

University of Waterloo  
CS 251 Midterm Exam  
Fall 2011

---

CS 251: Computer Organization & Design  
Instructor: Elodie Fourquet and Stephen Mann

---

October 25, 2011

Start time: 4:30 PM. End Time: 6:20 PM.

Duration: 110 minutes

Number of exam pages (including this cover sheet): 12  
CLOSED BOOK

---

Signature: \_\_\_\_\_

#	Points Possible	Score	Comments
1	8		
2	7		
3	6		
4	9		
5	6		
6	5		
7	6		
8	8		
Total	55		
Mark	100		

Directions

The instructor reserves the right to adjust the value of individual problems if they are deemed too hard or too easy.

If you need extra paper for a question, use the back of that question's page. If you need more paper beyond this, ask the proctors for blank pages. IT IS YOUR RESPONSIBILITY TO ENSURE THAT ANY EXTRA PAGES ARE STAPLED TO THIS EXAM.

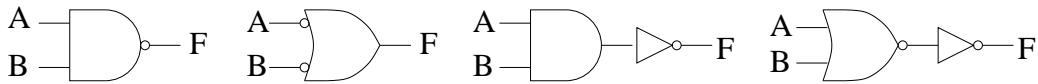
Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

1. (8 Points) Short Answer

- (a) (2 pts) Give the *unreduced, minterm* expression for F based on the following truth table.

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

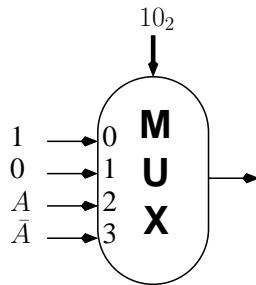
- (b) (2 pts) Circle the gate(s)/circuit(s) that are NOT equivalent to a NAND gate.



- (c) (1 pt) How many bits of information can one flip-flop store? Circle one.

- (a) 0                    (b) 1                    (c) 4                    (d) 8                    (e) 16                    (f) 32

- (d) (1 pt) Label the value of the output of the following circuit:



- (e) (2 pts) Draw a circuit for  $F = A\bar{B} + BC + A\bar{C}$  using only NAND gates.

Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

2. (7 Points)

In this question you are analyzing the following function  $F$

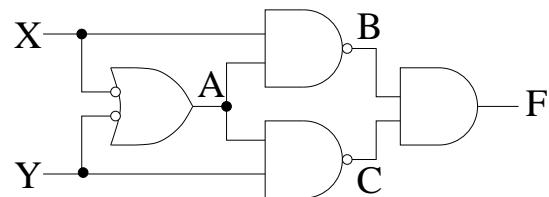
$$F = \overline{A\bar{B}} + \overline{A} + \overline{C}$$

- (a) (4 pts) Give the truth table for  $F$  and deduce the minimal expression for  $F$ .

A	B	C	F
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

$$F =$$

- (b) (3 pts) Derive the truth table for the following circuit. Fill in for each label the intermediate gate outputs.



X	Y	A	B	C	F
0	0				
0	1				
1	0				
1	1				

Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

3. (6 Points)

- (a) (1 pt) Negate the following number, expressed in 8-bit 2's complement representation.

0101 1010

- (b) (2 pts) Compute the following bitwise logical operations.

$$\begin{array}{r} 0101\ 1110 \\ \text{XOR}\ 0000\ 1111 \\ \hline \end{array} \qquad \begin{array}{r} 1100\ 1110 \\ \text{OR}\ 0000\ 1111 \\ \hline \end{array}$$

- (c) (3 pts) Using 8-bit 2's complement **addition**, perform the following computations. You should take the 2's complement of any of the operands as needed. For each computation, circle if an overflow occurred or not. Show your work.

$$\begin{array}{r} 1101\ 0111 \\ +1011\ 1101 \\ \hline \end{array} \qquad \begin{array}{r} 1010\ 0111 \\ -0101\ 1010 \\ \hline \end{array}$$

Overflow: YES \ NO                                    YES \ NO

Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

4. (9 Points)

- (a) (2 pts) Sign extend the following 4-bit 2's complement number to an 8-bit 2's complement number, and give the signed decimal value of the extended number.

Given number : 1011

Sign extension : Signed Decimal :

- (b) (2 pts) Draw the hardware to realize sign extension from a 4-bit 2's complement number,  $B_3B_2B_1B_0$ , to a 8-bit one,  $C_7C_6C_5C_4C_3C_2C_1C_0$ . Fill in how many gates are needed for this hardware.

Total Number of gates:

$B_3$  —

$B_2$  —

$B_1$  —

$B_0$  —

- (c) (3 pts) Write the following number  $-10.01_2 \times 2^3$  (the significant is given in binary form, while the exponent is in decimal) as a 32-bit, IEEE normalized floating point number with biased exponent. Show your work.

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

- (d) (2 pts) Circle the normalized binary floating point number that represents the value 5.625 (a decimal floating point).

- (a)  $1.0001 \times 2^2$       (b)  $1.010001 \times 2^2$       (c)  $101.101 \times 2^0$   
(d)  $0.101011 \times 2^2$       (e)  $1.010001 \times 2^0$       (f)  $1.01101 \times 2^2$   
(g)  $101.0001 \times 2^0$       (h)  $1.01011 \times 2^2$       (i)  $1.101011 \times 2^0$

Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

5. (6 Points) A *full subtractor* is a combinational circuit which operates on three binary input digits,  $a$ ,  $c$  and  $b_{in}$ . It computes as outputs the resulting difference,  $r = a - c - b_{in}$  and a new borrow digit,  $b_{out}$ .

The truth table for a full subtractor is as follows:

$a$	$c$	$b_{in}$	$b_{out}$	$r$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

- (a) (2 pts) Give a sum of minterms expression for  $b_{out}$  and  $r$ . Do **not** reduce this expression.

$$b_{out} =$$

$$r =$$

- (b) (1 pt) The expression for  $r$  can be reduced significantly. Give a reduced form for  $r$ .

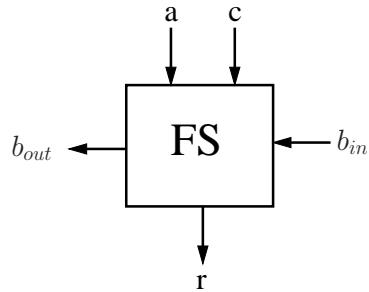
$$r =$$

(continued on next page)

Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

- (c) (3 pts) Connect a number of full subtractors to make a 4-bit subtractor unit that subtracts two 4-bit numbers  $a_3a_2a_1a_0$  and  $c_3c_2c_1c_0$ , computing both the result  $r_3r_2r_1r_0$  and its borrow out  $b_{out}$ . To build your 4-bit subtractor, use the following symbol FS as the full-subtractor unit that performs 1-bit subtraction for binary digits  $a$  and  $c$  with borrow in,  $b_{in}$ .

Be sure to label all inputs and outputs.



Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

6. (5 Points) Consider the following next-state table and corresponding output table:

$S_1$	$S_0$	$A$	$S'_1$	$S'_0$		$S_1$	$S_0$	$B$
0	0	0	1	0		0	0	0
0	0	1	1	1		0	1	X
0	1	X	X	X		1	0	1
1	0	0	1	1		1	1	1
1	0	1	1	0				
1	1	0	0	0				
1	1	1	0	0				

(a) (3 pts) Draw the finite state machine for this next state and output table pair.

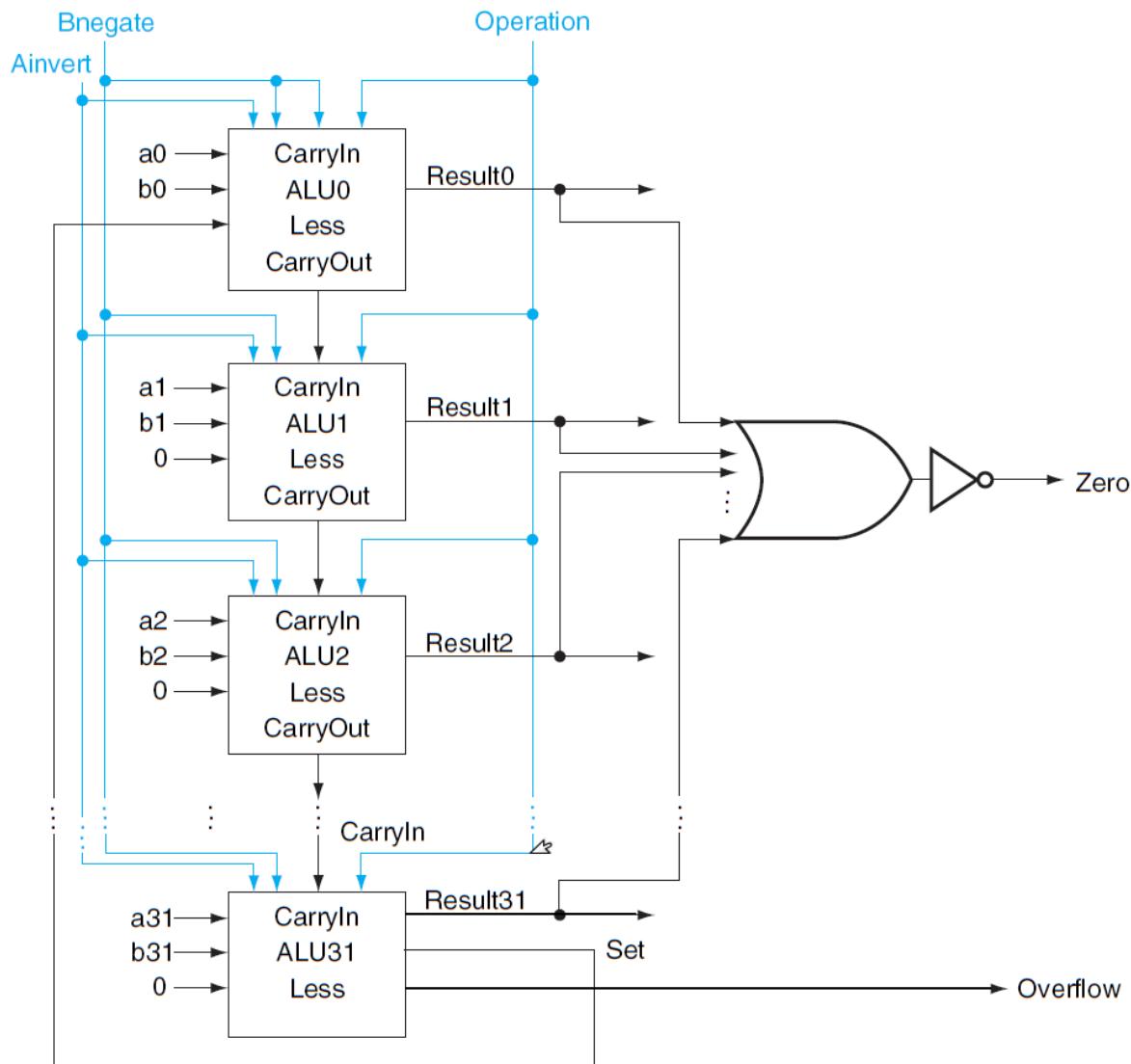
(b) (2 pts) Complete the table below tracing the above finite state machine on a particular sequence of input bits  $A$ . Note that the Next State of one column is the Current State for the next column.

Current State	00						
$A$	0	1	0	1	1	0	
$B$							
Next State							

Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

7. (6 Points) Below is the diagram for a 32-bit ALU. Modify this diagram to have the following three additional outputs. In the figure, label each output with the name given in bold in the question.

- (a) (2 pts) **Positive**: This should be HIGH (1) when the output is strictly greater than 0.
- (b) (2 pts) **Negative Even**: This should be HIGH (1) when the output is a negative even number.
- (c) (2 pts) **Multiple-of-4**: This should be HIGH (1) when the output is a multiple of 4.



Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

8. (8 Points) Consider the following assembly language instruction

5000: lw \$1,200(\$2)

You will find MIPS instruction formats on the last page of this exam.

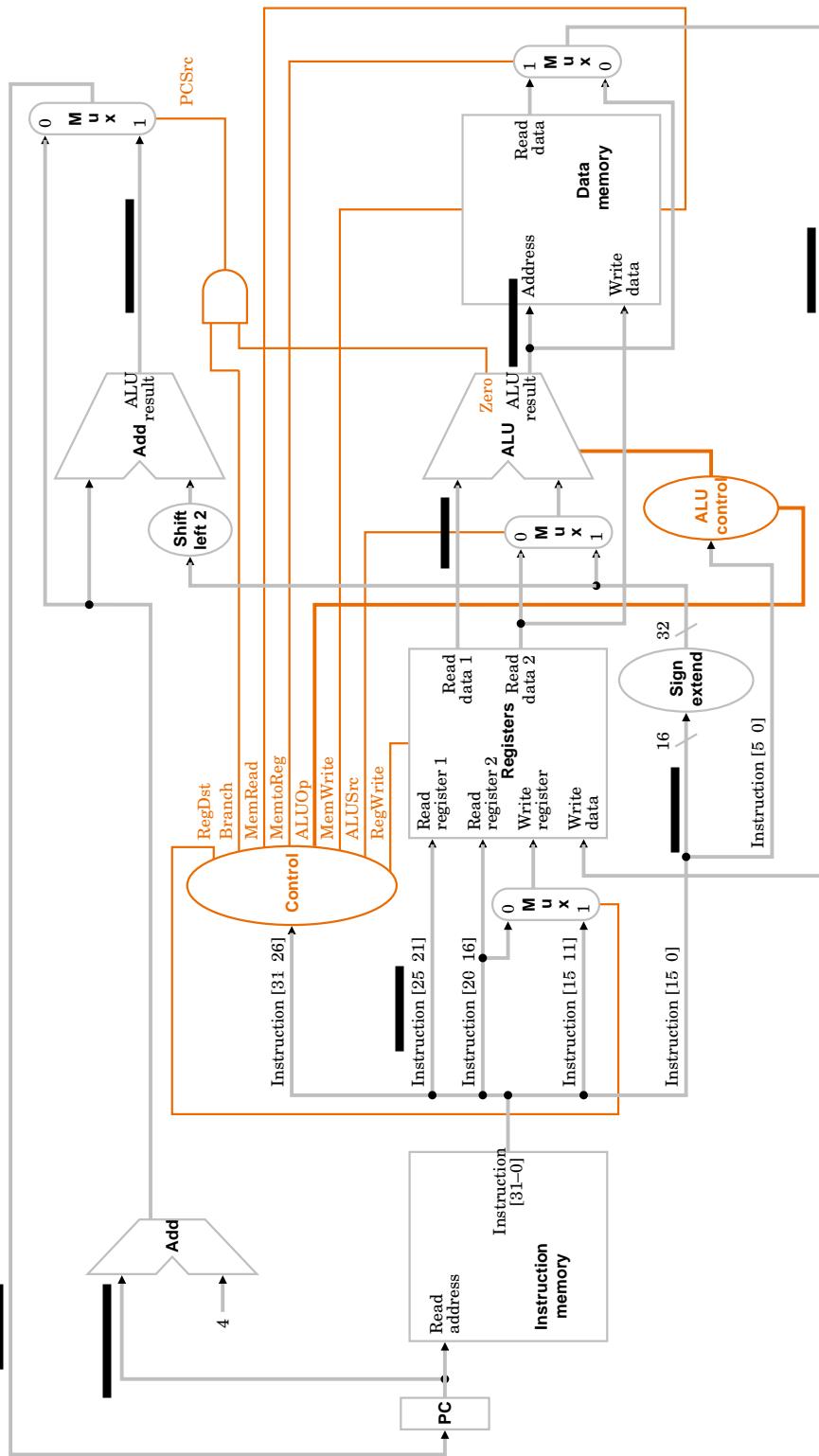
In the figure on the next page, there are **eight** dark lines. On each line, write in the value that travels along the corresponding wires when executing this assembly language instruction. Note that all the marked wires are horizontal.

Assume that each register  $\$i$  (with  $i > 0$ ) contains the decimal value  $30 + i$ . In the data memory each address,  $a$ , contains the decimal value  $40 + a$ .

Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

(Question 8, continued)

**5000: lw \$1, 200(\$2)**



Name: \_\_\_\_\_ Student Number: \_\_\_\_\_

## MIPS and architecture details

(You may remove this page from your exam.)

- Instruction formats

R-format: add rd,rs,rt

31	26 25	21 20	16 15	11 10	6 5	0
0	rs	rt	rd	shamt	funct	

6 bits      5 bits      5 bits      5 bits      5 bits      6 bits

Load/store: lw rt,50(rs)

31	26 25	21 20	16 15	0
35/43	rs	rt	offset	

6 bits      5 bits      5 bits      16 bits

Jump: j 3000

31	26 25	0
4	address	

6 bits      26 bits

Instruction	Action
add rd,rs,rt	$R[rd] = R[rs] + R[rt]$
lw rt, 50(rs)	$R[rt] = M[rs + se\ 50]$
sw rt, 50(rs)	$M[rs + se\ 50] = R[rt]$
beq rs,rt,50	if( $R[rs] == R[rt]$ ) $PC = PC + 4 + se\ 50 \times 4$