Philadelphia University
Faculty of Engineering

Marking Scheme

Examination Paper
Department of CE

Module: Microprocessors (630313)

Final Exam  Second Semester  Date: 08/06/2014
Section 1
Weighting 40% of the module total

Lecturer:  Dr. Qadri Hamarsheh
Coordinator:  Dr. Qadri Hamarsheh
Internal Examiner:  Eng. Anis Nazer
Marking Scheme
Microprocessors (630313)

The presented exam questions are organized to overcome course material, the exam contains 6 questions; all questions are compulsory requested to be answered. Thus, the student is permitted to answer any question out of the existing ones in this section.

Marking Assignments

The following scheme shows the marks assignments for each question. They show also the steps for which a student can get marks along the related procedure he/she achieves.

**Question 1**
This question is attributed with 10 marks if answered properly
The answer for this question as the following:

1) Since the x86 has an address bus of 20-bits, its memory is segmented into 1 M segments (i.e. 2^20).
   a) True b) False

2) If CS = 7FA2H, SS = 0801H, SI = 0100H and IP = 438EH the address of the next instruction is:
   a) 83DAEH  b) 438EH  c) 83DA0H  d) None of the above

3) One of the following is not a valid segment address
   a) 00000   b) E0840   c) 8CE90   d) 8CE91

4) Pipelining improves CPU performance due to
   a) increased clock speed  b) the introduction of parallelism  c) reduced memory access time  d) additional functional units

5) The basic parts of an instruction, in order from left to right, are:
   a) label, mnemonic, operand(s), comment
      b) comment, label, mnemonic, operand(s)
      c) label, mnemonic, comment
      d) mnemonic, operand(s), comment

6) Which of the following will generate assembly errors?
   a) var1 BYTE 1101b, 22, 35
      b) var3 BYTE 'S', '98778',
      c) var2 BYTE "ABCDE", 18
      d) None of the above

7) Which directive is used when defining BCD numbers?
   a) DWORD    b) FWORD
      c) QWORD    d) TBYTE

8) Given that the BL register contains ‘B’, the effect of the following instruction
   OR BL, 0010 0000
   is to
   a) clear BL  b) store 0010 0000 in BL  c) store ‘b’ in BL  d) leave BL unchanged

9) What is the result in AL after executing the following instructions?
   XOR AL, AL
   OR AL, 80H
   a) 80H b) 88H  c) 00H  d) None of the above

10) Which of the following are performed when an interrupt occurs:
    (I) FLAGS register is pushed to the stack
    (II) CS register is pushed to the stack
    (III) IP register is pushed to the stack
    a) (I) and (II) and (III)  b) (I) and (II) only
        c) (II) and (III) only  d) (I) and (III) only
Question 2 This question is attributed with 8 marks if answered properly

Question 2.a (3 marks)

A 3X8 decode is used to decode the instruction. The block diagram of the ALU is shown in figure

It performs the operation as: \( C = A \text{ op } B \)

Question 2.b (1.5 marks)

Using TYPEDEF

- The TYPEDEF operator creates a user-defined type.
- TYPEDEF is ideal for creating pointer variables.
- Syntax:
  - `name TYPEDEF type`
- For example, PBYTE and PWORD are pointers to bytes and words, respectively:
  ```
  PBYTE TYPEDEF PTR BYTE
  PWORD TYPEDEF PTR WORD
  ```

```
.data
arrayB BYTE 10h,20h,30h,40h
ptr1 PBYTE ?
ptr2 PBYTE arrayB
```

Question 2.c (1.5 marks)

**NOT**: Logical inversion or the one’s complement and **NEG**: arithmetic sign inversion or the two’s complement.

**AND**: Performs the AND operation and changes the destination operand.

**TEST**: Test instruction performs the AND operation and it wont changes destination operand but it only affects the condition of the flag register.

**SUB**: Performs the subtraction operation and changes the destination operand.

**CMP**: Comparison instruction is a subtraction that changes only the flag bits; the destination operand never changes.
**Question 2.d**

**Solution**

**Software Interrupt - Internal - from int or into**
- The INT instruction executes a software interrupt.
- The code that handles the interrupt is called an interrupt handler.
- The Interrupt Vector Table (IVT) holds a 32-bit segment-offset address for each possible interrupt handler.
- Interrupt Service Routine (ISR) is another name for interrupt handler.

**Hardware Interrupt - External Uses INTR and NMI**
- Generated by the Intel 8259 Programmable Interrupt Controller (PIC) in response to a hardware signal

**Interrupt Control Instructions**
- STI – set interrupt flag
- CLI – clear interrupt flag

---

**Question 3** This question is attributed with 6 marks, if answered properly.

**Question 3.a**

**Solution**

```
MOV EAX,L-4 ; EAX=5
MOV EBX,L+4 ; EBX=3
MOV ECX,L+12 ; ECX=6
```

**Question 3.b**

**Solution**

```
mov ax, J
cmp ax, K
jne DoElse
inc L
jmp ifDone

DoElse:
    dec L

ifDone:
```

**Question 3.c**

**Solution**

```
WhlLoop:    mov ax, j
            cmp ax, k
            jnae QuitLoop
            dec j
            inc k
            jmp WhlLoop

QuitLoop:   exit
            Main endp
```

**Question 3.d**

**Solution**

```
mov cx, 100
mov si, OFFSET C ; si = address of C array
Check:
cmp BYTE PTR [si], '$' ; if [si] == '$'
je HasDollar ; then go to HasDollar
inc si ; si = si + 1
dec cx
    jnz Check
```
Question 4

This question is attributed with 5 marks, if answered properly.

The complete code for this question as the following:

<table>
<thead>
<tr>
<th>Solution</th>
</tr>
</thead>
</table>
| INCLUDE Irvine32.inc  
.data  
Uarray WORD 1000h,2000h,3000h,4000h  
Sarray SWORD -1,-2,-3,-4  
.code  
main PROC  
; Move with zero extension:  
    movzx eax,Uarray  
    movzx ebx,Uarray+2  
    movzx ecx,Uarray+4  
    movzx edx,Uarray+6  
    call DumpRegs  
; Move with sign extension:  
    movsx eax,Sarray  
    movsx ebx,Sarray+2  
    movsx ecx,Sarray+4  
    movsx edx,Sarray+6  
    call DumpRegs  
exit  
main ENDP  
END main |

Question 5

This question is attributed with 5 marks, if answered properly.

The answer for this question as the following:

<table>
<thead>
<tr>
<th>Solution</th>
</tr>
</thead>
</table>
| INCLUDE Irvine32.inc  
COUNT = 50  
.data  
commaStr BYTE ",",0  
.code  
main PROC  
call Clrscr  
call Randomize  
mov  ecx,COUNT  
L1:  
    mov  eax,41  
    call RandomRange  
    sub  eax,20  
    call WriteInt  
    call WriteString  
loop L1  
exit  
main ENDP  
END main |

Question 6

This question is attributed with 6 marks, if answered properly.

The answer for this question as the following:

<table>
<thead>
<tr>
<th>Solution</th>
</tr>
</thead>
</table>
| TITLE (PrnEncStr.asm)  
.MODEL small  
INCLUDE Irvine16.inc  
BUFMAX = 200  
.data  |
sPrompt1 BYTE "Enter the plain text: ",0
sPrompt2 BYTE "Enter the Key string: ",0
bufSize WORD ?
buffer BYTE BUFMAX+1 DUP(0)
KEY BYTE BUFMAX+1 DUP(0)

.code
main PROC
mov dx,OFFSET sPrompt1
call WriteString
mov cx, BUFMAX
mov dx,offset buffer
call readstring
mov bufSize, ax
mov dx,OFFSET sPrompt2
call WriteString
mov cx, BUFMAX
mov dx,offset KEY
call readstring
cmp ax, bufSize
jnz finish
mov cx,bufSize
mov si,0
L1: mov al, buffer[si] ; get character
    xor al, KEY[si]
    inc si
    loop L1
finish:
.exit
main ENDP
END main