YORK
U
UNIVERSITÉ
UNIVERSITY

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
U
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 representation $15_{ten} = F_{hex}$ is equivalent to 0x0000000F or 0xF in short.

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

.

1.
```
            .data
    x:    .byte     1, 2, 3, 4, 5
    y:    .word     5, 10, 15, 20, 25
            .text
            addi $t0, $0, 8
            lw   $t1, x($t0)
            sll  $t2, $t1, 4
            ori  $t3, $t2, 12
```

$t2 (in hex) =

$t3 (in hex) =

2.
```
                .data
    aa:         .asciiz    "abcde"
    bb:         .ascii     "12345"
    cc:         .asciiz    "xyz"
                .text
    main:       la    $t1, aa
                la    $t2, bb
                la    $t3, cc
                addi $t2, $t3, -2
                sub  $t2, $t2, $t1
                sub  $t1, $t3, $t1
```

$t1 (in decimal) =

$t2 (in decimal) =

3.
```
                .data
    an:         .asciiz   "1234abcd"
    bn:         .byte     1,2,3,8,9,10,11
                .text
    main:       la    $t1, an
                la    $t4, bn
                lw    $t2, 3($t1)
                lb    $t3, 1($t4)
                add   $t3, $t2, $t3
```

$t2 (in hex) =

$t3 (in hex) =

4.
```
    main:       add  $t2, $0, $0
                lui  $t2, 0x001f
                addi $t3, $t2, 0x7fff
```

$t2 (in hex) =

$t3 (in hex) =

5.
```
    main:  la    $t0, array
           addi $t1, $0, 12
           lw    $t2, array($t1)
           add   $t3, $t1, $t2
           .data
    array: .word 10, 27, 36, 57
```

$t2 (in decimal) =

$t3 (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.
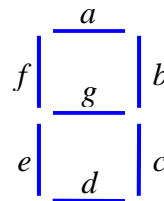
Fig. 1: Segment designation in a seven segment decoder

A particular design of the binary-to-seven segment decoder has three input bits, say x, y, and z that contain 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.
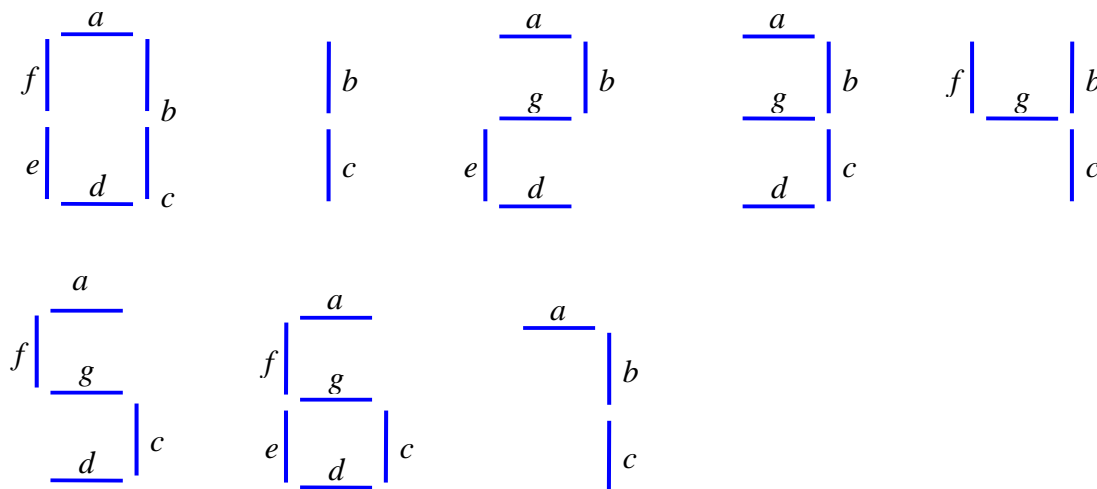
Fig. 2: Numerical designation for display of a number (from 0 to 7)

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 outputs (*d* to *g*).

Inputs			Outputs			
x	*y*	*z*	*d*	*e*	*f*	*g*
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 outputs (*d* to *g*).

c)   Simplify the Boolean expressions from the previous question to minimize the number of logic gates required.

d)   Using AND, OR, and NOT gates, draw digital circuits that implement the 4 outputs (*d* to *g*) from the
        inputs (*x*, *y*, and *z*).

.

7. The figure below shows the schematic diagram of a NAND gate

a

b

c

Fig.1: Schematic diagram of the NAND gate

which has the following truth table

a	b	$c = a$ NAND b
0	0	1
0	1	1
1	0	1
1	1	0

A NAND gate is a universal gate because any digital component can be implemented using NAND gates only. Implement (a) a NOT gate, (b) an AND gate, and (c) an OR gate using only NAND gates. You may use more than one NAND 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. For this question assume the following:

    variables a, b, c, d, e are mapped to registers $s1 to $s5
    starting address of **byte array** "small" is stored in $s7

    Write a minimal sequence of MIPS instructions that execute the following operation:

    small[5] = b + small[5]

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

SHORT ANSWER

1. ANS:

   $t2 (in hex) = 0x50

   $t3 (in hex) = 0x5c

   PTS: 1   OBJ: Section A

2. ANS:

   $t1 (in decimal) = 11

   $t2 (in decimal) = 9

   PTS: 1   OBJ: Section A

3. ANS:

   $t2 (in hex) = 0x63626134,

   $t3 (in hex) = 0x63626136

   PTS: 1   OBJ: Section A

4. ANS:

   $t2 (in hex) = 0x001f0000

   $t3 (in hex) = 0x001f7fff

   PTS: 1   OBJ: Section A

5. ANS:

```
$t2 (in decimal) = 57

$t3 (in decimal) = 69
```

PTS: 1   OBJ: Section A

PROBLEM

6. ANS:

(a) The truth table for the binary-to-seven segment decoder is shown below:

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 seven outputs) are given below only d through g are required for this question:

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

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

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

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

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

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

and
$$g = x'yz' + x'yz + xy'z' + xy'z + xyz'.$$

(c) The expressions for *d, e, f* and *g* can be simplified fairly considerably:

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

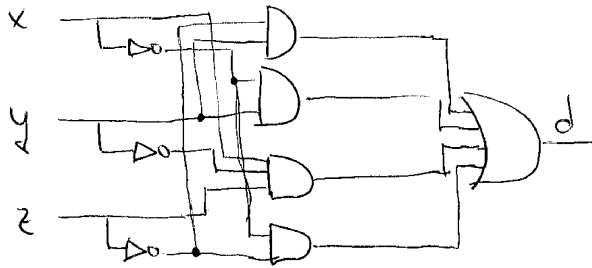$$e = x'y'z' + x'yz' + xyz' = x'z'+xyz'= (x'+xy)z'= x'z'+yz'$$

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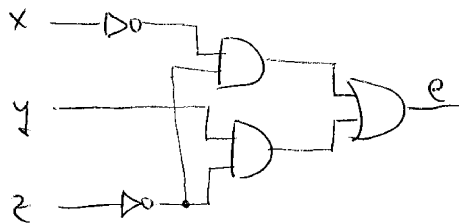
$$g = x'yz' + x'yz + xy'z' + xy'z + xyz' = x' y + xy'$$

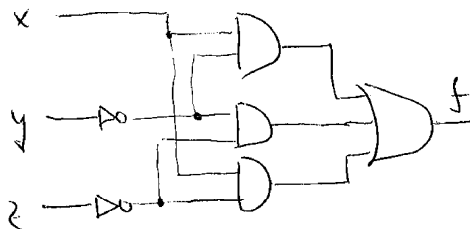(d)The digital circuits can be drawn from the above Boolean expressions.

d: $d = x'y + x'z' + xy'z + yz'$

e: $e = x'z' + yz'$
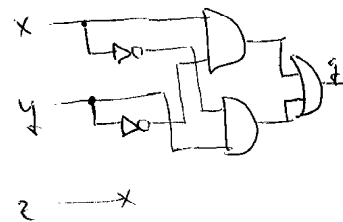

f: $f = xy' + xz' + y'z'$ g: $g = x'y + xy'$

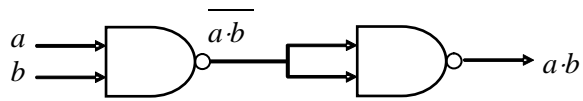PTS: 1   OBJ: Section B.2

7. ANS:
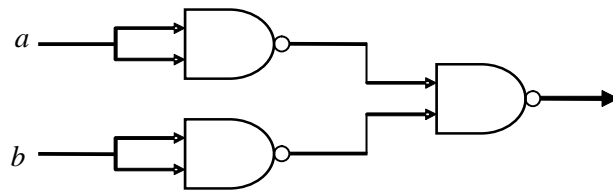   (a) NOT

    (b) AND

    (c) OR

    PTS: 1   OBJ: Section B.1

8. ANS:

```
lb    $t0, 5($s7)    # $t0 = small[5]
add   $t0, $t0, $s2  # $t0 = b + small[5]
sb    $t0, 5($s7)    # small[5] <-- $t0
```

    PTS: 1   OBJ: Section C