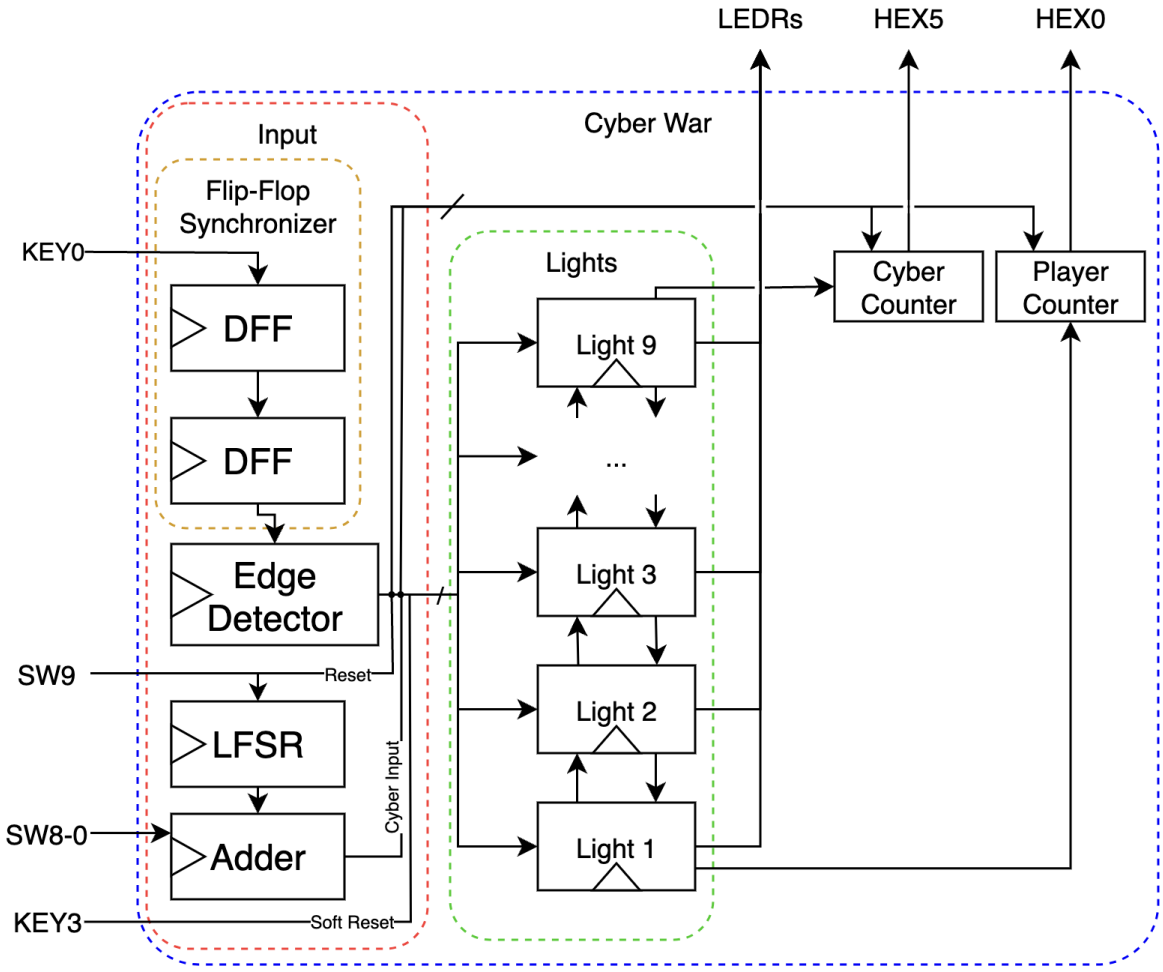
CSE 369 Lab 7

Useful Components

Isaac Wu
2360957
November 20, 2024

1. Top-Level Block Diagram



This is my Top-Level Block Diagram for the Cyber War module. The Cyber War module takes in 12 inputs, KEY0 (for Player Input), KEY3 (to reset the playfield after a victory), SW9 (to reset the game), and SW8-0 (to calibrate the difficulty of the cyber player). Only the player input needs to be synchronized, so it gets passed through two D-flip-flops to account for metastability. I then pass in the user input into an Edge Detector to reduce each button press to one cycle in length since our clock is very fast.

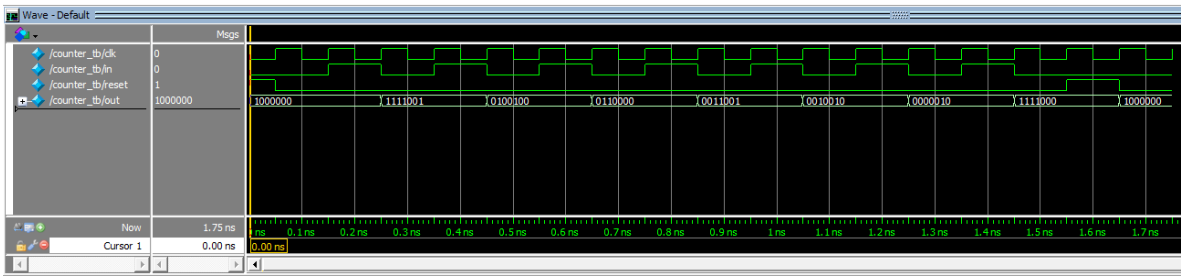
A 9-Bit LFSR is initialized and its status is read as an unsigned number. The current state of the LFSR is added with the unsigned number interpreted from SW8-0. If the resulting sum is above a certain threshold, then the cyber player will input on the next clock edge.

The inputs for both players, along with the reset and soft reset inputs are all passed into each of the 9 light modules. These light modules have their own state and will determine whether or not their linked LED will be lit, which is output to the LEDRs outside of the top level module.

The status of the lights on each edge are passed in along with the players' inputs and the reset input to two counter modules, one to keep track of each player's win count. These counters are responsible for updating their linked HEX display (HEX5 for the cyber player and HEX0 for the player) when their corresponding player has won.

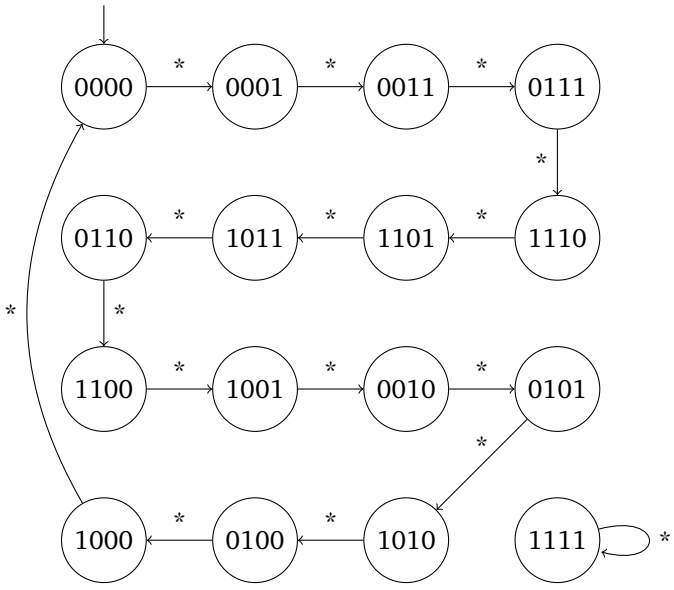
2. 3-Bit Counter Simulation

These are the resulting waves of a test bench to verify functionality of my 3-bit counter. The last combined wave is the binary output assigned to HEX display modules that have been predefined to display the numbers 0-7. The input is toggling after each rising clock edge, and each toggle on increments the counter at the next rising clock edge, and reset inputs will revert the output to the defined binary for 0.



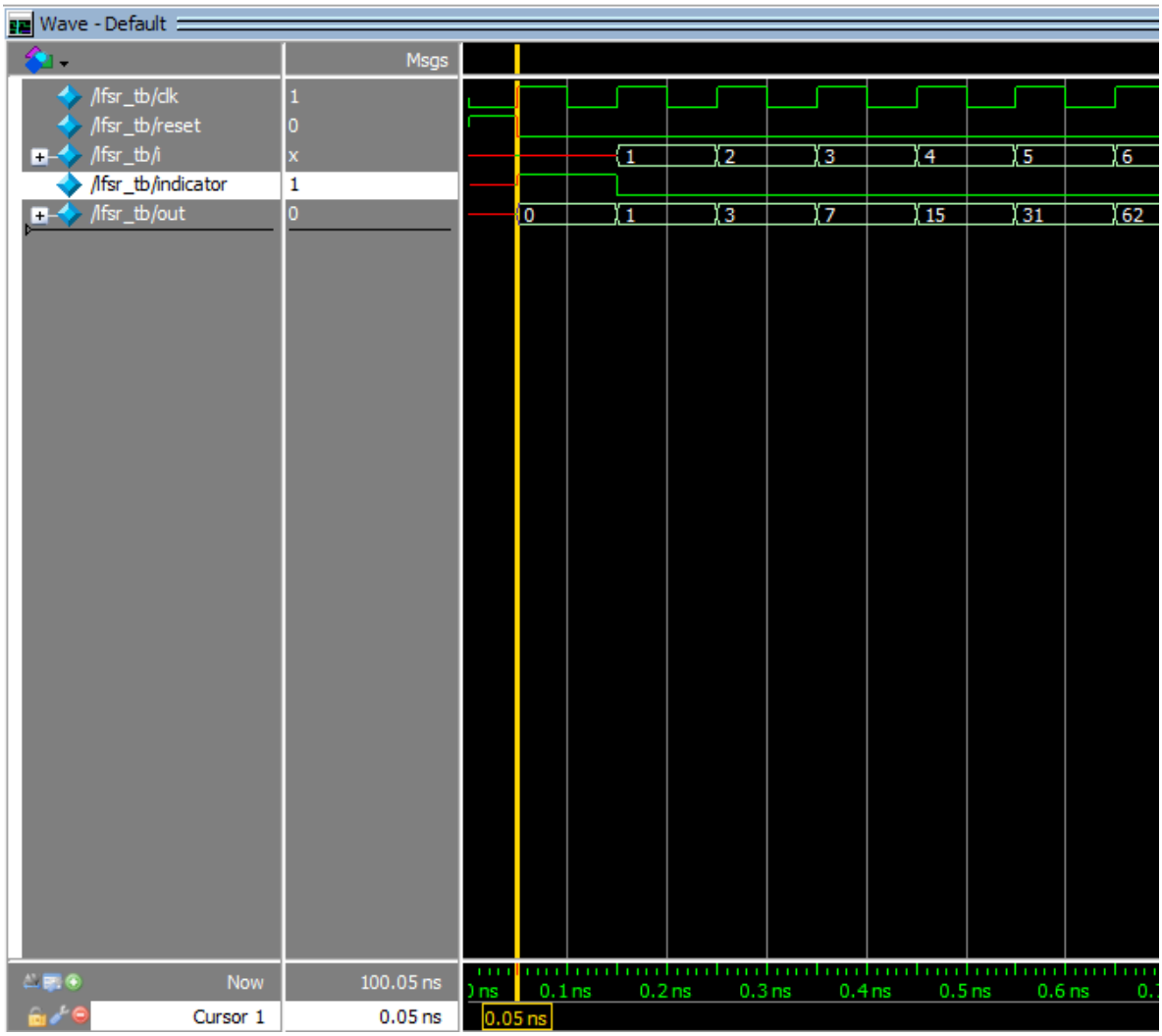
3. 4-Bit LFSR State Diagram

This is the state diagram for a 4-Bit Linear Feedback Shift Register. The next state is determined by shifting the 3 least significant bits to the 3 most significant bits and setting the least significant bit to the XNOR of the two most significant bits. In the diagram, the arrows with a * represent that the transition will always be taken. Note that the 1111 state is separate from the diagram and can only lead to itself.

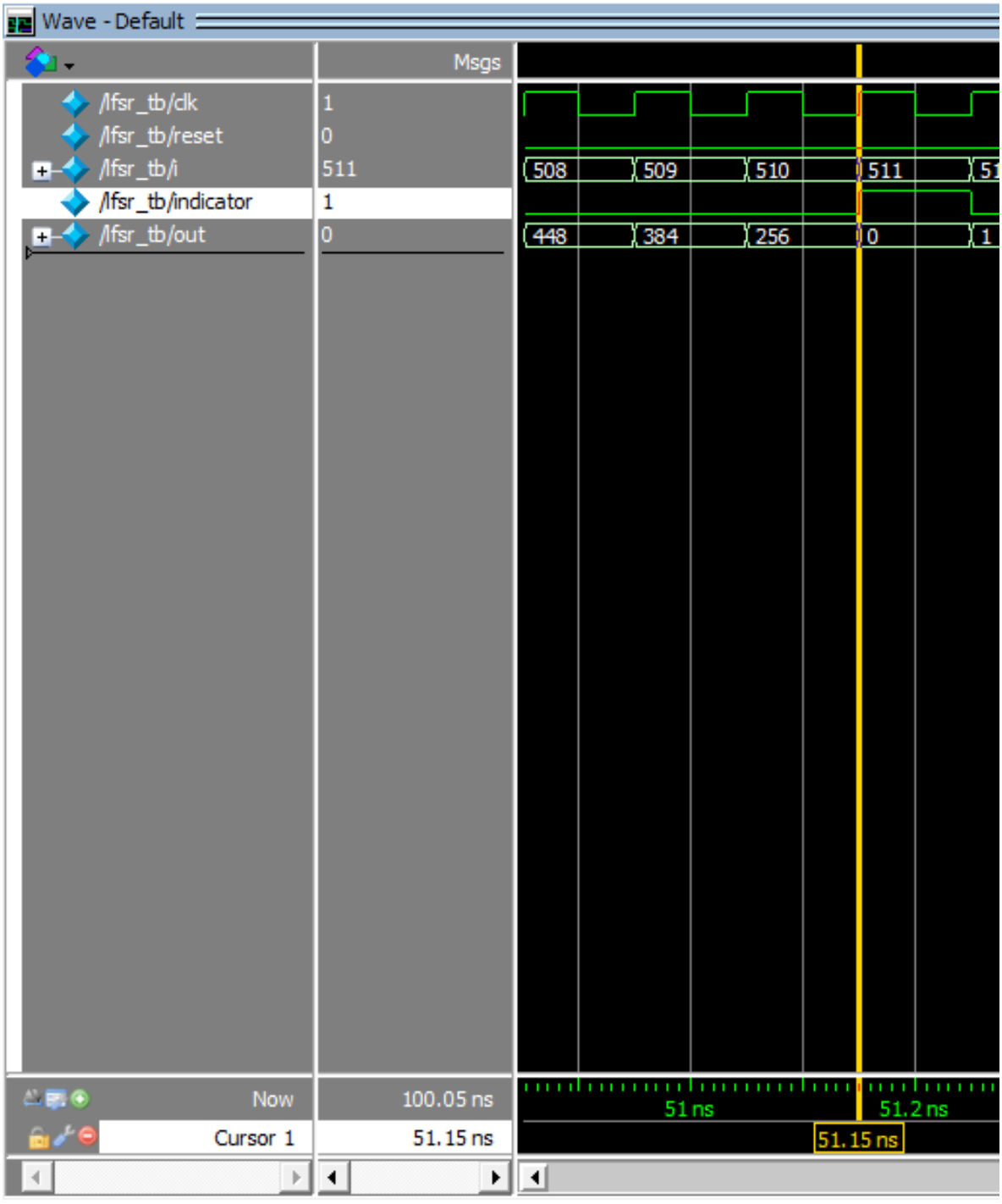


4. 9-Bit LFSR Simulation

The beginning of my LFSR simulation. You can see that i is counting the number of states after the initial state, for which I have an indicator signal set to be true only when the out state is 0, and these change every rising clock edge.



This diagram shows the next edge that my indicator is signal is true and the out state is once again 0. Here, it shows that i is now 511, indicating to us that the maximum state cycle length is 511 states.



5. 10-Bit Adder Simulation

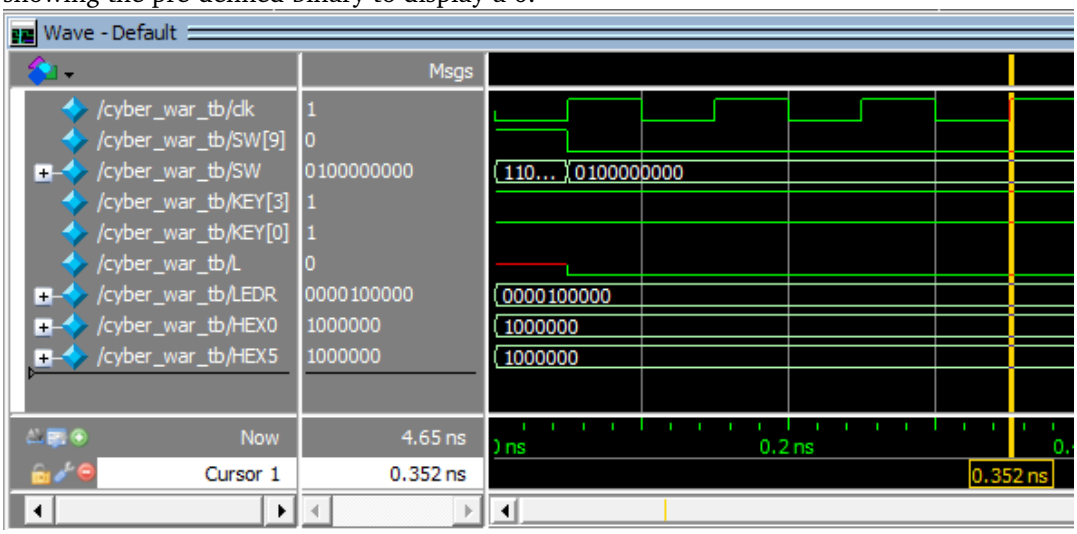
The top 3 waves are unsigned, the bottom 3 are the same values, just signed.

1) An addition with one input being 0: 0 + 1 = 0
2) An addition whose result is 511: 369 + 142 = 511
3) An addition whose result is 0: 369 + -369 = 0
4) An example of unsigned overflow: 369 + 655 = 0, 1023 + 1023 = 1022
5) An example of positive signed overflow: 511 + 511 = -2
6) An example of negative signed overflow: -512 + -512 = 0

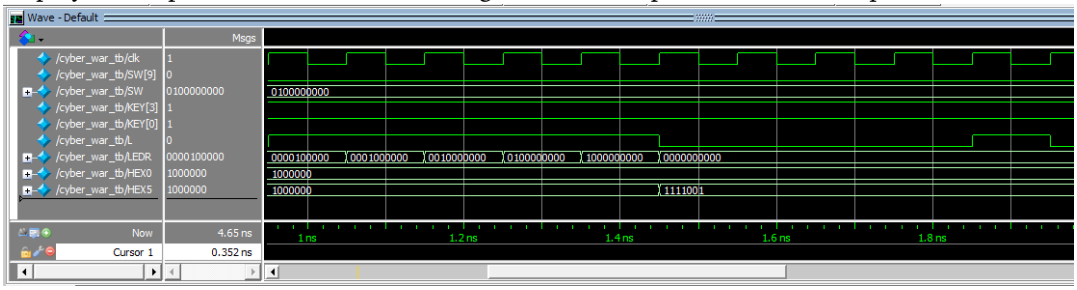
6. Top-Level ModelSim Simulation

As a reminder, SW[9] represents a hard reset (resets the game), KEY[3] represents a soft reset (resets the playing field), KEY[0] is the user input, SW[8-0] represents the difficulty of the Cyber player, and L is the Cyber player "input".

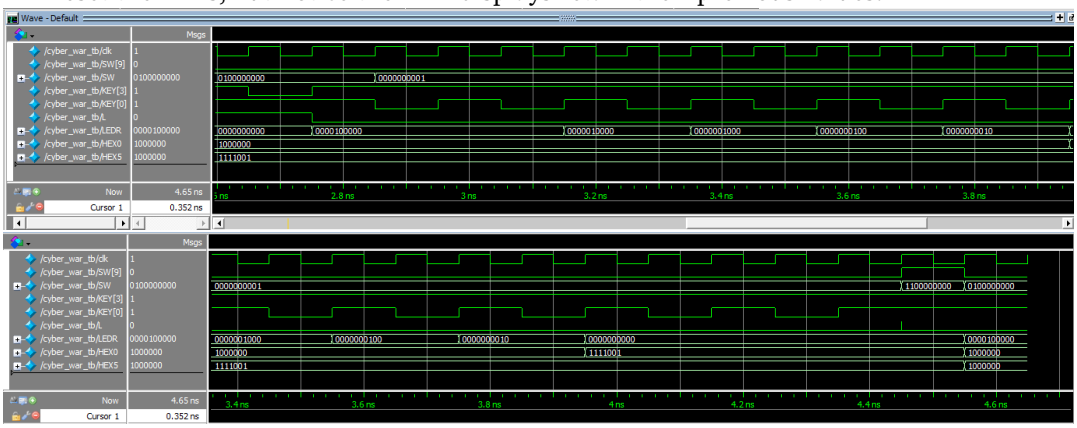
This is the wave diagram for the initial state of Cyber War, right after a reset input. In this scenario, I set the Cyber player's difficulty to 512/1023. LED 5 is on, the rest are inactive. Both HEX displays are showing the pre-defined binary to display a 0.



This is what a wave diagram of the Cyber player winning looks like. The Cyber player pseudo-randomly "presses" an input and the LED status light moves to the left at the next clock cycle after each input (since we don't have to synchronize the Cyber player's inputs). After the status light moves off of the final LED position, the HEX display for the cyber player will increment to the pre-defined binary to display a 1. Inputs after a win don't change the LED output or the HEX output.



This is a wave diagram of Player winning. The same thing happens, just in reverse. The Cyber player has been slowed down to 1/1023, and does not make an input in this sequence. The LED status light starts in the center and moves to the right. After it falls off the right, the HEX display for the player will change to display a 1. Again, any inputs after a victory will not change any displays. A soft reset input will reset the LEDs, but notice the HEX displays retain their previous values.



7. Resource Utilization

Analysis & Synthesis Resource Utilization by Entity

<<Filter>>

	Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Block Memory Bits	DSP Blocks	Pins	Virtual Pins	Full Hierarchy Name	Entity Name	Library Name
1	|cyber_war	66 (1)	47 (0)	0	0	40	0	|cyber_war	cyber_war	work
1	|adder:add|	10 (10)	0 (0)	0	0	0	0	|cyber_war|adder:add	adder	work
2	|clock_divider:cdiv|	20 (20)	20 (20)	0	0	0	0	|cyber_war|...ivider:cdiv	clock_divider	work
3	|counter:l_count|	11 (11)	3 (3)	0	0	0	0	|cyber_war|...ter:l_count	counter	work
4	|counter:r_count|	11 (11)	3 (3)	0	0	0	0	|cyber_war|...ter:r_count	counter	work
5	|edge_detector:right|	2 (2)	1 (1)	0	0	0	0	|cyber_war|...ector:right	edge_detector	work
6	|lfsr:r|	1 (1)	9 (9)	0	0	0	0	|cyber_war|lfsr:r	lfsr	work
7	|light:L1|	1 (1)	1 (1)	0	0	0	0	|cyber_war|light:L1	light	work
8	|light:L2|	1 (1)	1 (1)	0	0	0	0	|cyber_war|light:L2	light	work
9	|light:L3|	1 (1)	1 (1)	0	0	0	0	|cyber_war|light:L3	light	work
10	|light:L4|	1 (1)	1 (1)	0	0	0	0	|cyber_war|light:L4	light	work
11	|light:L5|	1 (1)	1 (1)	0	0	0	0	|cyber_war|light:L5	light	work
12	|light:L6|	1 (1)	1 (1)	0	0	0	0	|cyber_war|light:L6	light	work
13	|light:L7|	1 (1)	1 (1)	0	0	0	0	|cyber_war|light:L7	light	work
14	|light:L8|	1 (1)	1 (1)	0	0	0	0	|cyber_war|light:L8	light	work
15	|light:L9|	1 (1)	1 (1)	0	0	0	0	|cyber_war|light:L9	light	work
16	|sync:sync|	1 (0)	2 (0)	0	0	0	0	|cyber_war|sync:sync	sync	work
1	|d_ff:d0|	1 (1)	1 (1)	0	0	0	0	|cyber_war|...ync|d_ff:d0	d_ff	work
2	|d_ff:d1|	0 (0)	1 (1)	0	0	0	0	|cyber_war|...ync|d_ff:d1	d_ff	work

8. Misc.

How many hours (estimated) it took to complete this lab in total, including reading, planning, designing, coding, debugging, and testing.

It took around 10 hours to complete this lab.