The New York City College of Technology – Computer Engineering Technology

CET 3510 Microcomputer System Technology Instructor: Prof. Y.Wang

Study and Review I

Q1 What are the main components *for the Von Neumann* computer systems?

Q2 What are the main components *for Central Processing Unit* (CPU) contains?

Q3 What are the main buses for computer system? What is the purpose of each bus inside the computer?

Q4 What is the data bus sizes of Pentium I ?

Q5 In order to compile a **Demo.hla** program under the Linux operating system, which compiler do you need to call?

Q6 In order to rename **Demo.hla** program to Demonstration.hla, which command is used under the Linux operating system? Write a Linux command for this.

Q7 (5 points) Inside the Control Unit, which register contains the instruction that is being executed?

Q8 (5 points) How many bits for each nibble, byte, word, and double word, respectively?

Q9 (5 points) How many hexadecimal digits for data types of char, byte, and word, respectively?

Q10 Computer uses each cycle per instruction. How many steps in each cycle? What are they?

Q11 A computer has 128 MB of memory. How many bits are needed to address any single byte in memory? If there are 8 bytes (namely c1:=$A1, c2 :=$AB, c3 :=$10, c4 :=$EF, c5=$AB12, and c6=FE01), stored at this memory, draw the table and place them (c1, c2, c3, c4, c5, and c6) at the memory with associated address (Little Endian computer. Assume the address for first variable c1 is 000EF110 H).

Q12 What is the 8-bit binary information, hexadecimal information for 4, 2, 0, -2, -4, respectively?

Q13 Computer register AL hold the binary information of "1111 1101", what is the *unsigned number* in decimal format?

Q14 Computer register AL hold the binary information of "1111 1101", what is the *signed number* in decimal format?

Q15 Suppose AX holds the hexadecimal value of 82FE H, what is the hexadecimal value for EAX after sign extends AX to EAX? And what is the hexadecimal value for EAX after zero extends AX to EAX?

Study and Review II

Q1 What are the main buses for computer system? What is the purpose of each bus inside the computer?

Q2 What are the main components *for Central Processing Unit* (CPU) contains?

Q3 In order to compile a **TEST.hla** program under the Linux operating system, which compiler do you need to call?

Q4 In order to copy Demonstration.hla program to **TestDemo.hla**, which command is used under the Linux operating system? Write a Linux command for this.

Q5 Inside the Control Unit, which register contains the instruction that is being executed?

Q6 How many hexadecimal digits for data types of char, byte, word, and double word, respectively?

Q7 Computer uses each cycle per instruction. How many steps in each cycle? What are they?

Q8 What is the 16-bit Hexadecimal information for 3 and -5, respectively?

Q9 Computer register AH hold the binary information of "1111 1100", what is the *unsigned number* in decimal format?

Q10 Computer register AL hold the binary information of "1111 1100", what is the *signed number* in decimal format?

Q11 A computer has 512 MB of memory. How many bits are needed to address any single byte in memory? If there are 8 bytes (namely c1:=$A1, c2 :=$AB, c3 :=$10EF, c4 :=$EF, c5=$AB12, and c6=FE01), stored at this memory, draw the table and place them (c1, c2, c3, c4, c5, and c6) at the memory with associated address (Little Endian computer. Assume the address for first variable c1 is 000EF000 H).

Q12 Suppose AX holds the hexadecimal value of 82FE H,

(1) what is the hexadecimal value for EAX after sign extends AX to EAX? Write HLA x86 Intel instruction to sign extend AX to EAX
(2) And what is the hexadecimal value for EAX after zero extends AX to EAX? Write HLA x86 Intel instruction to zero extend AX to EAX

Q13 A single precision floating point number can be represented in hexidecimal format

A8200000.

(1) How many total bits needed to represent a single precision floating point number?
(2) Howmany bits are needed to represent the sign?
(3) Howmany bits are needed to represent the integer part?
(4) Howmany bits are needed to represent the fraction part?
(5) This floating point number is positive or negative? Why?
(6) What is a biased exponent for this floating point number?

Q14 Write a program that generates a "Powers of Four" table, where the exponent will be 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10, namely 4^0, 4^1, 4^2, 4^3, 4^4, 4^5, 4^6, 4^7, 4^8, 4^9, 4^{10}. Following flow control can be used in power of four table generation

(1) Use **jmp with label**
(2) while….endwhile
(3) for… endfor

Q15 The Stack is a LIFO Data Structure. The 80x86 controls its stack via the ESP (stack pointer) registers. The 80x86 decrements the stack pointer by the size of the data you are pushing (PUSH instruction) and then it copies the data to memory where ESP is then pointing. POP instruction copies the data from memory location ESP before adjusting the value in ESP. Study the following example and understand ESP

```
program  stk;
#include ("stdlib.hhf");

static
   input32_1: int32 :=600;
  input32_2: int32 :=700;
  input32_3: int32 :=800;
    value32:  int32;       //to retrieve value from memory address
  count: uns8;  //to count the number of memory address push in the stack
  LoopCntr: uns8;        //loop counter used in for loop

begin stk;
   stdout.put("Stack Structure", nl);
   stdout.put("The address for stack pointer: ", ESP, nl);   //show the address for Stack
    push(input32_1);
    stdout.put("The address for stack pointer: ", ESP, nl);
    mov( 1, count);
    push(input32_2);
    stdout.put("The address for stack pointer: ", ESP, nl);
    inc( count);
     push(input32_3);
     stdout.put("The address for stack pointer: ", ESP, nl);
     inc( count);

     mov(count, BL);
     mov( 0, LoopCntr );
    for ( mov( 0, LoopCntr ); LoopCntr < BL; inc(LoopCntr) ) do
            stdout.put("The address for stack pointer: ", ESP,nl);
            pop(EAX);
            mov(EAX, value32);
            stdout.put("The value pop from stack is ",value32,  nl);
     endfor;
end stk;
```

Q16. Review the **lea (mem, EAX)** instruction. Recall the declared variable and its memory address. Practice register addressing mode, the direct addressing mode (memory addressing mode), and the indirect addressing modes. Understand the difference between EAX and [EAX]. Understand the different addressing mode for
1) mov([ebx], al); 2) mov ($A5, AL); 3) mov(BL, AL)
4) mov($1234_5678, ebx);
mov([ebx], al);

Q17. Call linux.ioperm() to get permission to access the ports
Use in() to read from the port (check x86 Instructions)
Use out() to write to the port (check x86 Instructions)
#include("linux.hhf") provides the library for the functions of linux.ioperm()
#include("stdlib.hhf") provides the library for the functions of stdin.get() and stdout.put()

1. The function ioperm (...) is Linux specific. It allows an application program to access to the 80x86 I/O-addresses within the range 0x000 to 0x3ff.

2. in(DX, AL) to read date from the port.These instructions read a byte from an input port and place the input date into the accumulator register AL.
Must use DX register to hold the 16-bit port number.
Example, (1) mov($379, DX), in($379, AL); (2) mov ($61, DX), in($DX, AL)

3. out(AL, DX) provides a HLA language interface to the machine instruction that outputs the accumulator to the specified port (i.e., I/O port) using the I/O address space instead of the memory address space. Must use DX register to hold the 16-bit port number
Example, (1) mov($379, DX), in($379, AL); (2) mov ($61, DX), in($DX, AL)