50.002 COMPUTATION STRUCTURES

INFORMATION SYSTEMS TECHNOLOGY AND DESIGN

Extras on Sequential Logic Timing

Natalie Agus (Fall 2018)

1 Sequential Logic

Recall that a sequential logic circuit is a circuit made up of various registers, and various combinational logic *unit* (i.e: adder unit, bitshift unit, any kind of logic units that are made up of MOSFETS), a single CLK, with input(s) and output(s).

2 Finding Setup Time for Input

The setup time t_S of a D-latch is defined as:

The time required for its *input* to be stable before the clock changes from 1 to 0

But since in the case of a register, the master sees the *inverted* clock, the setup time for a Register is defined as:

The time required for its *input* to be stablebefore the clock changes from 0 to 1

Let's consider several cases of Input to a sequential logic circuit:

1. **CASE 1: INPUT → REGISTER**

   In this case, the input to a combinational circuit **directly meets a register**, that's connected to another combinational logic circuit. The three dots on the right represents the rest of the circuits that are not drawn.

   The setup time for input in this case, since the input directly meets a register R1, is simply the setup time of Register R1

$$t_S = t_{S_{R1}}$$

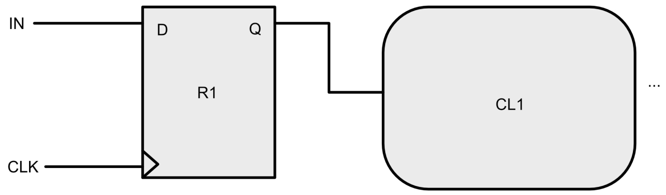

Figure 1

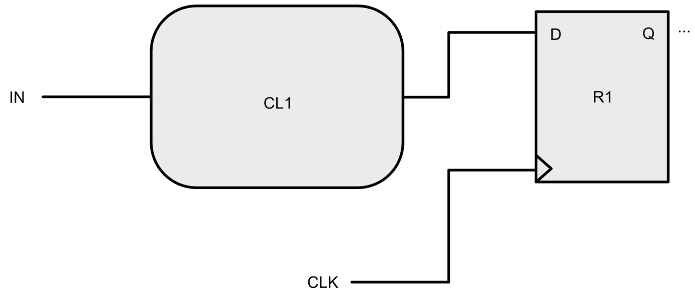

Figure 2

2. **CASE 2: INPUT → CL → REGISTER**

   In this case, there's the presence of some combinational circuit **before** the input meets the first register. The setup time for input in this case, is

   $$t_S = t_{pd_{CL1}} + t_{S_{R1}}$$

   This is because the input has to be firstly held stable and wait until CL1 is able to produce a valid output, and only then it can further fulfil the setup time of R1 and be held stable for as much as $t_{S_{R1}}$.

3 Finding Hold Time for Input

The hold time t_H of a D-latch is defined as:

The time required for its *input* to be stable after the clock changes from 1 to 0

But since in the case of a register, the master sees the *inverted* clock, the hold time for a Register is defined as:

The time required for its *input* to be stable after the clock changes from 0 to 1

Similarly, lets consider several cases of Input to a sequential logic circuit:

1. **CASE 1: INPUT → REGISTER**
   The hold time for input in this case is,

   $$t_H = t_{H_{R1}}$$

   This is because the input directly meets the register in the circuit, so the hold time of the input is simply the hold time required by the register.

2. **CASE 2: INPUT → CL → REGISTER**
   The hold time for this case is,

   $$t_H = t_{H_{R1}} - t_{cd_{CL1}}$$

   The hold time required for R1 is as specified, which is $t_{H_{R1}}$. But recall that CL1 does not immediately produce *invalid* value once the input turns invalid, because there is contamination delay $t_{CD_{CL1}}$. Therefore, the minimum hold time for the input is the **hold** time of R1 subtracted by the contamination delay of CL1.

   To further understand why we need to subtract the hold-time of the register with the contamination delay of the combinational logic unit, take a look at the diagram below,

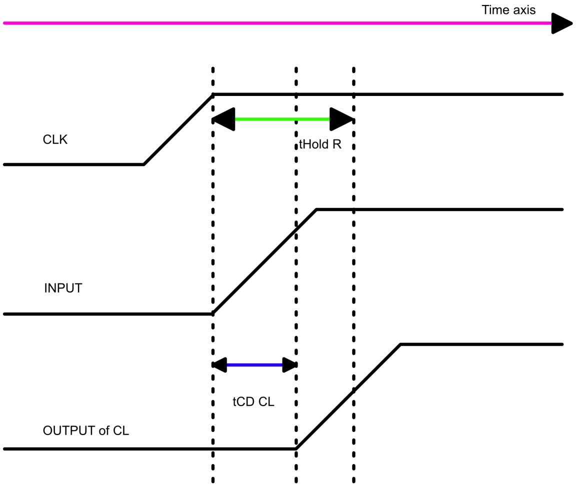

Figure 3

In the case above, you can see that **if input changes right away after CLK becomes stable 1**, then even though there's tCD of the CL [1], in this case it is not enough to fulfil the hold time (green line) of the register R [2].

[1] blue line, time taken for the output of CL to be invalid after the input to CL is invalid

[2] t-hold definition is the time required for the input to the D-latch to be stable after the clock changes from 1 to 0. The graph shows CLK changes from 0 to 1. But recall in flip-flop, the master latch **sees the inverse of the CLK**

So we need the hold time of the input,
which is the red line, to fulfil the green t-hold requirement of the register.

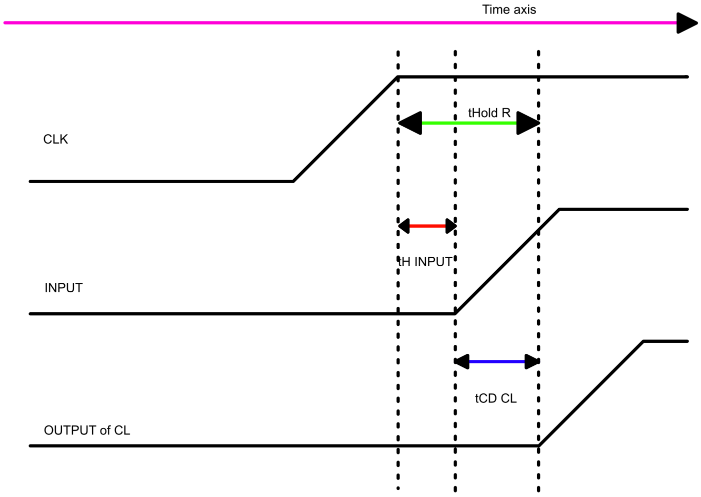

Figure 4

4 Finding the propagation delay and contamination delay of the entire circuit

The propagation delay t_{pd} of the entire sequential logic circuit is defined as:

The time taken to produce a valid *output* after the CLK turns valid

The contamination delay t_{cd} of the entire sequential logic circuit is defined as:

The time taken to produce an invalid *output* after the CLK turns invalid

Lets consider these two cases:

1. **CASE 1: REGISTER → OUTPUT**
   The t_{pd} of the circuit in this case is simple,

   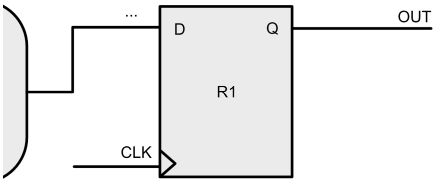

   Figure 5

   $$t_{pd} = t_{pd_{R1}}$$

   Similarly the t_{cd} of the circuit,

   $$t_{cd} = t_{cd_{R1}}$$

   By definition of propagation delay in sequential logic circuit, after the CLK rise becomes valid, the signal simply just have to pass through the register for $t_{pd_{R1}}$ before a valid OUT is produced. Same logic applies to the contamination delay. The invalid signal will be reflected at the OUT after $t_{cd_{R1}}$.

2. **CASE 2: REGISTER → CL → OUTPUT**
   The t_{pd} of the circuit in this case is,

   $$t_{pd} = t_{pd_{R1}} + t_{pd_{CL1}}$$

   The t_{cd} of the circuit is similarly,

   $$t_{cd} = t_{cd_{R1}} + t_{cd_{CL1}}$$

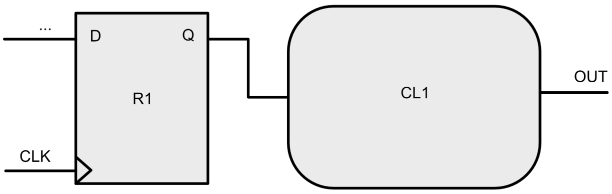

Figure 6

For propagation delay, this is because after the clock rise becomes valid, the signal is passed through by the R1 towards CL1, but one has to wait for as long as the propagation delay of CL1 before getting a valid OUT. Similar logic applies to contamination delay. The INVALID signal from CLK does not directly get reflected at the output of the R1, but one has to wait for as long as $t_{cd_{R1}}$ and only then the output of R1 becomes invalid. Afterwards, CL1 receives an invalid input, but similarly one has to wait for another $t_{cd_{CL1}}$ before the invalid signal OUT is reflected at the end of the sequential logic circuit.

5 Finding the minimum CLK period

According to timing constraint t_2 in the previous notes (Seq logic and Synchro), the clock period has to be larger than the time taken to finish the 'work' (propagation delays plus the setup time of the downstream register) between two registers.

Take a look at this slightly complicated sequential logic circuit below,

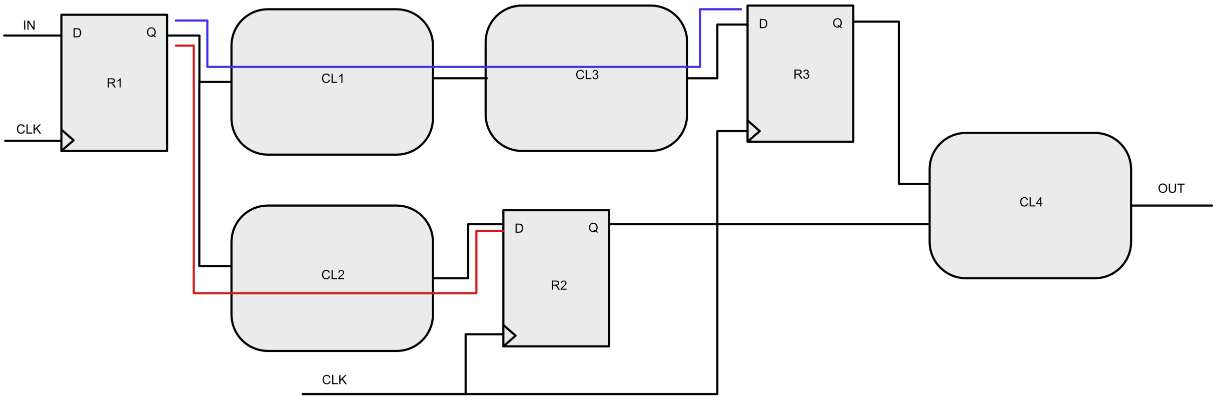

Figure 7

1. At each clock period (each time the clock rise from 0 to 1), a new input is being "loaded" to the registers, and the previous input is passed through to the rest

of the components downstream [3].

2. So before the next clock rise, the input at R1 has to propagate through R1, through the combinational logic, and finish setting up at the downstream register (R3 or R2)

3. There are two paths, blue and red where the output of R1 will flow downstream.

4. The time taken R1 to produce an output and reach R3, following the blue path is:

$$t_{\textbf{blue}} = t_{pd_{R1}} + t_{pd_{CL1}} + t_{pd_{CL3}} + t_{S_{R3}}$$

5. The time taken R1 to produce an output and reach **R2**, following the red path is:

$$t_{\textbf{red}} = t_{pd_{R1}} + t_{pd_{CL2}} + t_{S_{R2}}$$

6. The output of R1 has to traverse through both blue path and red path has to be done before the next clock rise

7. Hence the **minimum CLK period** is $max\{t_{\textbf{blue}}, t_{\textbf{red}}\}$.

6 Summary

1. When you begin to analyse timing requirements for sequential circuits, take note of **(1) paths between two registers**, **(2) paths from input to the FIRST register**, **(3) paths from the LAST register to the output**

2. To compute clk period: compute time taken to propagate from paths between two registers, take the *maximum* since clk period has to satisfy all paths

3. To compute tpd and tcd of entire circuit: compute time taken to propagate (and equivalently, the tcd time) from the last register to the output. If there's multiply ”last” registers and multiple output, compute the tpd as the max among them, and the tcd as the min among them (just like how you'd compute tpd and tcd of combinational logic circuit)

4. To compute the tH and tS of input, compute both values from all input to all ”first” registers in the circuit, and take the max of all.

[3] downstream means whichever unit after a specific component. For example CL1 is the downstream of R1. CL3 is the downstream of both CL1 and R1, etc.