

THE HONG KONG UNIVERSITY OF SCIENCE & TECHNOLOGY

COMP180: Computer Organization

Spring Semester, 2001

MIDTERM EXAMINATION

March 19, 2001

STUDENT NAME:	_____
STUDENT NUMBER:	_____
LAB NUMBER:	_____

Instructions to Students

1. There are 6 problems and 9 pages. Check that you have all 9 pages.
2. Write your name, student number and lab number on this page.
3. Write your student number on each of the following pages.
4. Answer all questions in the space provided. Rough work should be done only on the back pages and afterwards drawn a diagonal line through it to show that it is not part of your answer.
5. Leave all pages stapled together.
6. The examination period will last for **1 hour and 30 minutes**.
7. Stop writing immediately when the time is up.

For Grading Purpose Only

Problem 1	_____ / 20
Problem 2	_____ / 15
Problem 3	_____ / 20
Problem 4	_____ / 15
Problem 5	_____ / 10
Problem 6	_____ / 20
 TOTAL:	 _____ / 100

1. (20 points)

(a) Show the single MIPS instruction or minimal sequence of instructions for this C statement:

$$x[10] = x[11] + c$$

Assume that c corresponds to register $\$t0$ and the integer array x has a base address of 4,000 (decimal).

Note that integers are represented by 32-bit words here.

(b) Explain why the following instruction cannot be translated into one single MIPS machine instruction.

`sw $a0, 80000($t0)`

(c) Translate the statement in b) into exactly two MIPS instructions so that it works.

2. (15 points)

The following code fragment processes an array and produces two important values in register \$v0 and \$v1. Assume that the array consists of 5000 words indexed 0 through 4999, and its base address is stored in \$a0 and its size (5000) in \$a1. Describe in one sentence what this code does. Specifically, what will be returned in \$v0 and \$v1?

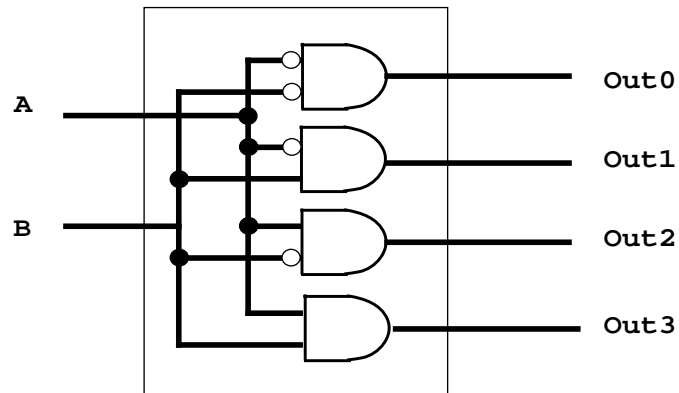
```

                                add    $a1,    $a1,    $a1
                                add    $a1,    $a1,    $a1
                                add    $v0,    $zero,    $zero
                                add    $t0,    $zero,    $zero
outer:      add    $t4,    $a0,    $t0
                                lw      $t4,    0($t4)
                                add    $t5,    $zero,    $zero
                                add    $t1,    $zero,    $zero
inner:      add    $t3,    $a0,    $t1
                                lw      $t3,    0($t3)
                                bne     $t3,    $t4,    skip
                                addi    $t5,    $t5,    1
skip:       addi    $t1,    $t1,    4
                                bne     $t1,    $a1,    inner
                                slt     $t2,    $t5,    $v0
                                bne     $t2,    $zero,    next
                                add     $v0,    $t5,    $zero
                                add     $v1,    $t4,    $zero
next:      addi    $t0,    $t0,    4
                                bne     $t0,    $a1,    outer

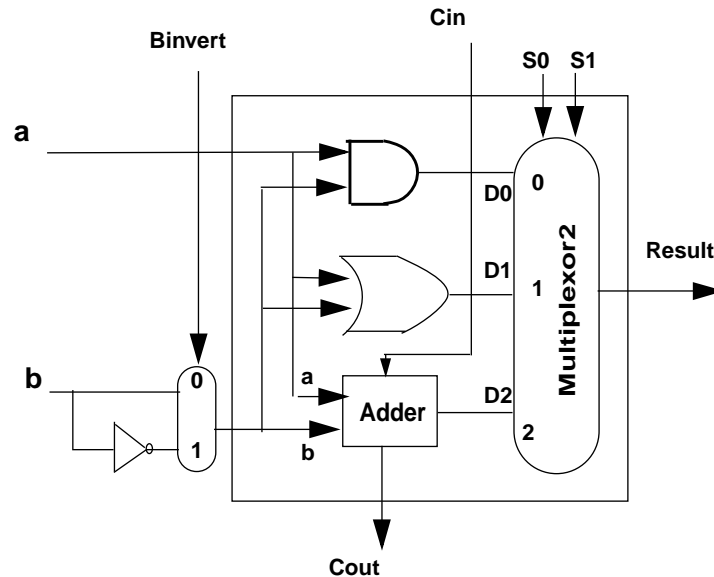
```

3. (20 points)

(a) Give the truth table of the circuit in the following figure, deduce in one word what is the function of this circuit.



Given the 1-bit ALU shown in the following figure, answer the questions in part b to f



(b) Provide the truth table of the adder showing both the Sum and the Cout. Be sure to use labels a, b, and Cin.

(c) Write the equations for Sum and Cout.

(d) Give the implementation of the 1-bit Adder using only simple gates (AND, OR, and NOT). To avoid any confusion, you may separate the Sum and Cout into two circuits.

(e) Show the truth table of multiplexor2 with two inputs (S0 and S1) and 1 output (Result). Be sure to use the labels D0, D1, D2, S0, and S1.

(f) Give an implementation of the multiplexor2 using simple gates.

4. (15 points)

A processor P with a clock rate of 100 MHz has the following characteristics:

<u>Instruction class</u>	<u>CPI</u>	<u>Frequency</u>
A	2	40%
B	3	30%
C	4	20%
D	5	10%

It is possible to have an improved processor design P' with a clock rate of 120 MHz. The instruction classes now have shorter CPIs as follow:

<u>Instruction class</u>	<u>CPI</u>	<u>Frequency</u>
A	2	40%
B	2	30%
C	3	20%
D	4	10%

(a) What is the CPI of processors P and P'?

(b) How much faster is P' compared to P?

It is possible to improve the compiler for P such that the mean frequencies of the instruction classes are changed as follows (clock rate remains at 100 MHz):

<u>Instruction class</u>	<u>CPI</u>	<u>Frequency</u>
A	2	50%
B	3	35%
C	4	10%
D	5	5%

(c) What is the CPI of P with the new compiler?

(d) How much faster is P with the new compiler?

5. (10 points)

Find the shortest sequence of MIPS instructions to determine if there is a carry out from the addition of two registers, say, register \$t3 and \$t4. Place a 0 or 1 in register \$t2 if the carry out is 0 or 1, respectively. (Hint: It can be done in two instructions.)

6. (20 points)

The following MIPS code is a fragment of a program that calculates in register \$s3 an approximation of the exponential of a number x given in register \$s0 up to the k th term where k is given in register \$s1 using the following approximation:

$$\exp\{x\} = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^j}{j!} + \dots$$

Assuming the main program and the two procedures are loaded into memory at the addresses shown on the left of the instructions (in decimal representation):

```

00004100      addi    $s3,    $zero, 1
00004104  loop:  add    $a0,    $zero, $s0
00004108      add    $a1,    $zero, $s1
00004112      jal    term
00004116      add    $s3,    $s3,    $v0
00004120      addi    $s1,    $s1,    -1
00004124      beq    $s1,    $zero, end
00004128      j      loop
00004132  end:    /* syscall to terminate the program */
...
...

00004200  term:  ...
00004204      add    $t2,    $zero, $a1
00004208      jal    fact
00004212      add    $t0,    zero,    $a0
00004216  num:  addi    $t2,    $t2,    -1
00004220      beq    $t2,    $zero, ret1
00004224      mult   $t0,    $t0,    $a0
00004228      j      num
00004232  ret1:  div    $v0,    $t0,    $v0
00004236      jr     $ra

00004440  fact:  ...
00004444      addi    $v0,    $zero, 1
00004448      mult   $v0,    $v0,    $a1
00004452      addi    $a1,    $a1,    -1
00004456      beq    $a1,    $zero, ret2
00004460      j      fact
00004464  ret2:  jr     $ra

```

(a) Expand each of the two instructions at addresses 00004112 and 00004208 into two instructions each and explain why.

(b) What is the content of register \$ra when the PC content is 00004204

(c) What is the content of register \$ra when the PC content is 00004444

(d) What is the content of register \$ra when the PC content is 00004212

(e) What is the PC content when the instruction at address 00004236 is executed? If there is a problem explain it and why it occurred.

(f) What should be done to solve the problem in e)? Just describe, no code is needed.