WASHINGTON STATE UNIVERSITY

School of Electrical Engineering and Computer Science

Introduction to Computer Architecture, Fall 2016

Midterm #1
October 5, 2016
Duration: 60 minutes

NAME: ID:

	Total Points	Earned
Problem 1	10	
Problem 2	20	
Problem 3	20	
Problem 4	20	
Problem 5	15	
Problem 6	15	

Notes:

- You may bring a double-sided letter-size cheat-sheet and a calculator to the test. No other resources are allowed! In particular, NO textbook, lecture notes, internet access, smartphone usage, etc. are allowed!
- Make sure to write your name and WSU ID down on all pages.
- You may use both sides of each page if needed!
- Show your work for each question. Even if you don't know exact answer to a question, show your work to get partial credit.
- MIPS reference data is provided on the last page of this test.

1. **(10 points).** For each sentence below, indicate if it is a *True* or *False* statement.

   a. A CPU with a faster clock (i.e., higher clock rate) always has higher performance compared to a computer one with a slower clock (i.e., lower clock rate).

   False

[redacted]

   d. Assuming a 2's complement representation, the hexadecimal number '0x$\boxed{0}$C00000' is a positive number.

   True, the MSB is '0'. $\rightarrow$ ($\boxed{0}$000)

   most significand bit

[redacted]

2. **(20 Points)** Consider the decimal number "63.25". Write down its binary representation using the IEEE 745 single precision format.

Single precision Format :

Sign	Exponent	Fraction
1bit	8 bits	23 bits

$$(-1)^{Sign} \times (1 + Fraction) \times 2^{(Exponent - \overset{127}{Bias})}$$

$63 = (111111)_2$ $\xleftarrow{\text{Integer part}}$ 63.25 $\xrightarrow{\text{Fraction Part}}$ $0.25 \times 2 = 0.5 \rightarrow 0$

$\parallel$ 111111.01 $\qquad$ $0.5 \times 2 = 1.\underline{0} \rightarrow 1$

done

normalization $\longrightarrow$ $1.1111101 \times 2^5 = (-1)^{\overset{\text{sign}}{0}} \times (1 + \underbrace{0.1111101}_{\text{Fraction}}) \times 2^{\overset{\text{Exp-Bias}}{5}}$

$Exp - \overset{127}{\cancel{Bias}} = 5 \Rightarrow Exponent = 132 = (10000100)_2$

$Fraction = 1111101\underbrace{0\ldots0}_{16\,bits}$

$S = 0$

$\Rightarrow 63.25 = 01000010011111010\underbrace{\cdots0}_{16\,bits}$

3. **(20 points)** Assume for a given program, 70% of the executed instructions are arithmetic, 10% are load/store, and 20% are branch.

a) Given this 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, compute he average CPI.

$$CPI_{avg} = \sum_{i=1}^{class\#} percent_i \; CPI_i$$

$$CPI_{avg} = \frac{70}{100} \times 2 + \frac{10}{100} \times 6 + \frac{20}{100} \times 3 = 0.7 \times 2 + 0.1 \times 6 + 0.2 \times 3$$

$$CPI_{avg} = 2.6 \; \frac{cycles}{Instructions}$$

b) How many cycles, on average, an arithmetic instruction should require to achieve 25% improvement in the overall performance, assuming that load/store and branch instructions are not improved at all.

$$Performance_{new} = 1.25 \times Performance_{old} \Rightarrow CPI_{old} = 1.25 \; CPI_{new}$$

$$\Rightarrow \frac{CPI_{old}}{CPI_{new}} = 1.25$$

$$\Rightarrow \frac{2.6}{0.7 \times CPI_A + \underbrace{0.1 \times 6 + 0.2 \times 3}_{1.2}} = 1.25 \Rightarrow 0.7\,CPI_A + 1.2 = 2.08$$

$$\Rightarrow 0.7\,CPI_A = 0.88 \Rightarrow CPI_A = 1.25 \simeq 1.2 \; \frac{Cycles}{Instructions}$$

4. **(20points).** You are asked to multiply two binary numbers using a sequential multiplier. These two binary numbers are 1010 and 10010. Show the value of different registers during each clock cycle in the following table.

Solution 1: 1010 → multiplier, 10010 → multiplicand

Solution 2: 10010 → multiplier, 1010 → multiplicand

	Cycle 1	**Cycle 2**	**Cycle 3**	**Cycle 4**	**Cycle 5**
9 bits ← **Product**	000000000	000000000	000100100	000100100	010110100
9 bits ← **Multiplicand**	0000 10010	000100100	001001000	010010000	100 100000
4 bits ← **Multiplier**	1010	0101	0010	0001	0000

	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle5	Cycle 6
9 bits ← product	000000000	000000000	000010100	000010100	000010100	010110100
9 bits ← multiplicand	00000 1010	000010100	000101000	001010000	0101 00000	101000000
5 bits ← multiplier	10010	01001	00100	00010	00001	00000

5. **(15 points)** Implement the following code in MIPS assembly. Assume variables 'm' and 'n' are stored in \$s0 and \$s1 respectively.

```
if (m>n)
    m = n;
else
    n = m;
```

Solution 1 (bne)

```
slt $t0, $s1, $s0
bne $t0, $zero, L1
add $s1, $s0, $zero
J Exit
L1: add $s0, $s1, $zero
Exit:
```

Solution 2 (beq)

```
slt $t0, $s1, $s0
beq $t0, $zero, L1
add $s0, $s1, $zero
J Exit
L1: add $s1, $s0, $zero
Exit:
```

6. **(15 points)** The following MIPS assembly code is expected to implement the 'for' loop mentioned below. There are several mistakes in the assembly code that you need to fix. Assume that the base address of array 'A' is in register \$s0. Rewrite the assembly code in the space provided below. Note that you will need to find each incorrect instruction and write the correct instruction with correct arguments for each.

for (i=0; i<100; i++)
{
A[i] = 256;
}
	The correct code is included below with correct instructions marked.

li \$t0, \$s0 X li \$t1, 100d sll \$t1, \$t1, 4 X add \$t1, \$t1, \$s0 ori \$t2, \$zero, 256d top: sltu \$t3, \$s0, \$t1 X beq \$t3, \$zero, done sw \$t2, 12(\$t0) X addi \$t0, \$t0, 1 X j top done:	la \$t0 , \$s0 li \$t1 , 100d sll \$t1, \$t1 , 2 add \$t1 , \$t1 , \$s0. ori \$t2, \$zero, 256d top: sltu \$t3, \$t0 , \$t1 beq \$t3 , \$zero , done sw \$t2 , 0(\$t0) addi \$t0 , \$t0 ,4 j top done: