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	**/ 30**
Part B	**/ 24**
Part C	**/ 36**
Total	**/ 100**

Part A: Calculation (30 marks)

Answer the following questions in the space Assume computer A (CPI of 1.3 at 600MHz) and computer B (CPI of 2.5 at 750Mhz). A particular program, when compiled for computer A, has 100,000 instructions. How many instructions would the program need to have when compiled for Computer B, in order for the two computers to have exactly the same execution time for this program? **(5 marks)**

(CPUTime)A = (Instruction count)A * (CPI)A * (Clock cycle Time)A
= (100,000)*(1.3)/(600*10^6) ns

(CPUTime)B = (Instruction count)B * (CPI)B * (Clock cycle Time)B
= (I)B*(2.5)/(750*10^6) ns

Since (CPUTime)A = (CPUTime)B, we have to solve for (I)B and get 65000

1. Consider two processor variants, M1 and M2 that have identical instruction sets. There are three classes of instructions (A, B, and C) in the instruction set. M1 runs at 100MHz and M2 at 120MHz. The average number of cycles for each instruction class and their frequencies (for a typical program) are as follows:

Int. class	M1 CPI	M2 CPI	Frequency
A	1	2	60%
B	3	4	30%
C	5	6	10%

a) Calculate the average CPI for M1 and M2. **(4 marks)**

M1: CPI = 1*0.6 + 3*0.3 + 5*0.1 = 2.0
M2: CPI = 3*0.6 + 3*0.3 + 6*0.1 = 3.0

b) Calculate the average MIPS ratings for M1 and M2. Which machine has a smaller MIPS rating? **(5 marks)**

M1: Average MIPS rating = Clock Rate/(CPI * 10^6)
= (100 * 10^6) / (2.0*10^6)
= 50.0
M2: Average MIPS rating = Clock Rate/(CPI * 10^6)
= (120 * 10^6) / (3.0*10^6)
= 40.0
Machine M2 has a smaller MIPS rating.

2. The design team for a simple, single-issue processor is choosing between a pipelined or non-pipelined implementation. Here are some design parameters for the two possibilities:

Parameter	Normal Version	Enhanced Version
Clock Rate	500 MHz	350 MHz
CPI for ALU instructions	1	1
CPI for Control instructions	2	1
CPI for Memory instructions	2.7	1

a) For a program with 20% ALU instructions, 10% control instructions and 75% memory instructions, which design will be faster? Give a quantitative CPI average for each case. **(5 marks)**

Average CPI for Normal Version = (0.2*1 + 0.1*2 + 0.75*2.7)
= 2.425
Average CPI for Enhanced Version = (0.2*1 + 0.1*1 + 0.75*1)
= 1.05

CPU execution time for Normal version = 2.425/(500 Mhz)
= 4.85ns
CPU execution time for Enhanced version = 1.05/(350 Mhz)
= 3.0ns

The Enhanced version is faster.

b) For a program with 80% ALU instructions, 10% control instructions and 10% memory instructions, which design will be faster? Give a quantitative CPI average for each case. **(5 marks)**

Average CPI for Normal Version = (0.8*1 + 0.1*2 + 0.1*2.7)
= 1.27
Average CPI for Enhanced Version = (0.8*1 + 0.1*1 + 0.1*1)
= 1.0
CPU execution time for Normal version = 1.27/(500 Mhz)
= 2.54ns
CPU execution time for Enhanced version = 1.0/(350 Mhz)
= 2.86ns

The pipelined version is faster.

3. 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. **(6 marks)**

Register	Value	Memory Location
R1	12	12
R2	16	16
R3	20	20
R4	24	24

a) `add R3, R2, R1`

R1	12
R2	16
R3	28

b) `ld R3, 12(R1)`

R1	12
R2	16
R3	24

c) `addi R2, R3, 16`

R1	12
R2	36
R3	20

Part B: RISC V Machine Code conversion (24

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) `add x8, x18, x1`

0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1

b) `slt x5, x1, x19`

0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1

c) `sw ra, 8(sp)`

0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	1

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

0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	1	1	0	0	1	1

a) `___xor x4, x2, x3_____`

0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	1

b) `___ld x12, 4(x9)_____`

0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	1	1

c) `___sub x17,x16,x15_____`

Part C: 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) Multiply the value stored in `x1` by 4 and store it back in x1. **(3 marks)**

```
sll x1, x1, 2
```

b) Store the remainder of dividing `x3` by 32 in `x9`. **(3 marks)**

```
xori x9, x3, 32
or
srli x9, x3, 5
```

c) Store the integer 258 into register `x6`. **(3 marks)**

```
addi x6, x0, 258
```

d) Invert all the bits of register `x7`. **(3 marks)**

```
xori x7, x7, -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 x4 is an even number and if it is, then branch to label 'Even'? **(4 marks)**

```
andi   x1, x4, 1
beq    x1, x0, Even
```

b) Push the value in `x8` onto the stack. **(4 marks)**

```
addi   sp, sp, -8
sw     x8, 0(sp)
```

c) Set content of `x10` to `0x770990BB`. **(4 marks)**

```
lui x10, 0x77099      # x10=0x77099000
addi x10, x10, 0x0BB  # x10=0x770990BB
```

All students get 3 marks for this section because of typo errors.

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

```
x8 contains base address of array
A.
          add x9, x8, x0
          add x10, x0, x0
          add x11, x0, x0
Loop:     ld x12, 0(x9)
          add x10,x10,x12
          addi x9,x9,8
          addi x11,x11,1
          addi x13,x0,20
          blt x11,x13,Loop
```

Calculate sum of all elements of A

```
X8 and x9 contain base address of
arrays A and B respectively

          addi x11, x0, 1
          addi x12, x0, 100
Loop:     ld x1, 0(x8)
          ld x2, 8(x9)
          add x3, x1, x2
          sd x3, 8(x8)
          addi x8, x8, 8
          addi x9, x9, 8
          addi x11, x11, 1
          bne x11, x12, Loop
```

A[i] = A[i-1] + B[i];

```
x8 contains base address of array
A.

main:     addi x9, x0, 5
alpha:    ld x1, 0(x8)
          bge x1, x0, beta
          sub x1, x0, x1
          sd x1, 0(x8)
beta:     addi x9, x9, -1
          addi x8, x8, 8
          bge x9, x0, alpha
          jalr ra
```

Turn array items positive

```
          addi x18, x0, 3
          addi x19, x0, 5
          addi x10, x0, 1

Loop:     beq x19,x0,End
          sll x18,x18,x10
          addi x19,x19,-1
          jal ra, Loop
End:
```

'3' multiply by 20 [or 3 shift-left 6]