Name: Michael Campos

Midterm Exam (A)

**CSE222: Computer Architecture & Organization**

(**March 2020, SCCC**)

**Part I (40):** **Write answers in this file and submit**

1. (**3**) (a) Give a brief description about how a machine instruction is executed in Von Neumann Architecture machine

First the PC (Program Counter) fetches the address of the memory location that has the next instruction. Then the address is copied to the MAR (Memory Address Register) via the address bus. After the address is copied, the MAR copies the contents located at the copied address and copies the contents to the MDR (Memory Data Register). Then the MDR copies the contents and stores it in the CIR (Current Instruction Register). The PC moves to the next instruction and fetches it. The instruction is finally decoded and then executed.

(**3**) (b) What is **big-endian** and what is **little-endian**? Given the following number **0x1324ACBD**, fill in the following table(memory) to indicate how this number is saved in memory as big- and little-endian:

Little-endian – **the least significant byte of data is placed at the byte with the lowest address**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | BD | CA | 42 | 21 |
| **Address**: | 0x1003 | 0x1002 | 0x1001 | 0x1000 |

Big-endian – **places the most significant byte first and the least significant byte last.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 13 | 24 | AC | BD |
| **Address**: | 0x1003 | 0x1002 | 0x1001 | 0x1000 |

1. (**12**) Convert numbers among different number systems:

( **1359** )10 = ( 10101001111 )2 = ( 54F )16

( 2859 )10 = ( 101100101011 )2 = ( **B2B** )16

( **357** )10 = ( 2A5 )11

( 425 )10 = ( **357** )11

1. (**7**) Convert numbers:
2. **10110110** is 8-bit two’s complement number. Convert it to decimal

-74

1. **101011** is a 6-bit two’s complement number. Expand it to 8-bit two’s complement number

1110 1011

1. Convert the following 2 decimal numbers to **8-bit** two’s complement binary numbers then perform addition. Write the result as a binary number

(**-57**) + (**39**) = ( 11000111 )2  + ( 00100111 ) 2 = ( 11101110 )2

1. (**10**) **-47.3125** is a decimal number, write this number in the formats as list below:

(You should write the results as hexadecimal number)

* + - * 1. 16-bit fixed-point sign/magnitude format with 8 integer bits and 8 fraction bits;

10111111.01010000

* + - * 1. 16-bit fixed-point 2’s complement format with 8 integer bits and 8 fraction bits;

11011000.10110000

* + - * 1. Single precision (32-bit) IEEE 754 floating-point format

0xC23D4000 = 1100 0010 0011 1101 0100 0000 0000 0000

1. (**5**) Give brief description of the following instructions:
2. **bne $s0, $s1, label**

Branch if not equal to. The program branches to label is $s0 does NOT equal to $s1.

1. **slt $t0, $t1, $t2**

Set less than or equal to. If $t1 is less than $t2 then $t0 will set equal to 1. If the opposite is true ($t2 < $t1) then $t0 will be set to 0. It’s a R-Type instruction.

1. **jal func**

Jump and link. It’s a J-Type instruction. Jumps to the label func and saves the whoever called the function to $ra.

1. **srav $s0, $s1, $s2**

Shift right arithmetic variable. Shifts the contents inside $s1 by X times (contents inside $s2) and stores the result to $s0. It’s a R-Type instruction.

1. **sw $s0, 4($a0)**

Save word. Saves the contents stored in $s0 to $a0 starting 4 bytes after where the memory address starts of $a0