Spring 2020 *Name*: David Pace ID:_____
CSCI 3301 Homework # 1

Due: **Tuesday March 09, 2020 (11:59 pm), via Moodle.**

The rules:

- All work must be your own. You are not to work in teams on this assignment. You are not to use materials from previous offerings of this course.
- Format: Submit as a single file (via moodle) containing a PDF file. Email me (ayn@cs.uno.edu) assignment only if moodle is not working.
- You may use the textbook and lecture notes, but do NOT search the Internet for solutions.
- The submission deadline is strict. Therefore, please submit on time.

Total Marks = 100

(1) [2 x 4=8 *points*]

(a) For the following C statement, what is the corresponding MIPS assembly code? Assume that the variables *f*, *g*, *h*, and *i* are given and could be considered 32-bit integers as declared in a C program. Use a minimal number of MIPS assembly instructions.

f = *g* + (*h* – 5);

addi $t0, $s1, −5 # $s1 = h
add $t0, $t0, $s0 # $s0 = g

(b) For the following MIPS assembly instructions below, what will be the corresponding possible minimal Java/C statement?

add *f*, *g*, *h*
add *f*, *i*, *f*

int f,g,h,i;
int f = g+h;
int f = i+f;

(2) [3 × 5=15 *points*] Assume the following register contents:

$t0 = 0xAAAAAAAA, $t1 = 0x12345678

(a) For the register values shown above, what is the value of $t2 for the following sequence of instructions?

```
sll $t2, $t0, 4
or $t2, $t2, $t1
```

0xbabefef8

(b) For the register values shown above, what is the value of $t2 for the following sequence of instructions?

```
sll $t2, $t0, 4
andi $t2, $t2, –1
```

0xaaaaaaa0

(c) For the register values shown above, what is the value of $t2 for the following sequence of instructions?

```
srl $t2, $t0, 3
andi $t2, $t2, 0xFFFFFFFF
```

0x1555555

(3) [15 *points*]

Assume $t0 holds the value 0x00101000. What is the value of $t2 after the following instructions? Explain.

```
slt $t2, $zero, $t0
bne $t2, $zero, ELSE
j DONE
ELSE: addi $t2, $t2, 2
DONE:
```

$t2 = 3

(4) **[**3 x 3 = 9 *points***] DO NOT** use calculator or converters from the Internet. Show your calculation results step by step. Note that 0x means HEX.

(a) Translate 0xabcdef12 into binary.

1010 1011 1100 1101 1111 0001 0010

(b) Translate 0xabcdef12 into decimal.

2882400018

(c) Translate 8985 into Hex.

2319

(5) [2 x 7 = 14 *points*]
(a) Provide the ***type***, ***assembly language instruction***, and ***binary representation*** of instruction described by the following MIPS fields:

op=0, rs=3, rt=2, rd=3, shamt=0, funct=34

Type	instr	op	rs	rt	rd	shamt	funct
R-format	sub	000000	00011	00010	00011	00000	010000

(b) Provide the ***type***, ***assembly language instruction***, and ***binary representation of instruction*** described by the following MIPS fields:

op=0x23, rs=1, rt=2, const=0x4

Type	instr	op	rs	rt	const
I-format	lw	100011	00001	00010	00100

(6) **[**3 × 7 =21 *points***]** Consider the following MIPS loop:

```
LOOP: slt $t2, $zero, $t1
      beq $t2, $zero, DONE
      addi $t1, $t1, -1
      addi $s2, $s2, 2
      j LOOP
DONE:
```

(a) Assume that the register $t1 is initialized to the value 10. What is the value in register $s2 assuming the $s2 is initially zero?
20

(b) For each of the loops above, write the equivalent *Java* / C code routine. Assume that the registers $s1, $s2, $t1, and $t2 are integers A, B, *i*, and temp, respectively.

```
Int b = 0;
for(int i = 10; 0 < i;  i-- ) {
    b += 2;
}
```

(c) For the loops written in MIPS assembly above, assume that the register $t1 is initialized to the value N. How many MIPS instructions are executed? **5N**

(7) [3 × 6 =18 *points***]** Assume that for a given program 70% of the executed instructions are arithmetic, 10% are load/store, and 20% are branch.

(a) Given the instruction mix and the assumption that an arithmetic instruction requires 2 cycles, a load/store instruction takes 6 cycles, and a branch instruction takes 3 cycles, find the average CPI.

$$CPI = \frac{cycles}{instructions} =$$

2 cycles/arithmetic
6 cycles/load or store
3 cycles/branch instr

Let x be the number of instructions
f(x) = .7x(2 CPI) + .1x(6 CPI) + .2x(3 CPI)
f(100) = 260 cycles total

So whats the average?

260 cycles/100 instru = 2.6 CPI on average for this program

(b) For a 25% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

$$260(1\text{-}0.25) = (.7x(2)/IF) + .1x(6) + .2x(3)$$
$$195 - .1x(6) - .2x(3) = .7x(2)/IF$$
$$IF = .7x(2) / (195 - .1x(6) - .2x(3))$$
$$= 1.87 \text{ cycles}$$

$$2CPI/1.87 \text{ cycles} = 1.07 \text{ CPI}$$

$$T_{improved} = \frac{T_{affected}}{improvement\ factor} + T_{unaffected}$$

(c) For a 50% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

$$260(1\text{-}0.50) = (.7x(2)/IF) + .1x(6) + .2x(3)$$
$$130 - .1x(6) - .2x(3) = .7x(2)/IF$$
$$IF = .7x(2) / (130 - .1x(6) - .2x(3))$$
$$= 14 \text{cycles}$$

$$2CPI/14 \text{ cycles} = 0.142 \text{ CPI}$$

0xabcdef12 → Decimal, Binary

4.) a.) a = 10 = 1010 = $2^3 + 2^1$

b = 11 = 1011 = $2^3 + 2^1 + 2^0$

c = 12 = 1100 = $2^4 + 2^3$

d = 13 = 1101 = $2^4 + 2^3 + 2^2$

e = 14 = 1110 = $2^4 + 2^3 + 2^2 + 2^1$ $\sum_{n=1}^{4}$

f = 15 = 1111 = $\sum_{n=0}^{4} 2^n$

b.) s = 0xabcdef12

where x is a decimal:

$f(x \in S) = \sum_{n=0}^{7} a_n \qquad a_n = 16^n \cdot x$

$(16^7 \cdot 10) + (16^6 \cdot 11) + (16^5 \cdot 12) + (16^4 \cdot 13) + (16^3 \cdot 14) + (16^2 \cdot 15) + (16^1 \cdot 1) + (16^0 \cdot 2)$

= 2882400018

c.) $f(d) = \{n \in \mathbb{N} : n/1 = \frac{d}{16^n}\}^4$ $\qquad g(d) = (d \% 16) \times 16$

$f(8985) = \frac{8985}{16^1} = 561$ $\qquad g(8985) = 0.5625 \times 16 = 9$

$= \frac{8985}{16^2} = 35$ $\qquad g(f(8985)) = 0.0625 \times 16 = 1$

$= 0.1875 \times 16 = 3$

$= \frac{8985}{16^3} = 2$ $\qquad = 0.125 \times 16 = 2$

$= \frac{8985}{16^4} = 0$

9319

2.) $t0 = 0xAAAAAAAA $t1 = 0x12345678

a.) 1010 1010 1010 1010 1010 1010 1010 1010

1010 1010 1010 1010 1010 1010 1010 1010

OR 0001 0010 0011 0100 0101 0110 0111 1000

$t4 = 1011 1010 1011 1110 1111 1110 1111 1010

= 0xbabefef8

b.) 1010 1010 1010 1010 1010 1010 1010 0000

1111 1111 1111 1111 1111 1111 1111 1110

1010 1010 1010 1010 1010 1010 1010 0000

= 0xA A A A A A A 0

c.) $t2 → 1010 1010 1010 1010 1010 1010 1010 1010

1111 0101 0101 0101 0101 0101 0101 0101

0x F 5 5 5 5 5 5