Lab 4: Basic Computer Organization

**CEG 2136 B - Computer Architecture**

**Fall 2017**

**School of Electrical Engineering and Computer Science University of Ottawa**

Course Coordinator: Dr. Sawsan Abdul-majid

Group 18

Felix Singerman 7970742

Nasser Haidar  8543506

Submission Date: December 4, 2017

# Theoretical Part

**1. Introduction of problem or to lab**

In this lab, we are creating and analyzing a basic computer control unit. We will also use opcodes to write simple programs in machine code. This is important because it will help us understand how instructions are executed and how the memory addresses are retrieved.

In the prelab, we derived the equations of all the control signals which have to be generated by the Control Unit to control the registers and ALU for the CPU data path, the bus and the memory. For the hardware aspect of the lab, we analyzed the RTL expressions of Table 2, Table 3, Table 4 in the lab manual and wrote the logic expression for each of the control signals to draw the controller. For the programming component we designed a program to add the sequence given in the lab manual.

**2. Discussion of problem**

The problem for this lab consisted of designing, simulating, building and testing a control unit.

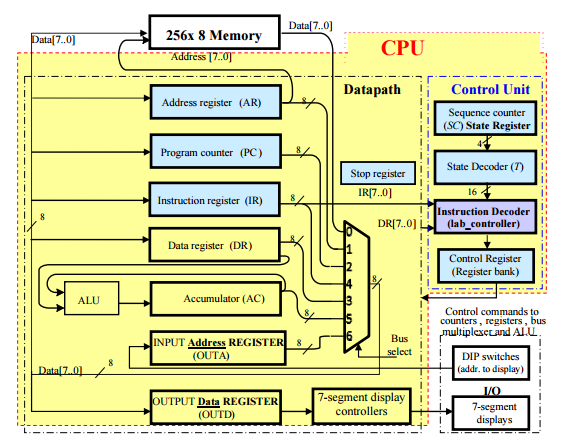


Figure 1: Computer block diagram

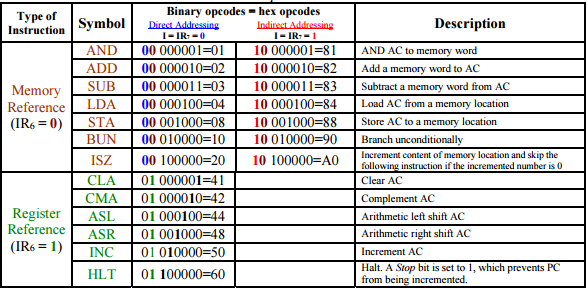


Figure 2: Computer Instructions List

3. Discussion of algorithmic solution Explain the used algorithm to solve it and the block components you are going to use.

To start we needed to find the hardware equations for part 1 using the provided tables from the lab4 document. We then added the logic diagrams for these equations into lab4top file to complete the circuit diagram.

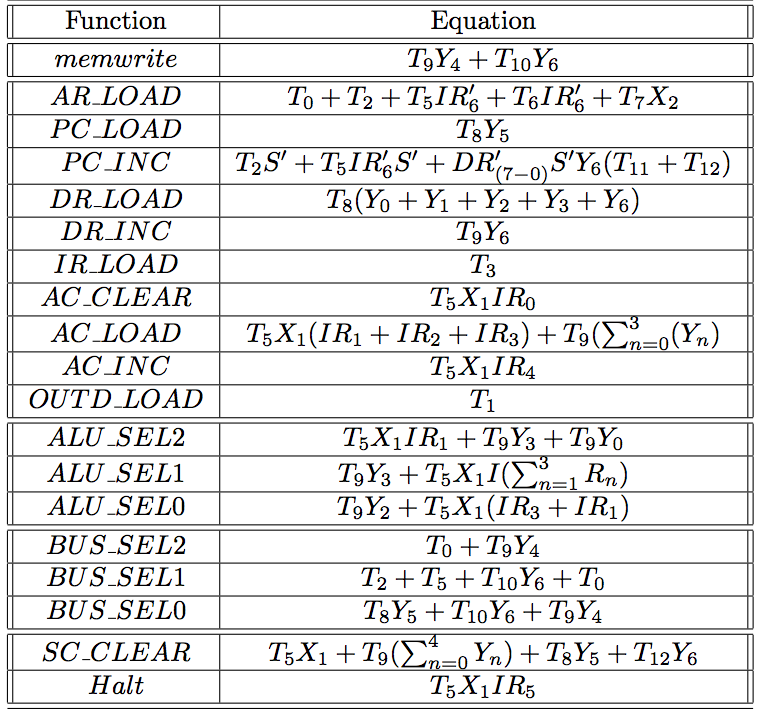
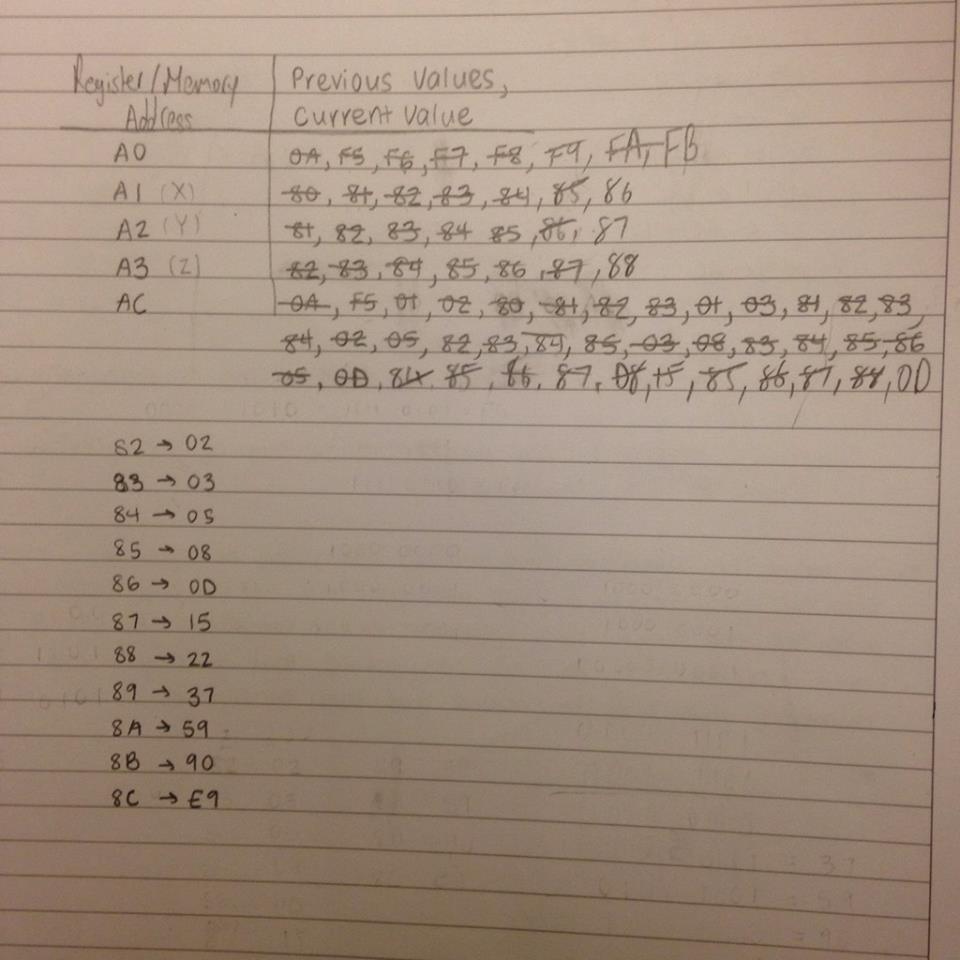


Figure 3: Hardware Equations

For the second part of the lab we needed to write a program that consecutively added each number of a given sequence of hexadecimal numbers and displays the hexadecimal number that causes the sum to be equal to zero.



Design Part

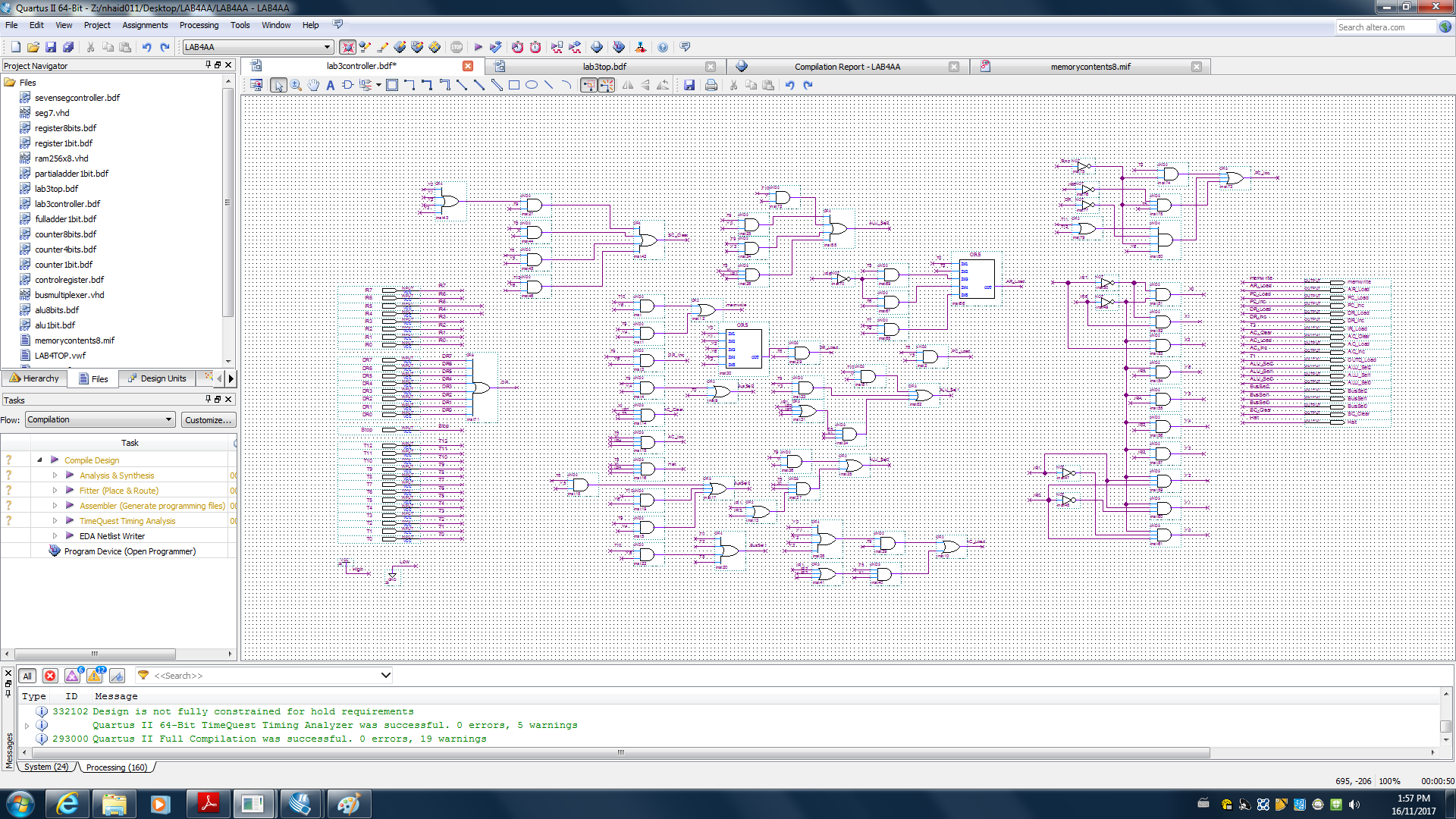


Figure 6: Completed Circuit Diagram lab4top

2. **Discussion of used components**

**AND Gate**

A gate where the output is 1 if and only if both inputs have a value of 1, otherwise, its output is 0.

**NOT Gate**

A gate where the output is 1 if the input is 0, and vice versa.

**XOR Gate**

A gate where the output is 1 if only one of the inputs is 1.

**NOR Gate**

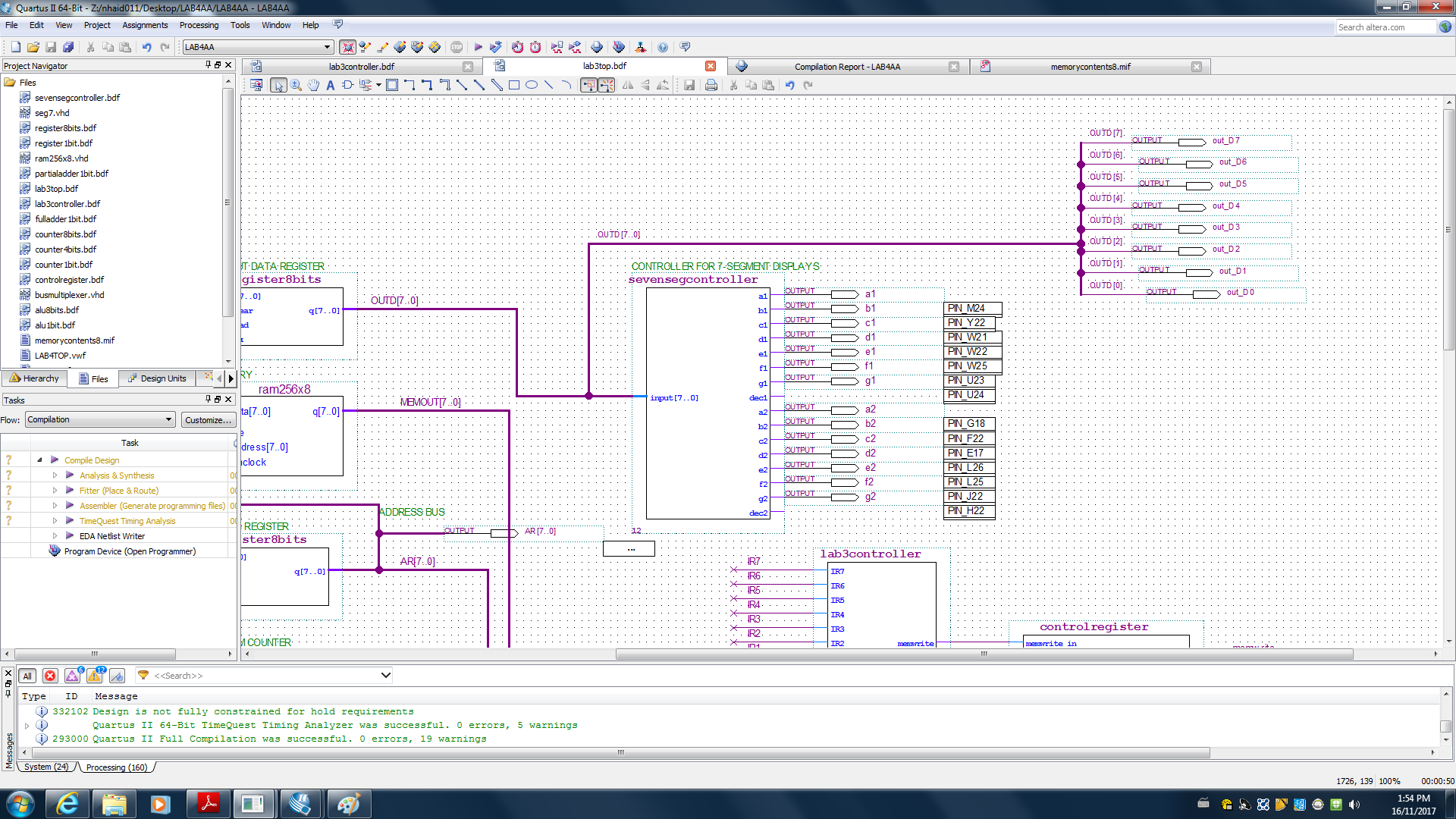
A gate where the output is 1 if and only if both inputs are 0, otherwise, its output is 0.

**Circuit Diagrams**

*Refer to “Design Part”*

3. Discussion of actual solution

After deriving the equations for the control signals we implemented them in Quartus.

****

We then created a mif file for the second part of the lab.

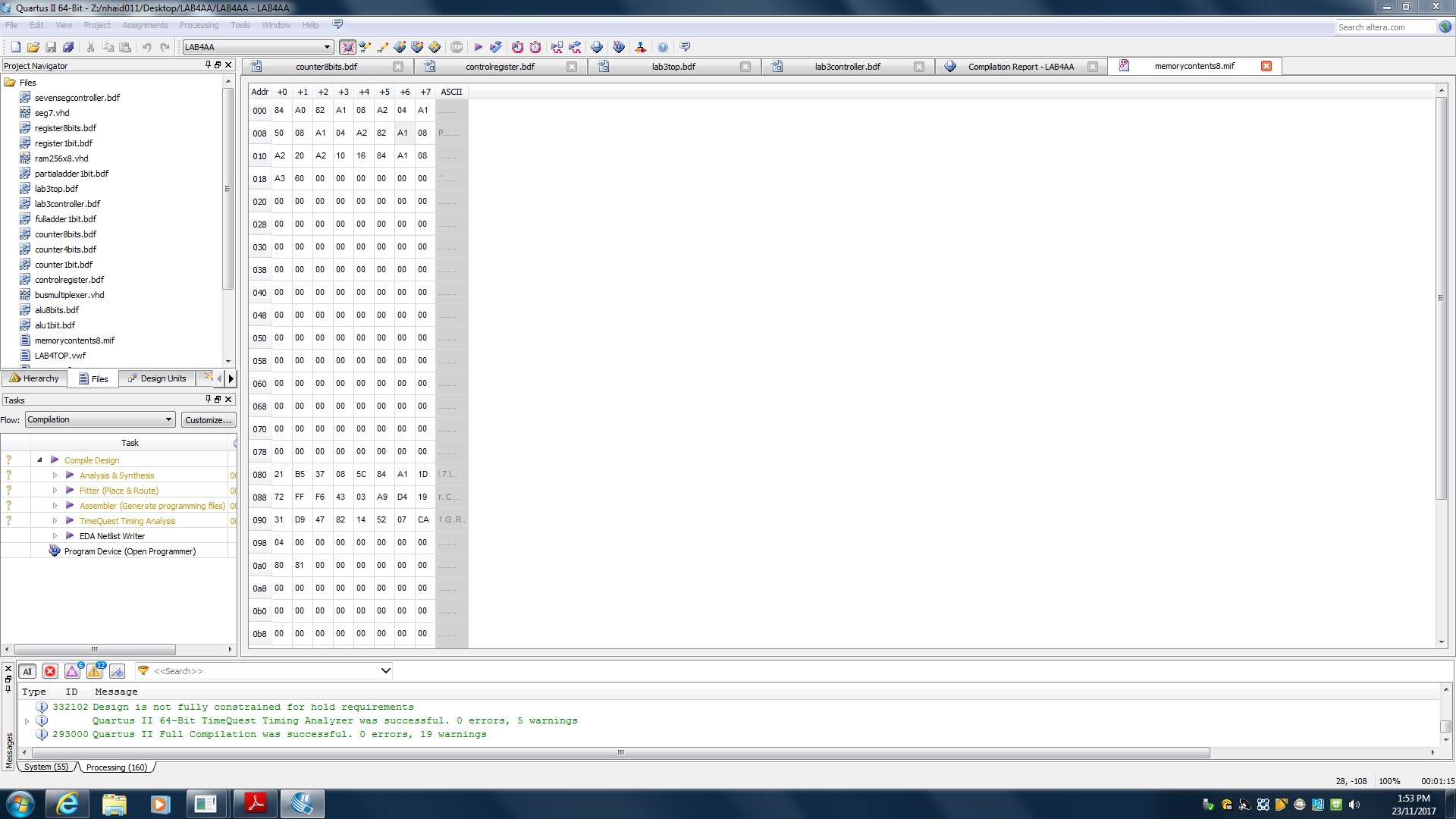
****

Figure 4: .mif file for addition program

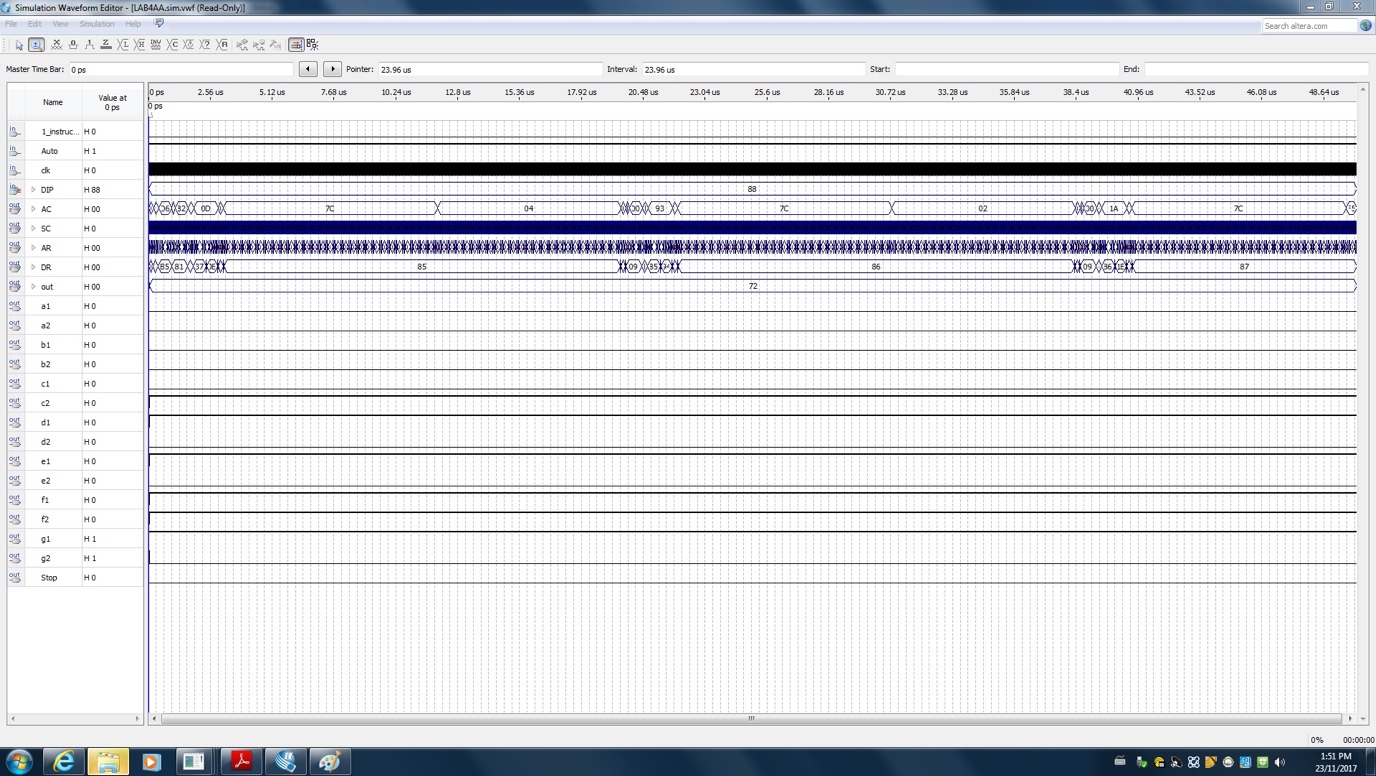


Figure 5: Simulation for the addition program

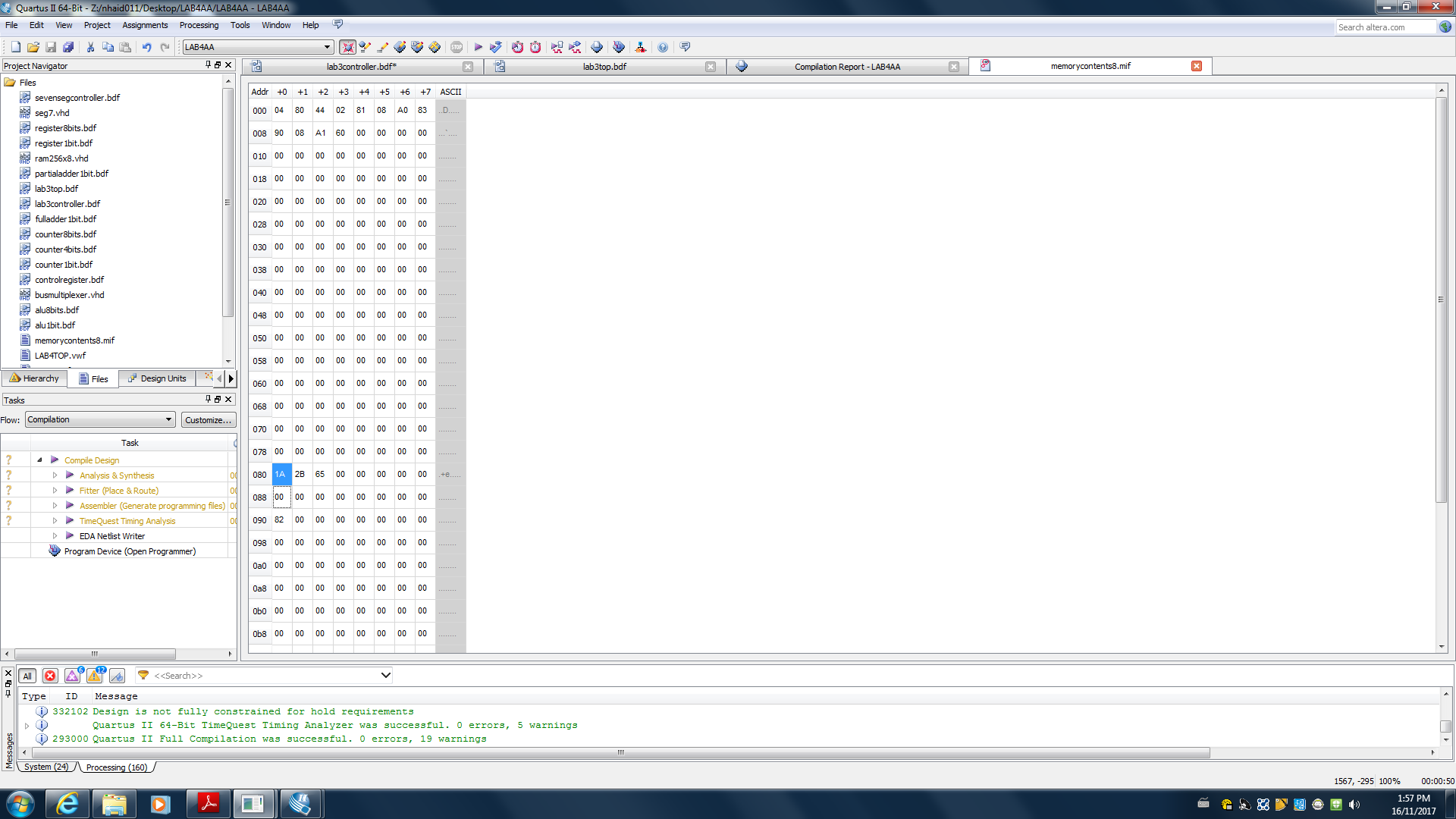


Figure 6: .mif file for consecutive numbers added

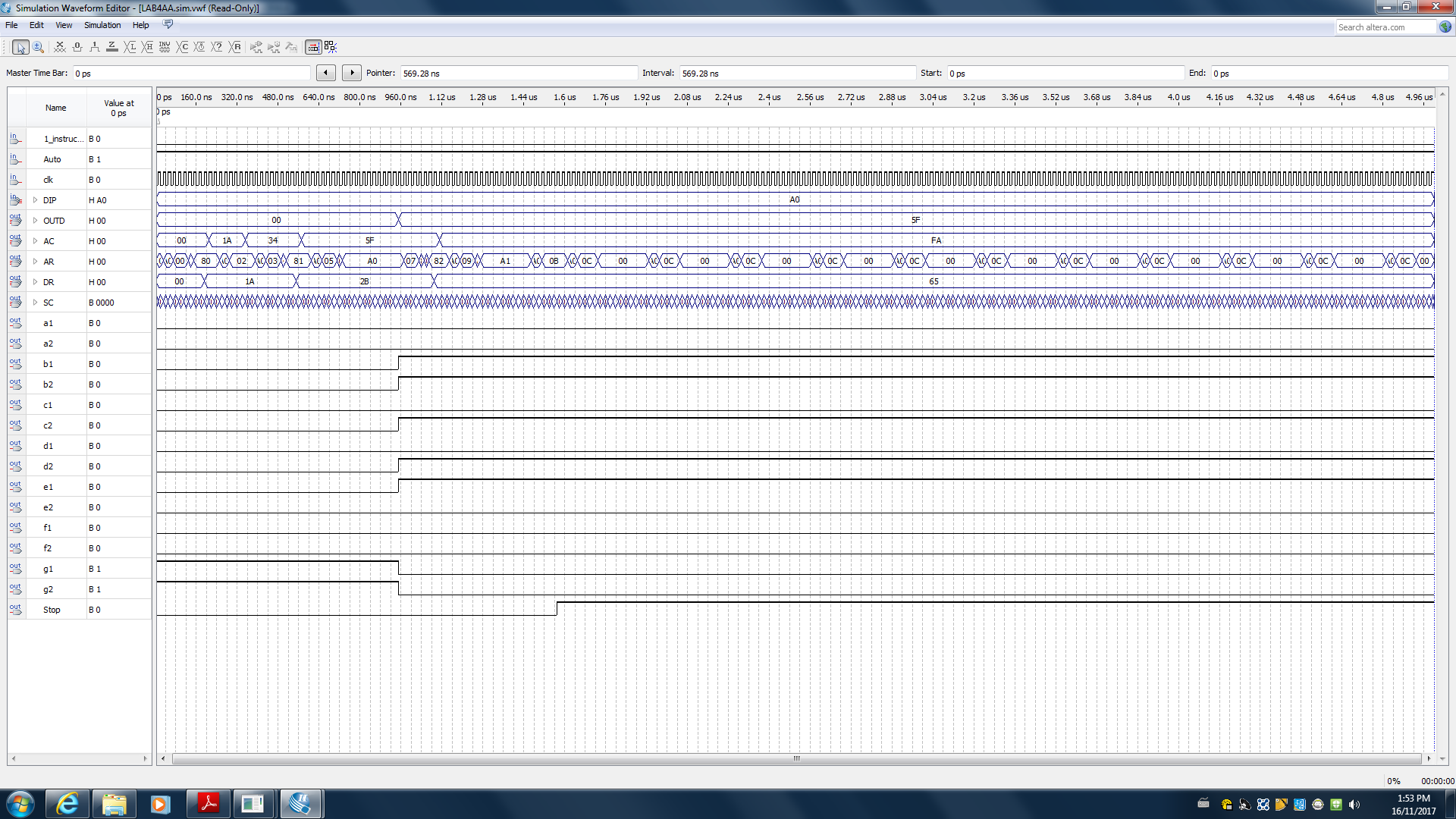


Figure 7: Simulation for the consecutive numbers added program

4. Discussion of tool

**Altera DE2-115 Board**

This is a circuit board which consists of multiple pins, buttons, and LEDs. It allows us to visualize and test our designed circuits from the Quartus software.

5. Discussion of challenging problems

We did not come across many issues during this lab. Apart from a few issues in equation calculation, the lab went off without a hitch. We just had to go over our equations with the TA to see where we went wrong.

By conducting this lab, we learned the design of controllers for a basic computer and how machine code is used within the basic computer to get desired results. Although we encountered some issues, overall we were successful with our implementation of this lab