**Lab 3. CPU**

1. **Objectives:**
2. Learn how to design single cycle CPU.
3. **Resources:**
4. Download Logisim 2.7.1 from canvas
5. Download “Lab3 - CPU.circ” from Canvas. Inputs, outputs, and most of the necessary electronic components have been given in each circuit. **Do not make any changes on the given Inputs and outputs in the circuits, and Do not change the appearances of electronic components.**
6. **Requirements**
7. Write down your answers to the questions in the following “4. Tasks” section in your lab report. Do not forget to show your demo to TA if required.
8. This lab is separated into two parts. Please submit the report for each part on time.

**<1> Part I (50 points): tasks 4.1 – 4.4, the deadline is 4/14.**

**<2> Part II (50 points): tasks 4.5, the deadline is 4/21.**

1. **Tasks**:

Design a 32-bit CPU that can realize the following functions:

1. Memory-reference instructions: LDUR, STUR
2. Arithmetic-logical instructions: ADD, SUB, AND, and ORR
3. Branch instructions: compare and branch on zero (CBZ)

**4.1 Design a register file with 32 32-bit registers (10 points)**

Explain your design method:

Each register is enabled by RegisterWrite AND its own write bit (WX). The actual data is from the WriteData and is outputted to its own read bit (RX). 32 of these are in parallel and are all clocked and reset by the clock and reset.

Circuit:

Diagram

Description automatically generated with low confidence

**4.2 Design a sign-extend unit for the CPU (10 points)**

Explain your design method:

32-bit code is split and sent through a signal extender and fed into a multiplexer which is selected by bit 26 from the original 32-bit code.

Circuit:

Diagram, schematic

Description automatically generated

**4.3 Design an ALU control unit (15 points)**

Truth table:

A picture containing text, chain, metalware

Description automatically generated

Generated circuit by Logisim:

Diagram, schematic

Description automatically generated

**4.4 Design a CPU control unit (15 points)**

Explain your design method:

The eleven bits are sent through large AND gates to resemble codes for CBZ, STUR, LDUR, and R-Type instructions. The output gets sent to OR gates and to the circuit output.

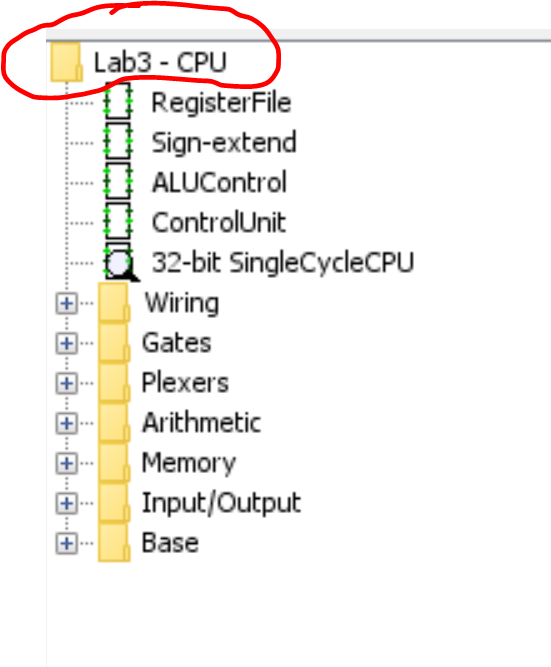
Circuit:

Diagram, schematic

Description automatically generated

**4.5 Design a 32-bit single cycle CPU and demo your design to TA (50 points)**

Please note: The 32-bit ALU designed in Lab2 is needed for the CPU design. (1) Before using the ALU in the CPU design, change the “set on less than” function to “pass b”. (2) The ALU circuit can be loaded through this process: right click the project name which is highlighted in the following figure, select “load library”, select “Logisim library”, select the ALU circuit you designed.



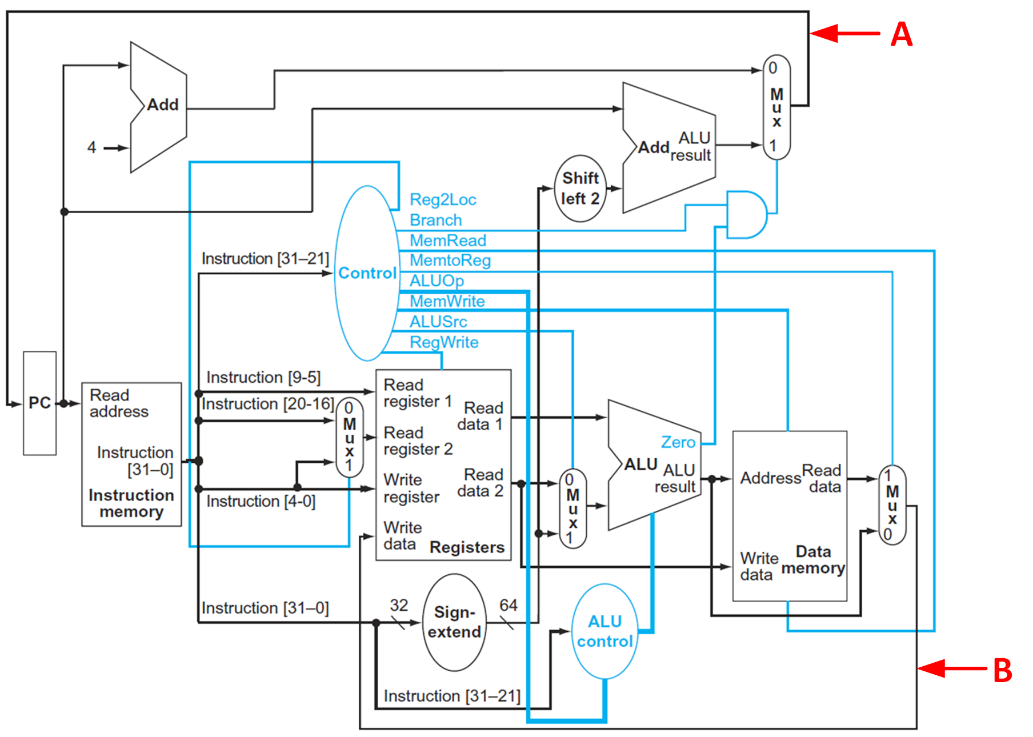
Explain your design method:

Implement the circuit from the slides exactly how it is presented. I did have to go back and fix a couple of circuits from lab 2 that were used in the ALU.

Circuit:

Diagram, schematic

Description automatically generated



Add binary or hex probes to location A and B to show the values after each clock. Include the results in the report.

Testing cases:

<1> Testing instructions have been loaded to ROM. Do not change its value.

|  |
| --- |
| f841804d |
| f845006e |
| 8b0e01a9 |
| 8a0901aa |
| aa0a012b |
| cb09016c |
| f802004c |
| b4ffff33 |

<2> Before demo, make sure “00012345” and “00067890” are loaded to the word address of 6 and 20 in RAM, respectively. You can right click the RAM and select “edit contents” to initialize the values in RAM. You can reference the following figure.

