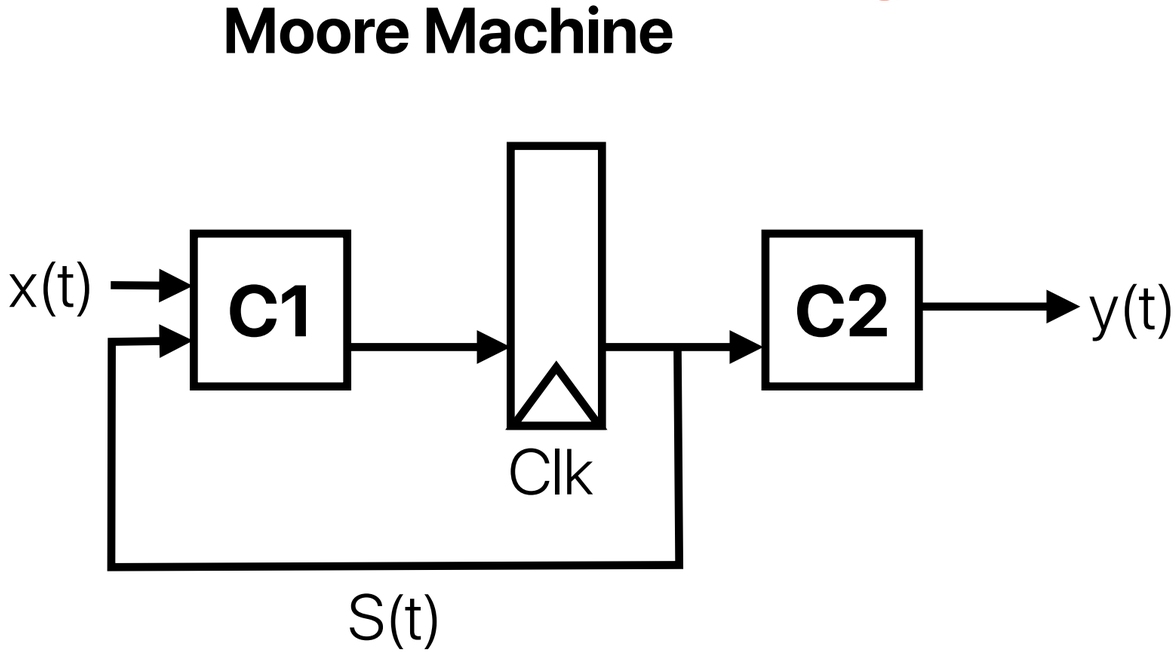




Timing Constraints on Sequential Logic

Timing Constraints in Sequential Circuit Designs





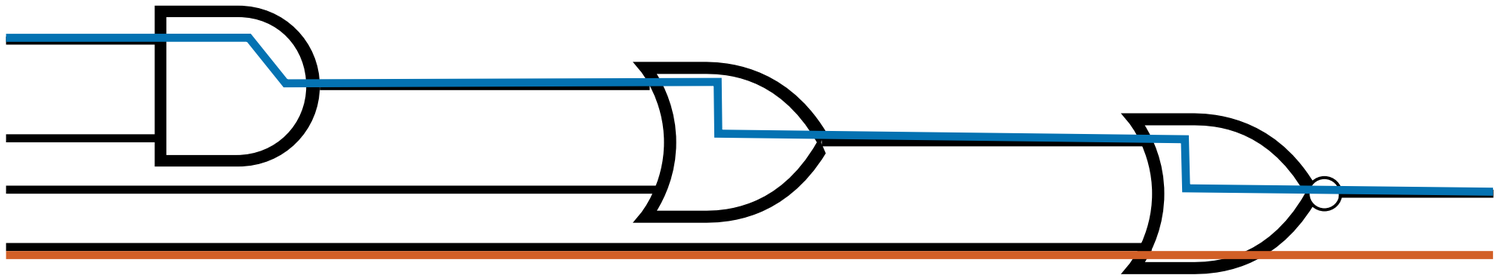
- A seemingly logically correct design can go wrong – signals don't travel in zero time
- We next look at timing constraints for combinational and sequential logic.

Combinational Logic Timing

- Min delay of a gate, also called contamination delay: t_{cd}
  - Minimum time from when an input changes until the output starts to change
- Max delay of a gate, also called propagation delay: t_{pd}
  - Maximum time from when an input changes until the output is guaranteed to reach its final value (i.e., stop changing)

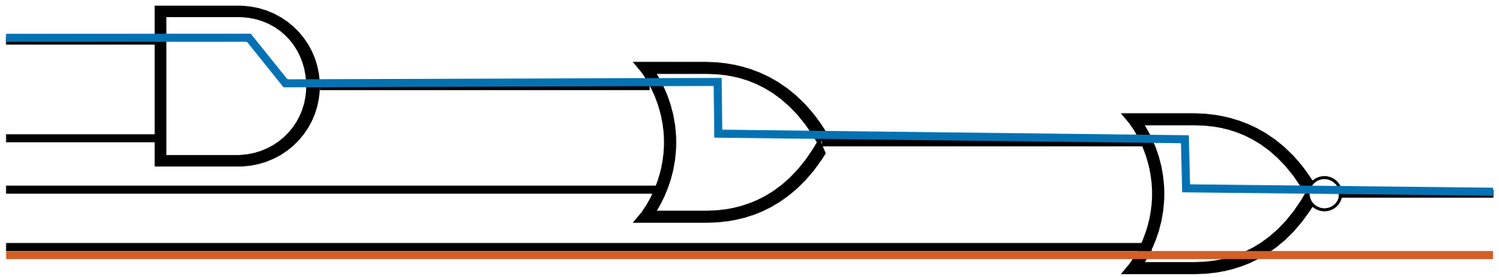
Poll close in 1:30

Combinational Logic: Output Timing Constraints



- Which path in the above circuit determines the **contamination** delay of the circuit (assuming the delay of all the gates is the same)?
  A. Blue path
  B. Red path
  C. Both
  D. Neither

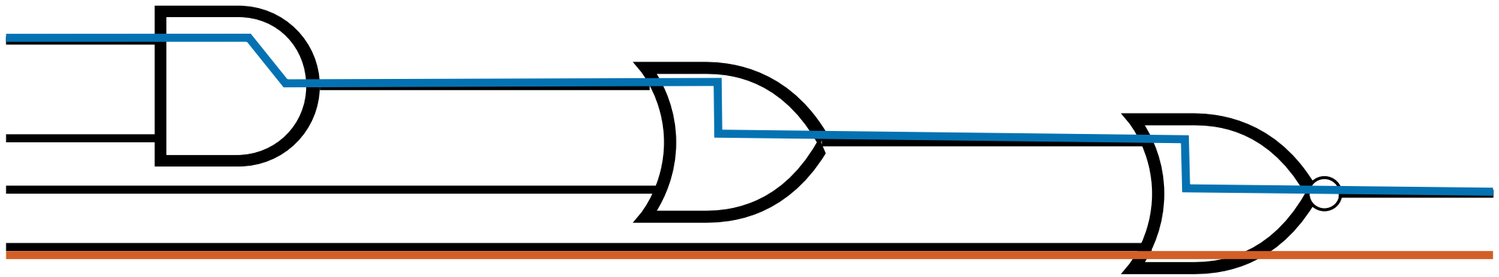
Combinational Logic: Output Timing Constraints



- Which path in the above circuit determines the **contamination** delay of the circuit (assuming the delay of all the gates is the same)?
  A. Blue path
  B. Red path
  C. Both
  D. Neither

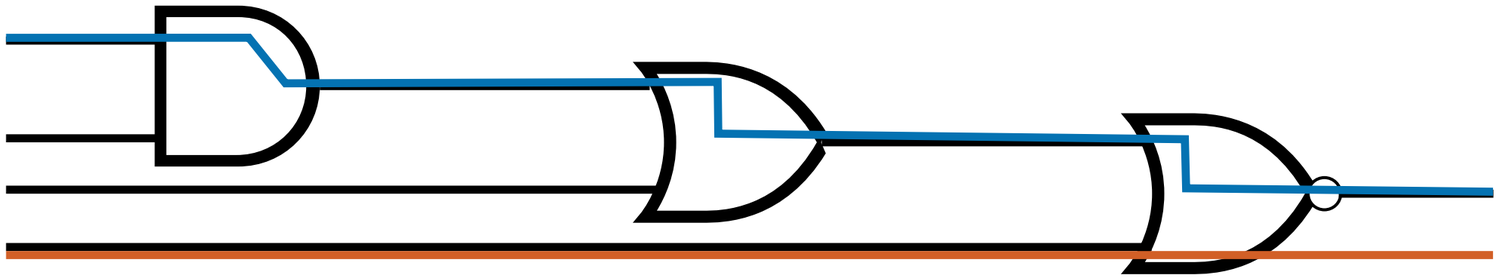
Poll close in 1:30

Combinational Logic: Output Timing Constraints



- Which path in the above circuit determines the **propagation** delay of the circuit (assuming the delay of all the gates is the same)?
  A. Blue path
  B. Red path
  C. Both
  D. Neither

Combinational Logic: Output Timing Constraints



- Which path in the above circuit determines the **propagation** delay of the circuit (assuming the delay of all the gates is the same)?

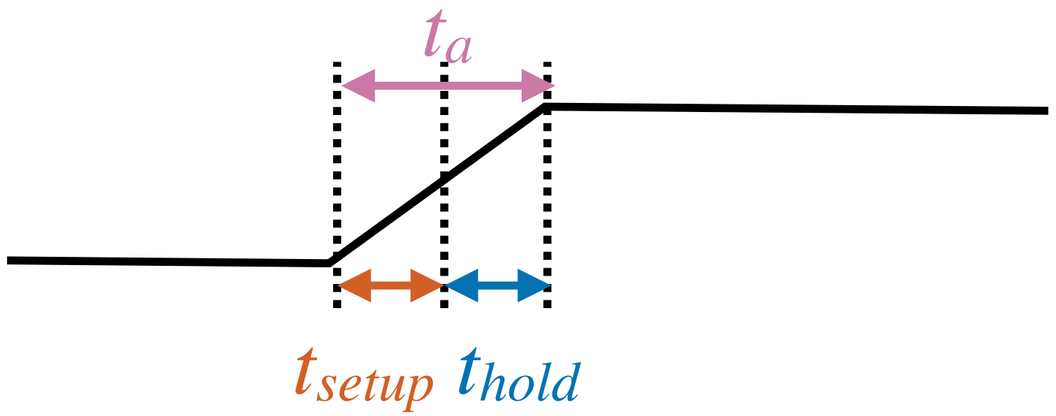
  A. Blue path

  B. Red path

  C. Both

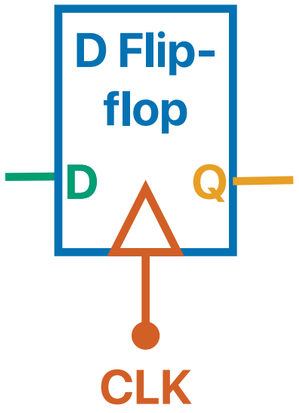
  D. Neither

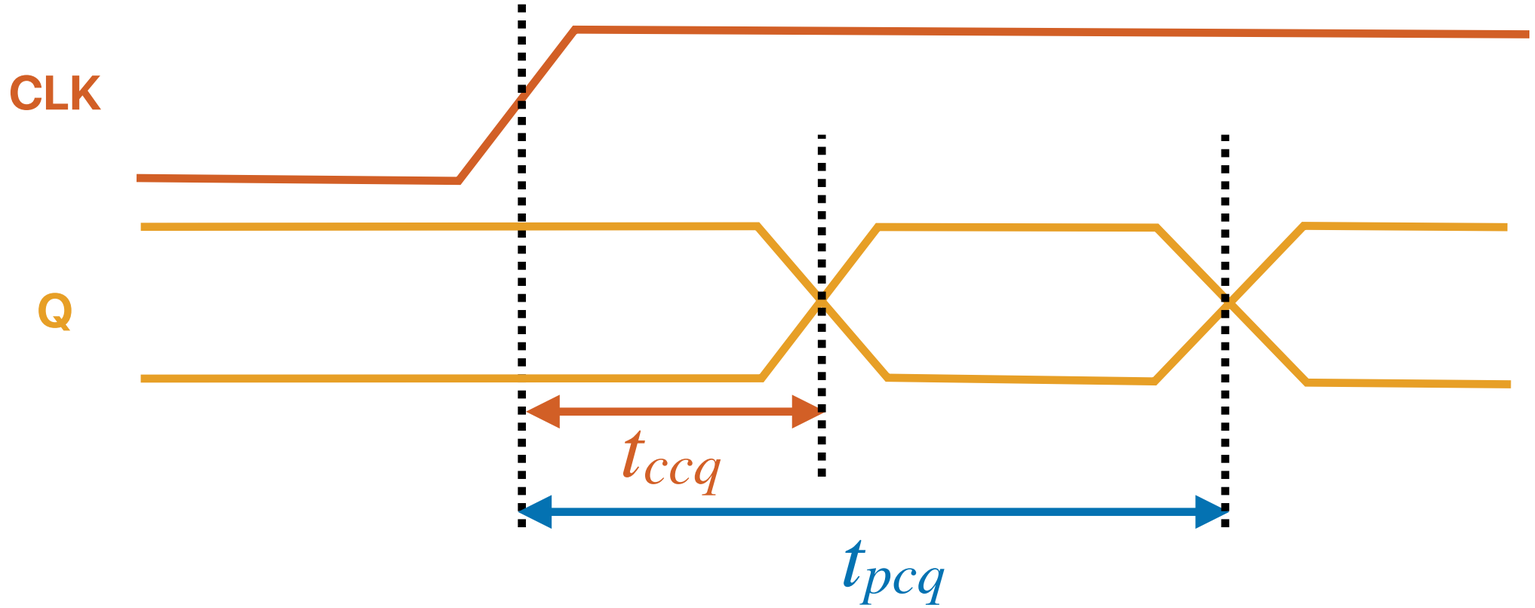
Setup and hold times



- Setup time: t_{setup}
  - Time before the clock edge that data must be stable (i.e. not change)
- Hold time: t_{hold}
  - Time after the clock edge that data must be stable
- Aperture time: t_a
  - Time around clock edge that data must be stable ($t_a = t_{setup} + t_{hold}$)

Output Timing Constraints





- Min delay of FF, also called contamination delay or min CLK to Q delay: t_{ccq}
  - Time after clock edge that Q might be unstable (i.e., starts changing)
- Max delay of FF, also called propagation delay or maximum CLK to Q delay: t_{pcq}
  - Time after clock edge that the output Q is guaranteed to be stable (i.e. stops changing)

Announcement

- iEval — Capture your screenshot, submit through iLearn and you will receive a full credit assignment
- Lab 6 is up — due on 6/2
  - Watch the video and read the instruction BEFORE your session
  - There are links on both course webpage and iLearn lab section
  - Submit through iLearn > Labs
- Office Hours
  - All office hours share the same meeting instance — if you have registered once, you cannot do it again.
  - Zoom does not resend registration confirmation and does not allow us to "re-approve" if you have registered
  - The only way is to dig out the e-mail from Zoom
- Final exam will be held during the campus scheduled period to avoid conflicts
  - 6/11 11:30am — 2:59:59pm
  - About the same format as midterm, but longer
  - Will have a final review on 6/6 to help you prepare

Electrical
Computer Science
Engineering

120A

つづく