Computation II: embedded system design (5EIB0)

Part 2e: 2022-04-21 Design a parametrized counter - 6pt

Due date: Thursday, 21 April 2022, 5:45 PM
Requested files: parametrized_counter.v (Download)
Type of work: Individual work

Assignment

In this assignment, you have to create a Verilog implementation of a parametrized counter (**parametrized_counter**). When the counter is triggered, it will run for a number of cycles, and then stop counting until it is triggered again.

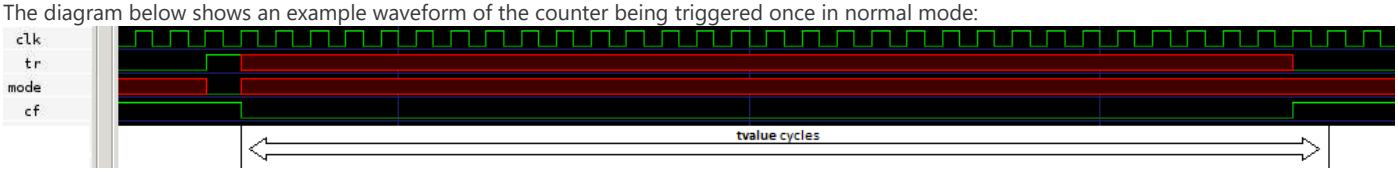
Compared to a normal counter, this parametrized counter has two special features:

1. The number of cycles it will count can be customized using a Verilog parameter.
2. It has a *slow* mode in which it runs three times as long.

It has the following ports:

- **clk (1 bit input, positive edge):** Clock for the module
- **reset (1 bit input, active high, synchronous):** Reset for the module.
- **tr (1 bit input, active high):** Trigger (i.e. start) the counter. *If the counter was already running, nothing should happen.*
- **cf (1 bit output, active low):** Indicates the counter is running. In other words, it is high only when the counter is **not** running.
- **tvalue (parameter):** Number of cycles to count in *normal* mode (i.e. **cf** should be low for ***tvalue*** - *1* cycles). *Your module must support **tvalue**s up to 32 cycles.*
- **mode (1 bit input, active high):** Enable *slow* mode (i.e. if **tr** is asserted while **mode** = 1, cf should be low for *3* × ***tvalue*** - *1* cycles). *If **mode** changes while the counter is running, nothing should happen.*

The diagram below shows an example waveform of the counter being triggered once in normal mode:



Debug

Click on the symbol marked below to see the waveforms produced by your design. Please note that no waveforms will be produced if your code has an error that prevents it from being simulated.

*Note: in the waveforms you might see that some of the inputs are undefined (x) at some moments in time. This means that the behaviour of your module should **not** depend on those values at that moment in time. (If your module incorrectly does use them anyway, it might cause undefined values (x) to appear at its outputs, and it will fail the test.) In particular, the testbench sets **mode** and **tr** to x while the counter is supposed to be running.*

Requested files

parametrized_counter.v

```
1  `timescale 1ns / 1ps
2  module parametrized_counter (
3          input clk,
4          input reset,
5          input tr,
6          output cf,
7          input mode
8      );
9
10     parameter tvalue = ...; // Replace the ... with a sane default value
11
12     // Add your implementation here
13
14 endmodule
```

VPL

◄ Part 2d: 2022-04-21 Finite State Machine (Design Moore AC controller - 6pt)

Jump to...

Part 2f: 2022-04-21 Finite State Machine (Design Washing Machine Control Unit - 7pt) ►