Lab 11 – Design of a Single Cycle RISC Processor

Objectives

In this lab, we will integrate previously designed modules to build a single-cycle RISC-V processor.

Section	⏱
a) Introduction A brief overview of this lab.	**01**
b) Implementation In this section, you will implement a single-cycle RISC processor by integrating previously designed modules.	**60**
Exercise An exercise to verify the functionality of a single-cycle processor including some theoretical explanation.	**30**

a. Introduction

We have developed all the necessary modules of a single cycle processor. We can now combine all the pieces to make a simple datapath for the core RISC-V architecture to run specific set of instructions, which include R-Type, I-Type and Branch-Type instructions.

b. Implementation

In this lab, we will use the modules developed in the previous labs and integrate them together to construct a single-cycle datapath processor. Copy the following modules in a new file path, named Lab11/design:

1. ImmGen.v, Mux.v, InsParser.v from ..Lab06 folder.
2. RegisterFile.v from ../Lab07 folder.
3. ALU_64_bit.v from ../Lab05 folder.
4. Data_Memory.v and Instruction_Memory.v from ../Lab08 folder.
5. Program_Counter.v and Adder.v from ../Lab09 folder.
6. ALU_Control.v and Control_Unit.v from ../Lab10 folder.

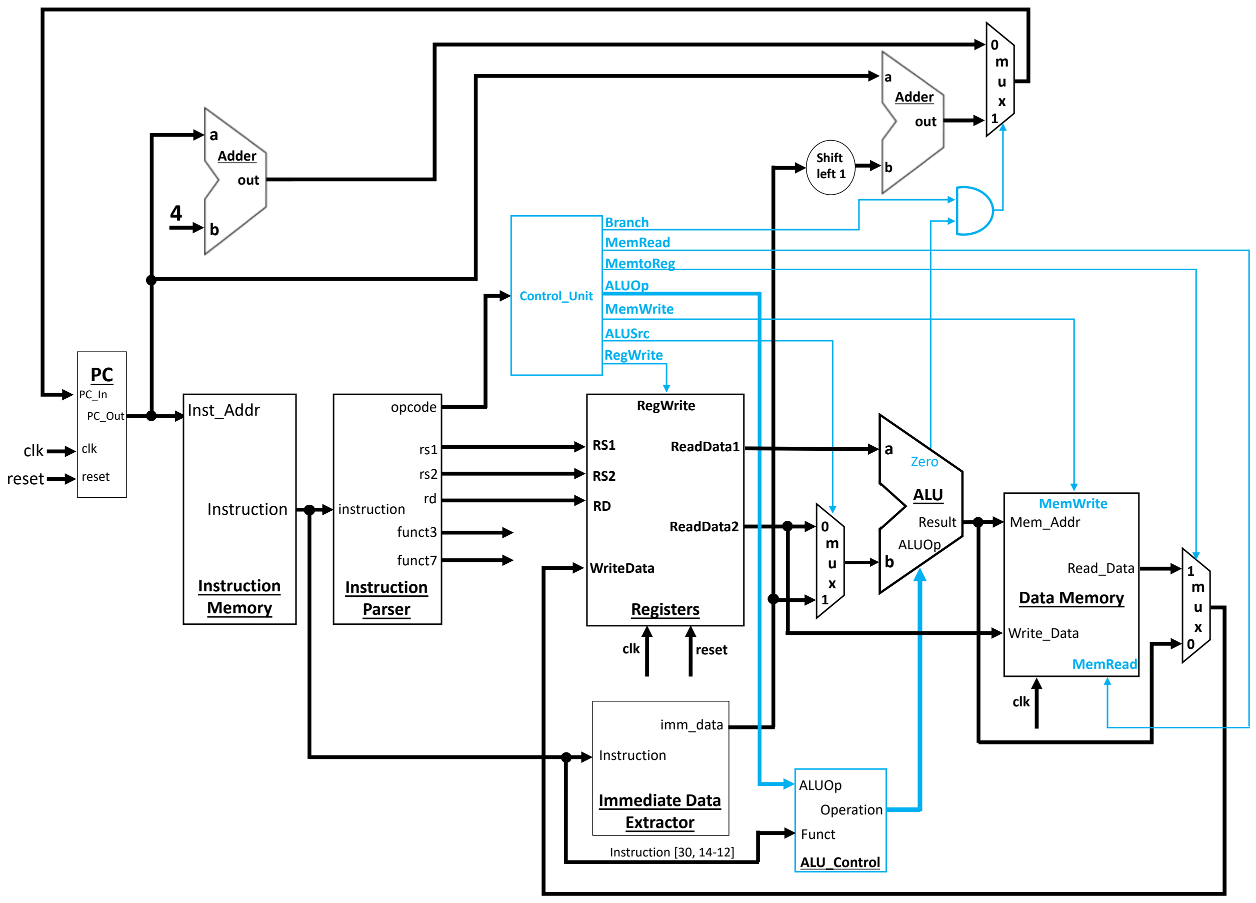

Fig. 11.1. Single cycle processor.

Now integrate all the above modules in a top module named, `RISC_V_Processor`, according to the processor diagram shown in Figure 11.1. The inputs to the top-level module are `clk` and `reset`, and there is no output.

Exercise

Load instruction memory with the contents shown in Figure 11.2 below.

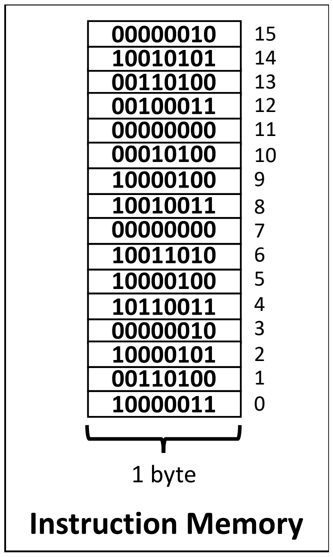

Fig. 11.2. Instruction memory contents for exercise.

Make sure that you already have appropriate values initialized in the Register File and Data Memory. Also make sure that there are 32 registers in the Register File. Perform the following tasks.

1. Write a testbench. Initialize the simulation by first resetting the module under test for, say, 10ns. Toggle the clock signal at every 5ns.

2. Decode the instructions placed in instruction memory, and mention their assembly code below.

Calculate the expected results of decoded instructions using the contents of registers and memory array initialized in modules. Compare them with the simulation results and briefly describe.

Since the top module doesn't have any particular output, so you can verify results by observing the following signals:
Instruction Address
Instruction
Rs1, Rs2, Rd
WriteData
ReadData1, ReadData2
Immediate Data
ALU inputs and ALU_Result
Mem_Address
Data_Memory Read Data

3. Write down its corresponding C code below. Assume that the relevant register used in this exercise is given a variable name “h” and the memory array is given a name “A”.

Assessment Rubric
Computer Architecture Lab
Lab 11
Design of a Single Cycle RISC V Processor

Name:	**Student ID:**

Points Distribution

Task No.	**LR 2** **Code**	**LR 5** **Results**	**AR 7** **Report Submission**
Task 1 **RISC V Processor**	/30	-	/20
Task 2 **Test Bench and Decoded Instructions**	/15	/25	
Exercise (3) **C-Equivalent**	/10	-	
Total Points	/100 Points		
CLO Mapped	CLO 1		

For description of different levels of the mapped rubrics, please refer to the provided Lab Evaluation Assessment Rubrics.

#	**Assessment Elements**	**Level 1: Unsatisfactory Points 0-1**	**Level 2: Developing Points 2**	**Level 3: Good Points 3**	**Level 4: Exemplary Points 4**
LR2	**Program/Code/ Simulation Model/ Network Model**	Program/code/simulation model/network model does not implement the required functionality and has several errors. The student is not able to utilize even the basic tools of the software.	Program/code/simulation model/network model has some errors and does not produce completely accurate results. Student has limited command on the basic tools of the software.	Program/code/simulation model/network model gives correct output but not efficiently implemented or implemented by computationally complex routine.	Program/code/simulation /network model is efficiently implemented and gives correct output. Student has full command on the basic tools of the software.
LR5	**Results & Plots**	Figures/ graphs / tables are not developed or are poorly constructed with erroneous results. Titles, captions, units are not mentioned. Data is presented in an obscure manner.	Figures, graphs and tables are drawn but contain errors. Titles, captions, units are not accurate. Data presentation is not too clear.	All figures, graphs, tables are correctly drawn but contain minor errors or some of the details are missing.	Figures / graphs / tables are correctly drawn and appropriate titles/captions and proper units are mentioned. Data presentation is systematic.
AR7	**Report Content/Code Comments**	Most of the questions are not answered / figures are not labelled/ titles are not mentioned / units are not mentioned. No comments are present in the code.	Some of the questions are answered, figures are labelled, titles are mentioned and units are mentioned. Few comments are stated in the code.	Majority of the questions are answered, figures are labelled, titles are mentioned and units are mentioned. Comments are stated in the code.	All the questions are answered, figures are labelled, titles are mentioned and units are properly mentioned. Proper comments are stated in the code.