Name: . . . . . .

ID Number: . . . . . .

COMP203: Mid-Term Test

20 August, 2003

Instructions

- Maximum time: **90 minutes**.
- Answer **all** the questions.
- There are 90 marks in total.
- Write your answers in the boxes in this test paper and hand in all sheets.
- Non-programmable calculators are allowed.
- Every box with a heavy outline requires an answer.
- Page 11 shows some commonly used MIPS instruction names and registers.

Questions	**Marks**
1. Basic Concepts	[10]
2. Registers, Memory and Big Constants	[10]
3. Decision Making	[10]
4. Addressing Modes and Instruction Formats	[10]
5. Number Conversion	[10]
6. Boolean Expression and Logic Gates	[10]
7. Multiplication	[10]
8. Overflow Detection and manipulation	[10]
9. Procedures/Functions	[10]
Total Marks	[90]

Question 1. Basic Concepts [10 marks]

Briefly define each of the following terms:

(a) [3 marks] Assembler:

(b) [3 marks] Compiler:

(c) [2 marks] Operating system:

(d) [2 marks] Sequential logic blocks:

Question 2. Registers, Memory and Big Constants [10 marks]

(a) [6 marks] Consider the following C statement:

```
A[10] = A[5] + h;
```

Assume that register `$s0` holds variable `h` and that register `$s1` holds the base address of the integer array `A`. Write a sequence of MIPS instructions that directly corresponds to this statement. Use temporary registers if necessary.

(b) [4 marks] Consider the following sequence of MIPS instructions:

```
lui $t1, 0x0123
ori $t2, $t1, 0xa021
addi $t3, $t1, 0xa021
```

What values will be stored in registers `$t1, $t2, $t3` after the above instructions are executed?

Question 3. Decision Making [10 marks]

Consider the following C code segment:

```
if (x < m)
   x = x - n;
else
   x = x + n;
x = x + 2;
```

Assume that the registers `$s0, $s1, $s2` hold the variables `x, m, n`, respectively.

Write a sequence of MIPS instructions that directly corresponds to this C code segment. Use temporary registers if necessary.

Question 4. Addressing Modes and Instruction Formats [10 marks]

(a) [5 marks] In MIPS assembly language, there are five addressing modes: (1) immediate/constant addressing, (2) register addressing, (3) base addressing, (4) PC-relative addressing, and (5) pseudodirect addressing. There are three instruction formats: *R–type*, *I–type* and *J–type*.

Give one example instruction for each of the five addressing modes and state the instruction format of each given instruction.

(b) [5 marks] Given the following MIPS instructions:

```
     slt $t0, $s1, $s0
     bne $t0, $zero, Else
     sub $s1, $s1, $s0
     j Exit
Else:addi $s1, $s1, 1
Exit:add $s1, $s1, $s2
```

Calculate the value of the branch relative address `Else` (*the offset*) in the second instruction `bne $t0, $zero, Else`. **Present the final result only**.

Question 5. Number Conversion [10 marks]

This question concerns different formats of numbers. **Write only the final answer into the boxes.**

(a) [2 marks] Convert the decimal number `-1025` into a 16-bit two's complement binary number.

(b) [2 marks] Convert the 16-bit two's complement binary number `1111 1111 1100 0000` into a decimal number.

(c) [3 marks] Show the IEEE 754 binary representation of the the decimal floating point number `-3.625` in single precision.

(d) [3 marks] Convert the following IEEE 754 single precision binary number
`1100 0000 1000 0000 0000 0000 0000 0000` into a decimal number.

Question 6. Boolean Expression and Logic Gates [10 marks]

Given the following truth table for a PLA (Programmable Logic Array):

Input			Output	
A	B	C	D	E
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(a) [4 marks] Give a boolean expression for each of D and E based on the truth table.

(b) [6 marks] Design a PLA (Programmable Logic Array) to implement the truth table based on the boolean expressions you gave in part (a).

Question 7. Multiplication [10 marks]

Calculate the following multiplication using the Booth's algorithm:

`0011` $\times$ `1111`

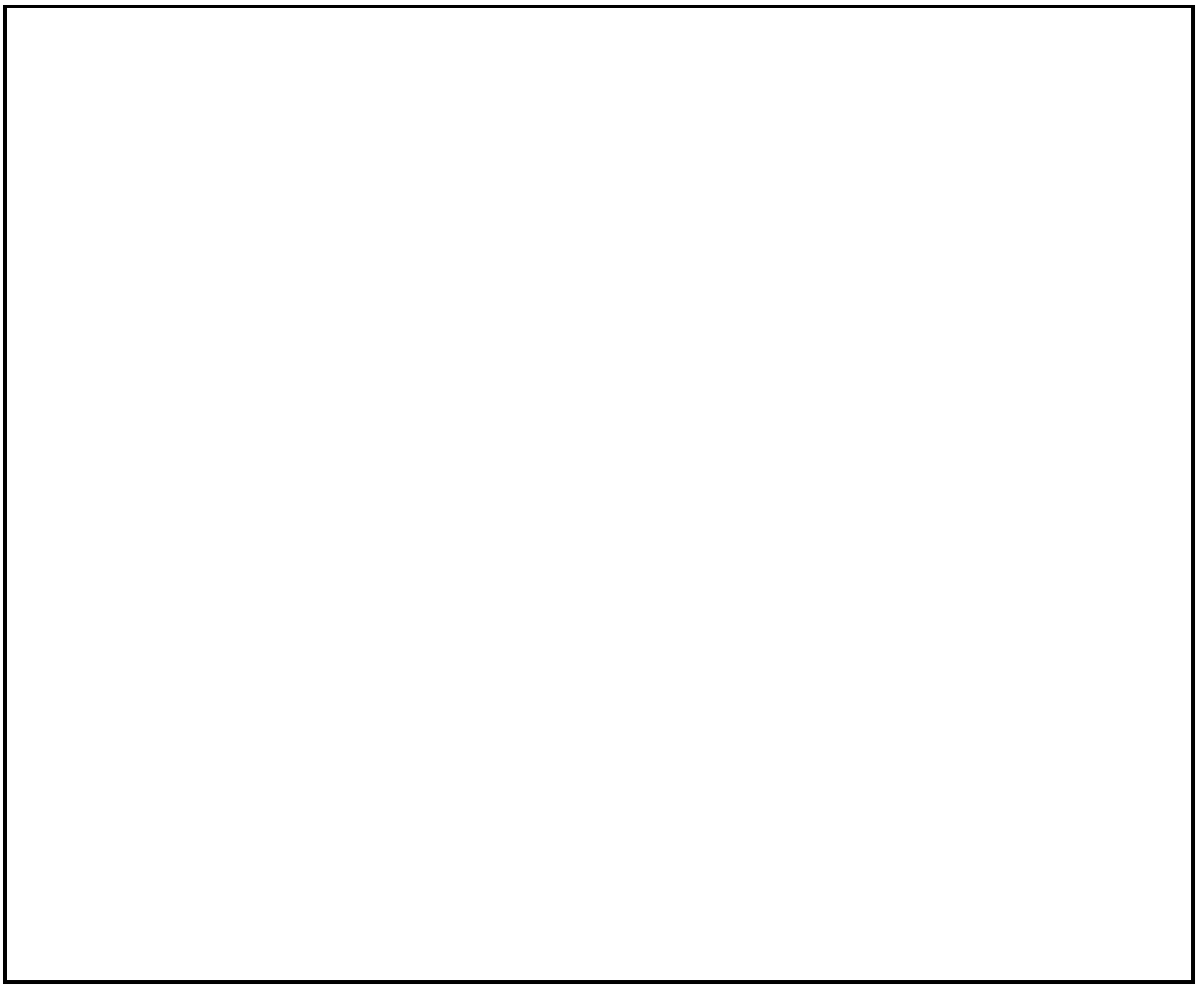
Assume that the multiplicand and the multiplier are 4-bit 2's complement integers (consider the sign). Show your work in a table and identify your final result.

Question 8. Overflow Detection and Manipulation [10 marks]

Assume that A is a negative integer, that B is a positive integer, and that variables A, B, and C are placed in registers `$s1, $s2` and `$s3`, respectively. Write **at most 10** MIPS instructions in total to perform the following tasks:

- $C = A - B$;
- If there is no overflow, then add decimal constant 100 to C (`$s3`) and place the result in register `$s4`;
- Otherwise, set the most significant bit of C (`$s3`) to 1.

Use temporary registers if necessary.



Question 9. Procedures/Functions [10 marks]

Given the following C procedure/function:

```
int test(int x, int y)
{
   int z;

   z = x + y - 1;

   return z;
}
```

Assume that register `$s0` holds the variable `z`. Write a sequence of MIPS instructions that directly corresponds to this function. Use temporary registers if necessary.

A Commonly Used MIPS Instructions

add	sub
lw	sw
addi	lui
and	or
andi	ori
sll	srl
jal	jr
j	
beq	bne
slt	slti
mult	div
mul	
lb	sb

B MIPS Registers — Numbers and Names

Name	Number	Usage
$zero	0	constant value 0
$at	1	reserved for assembler
$v0–$v1	2–3	values for results and expression evaluation
$a0–$a3	4-7	arguments, for functions/procedures
$t0–$t7	8-15	temporaries
$s0–$s7	16-23	saved. Fast locations for data
$t8–$t9	24-25	more temporaries
$k0–$k1	26-27	reserved for the OS
$gp	28	global pointer
$sp	29	stack pointer
$fp	30	frame pointer
$ra	31	return address, for functions/procedures

****