# CS152A Lab 1 Workshop 1

## Clock Dividers

1. Add clk\_en to the simulation’s waveform tab and then run the simulation again. Use the cursor to find the periodicity of this signal (you can select the signal and use arrow keys to reach the exact edges). Capture a waveform picture that shows two occurrences of clk\_en, and include it in the lab report. Indicate the exact period of the signal in the report.![A screen shot of a computer

   Description automatically generated]()
2. A duty cycle is the percentage of one period in which a signal or system is active: , where D is the duty cycle, T is the interval where the signal is high, and p is the period. What is the exact duty cycle of clk\_en signal?
3. What is the value of clk\_dv signal during the clock cycle that clk\_en is high?

At the positive edge of clk\_en, all 16 clk\_dv are zero and at the falling edge, only clk\_dv[0] is 1 and the rest remain as zeros.

1. Draw a simple schematic/diagram of signals clk\_dv, clk\_en, and clk\_en\_d signals. It should be a translation of the corresponding Verilog code.

A diagram of a program

Description automatically generated

## Debouncing

1. What is the purpose of clk\_en\_d signal when used in expression ~step\_d[0] & step\_d[1] & clk\_en\_d? Why don't we use clk\_en?

Because clk\_en has the same start time as the step\_d[0:2] and we want to delay it to make sure that we are &-ing it with a positive rising edge.

1. Instead of clk\_en <= clk\_dv\_inc[17], can we do clk\_en <= clk\_dv[16], making the duty cycle of clk\_en 50%? Why?

By doing clk\_dv[16], we are making the duty cycle of clk\_en to 2^16 instead of 2^17. This means that by looking at the 17th bit instead of the 18th bit, then we are essentially clock frequency by two, leading to a 50% duty cycle.

We don’t want to make it 50% duty cycle because it will be on high for half the time and low for the other half and this might make us execute the same instructions over and over.

1. Include waveform captures that clearly show the timing relationship between clk\_en, step\_d[1], step\_d[0], btnS, clk\_en\_d, and inst\_vld.

![A green and black background

Description automatically generated]()

1. Draw a simple schematic/diagram of the signals above. It should be a translation of the corresponding Verilog code.

A diagram of a computer program

Description automatically generated

## Register File

1. Find the line of code where a register is written a non-zero value. Is this sequential logic or combinatorial logic?

Line: rf[i\_wsel] <= i\_wdata;

This is sequential because it depends on the clock: always @ (posedge clk)

1. Find the lines of code where the register values are read out from the register file. Is this sequential or combinatorial logic? If you were to manually implement the readout logic, what kind of logic elements would you use?

assign o\_data\_a = rf[i\_sel\_a];

assign o\_data\_b = rf[i\_sel\_b];

This logic is combinatorial logic because the output values o\_data\_a and o\_data\_b are determined directly by the current values of i\_sel\_a and i\_sel\_b, without any dependence on clock signals or past states.

1. Draw a circuit diagram of the register file block. It should be a translation of the corresponding Verilog code.

A diagram of a computer program

Description automatically generated

1. Capture a waveform that shows the first time register 3 is written with a non-zero value.![A screen shot of a video game

   Description automatically generated]()

# CS152A Lab 1 Workshop 2

## Nicer UART Output

In class demo.

## An Easier Way to Load Sequencer Program

1. Identify the part of the tb.v where the instructions are sent to the UUT.

The instructions sent to the UUT are executed within the initial block, specifically in the sequence of tasks that represent different instructions:

1. Which user tasks are called in this process?

tskRunPUSH(0,4);

tskRunPUSH(0,0);

tskRunPUSH(1,3);

tskRunMULT(0,1,2);

tskRunADD(2,0,3);

tskRunSEND(0);

tskRunSEND(1);

tskRunSEND(2);

tskRunSEND(3);

## Fibonacci Numbers

In class demo.