**Assignment #8 – MIPS Processor Design**

CDA 3100, Computer Organization I

⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

**Problem 1 (50 points)** Design a MIPS processor supporting **only** the **R-type** and the **jset rs, rt, rd** instruction. The jset rs, rt, rd instruction does the following:

**If rs is not equal to rt, the next PC is the memory content at rs-rt and rd is set to be rs-rt; otherwise, the next PC is PC+4 and register $ra is set to be PC+4.**

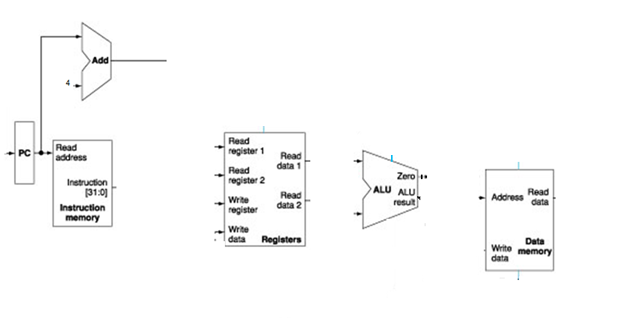
Recall that **$ra** is the special register used in the jal instruction.

For example,

* if rsandrtare 20 and 8 respectively, rd will be 12, and PC will be the memory content at 12.
* if rsandrtare 20 and 20 respectively, rd will not be changed, PC will be **PC+4**, and **$ra** will be **PC+4.**

For this problem, assume that the opcode of **R-type** is 000000 and the opcode of jset is 100000.

1. (30 points) Show the data path of this processor, add 2-1 MUX when necessary. Besides a group of wires, please show clearly the indices of the bits.
2. (20 points) Consider the control signals for the 2-1 MUX. Please fill in the table (you should not need more than 4 2-1 MUXes). In case of “don’t care,” write down 0. Then write down the logic functions. Certain bits in the instruction can be denoted as, for example, ins[31]. The ALU zero output can be referred to simply as “z.”



|  |  |  |  |
| --- | --- | --- | --- |
|  | R-type | jset (rs != rt) | jset (rs == rt) |
| MUXCtrl1 |  |  |  |
| MUXCtrl2 |  |  |  |
| MUXCtrl3 |  |  |  |
| MUXCtrl4 |  |  |  |

**Problem 2 (50 points)** Design a MIPS processor supporting **only** the **lw** and the **mgt rs, rt, rd** instruction. The mqt rs, rt, rd instruction does the following:

**if rs >= rt, rd is set to be rs; otherwise the next PC is set to be rt.**

For example,

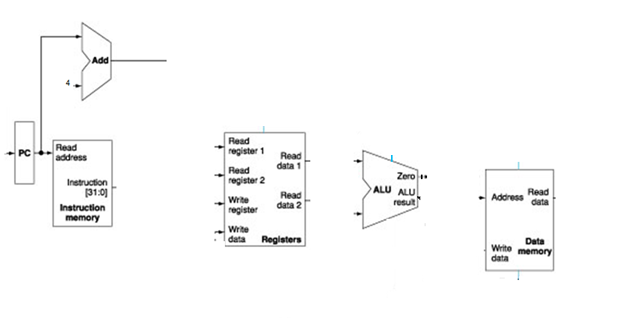
* if rsandrtare 20 and 8 respectively, rd will be set to 20, and PC will be PC+4.
* if rsandrtare 20 and 32 respectively, rd will not be changed, and PC will be 32.

**In this problem, the simplifying assumption is that the values in rs and rt are always in [-230,230].**

For this problem, assume that the opcode of lw is 000000 and the opcode of mgt is 100000.

Recall that in MIPS, ALUCtrl is 0000 for and, 0001 for or, 0100 for add, 0110 for sub.

1. (30 points) Show the data path of this processor, add 2-1 MUX when necessary. Besides a group of wires, please show clearly the indices of the bits.
2. (20 points) The control signals include ALUCtrl, RegWrite, and the signals to control the added 2-1 MUX. Please fill in the table (you should not need more than 4 2-1 MUXes). In case of “don’t care,” write down 0. Then write down the logic functions for RegWrite and the MUXCtrls. Certain bits in the instruction or in ALUResult can be denoted as instruct[31].



|  |  |  |  |
| --- | --- | --- | --- |
|  | lw | mgt (rs >= rt) | mgt (rs < rt) |
| ALUCtrl |  |  |  |
| RegWrite |  |  |  |
| MUXCtrl1 |  |  |  |
| MUXCtrl2 |  |  |  |
| MUXCtrl3 |  |  |  |
| MUXCtrl4 |  |  |  |