COE 205 Dr. Aiman El-Maleh

HW#1

Q1 The instruction set architecture (ISA) is the instruction set, the machine's memory, and all of the programmer-accessible registers in the computer.

Q2 - The program counter (PC) or the instruction pointer (IP) is the register in the CPU that holds the address of the next instruction to be fetched from memory

- The instruction register (IR) is the register in the CPU that stores the machine language instructions, temporarily, after the instructions are fetched from memory.

- In the fetch-execute process, the CPU takes the address stored in the program counter and reads from memory the instruction stored at that address. The instruction read from memory is stored in the instruction register. The program counter is then incremented to point to the next instruction to be fetched from memory. Then, the CPU executes the instruction stored in the instruction register. This process is performed repeatedly until the machine is halted.

Q3 Advantages of programming in assembly:

1. Machine designers need to know and understand the instruction sets of various machines to view the trade-offs implicit in a new machine design.

2. Compiler writers need to be intimately familiar with all the details of the machine languge.

3. Compilers do not always generate optimum code.

4. To program small embedded controllers in toys, appliances and cameras.

Advantages of programming in high-level languages:

1. Programs are easier to understand, write, and maintain.
2. Programs are portable, i.e. they can run on different machines.
3. High-level languages have rich data type structures that help prevent the programmer from making errors due to misuse of languge constructs.

Q4 (i) 250.375

$= (1111\ 1010.0110)_2 = (372.3)_8$

$= (FA.6)_{16}$

(ii) 4444.4

$= (1\ 0001\ 0101\ 1100.011)_2$

$= (10534.315)_8$

$= (115C.666)_{16}$

Q5 (i) $(10E)_{16} + (13F)_{16}$

$$\begin{array}{r} 10E \\ 13F \\ \hline 24D \end{array} \qquad \begin{array}{r} 270 \\ +319 \\ \hline 589 \end{array}$$

$(24D)_{16} = 13 + 4 \times 16 + 2 \times 16^2 = 589$

(ii) $(1E)_{16} * (10)_{16}$

$$\begin{array}{r} 1E \\ \times\ 10 \\ \hline 00 \\ 1E \\ \hline 1E0 \end{array} \qquad\qquad \begin{array}{r} 30 \\ \times\ 16 \\ \hline 180 \\ 30 \\ \hline 480 \end{array}$$

$$(1E0)_{16} = 14 \times 16 + 16^2 = 480$$

(iii) $(1101)_2 * (1000)_2$

$$\begin{array}{r} 1101 \\ \times\ 1000 \\ \hline 0000 \\ 0000 \\ 0000 \\ 1101 \\ \hline 1101000 \end{array} \qquad\qquad \begin{array}{r} \scriptstyle 2 \\ 13 \\ \times\ 8 \\ \hline 104 \end{array}$$

$$(1101000)_2 = 2^3 + 2^5 + 2^6 = 8 + 32 + 64 = 104$$

Q6 (i) -119

$+119$: 0111 0111

Sign-magnitude	1's complement	2's Complement
1111 0111	1000 1000	1000 1001

(ii) -55

$+55$: 0011 0111

Sign-magnitude	1's Complement	2's Complement
1011 0111	1100 1000	1100 1001

Q7 +120 : 0111 1000

(i) 8 bits : 1000 1000

(ii) 16 bits : 1111 1111 1000 1000

Q8 (i) 0101 + 1111

- Sign magnitude :

This is equivalent to +5 − 7 = −2

```
  111
- 101
-----
- 010
```

So, the result will be 1010.

There is no overflow because the effective operation is subtraction.

- 2's complement :

```
   111
   0101   (+5)
 + 1111   (−1)
 ------
   0100   (+4)
```

① ⊕ ① = 0 So, there is no overflow.

(ii) 1011 − 0111

- Sign magnitude:

This is equivalent to −3 −7 = −10

```
   11
   011
 + 111
 -----
①  010
```

So, there is overflow since there is a carry out from adding the magnitudes.

- 2's complement :

1011 − 0111 = 1011 + 1001

```
   1 1
   1011   (−5)
 + 1001   (−7)
 ------
①  0100
```

① ⊕ ⓪ = 1 So, there is overflow.

Q9 8-bits

(i) Maximum unsigned number:

$= 2^8 - 1 = 255 = (1111\ 1111)_2$

(ii) Signed-magnitude notation:

Largest positive number $= 2^7 - 1 = +127$
$= (0111\ 1111)_2$

Smallest negative number $= -127 = (1111\ 1111)_2$

(iii) 2's complement notation:

Largest positive number $= +127 = (0111\ 1111)_2$

Smallest negative number $= -128 = (1000\ 0000)_2$

Q10

C	O	E	2
1100 0011	1100 1111	1100 0101	1011 0010

∅	5
0011 0000	0011 0101

Q11 The message is : Attack at Dawn

Q12 To convert the ASCII code of a decimal digit to a number we need to subtract from it 30H, i.e. the ASCII code of character '0'.