CSCE 3301

Computer Architecture

Project 1

Milestone 4

Ahmed Ehab Hamouda                Khalid A. Mohamed

   900172637                                 900153041

Mohamed Elsayed Ehshabshiri (Team Leader)

900171292

**Objective:**

Developing a pipelined datapath block diagram and Verilog description supporting the entirety of RV32I instructions. Basic test cases are needed to cover all supported instructions. A single single-ported memory is used for both instructions and data.

**Apparatus:**

Vivado Design Suite software suite produced by Xilinx for synthesis and analysis of HDL designs: using Verilog HDL. As well as graphical block diagram representations of the mentioned datapath design.

**Method:**

A pipelined processor was designed in the previous milestone in the project. Some - or most – of the cases needed debugging and adjustments. Also, bonuses were attempted to implement in this stage.

Furthermore, the tasks were divided amongst the team members as follows:

1.    Mohamed Elsayed:

-       JAL/JALR debugging & modifications.

-       Flushing unit for jumping and branching.

-       Single memory implementation rechecking and modifications.

-       FPGA implementation.

-       Constraint file.

-       Forwarding unit adjustment.

-       Test cases programs for R,I,S,J,U formats.

-       Drawing the datapath.

2.    Ahmed Ehab:

-       BONUS: RV32C “compressed instructions” support.

-       BONUS: Multiplier unit.

-       System Calls.

-       Readme file.

- Data hazard forwarding unit

- Modifying the ALU ctrl unit

Modifying the control unit

Cleaned the code of my part

Implemented the R and SB format

Test cases for all for all of the above

Synthesized the design

Drawing the datapath

Writing the conclusion in the report

Collecting all test cases in some test files

Submitting the report

3.    Khalid Ahmed:

-       JAL/JALR debugging & modifications.

-       Reorganizing the code’s hierarchy and documentation.

-       Report writing.

-       Test cases and debugging for J instructions.

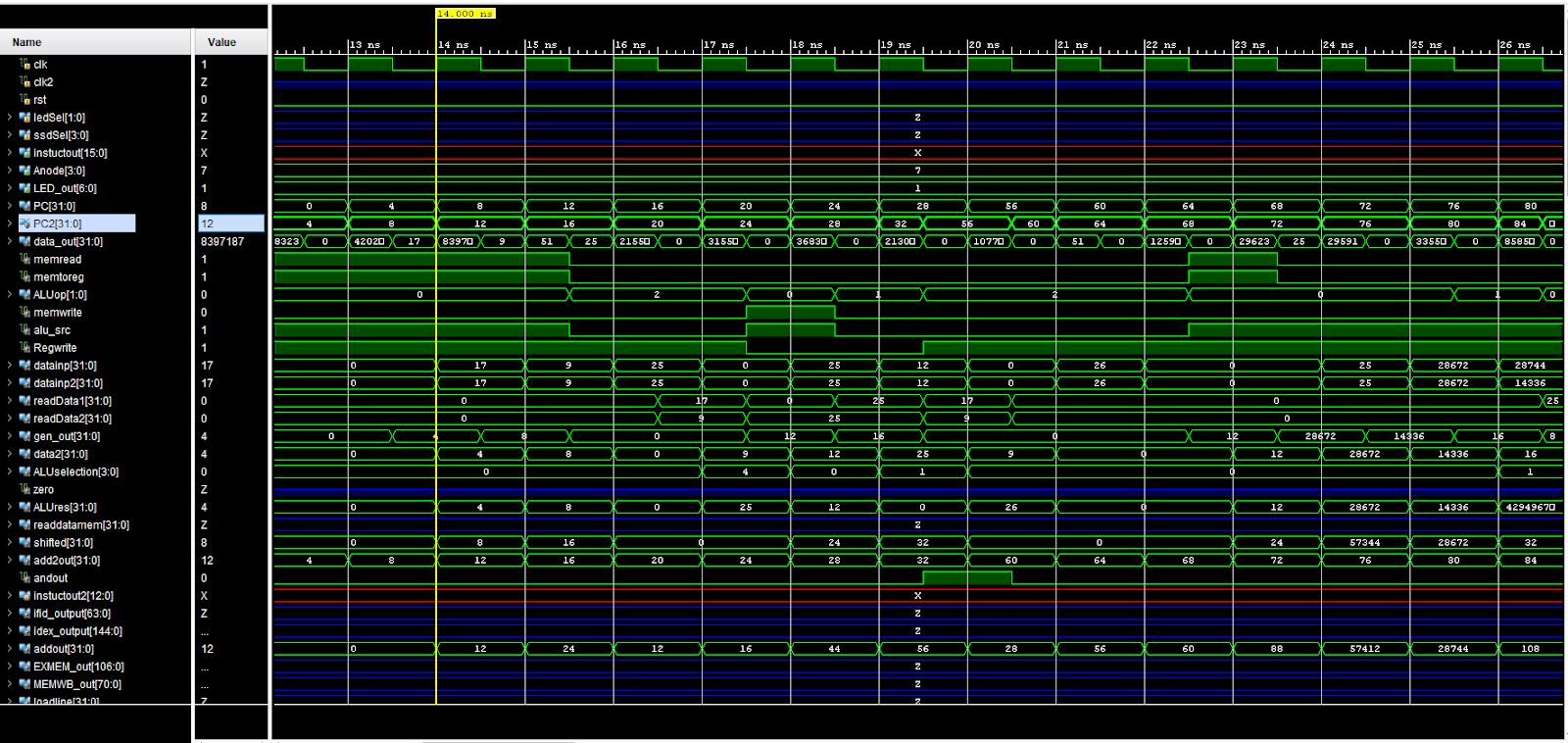
In addition, all the team members contributed variously on the datapath’s modifications, debugging, simulation and test cases.

As an initial step, the tasks were divided to revisit the difficulties faced in the previous milestone. As well as working on the branch predictor bonus. Unfortunately, the branch predictor did not operate properly hence, excluded. Furthermore, though there were some issues with the clock’s level regarding the register file with *store* instructions, the single memory implementation was tackled and successfully implemented. Meanwhile, the compressed instructions’ support and multiplier were designed but not yet included in the code, while debugging the entirety of the RV32I support.

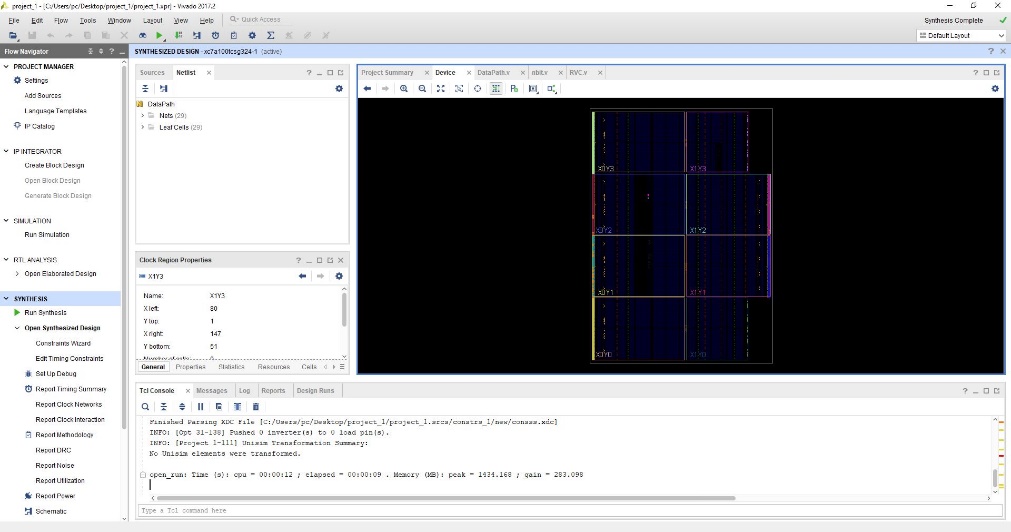
Afterwards, the code was cleaned up, fully documented in comments and organized by stages - graphically categorized. The dot notations were also implemented for the ports. Alas, the code was altered a fair amount during parallel debugging and appending adjustments. It concluded in discarding several steps of the organization to work up with the debugging in combining the running parts of the code. As a slight difference in connections would cost hours of debugging.

Mentioning the alterations happened in parallel, they started with the flushing unit. After it was implemented successfully with minimum trouble, the JAL, JALR, LUI and AUIPC were retraced. Most of the problems faced earlier were regarding the propagation of the PC counter in the pipeline, disregarding the EXE, MEM and WB stages. Caused some logical errors with jumping and branching but solved successfully. Adding an adder after the ID register also helped with adding the right PC + 4 to the *rd* register for JAL & JALR.

**Simulation**

Simulation of the program that tests the I,R,S,J, and U formats. The program can be found in the testbench file in the submission folder.

**Synthesized Design:**

****

**Conclusion and Remarks:**

The pipelined processor was successfully designed, overcoming any hurdles with simulation, synthesis, elaboration and bitstream generation. Connecting and programming to the FPGA had no issues. On the contrary, for some reason the program could not be loaded properly to the board, even though it was tested using various devices and IDEs. It seems that the program is too large for the FPGA board. Implementing a smaller memory might seem to help resolve the issue but the multiplication unit seems to still cause an overload in the gate arrays. In the end, this project helped to achieve an elaborated deep understanding of what’s “under the hood’.