**Course**: ENCM 369

**Lab Section:** B03

**Lab 7**

**Student Name**: Mitchell Sawatzky

**Date Submitted**: March 11, 2016

Exercise A

Part I

lw $t0, 8($s2)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

|  |  |
| --- | --- |
| Signal | Value |
| MemtoReg | 0b1 |
| MemWrite | 0b0 |
| Branch | 0b0 |
| ALUControl | 0b010 |
| ALUSrc | 0b1 |
| RegDst | 0b0 |
| RegWrite | 0b1 |
| A1 | 0b10010 |
| A2 | 0b01000 |
| A3 | 0b01000 |
| SrcA | 0x1001003c |
| SrcB | 0x00000008 |
| ALUResult | 0x10010044 |
| WD3 | 0x00004444 |
| PCBranch | 0x004000c4 |

Part II

and $s2, $s2, $t0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

|  |  |
| --- | --- |
| Signal | Value |
| MemtoReg | 0b0 |
| MemWrite | 0b0 |
| Branch | 0b0 |
| ALUControl | 0b000 |
| ALUSrc | 0b0 |
| RegDst | 0b1 |
| RegWrite | 0b1 |
| A1 | 0b10010 |
| A2 | 0b01000 |
| A3 | 0b10010 |
| SrcA | 0x1001003c |
| SrcB | 0xffffffe0 |
| ALUResult | 0x10010020 |
| WD3 | 0x10010020 |
| PCBranch | 0x003e4134 |

Exercise B

The main and ALU decoders are modified by simply connecting wires to the relevant places on the decoder, as every new instruction that is being added has a unique opcode. The ALU Decoder will need to be changed to support the new 3-bit ALUOp bus. The SZ input of the Extend unit should be connected to the Control Unit so that main decoder can power it off when it encounters andi or ori.

ALU Decoder Truth Table

|  |  |  |
| --- | --- | --- |
| ALUOp | Funct | ALUControl |
| 000 | X | 010 (add) |
| 001 | X | 110 (sub) |
| 010 | 100000 (add) | 010 (add) |
| 010 | 100010 (sub) | 110 (sub) |
| 010 | 100100 (and) | 000 (and) |
| 010 | 100101 (or ) | 001 (or ) |
| 010 | 101010 (slt) | 111 (slt) |
| 011 | X | 111 (slt) |
| 1X0 | X | 000 (and) |
| 1X1 | X | 001 (or ) |

Main Decoder Truth Table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction | Opcode | RegWrite | RegDst | ALUSrc | Branch | MemWrite | MemtoReg | SZ | ALUOp |
| R-type | 000000 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 010 |
| lw | 100011 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 000 |
| sw | 101011 | 0 | X | 1 | 0 | 1 | X | 1 | 000 |
| beq | 000100 | 0 | X | 0 | 1 | 0 | X | 1 | 001 |
| addi | 001000 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 000 |
| slti | 001010 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 011 |
| andi | 001100 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 100 |
| ori | 001101 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 101 |

Exercise C

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction | ShifttoReg | ShiftDir | RegWrite | RegDst | ALUSrc | ALUControl | Branch | MemWrite | MemtoReg |
| ADD | 0 | X | 1 | 1 | 0 | 010 | 0 | 0 | 0 |
| SUB | 0 | X | 1 | 1 | 0 | 110 | 0 | 0 | 0 |
| AND | 0 | X | 1 | 1 | 0 | 000 | 0 | 0 | 0 |
| OR | 0 | X | 1 | 1 | 0 | 001 | 0 | 0 | 0 |
| SLT | 0 | X | 1 | 1 | 0 | 111 | 0 | 0 | 0 |
| LW | 0 | X | 1 | 0 | 1 | 010 | 0 | 0 | 1 |
| SW | 0 | X | 0 | X | 1 | 010 | 0 | 1 | X |
| BEQ | 0 | X | 0 | X | 0 | 110 | 1 | 0 | X |
| SLL | 1 | 1 | 1 | 1 | X | XXX | 0 | 0 | X |
| SRL | 1 | 0 | 1 | 1 | X | XXX | 0 | 0 | X |

