Digital Design and Computer Architecture

J. Spjut and R. Wang **Final** **Spring, 2013**

This is an open book, open notes take-home exam. You are allowed at most **3 hours** to take the exam. The exam is due back to Prof. J. Spjut's office no later than 2:00 p.m. on Monday May 6, 2013. If Prof. Spjut is not present, you may leave the exam in the box or under the door.

You may not discuss the exam with anybody else except Prof. Wang or Prof. Spjut and you may not consult any references outside of the book and your notes or use a computer until you have turned in the exam. Remember that the Harvey Mudd College Honor Code applies.

The number of points for questions on each page is shown below. The entire exam is worth 100 points. All work and answers should be written directly on this examination booklet. Use the backs of pages if necessary. Write neatly; illegible answers will be marked wrong. **Show your work** for partial credit. If you need to separate the pages, be sure to write your name on the top of each page and staple the test before handing it in. Clearly label any work you do on the extra sheet of paper in the back of the exam.

Name: _____

Page 2: _____/8

Page 3: _____/11

Page 4: _____/10

Page 5: _____/12

Page 6: _____/8

Page 7: _____/12

Page 8: _____/12

Page 9: _____/11

Page 10: _____/16

Total: _____/100

Name:_____

1. [4] What is the **minimum** number of instructions past the current (i.e. the jump) instruction that the jump instruction `j` can jump to? Hint: Think of the extreme case.

Answer:_____

2. [4] What is the **maximum** number of instructions past the current (i.e. the jump) instruction that the jump instruction `j` can jump to? Hint: Again, think of the extreme case.

Answer:_____

Name:_____

3. [5] Convert the two's complement number 0x583C shown in hex to decimal.

Answer:_____

4. [6] Add the following IEEE single-precision floating point numbers (shown in binary). You may leave your result in binary form.

```
  1 10000101 10110100100000000001000
+0 10000101 10011000000000000001000
```

Sum = _____

Name:_____

5. a) [10] Translate the following procedure into MIPS assembly language. Remember to properly save and restore registers onto the stack, and appropriately decrement and increment the stack pointer. Be sure to comment your code thoroughly. Argument 'a' should be in register $a0 and argument 'b' should be in register $a1.

```
int test(int a, int b) {

     if (a  > b)
          return 1;
     else
          return (a + test(a+2, b));

}
```

b) [7] What is the final returned value from your procedure `test` from part (a) if the argument `a` is -4 and b is 7 when `test` is first called? You may find it useful to draw a picture of the stack during execution.

Answer:_____

c) [5] How would your code function if you failed to store `$ra` on the stack? Be specific in your answer.

Answer:_____

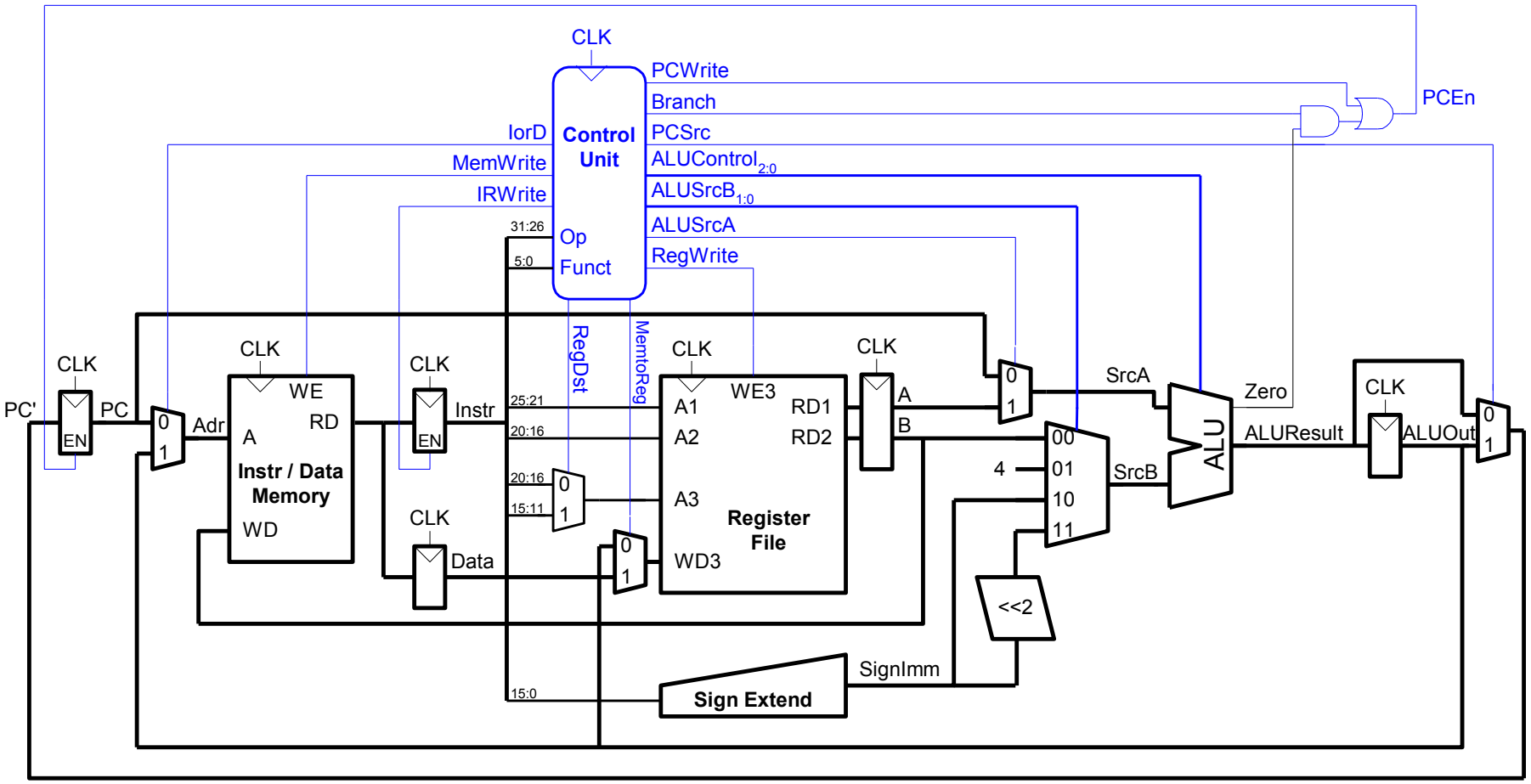
6. You would like to add the `slti` command to your multicycle MIPS processor. Recall that

```
slti $t0, $t1, 42
```

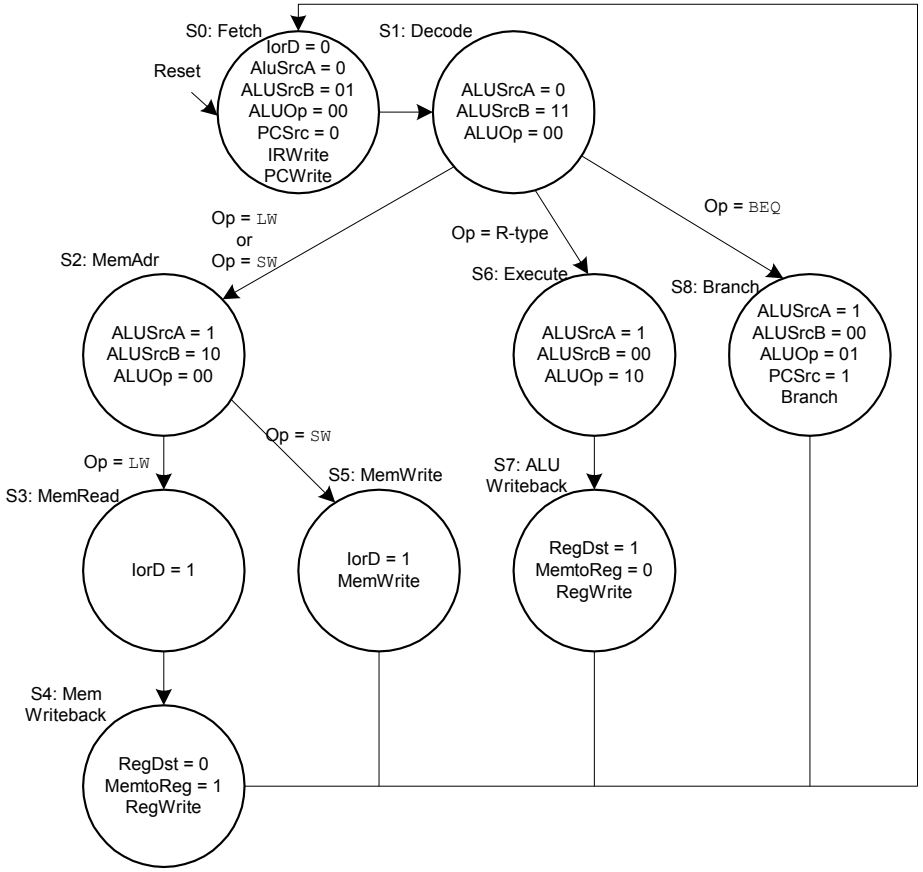
stores 1 in $t0 if $t1 is less than 42 and stores 0 in $t0 otherwise.

a) [8] Mark up the datapath on the next page with your changes. Do not change the number of bits in existing control signals. Use as few new control signals as possible. Unreadable, incomplete, or messy answers will receive no credit.

Name:_____


CLK
PCWrite
Branch
PCSrc
ALUControl2:0
ALUSrcB1:0
ALUSrcA
RegWrite
IorD
MemWrite
IRWrite
31:26
5:0
Op
Funct
Control Unit
PCEn
RegDst
MemtoReg
PC'
PC
Adr
WE
RD
A
Instr / Data Memory
WD
Instr
Data
25:21
20:16
15:11
A1
A2
A3
WE3
WD3
RD1
RD2
Register File
A
B
SrcA
SrcB
4
00
01
10
11
ALU
Zero
ALUResult
ALUOut
<<2
SignImm
15:0
Sign Extend
EN
0
1

b) [12] Add states to the multicycle FSM shown below to support `slti`. Use as few new states as reasonable.



(modified state machine)

7. Consider a memory hierarchy with a 65536 (64k) byte cache memory and a 4294967296 (4G) byte main memory. Assume a set-associative cache with 32 byte blocks and 16 blocks per set.

a) [6] For the CPU to fetch a byte from main memory, how many address bits are needed?

b) [6] How are the address bits organized into TAG, SET, and OFFSET fields. Specify in order from most significant bit to least significant bit.

c) [5] Given a memory address issued by the CPU, specify the steps to find the set, block containing the byte needed. How do you find out whether the byte is in the cache or not?

d) [6] For the addresses, 0x01234C4D and 0x9ABEFC52, specify which sets need to be searched to find them.

8. [16] Draw a circuit schematic for the Verilog module listed below.

```
module my_design( input  logic [31:0] x, y,
                  input  logic [2:0]  w,
                  output logic [31:0] z);

  logic [31:0] bb, b2, r1, r2, r3, r4;

  assign y2 = ~y;
  assign y3 = w[2] ? y2 : y;
  assign r1 = x + y3 + w[2];
  assign r2 = x & y3;
  assign r3 = x | y3;
  assign r4 = r1[31] ? 32'b1 : 32'b0;

  always_comb
    case (w[1:0])
      2'b00: z = r2;
      2'b01: z = r3;
      2'b10: z = r1;
      2'b11: z = r4;
    endcase

endmodule
```

Name:_____