ID: B

CSE 2021 Computer Organization Midterm Test (Fall 2009)

Instructions

- ❑ This is a closed book, 80 minutes exam.
- ❑ The MIPS reference sheet may be used as an aid for this test.
- ❑ An 8.5” x 11” “Cheat Sheet” may also be used as an aid for this test. MUST be original handwriting.
- ❑ This is a question/answer booklet: Write your answers in the space provided and as indicated in each question. Use the backside for scratch work. *Do not* hand in anything other than this booklet.
- ❑ Fill in your personal data in the box below *before the start of the exam* and then wait until the instructor has distributed the exams to all students. *Do not* turn this page over until the instructor has announced that you may do so.
- ❑ Keep your York photo ID (or any other acceptable photo ID) on the desk in front of you so that the instructor may inspect it without disturbing you.
- ❑ You may use ONLY those instructions that appear in the MIPS sheet. Whenever needed, assume that **the machine is little endian.**
- ❑ No questions during the exam. Write your final answer with a pen.

LAST NAME: ______________________________

FIRST NAME: ______________________________

STUDENT NUMBER: ______________________________

PRISM LOGIN: ______________________________

Section	Points	Mark
A	20	
B	20	
C	10	
TOTAL	**50**	

Name:___________________________

Student #:________________________

YORK
UNIVERSITÉ
UNIVERSITY

Section A <5 questions x 4 points each = 20 points>

For questions in this section, specify the content of the required register(s), in the required radix after the specified code is executed. Note the following clarifications:

A We specify the hexadecimal notation with the prefix of 0x such that representation15_{ten} = F_{hex} is equivalent to 0x0000000F or 0xF in short.

B Assume data specified in MIPS instruction is stored as a contiguous block.

.

1.

```
addi $t1, $0, 64
addi $t2, $t1, -45
div  $t1, $t2
mfhi $t3
mflo $t4
```

$t3 (in decimal) =

$t4 (in decimal) =

2.

```
addi $t0, $0, 0x7
addi $t1, $t0, 0xc
addi $t2, $0, 40
mult $t1, $t2
mflo $t1
```

$t1 (in hex) =

$t2 (in hex) =

3.

```
                  la   $t0, char
                  lb   $t2, 5($t0)
                  addi $t2, $t2, -43
                  add  $t0, $t0, $t2
                  lb   $t3, 0($t0)
                  .data
        char:     .asciiz   “a1b2c3d4e5”
```

$t2 (in decimal) =

$t3 (in decimal) =

4.

```
                  .data
        an:       .word     19, 59, -13
        bn:       .word     15, 9, -2, -7
                  .text
        main:     addi $t2, $0, 24
                  lb   $t3, an($t2)
                  lb   $t2, an($0)
```

$t2 (in decimal) =

$t3 (in decimal) =

5.

```
        start:    addi $a0, $0, 15
                  addi $t0, $0, 0
                  addi $t1, $0, 3
        again:    slt  $t2, $a0, $t1
                  bne  $t2, $0, end
                  add  $t0, $t0, $t1
                  addi $t1, $t1, 3
                  j    again
        end:      add  $v0, $t0, $0
```

$v0 (in decimal) =

Section B <2 questions x 10 points each = 20 points>

For each question, write your answer in the box provided.

6. A binary-to-seven segment decoder is a logic circuit that converts a number expressed in binary to an appropriate code for the selection of segments in a display indicator illustrated in fig. 1.

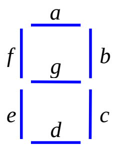


Fig. 1: Segment designation in a seven segment decoder

The binary-to-seven segment decoder has three input bits, say *x*, *y*, and *z* that hold the 3-bit binary representation of the number to be displayed. The seven outputs of the decoder (*a*, *b*, *c*, *d*, *e*, f, and *g*) select the corresponding segments in the display are shown in fig. 2 to highlight the numeric digits being displayed.

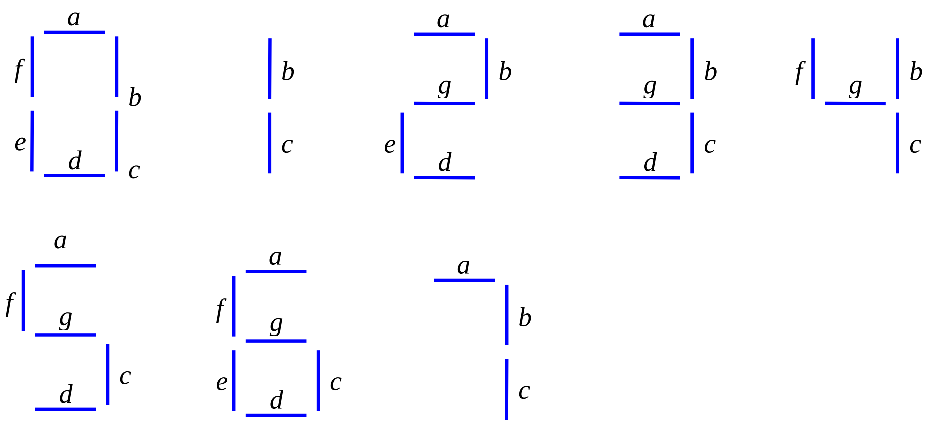


Fig. 2: Numerical designation for display of a number (from 0 to 7)

As an example, if binary number 001 is to be displayed, the binary inputs to the decoder are $x = y = 0$ and $z = 1$ with the outputs $b = c = 1$. The remaining five outputs *a*, *d*, *e*, *f*, and *g* are all 0. For the binary-to-seven segment decoder, shown in figs. 1 and 2:

a) Draw the truth table with three inputs (*x*, *y*, and *z*) and ONLY the first three outputs (*a*, *b*, *c*).

Inputs			*Outputs*		
x	*y*	*z*	*a*	*b*	*c*
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

b) Based on the truth table drawn in a), derive a Boolean expressions for each of the first three outputs (*a*, *b*, *c*).

c) Provide simplified Boolean expressions for each of the outputs in part (b).

c) Using AND, OR, and NOT gates, draw digital circuits that implement the first 3 outputs (*a*, *b*, *c*) from the inputs (*x*, *y*, and *z*).

.

7. Fig. 1 shows the schematic diagram of a `NOR` gate

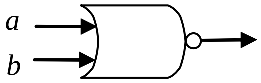


Fig.1: Schematic diagram of the `NOR` gate

which has the following truth table

a	*b*	*c* = *a* `NOR` *b*
0	0	1
0	1	0
1	0	0
1	1	0

A `NOR` gate is a universal gate because any digital component can be implemented using `NOR` gates only. Implement (a) a `NOT` gate, (b) an `AND` gate and (c) an `OR` gate using only `NOR` gates. You may use more than one `NOR` gate but no other type of gate may be used.

a) a `NOT` gate

(b) an `AND` gate

(c) an OR gate

.

Section C <1 question x 10 points each =10 points>

Write your answer in the box provided. You have to follow the MIPS standard conventions of using the registers, i.e. registers $a0 - $a4 are used for passing arguments to a procedure, registers $v0 - $v1 for returning results, etc.

8. Write a minimal sequence of MIPS instructions that swaps the contents of registers $s1 and $s2 if the value in $s1 is greater than or equal to the value stored in $s2. Otherwise, it leaves the contents of the registers unchanged.

CSE 2021 Computer Organization Midterm Test (Fall 2009)
Answer Section

SHORT ANSWER

1. ANS:

```
$t3 (in decimal) = 7

$t4 (in decimal) = 3
```

PTS: 1 OBJ: Section A

2. ANS:

$t1 (in hex) = 760_{10} = 0x2F8

$t2 (in hex) = 40_{10} = 0x28

PTS: 1 OBJ: Section A

3. ANS:

```
$t2 (in decimal) = 8

$t3 (in decimal) = 101
```

PTS: 1 OBJ: Section A

4. ANS:

```
$t2 (in decimal) = 19

$t3 (in decimal) = -7
```

PTS: 1 OBJ: Section A

5. ANS:

```
$v0 (in decimal) = 45
```

PTS: 1 OBJ: Section A

PROBLEM

6. ANS:

(a) The truth table for the binary-to-seven segment decoder is shown below (all 7 outputs are shown only a, b, c required for this question):

Inputs			Outputs						
x	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*
0	0	0	1	1	1	1	1	1	0
0	0	1	0	1	1	0	0	0	0
0	1	0	1	1	0	1	1	0	1
0	1	1	1	1	1	1	0	0	1
1	0	0	0	1	1	0	0	1	1
1	0	1	1	0	1	1	0	1	1
1	1	0	1	0	1	1	1	1	1
1	1	1	1	1	1	0	0	0	0

(b) The Boolean expressions (for all 7, only a, b, c required) are given below

$$a = x'y'z' + x'yz' + x'yz + xy'z + xyz' + xyz\,,$$

$$b = x'y'z' + x'y'z + x'yz' + x'yz + xy'z' + xyz\,,$$

$$c = x'y'z' + x'y'z + x'yz + xy'z' + xy'z + xyz' + xyz\,,$$

$$d = x'y'z' + x'yz' + x'yz + xy'z + xyz'\,,$$

$$e = x'y'z' + x'yz' + xyz'\,,$$

$$f = x'y'z' + xy'z' + xy'z + xyz'\,,$$

and

$$g = x'yz' + x'yz + xy'z' + xy'z + xyz'\,.$$

(c) The expressions for *a*, *b*, *c* can be simplified fairly considerably:

$$\mathrm{a} = \mathrm{x'z'} + (\mathrm{x'y} + \mathrm{xy'})\mathrm{z} + \mathrm{xy} = \mathrm{x'z'} + (\mathrm{x} + \mathrm{y})(\mathrm{xy})'\mathrm{z} + \mathrm{xy}$$

$$= \mathrm{x'z'} + \mathrm{xz} + \mathrm{yz} + \mathrm{xy} = (\mathrm{x} + \mathrm{z})' + \mathrm{xz} + \mathrm{y}(\mathrm{x} + \mathrm{z}) = \mathrm{y} + \mathrm{x'z'} + \mathrm{xz}$$

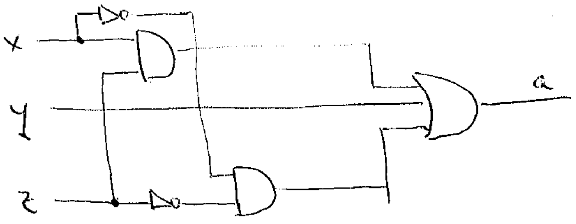
$$\mathrm{b} = \mathrm{x'z'} + \mathrm{yz} + (\mathrm{x'z} + \mathrm{xz'})\mathrm{y'} = \mathrm{x'z'} + \mathrm{yz} + (\mathrm{x} + \mathrm{z})(\mathrm{xz})'\mathrm{y'} = (\mathrm{x} + \mathrm{z})' + \mathrm{yz} + (\mathrm{x} + \mathrm{z})(\mathrm{xz})'\mathrm{y'}$$

$$= \mathrm{x'z'} + \mathrm{yz} + \mathrm{x'y'} + \mathrm{z'y'} = \mathrm{x'}(\mathrm{z'} + \mathrm{y'}) + \mathrm{yz} + \mathrm{y'z'} = \mathrm{x'}(\mathrm{yz})' + \mathrm{yz} + \mathrm{y'z'} = \mathrm{x'} + \mathrm{yz} + \mathrm{y'z'}$$

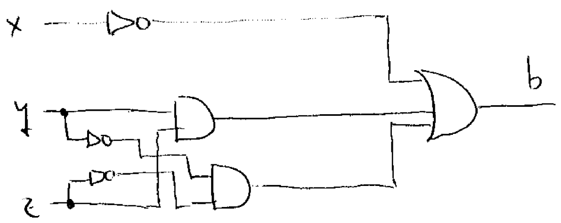
$$c = x'y' + yz + xy' + xyz' = y' + y(z + xz') = y' + y(z + x) = x + y' + z$$

(d)The digital circuits can be drawn from the above Boolean expressions.

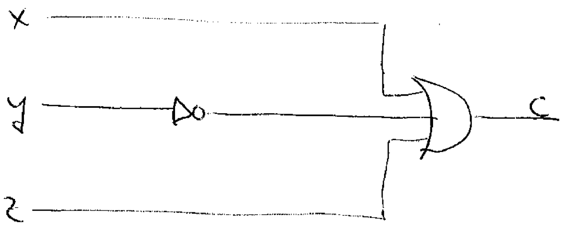
a: $a = y + x'z' + xz$



b: $b = x' + \cancel{x}y'z' + yz$



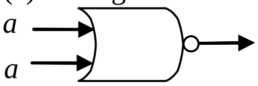
c: $c = x + y' + z$



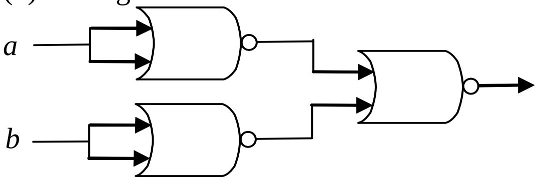
PTS: 1 OBJ: Section B.2

7. ANS:

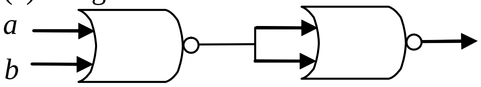
(a) NOT gate



(b) AND gate



(c) OR gate



PTS: 1

8. ANS:

```
slt   $t0, $s1, $s2
addi  $t1, $0, 1
beq   $t0, $t1, exit
addi  $t2, $s1, 0
addi  $s1, $s2, 0
addi  $s2, $t2, 0
exit:
```

PTS: 1 OBJ: Section C