جامعة بنـها – كلية العلوم – قسم الرياضيات

مادة من المستوي الثالث (علوم حاسب)

يـوم الامتحان: الاثنين

تـاريخ الامتحان: 4 / 1 / 2016 م

المادة : معالجات دقيقة (325 رس)

الممتحن: د/ مصعب عبد الحميد محمد حسان

مدرس بقسم الرياضيات بكلية العلوم

الاسئلة و نموذج الإجابة

ورقة كاملة

Answer the following questions:

Question 1. (13 marks)

A- Define registers, cach memory, data bus. (2 marks)

B- Discuss 8086, 80286, 80386, and 80486 processors. (4 marks)

C- List and Discuss segment registers. (4 marks)

D- Show the purpose of EAX, EBX, ECX, EDX, and EBP registers.(3 marks)

Question 2. (15 marks)

A- Name the four basic parts of an assembly language instruction.(2 marks)

B- List the rules of MOV Instruction. (2 marks)

C- Create an uninitialized data declaration for an 8-bit unsigned integer and 32-bit signed integer. (2 marks)

D- Use the following data definitions (6 marks)

```
oneByte     BYTE     27h;
oneWord     WORD     1100h;
oneDword    DWORD    35126579h ;
arrayB      BYTE     60h, 10h, 20h, 50h, 90h
arrayW      WORD     200h, 500h, 100h
arrayD      DWORD    30000h, 20000h, 50000h
myWords WORD 3 DUP(?),2000h
myString BYTE "ABCDE"
```

What will be the value of EAX after each of the following instructions execute?

(a) mov eax, 0; (b) mov al, oneByte; (c) mov ax, oneWord;
(d) mov eax, oneDword; (e) mov al, 0 ; (f) mov al, [arrayB+3] ;
(g) mov ax, [arrayW+4] ; (h) mov eax, [arrayD+8] ;
(i) mov eax, LENGTHOF myWords (j) mov eax, SIZEOF myString

E- What the meaning of the following statements (3 marks)

(a) array1 WORD 20 DUP(?),0,0 ;
(b) array2 WORD 6 DUP(2 DUP(?)) ;
(c) ALIGN 2;
(d) movzx eax, 1269h;
(e) mov edx, OFFSET buffer;

Question 3. (20 marks)

A- Discuss the nested loops of assembly language . (4 marks)

B- Write assembly program to execute the following arithmetic expression

Rval = - Xval + (Yval - Zval); (4 marks)

C- Write assembly program to display a graphical popup message box with Yes and No buttons. (4 marks)

D- Write assembly program that copies a string from source string to target string? (4 marks)

E-Write assembly program to sum elements of 32-BitInteger array(4 marks)

Model Answer

Answer of Question 1

A- registers are some internal memory storage locations in processor to speed up the processor operations.

cach memory: is a type of memory used to hold frequently used data. Cache memory is relatively small, but very fast

data bus transfers instructions and data between the CPU and memory

B-

Processor	Speed	Data Bus Width	Memory Size
8086	10MHZ	16	1M
80286	12.5MHZ	16	16M
80386	40MHZ	32	4G
80486	100MHZ	32	4G + 8K cache

C-

Segment Registers

Segments are specific areas defined in a program for containing data, code and stack. There are three main segment registers

1- CS register (code segment register) stores the starting address of the code segment (contains all the instructions to be executed). Instruction pointer register is used to determine the offset

2- DS register (data segment register) stores the starting address of the data segment

3- SS register (Stack segment register) stores the starting address of the stack segment (holds local variables and functions parameters). ESP register is used to determine the offset

D-

EAX is automatically used by multiplication and division instructions. It is often called the *extended accumulator* register.

EBX (base register) used in indexed addressing

ECX (count register) store the loop count in iterative operations

EDX (data register) used in I/O operations

EBP is used by high-level languages to reference function parameters and local variables on the stack. It should not be used for ordinary arithmetic or data transfer except at an advanced level of programming. It is often called the *extended frame pointer* register.

Answer of Question 2

A- An instruction contains four basic parts:

- **-Label (optional)**
- **-Instruction mnemonic (required)**
- **-Operand(s) (usually required)**
- **-Comment (optional)**

This is the basic syntax:

[*label*:] *mnemonic* [*operands*] [;*comment*]

B- MOV is very flexible in its use of operands, as long as the following rules are observed:

-Both operands must be the same size.

-Both operands cannot be memory operands.

-CS, EIP, and IP cannot be destination operands.

-An immediate value cannot be moved to a segment register.

Here is a list of the general variants of MOV, excluding segment registers:

MOV *reg, reg*
MOV *mem, reg*
MOV *reg, mem*
MOV *mem, imm*
MOV *reg, imm*

C-val1 BYTE ?
val2 SDWORD ?

D-

a- EAX=00000000
b- EAX=00000027
c- EAX=00001100
d- EAX=35126579
e- EAX=35126500
f- EAX=35126550
g-EAX=35120100
h-EAX=00050000
i- EAX=00000004
j- EAX=00000005

E-

a- array of word type and its length is 22
b- array of word type and its length is 12
c- the next variable is aligned on an even-numbered address
d- EAX = 00001269
e- The OFFSET operator returns the offset of a data label. The offset represents the distance, in bytes, of the label from the beginning of the data segment.

Example:

.data

```
  buffer BYTE ?
  buffer1 WORD ?
  buffer2  DWORD ?
  mov edx,OFFSET buffer ;   EDX = 00404000
  mov edx,OFFSET buffer1 ; EDX = 00404001
  mov edx,OFFSET buffer2 ; EDX = 00404003
```

Answer of Question 3

A-

Nested Loops **When creating a loop inside another loop, special consideration must be given to the outer loop counter in ECX. You can save it in a variable:**

```
.data
count DWORD ?
.code
mov ecx,100 ; set outer loop count
L1:
mov count,ecx ; save outer loop count
mov ecx,20 ; set inner loop count
L2:
.
.
loop L2 ; repeat the inner loop
mov ecx,count ; restore outer loop count
loop L1 ; repeat the outer loop
```

As a general rule, nested loops more than two levels deep are difficult to write. If the algorithm
you’re using requires deep loop nesting, move some of the inner loops into subroutines.

B-

```
TITLE arithmetic expression :  Rval = - Xval + (Yval - Zval);
INCLUDE Irvine32.inc
.data
Rval SDWORD ?
Xval SDWORD 26
Yval SDWORD 30
Zval SDWORD 40
.code
main PROC
; first term: -Xval
mov eax,Xval
neg eax ; EAX = -26
; second term: (Yval - Zval)
mov ebx,Yval
sub ebx,Zval ; EBX = -10
; add the terms and store:
add eax,ebx
```

```
mov Rval,eax ; -36
exit
main ENDP
END main
```

C-

```
TITLE MsgBoxAsk
INCLUDE Irvine32.inc
.data
caption BYTE "Survey Completed",0
question BYTE "Thank you for completing the survey."
BYTE 0dh,0ah
BYTE "Would you like to receive the results?",0
.code
main PROC
mov ebx,OFFSET caption
mov edx,OFFSET question
call MsgBoxAsk
;(check return value in EAX)
exit
main ENDP
END main
```

D-

```
TITLE Copying a String (CopyStr.asm)
INCLUDE Irvine32.inc
.data
source BYTE "This is the source string",0
target BYTE SIZEOF source DUP(0)
.code
main PROC
mov esi,0 ; index register
mov ecx,SIZEOF source ; loop counter
L1:
mov al,source[esi] ; get a character from source
mov target[esi],al ; store it in the target
inc esi ; move to next character
loop L1 ; repeat for entire string
exit
main ENDP
END main
```

E-

```
ArraySum PROC
;
; Calculates the sum of an array of 32-bit integers.
; Receives: ESI = the array offset
; ECX = number of elements in the array
; Returns: EAX = sum of the array elements
;--------------------------------------------------
push esi ; save ESI, ECX
push ecx
mov eax,0 ; set the sum to zero
L1: add eax,[esi] ; add each integer to sum
add esi,TYPE DWORD ; point to next integer
loop L1 ; repeat for array size
pop ecx ; restore ECX, ESI
pop esi
ret ; sum is in EAX
ArraySum ENDP
```