Using a D register as the memory component in our sequential logic system works great!

At each rising edge of the clock, the register loads the new state, which then appears at the register's output as the current state for the rest of the clock period.

The combinational logic uses the current state and the value of the inputs to calculate the next state and the values for the outputs.

A sequence of rising clock edges and inputs will produce a sequence of states, which leads to a sequence of outputs.

In the next chapter we'll introduce a new abstraction, finite state machines, that will make it easy to design sequential logic systems.

Let's use the timing analysis techniques we've learned on the sequential logic system shown here.

The timing specifications for the register and combinational logic are as shown.

Here are the questions we need to answer.

The contamination delay of the combinational logic isn't specified.

What does it have to be in order for the system to work correctly?

Well, we know that the sum of register and logic contamination delays has to be greater than or equal to the hold time of the register.

Using the timing parameters we do know along with a little arithmetic tells us that the contamination delay of the logic has to be at least 1ns.

What is the minimum value for the clock period tCLK?

The second timing inequality from the previous section tells us that tCLK has be greater than than the sum of the register and logic propagation delays plus the setup time of the register.

Using the known values for these parameters gives us a minimum clock period of 10ns.

What are the timing constraints for the Input signal relative to the rising edge of the clock?

For this we'll need a diagram!

The Next State signal is the input to the register so it has to meet the setup and hold times as shown here.

Next we show the Input signal and how the timing of its transitions affect to the timing of the Next State signal.

Now it’s pretty easy to figure out when Input has to be stable before the rising clock edge, i.e., the setup time for Input.

The setup time for Input is the sum of propagation delay of the logic plus the setup time for the register, which we calculate as 7ns.

In other words, if the Input signal is stable at least 7ns before the rising clock edge, then Next State will be stable at least 2ns before the rising clock edge and hence meet the register’s specified setup time.

Similarly, the hold time of Input has to be the hold time of the register minus the contamination delay of the logic, which we calculate as 1ns.

In other words, if Input is stable at least 1ns after the rising clock edge, then Next State will be stable for another 1ns, i.e., a total of 2ns after the rising clock edge.

This meets the specified hold time of the register.

This completes our introduction to sequential logic.

Pretty much every digital system out there is a sequential logic system and hence is obeying the timing constraints imposed by the dynamic discipline.

So next time you see an ad for a 1.7 GHz processor chip, you’ll know where the “1.7” came from!