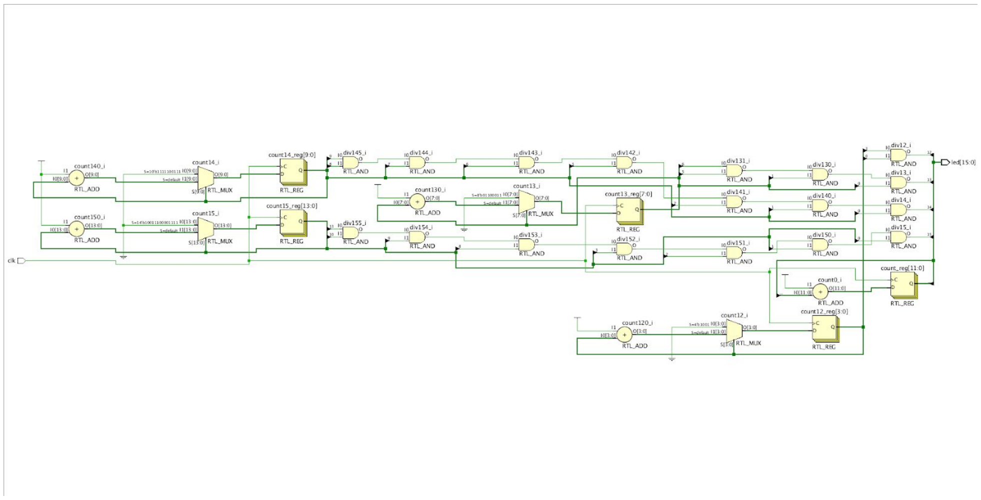
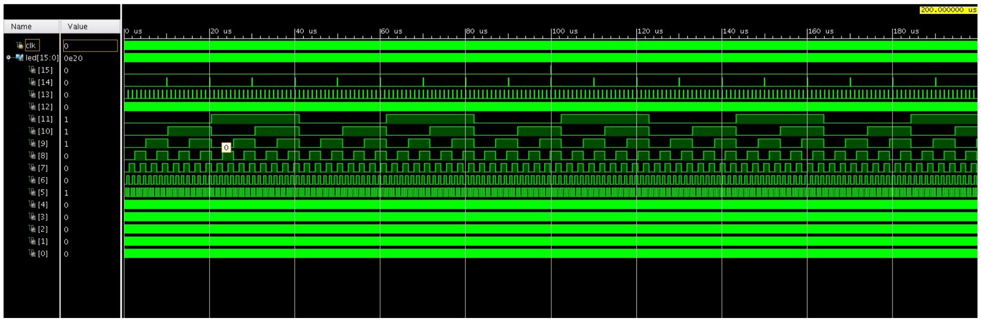
# Lab4 - Sequential Logic Design

Colby Janecka
CDJ2326
Telang ~ Wed. 12-1

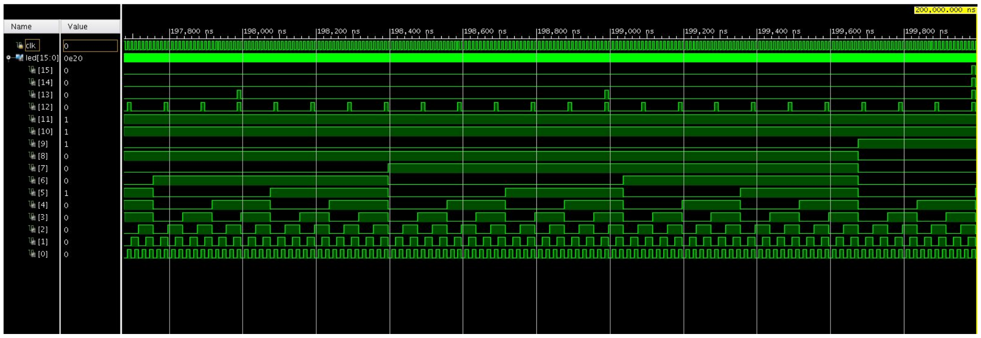
## a. Clock divider bank schematic.



## b. Clock divider bank waveform(s).

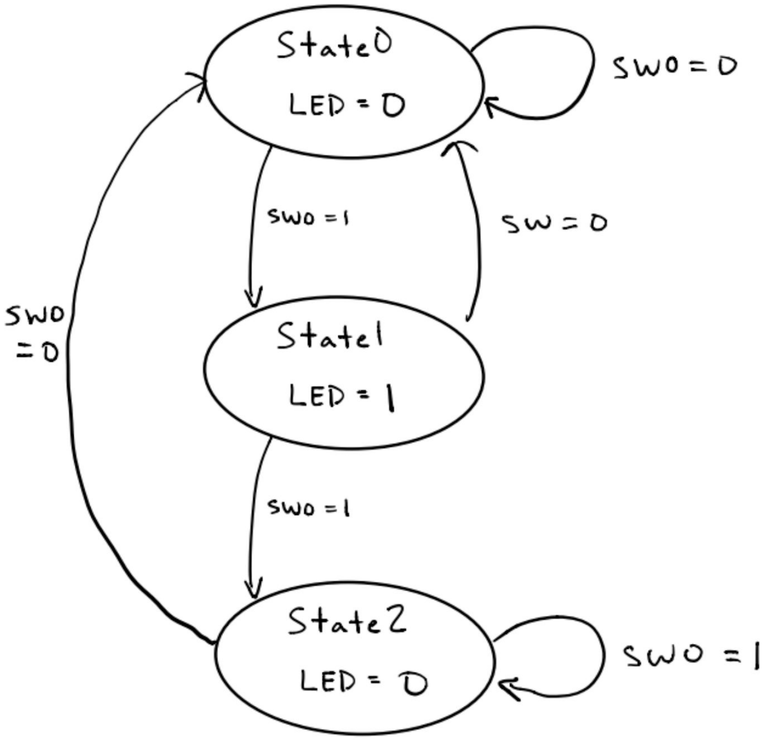


| Name | Value |
|---|---|
| clk | 0 |
| led[15:0] | 0e20 |
| [15] | 0 |
| [14] | 0 |
| [13] | 0 |
| [12] | 0 |
| [11] | 1 |
| [10] | 1 |
| [9] | 1 |
| [8] | 0 |
| [7] | 0 |
| [6] | 0 |
| [5] | 1 |
| [4] | 0 |
| [3] | 0 |
| [2] | 0 |
| [1] | 0 |
| [0] | 0 |



| Name | Value |
|---|---|
| clk | 0 |
| led[15:0] | 0e20 |
| [15] | 0 |
| [14] | 0 |
| [13] | 0 |
| [12] | 0 |
| [11] | 1 |
| [10] | 1 |
| [9] | 1 |
| [8] | 0 |
| [7] | 0 |
| [6] | 0 |
| [5] | 1 |
| [4] | 0 |
| [3] | 0 |
| [2] | 0 |
| [1] | 0 |
| [0] | 0 |

## c. Rising-edge detector Moore FSM.



## d. Rising edge detector frequency calculations.

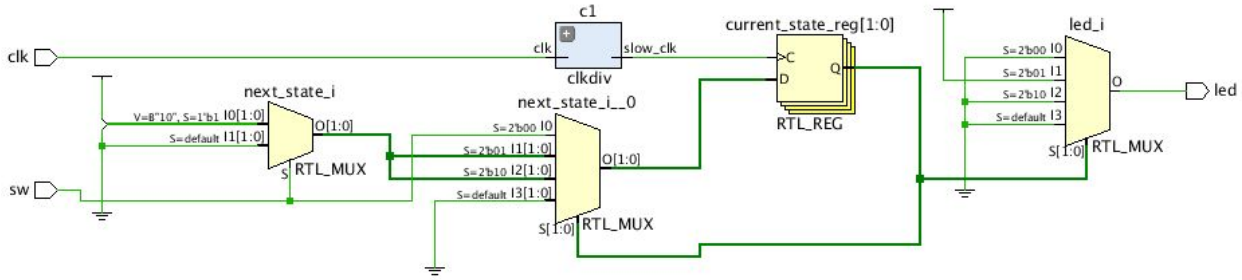
To remain at State 3 for 50ns:

$$\text{freq} = \frac{1}{50\text{ns}} = 20 \text{ MHz}$$

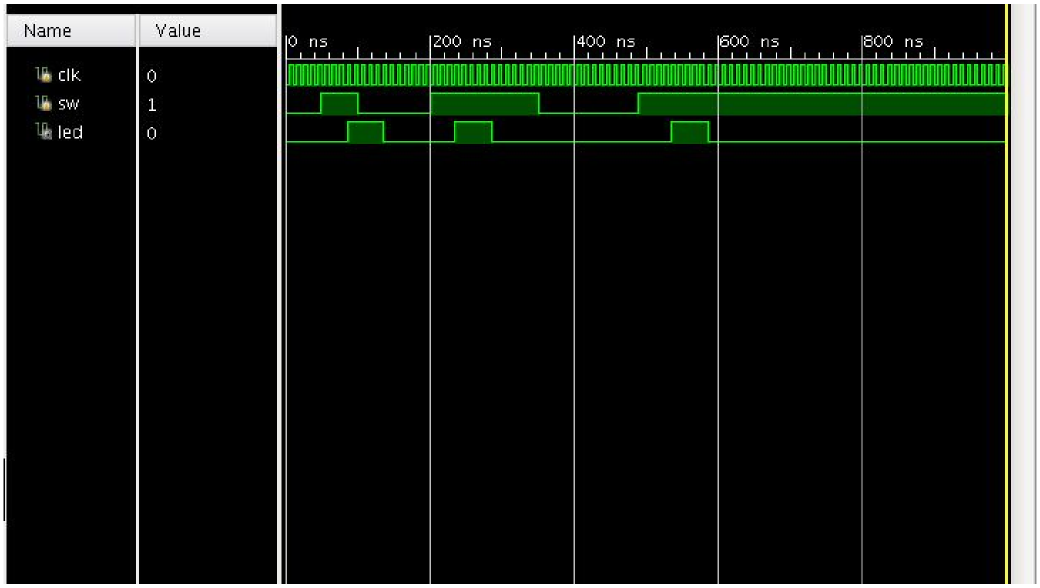
board clock = 100 MHz = 10 ns

So, we need to toggle the slow clock for every 5 board clock cycles.

## e. Rising edge detector schematic.

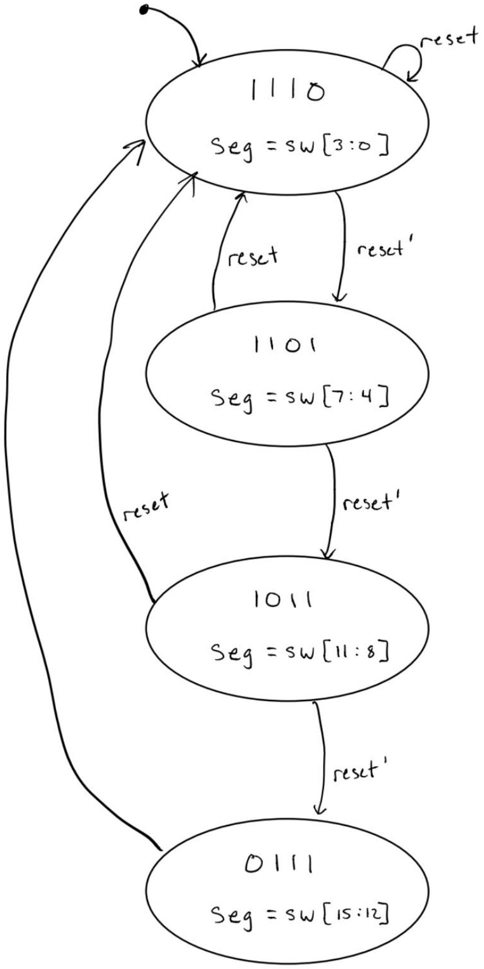


## f. Rising edge detector waveform screenshot.



| Name | Value |
|---|---|
| clk | 0 |
| sw | 1 |
| led | 0 |

g. **Seven-segment display Moore FSM.**

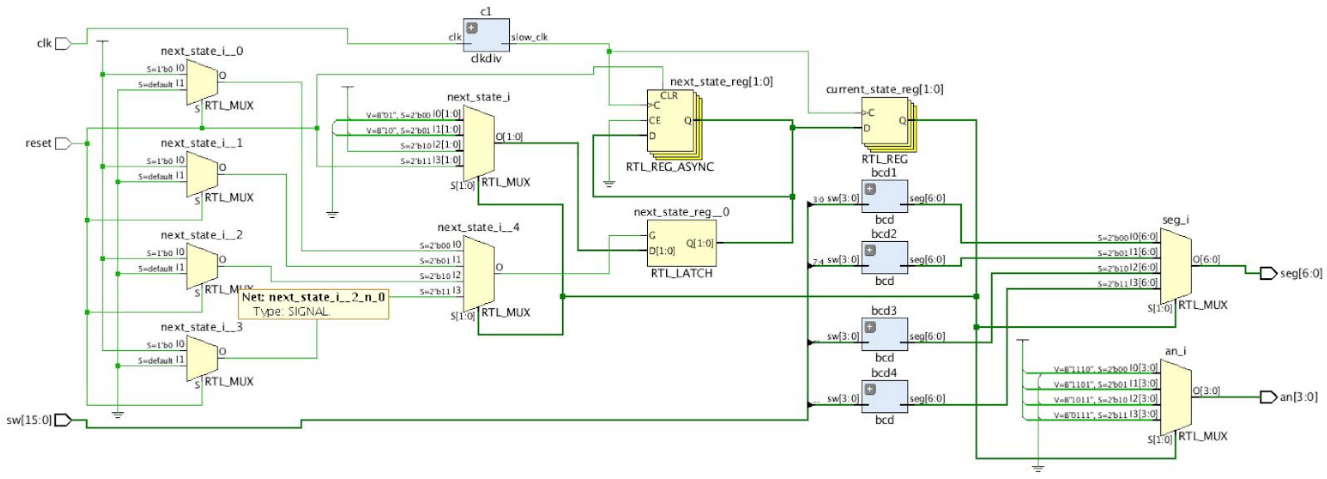


h. **Seven-segment display frequency calculations.**

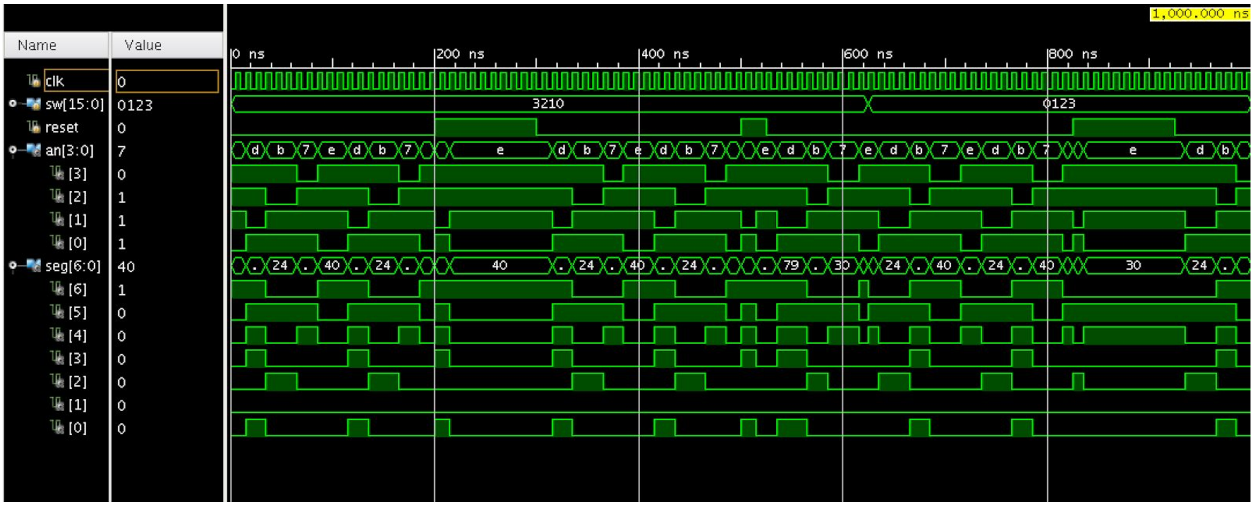
$\frac{1}{40MHz} = 25ns$, board = 10ns,

so we must wait 2.5 board clock cycles.

## i. Seven-segment display schematic.



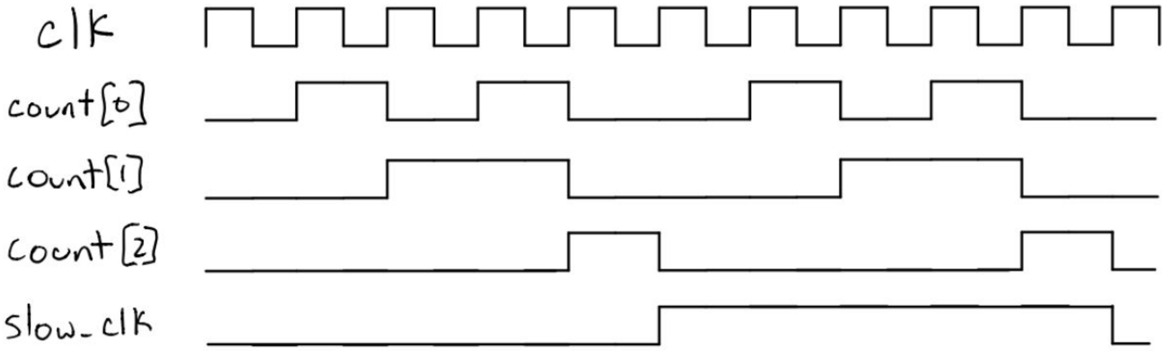
## j. Seven-segment display waveform screenshot.



| Name | Value |
|---|---|
| clk | 0 |
| sw[15:0] | 0123 |
| reset | 0 |
| an[3:0] | 7 |
| [3] | 0 |
| [2] | 1 |
| [1] | 1 |
| [0] | 1 |
| seg[6:0] | 40 |
| [6] | 1 |
| [5] | 0 |
| [4] | 0 |
| [3] | 0 |
| [2] | 0 |
| [1] | 0 |
| [0] | 0 |

k. Answers to the 8 checkout questions listed below.

1. In less than 20 words, how would you design a divide-by-5 clock? Draw a timing diagram of your divide-by-5 clock.
    a. Create a counter on the input clock, and whenever it reaches 5 toggles the slow_clock and reset the count.

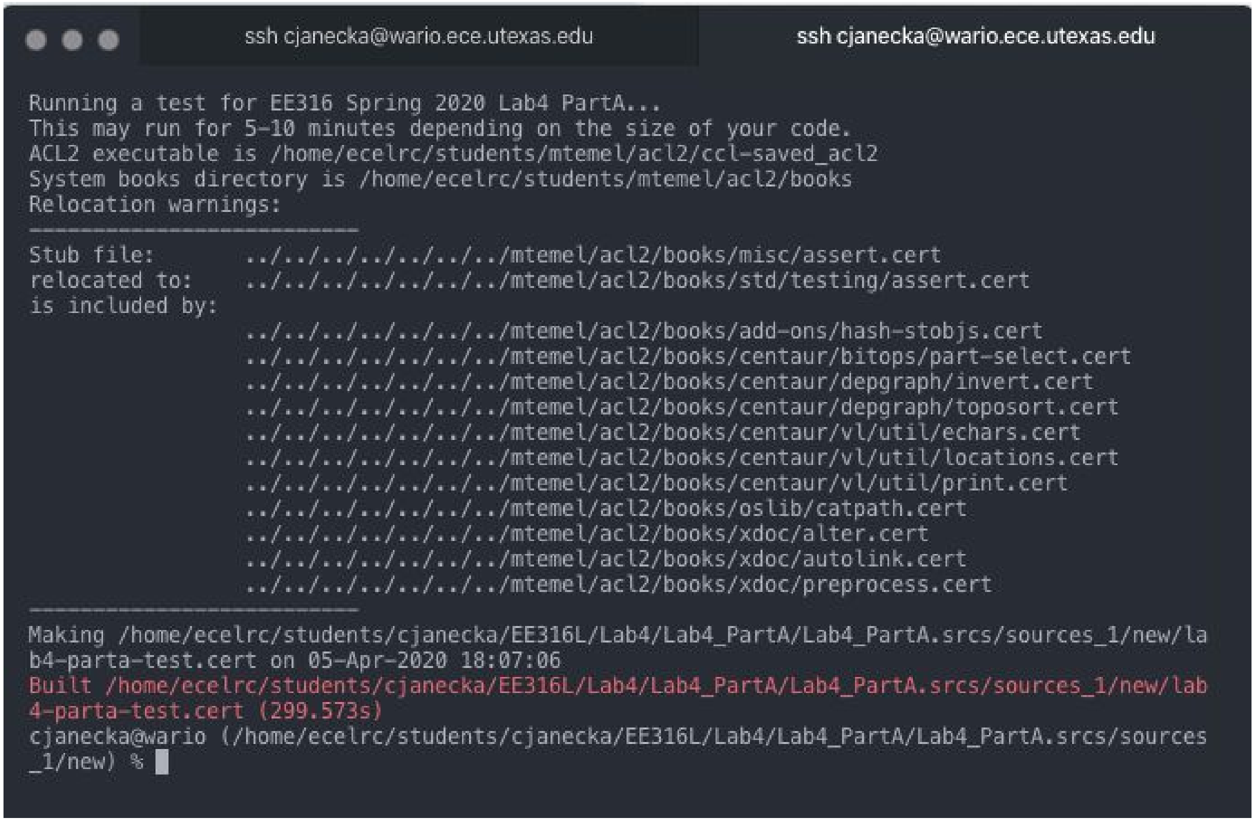


2. What is a sensitivity list? Why is it important?
    a. Sensitivity lists are conditions used to evaluate when an always block should be executed. It is very important for creating conditional statements that activates a process when a specified signal has changed.
3. Define blocking and non-blocking. When is each used?
    a. Blocking statements ('=') are executed sequentially, meaning they are executed in order one at a time. Non-blocking statements ('<=') on the other hand, execute simultaneously. Blocking statements are used when the code needs to be executed in a

**specific order, whereas non-blocking statements are used to assign values to multiple registers at the same exact time.**

4. **Why is it impossible to get a clock of exactly 10Hz by using bit-masking in Verilog?**
    a. **A bit mask dividers are not exactly precise due to the time it takes to run the bit mask operations, which may result in slight variations (1%).**
5. **Define synchronous and asynchronous. Is an FSM synchronous or asynchronous?**
    a. **Synchronous tasks operate simultaneously, or at the same time. On the other hand, asynchronous tasks operate on timescales independent of one another. Because the FSM behavior in this lab was driven by the clock, it is considered synchronous. If this was not the case, the FSM would be considered asynchronous.**
6. **How do we get around the "problem" of not being able to display four different numbers on the seven-seg at the same time?**
    a. **In order to display data to multiple displays, we used time multiplexing. This is the practice of alternating the digit that you are sending data to over time. As it cycles through anodes at a desired frequency (40MHz in our case), it appears to the naked eye that they are all actively displaying numbers simultaneously.**
7. **What do the keywords `posedge` and `negedge` mean, and where in your code do you use them?**
    a. **Posedge and negedge tell the program which side of a clock pulse to use for running statements. If posedge is used as a conditional, the statement is run when the rising edge occurs, while negedge would cause it to run on the falling edge.**
8. **How would you change your edge detector state machine to create an edge detector that senses both rising and falling edges?**
    a. **In order to do this, I would add an additional condition to the always block using an OR statement. For example, `always @(posedge clk or negedge clk)` would sense both edges.**

## Proof of working PartA:


ssh cjanecka@wario.ece.utexas.edu
ssh cjanecka@wario.ece.utexas.edu

```
Running a test for EE316 Spring 2020 Lab4 PartA...
This may run for 5-10 minutes depending on the size of your code.
ACL2 executable is /home/ecelrc/students/mtemel/acl2/ccl-saved_acl2
System books directory is /home/ecelrc/students/mtemel/acl2/books
Relocation warnings:
--------------------------
Stub file:       ../../../../../../../mtemel/acl2/books/misc/assert.cert
relocated to:    ../../../../../../../mtemel/acl2/books/std/testing/assert.cert
is included by:
                 ../../../../../../../mtemel/acl2/books/add-ons/hash-stobjs.cert
                 ../../../../../../../mtemel/acl2/books/centaur/bitops/part-select.cert
                 ../../../../../../../mtemel/acl2/books/centaur/depgraph/invert.cert
                 ../../../../../../../mtemel/acl2/books/centaur/depgraph/toposort.cert
                 ../../../../../../../mtemel/acl2/books/centaur/vl/util/echars.cert
                 ../../../../../../../mtemel/acl2/books/centaur/vl/util/locations.cert
                 ../../../../../../../mtemel/acl2/books/centaur/vl/util/print.cert
                 ../../../../../../../mtemel/acl2/books/oslib/catpath.cert
                 ../../../../../../../mtemel/acl2/books/xdoc/alter.cert
                 ../../../../../../../mtemel/acl2/books/xdoc/autolink.cert
                 ../../../../../../../mtemel/acl2/books/xdoc/preprocess.cert
--------------------------
Making /home/ecelrc/students/cjanecka/EE316L/Lab4/Lab4_PartA/Lab4_PartA.srcs/sources_1/new/la
b4-parta-test.cert on 05-Apr-2020 18:07:06
Built /home/ecelrc/students/cjanecka/EE316L/Lab4/Lab4_PartA/Lab4_PartA.srcs/sources_1/new/lab
4-parta-test.cert (299.573s)
cjanecka@wario (/home/ecelrc/students/cjanecka/EE316L/Lab4/Lab4_PartA/Lab4_PartA.srcs/sources
_1/new) %
```