A BEng Project Interim Report

Author:
Minduagas
JARMOLOVICIUS
zceemja@ucl.ac.uk

Supervisor:
Prof. Robert
KILLEY
r.killey@ucl.ac.uk

Second Assessor:
Dr. Ed
ROMANS
e.romans@ucl.ac.uk

DEPT. OF ELECTRONIC AND ELECTRICAL ENGINEERING
UNIVERSITY COLLEGE LONDON

December 13, 2019

1 Abstract

This is abstract.

2 Introduction

Designed RISC has single-cycle datapath with 8bit data bus and 4 general purpose registers.

3 The Work Performed to Date

3.1 Project Scheduling

In this section present your Gantt Chart showing time management. Describe it. Be sure to distinguish term 1 work from term 2 work. Identify the dates of Progress Reports, the Viva, and the due date of the Final Project Report

3.2 Supporting Theory

This section outlines the theory supporting your work. Presumably you have a question that needs to be answered or a problem that needs to be solved. You may have a postulate you are trying to confirm. Whatever it might be, there is some sort of theory underlying it, otherwise you can make no deductions. Without deductions, you cannot identify needed work. Explain this to the readers. In this section the theory

Figure 3.2.1 represents simplified diagrams of RISC and OISC architectures. In RISC architecture, program data travels from program memory to control block where instruction is decoded and further decided where what data is directed. Such structure requires complicated control block and additional data routing blocks. In order to increase performance of such processor one would need to add pipelining or add multiple cores. Both methods bring big disadvantages - multicore processor requires software adjustments and each core doubles the control and datapath substantially increasing die area; pipelinig allow operation at higher frequencies however it brings design complications such as complicated hazard prevention logic and instruction lookup. Simplicity of OISC architecture overcomes these disadvantages by following: Pipelining can be done by individual blocks and programmibly waiting for results, multicore can be simulated by adding more data and instruction buses, hazards can be prevented with software or/and integrated into address registers. In addition, ALU and any other processor component can be divided by adding different address registers thus allowing to utilise multiple components at the same time given that multiple data buses are used.

3.3 Progress to Date

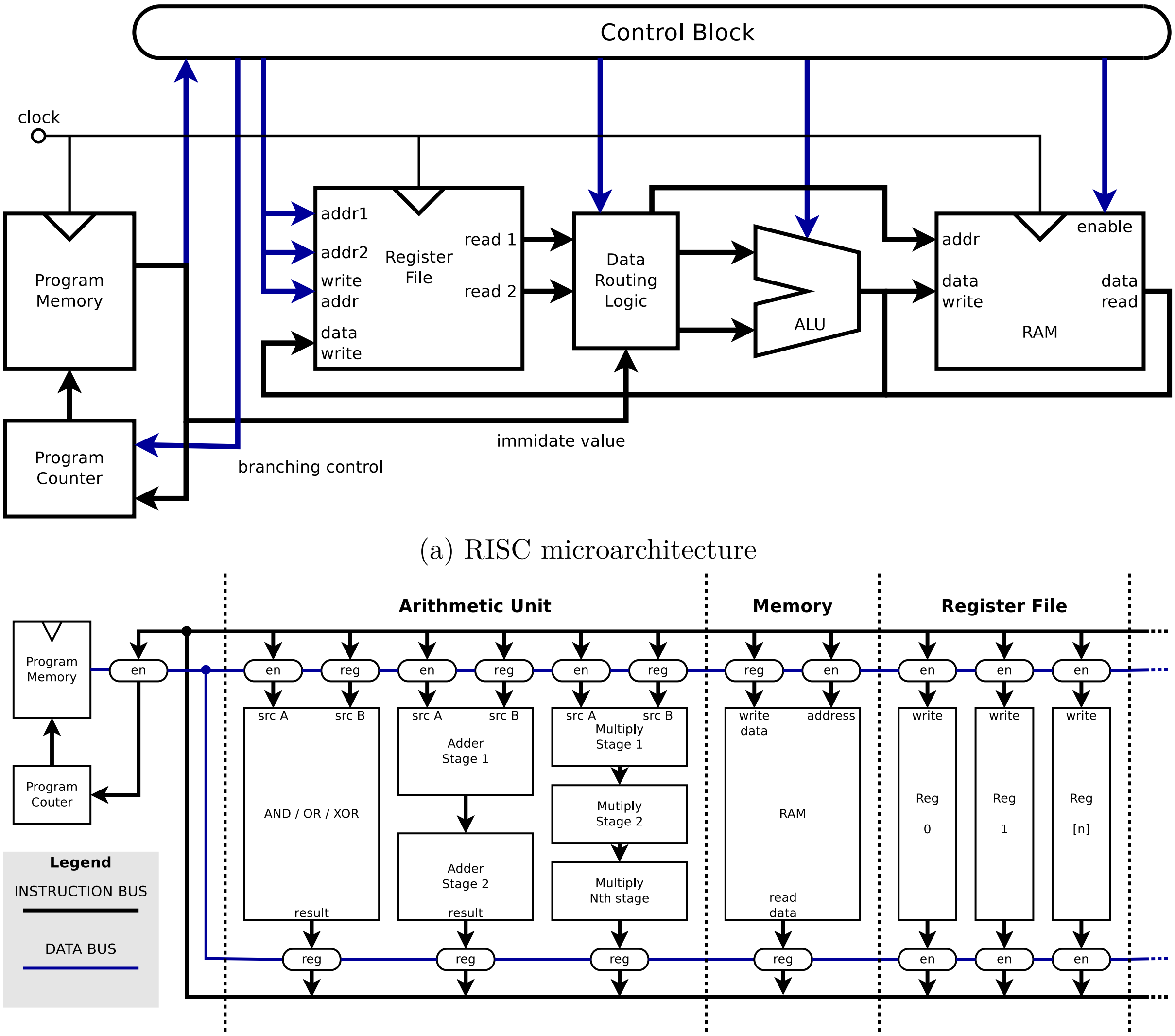
Use as many sections as you need to in order to discuss your progress so far. For example, you can use a different section to discuss each bit of the system to be constructed or designed.Include difficulties and issues impeding progress.

3.3.1 Program Counter

3.3.2 Stack Pointer

4 Summary of Difficulties and Issues

This chapter focuses on any difficulties or issues that are hindering you from moving forward or are slowing you down. Your Supervisor needs to know whether other coursework is keeping you from progressing more rapidly or if parts are not arriving as expected, or if something else is bothering you. You may have made notes of these already in Chapter 2, but now is the time to bring them together and summarise them for your Supervisor.



(a) RISC microarchitecture

(b) Single data bus OISC microarchitecture

Figure 3.2.1: Simplified diagrams of both architectures. Blue lines indicate control/instruction buses and black - data buses

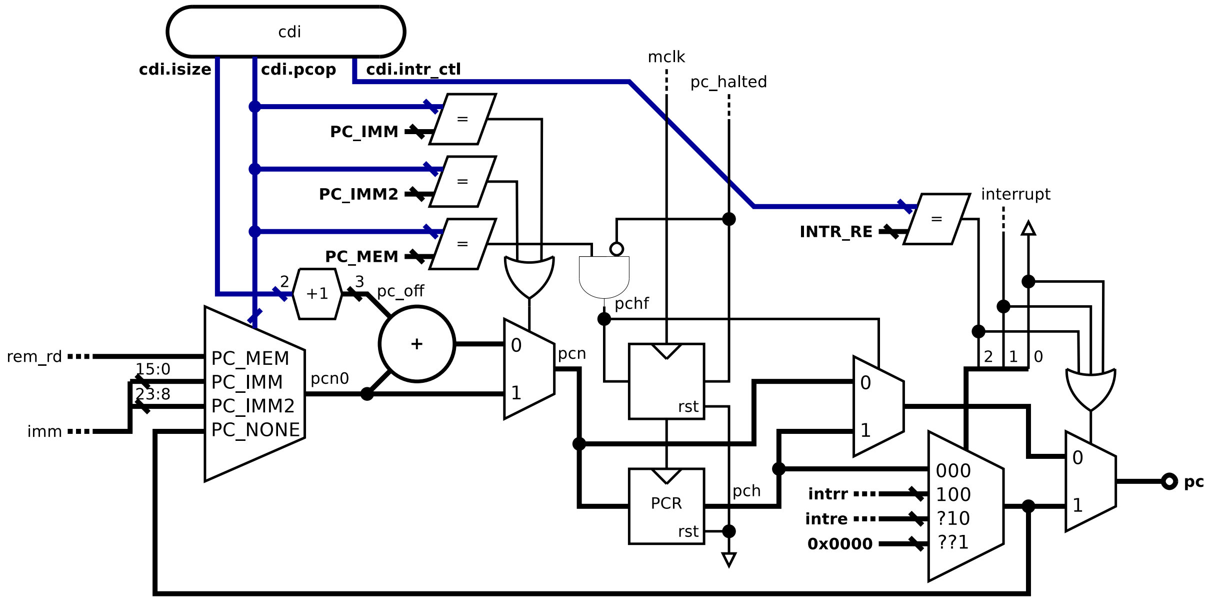


Figure 3.3.1: Digital diagram of RISC8 program counter

4.1 List of Difficulties

Use a bullet list and then explain each one in more detail. You may also refer back to sections in Chapter 2

- Assembler - writing a flexible code that would convert assembly code with labels, sections and macros to machine code;

- Memory - trying to develop easy to use ROM and RAM memory;

- Timing - problems with multiple components in datapath due to RAM and ROM registers;

- JTAG - issues with JTAG interface that communicates with FPGA.

4.1.1 Assembler

4.1.2 Memory

Initial plan was to use 32MBit 16-bit data width, SDRAM chip located at FPGA board. After successfully simulating most of processor functions, next step was to synthesise and run it on FPGA which brought 2 problems:

1. Uploading program into ROM is not simple as generic Verilog unpacked register array cannot be initialised from file. This problem was solved by using FPGA built-in M9K memory that allows flexible RAM/ROM configuration. In addition, this memory can be read from/written to via JTAG connection without affecting FPGA operation. This also enabled quick method to upload programs without need to resynthesising processor code;

2. Timing problems with SDRAM memory controller as it runs at higher frequency than processor (at 100MHz versus 1MHz), used interface between them was multiple 1-word length FIFO registers which caused memory read operation take 2 processor cycles. 3 possible solutions were considered - suspend processor clock while memory data is read, remove FIFO registers and rely on SDRAM clock being much greater than processor clock, use M9K. Last option was chosen due to ease of implementation and ability to read RAM content via JTAG for debugging purposes.

4.1.3 Timing

4.1.4 JTAG

At multiple occasions FPGA could not be controlled due to JTAG error code that indicated that there is problem with commutation between JTAG controller on the board and the rest of a chain. After long investigation it was concluded that this problem is caused by Linux JTAG Daemon service that needs to be simply restarted.

4.2 Failure Assessment

Focus on the possibility of failure in this section and assessthe likelihood ofthe most important steps of the project not reaching a conclusion. Identify all those things that must come together in order for the project to be successful and discover which are the most likely to fail and by what measure. Identify mitigation procedures.

4.3 Updated Safety Risk Assessment

Consider again your Safety Risk Assessment. Is the venue where you are working on the project changing or going to change? Will additional equipment be used that was not included in the first assessment.Will you be doing something hazardous that you did not think about in the first assessment. Update your assessment and submit it to your Supervisor and then to RiskNet.

5 Updated Schedule

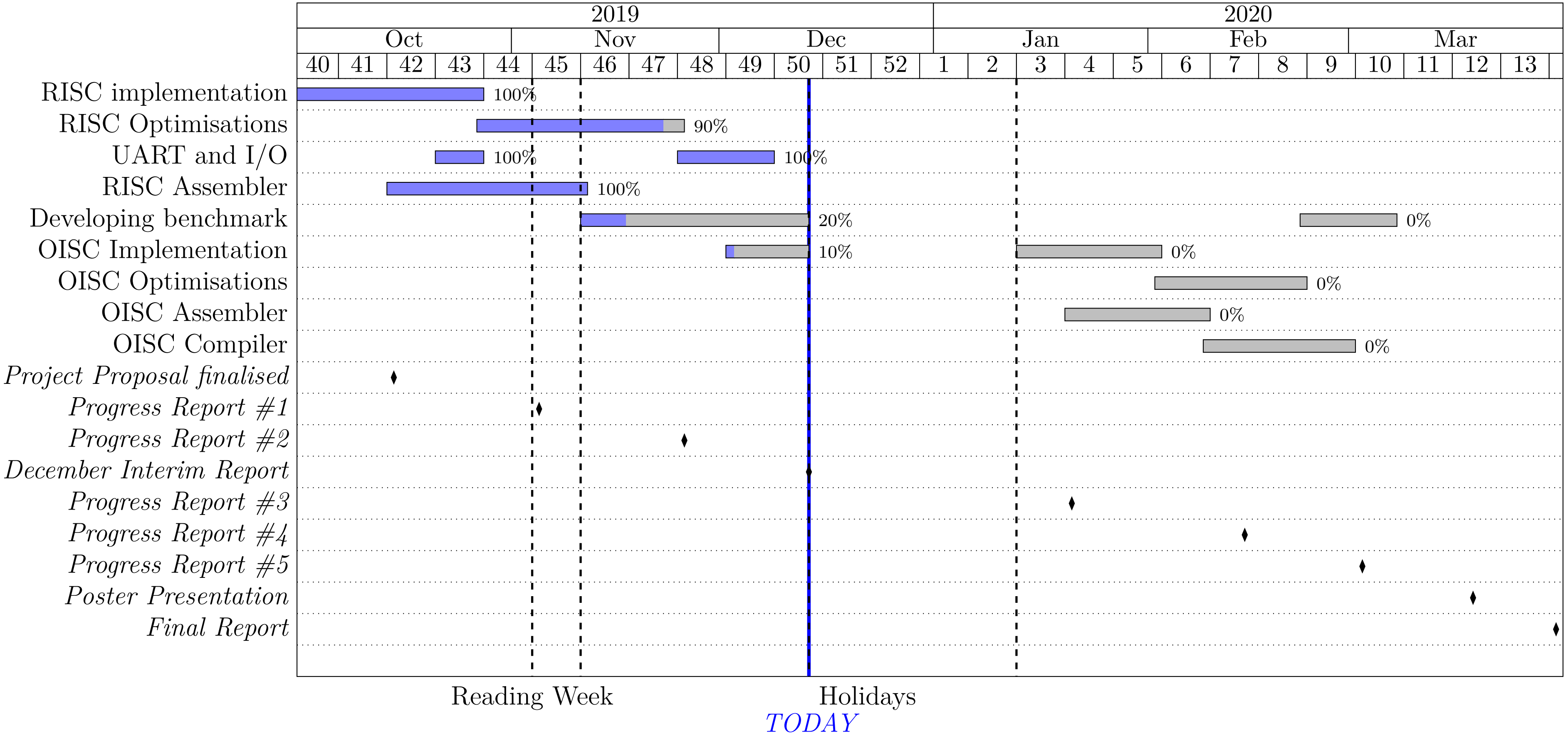


Table 1: Updated project schedule Grantt chart

6 Appendix A: Safety Risk Assessment

Append your updated Safety Risk Assessment whether approved in RiskNet or not. This should be the page that is signed by your Supervisor and by your lab manager(s).

7 Appendix B: Computer Code

Append any code you may need here. Reference it in the text as "Appendix B, code snippet #"; for example,"Finda sample of the code usedfor this experiment in Appendix B, code snippet 2".

8 References