

EE-229: Computer Organization and Assembly Language

Serial No:

Sessional Exam 1

Total Time: 1 Hour

Total Marks: 35

Wednesday, 14th October, 2020

Course Instructors

Ameen Chilwan, Shams Farooq, Farwa Batool,
Rohail Gulbaz.

Signature of Invigilator

Student Name

Roll No

Section

Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. Use the back of the last page for rough work.
3. If you need more space write on the back side of the paper and clearly mark question and part number etc.
4. After asked to commence the exam, please verify that you have SIX (6) different printed pages including this title page. There are a total of 5 questions.
5. Calculator sharing is strictly prohibited.
6. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.

| | Q-1 | Q-2 | Q-3 | Q-4 | Q-5 | Total |
|-------------------|-----|-----|-----|-----|-----|-------|
| Marks Obtained | | | | | | |
| Total Marks | 8 | 5 | 8 | 6 | 8 | 35 |

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2020

Islamabad Campus

Question 1 [8 Marks]

Given that we have a 16-bit architecture with FLAGS register given as follows:

| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|----|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | NT | IOPL | | OF | DF | IF | TF | SF | ZF | 0 | AF | 0 | PF | 1 | CF |

Assuming that all system and control flags are set to zero after each arithmetic operation, find the values (in hexadecimal) saved in the FLAGS register after the following operations. It should be noted that the numbers are presented as signed 2's complement integers.

a. 8102h + FFFEh [2 Marks]

$$\begin{array}{r} 1000\ 0001\ 0000\ 0010 \\ +\ 1111\ 1111\ 1111\ 1110 \\ \hline 1\ 1000\ 0001\ 0000\ 0000 \end{array} \quad \text{AC}=1, \text{P}=1, \text{Z}=0, \text{S}=1, \text{C}=1, \text{O}=0$$

FLAGS = 0000 0000 1001 0111 (0097h)

b. 15A0h + 8547h [2 Marks]

$$\begin{array}{r} 0001\ 0101\ 1010\ 0000 \\ +\ 1000\ 0101\ 0100\ 0111 \\ \hline 0\ 1001\ 1010\ 1110\ 0111 \end{array} \quad \text{AC}=0, \text{P}=0, \text{Z}=0, \text{S}=1, \text{C}=0, \text{O}=0$$

FLAGS = 0000 0000 1000 0010 (0082h)

c. Write an assembly code that finds even parity for the most significant BYTE of si register, where si register is a 16-bit register. Write code after given lines that will update parity bit in the FLAGS register. [4 Marks]

```
mov si,0F798H
add si,01234h
```

```
.code
mov si, 0F798h      ;Saves the given value in si register
mov ax, si          ;Saves the word in si into register ax
mov bl, ah           ;Saves the most significant BYTE in bl
add bl, 00h          ;This will set all flags in the FLAGS register
                    ;including the parity flag
```

Question 2 [5 Marks]

List the basic components of a computer system and describe their functions and interconnection. Suppose the system is 32-bits. Briefly describe the effect of 32 bits architecture on the functionality of basic components. [5 marks]

There are three basic components of a computer system;

1. Processor:

This is the brain of the computer. It controls all the components and operations of the computer system. In turn, the processor is also divided into three parts: The Control Unit; that controls all the operations, the Arithmetic and Logical Unit; that performs the operations, and the Registers; where operands and results are stored.

2. Memory:

Memory refers to the main memory which is composed of RAM, ROM and, to some extent, Cache.

3. Input/Output:

These are devices used for interfacing with the computer users. These include keyboard, mouse, monitor, printer, joystick etc. These devices are controlled by an I/O controller that is connected to the rest of computer components via buses.

4. Interconnect:

Different components of the computer system are interconnected using buses. There are at least three types of buses: the Data Bus; which transfers data between different computer components, the Address Bus; which identifies which memory location or I/O device needs to be read from or written into, and the Control Bus; that communicates the control messages, e.g. whether data bus needs to be read from or written into.

Question 3 [8 Marks]

Implement following C++ code using LOOP statement in Assembly. Update the final value of SI after execution of the program.

```
int si=0;

for(int a=4;a>0;a--)
{
    for(int b=10;b>0;b=b-2)
    {
        for(int d=3;d>0;d--)
        {
            si++;
        }
    }
}
```

```
.data
varA DWORD    ?
varB DWORD    ?
varD DWORD    ?
vSI  DWORD    ?
```

```
.code
mov  vSI, 0
mov  varA, 4
mov  varB, 10
mov  varD, 3
```

LoopA:

LoopB:

```
LoopD:
mov  ecx, varD
add  vSI, 1
loop LoopD
```

```
mov  ecx, varB
sub  varB, 2
loop LoopB
```

```
mov  ecx, varA
sub  varA, 1
loop LoopA
```

SI = 60

Question 4 [6 Marks]

Write an assembly program that copies contents of *string1* to *copystring* in reverse order.

copystring should look like: 'ssergorp ni si mretdim ruoy'

```
string1 db 'your midterm is in progress',0
```

```
copystring db LENGTHOF string1 DUP('a')
```

```
.data
```

```
    string1 db 'your midterm is in progress',0
```

```
    copystring db lengthof string1 dup('a')
```

```
.code
```

```
    mov ax,@data
```

```
    mov ds,ax
```

```
    mov ax,0          ; zeroing ax registers
```

```
    mov si,offset string1
```

```
    mov CX, LENGTHOF STRING1
```

```
    dec cx
```

```
    mov si,0
```

```
    mov di,cx
```

```
l1:
```

```
    mov bl,string1[si]
```

```
    mov copystring[di],bl
```

```
    inc si
```

```
    dec di
```

```
loop l1
```

```
    mov copystring[di],byte ptr '\0'
```

Question 5 [8 Marks]

Consider the following data declaration in hexadecimal (h). Fill in the given memory:

```
.data
    word2      dw      -1
    list1      BYTE    1,2
    quad1      dq      0123456789ABCDEFh
    list2      db      10, 041h, 'A', 00111111b
    string     BYTE    'ABC',0
    list4      WORD    2 DUP(0AB12h)
    quad3      QWORD   'AB'
```

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0000 | FF | FF | 01 | 02 | EF | CD | AB | 89 | 67 | 45 | 23 | 01 | 0A | 41 | 41 | 3F |
| 0010 | 41 | 42 | 43 | 00 | 12 | AB | 12 | AB | 42 | 41 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0020 | | | | | | | | | | | | | | | | |

ROUGH SPACE