Design Exercise

RTL Block Diagrams for a timing recovery circuit

- 10 pts
- Create your own design (but can discuss with teams)
- Write short description for each block, state transition diagrams for any state machines, indicate number of Flip-flops in each block
- Include your name, lab section, mg account, and date at top
- Collaboration with one or two other people is permitted, but each person is expected to prepare their own solution.
- If you collaborated, give your collaborator's name and briefly describe how you collaborated. Or state that you worked alone.
- The solution should either be prepared electronically or very nearly drawn and scanned and then uploaded in Brightspace

Updated 9/1/2020

Concept

- Some serial interfaces rely on the receiving interface to produce a sequence of timing pulses that are synchronized to the incoming data.
- The sequence of pulses will not stay synchronized to the incoming data unless your design continually adjusts to the incoming data bits.
- Your task: draw a hierarchical RTL block diagram for a circuit which produced a sequence of pulses, one per incoming data bit.

Updated 9/1/2020

Design Exercise Specs

Clock/Reset	Inputs	Outputs
60 MHz clock called **Clk,** (All flip-flops must run on this clock)	**D_i:** (async.) Serial data input. Input should be 1 when no data is transmitted	**Shift_en_o:** a timing pulse that is asserted each time a new data bit is received
Asynchronous active low reset, **nRst** Input of zero forces count to 0.		Timing pulses should cease if D_i does not change for more than 6 consecutive bit periods

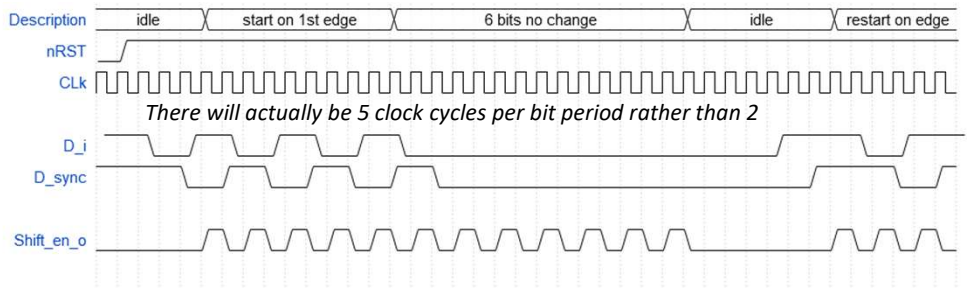
Asynchronous inputs must be passed through a synchronizer.

Updated 9/1/2020

Design Exercise Specs

Top Level Block per...	Diagram for each top level block	Other Requirements
Counter	**Registers**	**Indicate number of bits on all busses**
State Machine	**Next state logic**	**compute number of flip-flops required**
Other major functional blocks	**Output logic**	**Synchronizers for all asynchronous inputs ~~including reset~~**
	Other combinational functions	

d 9/1/2020



nRst = 0 forces reset of all registers in design.

D_i is initially idle, i.e., D_i = 1

D_sync refers to **D_i** after going through synchronizers

If D_sync doesn't change for 6 bit periods, stop the shift enable

ShiftEn: is output of the timing recovery block. Incoming data will be at a frequency of 12Mhz. Thus, shift enable pulses will also be generated at 12MHz, but exact timing must be adjusted to match the incoming data which may deviate slightly. The clock period and Shift_en pulses will be much shorter than they are shown in this diagram.

5

Updated 9/1/2020

5

Tips

- Draw a **functional block diagram** 1st and make sure that it accounts for the required functions and interface signals. Then add clock and reset signals were necessary. This will become the top diagram in your hierarchical RTL block diagram.
- For each block, you will need to prepare a supporting diagram (see sample solution to prior RTL homework)
- A sample of a prior assignment will be posted in BrS to give you an idea of the expected format.

6

Updated 9/1/2020

6

Tips

- Your design should include:
  - A synchronous edge detector (to detect rising and falling edges on D_i). Resembles a synchronizer + a gate to compare the FF outputs.
  - A clock divider that produces pulses at 12MHz. Every transition on D_i should cause the clock divider to restart.
- Make sure that the shift enable pulses stop if 6 clock divider pulses occur consecutively without a change in D_i

Updated 9/1/2020

Tips

- Include a synchronizer on the input D_i
- You are expected to partition the design into small enough functional blocks so that each block would be relatively simple to design and debug.
- If you come to the Tuesday morning lecture review or office hours, we can discuss some of the blocks you will need and how to get started preparing the solution.

Updated 9/1/2020