UNIVERSITY OF TORONTO

Faculty of Arts and Science

April 2015 Examinations

CSC258H1S: Computer Organization

Duration: 3 hours

Permitted Aids: one ruler, one highlighter

Last Name: _	
First Name:	
Student Num	ber:
Instructors:	Steve Engels (L0101)
	Larry Zhang (L0201)
	Myrto Papadopoulou (L5101)

Mark Breakdown Instructions: / 34 Part A: Write your name on every page of this exam. / 26 Part B: • Do not open this exam until you hear the signal / 17 to start. Part C: Have your student ID on your desk. / 27 Part D: No aids permitted other than writing tools. Keep all bags and notes far from your desk before the / 12 Part E: exam begins. • There are 6 questions on 16 pages. When you hear / 24 Part F: the signal to start, make sure that your exam is /1 complete before you begin. **Bonus:** • Read over the entire exam before starting. If you use any space for rough work or have to use Total: **/ 140** the overflow page, clearly indicate the section(s) that you want marked.

Part A: Short Answer (34 marks)

Answer the following questions in the space provided. When providing a written answer, write \underline{as} clearly and legibly as possible. Marks will not be awarded to unreadable answers.

clearly and legibly as pos	s sible . Marks will not be awa	rded to	unreadable answers.
	llowing hexadecimal addresses? Circle all that apply.		e valid PC values when translated
a)	E45B23C1	b)	4C32A330
c)	9B31FF1C	d)	1234ABCD
2. Which is of the follow apply. (2 marks)	ving registers always store	es a valu	e that is divisible by 4? Circle all that
a)	\$zero	b)	\$t0
c)	\$sp	d)	\$ra
3. In an R-type instruction	on, how many bits are res	erved fo	or the shift amount? (1 mark)
а) 5	b) 6	
С) 16	d) 26	
4. In an I-type instruction Circle all that apply. (2 r	•	need to	be performed on the immediate value
а) sign extension	ŀ	zero extension
c) ones extension	(d) inversion
5. Which of the following Circle all that apply. (2 r	•	nmetic o	perations will cause an overflow?
а) 1001 + 0100	k	b) 1010 + 1110

c) 1111 + 0001 d) 0100 + 0100

6. If a one-hot decoder has a binary output of 1000000000000, what would the input be? **(2 marks)**

7. What is the maximum distance that a branch instruction can jump (in bytes)? (1 mark) a) 2^{16} **b)** 2^{18} c) 2^{26} d) 2^{28} 8. What is the maximum distance that a jump instruction can jump (in bytes)? (1 mark) a) 2^{16} **b)** 2¹⁸ c) 2^{26} d) 2^{28} 9. What is the output of a tri-state buffer when both inputs are zero? (1 mark) 10. How many bytes can a register unit store, with 6 address bits and 32-bit words? (2 marks) 11. How many bits does a register unit need to address each register, if it stores 1024 bytes in 32-bit integers? (2 marks) **12.** To load the value 1101101100110110 into a 16-bit shift register, it takes X clock cycles. To load the same value into a 16-bit load register, it takes Y clock cycles. What are the values of X and Y? (2 marks) **13.** Which of the following assembly code segments would implement the pseudo-instruction bge \$s, \$t, Label? Circle all that apply. (2 marks) slt \$d, \$t, \$s slt \$d, \$s, \$t beq \$d, \$zero, Label beq \$d, \$zero, Label beq \$s, \$t, Label beq \$s, \$t, Label slt \$d, \$s, \$t slt \$d, \$t, \$s bne \$d, \$zero, Label bne \$d, \$zero, Label

14. Which of the following Verilog statements cannot appear inside an always block? Circle all that apply. (2 marks)

a)
$$a = b$$
;

a)
$$a = b;$$
 b) assign $a = b;$

c)
$$a \le b$$
;

15. In Lab 1, you had to implement a 5-to-1 multiplexer with 2-to-1 multiplexers. What is the minimum number of 2-to-1 multiplexers that you would need to implement a 15-to-1 multiplexer? (1 mark)

16. Which of the following are equivalent to \overline{XY} ? Circle all that apply. (2 marks)

a)
$$(\overline{X} + Y)$$

a)
$$(\overline{\overline{X} + Y})$$
 b) $X(\overline{Y}Z + \overline{Y}\overline{Z}X)$

c)
$$(X + \overline{Y})$$

c)
$$(X + \overline{Y})$$
 d) $X\overline{Y}Z + \overline{X}Y\overline{Z}$

17. The MOSFET is made up of three types of material, listed below. Sort these materials according to the conductivity, from most conductive (1) to least conductive (3). (2 marks)

Metal ____ Oxide ___ Semiconductor ____

18. Other than the proposal, there were 5 parts of Lab 6 that you needed to do. What were those parts? Fill in your answer in the spaces below. (5 marks)

b) _____

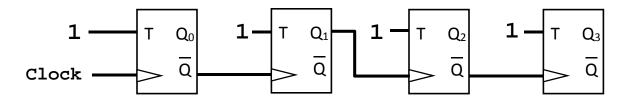
c) _____

d) _____

e) _____

Part B: Design and Analysis (26 marks)

1. Consider the flip-flop diagram below. What values will be on Q_3 , Q_2 , Q_1 , and Q_0 , after each consecutive clock cycle? Write your answer in the spaces provided, one space per clock cycle. **(6 marks)**



 $Q_3 \; Q_2 \; Q_1 \; Q_0$

Initial value	(before	first c	clock	pulse') :

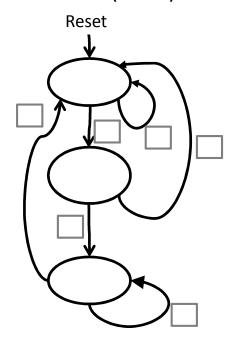
0	0	0	0	

0	0	0	1

2. Consider the Verilog code on the left, which implements a Finite State Machine. Refer to this code when answering the questions on the right. **(14 marks)**

```
module fsm(clock, reset, X, Z);
  input clock, reset, X;
  output reg Z;
  reg [2:0] state, next_state;
  always @ (posedge clock)
     if (reset)
        state <= 1;
     else
        state <= next state;</pre>
  always @ (state)
     case (state)
       2: Z = 1;
       default: Z = 0;
     endcase
  always @ (state, X)
     case (state)
       1: if (!X)
            next_state = 4;
           else
            next state = 1;
       2: if (X)
            next_state = 1;
          else
            next_state = 2;
       4: if (!X)
            next_state = 2;
          else
            next_state = 1;
       default: next_state = 1;
     endcase
endmodule
```

i) Complete the state diagram of this FSM by filling in each state with the correct state number, and labeling each transition with the matching values for X (0 or 1). Ignore any transitions due to reset. (9 marks)



ii) Given the above state transition diagram, fill in the output for each state. **(3 marks)**

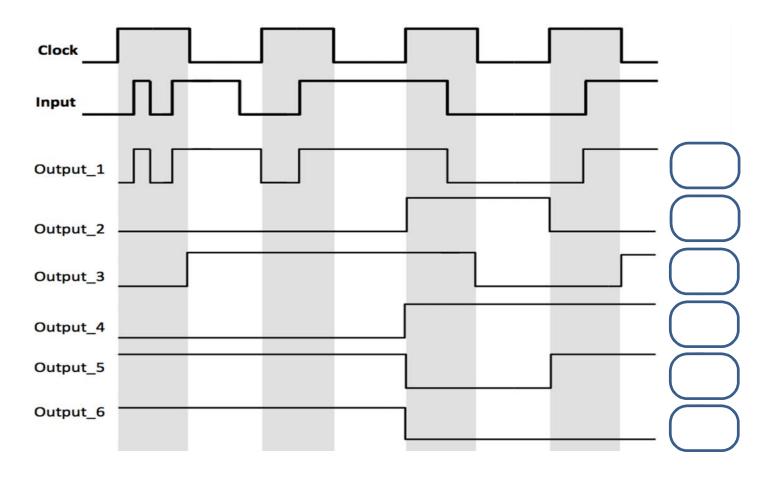
State	Output
1	
2	
4	

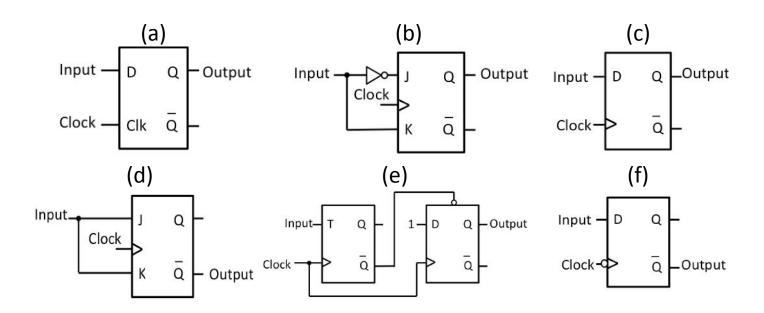
iii) Given the sequence of inputs 0, 1, 0, which will be the resulting FSM state? **(1 mark)**

iv) What type of FSM is this? (1 mark)

a) Mealy b) Moore

3. For every output waveform below, specify in the spaces on the right which sequential circuits at the bottom would generate this output behaviour. If you think there is no match, write "None". Assume functional simulation and that the initial Q state, when unknown, was zero. **(6 marks)**





Part C: Processor Operations (17 marks)

1. When Booth's Algorithm is performed on the binary inputs A=1011 and B=1001, the values for A and P change at each step of the algorithm. The framework is provided below, with a few values filled in for you. Fill in the rest, according to the steps shown in class. **(5 marks)**

Initial Values:

Step #1:

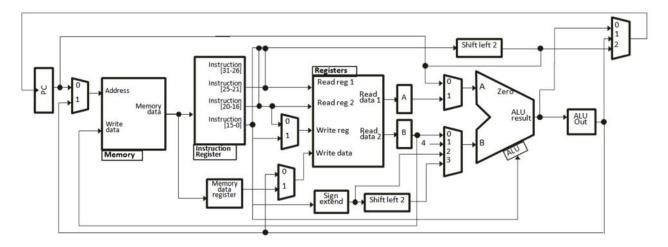
Step #2:

Step #3:

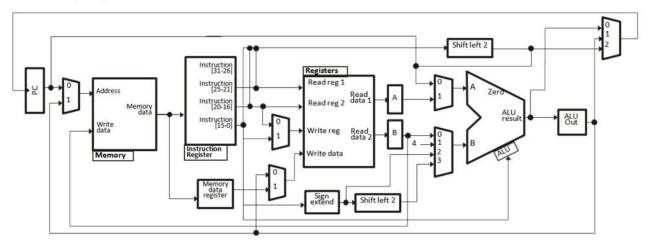
New P value =	

Step #4:

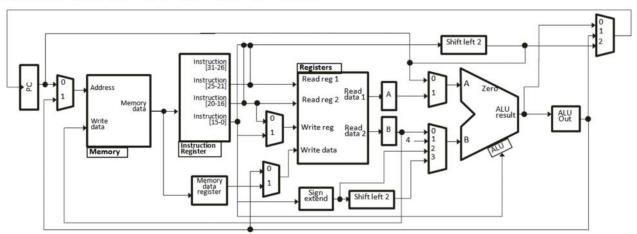
- **2.** Consider the datapaths below. For each of the following operations, highlight the path that the data needs to take, from start to finish. **(12 marks)**
 - a) Store \$ra into the program counter.



b) Store the program counter in \$ra.



c) Push the value stored in \$a0 onto the stack.



Part D: Processor Instructions (27 marks)

1. Each assembly language instructions below matches a machine code instruction at the
bottom. However, the opcodes have been removed from each machine code instruction. Match
the assembly language instructions to their corresponding machine code instructions, and label
each machine code instruction with its missing opcode. (12 marks)

A) or \$	t8, 9	\$t0,	\$t1
----------	-------	-------	------

D)
$$lw $v0, -1($sp)$$

i.	11101000101111111111111111
`	

Assembly code instruction: _____

- **2.** For each of the processor tasks below, indicate what the values of the following control unit signals will be by filling in the boxes next to each signal with the signal values. **(15 marks)**
 - If a control signal doesn't affect the operation, fill in its value with an X.
 - For ALUOp, full marks will only be given for binary values. If you don't know what the values are, just write what kind of operation is taking place instead.

Add 4 to the stack pointer value.								
PCWrite MemToReg ALUSrcA	PCWriteCond IorD MemRead MemWrite IRWrite PCSource ALUOp ALUSrcB RegWrite RegDst							
Add 100 to PCWrite MemToReg ALUSrcA	to the program counter, and store the result in \$ra. PCWriteCond IorD MemRead MemWrite IRWrite PCSource ALUOp ALUSrcB RegWrite RegDst							
Fetch and PCWrite MemToReg ALUSrcA	PCWriteCond IorD MemRead MemWrite IRWrite PCSource ALUOp ALUSrcB RegWrite RegDst							

Part E: Verilog (12 marks)

Consider the piece of Verilog code on the right.

1. In a sentence or less, describe what operation this code performs. **(4 marks)**

2. Based on your answer above, what do the i and O reg values represent? (2 marks)

```
i: _____
```

o: _____

```
module one (o, s, t, d, i, m, w);
input [31:0] m;
input w;
output reg [5:0] o;
output reg [4:0] s, t, d;
output reg [15:0] i;
reg [31:0] c;
always @ (*)
 begin
    if (w) c <= m;
    o <= c[31:26];
    s <= c[25:21];
    t <= c[20:16];
    d <= c[15:11];</pre>
    i <= c[15:0];
  end
endmodule
```

Consider the piece of Verilog code on the right.

3. In one sentence, describe what function this code performs. **(4 marks)**

4. Given your answer to part 1, what functions do input signals e and w perform? **(2 marks)**

```
module two (c, e, w, a, di, do);
 input c, w, e;
 input [7:0] a;
 input [3:0] di;
 output [3:0] do;
 reg [3:0] M [255:0];
       [3:0] do;
 reg
 always @(posedge c)
 begin
   if (e) begin
      if (w)
         M[a] <= di;
      else
         do <= M[a];
   end
 end
endmodule
```

Part F: Assembly Language (24 marks)

1. In the spaces provided below, write the assembly language instruction(s) that perform the following tasks. Full marks will only be given for one-instruction answers. (12 marks total) a) Set the value stored in \$s0 to one eighth of its original value. (3 marks) b) Invert all the bits of the value stored in \$a0. (3 marks)

c) Pop a character off the stack and store it in \$t0. (3 marks)

d) Set the program counter to zero. (3 marks)

2. In the space provided, write the resulting list values for each assembly program. (12 marks)

```
.data
len:
          .word
          .word -4, -1, 0, 1, 4
list:
          .text
          addi $s0, $zero, list
main:
          addi $s1, $zero, len
          lw $t1, 0($s1)
top:
          lw $t0, 0($s0)
          add $t0, $t0, $t0
          sw $t0, 0($s0)
          addi $t1, $t1, -1
          addi $s0, $s0, 4
          bne $t1, $zero, top
end:
          jr $ra
```

```
.data
len:
          .word
          .word -4, -1, 0, 1, 4
list:
          .text
          addi $s0, $zero, list
main:
          addi $s1, $zero, len
          lw $t1, 0($s1)
top:
          lw $t0, 0($s0)
          sub $t0, $t0, $t0
          sw $t0, 0($s0)
          addi $t1, $t1, -1
          addi $s0, $s0, 4
          bne $t0, $zero, top
end:
          jr $ra
```

```
.data
len:
          .word
          .word -4, -1, 0, 1, 4
list:
          .text
          addi $s0, $zero, list
main:
          addi $s1, $zero, len
          lw $t1, 0($s1)
top:
          lw $t0, 0($s0)
          addi $t1, $t1, -1
          addi $s0, $s0, 4
          add $t0, $t0, $t0
          sw $t0, 0($s0)
          bne $t1, $zero, top
end:
          jr $ra
```

```
.data
len:
          .word
          .word -4, -1, 0, 1, 4
list:
          .text
          addi $s0, $zero, list
main:
          addi $s1, $zero, len
          lw $t1, 0($s1)
          lw $t0, 0($s0)
          addi $t1, $t1, -1
top:
          add $t0, $t0, $t1
          bne $t1, $zero, top
          addi $s0, $s0, 4
          sw $t0, 0($s0)
end:
          jr $zero
```

Reference Information

ALU arithmetic input table:

Select		Input	Operation		
S ₁	S ₀	Y	C _{in} =0	C _{in} =1	
0	0	All 0s	G=A	G=A+1	
0	1	В	G=A+B	G=A+B+1	
1	0	В	G=A-B-1	G=A-B	
1	1	All 1s	G=A-1	G=A	

Register assignments:

Register values: Processor role

- Register 0 (\$zero): reserved value.
- Register 1 (\$at): reserved for the assembler.
- Registers 2-3 (\$v0, \$v1): return values
- Registers 4-7 (\$a0-\$a3): function arguments
- Registers 8-15, 24-25 (\$t0-\$t9): temporaries
- Registers 16-23 (\$s0-\$s7): saved temporaries
- Registers 28-31 (\$gp, \$sp, \$fp, \$ra)

Bonus Question: (1 mark)

Draw something in the space below.



Instruction table:

Instruction Type Op/Func Syntax add R 100000 \$d, \$s, \$t addiu I 001000 \$t, \$s, i addiu I 001001 \$t, \$s, i addiu I 001001 \$t, \$s, i addiu R 011010 \$s, \$t divu R 011010 \$s, \$t mult R 011000 \$s, \$t multu R 011001 \$s, \$t sub R 100010 \$d, \$s, \$t sub R 100010 \$d, \$s, \$t sub R 100010 \$d, \$s, \$t subu R 100010 \$d, \$s, \$t and R 100100 \$d, \$s, \$t and R 100100 \$d, \$s, \$t and R 100110 \$t, \$s, i and R 100111 \$d, \$s, \$t or R 100110 \$t, \$s, i and R <th></th> <th></th> <th></th> <th></th>				
addu R 100001 \$d, \$s, \$t addi I 001000 \$t, \$s, i addiu I 001001 \$t, \$s, i addiu I 001001 \$t, \$s, i addiu I 001001 \$t, \$s, i addiu R 011010 \$s, \$t adiv R 011011 \$s, \$t mult R 011000 \$s, \$t and R 100010 \$d, \$s, \$t and R 100100 \$d, \$s, \$t and I 001100 \$t, \$s, i and I 001101 \$d, \$s, \$t or R 100111 \$d, \$s, \$t or R 100111 \$d, \$s, \$t and I 001101 \$t, \$s, i and I 001101 \$t, i and I 00111 \$t, i and I 00111 \$t, i and I 00111 \$t, i and I i 00111 \$t, i and I i 00111 \$t, i and I i i i i and I i i i i and I i i i and I i i i i i i i i i i i i i i i i i i	Instruction	Type	Op/Func	Syntax
addi	add	R	100000	\$d, \$s, \$t
addiu I 001001 \$t, \$s, i div R 011010 \$s, \$t divu R 011011 \$s, \$t mult R 011000 \$s, \$t multu R 01001 \$s, \$t sub R 100010 \$d, \$s, \$t subu R 100011 \$d, \$s, \$t and R 100100 \$d, \$s, \$t andi I 001100 \$t, \$s, i nor R 100111 \$d, \$s, \$t or R 100110 \$t, \$s, i xor R 100010 \$d, \$t, \$s xor R 0001	addu	R	100001	\$d, \$s, \$t
div R 011010 \$5, \$t divu R 011011 \$5, \$t mult R 011000 \$5, \$t multu R 011001 \$5, \$t sub R 100010 \$d, \$5, \$t subu R 100100 \$d, \$5, \$t and R 100100 \$d, \$5, \$t andi I 001100 \$t, \$5, \$t nor R 100111 \$d, \$5, \$t or R 100110 \$t, \$5, \$t sor R 100110 \$t, \$5, \$t sor R 100110 \$t, \$5, \$t sra R 000111 \$d, \$t, \$t sra R 0	addi	I	001000	\$t, \$s, i
divu R 011011 \$5, \$t multu R 011000 \$5, \$t multu R 011001 \$5, \$t sub R 100010 \$d, \$5, \$t subu R 100010 \$d, \$5, \$t and R 100100 \$d, \$5, \$t andi I 001100 \$t, \$5, \$t nor R 100111 \$d, \$5, \$t or R 100101 \$t, \$5, \$t or R 100110 \$t, \$5, \$t or R 100110 \$t, \$5, \$t ori I 00110 \$t, \$5, \$t sori I 00110 \$t, \$t, \$s sra R 00011 \$d, \$t, \$s sra R <td< td=""><td>addiu</td><td>I</td><td>001001</td><td>\$t, \$s, i</td></td<>	addiu	I	001001	\$t, \$s, i
multu R 011000 \$5, \$t multu R 011001 \$5, \$t sub R 100010 \$d, \$s, \$t subu R 100011 \$d, \$s, \$t and R 100100 \$d, \$s, \$t andi I 001100 \$t, \$s, i nor R 100111 \$d, \$s, \$t or R 100101 \$t, \$s, i xor R 100110 \$t, \$s, i xor R 100111 \$t, \$s, i xor R 100111 \$t, \$s, i xor R 100111 \$t, \$s, i xor R	div	R	011010	\$s, \$t
multu R 011001 \$5, \$t sub R 100010 \$d, \$s, \$t subu R 100010 \$d, \$s, \$t and R 100100 \$d, \$s, \$t andi I 001100 \$t, \$s, i nor R 100101 \$d, \$s, \$t or R 100101 \$t, \$s, i ori I 001101 \$t, \$s, i xor R 100110 \$t, \$s, i xor R 100100 \$d, \$t, a xor R 00010 \$d, \$t, a xor R 00011 \$d, \$t, s xor R 00010 \$d, \$t, \$s xr R 00010 \$s, \$t, label beq I 00010 \$s, \$t, label beq I	divu	R	011011	\$s, \$t
sub R 100010 \$d, \$s, \$t subu R 100011 \$d, \$s, \$t and R 100100 \$d, \$s, \$t andi I 001100 \$t, \$s, i nor R 100111 \$d, \$s, \$t or R 100101 \$t, \$s, i xor R 100110 \$t, \$s, i xor R 100111 \$t, \$s, i xor R 100011 \$t, \$s, i xor R 100010 \$t, \$t, \$s xor R	mult	R	011000	\$s, \$t
subu R 100011 \$d, \$s, \$t and R 100100 \$d, \$s, \$t andi I 001100 \$t, \$s, i nor R 100111 \$d, \$s, \$t or R 100101 \$d, \$s, \$t ori I 00110 \$t, \$s, i xor R 100110 \$t, \$s, i sll R 000100 \$d, \$t, a sll R 00011 \$d, \$t, a srav R 00011 \$d, \$t, a srlv R 00011 \$d, \$t, a srlv R 00010 \$d, \$t, a srlv R 00010 \$s, \$t, label bgtz I 00010 \$s, \$t, label bgtz I 00010 \$s, \$t, label jal J	multu	R	011001	\$s, \$t
and R 100100 \$d, \$s, \$t andi I 001100 \$t, \$s, i nor R 100111 \$d, \$s, \$t or R 100101 \$d, \$s, \$t or I 001101 \$t, \$s, i xor R 100110 \$t, \$s, i xor R 100110 \$t, \$s, i xor I 001110 \$t, \$s, i xor I R 000000 \$d, \$t, a sllv R 000100 \$d, \$t, a xor I R 000111 \$d, \$t, a xor I R 000111 \$d, \$t, a xor I R 000110 \$d, \$t, a xor I xor	sub	R	100010	\$d, \$s, \$t
andi	subu	R	100011	\$d, \$s, \$t
nor R 100111 \$d, \$s, \$t or R 100101 \$d, \$s, \$t ori I 001101 \$t, \$s, i xor R 100110 \$t, \$s, i xori I 00110 \$t, \$s, i xori I 00110 \$t, \$s, i sll R 000100 \$d, \$t, a sllv R 00011 \$d, \$t, a srav R 00011 \$d, \$t, s srl R 00011 \$d, \$t, a srav R 00011 \$d, \$t, a srlv R 00011 \$d, \$t, a srlv R 00010 \$d, \$t, s srl R 00010 \$d, \$t, \$s srl R 00010 \$d, \$t, \$s srl R 00010 \$s, \$t, label bgtz I 00010 \$s, \$t, label blez I 00000 \$s jal J 000	and	R	100100	\$d, \$s, \$t
or R 100101 \$d, \$s, \$t ori I 001101 \$t, \$s, i xor R 100110 \$d, \$s, \$t xori I 00110 \$t, \$s, i soll R 000000 \$d, \$t, a sllv R 000011 \$d, \$t, a sra R 000011 \$d, \$t, a srav R 000111 \$d, \$t, a srav R 000110 \$d, \$t, a srlv R 000110 \$s, slabel blez I 000101 \$s, \$t, label jal Joo	andi	I	001100	\$t, \$s, i
ori I 001101 \$t, \$s, i xor R 100110 \$d, \$s, \$t xori I 001110 \$t, \$s, i sll R 000000 \$d, \$t, a sllv R 000101 \$d, \$t, a sra R 00011 \$d, \$t, s srav R 00011 \$d, \$t, \$s srlv R 00010 \$d, \$t, \$s srlv R 00010 \$d, \$t, \$s srlv R 000110 \$d, \$t, \$s srlv R 000110 \$d, \$t, \$s srlv R 000110 \$s, \$t, label bgtz I 000111 \$s, label bgtz I 000101 \$s, \$t, label j J 000010 \$s, \$t, label jal J 000011 \$s jal J 000010 \$s jr R 001000 \$s lb I 1000	nor	R	100111	\$d, \$s, \$t
xor R 100110 \$d, \$s, \$t xori I 001110 \$t, \$s, i sll R 000000 \$d, \$t, a sllv R 000100 \$d, \$t, a sra R 000011 \$d, \$t, a srav R 000111 \$d, \$t, s srl R 000010 \$d, \$t, a srlv R 000110 \$d, \$t, a srlv R 000100 \$d, \$t, a srlv R 000100 \$s, \$t, label bgtz I 000101 \$s, \$t, label bne I 000101 \$s, \$t, label jal J 00001 \$s jr R 001001 \$s jr R 001000 \$s lbu I 1000	or	R	100101	\$d, \$s, \$t
xori I 001110 \$t, \$s, i sll R 000000 \$d, \$t, a sllv R 000100 \$d, \$t, \$s sra R 000011 \$d, \$t, \$s srav R 000110 \$d, \$t, \$s srl R 000110 \$d, \$t, \$s srlv R 000110 \$d, \$t, \$s sp, label beq I 000111 \$s, label bgtz I 000101 \$s, \$t, label jal J 000010 \$s, \$t, label jal J 000011 \$s jr R 001001 \$s lb I 100000 \$t, i (\$s) lb I 100001 \$t, i (\$s) lw I 10001 \$t, i (\$s) sh I 101011<	ori	I	001101	\$t, \$s, i
sll R 000000 \$d, \$t, a sllv R 000100 \$d, \$t, \$s sra R 000011 \$d, \$t, a srav R 000111 \$d, \$t, \$s srl R 000110 \$d, \$t, \$s srlv R 000110 \$d, \$t, \$s splad I 000100 \$s, \$t, label bgtz I 000101 \$s, \$t, label bne I 000101 \$s, \$t, label jal J 000011 \$s jr R 001001 \$s jr R 001001 \$s lb I 100000 \$t, i(\$s) lb I 100001 \$t, i(\$s) lb I 101001 \$t, i(\$s) sh I 101001 \$t, i(\$s)	xor	R	100110	\$d, \$s, \$t
sllv R 000100 \$d, \$t, \$s sra R 000011 \$d, \$t, a srav R 000111 \$d, \$t, \$s srl R 000010 \$d, \$t, a srlv R 000110 \$d, \$t, \$s beq I 000100 \$s, \$t, label bgtz I 000111 \$s, label blez I 000110 \$s, \$t, label blez I 000101 \$s, \$t, label j J 000010 \$s, \$t, label jal J 000010 \$s, \$t, label jal J 000010 \$s, \$t, label jal J 000011 \$s, \$t, label jal J 000011 \$s jr R 001001 \$s lb I 100000 \$t, i(\$s) lbu I 100001 \$t, i(\$s) lw I 100001 \$t, i(\$s) sb I 101001 \$t, i(\$s) sw I 101011 \$t,	xori	I	001110	\$t, \$s, i
sra R 000011 \$d, \$t, a srav R 000111 \$d, \$t, \$s srl R 000010 \$d, \$t, a srlv R 000110 \$d, \$t, \$s beq I 000100 \$s, \$t, label bgtz I 000110 \$s, label blez I 000101 \$s, \$t, label bne I 000101 \$s, \$t, label jal J 000010 label jalr R 001001 \$s jr R 001001 \$s jr R 001000 \$s lb I 100000 \$t, i(\$s) lbu I 100000 \$t, i(\$s) lw I 100001 \$t, i(\$s) lw I 101001 \$t, i(\$s) sh I 101001 \$t, i(\$s) sw I 101011 \$t, i(\$s) trap I 001100 i	sll	R	000000	\$d, \$t, a
srav R 000111 \$d, \$t, \$s srl R 000010 \$d, \$t, \$s srlv R 000110 \$d, \$t, \$s beq I 000100 \$s, \$t, label bgtz I 000111 \$s, label blez I 000101 \$s, \$t, label bne I 000101 \$s, \$t, label jal J 000010 label jal J 000011 label jal J 000011 \$s jr R 001001 \$s jr R 001000 \$s lb I 100000 \$t, i(\$s) lbu I 100001 \$t, i(\$s) lw I 100011 \$t, i(\$s) sb I 101001 \$t, i(\$s) sw I 101011 \$t, i(\$s) trap I 001100 i	sllv	R	000100	\$d, \$t, \$s
srl R 000010 \$d, \$t, \$s beq I 000100 \$d, \$t, \$s beq I 000100 \$s, \$t, label bgtz I 000111 \$s, label blez I 000110 \$s, \$t, label bne I 000101 \$s, \$t, label j J 000010 label jal J 000011 label jal R 001001 \$s jr R 001001 \$s jr R 001000 \$s lb I 100000 \$t, i(\$s) lb I 100001 \$t, i(\$s) lh I 100011 \$t, i(\$s) lw I 101001 \$t, i(\$s) sh I 101011 \$t, i(\$s) sw I 101011 \$t, i(\$s) trap I 001100 i	sra	R	000011	\$d, \$t, a
srlv R 000110 \$d, \$t, \$s beq I 000100 \$s, \$t, label bgtz I 000111 \$s, label blez I 000110 \$s, \$t, label bne I 000101 \$s, \$t, label j J 000010 label jal J 000011 label jalr R 001001 \$s jr R 001000 \$s lb I 100000 \$t, i(\$s) lbu I 100100 \$t, i(\$s) lhu I 100101 \$t, i(\$s) lw I 101001 \$t, i(\$s) sb I 101001 \$t, i(\$s) sw I 101011 \$t, i(\$s) trap I 001100 i	srav	R	000111	\$d, \$t, \$s
beq I 000100 \$\$, \$\$t, label bgtz I 000111 \$\$, label blez I 000110 \$\$, label bne I 000101 \$\$\$, \$\$t, label j J 000010 label jal J 000011 label jalr R 001001 \$\$\$ jr R 001000 \$\$\$ lb I 100000 \$\$\$ lb I 100100 \$\$\$t, i (\$\$\$) lh I 100001 \$\$\$t, i (\$\$\$) lw I 100001 \$\$\$t, i (\$\$\$) sb I 101001 \$\$\$t, i (\$\$\$) sh I 101001 \$\$\$t, i (\$\$\$) trap I 001100 i	srl	R	000010	\$d, \$t, a
bgtz I 000111 \$\$, label blez I 000110 \$\$, label bne I 000101 \$\$\$, \$\$\$, label j J 000010 label jal J 000011 label jalr R 001001 \$\$\$ jr R 001000 \$\$\$ lb I 100000 \$\$\$ lb I 100100 \$\$\$ lbu I 100100 \$\$\$ lhu I 100101 \$\$\$ lw I 101001 \$\$\$ sb I 101001 \$\$\$ sh I 101011 \$\$\$ sw I 10101 \$\$\$ trap I 001100 i	srlv	R	000110	\$d, \$t, \$s
blez I 000110 \$\$, label bne I 000101 \$\$, \$\$t, label j J 000010 label jal J 000011 label jalr R 001001 \$\$\$ jr R 001000 \$\$\$ lb I 100000 \$\$\$\$t, i (\$\$\$) lbu I 100100 \$\$\$\$\$t, i (\$\$\$) lhu I 100001 \$\$\$\$\$t, i (\$\$\$) lw I 101001 \$\$\$\$\$\$t, i (\$\$\$) sb I 101001 \$\$\$\$\$t, i (\$\$\$) sh I 101011 \$\$\$\$\$t, i (\$\$\$) trap I 001100 i	beq	I	000100	\$s, \$t, label
bne I 000101 \$5, \$t, label j J 000010 label jal J 000011 label jalr R 001001 \$5 jr R 001000 \$5 lb I 100000 \$t, i(\$\$\$) lbu I 100100 \$t, i(\$\$\$) lh I 100001 \$t, i(\$\$\$) lw I 100001 \$t, i(\$\$\$) sb I 101000 \$t, i(\$\$\$) sh I 101011 \$t, i(\$\$\$) sw I 101011 \$t, i(\$\$\$) trap I 001100 i	bgtz	I	000111	\$s, label
j J 000010 label jal J 000011 label jalr R 001001 \$\$ jr R 001000 \$\$ lb I 100000 \$\$t, i (\$\$\$) lbu I 100100 \$\$t, i (\$\$\$) lhu I 100001 \$\$t, i (\$\$\$) lw I 101000 \$\$t, i (\$\$\$) sb I 101001 \$\$t, i (\$\$\$) sh I 101011 \$\$t, i (\$\$\$) trap I 001100 i	blez	I	000110	\$s, label
jal J 000011 label jalr R 001001 \$\$ jr R 001000 \$\$\$ lb I 100000 \$\$\$\$t, i(\$\$\$) lbu I 100100 \$	bne	I	000101	\$s, \$t, label
jalr R 001001 \$S jr R 001000 \$S lb I 100000 \$t,i(\$S) lbu I 100100 \$t,i(\$S) lh I 100001 \$t,i(\$S) lw I 100011 \$t,i(\$S) sb I 101000 \$t,i(\$S) sh I 101011 \$t,i(\$S) sw I 101011 \$t,i(\$S) trap I 001100 i	j	J	000010	label
jr R 001000 \$5 lb I 100000 \$t, i(\$\$\$) lbu I 100100 \$t, i(\$\$\$) lh I 100001 \$t, i(\$\$\$) lw I 100011 \$t, i(\$\$\$) sb I 101000 \$t, i(\$\$\$) sh I 101001 \$t, i(\$\$\$) sw I 101011 \$t, i(\$\$\$) trap I 001100 i	jal	J	000011	label
1b I 100000 \$t, i (\$s) 1bu I 100100 \$t, i (\$s) 1h I 100001 \$t, i (\$s) 1hu I 100101 \$t, i (\$s) 1w I 100011 \$t, i (\$s) sb I 101000 \$t, i (\$s) sh I 101011 \$t, i (\$s) sw I 101011 \$t, i (\$s) trap I 001100 i	jalr	R	001001	\$ S
lbu I 100100 \$t, i (\$s) lh I 100001 \$t, i (\$s) lhu I 100101 \$t, i (\$s) lw I 100011 \$t, i (\$s) sb I 101000 \$t, i (\$s) sh I 101011 \$t, i (\$s) sw I 101011 \$t, i (\$s) trap I 001100 i	jr	R	001000	\$ S
1h I 100001 \$t, i (\$s) 1hu I 100101 \$t, i (\$s) 1w I 100011 \$t, i (\$s) sb I 101000 \$t, i (\$s) sh I 101011 \$t, i (\$s) sw I 101011 \$t, i (\$s) trap I 001100 i	lb	ı	100000	\$t, i (\$s)
lhu I 100101 \$t,i(\$s) lw I 100011 \$t,i(\$s) sb I 101000 \$t,i(\$s) sh I 101001 \$t,i(\$s) sw I 101011 \$t,i(\$s) trap I 001100 i	lbu	I	100100	\$t, i (\$s)
lw l 100011 \$t, i (\$s) sb l 101000 \$t, i (\$s) sh l 101001 \$t, i (\$s) sw l 101011 \$t, i (\$s) trap l 001100 i	lh	I	100001	\$t, i (\$s)
sb I 101000 \$t, i (\$s) sh I 101001 \$t, i (\$s) sw I 101011 \$t, i (\$s) trap I 001100 i	lhu	I	100101	\$t, i (\$s)
sh I 101001 \$t, i (\$s) sw I 101011 \$t, i (\$s) trap I 001100 i	lw	I	100011	\$t, i (\$s)
sw I 101011 \$t, i(\$s) trap I 001100 i	sb	I	101000	\$t, i (\$s)
trap	sh	I	101001	\$t, i (\$s)
	sw	I	101011	\$t, i (\$s)
mflo R 010010 \$d	trap	I	001100	i
	mflo	R	010010	\$d

 ${\it This page is left blank intentionally for answer overflows.}$

Total Marks = 140

Total Pages = 16

16