Electronics Proposal

Aim : Design and Implementation of an 8-bit Microprocessor with Pipelining in Verilog HDL.

1 Team Details :

1.1 Team Leader :

- Name : Saatvik Managari
- Roll Number : 240002035
- Github : https://github.com/SaatvikManagari
- LinkedIn : https://www.linkedin.com/in/saatvik-managari-2b2379342/
- Skills : C,C++,Python, Verilog, Machine Learning

1.2 Team Member :

- Name : Nemani Sandeep
- Roll Number : 240002044
- Github : https://github.com/Sandeep-i7
- LinkedIn : https://www.linkedin.com/in/sandeep-nemani-1b8b3a366/
- Skills : C, C++, Python, Verilog

1.3 Team Member :

- Name : Nagalla Abhisri Karthik
- Roll Number : 240002041
- Github : https://github.com/Abhi1517621
- LinkedIn : https://www.linkedin.com/in/nagalla-abhisri-karthik-1a8b9a334/
- Skills : C,C++, WebDev, Verilog, Canva

1.4 Team Member :

- Name : Sabbani Nishank
- Roll Number : 240002061
- Github : https://github.com/Nishank9126
- LinkedIn : https://www.linkedin.com/in/nishank-sabbani-4b1bba31b/
- Skills : C, C++, Python, Canva, Machine Learning, Verilog

2 Project

The aim of this project is to build a functioning 8-Bit microprocessor using verilog that should be able to perform tasks on a pipelined architecture. We have planned to achieve the process in the following manner.

2.1 Project Ideas

After researching the multiple microprocessors present on the market, we have temporarily noted a few important aspects involved in the architectures. This process led us to believe in making our own architecture involving multiple features abstracted from different processors and we are trying to implement all of them together into one processor.

The main ideas involve the following.

- Building of a proper 8-Bit register with one-way communication.
- Building of an ALU as well as a Load/Save Unit (LSU).
- Using a multiline instruction set one with the instruction and other with addresses.
- Memory management using the LSU.
- Dividing the registers into 2 categories : Working Registers and Drop Registers.
- Building a C-based instruction set to cooperate with the processor.

I would like to elaborate on a few ideas presented here.

2.2 Multiline Instruction set:

The idea of this instruction model is to break the entire instruction into 2 parts, one called the instruction set which determines the action to be performed by the processor and the other being the address set which determines the location of data transfer and operations performed in the processor. These 2 sets are sent one after the other to ensure the order of code

2.3 Dividing the registers into 2 categories:

The Address set consists of 8-bit that carry the addresses of memory locations as well as the memory location of the registers present in the microprocessor. These register are divided in 2 types the so called working registers and drop registers. As the name suggest the working register the registers that store the information that requires operations to be performed upon.These working registers contain an address of 3- bits i.e. there exists 8 registers. While the drop registers are the registers where the data is dropped after the ALU operations are performed. These are addressed using 2 bit addresses. This entire process ensures a one way data flow only in a circular manner, hence avoiding any data mistransfers or backlogging of the data.

2.4 Program counter:

This is the starting point of the microprocessor. This is where the instructions arrive from the memory unit and enter the microprocessor. Since our model involves a 2 step instruction first the instruction set is sent and then the address set. Here once the Instruction set arrives the instruction is modified and sent to either the LSU (Load/Save Unit) or the ALU (Arithmetic and Logic unit) , followed by the address set (which is sent only to the active unit) which determines the data locations. The address set which is sent to ALU by the registers is in 8-bits which signifies which two working registers(3-bit registers) should be used in operation and to which drop register(2-bit register) the output should go.

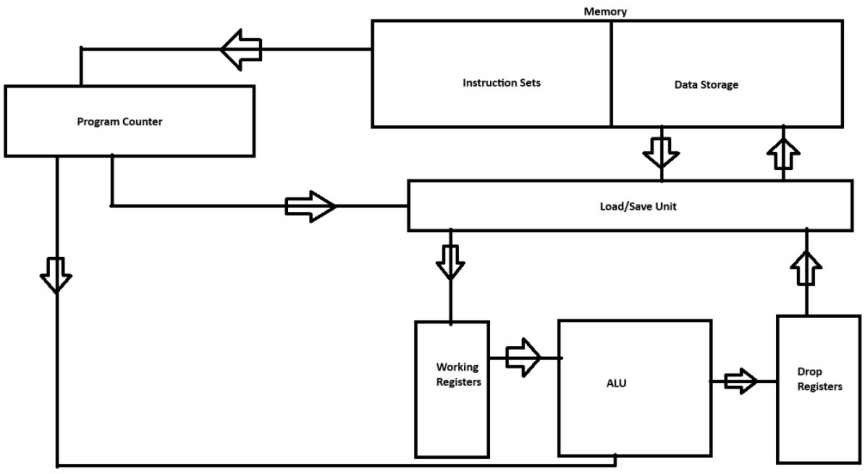

Figure 1: Basic architecture

2.5 LSU - Load/Save Unit:

It is the unit that controls the entire data transfer in the microprocessor. It connects the memory unit with the processor. It loads the data into the working registers and saves the data from the drop registers. It also plays a role in the data transfer from the drop registers to the working registers completing the loop.

2.6 ALU -Arithmetic and logic unit:

It is the unit where all the required operation occur. It connects the working registers to the drop registers. It also plays the important role in the processor as it is the processing power of the processor.

This is the basic architecture of the microprocessor. We are still researching on different units and their internal architecture.

2.7 Pipelining:

Pipelining in computer architecture is a technique that improves processor performance by enabling the execution of multiple instructions simultaneously through a series of stages. It's like an assembly line where different instructions are at different stages of execution at the same time, allowing for faster overall processing.

Example: Imagine an instruction like "add A + B and put the result in C." In a pipelined processor, while the LSU is fetching the values of A and B, the arithmetic-logic unit (ALU) can be simultaneously working on adding another pair of values, and the LSU can be storing the result of a previous calculation.

Major challenge in Pipelining is Hazards.In pipelined processors, hazards are situations that disrupt the smooth, sequential flow of instructions through the pipeline stages, leading to delays or incorrect execution. They can be categorized into three main types: structural hazards, data hazards, and control hazards. We have understood the basic concept of the pipeling and it's importance in increasing the efficiency of the microprocessor and have to research more about it regarding hazard management but the basic architecture is designed taking pipeling as one of the key feature.

2.8 Extra Feature:

We plan on implement a C-Based language to program the microprocessor and use it as per the user's interest. But this idea will be an extra feature and will be implemented if the microprocessor is completed before the stipulated time.

3 Conclusions and important notes

The entire plan is built roughly based on our current research and may be subjected to some changes in the future for better ideas. We plan to research more on the architecture and its flaws and strengths and try to eliminate all of the flaws present in the system and try to build a working 8 bit-microprocessor.

We have done a lot of research on the topic and have come with this plan and are still working on minimizing all the possible mistakes and flaws present in this model.

4 Project Timeline and Work Distribution

4.1 Week 1: Defining our ISA and modeling of the architecture involved

4.2 Week 2: Building of the register systems and modeling of the architecture

4.3 Week 3-4: Building of the ALU and LSU,Memory Buidling

4.4 Week 5: Program counter and instruction set division

4.5 Week 6: Pipeline management and hazard management

4.6 Week 7 -8: Buffer-weeks + Extra feature implementation

5 Verilog Problem Statement

All the members of this team know how to use verilog and we have also attended the verilog workshop conducted by Electronic club of IITI which boosted our knowledge in verilog. So we wrote the code of 7458chip in verilog in the website which is mentioned in the Electronics problem statements of IITI SoC. We wrote them individually , so we are adding four screen shots. And we will also refer the verilog tutorials provided in Problem Statement for any further complicated codes that has to be used in construction of 8-bit processor.

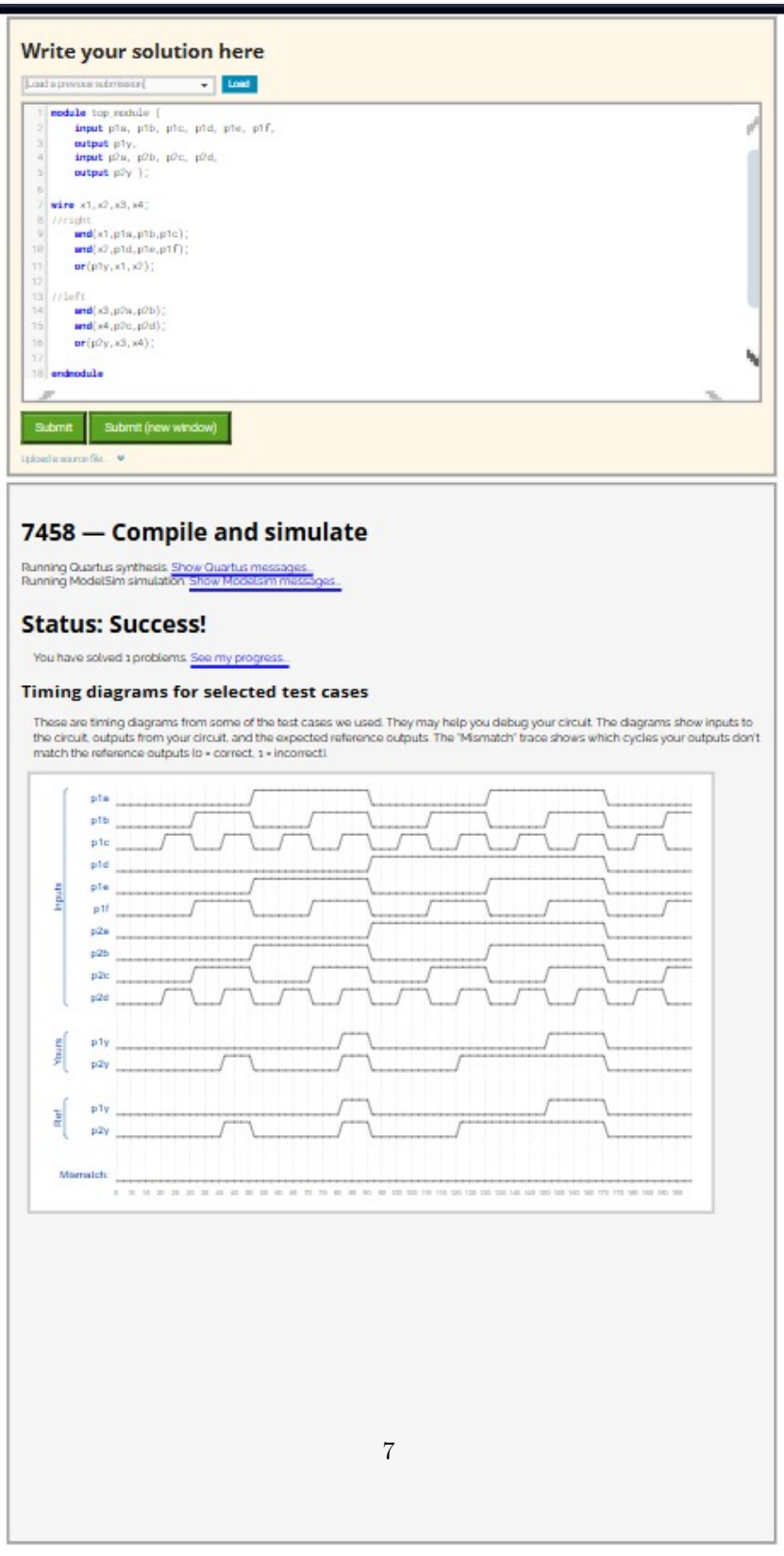

Figure 2: Done by Saatvik

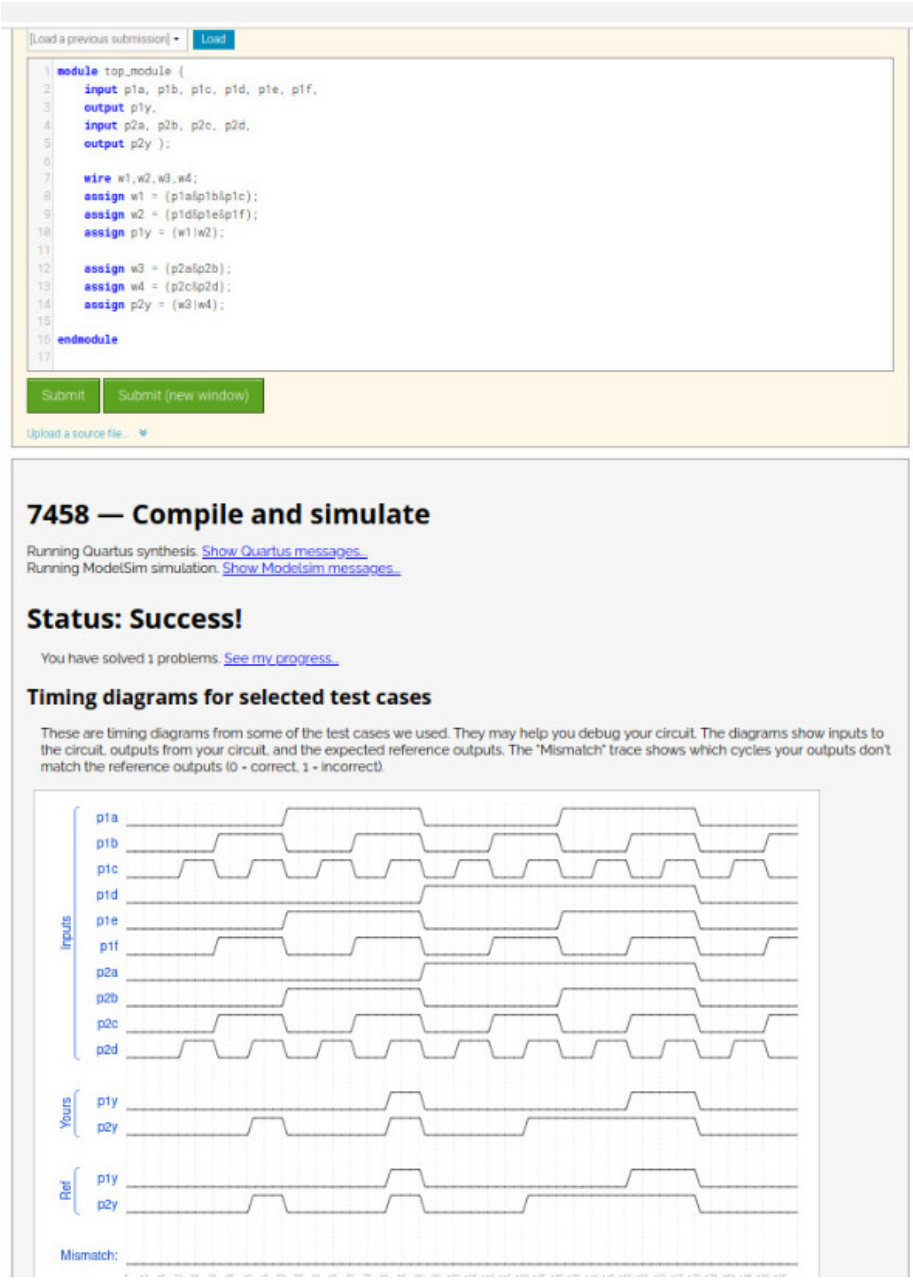

Figure 3: Done by Sandeep

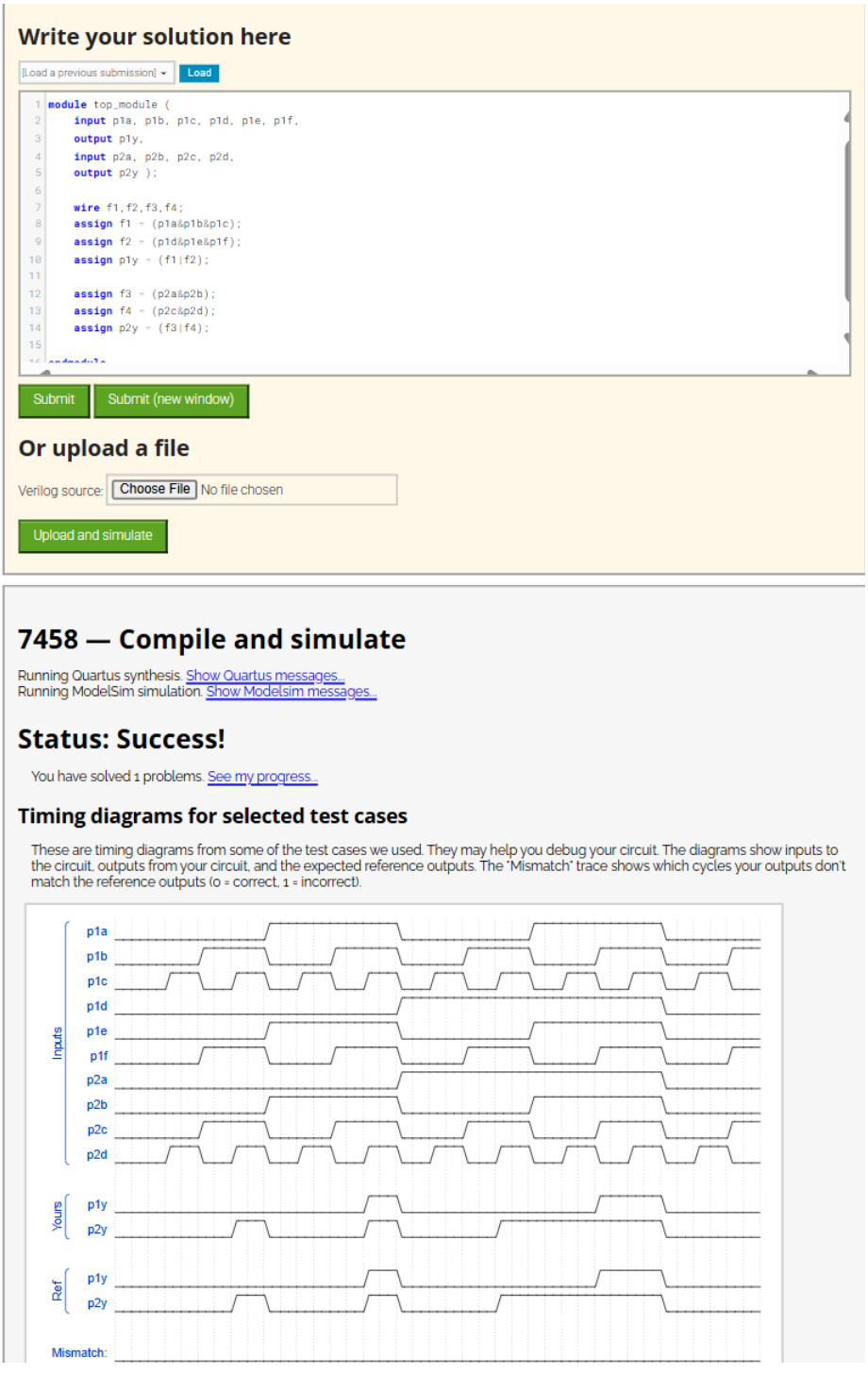

Figure 4: Done by Abhisri Karthik

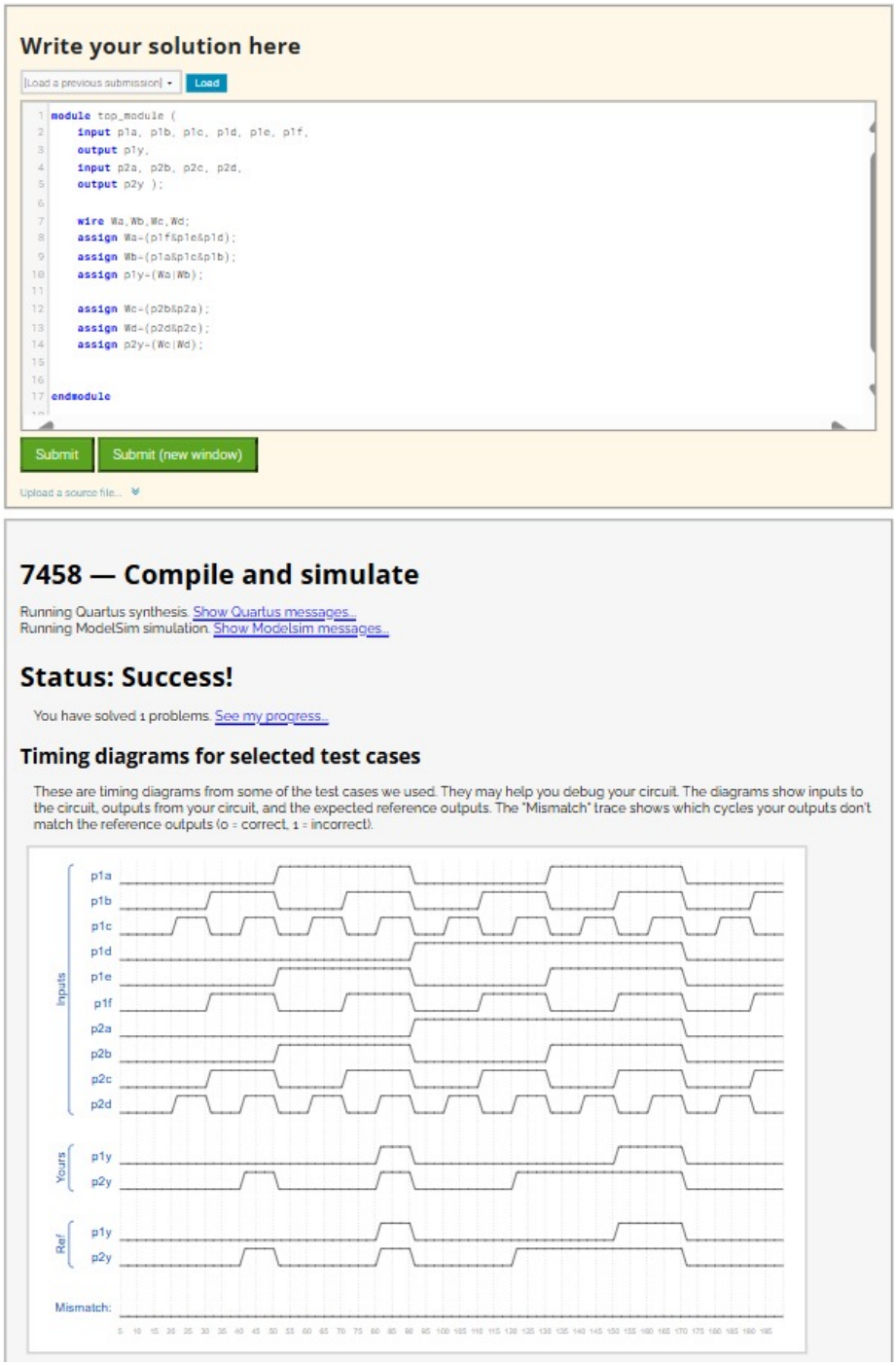

Figure 5: Done by Nishank