UNIVERSITY OF TORONTO

Faculty of Arts and Science

April 2012 Examinations

CSC258H1S: Computer Organization

Duration: 3 hours

No Aids Allowed

Last Name: _____

First Name: _____

Student Number: _____

Instructor: **Steve Engels**

Instructions:

- **Write your name on every page of this exam.**
- **Do not open this exam until you hear the signal to start.**
- Have your student ID on your desk.
- No aids permitted other than writing tools. Keep all bags and notes far from your desk before the exam begins.
- There are 4 questions on 16 pages. When you hear the signal to start, make sure that your exam is complete before you begin.
- Read over the entire exam before starting.
- If you use any space for rough work or have to user the overflow page, clearly indicate the section(s) that you want marked.

Mark Breakdown

Part A:	**/ 30**
Part B:	**/ 58**
Part C:	**/ 24**
Part D:	**/ 20**
Total:	**/ 132**

Part A: Short Answer (23 marks)

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

1. A memory unit stores 2^{16} bytes in 32-bit words. How many address bits are necessary in order to retrieve a single word from this memory unit? **(2 marks)**

a) 11 **b)** 12

c) 14 **d)** 16

2. True or False? The stack pointer gets smaller as the stack gets bigger. **(1 mark)**

True **False**

3. What is MOSFET short for? **(2 marks)**

4. True or False? A ripple counter is asynchronous because its values can change at any time. **(1 mark)**

True **False**

5. What are the HI and LO registers used for in the context of integer division? **(1 mark)**

6. What Verilog operator assigns the output of one gate to the input of another? **(1 mark)**

7. When does majority logic set an output signal high? **(1 mark)**

8. p-type semiconductor materials are said to have an abundance of _____. **(1 mark)**

9. If the binary value 1010 is shifted right arithmetically by one bit, what is the result? **(1 mark)**

10. When a jump-and-link instruction is called, what gets stored in $ra? **(1 mark)**

11. How can you test if a register is storing a valid instruction address? **(1 mark)**

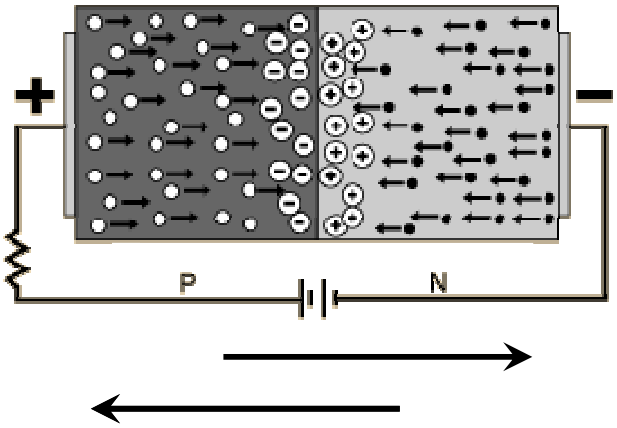
12. What do the following output signals from the ALU signify? **(4 marks)**

C: _____ **V: _____**

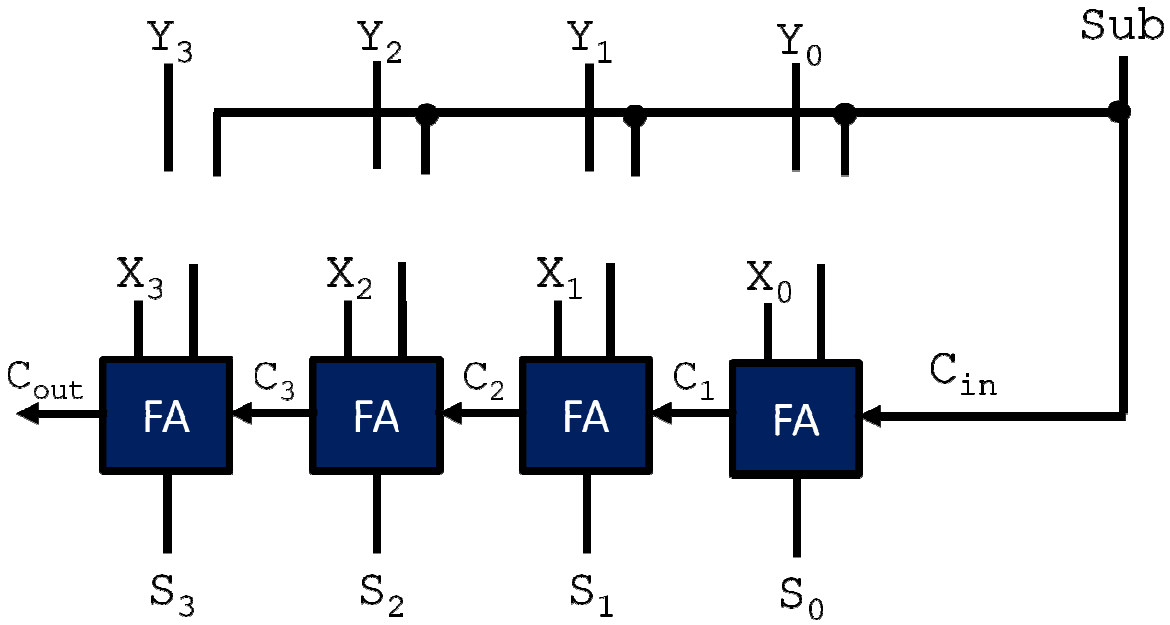
N: _____ **Z: _____**

13. Give an example of when the **V** signal might go high during an ALU operation. **(1 mark)**

14. Consider the pn-junction below. What are the names of the natural flows of current to the left and to the right that exist independent of any external bias? Label the arrows on the diagram below. **(2 marks)**



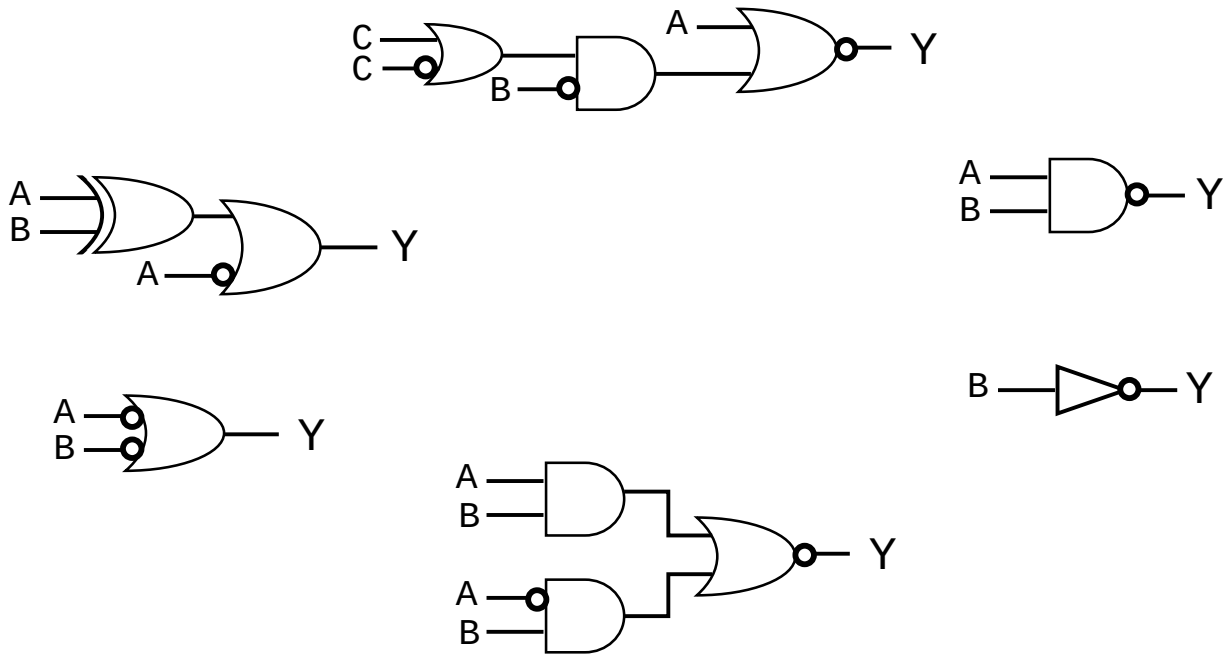
15. Draw gates into the following diagram to implement a subtractor circuit. **(2 marks)**



es the value in $t0 by

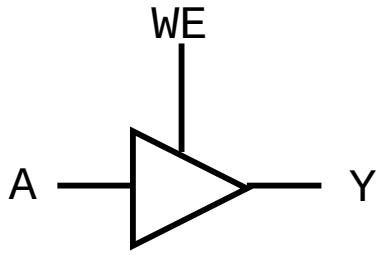
(continued)

17. Which of the circuits below have equivalent behaviour? Draw lines that connect any circuits that match. **(3 marks)**



18. Consider the tri-state buffer unit on the right. In the table below, fill in the output values for this buffer. **(3 marks)**

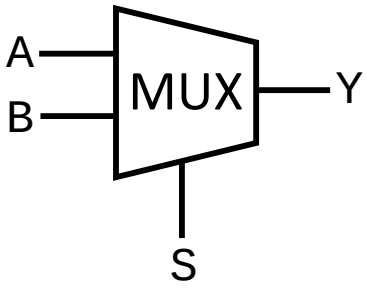
WE	A	Y



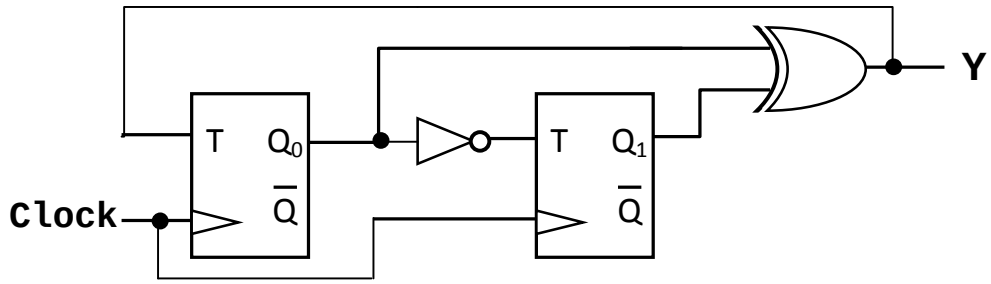
19. What do the letters in LFSR stand for? **(1 mark)**

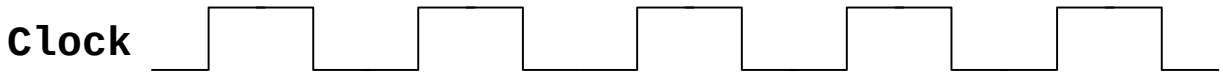
Part B: Design and Analysis (45 marks)

1. Consider the multiplexer device on the right. In the space below, draw a diagram that implements a **half adder** as a combination of NOT gates and 2-to-1 multiplexers. **(4 marks)**



2. Given the following circuit, show what the output value of Q_0, Q_1 and Y will be in the waveform diagram. Assume that Q_0 and Q_1 start with initial values of zero. **(6 marks)**



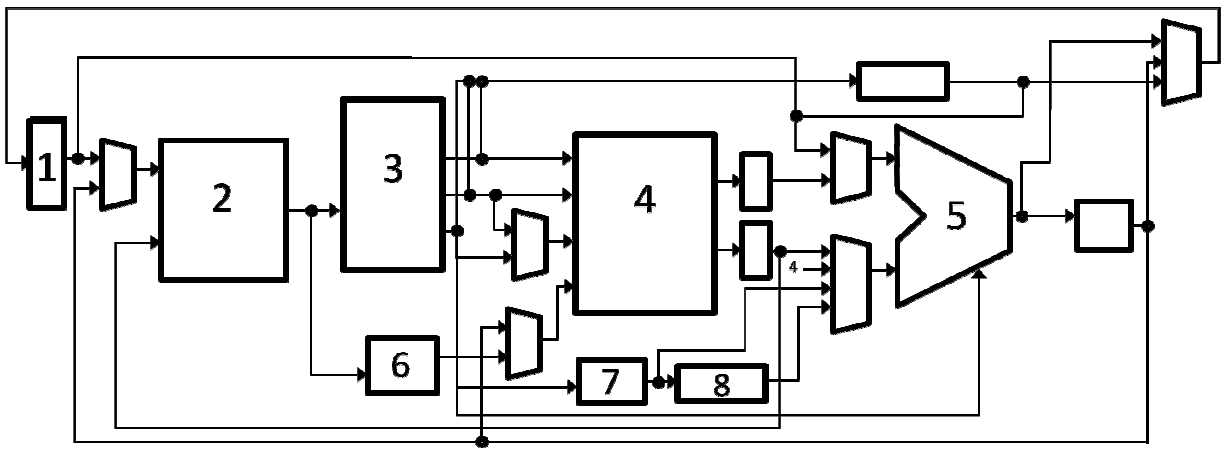


Q_0

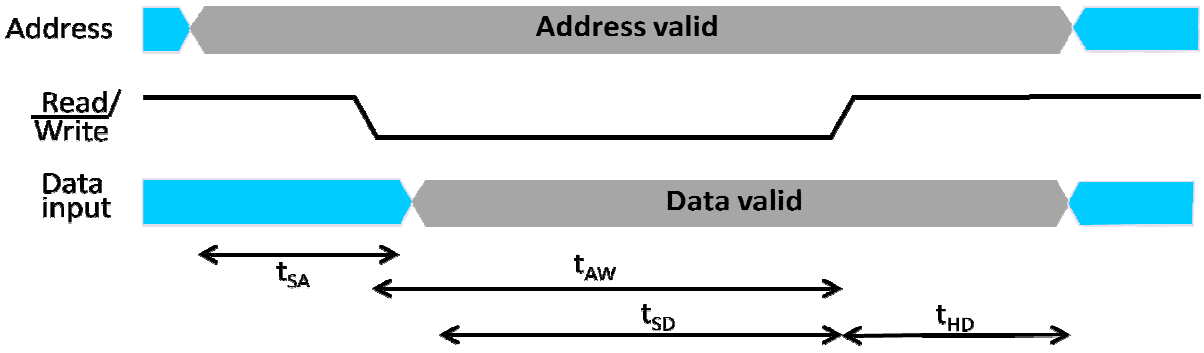
Q_1

Y

ıponents that are



4. For the following diagram, describe what each labeled time segment is called, and what it signifies during a memory write operation. **(8 marks)**



(continued)

5. In the space below, perform Booth's Algorithm on the binary values A=10101 and B=01010. Show your steps in the space provided. **(6 marks)**

P =

Step 1:

Step 2:

Step 3:

Step 4:

Step 5:

6. Verify your answer above by writing the decimal values for A, B and the final product in the spaces below. **(2 marks)**

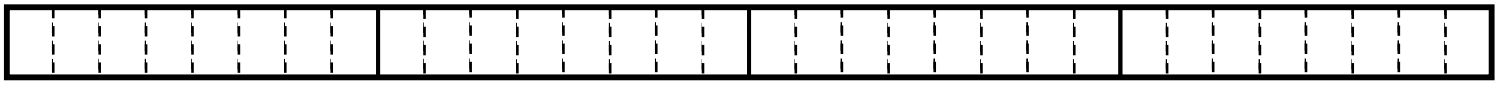
A = ☐

B = ☐

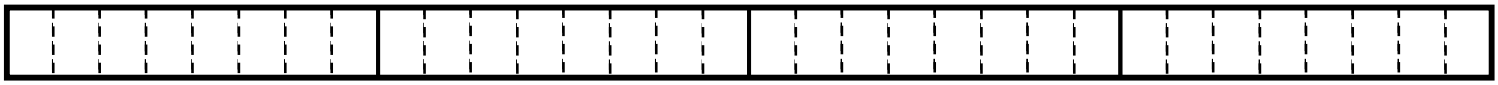
Product = ☐

7. For the following assembly language instructions, write the equivalent machine code instruction in the space provided. You might find the reference information in the appendix helpful for this question. **(6 marks)**

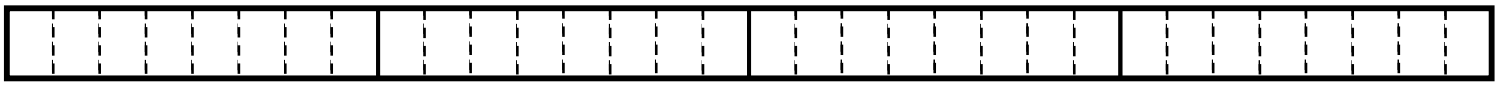
a) `sub $t0, $s0, $s1`



b) `bne $t6, $t9, loop   (where loop occurs 5 instructions earlier)`



c) `sb $t0, 12($s0)`



8. For the following machine code instructions, provide the equivalent assembly language instruction in the space provided. **(6 marks)**

a) `00100011111000000000000000000000`

b) `00000001010010010100000000100110`

c) `10001100100100000000000000000100`

9. For each of the processor instructions below, indicate which control unit signals will be on and off by filling in the boxes next to each signal with the signal values. **(12 marks)**

- If a control signal doesn't affect the operation, fill in its value with an X.
- For ALUOp, just write what kind of operation is taking place.

Increment the program counter to the address of the next instruction.

PCWrite ☐ PCWriteCond ☐ IorD ☐ MemRead ☐ MemWrite ☐
MemToReg ☐ IRWrite ☐ PCSource ☐ ALUOp ☐
ALUSrcA ☐ ALUSrcB ☐ RegWrite ☐ RegDst ☐

Ask memory for a data value whose address is stored in $a0.

PCWrite ☐ PCWriteCond ☐ IorD ☐ MemRead ☐ MemWrite ☐
MemToReg ☐ IRWrite ☐ PCSource ☐ ALUOp ☐
ALUSrcA ☐ ALUSrcB ☐ RegWrite ☐ RegDst ☐

Store the data value that was fetched from memory into register $t0.

PCWrite ☐ PCWriteCond ☐ IorD ☐ MemRead ☐ MemWrite ☐
MemToReg ☐ IRWrite ☐ PCSource ☐ ALUOp ☐
ALUSrcA ☐ ALUSrcB ☐ RegWrite ☐ RegDst ☐

Part C: Verilog (24 marks)

Consider the piece of Verilog code on the right.

```
module foo (out, enable, clk, reset);
  output [7:0] out;
  input enable, clk, reset;
  reg [7:0] out;
  always @ (posedge clk)
    if (reset) begin
      out <= 8'b0000_0001;
    end
    else if (enable) begin
      out <= {out[6],out[5],out[4],
              out[3],out[2],out[1],
              out[0],out[7]};
    end
endmodule
```

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

2. Under what condition(s) would `out[0]` be the only high output signal? **(3 marks)**

3. In the space below, write a Verilog module called `tff` that takes in input signals `data`, `clock` and `reset`, and has one output signal `q`. **(5 marks total)**

- Make the output `q` behave like a T flip-flop on the positive edge of the clock **(3 marks)**
- Implement an asynchronous reset that is negative-edge triggered. **(2 marks)**

4. Barcodes encode product numbers as a sequence of 1s and 0s. Certain barcodes start and end with the sequence `101`, to denote the beginning and end of the sequence.

In the space below, draw a FSM diagram that has a single input called IN and a single output called OUT. The FSM will set OUT high when the three most recent values of IN have been `101`. Keep the states to a minimum, and label your start state. **(5 marks)**

5. In the space below, write an `always` block in Verilog that implements the state transition component of your FSM. Ensure that the state names in your code are the same as in your diagram, and assume the input signal is still called IN. **(8 marks)**

Part D: Assembly Language (20 marks)

Consider the assembly language program in the box below.

```
          .data
list:     .word     3, 0, 1, 2, 6, -2, 4, 7, 3, 7
          .text
main:     addi $s0, $zero, list    #
          add $t0, $zero, $zero    #
          addi $t9, $zero, 10      #
          lw $t1, 0($s0)           #
top:      addi, $t0, $t0, 4        #
          beq $t0, $t9, end        #
          add $t2, $s0, $t0        #
          lw $t3, 0($t2)           #
          sub $t4, $t1, $t3        #
          bgtz $t4, top            #
          add $t1, $t3, $zero      #
          j top                    #
end:      sw $t1, 0($sp)           #
          addi $sp, $sp, -4        #
          jr $ra                   #
```

1. For each line in the code, provide a short descriptive comment on the right **(8 marks)**

2. In the space below, provide a one-sentence description of the overall task this code is performing. **(2 marks)**

3. In the space below, write a short assembly language program that is a translation of the program on the right. You can assume that `i` has been placed on the top of the stack, and that the return value should be placed on the stack as well before returning to the calling program. Make sure that you comment your code so that we understand what you're doing. **(10 marks)**

```
int sign (int i) {
   if (i > 0)
      return 1;
   else if (i < 0)
      return -1;
   else
      return 0;
```

Reference Information

Register values:

- Register 0 ($zero): value 0.
- Register 1 ($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)

Instruction	Op/Func	Syntax
add	100000	$d, $s, $t
addu	100001	$d, $s, $t
addi	001000	$t, $s, i
addiu	001001	$t, $s, i
div	011010	$s, $t
divu	011011	$s, $t
mult	011000	$s, $t
multu	011001	$s, $t
sub	100010	$d, $s, $t
subu	100011	$d, $s, $t
and	100100	$d, $s, $t
andi	001100	$t, $s, i
nor	100111	$d, $s, $t
or	100101	$d, $s, $t
ori	001101	$t, $s, i
xor	100110	$d, $s, $t
xori	001110	$t, $s, i
sll	000000	$d, $t, a
sllv	000100	$d, $t, $s
sra	000011	$d, $t, a
srav	000111	$d, $t, $s
srl	000010	$d, $t, a
srlv	000110	$d, $t, $s
beq	000100	$s, $t, label
bgtz	000111	$s, label
blez	000110	$s, label
bne	000101	$s, $t, label
j	000010	label
jal	000011	label
jalr	001001	$s
jr	001000	$s
lb	100000	$t, i ($s)
lbu	100100	$t, i ($s)
lh	100001	$t, i ($s)
lhu	100101	$t, i ($s)
lw	100011	$t, i ($s)
sb	101000	$t, i ($s)
sh	101001	$t, i ($s)
sw	101011	$t, i ($s)
trap	011010	i
mflo	010010	$d

This page is left blank intentionally for answer overflows.

Total Marks = 132

Total Pages = 16

End of exam