ICS 233, Term 072

Computer Architecture & Assembly Language

HW# 1 Solution

Q.1.

Exercise	Term	Description
1.1	Central Processor Unit(CPU)	Active part of the computer, following the instructions of the program to the letter. It adds numbers, tests numbers, controls other components, and so on.
1.2	Abstraction	Approach to the design of hardware or software. The system consists of layers, with each lower layer hiding details from the level above.
1.3	Bit	Binary digit.
1.4	Computer Family	Collection of implementation of the same set architecture. They are usually made by the same company and vary in the price and performance.
1.5	Memory	Component of the computer where all running programs and associated data reside.
1.6	Datapath	Component of the processor that performs arithmetic operations.
1.7	Control	The component of the processor that tells the datapath, memory, and I/O devices what to do according to the instructions of the program.
1.8	Desktop or personal computer	Computer designed for use by an individual , usually incorporating a graphics display, keyboard and mouse .
1.9	Embedded System	Computer inside another device used for running one predetermined application or collection of software.
1.10	Server	Computer used for running larger programs for multiple users often simultaneously and typically accessed only by a network.
1.11	Local area network (LAN)	Computer network that connects a group of computers by a common transmission cable or wireless link within a small geographic area (for example within the same floor of a building).
1.12	Wide area network (WAN)	Computer network that connects computers spanning great distances, the backbone of the internet.
1.13	Supercomputer	High performance machine, costing more than $1 million.
1.14	DRAM (dynamic random access memory)	Integrated circuit commonly used to construct main memory.

1.15	Defect	Microscopic flaw in a wafer.
1.16	Chip	Nickname for a die or integrated circuit.
1.17	Transistor	An on/off switch controlled by electricity.
1.18	Digital Video Disk (DVD)	Optical storage medium with a storage capacity of more than 4.7 GB.
1.19	Yield	The percentage of good dies from the total number of dies on the wafer
1.20	Assembler	Program that converts symbolic versions of instructions into their binary formats.
1.21	Operating system	Program that manages the resources of a computer for the benefit of the programs that run on that machine
1.22	Compiler	Program that translates from a higher-level notation to assembly language.
1.23	VLSI (very large integrated circuit)	Technology in which single chip contains hundreds of thousands to millions of transistors.
1.24	Instruction	Single software command to a processor
1.25	Cache	Small fast memory that acts as a buffer for the main memory.
1.26	Instruction set architecture	Specific interface that the hardware provides the low-level software.
1.27	Semiconductor	Substance that does not conduct electricity well but is the foundation of integrated circuits.
1.28	Wafer	Thin disk sliced from a silicon crystal ingot, which will be later divided into dies.

Q.2. 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.3. 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.4. 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.5. Assume you are in a company that will market a certain IC chip. The fixed costs, including R&D, fabrication and equipments, and so on, add up to $500,000. The cost per wafer is $6000, and each wafer can be diced into 1500 dies. The die yield is 50%. Finally, the dies are packaged and tested, with a cost of $10 per chip. The test yield is 90%; only those that pass the test will be sold to customers. If the retail price is 40% more than the cost, at least how many chips have to be sold to break even?

To break even we need to have the revenue equal to the cost of manufacturing.
Let us assume that the number of chips sold is n_s and the number of chips fabricated equal to n_f.
Cost of a good die = 6000/(0.5*1500) = $8
Cost of fabricating a chip = 8 + 10 = $18
The cost of fabricating n_f chips = 500,000 + 18* n_f
It should be observed that the number of chips sold $n_s = 0.90\, n_f \Rightarrow n_f = 1.11\, n_s$
Chip selling price = 18 *1.4 = $25.2
Revenue = $25.2 * n_s$
Thus, $25.2 * n_s = 500{,}000 + 18 * n_f$

$\Rightarrow 25.2*n_s = 500{,}000 + 18*1.11\ n_s$

$\Rightarrow n_s = 500{,}000/(5.22) = 95{,}785.44$

Thus, **95,786** chips need to be fabricated to break even.

Q.6. 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.7. 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.8. 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.9. 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.10. 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.11. 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.12. 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.