COSC 230, Spring 2023

Midterm 2

14 Apr 2023

Instructions

1. Make your responses legible. If the graders cannot read your answers clearly, then you may not receive credit for a response. **Only what you write inside of the provided boxes will be graded**.

2. Please remember that any cheating on an exam will result in a 0 for the course and a referral to the student conduct office.

3. Write your first name, last name, and net id as it appears on Canvas. The grades must be able to match your information with that in Canvas. Write your net id on each page in case one becomes lost.

4. Do not separate the exam pages.

5. Show your work. Most of these questions will require you to work out a problem. You must show your work to receive full credit for your response. The answer itself is only part of your grade.

First Name:

Last Name:

NetID:

COSC 230, Spring 2023
Midterm 2
14 Apr 2023

I. IEEE-754 (show ALL your work on this page)

1. Convert 13.5625F into IEEE-754 in hexadecimal.

```
13.5625F = 1101.1001 x 2^0
1101.1001 x 2^0 = 1.1011001 x 2^3
Sign = + (0)
Exponent = 3 + 127 = 130 = 1000_0010
Fraction = 1.1011001

0  1000_0010      1011001 0000000000000000
0100 0001 0101 1001 0000 0000 0000 0000
  4    1    5    9    0    0    0    0

 0x4159_0000
```

2. Convert the IEEE-754 value 0xbfd4_0000_0000_0000 into base 10.

```
0xbfd4_0000_0000_0000_0000
 1011 1111 1101 0100 0000 0000 0000 0000 0000 … 0000

[1] [01111111101] [0100 0000 0000 0000 0000 0000 … 0000

Sign = 1 (-)
Exponent = 01111111101 = 1021 - 1023 = -2
Fraction = 1.01

 -1.01 x 2^-2 = -0.0101 = -(0.25 + 0.0625) = -0.3125
```

II. Encoding Instructions

imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ

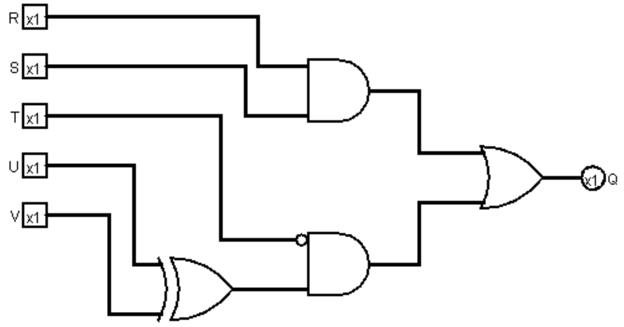
1. Encode `beq x7, x21, -16`

```
rs1 = 0b00111, rs2 = 0b10101, -16 = 1_1111_1111_0000
Bits 12 = [1], 11 = [1], 10:5 = [111111],  4:1 = [1000]
imm[12|10:5] = 1_111111, imm[4:1|11] = 1000_1
1_111111 10101  00111  000 1000_1  1100011
1111 1111 0101 0011 1000 1000 1110 0011
 F    F    5    3    8    8    E    3
```

II. Digital Logic

1. Draw the circuit diagram depicting the following circuit equation:

$$Q = RS + T'(U \oplus V)$$



2. Write the circuit equation given the following truth table:

A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Q = A’BC + AB’C + ABC

OR

Q = (A + B + C)(A + B + C’)(A + B’ + C)(A’ + B + C)(A’ + B’ + C)

III. Pipelining

1. Draw the 5-stage, RISC pipeline. Document each stage, what it does, and the functional unit(s) each stage uses.

IF – ID – EXE – MEM – WB uses IM – RF – ALU/FPU – DM – RF
IF – fetches an instruction from current PC.
ID – fetches source registers and/or sign extends immediates.
EXE – Calculates the requested operation, such as ADD/SUB/SLL/XOR/etc.
MEM – loads or stores values from/to data memory.
WB – writes the result into the destination register.

IV. Floating-point Assembly

1. Write the assembly that performs the following.

```
double myfunc(char op, double a, int b) {
   if (op) return a / b;
   else return a * b;
}
```

```
.section .text
.global myfunc
myfunc:
    # a0  - char op
    # fa0 - double a
    # a1  - int b
    fcvt.d.w  fa1, a1
    beqz      a0, 2f
    # If we get here, return a / b
    fdiv.d    fa0, fa0, fa1
    ret
2:
    # If we get here, return a * b
    fmul.d    fa0, fa0, fa1
    ret
```