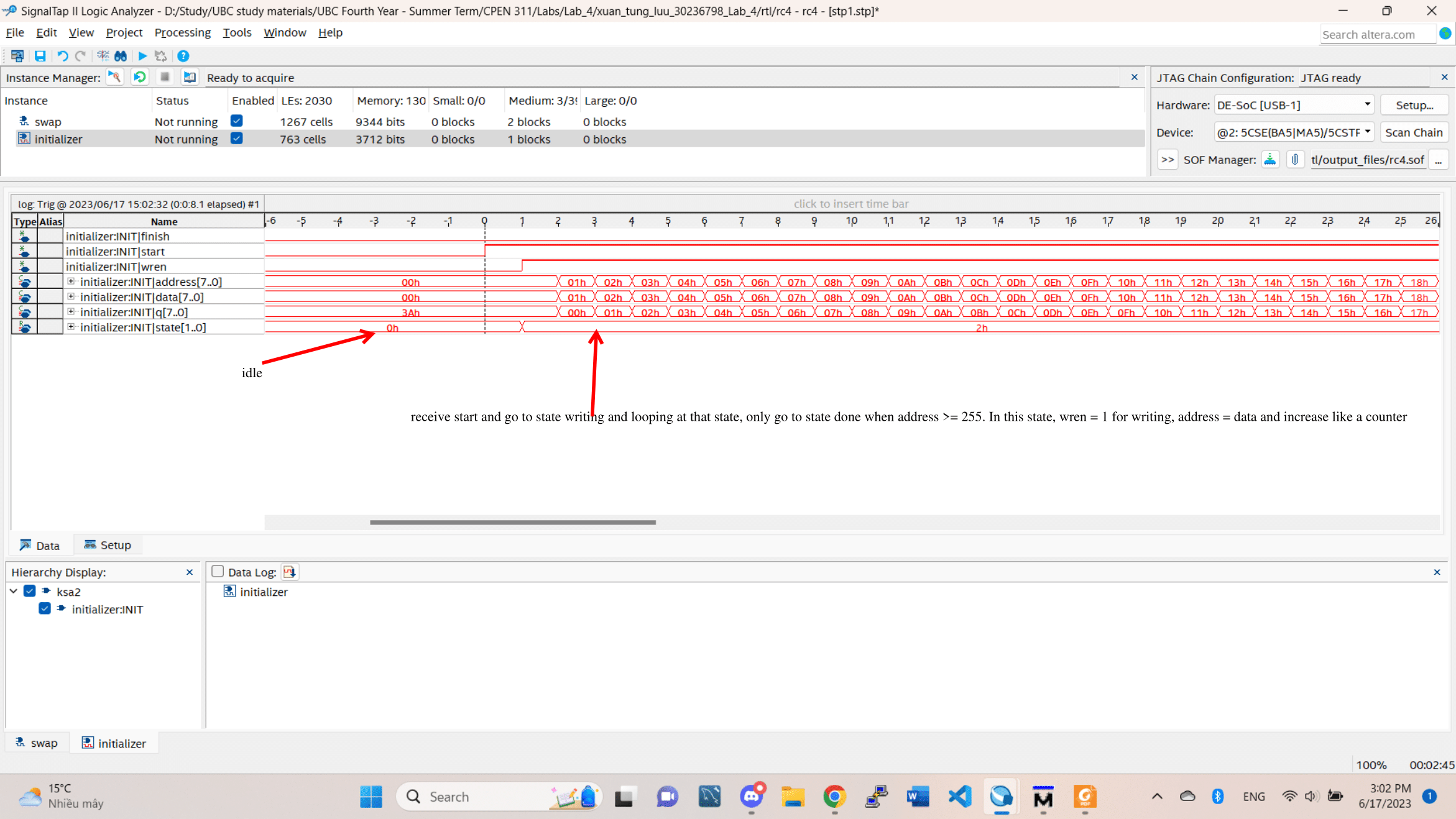
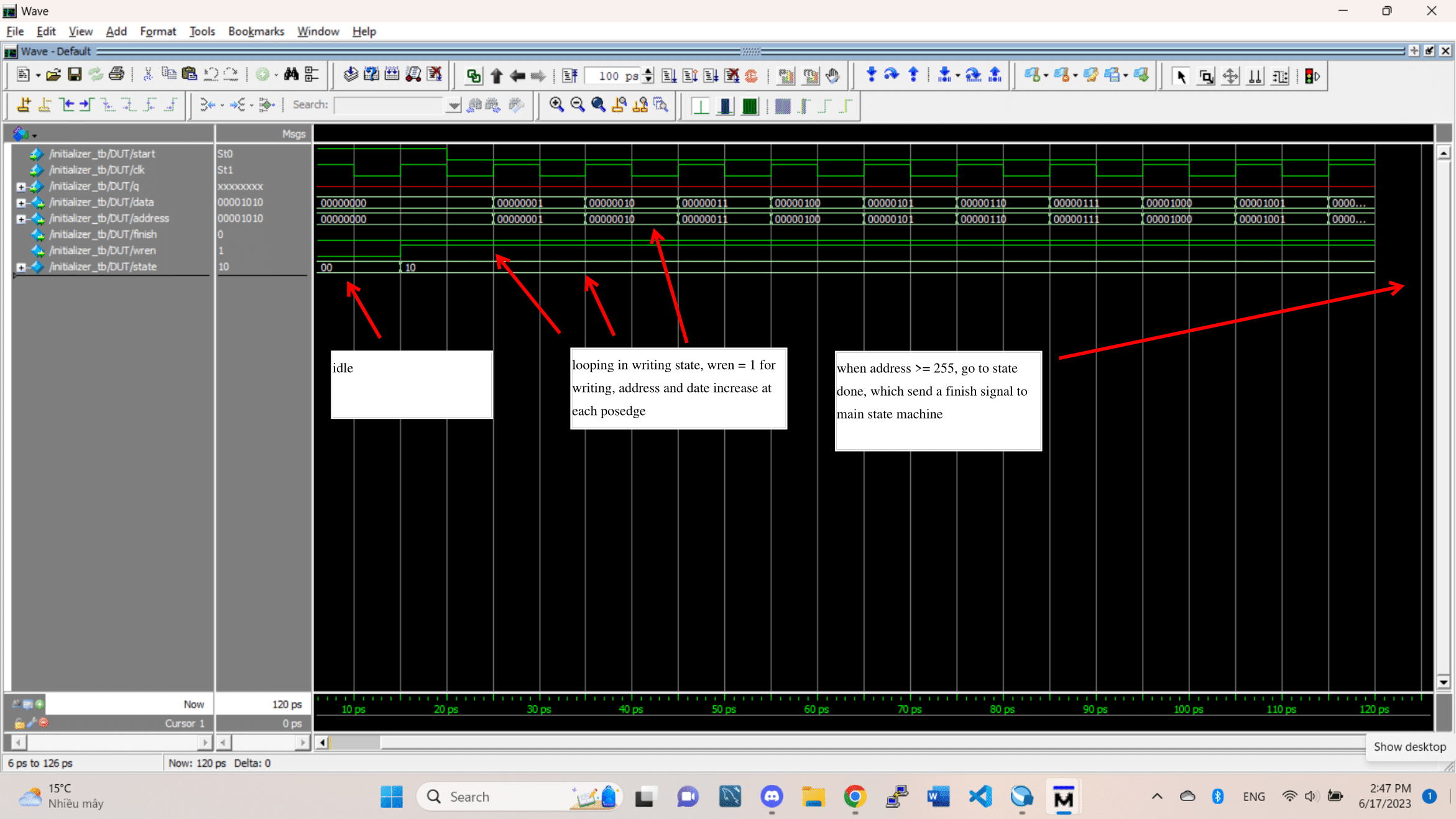
Tim Yang - 53414207 - Xuan Tung Luu - 30236798 - README

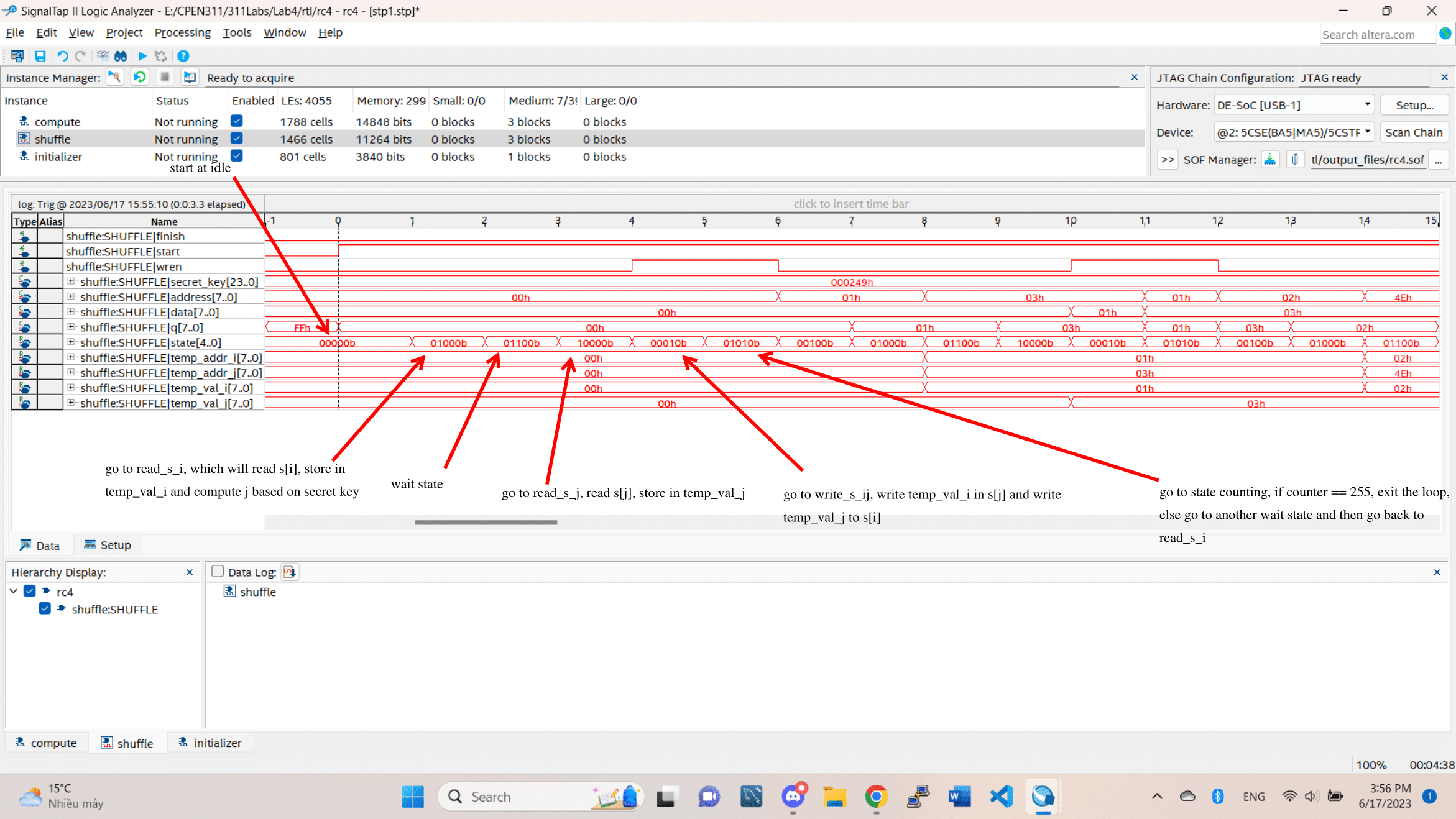
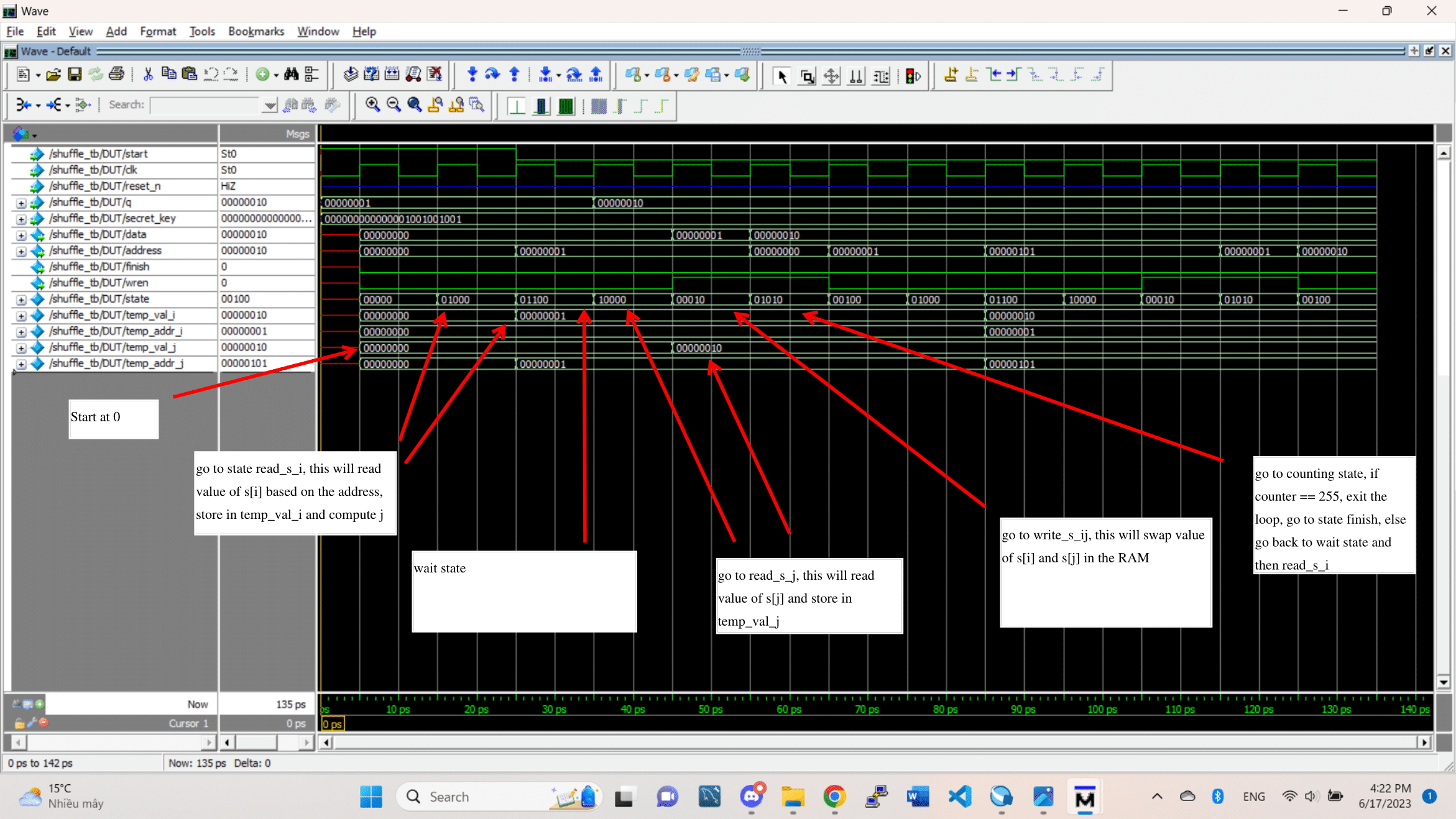
1. My SOF file is located at tim\_yang\_53414207\_xuan\_tung\_luu\_30236798\_Lab\_4/rtl/rc4.sof

2. Everything works.

3&4. We have 4 FSM. The first one is named initializer, which will initialize each address to store the same value of the address. 

Initializer

Next FSM is called shuffle, which will calculate the value j in each loop as the code in task 2a and swap the value at address i and j.



Shuffle

Third FSM is called computeEncyptedByte which will take the s array and use it to compute a decrypted output based on the encrypted rom.

A picture containing screenshot, text, diagram, line

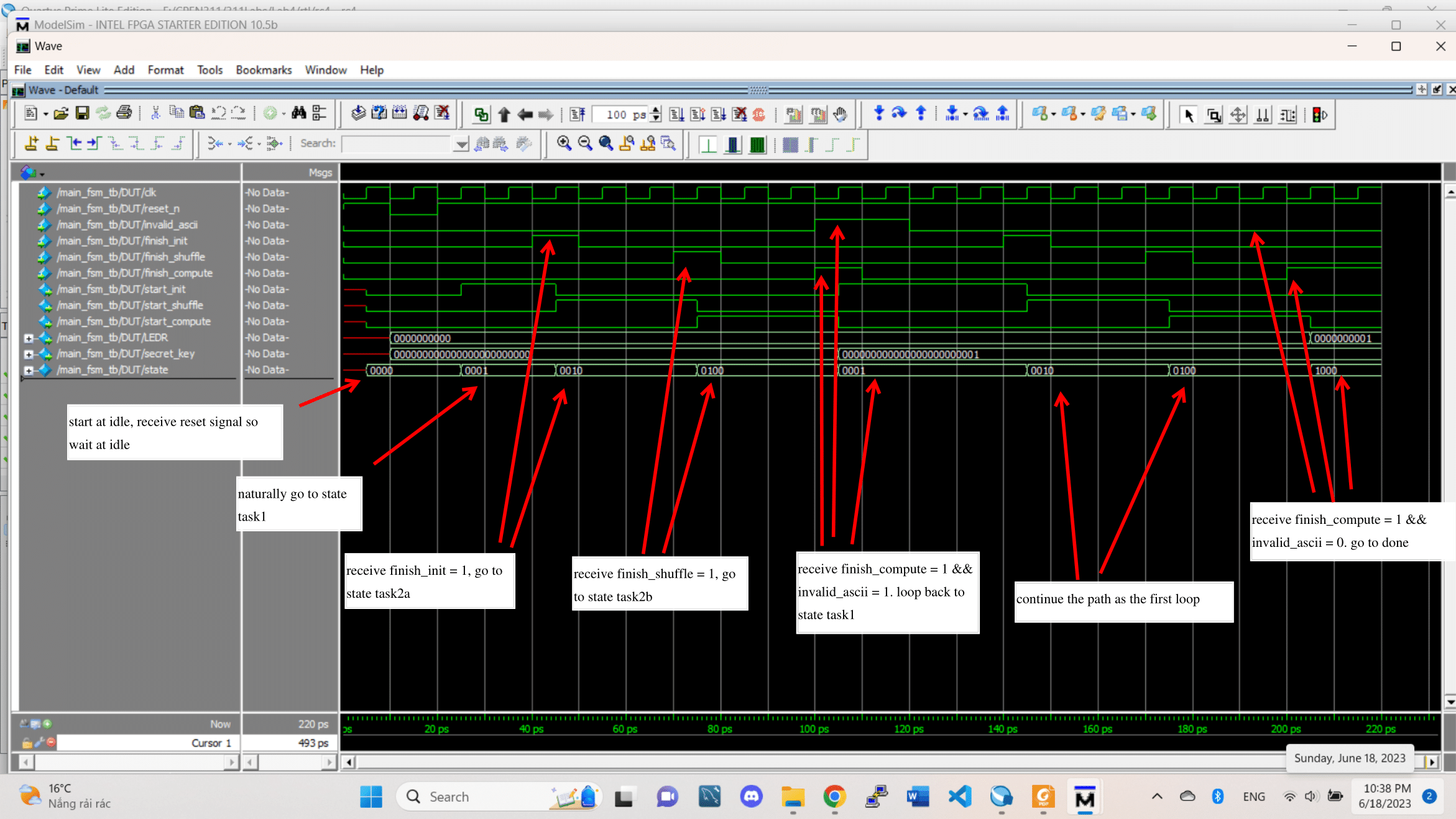
Description automatically generated

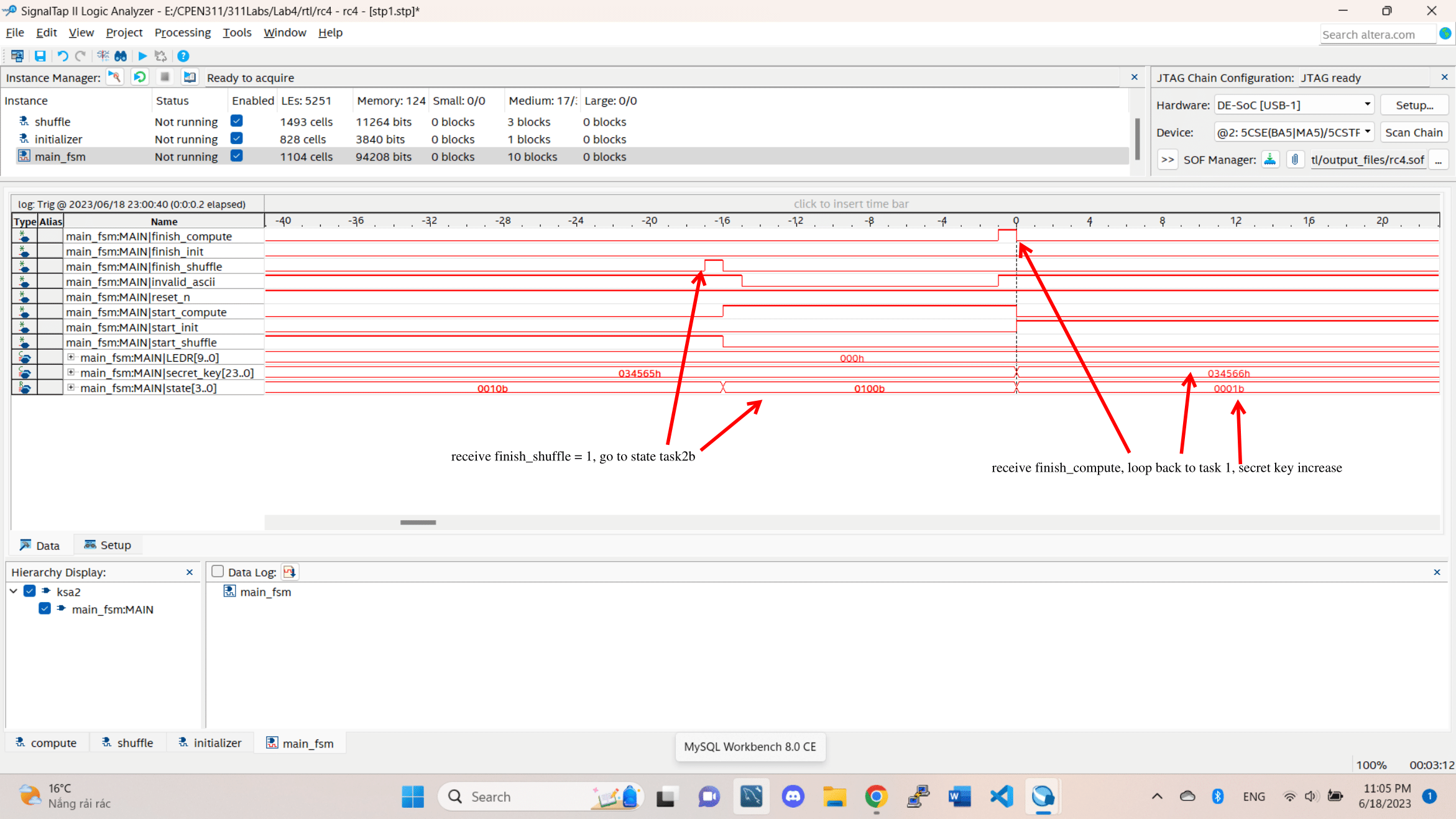
A screenshot of a computer

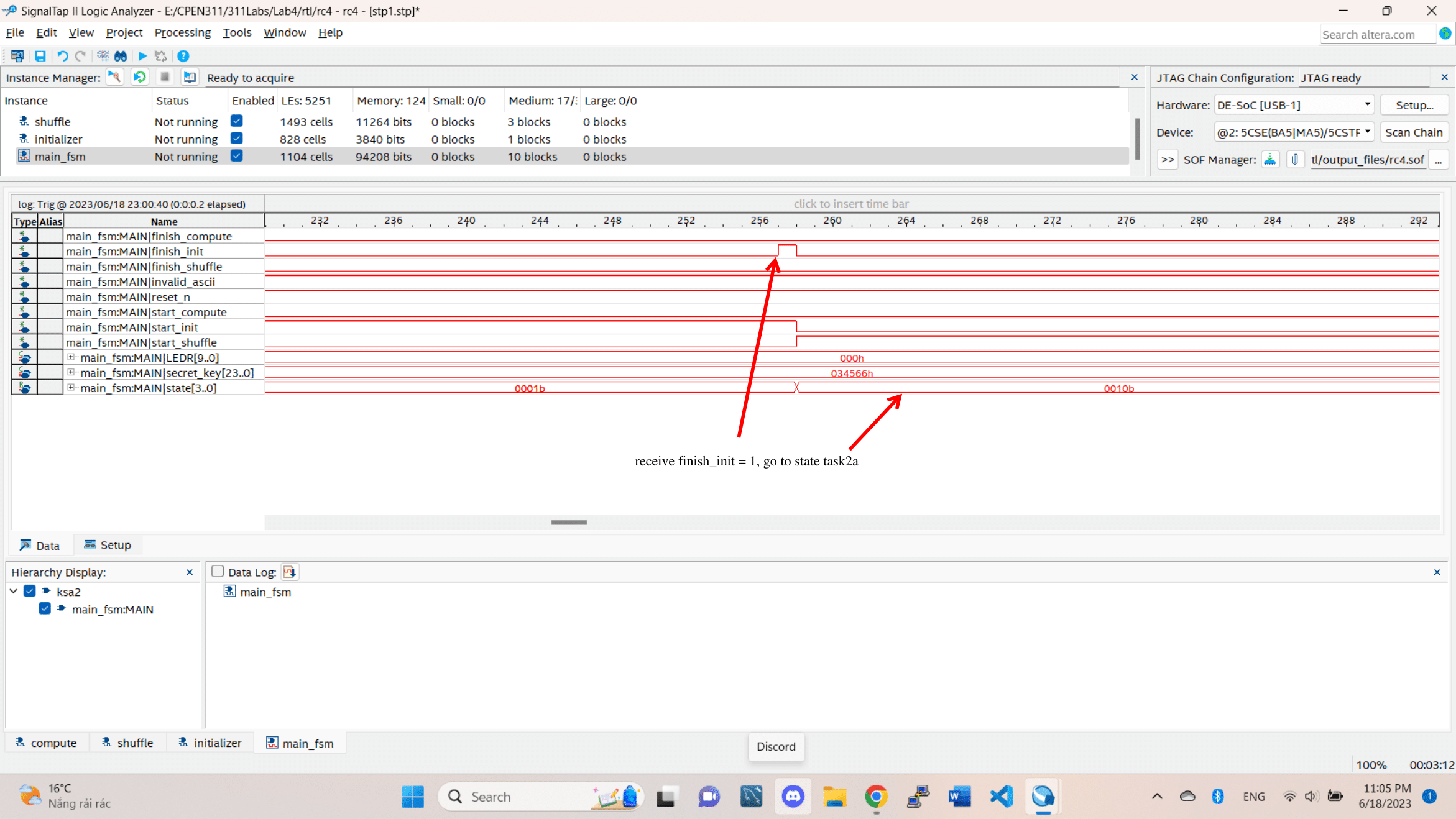
Description automatically generated with low confidence

ComputeEncryptedByte

Lastly, we have the FSM main\_fsm, which control when other state machine will start and loop around to find the secret key.







Main\_fsm

5. We use modelsim to run the simulation. The main files for each state machine are

tim\_yang\_53414207\_xuan\_tung\_luu\_30236798\_Lab\_4/rtl/initializer.sv

tim\_yang\_53414207\_xuan\_tung\_luu\_30236798\_Lab\_4/rtl/shuffle.sv

tim\_yang\_53414207\_xuan\_tung\_luu\_30236798\_Lab\_4/rtl/computeEncryptedByte.sv

tim\_yang\_53414207\_xuan\_tung\_luu\_30236798\_Lab\_4/rtl/main\_fsm.sv

The testbench files are

tim\_yang\_53414207\_xuan\_tung\_luu\_30236798\_Lab\_4/rtl/initializer\_tb.sv

tim\_yang\_53414207\_xuan\_tung\_luu\_30236798\_Lab\_4/rtl/shuffle\_tb.sv

tim\_yang\_53414207\_xuan\_tung\_luu\_30236798\_Lab\_4/rtl/computeEncryptedByte\_tb.sv

tim\_yang\_53414207\_xuan\_tung\_luu\_30236798\_Lab\_4/rtl/main\_fsm\_tb.sv

To simulate, just create the project with each couple of files (e.g initializer.sv with initializer\_tb.sv) and add the waves similar to the pictures (or any wave that you want).