Computer Architecture – Winter 2015

Assignment Two

**Name:** Julio Tain Sueiras\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Due Dates:** See Slate

**Instructions:**  This assignment is to complete individually

Show your work in the space provided where appropriate.

1. a) What is a memory Segment? \_as a way to divide memory into division and accessing them through segment plus the offset\_\_\_\_\_\_\_\_ **2 marks**

b) Describe one example of how memory segments is used in the 8086 processor.

8086 processor use segments by combing a 16 bit segment register and 16 bit offset register to access a 20 bit address space

**3 marks**

1. Explain briefly in your own words what is instruction encoding? Give at least one example of this.

Instruction encoding is the mapping of instruction set(MOV, ADD, etc) to the machine language, and this is done through hardware implementation in the design of the CPU processor itself, so each processor have different implementation of instruction encoding thus, have different instruction set, as in 8086 microprocessor(which uses intel asm dialect) map ADD or most instruction as 7 bit opcode

**10 marks**

1. List the two main components of a processor instruction and give a brief description of what each is used for.
2. The actual operation( a.k.a Opcode) in the left
3. the operands(which can be vary from no operands to multiple operands)

**4 marks**

1. a) What is meant by addressing mode? Different modes to access the registers/adress

**2 marks**

b) Give three examples of addressing modes in the 8086 processor

1. Direct Addressing
2. Registers Addressing
3. Base Index addressing

**3 marks**

1. How many 16 bit registers exist on the Intel 8086 processor? 8 registers

**1 mark**

1. Give the three classifications of the 8086 registers.
   * General Purpose Registers

* + Memory and IP Registers

* + Control and Status Registers

**3 marks**

1. Give the names of the accumulator registers. AX BX CX DX

**2 marks**

1. Give the names of the segment registers. CS DS ES SS

**2 marks**

1. Which two registers combine to give the effective address of an instruction that is stored in memory? Segment registers and Offset registers

**2 marks**

1. Give the effective address if the segment register is AB02 and the offset register is 0100.

AB120h

**1 mark**

1. a) How many bits are contained in the flags register? 16 bits

b) How many bits of the flags register are used? 1 bits for each flag register

**2 marks**

1. a) List five categories of instructions are implemented on the 8086 processor? Data transfer

Arithmetic

Logic

Control

String Manipulation

**5 mark**

1. The 32-bit value 30A79847 is stored starting at memory location 1000. Fill in the following tables.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Address | 10000 | 10001 | 10002 | 10003 |
| contents |  |  |  |  |

Little Endian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Address | 10000 | 10001 | 10002 | 10003 |
| contents |  |  |  |  |

Big Endian

**4 marks**

1. In class (and in the notes) we discussed a 6 step procedure in executing programs on a Von Neumann computer. List the steps.
   * Fetch the instruction from memory and place it in a register
   * Change PC(Program Counter, for indicating which instruction is processing) to point to following instruction
   * Check and Determine the type of instruction that just have been fetch
   * locate data in the memory if the instruction is using data,get the data and put it into a register in processor(only if the instruction uses data)
   * execute the instruction
   * go back to fetch for the next instruction.

**6 marks**

1. What is an assembler and why is it important to users?

A assembler is the component that translate the assembly code(instruction code) to machine code(processor readable code) and is important since is the bridge between processors component and users itself, if assembly code have not been through a assembler than it just a plain text that can't be executed to the processor

**2 marks**

1. Give the four ‘types’ of instructions that are required in assembler programming.
   * Register-format Operations
   * Memory R/W Operation
   * Immediate Operation
   * Control Operation(Branch/Jump)

**4 marks**

1. Write an 8086 assembler program (using the emulator) that will perform the following calculations. You should attach a printed copy of your code to this assignment.

a) ( 4 + 100) + ( 4 -5) + 35 Place the answer in the AX register

**7 marks**

b) 2010 \* ( 1004 \* 2) **6 marks** Where (in which registers) is the answer?

DX:AX registers **2 mark**

1. a) How many software interrupts are possible? 256

b) How many are currently supported by the emulator? 59

**2 marks**

1. Consider the following program:

org 100h

mov bx,0009h

mov ax,bx

back:

sub bx,01h

add ax,bx

cmp bx,00

jne back

hlt

Single step through the program and answer the following:

1. Complete the following table:

|  |  |  |
| --- | --- | --- |
| Instruction | Hex Value | Number of bytes |
| mov bx,0009h | BB 09 00 | 3 |
| mov ax,bx | 8B C3 | 2 |
| sub bx,01h | 83 EB 01 | 3 |
| add ax,bx | 03 C3 | 2 |
| cmp bx,00 | 83 FB 00 | 3 |
| jne back | 75 F6 | 2 |
| hlt | F4 | 1 |

**7 Marks**

1. Give the contents of the AX, BX and IP registers just before each execution of the JNE instruction: **For example:**  The first time the JNE instruction is highlighted the registers look like the following. You fill in the table:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **AX** | **BX** | **IP** |
| 1st | *0011* | *0008* | *010A* |
| 2nd | 0018 | 0007 | 010A |
| 3rd | 001E | 0006 | 010A |
| 4th | 0023 | 0005 | 010A |
| 5th | 0027 | 0004 | 010A |
| 6th | 002A | 0003 | 010A |
| 7th | 002C | 0002 | 010A |
| 8th | 002D | 0001 | 010A |
| 9th | 002D | 0000 | 010A |
| 10th |  |  |  |
| 11th |  |  |  |
| 12th |  |  |  |

**7 Marks**

1. How many times did the instruction JNE get executed? 9 time BX being the counter and compare it to 0 using CMP instruction

**1 Mark**

1. Using the emulator write a program that displays your name and the name of this course on the screen.

(Attach a printed copy of your source code to this assignment)

**10 Marks**