EE 371 Lab 4 Report

Jiarong Qian

I. Introduction

In this lab, I implemented a bit counter that counts the number of “1” from a input binary value and a binary search system that searches for an input value from a 32x8 RAM.

II. Lab Procedures

For the bit counter, I followed the ASMD diagram provided in the lab document. There were 3 states: initial state (S1), count state (S2), and finish state (S3). At the initial state, the counter value is set to 0 and the system loads the input value. Once the start signal is on, the system goes to count state and checks the last digit of the data. If the last digit is 1, counter increases by 1, otherwise it would not change. Then it shifts the data to right by 1. Once the data becomes 0, count finishes and the system stays at finish state until start signal is off.

Then I started drawing the ASMD diagram for the binary search system. I first started with 4 states: initial state (S1), search state (S2), found state (S3), and not found state (S4). In search state, the system starts with a search boundary from 0 to 31 and compares the current value from the RAM to the value to find. If they are equal, the search finishes. If the current value is larger, the upper bound for the search becomes 1 less than the current address, else the lower bound becomes 1 greater than the current address. Then the system repeats the cycle until it finds the value or the lower bound is larger than or equal to the upper bound, at which the search finishes.

I implemented the RAM with the built-in RAM from the IP catalog. After I finished coding and simulated the system, I found that the results were incorrect. I realized that there was delay in updating the address and reading the RAM so the system was always updating the address based on the comparison of the target value with the value read last clock cycle. I also realized that found state and not found state were redundant since the only difference is on the found signal. So I changed my ASMD diagram and the new states are: initial state (S1), read state (S2), address update state (S2_5), finish state (S3). The ASMD diagram is given below:

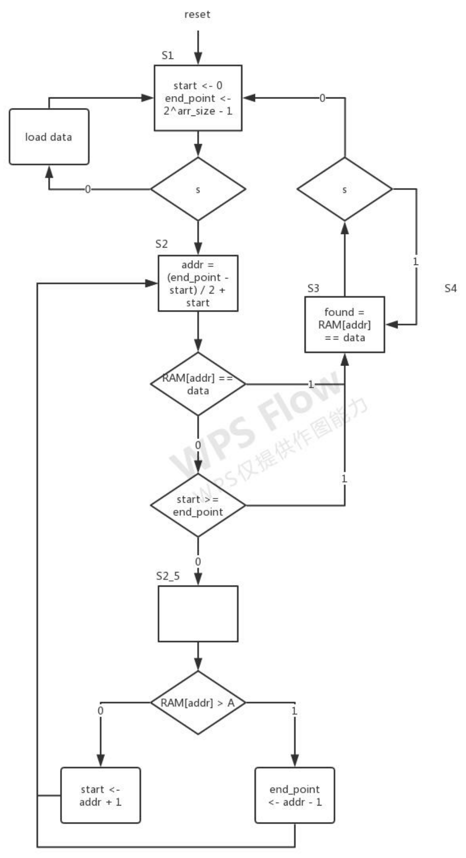


Fig 1. binary_search ASMD

III. Results

I tested the systems with several different inputs and the simulation results suggests that the systems were working correctly. After downloading the systems to the FPGA board, both systems worked correctly, though the binary search system sometimes does not stay at the the finished state but goes back directly to the initial state. I suspect this happens because the switch sometimes does not have a good contact to the internal circuit, since this happens more easily when I vibrate my finger on the switch slightly before turning it on.

IV. Feedback

When I first downloaded the binary search system onto the FBGA board, it did not work though the simulation was correct. The address was not updating in the way as it should be. I overcame this issue by introducing a new set of logic to store new address and new search boundaries. These values are assigned blockingly based on the current address and boundaries and then I assign address and boundaries to new values non-blockingly. Though the logic was the same, it solved the problem.

Because of this problem I encountered, I spent very long time on this lab. I think that I probably spent 40 hours on it.

Appendix

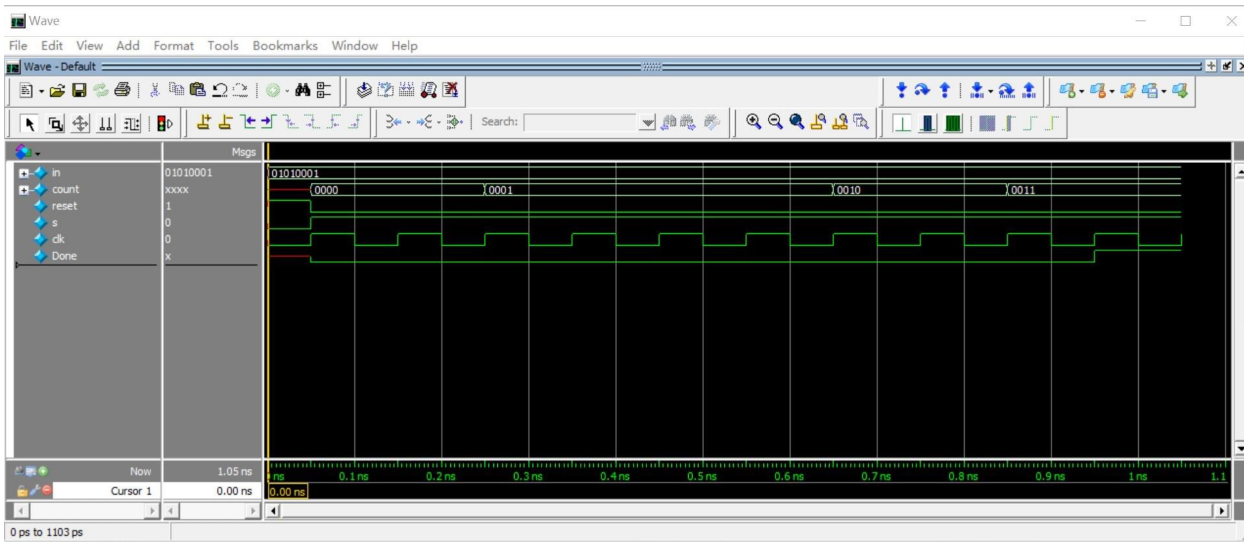


Fig 2. bitCounter Simulation

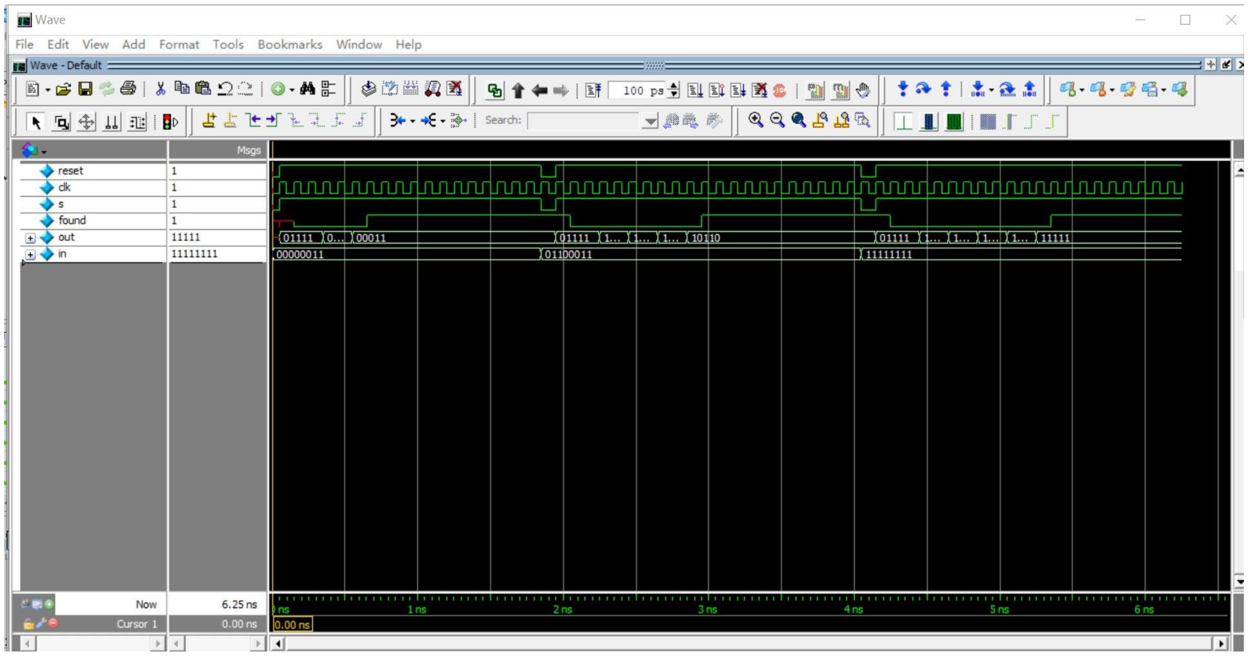
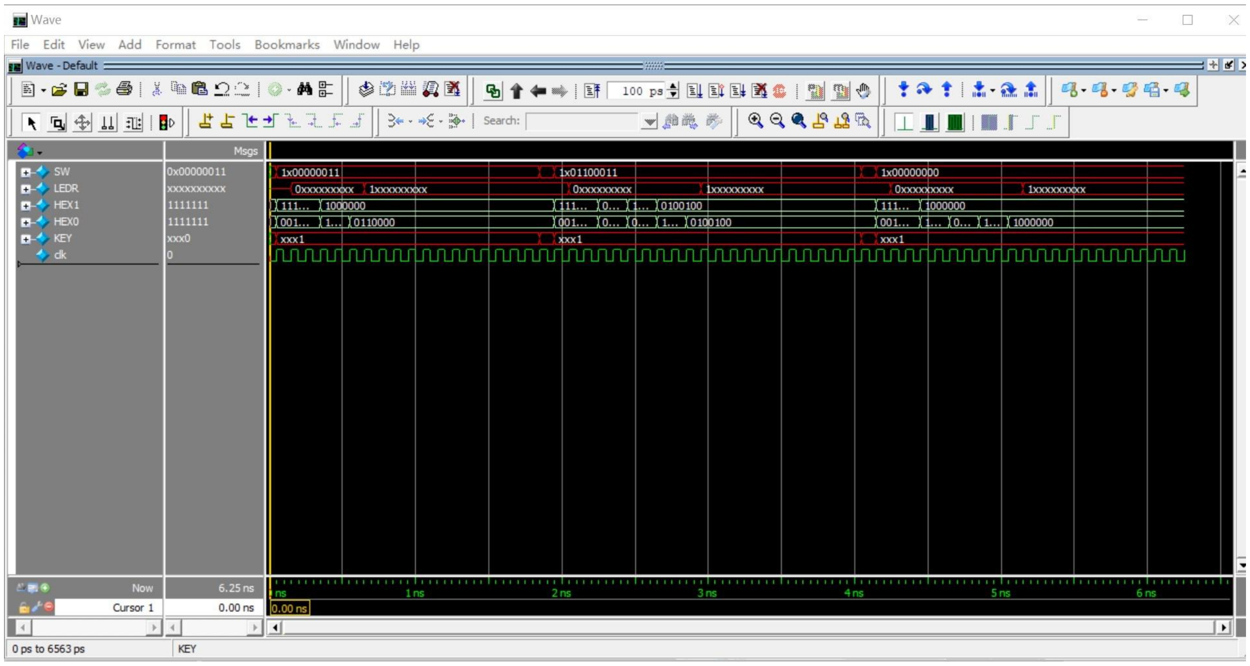
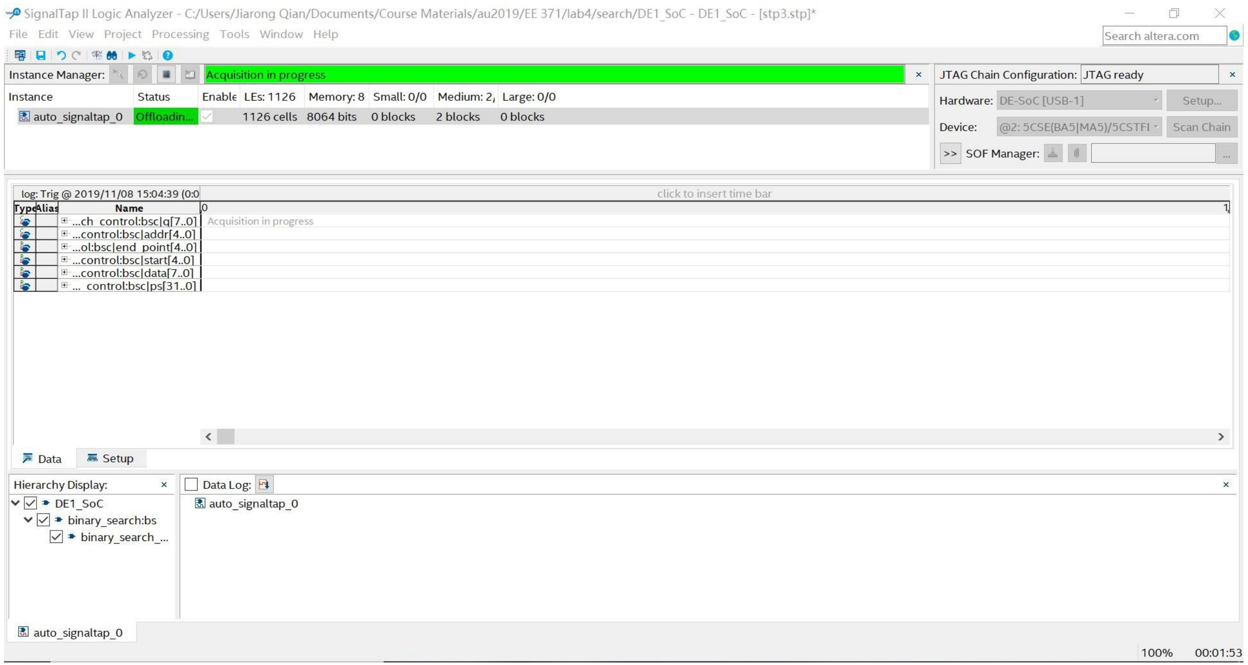


Fig 3. binary_search Simulation



```
SW     0x00000011   1x00000011 | 1x01100011 | 1x00000000
LEDR   xxxxxxxxxx   0xxxxxxxxx 1xxxxxxxxx | 0xxxxxxxxx 1xxxxxxxxx | 0xxxxxxxxx 1xxxxxxxxx
HEX1   1111111      111... 1000000 | 111... 0... 1... 0100100 | 111... 1000000
HEX0   1111111      001... 1... 0110000 | 001... 0... 0... 1... 0100100 | 001... 1... 0... 1... 1000000
KEY    xxx0         xxx1 | xxx1 | xxx1
clk    0
Now      6.25 ns
Cursor 1 0.00 ns
0 ps to 6563 ps    KEY
```

Fig 4. DE1_SoC (for binary_search) Simulation



```
SignalTap II Logic Analyzer - C:/Users/Jiarong Qian/Documents/Course Materials/au2019/EE 371/lab4/search/DE1_SoC - DE1_SoC - [stp3.stp]*
Instance Manager:   Acquisition in progress
Instance           Status      Enable  LEs: 1126    Memory: 8   Small: 0/0  Medium: 2,  Large: 0/0
auto_signaltap_0   Offloadin...        1126 cells   8064 bits   0 blocks    2 blocks    0 blocks

JTAG Chain Configuration: JTAG ready
Hardware: DE-SoC [USB-1]                 Setup...
Device: @2: 5CSE(BA5|MA5)/5CSTFI         Scan Chain
>> SOF Manager:

log: Trig @ 2019/11/08 15:04:39 (0:0
Type Alias Name                              0
     ...ch_control:bsc|q[7..0]               Acquisition in progress
     ...control:bsc|addr[4..0]
     ...ol:bsc|end_point[4..0]
     ...control:bsc|start[4..0]
     ...control:bsc|data[7..0]
     ... control:bsc|ps[31..0]

Hierarchy Display:              Data Log:
DE1_SoC                         auto_signaltap_0
  binary_search:bs
    binary_search_...
auto_signaltap_0
100%   00:01:53
```

Fig 5. Signal Tap Result for Binary Search