

*Homework 3 is being split in to homework 3 and 4. Homework 4 is really just homework 2b, but calling it that would ruin the numbering scheme. Solution templates can be found in /home/classes/ee4702/files/v and will be linked to the web page. Instructions for submission can be found in the homework template.*

**Problem 1:** Write an implicit structural description of the module designed in homework 2, either your design or the posted solution. Note: see the template for a workaround to a bug when using parameters.

**Problem 2:** Design a behavioral description of hardware similar to the one from homework 2, but that measures rotation speed by measuring the time between marks. The inputs and output are the same and the parameters are the same, except that `update_interval` is missing. In this module the output should be updated for each detected mark. (The update does not have to occur on the positive edge of `pd`, but it does have to be updated sometime.) The output must correctly indicate zero rotation rate. (You'll see why that needed to be specified.) Though the number of marks is known the width of the marks is not.

```
module tach2(rpx,pd,clk);
    input pd, clk;
    output rpx;
    wire pd, clk;
    reg [9:0] rpx;

    parameter freq = 500;    // Clock frequency.
    parameter marks = 4;    // Four pulses per revolution.
    parameter perwhat = 60; // Measure in revolutions per 60 seconds.
// Code here.

endmodule
```

Follow the following rules when writing the hardware description. (The rules do not apply to testbench code.)

- You can use multipliers or dividers. (Just use the usual operators, no need to instantiate anything.)
- Do not use delays: `#3 i=1;`. You can use event controls: `@(posedge clk)`.
- Use the initial block for parameter verification and register initialization only.