YORK UNIVERSITY

Winter 2019 Midterm Exam

EECS2021: Computer Organization

Duration: 1 hour 20 minutes

February 28th, 2019

Last Name: ____

First Name: ____

Student Number: ____________________________________

Instructor: **Dr. Rabia Bakhteri**

Lecture section: **M**

Instructions:

- **Do not open this exam until you hear the signal to start.**
- Have your student ID on your desk.
- No aids permitted other than writing tools. If you write in pencil, we reserve the right to deny any remark requests.
- Keep all bags and notes far from your desk before the exam begins.
- There are 4 parts on 9 pages. When you hear the signal to start, make sure that your exam paper has all the printed pages before you begin.
- Read over the entire exam before starting.
- Reference material is provided on the **Appendix** page at the back of this exam paper.

Mark Breakdown	
Part A	/ 10
Part B	/ 30
Part C	/ 24
Part D	/ 36
Total	**/ 100**

Part A: Short Answer (10 marks)

Answer the following questions in the space provided. When providing a written answer, write **as clearly and legibly as possible**. Marks will not be awarded to unreadable answers.

1. Which ones of the following hexadecimal addresses are valid PC values when translated into 32-bit binary addresses? Circle all that apply. **(2 marks)**

a) `E45B23C1` [✓ **b)** `4C32A330`]

[✓ **c)** `9B31FF1C`] **d)** `1234ABCD`

2. Which is of the following registers always stores a value that is divisible by 4? Circle all that apply. **(2 marks)**

[✓ **a)** `x0`] **b)** `x2`

c) `sp` [✓ **d)** `ra`]

3. Which operations cannot be performed by single assembly instruction? Circle all that apply. **(2 marks)**

a) divide by 32 **b)** decrementing

c) increment by 4 [✓ **d)** swap register contents]

4. The processor's control unit is basically an advanced __________. **(1 mark)**

a) adder circuit [✓ **b)** finite state machine]

c) register circuit **d)** multiplexer

5. Which of the following 4-bit signed binary arithmetic operations will cause an overflow? Circle all that apply. **(2 marks)**

a) `1001 + 0100` **b)** `1010 + 1110`

c) `1111 + 0001` [✓ **d)** `0100 + 0100`]

6. A one-hot decoder has a binary output of `0000010000000000`. What is the input? **(1 mark)**

01010

7. What is the 2's complement representation of the following decimal value: −76. **(2 marks)**

Binary = 10110100________ **Hexadecimal=**0xB4________

8. How many bytes can register x1 of the RISC V architecture store? **(1 mark)**

a) 64 **b)** 32

[✓ **c)** 16] **d)** 8

9. Which instruction can be used to copy an immediate value into the top half of a register? **(1 mark)**

load upper immediate or LUI

10. How many bytes can a register unit store, with 8 address bits and 32-bit words? **(2 marks)**

2^8 x 32/8 = 1024

As mentioned in class, this time the midterm exam will not have the short answer question sections or the theory questions. Instead there will be questions in the final exam.

Part B: Calculation (30 marks)

1. Computer X has a CPI of 1.2 and a 700MHz clock, while Computer Y has a CPI of 2.6 with a 1.3GHz clock. A particular program has 75000 instructions when compiled for Computer X. How many instructions would it need to have when compiled for Computer Y if the execution time on Computer Y is to be 25% faster than Computer X? **(5 marks)**

Theses calculation questions were solved in class on Tuesday, 26 Feb

2. Two processors, P_A, and P_B have identical instruction set. There are three classes of instructions (P, Q and R) in the instruction set. P_A runs at 100MHz and P_B at 80MHz. The average number of cycles for each instruction class and their frequencies (for a typical program) are as follows:

Int. class	P_A CPI	P_B CPI	Frequency
P	4	4	15%
Q	2	3	25%
R	1	2	60%

a) Calculate the average CPI for P_A, and P_B. **(4 marks)**

b) Calculate the average MIPS ratings for P_A, and P_B. Which machine has a smaller MIPS rating? **(5 marks)**

c) Which individual instruction class CPI do you need to change, and by how much, to have this machine have the same or better performance as the machine with the higher MIPS rating (you can only change the CPI for one of the instruction classes on the slower machine)?

3. If processor A has a higher clock rate than processor B, and processor A also has a higher MIPS rating than processor B, explain whether processor A will always execute faster than processor B. Suppose that there are two implementations of the same instruction set architecture. Machine A has a clock cycle time of 20ns and an effective CPI of 1.5 for some program, and machine B has a clock cycle time of 15ns and an effective CPI of 1.0 for the same program. Which machine is faster for this program, and by how much?

4. Use the register and memory values in the table below for the next questions. Assume all numbers are given in hexadecimal base. Assume each of the following questions starts from the table values; that is, **DO NOT** use value changes from one question as propagating into future parts of the question. Assume the values are in decimal **(6 marks)**

Register	Value
R1	0
R2	0
R3	400
R4	380

a) `sub R1, R3, R4`

R1	**20**
R3	**400**
R4	**380**

b) `beq R1, R2, 2000`

R1	**0**
R2	**0**
R3	**400**

c) `xori R2, R1, 9`

R1	**0**
R2	**9**
R3	**400**

Part D: Processor Instructions (24 marks)

1. For each assembly language instruction below, fill in the blanks with the corresponding 32-bit machine code instruction. For bits that could be either a **0** or a **1**, fill in the blank with an **X** value instead. **(12 marks)**

a) `ld x12, 4(x9)`

`000000000100  01001 011 01100 0000011`

b) `addi x3, x2, 19`

`000000010011  00010 000 00011 0010011`

c) `or x19, x17, x13`

`0000000 01101 10001 110 10011 0110011`

2. Given the machine code instructions below, write the corresponding assembly language instruction in the space below each machine code instruction. **(12 marks)**

a) `00001111111100010110000010010011`

`ori x1, x2, 255`

b) `01000000110001001000010100110011`

`sub x10, x9, x12`

c) `00000000001100101101000110010011`

`srli x3, x5, 3`

Part F: Assembly Language (36 marks)

1. In the spaces provided below, write the assembly language instruction(s) that perform the following tasks. ***Full marks will only be given for one-instruction answers***! **(12 marks total)**

a) Divide the value stored in `x11` by 16 and stored it back in `x11`. **(3 marks)**

```
srli x11, x11, 4
```

b) Store `0xFFFF FFFF FFFF FFFF` into register `x9`. **(3 marks)**

```
addi x9, x0, -1
```

c) Store the integer 55 into register `x2`. **(3 marks)**

```
addi x2, x0, 55
```

d) Store 1's complement of `x6` (inverting all the bits) in `x8`. **(3 marks)**

```
xori x8, x6, -1
```

2. Write the RISC V instructions that will perform that operation. Only implementations that use at most 2 operations will get full marks. **(12 marks total)**

a) How to determine if content of x11 is an odd number and if it is, then branch to label 'Odd'? **(4 marks)**

```
andi   x4, x11, 1
bne    x1, x0, Odd
```

b) Pop the value in the stack and save to x3. **(4 marks)**

```
ld     x3, 0(sp)
addi   sp, sp, 8
```

c) Set content of `x4` to `0xAA0BB0CC`. **(4 marks)**

```
lui x4, 0xAA0BB        # x4=0xAA0BB000
addi x4, x4, 0x0CC     # x4=0xAA0BB0CC
```

3. In the space provided, write the operation performed by each assembly program. **(12 marks)**

```
x8 contains base address of array
A.

main:     addi x9, x8, x0
          addi x10, x0, 5
top:      ld x1, 0(x9)
          add x1, x1, x1
          sd x1, 0(x9)
          addi x10, x10, -1
          addi x9, x9, 8
          bne x10, x0, top
end:
```

Multiply each element in A with 2 and save back into A

```
x8 contains base address of array
A.

          add x9, x8, x0
          addi x10, x0, 5
          add x11, x0, x0
alpha:    ld x12, 0(x9)
          add x11,x11,x12
          addi x10,x10,-1
          addi x9,x9,8
          blt x10, x0, alpha
```

Copy the first element of A into x11

Exam Tip: assume array A = [-3, 0, 1, 2, 3]. Run through the code and see what happens to the values as you loop through the assembly code. Thursday's exam Qs are slightly easier than these examples.

```
x8 contains base address of array
A.
main:     addi x9, x8, x0
          addi x10, x0, 5
top:      ld x15, 0(x9)
          addi x10, x10, -1
          addi x9, x9, 8
          add x15, x15, x15
          sd x15, 0(x9)
          bne x10, x0, top
end:
```

A = A[0] multiplied with [1,2,4,8,16]