WE5.2

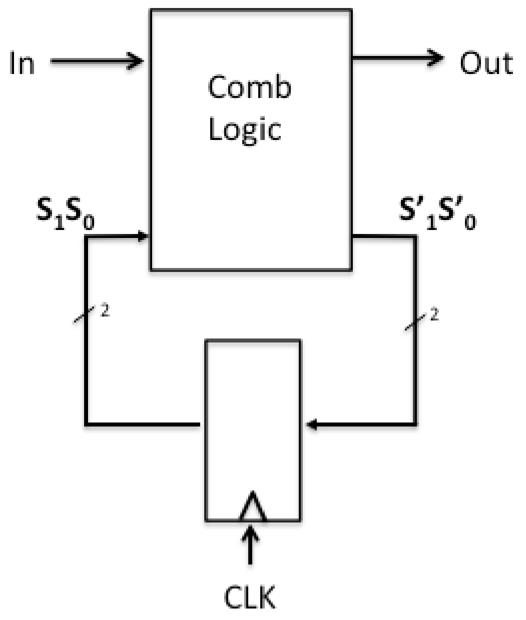
Bookmark this page

Sequential Logic Timing

3/4 points (ungraded)

OpenFSM.org, an organization dedicated to public disclosure of the transition diagrams of all commercially interesting FSMs, has hired you as an (unpaid) consultant. They've asked you to help reverse-engineer the BSM, an FSM embedded in the hot-selling consumer product, BlingSox.

You've been given the schematic diagram for the BSM below, and immediately recognize the diagram as an FSM having two state bits held in a single register.



OpenFSM has somehow been able to determine certain component timing specifications, listed below, but has been unable to come up with specifications for the BSM device as a whole. Moreover, they could not determine the hold time spec for the register.

Combinational Logic	
t_{PD}	3ns
t_{CD}	0.4ns

Register	
t_{PD}	1ns
t_{CD}	0.1ns
t_{SETUP}	1ns
t_{HOLD}	???

Nick O'Tyme, OpenFSM's performance expert, has asked you to find boundary values for timing specs that just guarantee reliable operation of the BSM.

What is the largest value for the register's hold time that allows the necessary timing specifications to be met?

Largest valid value for t_{HOLD} (ns): 0.5 ✓ Answer: 0.5

Explanation

In order to satisfy the t_{HOLD} of the register, the input to the register must remain stable until t_{HOLD} after the clock edge. The fastest that a new change can propagate to the input of the register is found by taking the sum of the contamination delays along the shortest path to that input. That is $t_{CD,register} + t_{CD,logic}$. So the constraint that $t_{HOLD,register}$ must satisfy is:

$$t_{HOLD,register} \leq t_{CD,register} + t_{CD,logic}$$

Plugging in the appropriate values, you'll get $t_{HOLD(register)} \leq 0.1 + 0.4$, so the most that the register's t_{HOLD} can be is 0.5ns.

What is the smallest value for the period of CLK that will meet the timing specifications?

Smallest value for t_{CLK} (ns): 5 ✓ Answer: 5

Explanation

The clock period must be long enough for data to pass through the entire circuit and be ready and stable for $t_{SETUP,register}$ before the next period begins. The propagation delay through the circuit must go through the register and the combinational logic. This results in the constraint:

Calculator

$$t_{CLK} \geq t_{PD,register} + t_{PD,logic} + t_{SETUP,register}$$

Again plugging in the appropriate values, you'll get $t_{CLK} \geq 1 + 3 + 1$, so the minimum clock period is 5ns.

Give the smallest setup and hold time specifications on IN with respect to the active edge of CLK that ensures the necessary timing specifications are met. **Assume that the t_{HOLD} spec for the register is 0.1ns.**

Setup time (ns): 4 ✓ Answer: 4

Hold time (ns): 1 ✗ Answer: -0.30000000000000004

Explanation

The IN input must be stable and valid long enough before the rising clock edge for it to propagate through the combinational logic and arrive at the register in time for it to setup. So it must be that:

$$t_{SETUP(input)} \geq t_{PD(logic)} + t_{SETUP(register)} = 3 + 1 = 4$$

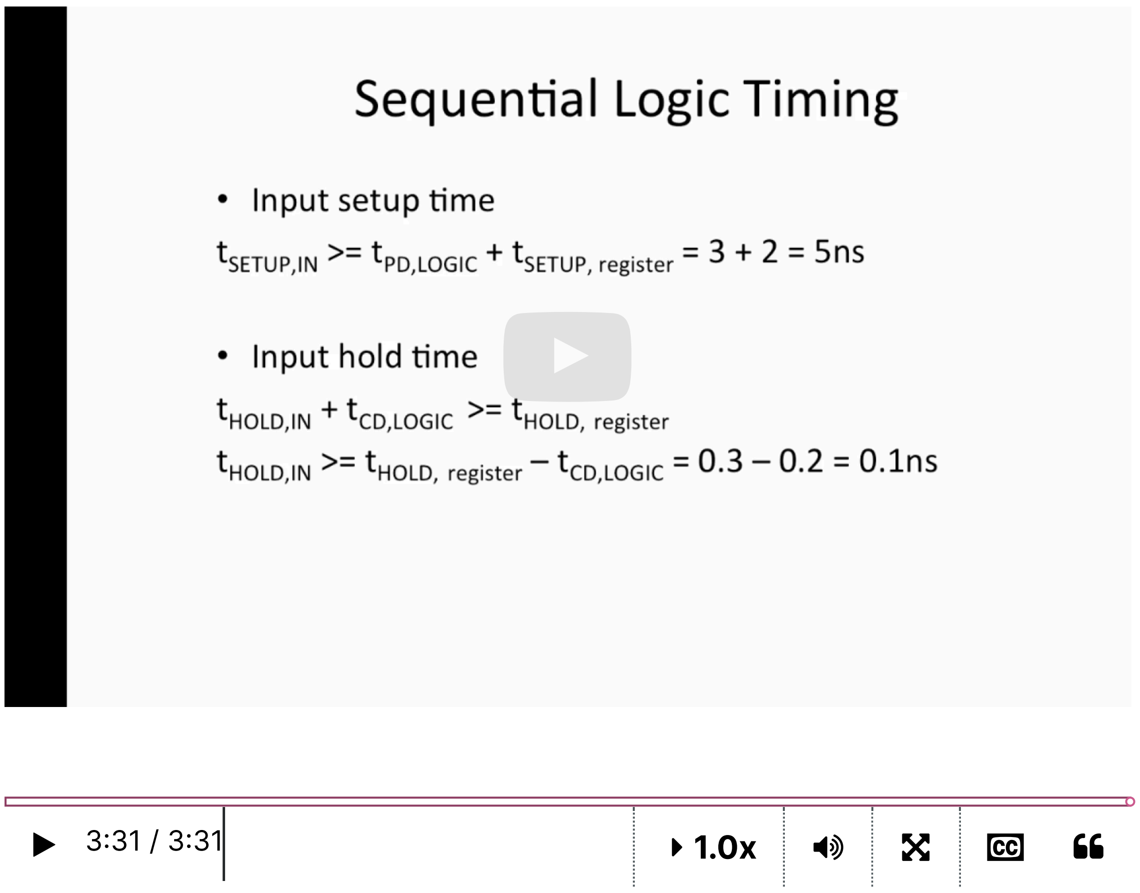
Once the IN input becomes invalid, the input to the register will become invalid after the contamination delay of the logic. IN must stay valid long enough to make sure that the input to the register doesn't become invalid before the register's hold time. So:

$$t_{HOLD(input)} + t_{CD(logic)} \geq t_{HOLD(register)}$$
$$t_{HOLD(input)} \geq t_{HOLD(register)} - t_{CD(logic)} = 0.1 - 0.4 = -0.30000000000000004$$

Submit

Answers are displayed within the problem

Sequential Logic Timing



Sequential Logic Timing

- Input setup time

$t_{SETUP,IN}$ >= $t_{PD,LOGIC}$ + $t_{SETUP, register}$ = 3 + 2 = 5ns

- Input hold time

$t_{HOLD,IN}$ + $t_{CD,LOGIC}$ >= $t_{HOLD, register}$

$t_{HOLD,IN}$ >= $t_{HOLD, register}$ – $t_{CD,LOGIC}$ = 0.3 – 0.2 = 0.1ns

Start of transcript. Skip to the end.

In this problem, we are going to take a look at the timing constraints associated with sequential logic circuits.

We are given a generic state machine diagram which has two state bits.

We are also given the timing parameters for the combinational logic and the state register.

Using this data, we want to determine the largest value for the register's hold time that allows the

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

Discussion

Hide Discussion

Topic: 5. Sequential Logic / WE5.2

Add a Post

Calculator

All Posts

tHOLD,register

question posted 7 years ago by **IanWRandman**

Why is tHOLD,register <= tCD,register + tCD,logic?

This post is visible to everyone.

silvinahw (Staff)
7 years ago - marked as answer 7 years ago by **silvinahw** (Staff)

This circuit is set up as a loop with a register in the middle. This means that data flows out of the register, through the combinational logic (or ROM) and then back to the input of the register. Since the hold time requirement for the register states that its input cannot change for tHold after the rising edge of the clock, it must be the case that the new data propagating through the loop does not make it back to the register input any faster the tHold,Reg. The fastest that data can propagate through the loop is determined by tCD,Reg + tCD,logic. So as long as this value is greater than tHold,Reg, then things will work properly and the hold time will be satisfied.

> The fastest that data can propagate through the loop is determined by tCD,Reg + tCD,logic.

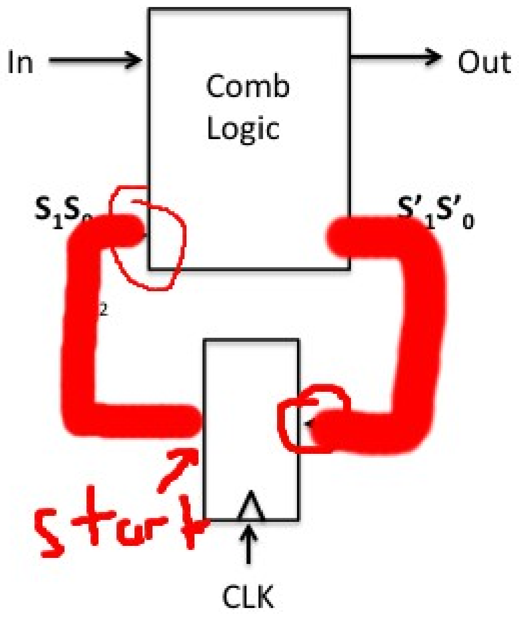
Why is the fastest that data can propagate through the loop not equal to tPD,Reg + tPD,logic?

posted 7 years ago by **IanWRandman**

Sorry for the confusion. The fastest that the data can propagate (correctly) through the circuit is tPD,Reg + tPD,logic. But tCD,Reg + tCD,logic is the fastest that a change might be seen on the input to the register. That value may not be the correct one but after tCD of the circuit you can no longer guarantee that the previous input is being held stable and valid any longer.

posted 7 years ago by **silvinahw** (Staff)

Is that for after the output of the register has already started changing, meaning that the input to the register changes after the tCD of the logic and then the tCD of the register after the data has made it back to the register? Sorry for my confusing explaination of my through process. I outlined how I envision it in the image below, with a circle representing tCD.



posted 7 years ago by **IanWRandman**

Your "start" should be on the right hand side of your register, at the register input. When you have a rising edge of the clock, the signal at the register input begins to flow through the register and then through the combinational logic until it eventually gets back to the register input once again.

The tHold requirement states that the value at the register input must not change for tHold after the rising edge of the clock. That means that the sum of the contamination delays of the register plus the logic must be greater than tHold to satisfy this requirement.

posted 7 years ago by **silvinahw** (Staff)

Add a comment

Preview

< Previous

Next >

edX

About
Affiliates
edX for Business
Open edX
Careers
News

Legal

Terms of Service & Honor Code
Privacy Policy
Accessibility Policy
Trademark Policy
Sitemap
Cookie Policy
Your Privacy Choices

Connect

Idea Hub
Contact Us
Help Center
Security
Media Kit

Calculator

Calculator