COE 301/ICS 233, Term 172

Computer Architecture & Assembly Language

HW# 1 Solution

Q.1. Briefly describe the main functionality of the program counter register (PC), the instruction register (IR), and the fetch-execute process in a computer.

Program counter register: is the register in the CPU that holds the address for the next instructor to be fetched from memory.

Instruction register: is the register in the CPU that stores the machine language instructions, temporarily, after the instructions are fetched from memory.

Fetch-execute process: 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. Execution of the instruction includes decoding the instruction, getting the operands, performing the instruction operation and storing the result back. The process is performed repeatedly until the machine is halted.

Q.2. Describe two advantages for programming in assembly and two advantages for programming in a high-level language.

Advantages of programming in assembly language:

1. Space and time efficiency as compilers do not always generate optimum code.
2. Accessibly to system hardware.

Advantages of programming in high-level language:

1. Programs are portable, i.e. they can run on different machines.
2. Programs are easier to understand, write and maintain.

Q.3. Given a magnetic disk with the following properties: Rotation speed = 7200 RPM (rotations per minute), Average seek = 8 ms, Sector = 512 bytes, Track = 200 sectors. Calculate the following:

(i) Time of one rotation (in milliseconds).

Number of rotations per second = 7200/60 = 120 RPS
Rotation time in milliseconds = 1000/120 = 8.33 ms

(ii) Average time to access a block of 32 consecutive sectors.

Average access time = Seek Time + Rotation Latency + Transfer Time
Average rotational latency = time of half rotation = 4.17 ms
Time to transfer 32 sectors = (32/200) * 8.33 = 1.33 ms
Average access time = 8 + 4.17 + 1.33 = 13.5 ms

Q.4. Represent the following numbers in binary and hexadecimal. Use as many bits as needed, and approximate the fraction up to 3 digits:

(i) 250.375

Binary = 1111 1010.011

Hexadecimal = FA.600

(ii) 4444.4
Binary = 1 0001 0101 1100.011
Hexadecimal = 115C.666

Q.5. Express the following numbers in sign-magnitude, 1`s complement, and 2`s complement notations, assuming 8-bit representation:

(i) -119

Sign-magnitude = 1111 0111

1`s complement = 1000 1000

2`s complement = 1000 1001

(ii) -55

Sign-magnitude = 1011 0111

1`s complement = 1100 1000

2`s complement = 1100 1001

Q.6. Show how the decimal integer -120 would be represented in 2`s complement notation using:

(i) 8 bits = 1000 1000

(ii) 16 bits = 1111 1111 1000 1000

Q.7. Perform the following operations assuming 8-bit 2`s complement representation of numbers. Indicate in your answer when an overflow occurs:

(i) 01010011 + 11111111= 0101 0010

(ii) 10110000 - 01110110=10110000 + 10001010=0011 1010 (overflow)

(iii) AF + FF= AE

(iv) AF - 70= AF + 90=3F (overflow)

Q.8. A microcontroller uses 8-bit registers. Give the following in both binary and decimal:

(i) The maximum unsigned number that can be stored.

Binary = 1111 1111

Decimal = 255

(ii) The smallest (negative) number and the largest (positive) number that can be stored using the sign-magnitude notation.

Smallest in Binary = 1111 1111

Smallest in Decimal = -127

Largest in Binary = 0111 1111

Largest in Decimal = +127

(iii) The smallest (negative) number and the largest (positive) number that can be stored using the 2`s complement notation.

Smallest in Binary =1000 0000

Smallest in Decimal = -128

Largest in Binary = 0111 1111

Largest in Decimal = +127

Q.9. If you type the phrase ICS233 on your keyboard, what is the binary sequence sent to the computer using 8-bit ASCII with the 8^{th} bit being an even parity bit.

I	C	S	2	3	3
1100 1001	1100 0011	0101 0011	1011 0010	0011 0011	0011 0011

Q.10. Suppose that a byte contains the ASCII code of a decimal digit; that is `0` to `9`. What hex number should be subtracted from the byte to convert it to the numerical form of the characters?

We need to subtract from it 30h.