CE100 Lab Report 6

Turkey Traffic

Justin Fortner

Lab 1C

TU/TH 1:30-3:20

11/22/17

**Description:**

This lab provides the task of designing a state machine that keeps track of where turkeys are located. This is accomplished when a turkey travels to the left a number is subtracted from the total and when a turkey moves to the right a number is added to the total. The purpose of this lab is to reiterate our knowledge of state machines and allow us to hone in our knowledge of verilog.

**Methods:**

**Part 1 - System:**

The state machine for this lab needed to handle a variety of different cases. These included turkeys crossing one or two sensors and changing its mind. Turkeys crossing from right to left as well as left to right. Turkeys getting stuck in the left, middle or right of the sensors. A state was given to handle each one of these cases. These states all branched off of an initial start state where the time and score counters do nothing. These states were formed through one hot encoding.

**Part 2 - Turkey Counter:**

The turkey counter is an eight bit up down counter capable of counting from -127 to 127. This is accomplished using 8 flip flops linked together in series with logic attached to the inputs that would provide the correct outputs.

**Part 3 - Time Counter:**

The time counter is a five bit up counter. The qsec signal is the clock for this module. Every increment in the count represents a quarter second. For this lab we needed a two second and an eight second counter. A high output was given when the counter reached 10001 to represent four seconds and 101 to represent one second.

**Part 4 - Sensor Inputs:**

The sensor inputs are given by btnL and btnR. When the button is pressed the corresponding led is turned off. This is accomplished by inverting the signal of the button to be the input for the LED. The sensor inputs are also the only input allowed to change the states outside of the time counter counting to four.

**Part 5 - Top Level:**

The top level for this lab tied all the different modules of this lab together. Allowed for the input of the buttons to simulate turkeys. Displayed the total number of turkeys. As well as keeping track of if there were more on the left or right by use of the negative side. Lastly this module contains the logic to display the numbers 1 through 4 when a sensor is blocked.

**Results:**

**Design:**

**Part 1 - System:**

The state machine starts in state D0. This state can be returned to at any time when none of the sensors are no longer blocked. The score is being shown in this state and both LEDs are on. The time counter is off.

If a turkey begins to cross from the left state D1 is entered. The turkey can change its mind and go back to D0 or continue through the sensors to D4. The score is being displayed but the LED corresponding to the blocked sensor is off. The timer begins to count to four.

If a turkey begins to cross from the right state D2 is entered. The turkey can change its mind and go back to D0 or continue through the sensors to D5. The score is being displayed but the LED corresponding to the blocked sensor is off. The timer begins to count to four.

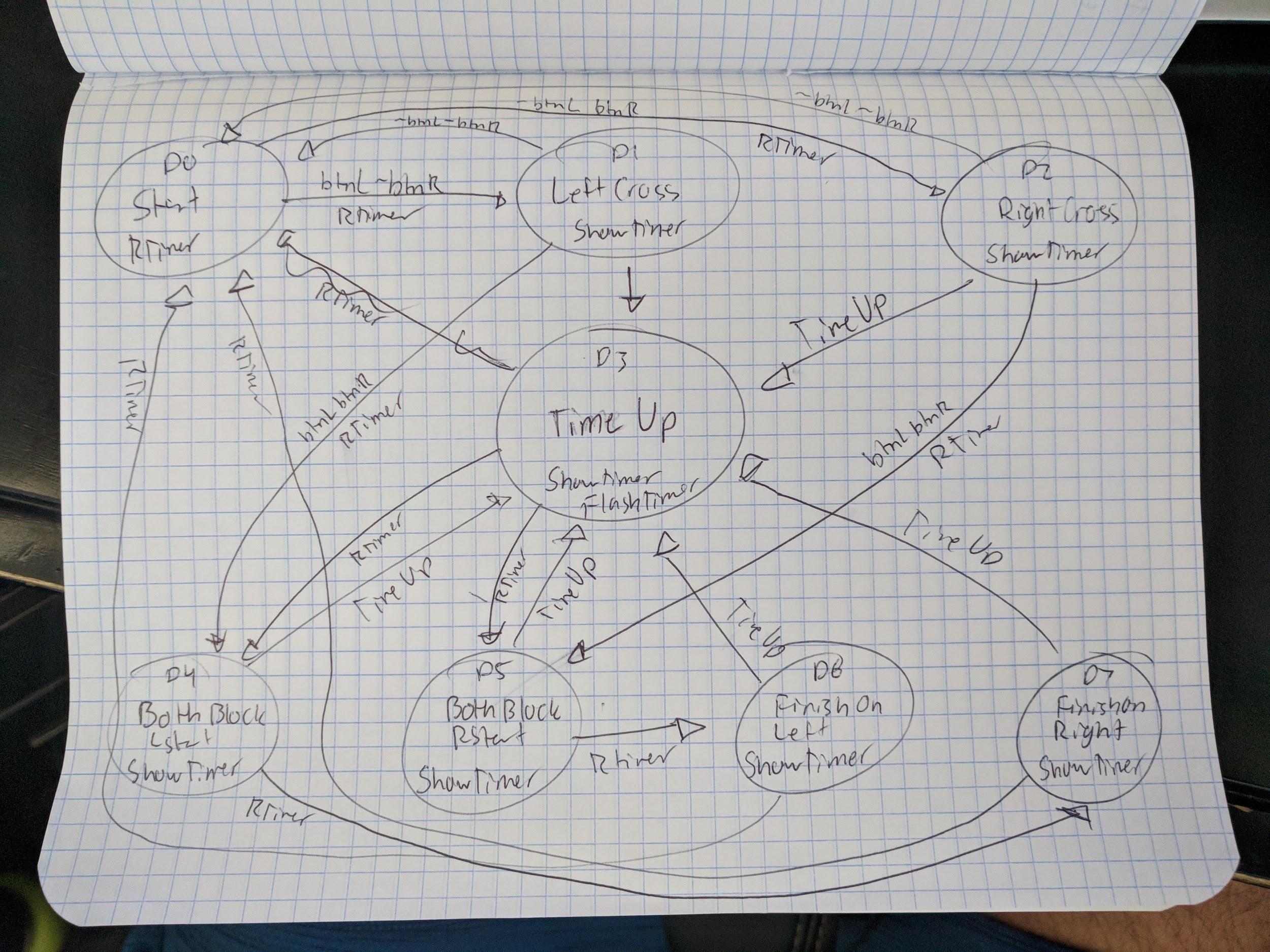
If a turkey blocks a sensor for too long they enter state D3. In this state the timer begins to blink. If a turkey moves on from this state the timer is reset.

If a turkey is blocking both sensors and starts on the left it will enter state D4. The turkey can move onto state D7 or change its mind and go back to state D1. The score is being displayed. Both LEDs are off. The timer begins to count to four.

If a turkey is blocking both sensors and starts on the right it will enter state D5. The turkey can move onto state D6 or change its mind and go back to state D2. The score is being displayed. Both LEDs are off. The timer begins to count to four.

If a turkey began on the left and is now completing his journey through the sensors state D7 is entered. The turkey can change its mind and go back to D4 or continue through the sensors to complete the journey. The score is being displayed but the LED corresponding to the blocked sensor is off. The timer begins to count to four.

If a turkey began on the right and is now completing his journey through the sensors state D6 is entered. The turkey can change its mind and go back to D5 or continue through the sensors to complete the journey. The score is being displayed but the LED corresponding to the blocked sensor is off. The timer begins to count to four.



//start

D[0] = (Q[0] & ~LBlock & ~RBlock)|(Q[3] & ~LBlock & ~RBlock)|(Q[1] & ~LBlock)|(Q[2] & ~LBlock)|(Q[4]& ~LBlock & ~RBlock)|(Q[5]& ~LBlock & ~RBlock)|(Q[6] & ~LBlock & ~RBlock)|(Q[7] & ~LBlock & ~RBlock);

//left crossed first

D[1] = (Q[0] & LBlock & ~RBlock & ~TimeUp)|(Q[1] & LBlock & ~RBlock & ~TimeUp)|(Q[4] & LBlock & ~RBlock & ~TimeUp);

//right crossed first

D[2] = (Q[0] & ~LBlock & RBlock & ~TimeUp)|(Q[2] & ~LBlock & RBlock & ~TimeUp)|(Q[5] & ~LBlock & RBlock & ~TimeUp);

//time ups

D[3] = (Q[1] & LBlock & ~RBlock & TimeUp)|(Q[2] & ~LBlock & RBlock & TimeUp)|(Q[4] & LBlock & RBlock & TimeUp)|(Q[3] & (LBlock|RBlock))|(Q[6] & LBlock & ~RBlock & TimeUp)|(Q[7] & ~LBlock & RBlock & TimeUp);

//both block when start left

D[4] = (Q[1] & LBlock & RBlock & ~TimeUp)|(Q[4]& & LBlock & RBlock & ~TimeUp)|(Q[7] & LBlock & RBlock & ~TimeUp);

//both block when start right

D[5] = (Q[2] & LBlock & RBlock & ~TimeUp)|(Q[5]& & LBlock & RBlock & ~TimeUp)|(Q[6] & LBlock & RBlock & ~TimeUp);

//finish left

D[6] = (Q[5] & LBlock & ~RBlock & ~TimeUp)|(Q[6] & LBlock & ~RBlock & ~TimeUp);

//finish right

D[7] = (Q[4] & ~LBlock & RBlock & ~TimeUp)|(Q[7] & ~LBlock & RBlock & ~TimeUp);

ShowTimer = Q[1]|Q[2]|Q[3]|Q[4];

RTimer = Q[0]|(Q[0] & LBlock & ~RBlock & ~TimeUp)|(Q[4] & LBlock & ~RBlock & ~TimeUp)|(Q[0] & ~LBlock & RBlock & ~TimeUp)|(Q[5] & ~LBlock & RBlock & ~TimeUp)|(Q[1] & LBlock & ~RBlock & TimeUp)|(Q[4] & LBlock & RBlock & TimeUp)|(Q[3] & (LBlock|RBlock))|(Q[6] & LBlock & ~RBlock & TimeUp)|(Q[7] & ~LBlock & RBlock & TimeUp)|

(Q[1] & LBlock & RBlock & ~TimeUp)|(Q[7] & LBlock & RBlock & ~TimeUp)|(Q[2] & LBlock & RBlock & ~TimeUp)|(Q[6] & LBlock & RBlock & ~TimeUp)|(Q[5] & LBlock & ~RBlock & ~TimeUp)|(Q[4] & ~LBlock & RBlock & ~TimeUp);

FlashTimer = Q[3];

crossComp = (Q[6] & ~LBlock & ~RBlock)|(Q[7] & ~LBlock & ~RBlock);

crossDirection = Q[6];

**Part 2 - Turkey Counter:**

This 8 bit turkey counter is comprised of two 4 bit counters. The second counter only counts up when the first is full and the first only counts down when the second is empty. This allows for a successful 8 bit up down counter.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UD** | **Q3** | **Q2** | **Q1** | **Q0** | **Q3+** | **Q2+** | **Q1+** | **Q0+** |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UD** | **Q3** | **Q2** | **Q1** | **Q0** | **Q3+** | **Q2+** | **Q1+** | **Q0+** |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

wD[0] = ~c[0];

wD[1] = (~UD&((~c[1]&~c[0])|(c[1]&c[0])))|(UD&((~c[1]&c[0])|(c[1]&~c[0])));

wD[2] = (~UD&((c[2]&c[0])|(c[2]&c[1])|(~c[2]&~c[1]&~c[0])))|(UD&((c[2]&~c[1])|(c[2]&~c[0])|(~c[2]&c[1]&c[0])));

wD[3] = (~UD&((c[3]&c[0])|(c[3]&c[1])|(c[3]&c[2])|(~c[3]&~c[2]&~c[1]&~c[0])))|(UD&((c[3]&~c[2])|(c[3]&~c[1])|(c[3]&~c[0])|(~c[3]&c[2]&c[1]&c[0])));

UStop = UD & ~c[7] & c[6] & c[5] & c[4] & c[3] & c[2] & c[1] & c[0];

DStop = ~UD & c[7] & ~c[6] & ~c[5] & ~c[4] & ~c[3] & ~c[2] & ~c[1] & c[0];

RTC = wRTC;

TC = wLTC & wRTC;

Z = wLZ & wRZ;

**Part 3 - Time Counter:**

Equations are obtained from the table below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CE** | **Q5** | **Q4** | **Q3** | **Q2** | **Q1** | **Q0** | **Q5+** | **Q4+** | **Q3** | **Q2+** | **Q1+** | **Q0+** |
| **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |
| **1** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **1** | **0** |
| **1** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **1** |
| **1** | **0** | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **0** | **1** | **0** | **0** |
| **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **1** |
| **1** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **1** | **1** | **0** |
| **1** | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **1** | **1** | **1** |
| **1** | **0** | **0** | **0** | **1** | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **1** |
| **1** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | **0** |
| **1** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **1** | **1** |
| **1** | **0** | **0** | **1** | **0** | **1** | **1** | **0** | **0** | **1** | **1** | **0** | **0** |
| **1** | **0** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **1** | **1** | **0** | **1** |
| **1** | **0** | **0** | **1** | **1** | **0** | **1** | **0** | **0** | **1** | **1** | **1** | **0** |
| **1** | **0** | **0** | **1** | **1** | **1** | **0** | **0** | **0** | **1** | **1** | **1** | **1** |
| **1** | **0** | **0** | **1** | **1** | **1** | **1** | **0** | **1** | **0** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **1** | **0** |
| **1** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **1** |
| **1** | **0** | **1** | **0** | **0** | **1** | **1** | **0** | **1** | **0** | **1** | **0** | **0** |
| **1** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **1** |
| **1** | **0** | **1** | **0** | **1** | **0** | **1** | **0** | **1** | **0** | **1** | **1** | **0** |
| **1** | **0** | **1** | **0** | **1** | **1** | **0** | **0** | **1** | **0** | **1** | **1** | **1** |
| **1** | **0** | **1** | **0** | **1** | **1** | **1** | **0** | **1** | **1** | **0** | **0** | **0** |
| **1** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **1** |
| **1** | **0** | **1** | **1** | **0** | **0** | **1** | **0** | **1** | **1** | **0** | **1** | **0** |
| **1** | **0** | **1** | **1** | **0** | **1** | **0** | **0** | **1** | **1** | **0** | **1** | **1** |
| **1** | **0** | **1** | **1** | **0** | **1** | **1** | **0** | **1** | **1** | **1** | **0** | **0** |
| **1** | **0** | **1** | **1** | **1** | **0** | **0** | **0** | **1** | **1** | **1** | **0** | **1** |
| **1** | **0** | **1** | **1** | **1** | **0** | **1** | **0** | **1** | **1** | **1** | **1** | **0** |
| **1** | **0** | **1** | **1** | **1** | **1** | **0** | **0** | **1** | **1** | **1** | **1** | **1** |
| **1** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** |
| **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |

wD[0] = (~CE&t[0])|(CE&~t[0]);

wD[1] = (~CE&t[1])|(t[1]&~t[0])|(CE&~t[1]&t[0]);

wD[2] = (~CE&t[2])|(t[2]&~t[1])|(t[2]&~t[0])|(CE&~t[2]&t[1]&t[0]);

wD[3] = (~CE&t[3])|(t[3]&~t[2])|(t[3]&~t[1])|(t[3]&~t[0])|(CE&~t[3]&t[2]&t[1]&t[0]);

wD[4] = (~CE&t[4])|(t[4]&~t[3])|(t[4]&~t[2])|(t[4]&~t[1])|(t[4]&~t[0])|(CE&~t[4]&t[3]&t[2]&t[1]&t[0]);

wD[5] = (~CE&t[5])|(t[5]&~t[4])|(t[5]&~t[3])|(t[5]&~t[2])|(t[5]&~t[1])|(t[5]&~t[0])|(CE&~t[5]&t[4]&t[3]&t[2]&t[1]&t[0]);

TimeUp = ~t[5] & t[4] & ~t[3] & ~t[2]& ~t[1]& ~t[0];

OneSec = ~t[5] & ~t[4] & ~t[3] & t[2]& ~t[1]& t[0];

**Part 4 - Sensor Inputs:**

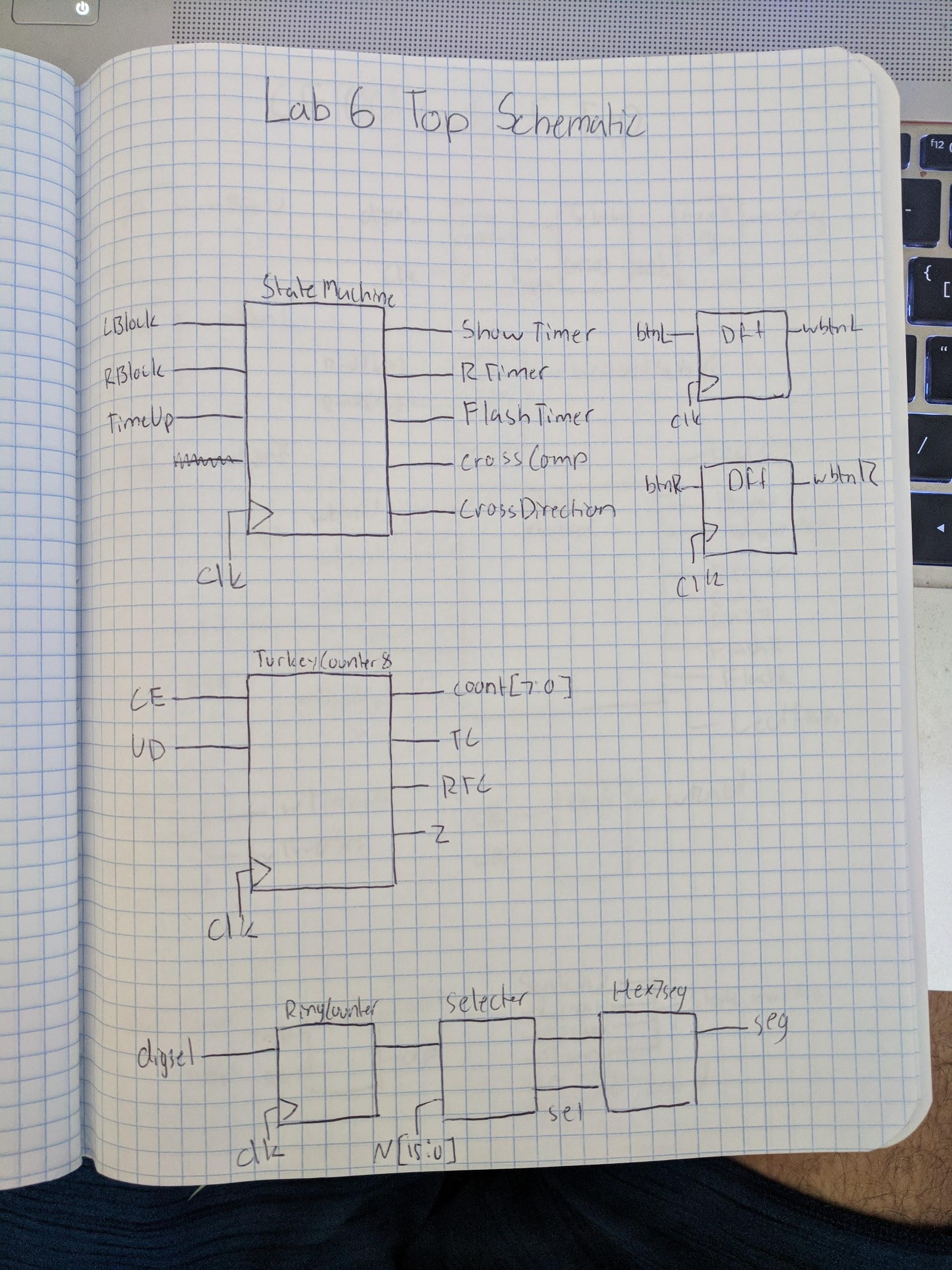
In order to synchronize the sensor inputs with the rest of the design the inputs are passed through their own flip flops. To indicate that these sensors are blocked an led is turned off. This is accomplished by inverting the signal from the button and passing that along to the led.

led[15] = ~btnL;

led[9] = ~btnR;

**Part 5 - Top Level:**

The top level for this lab tied all the different modules of this lab together. Allowed for the input of buttons. Kept track of the turkeys. Also within this top module is converting the turkey count from 2s complement to its positive counterpart if need be. Lastly this module contains the logic to display a negative sign when a negative count occurs and flash the timer when time runs out.



an [3] = ~((wShowTimer & wRingCounterOut[3] & ~wFour)|(wRingCounterOut[3] & wShowTimer & wTimeCounterOut1[1] & wFour));

an [2] = ~(wShowNeg & wRingCounterOut[2]);

an [1] = ~wRingCounterOut[1];

an [0] = ~wRingCounterOut[0];

**Testing & Simulation:**

This lab was tested using incremental testing. Each individual portion of this lab was tested before moving onto the next. This would ensure that I would always know where issues were located when they arose. My state machine was simulated most thoroughly in order to ensure that I would not enter two states at once and that there was no input that was given that would bring me outside of the state machine.

**Lab Questions:**

No questions for this lab.

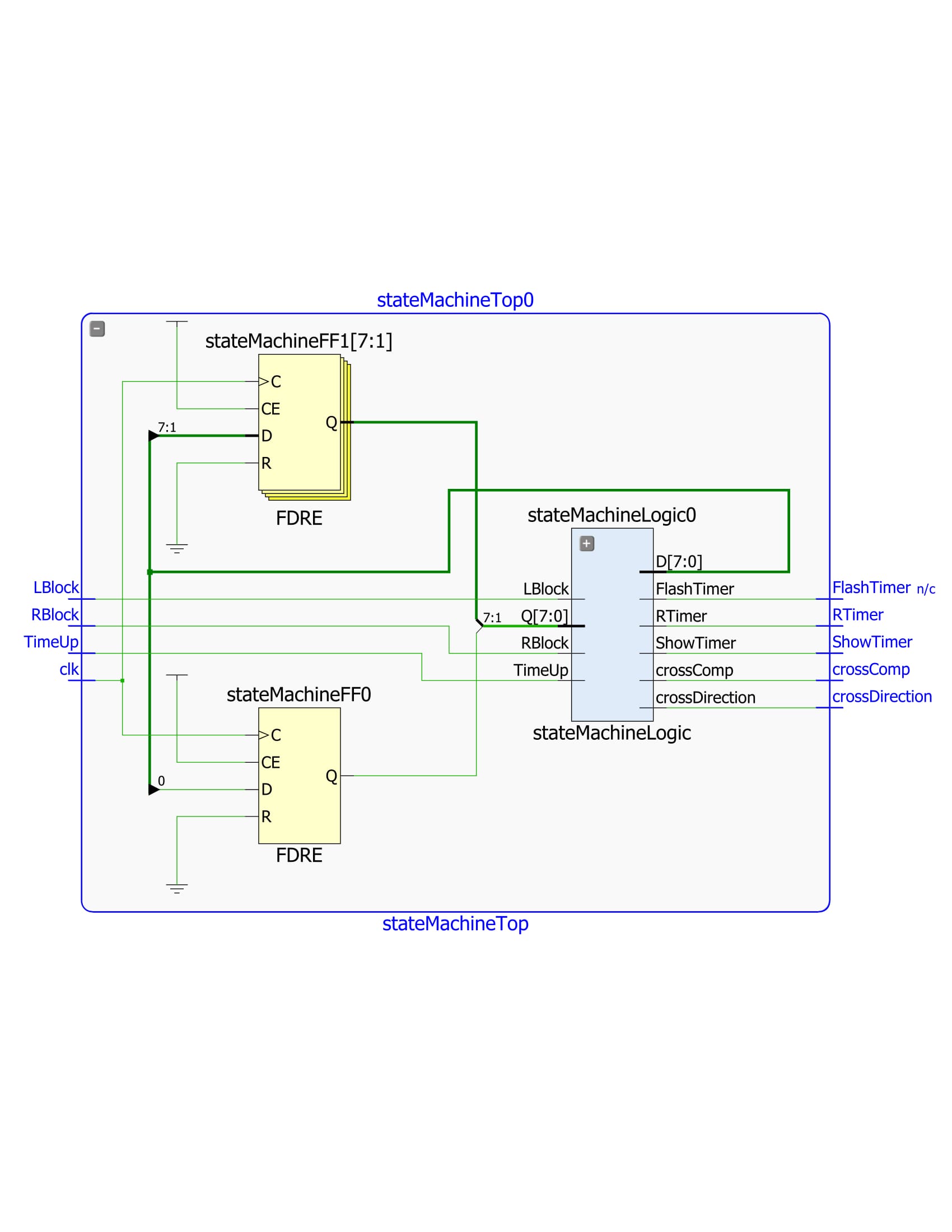
**Conclusion:**

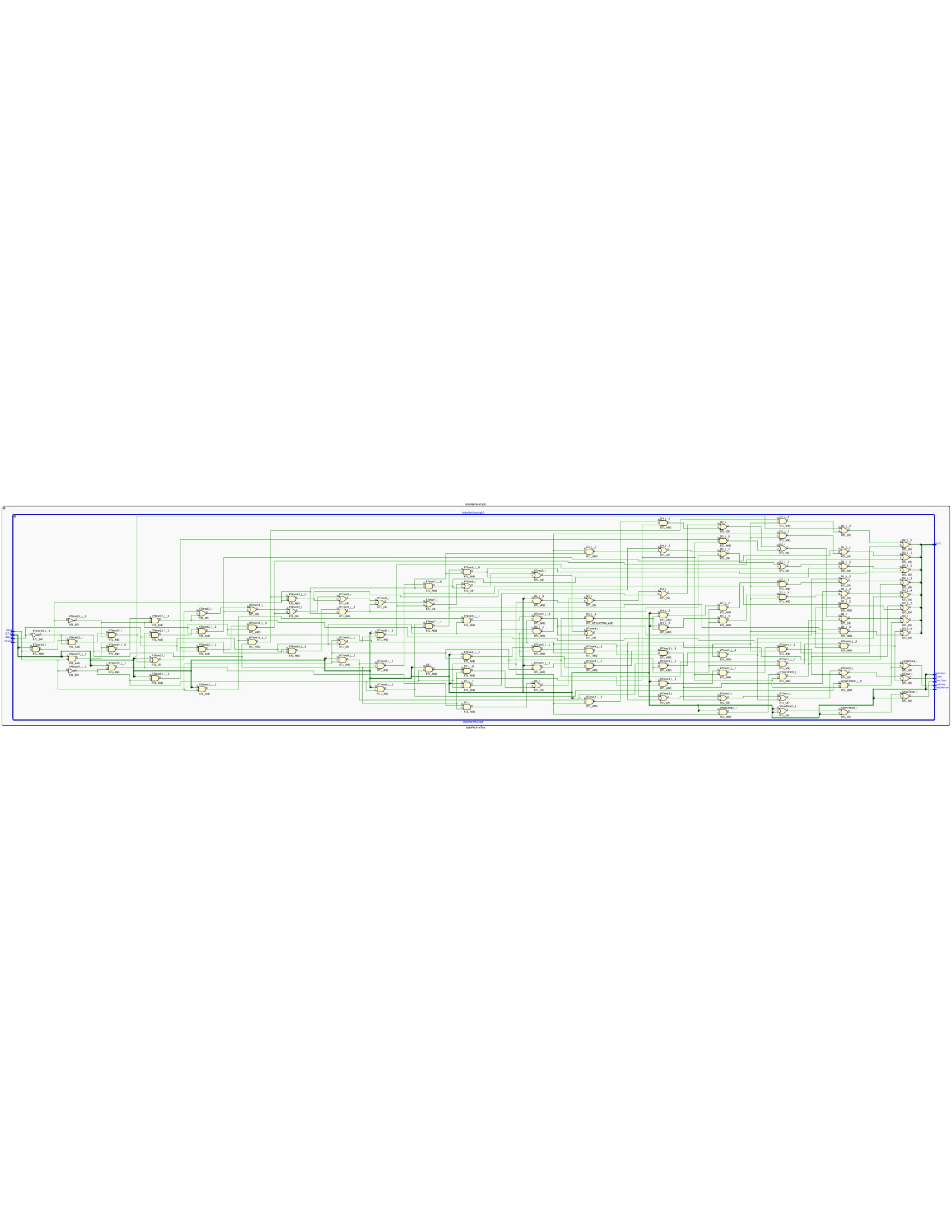
All in all this was a great lab. Not very many new skills were implemented in this lab, but many skills were reinforced. These skills included creating a state machine, creating a larger counter out of smaller ones, using inputs from the board to change states, and dealing with changing two's complement numbers into regular binary. Thankfully I did not have very many troubles with this lab. After a rather difficult lab 5 I was given less time than normal to complete this lab. I managed to complete this lab in an appropriate amount of time and only used one late day. I wish I had a better method to convert two's complement, but that is the only aspect of this lab that I would optimize.

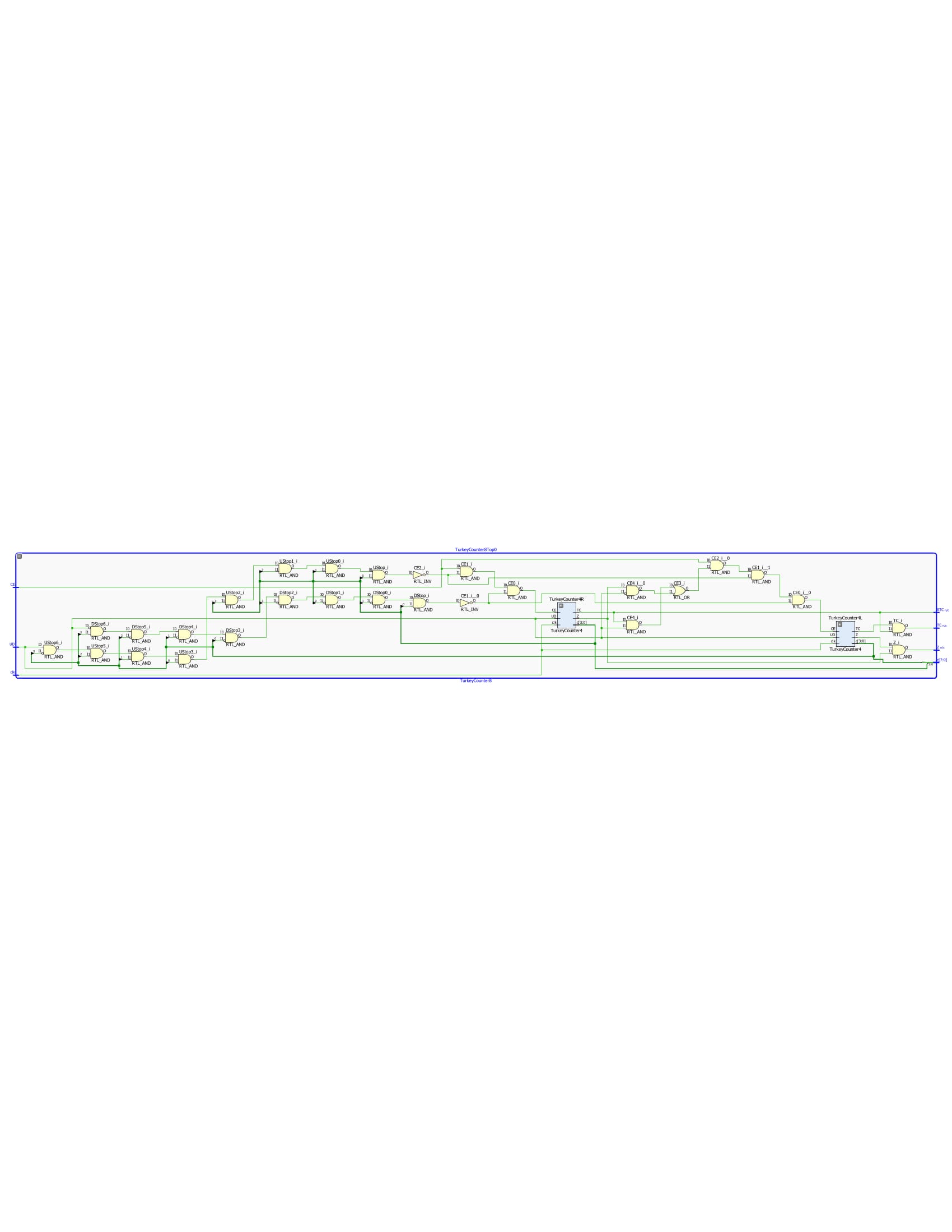
**Appendix:**

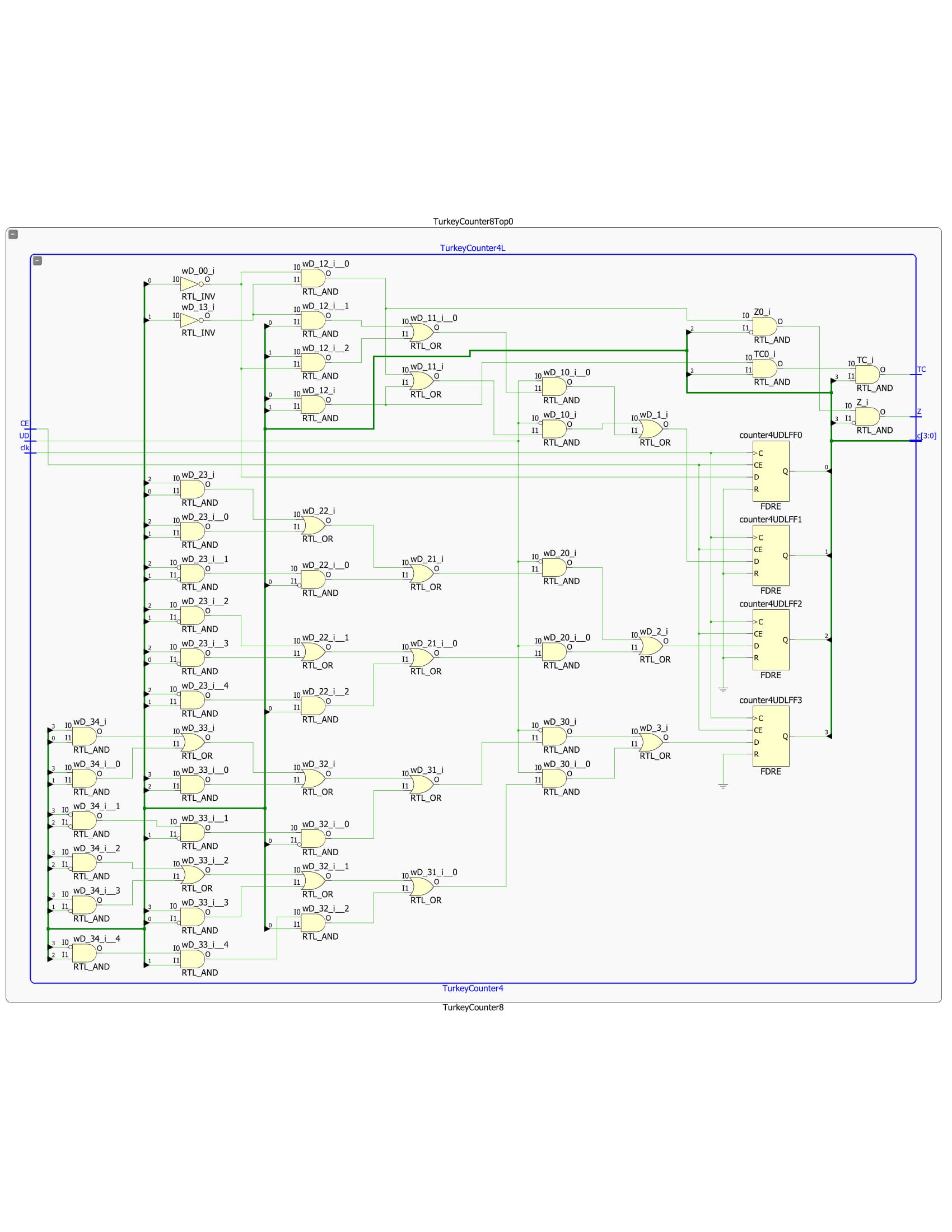
**Schematics:**

**Part 1 - System:**

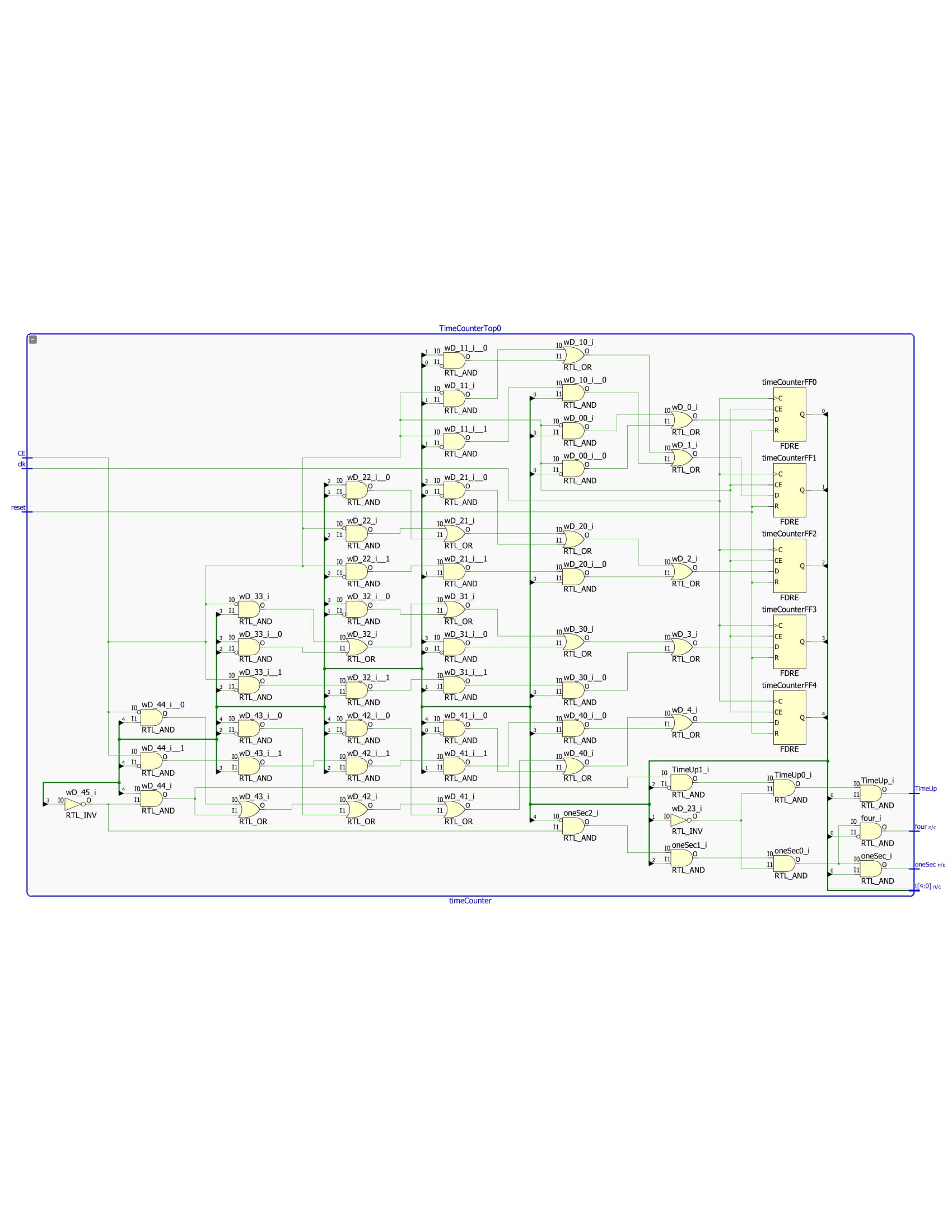
****

****

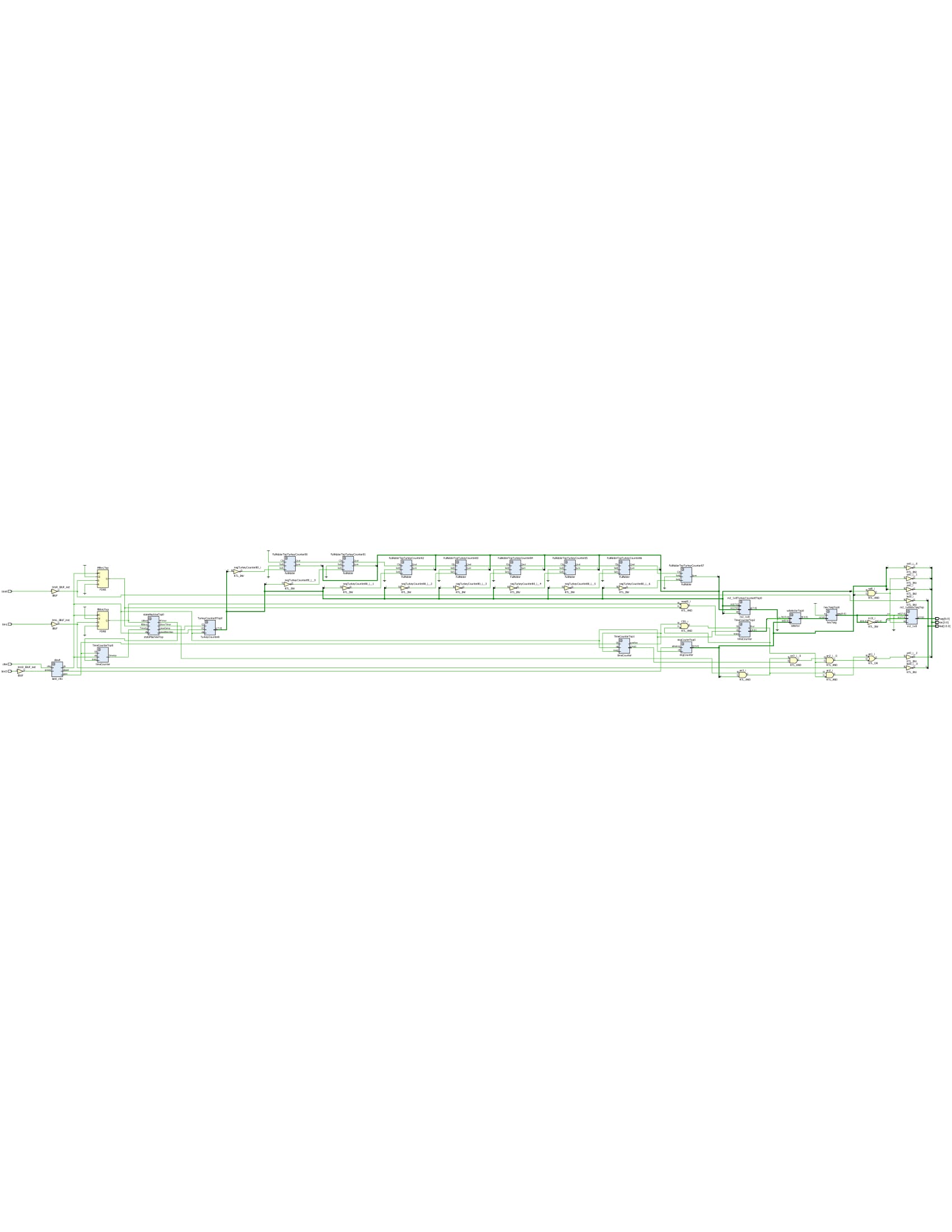
**Part 2 - Turkey Counter:**



**Part 3 - Time Counter:**

****

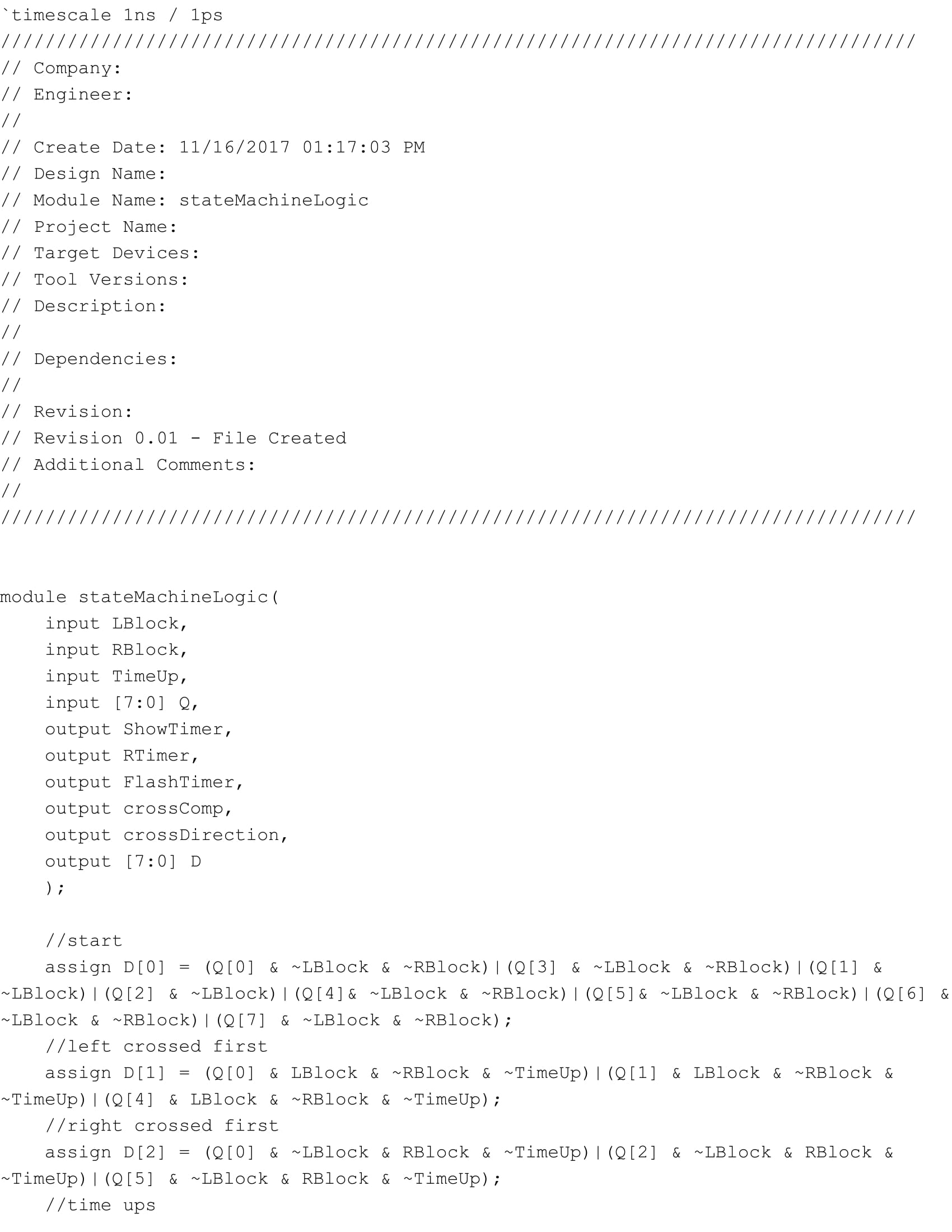
**Part 5 - Top Level:**

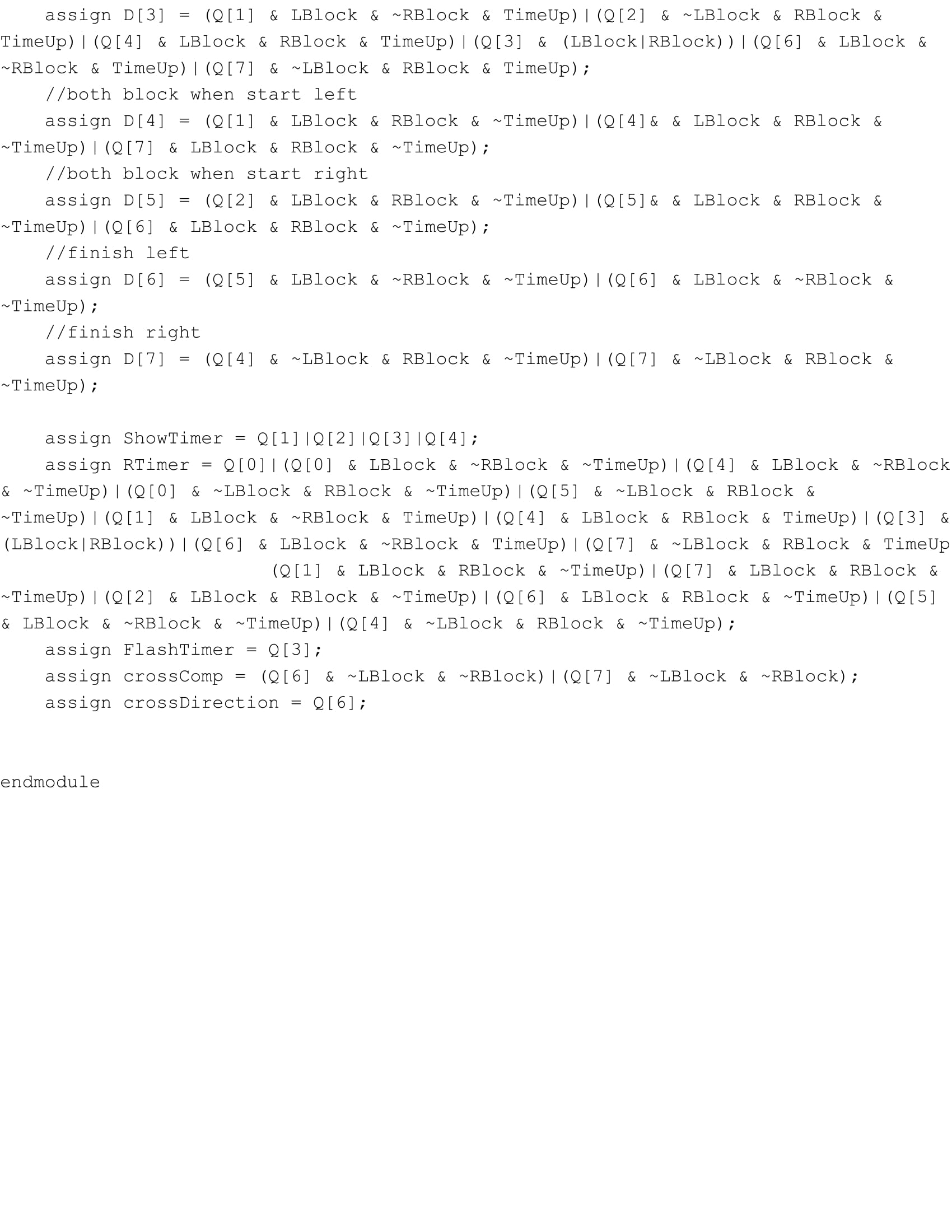
****

**Verilog:**

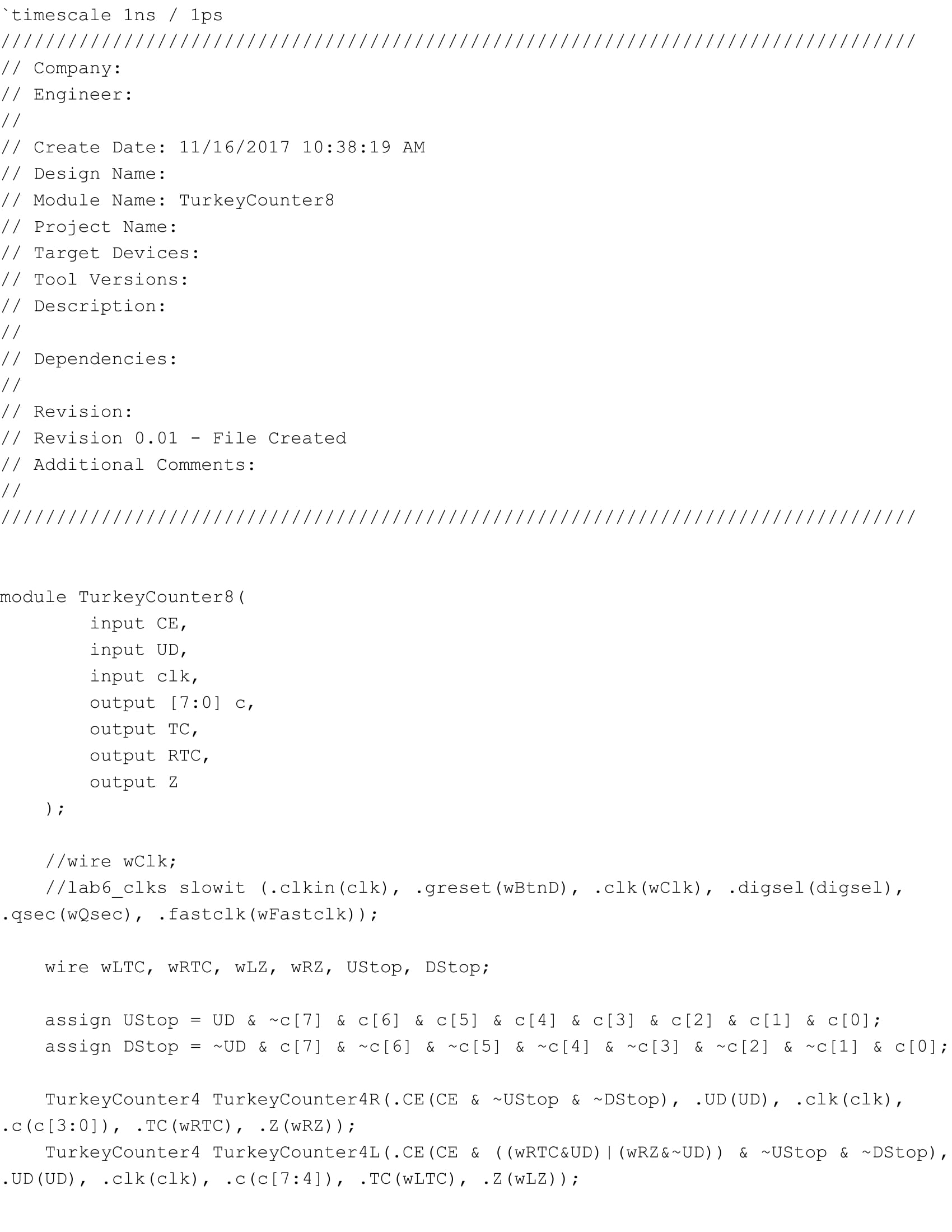
**Part 1 - System:**

****

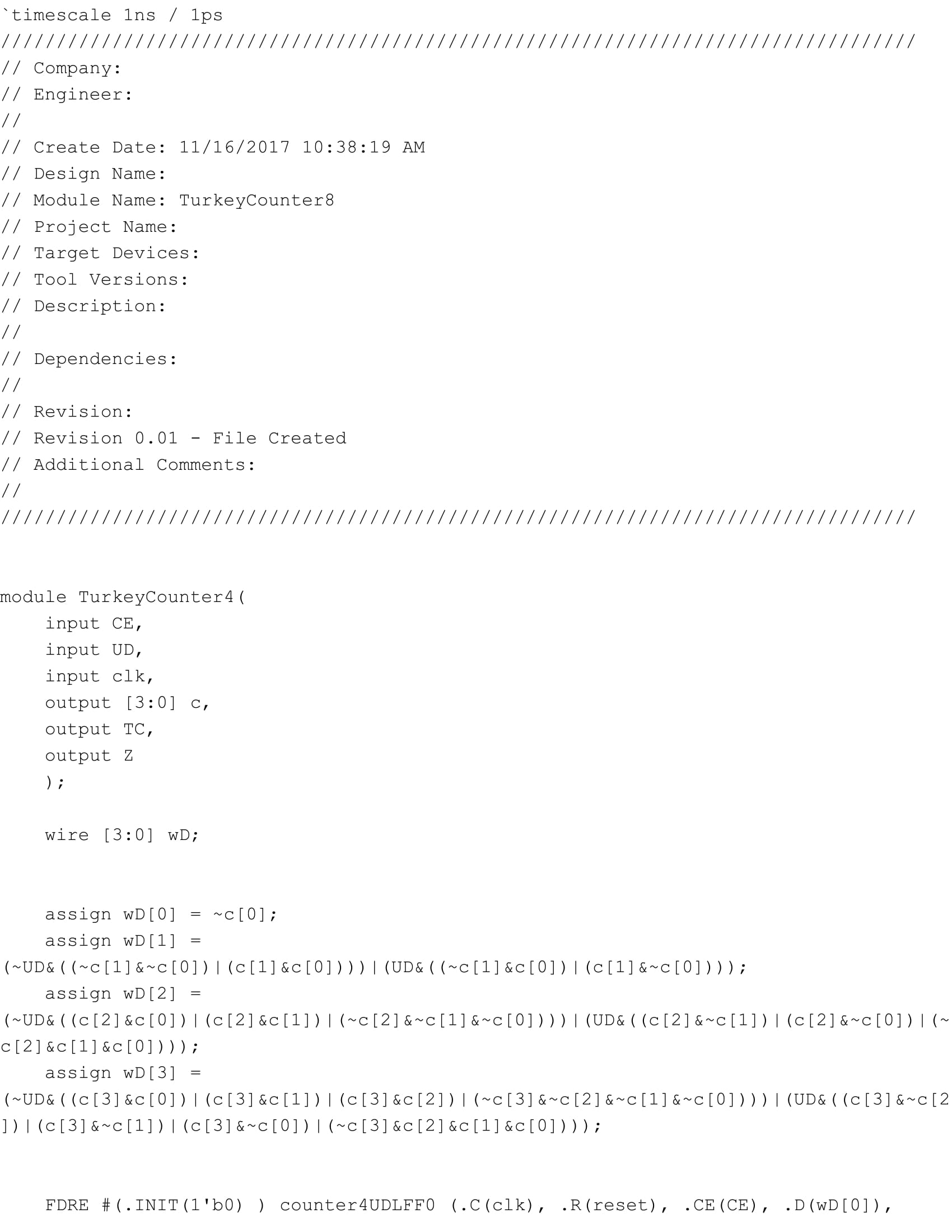
****

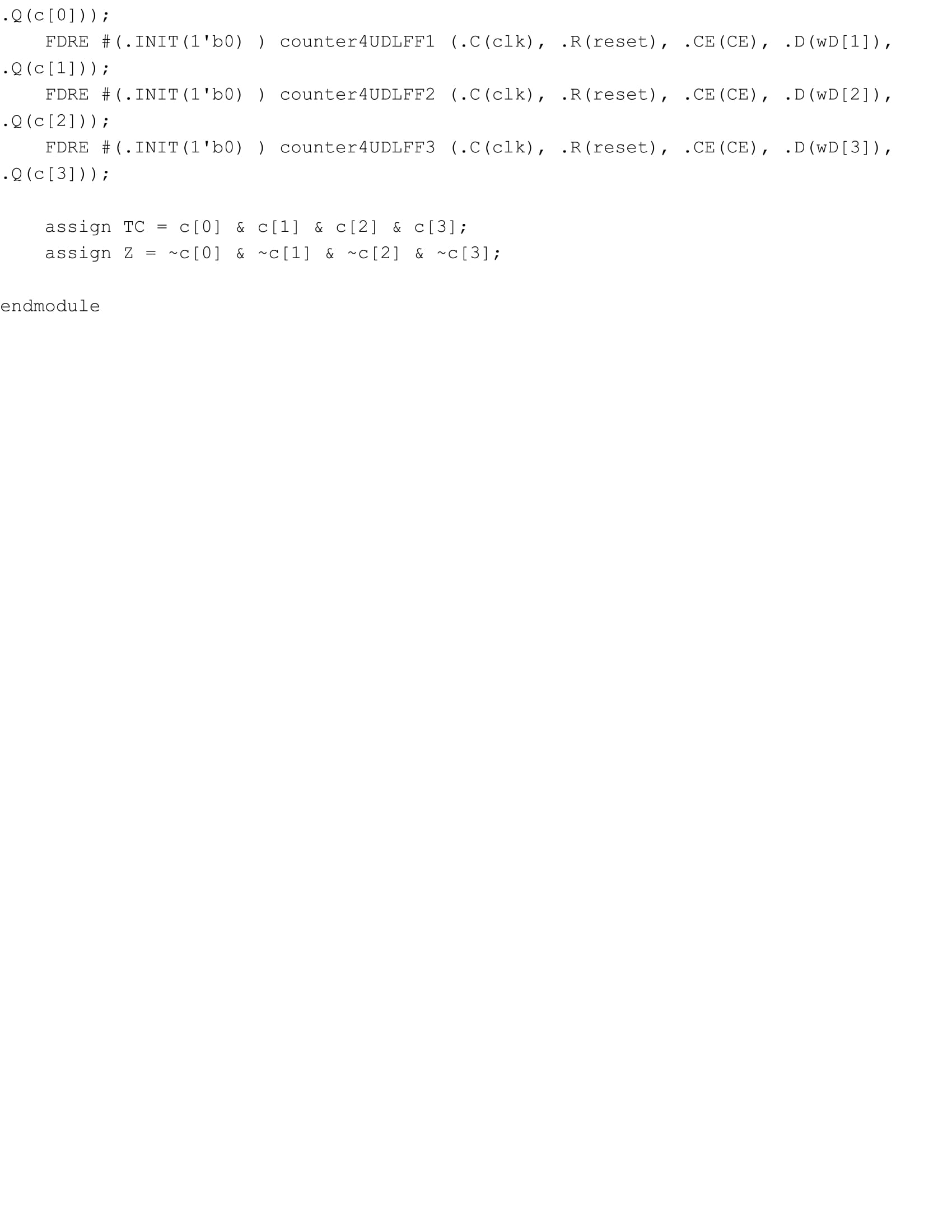
****

**Part 2 - Turkey Counter:**

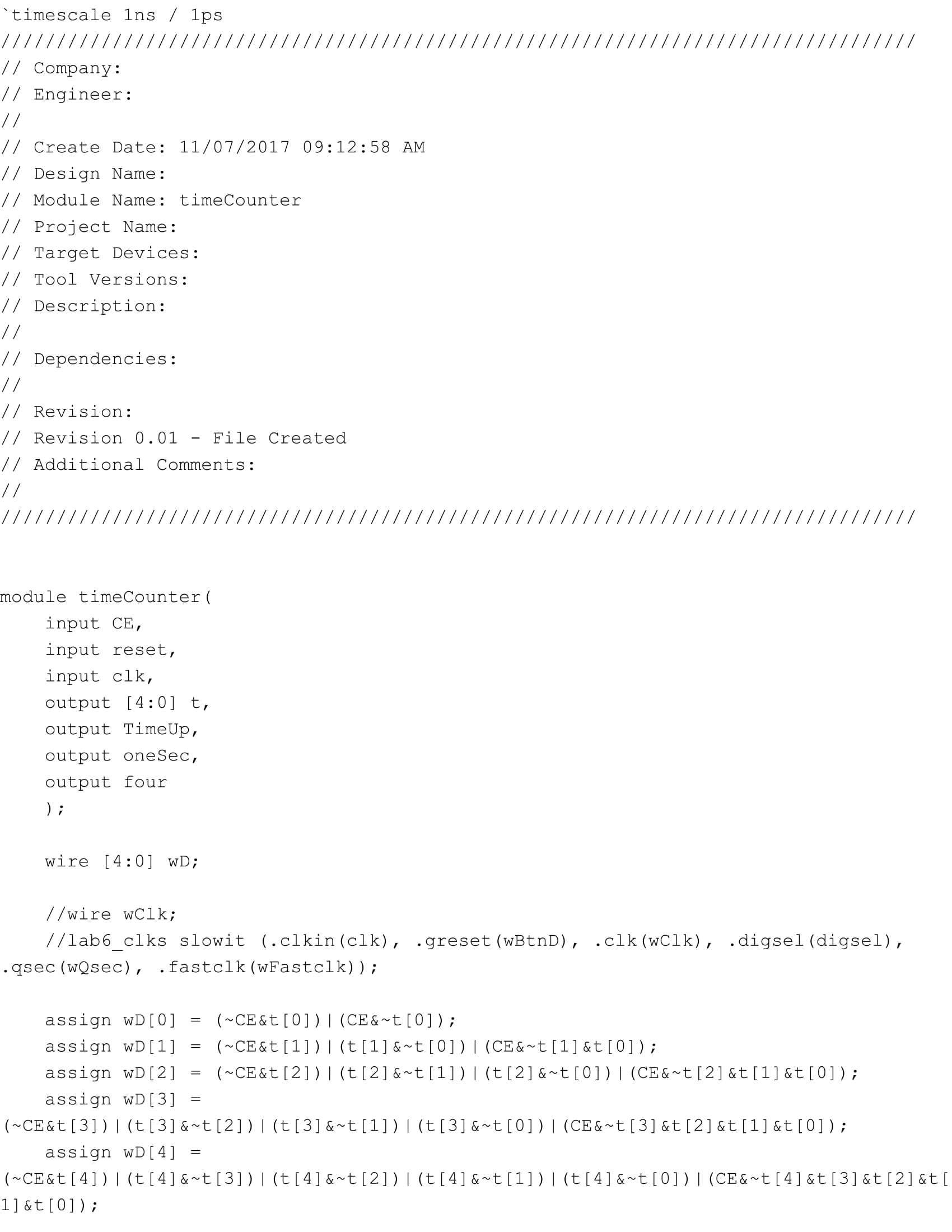
****

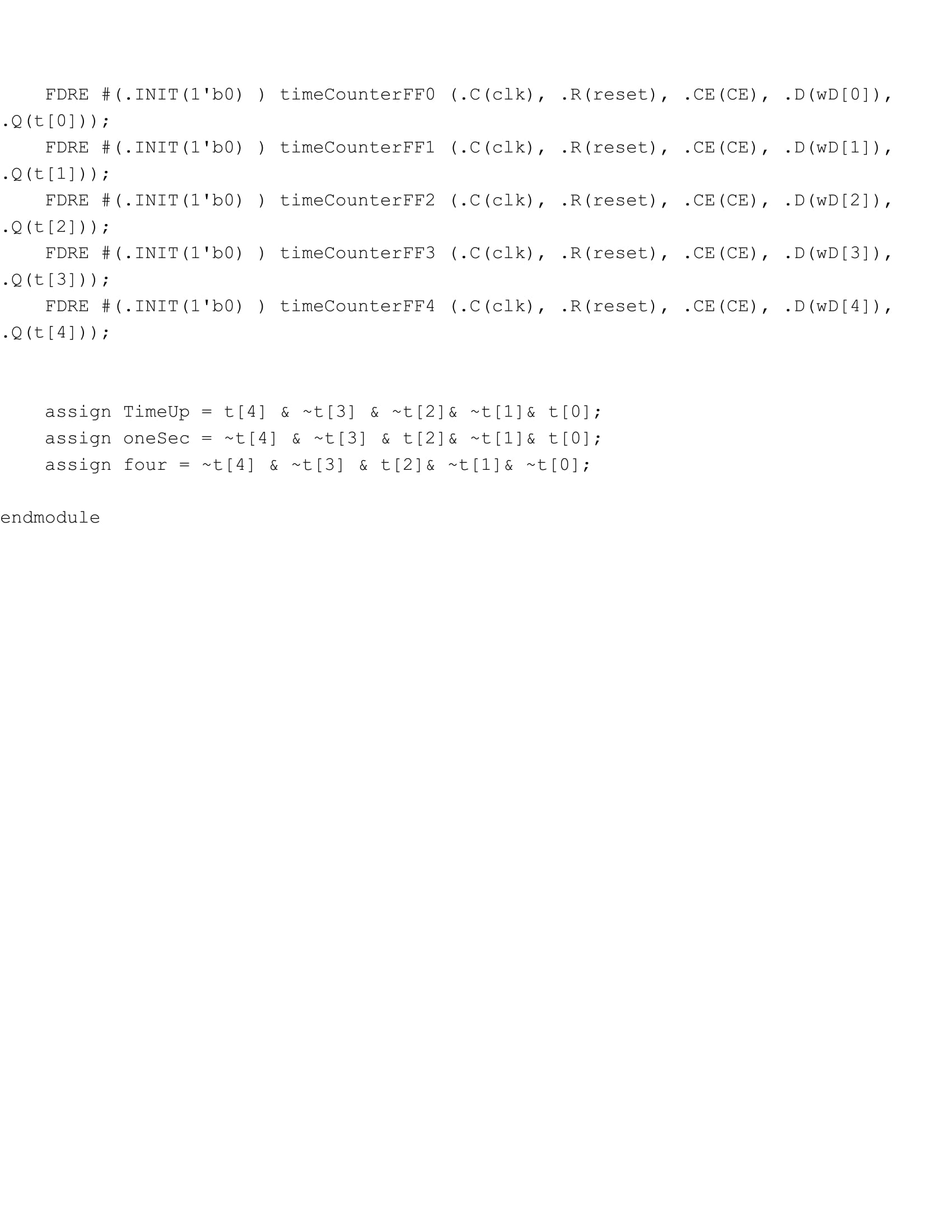
****

****

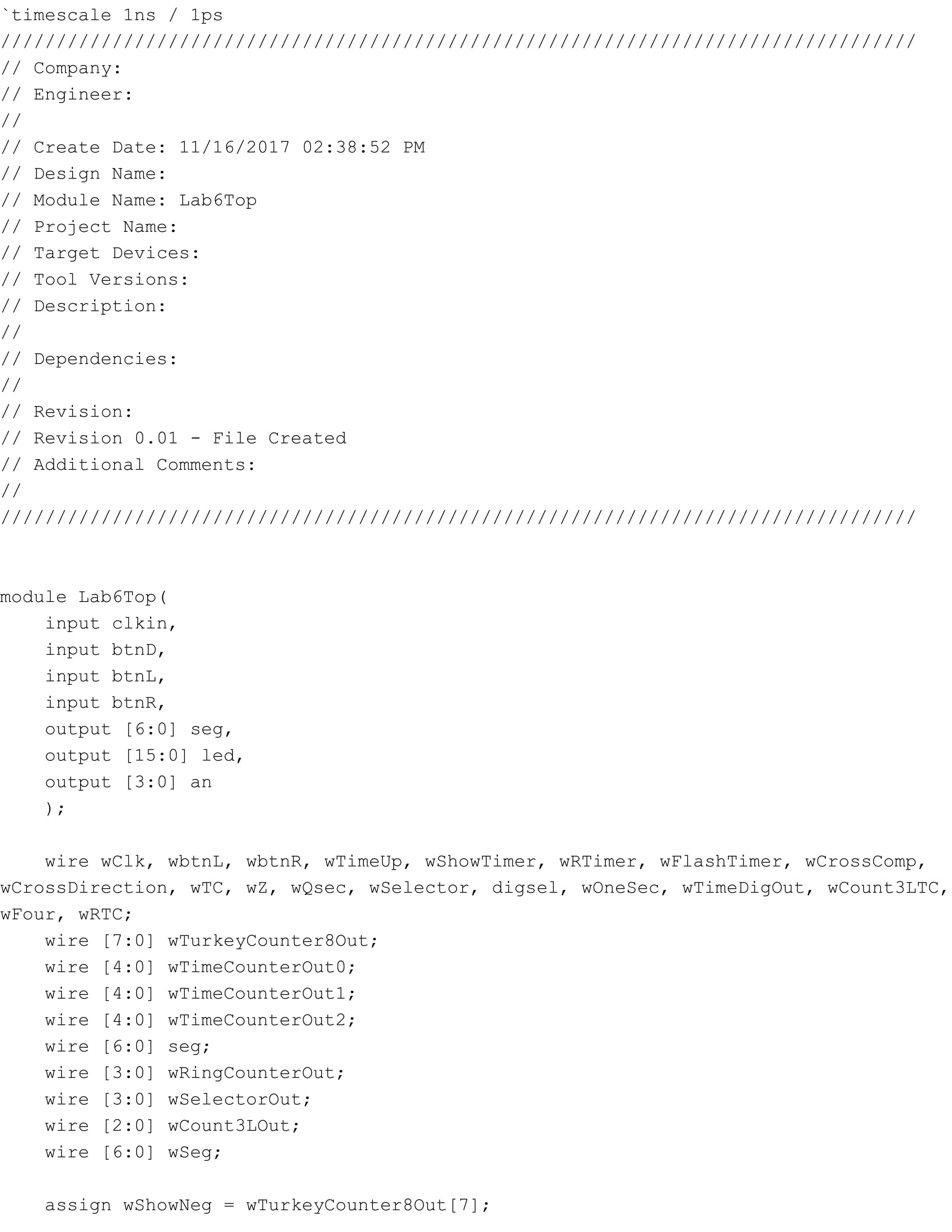
****

**Part 3 - Time Counter:**





**Part 5 - Top Level:**

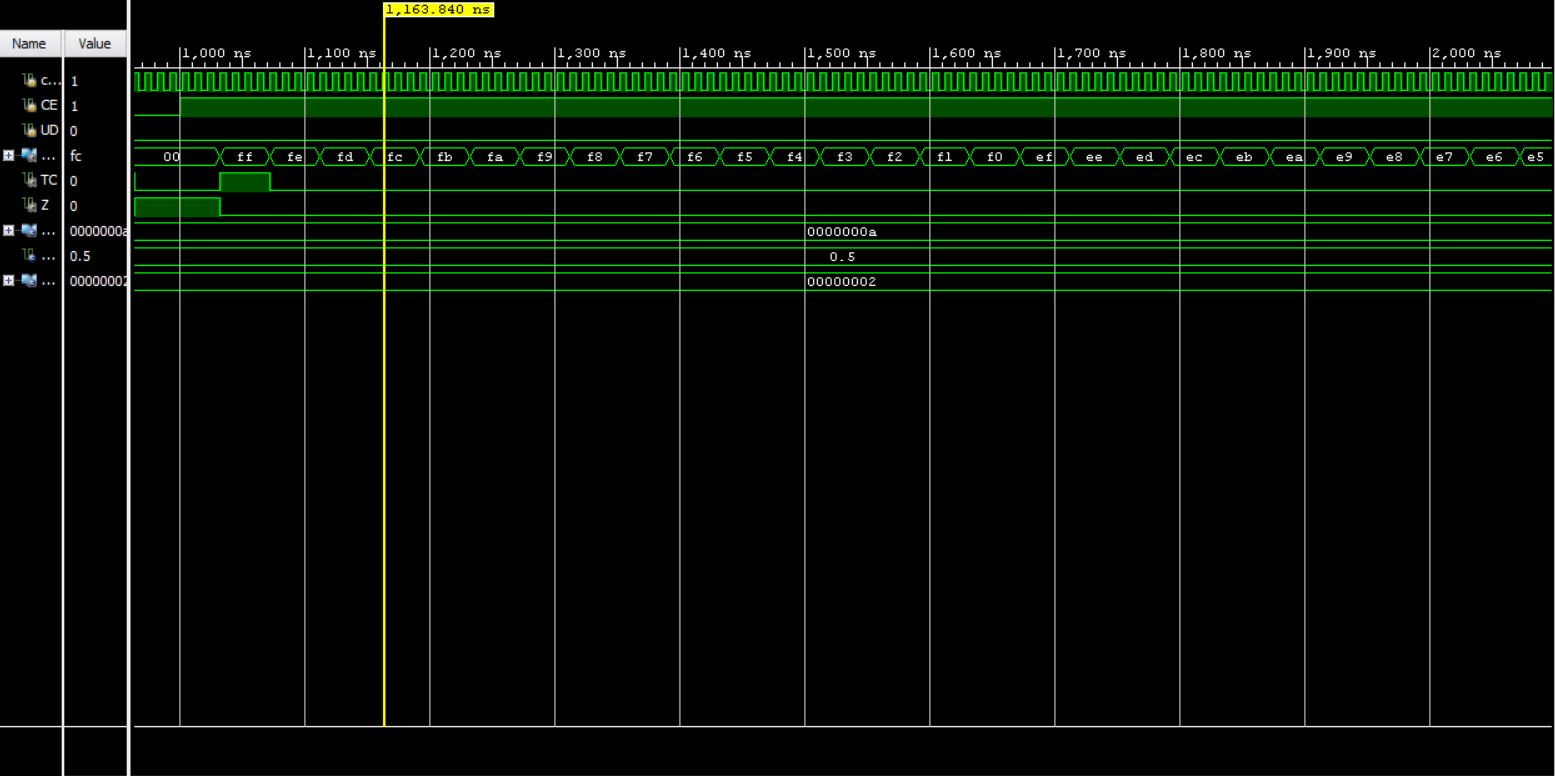
****

****

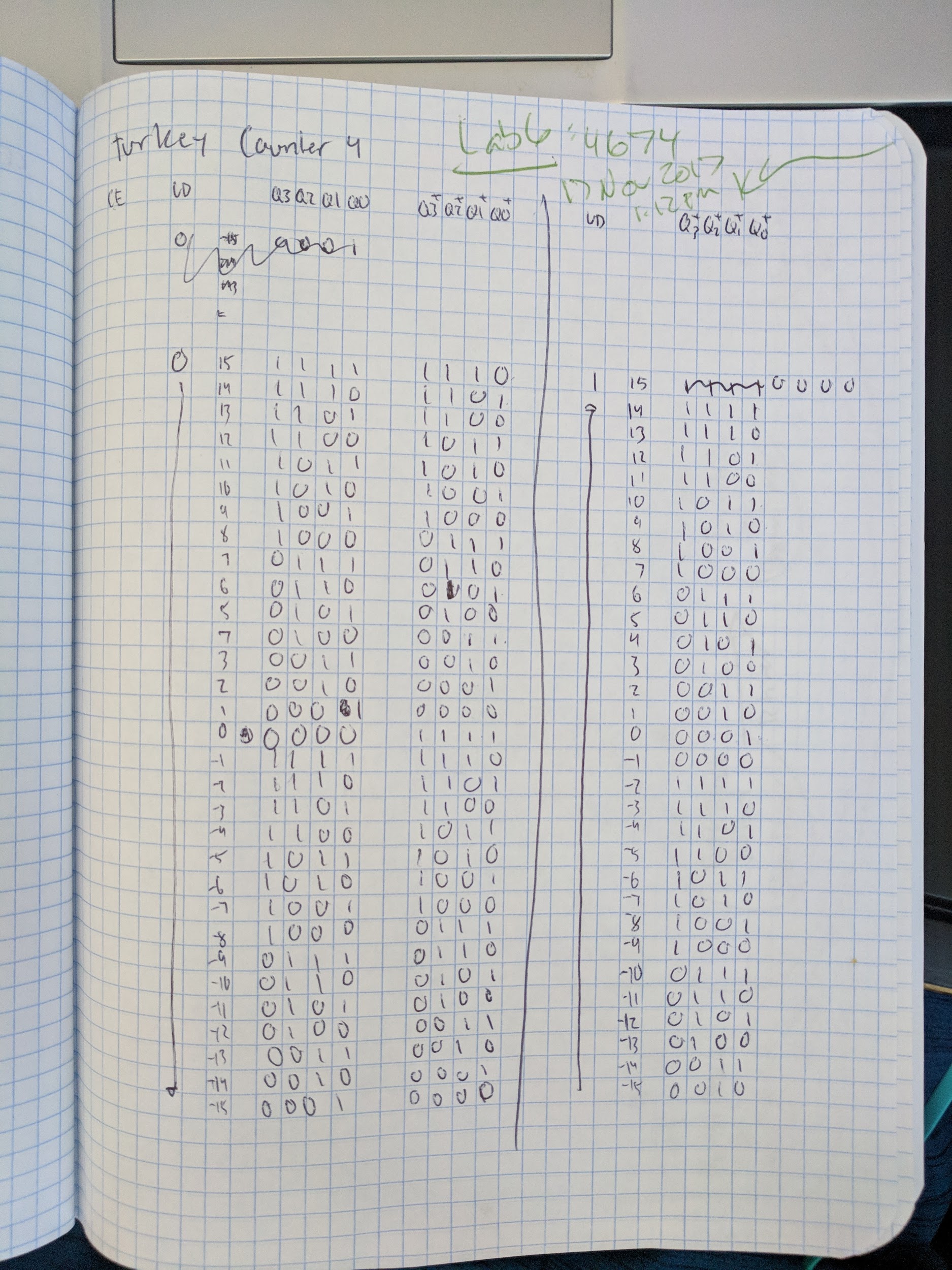
****

**Waveform:**

**Part 2 - Turkey Counter:**



**Notebook:**

****

