**Max Score = 15 points**

CS 250 2018 Spring Homework 06

This assignment is due at 11:59:00 pm Thursday, March 01, 2018.

Upload your typewritten answer document in either PDF or Word format to Blackboard. Then, download from Blackboard to be sure that your upload was successful.

Your last upload that is not marked “LATE” by Blackboard is the upload that will be graded. There is no “grace” period for late uploads.

1. Refer to textbook Figures 5.9 and 5.10.
   1. Name an instruction that implies that a register in the MIPS architecture can store a bit string that will be interpreted using IEEE 754 double precision format.  
      load word coprocessor
   2. Name an instruction that implies that a register can store a bit string having no associated representational form.   
      load word
   3. Name the instruction that corresponds to the C language code  
       if (a != b){ code\_to\_execute\_when\_condition\_is\_true }.  
      Assume that the values of a and b are held in registers.  
      branch not equal
   4. How many bits are necessary for the opcode field for the instructions in Figure 5.9?   
      5 bits
2. Imagine that Figure 6.9 has been modified to include the register-to-register type operation Shift Right Logical (SRL). Assume that a circuit is available within the ALU that takes as inputs a 32-bit operand to be shifted and a 5-bit unsigned integer shift amount. Bits shifted past the LSB are lost to the circuit; 0 bits shifted in at the MSB as needed.
   1. Design how SRL will use each field of the instruction format of Figure 6.2 and describe your design. As the designer, fill in the table with the bit strings of your choice where a fixed choice is possible. When a fixed choice is not possible, state the purpose of the instruction format field. Where neither a bit string nor a purpose is appropriate, state “Unused”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mnemonic | Opcode | Reg A | Reg B | Dst Reg | Offset |
| bits | 31 – 27 | 26 – 23 | 22 – 19 | 18 – 15 | 14 – 0 |
| SRL | **The opcode that we assigned for this operation** | **The 32-bit operand** | **unused** | **The bit string after shifting** | **5-bit unsigned integer shift amount** |

* 1. What is the result of the assembly language instruction  
      SRL r3, 4(r2) ; r3 🡨 4-bit-position right shift logical of contents of r2  
     if register r2 contains 0xA5A5FF00?

0000 1010 0101 1010 0101 1111 1111 0000

* 1. What is the result of the assembly language instruction  
      SRL r3, 40(r2)  
     if register r2 contains 0xA5A5FF00?

0000 0000 1010 0101 1010 0101 1111 1111

1. Modify the circuit in Figure 6.9 to include the branch-type instruction Branch Relative (BRR). BRR takes as operands the default\_next\_instr.ptr and the offset field for the integer adder in the ALU of Figure 6.9. The bit string result from the ALU is forwarded along the existing data buses so that it becomes the next\_instr\_ptr.
   1. Design the bit string for BRR by filling in the instruction format table. As the designer, fill in the table with actual bit strings of your choice where possible, show “unused” when appropriate, and state your chosen integer representation for the Offset field.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mnemonic | Opcode | Reg A | Reg B | Dst Reg | Offset |
| bits | 31 – 27 | 26 – 23 | 22 – 19 | 18 – 15 | 14 – 0 |
| BRR | **The opcode that we assigned for this operation** | **unused** | **unused** | **unused** | **2’s complement** |

* 1. What new device(s) must be added to the circuit in Figure 6.9 to support BRR? How is (are) these device(s) connected to the existing devices of Figure 6.9?  
     A multiplexer that inputs program counter and output to ALU.
  2. List all new buses to be added to Figure 6.9 in support of the implementation of BRR. Define each bus by stating its origin and its destination. Use the labels for components in Figure 6.9 as reference points for naming origins and destinations. Include the device(s) that you added in part b in your bus names as needed.

A bus from program counter to the new multiplexer and a bus from regA to the new multiplexer. And from the new multiplexer to ALU.

* 1. What existing bus(es) in Figure 6.9 must be re-wired to a new destination?  
     The original bus that connects register unit and ALU.
  2. When the circuit of Figure 6.9 is executing a BRR instruction what are the selections made by each multiplexer in the circuit? To answer this question for each mux, provide the name of the selected bus in terms of the source of the bus. For example, the bus between M2 and the register unit data in inputs would be called the M3 output bus.   
     M1: select from M3 output and default next instruction pointer

M2: select from M3 output and M1 output

M3: ALU and data out

M4: default next instruction pointer and regA

1. Here follows a schematic for the circuitry inside the register unit in textbook Figure 6.9. The triangular devices labeled BUF are buffers, circuits that allow signals to pass un-inverted when the buffer is selected, otherwise the buffer acts as an open switch. Clearly, and legibly label this schematic with the following information.
   1. reg A inputs
   2. reg B inputs
   3. dst reg inputs
   4. reg A outputs
   5. reg B outputs
   6. except for the 4-to-16 decoder inputs, indicate how many wires each bus in the schematic actually represents using slash/number notation.
   7. Two of the decoders have an unlabeled input; add a likely label for these two inputs  
        
      