Victoria University Of Bangladesh

Course title ~ CSE-108

Bachelor of Tourism & Hotel Management

Submitted By ~ Computer Fundamentals and Programming Techniques
Name ~ Khan Sifat
Student ID: 1521530011
Batch - 53
Program - BTHM

2.{a}

The main differences between the 3rd and 4th generations of computers lie in their technology, size, speed, and capabilities.

Third-generation computers, developed in the 1960s, primarily used integrated circuits (ICs) for processing. They were smaller, faster, more reliable, and less expensive compared to their predecessors. They also introduced high-level programming languages like COBOL and FORTRAN, making programming more accessible.

Fourth-generation computers, emerging in the mid-1970s, featured microprocessors, which significantly reduced the size and cost of computers while improving their performance. They also introduced features like graphical user interfaces (GUIs), personal computers, and networking capabilities, marking a shift towards more user-friendly and interconnected computing environments.

{B}

Certainly! The basic architecture of a CPU (Central Processing Unit) can be represented by several key components, including the Control Unit, Arithmetic Logic Unit (ALU), Registers, Cache, and Bus Interface. Here's a simplified diagram and explanation of each component:

1. Control Unit (CU):
    - The Control Unit manages the operation of the CPU by decoding instructions fetched from memory and coordinating the execution of these instructions.
    - It controls the flow of data between the CPU's components and ensures that instructions are executed in the correct sequence.
2. Arithmetic Logic Unit (ALU):
    - The ALU performs arithmetic and logical operations on data received from memory or registers.
    - Arithmetic operations include addition, subtraction, multiplication, and division.
    - Logical operations include AND, OR, NOT, and XOR.
    - The ALU processes data according to the instructions provided by the Control Unit.
3. Registers:
    - Registers are small, high-speed storage locations within the CPU used to store temporary data, memory addresses, and intermediate results of calculations.
    - Common types of registers include:
        - Program Counter (PC): Stores the memory address of the next instruction to be fetched.
        - Instruction Register (IR): Holds the current instruction being executed.
        - Accumulator (ACC): Stores the results of arithmetic and logical operations.
        - General-Purpose Registers (GPRs): Used for various purposes by the CPU, such as storing operands and intermediate results.
4. Cache:
    - Cache memory is a small, high-speed memory unit located inside or very close to the CPU.

    - It stores frequently accessed data and instructions to speed up the execution of programs.
    - The cache memory hierarchy typically includes multiple levels (L1, L2, L3), with each level having different capacities and access speeds.
5. Bus Interface:
    - The Bus Interface connects the CPU to other components of the computer system, such as memory, input/output devices, and secondary storage.
    - It consists of multiple buses, including the address bus, data bus, and control bus.
    - The address bus carries memory addresses from the CPU to memory or other devices.
    - The data bus transfers data between the CPU and memory or other devices.
    - The control bus carries control signals that coordinate the operation of the system, such as read/write signals and clock signals.

In summary, the CPU's basic architecture comprises the Control Unit, ALU, Registers, Cache, and Bus Interface, working together to fetch, decode, execute, and store instructions and data as part of the computation process within a computer system.

{C}

Primary storage and secondary storage serve different purposes in a computer system:

Primary Storage:

1. *Volatility*: Primary storage, such as RAM (Random Access Memory), is volatile, meaning its contents are lost when the power is turned off.

2. *Speed*: It is much faster in terms of access speed compared to secondary storage.

3. *Proximity to CPU*: Primary storage is located close to the CPU, allowing for quick access to data needed for processing.

4. *Capacity*: Typically, primary storage has a smaller capacity compared to secondary storage.

5. *Temporary Storage*: Primary storage is used for temporarily storing data and instructions that the CPU actively uses during processing.

Secondary Storage:

1. *Non-volatility*: Secondary storage, such as hard drives and SSDs (Solid State Drives), is non-volatile, meaning its contents are retained even when the power is turned off.

2. *Speed*: It is slower in terms of access speed compared to primary storage.

3. *Capacity*: Secondary storage has a much larger capacity compared to primary storage, allowing for long-term storage of data and programs.

4. *Permanence*: Data stored in secondary storage remains there until intentionally deleted or modified.

5. *External Storage*: Secondary storage devices are typically external to the CPU and are connected via interfaces such as SATA, USB, or PCIe.

3. A)

To convert 275.125 from base 10 to base 8, you can use the integer part and the fractional part separately.

The integer part 275 in base 8 is 423.

To convert the fractional part 0.125 to base 8, you multiply the fractional part by 8 repeatedly until the fractional part becomes zero or you reach your desired precision. So, 0.125 * 8 = 1.0, which is 1 in base 8.

Therefore, 275.125 in base 10 is equal to 423.1 in base 8.

B)

To convert the hexadecimal number ACO.1 ECF to base 10, you can calculate the decimal equivalent of each digit and sum them up:

ACO.1 ECF in hexadecimal is:

A * 16^3 + C * 16^2 + O * 16^1 + 1 * 16^0 + E * 16^-1 + C * 16^-2 + F * 16^-3

A = 10, C = 12, O = 0 (assuming O is the letter, not a digit), E = 14, and F = 15 in base 10.

So, the conversion would be:

10 * 16^3 + 12 * 16^2 + 0 * 16^1 + 1 * 16^0 + 14 * 16^-1 + 12 * 16^-2 + 15 * 16^-3

= 40960 + 3072 + 0 + 1 + (14/16) + (12/256) + (15/4096)

= 40960 + 3072 + 1 + 0.875 + 0.046875 + 0.003662109375

= 44033.925537109375

Therefore, (ACO.1 ECF)16 is approximately equal to 44033.925537109375 in base 10.

C)

To convert the hexadecimal number ACO.1ECF to base 10, you can calculate the decimal equivalent of each digit and sum them up:

ACO.1ECF in hexadecimal is:

A * 16^3 + C * 16^2 + O * 16^1 + 1 * 16^0 + E * 16^-1 + C * 16^-2 + F * 16^-3

A = 10, C = 12, O = 0 (assuming O is the letter, not a digit), E = 14, and F = 15 in base 10.

So, the conversion would be:

10 * 16^3 + 12 * 16^2 + 0 * 16^1 + 1 * 16^0 + 14 * 16^-1 + 12 * 16^-2 + 15 * 16^-3

= 40960 + 3072 + 0 + 1 + (14/16) + (12/256) + (15/4096)

= 40960 + 3072 + 1 + 0.875 + 0.046875 + 0.003662109375

= 44033.925537109375

Therefore, (ACO.1ECF)16 is approximately equal to 44033.925537109375

in base 10

D)

To convert the octal number $(15435.063)_8$ to base 10, you can separate the integer part from the fractional part and then calculate their decimal equivalents:

Integer part: 15435_8 = 1 * 8^4 + 5 * 8^3 + 4 * 8^2 + 3 * 8^1 + 5 * 8^0 = 4096 + 2560 + 256 + 24 + 5 = 6941

Fractional part: 0.063_8 = 0 * 8^-1 + 6 * 8^-2 + 3 * 8^-3 = 0 + 0.375 + 0.046875 = 0.421875

Combining the integer and fractional parts:

$(15435.063)_8$ = 6941 + 0.421875 = 6941.421875 in base 10.

4.

A)

LSB (Least Significant Bit) and MSB (Most Significant Bit) are terms used to describe the binary digits in a binary number.

- *LSB*: The LSB is the rightmost bit in a binary number, representing the smallest value in the binary place value system. It holds the least weight in determining the overall value of the number. For example, in the binary number 1011, the LSB is 1.

- *MSB*: The MSB is the leftmost bit in a binary number, representing the largest value in the binary place value system. It holds the most weight in determining the overall value of the number. For example, in the binary number 1011, the MSB is 1.

Here's an example to illustrate:

Binary number: 1011

- MSB (Most Significant Bit): 1
- LSB (Least Significant Bit): 1

In this example, the MSB carries the most weight, representing a value of 8 (2^3), while the LSB carries the least weight, representing a value of 1 (2^0).

B)

In some contexts, such as when dealing with hexadecimal (base 16) numbers, a leading zero may be used to denote that the number should be interpreted with a certain number of digits.

For example, if we consider "(2*4=08)" as an expression where 2 multiplied by 4 equals 8, it would typically be represented as "2 * 4 = 8" in base 10.

However, if we interpret this expression in base 8 (octal), then 2 multiplied by 4 equals 10 in base 8, but to represent it correctly in two digits, a leading zero is added, resulting in "10".

So, "(2*4=08)" could be understood as "2 * 4 = 10" in octal notation.

1.

To convert the decimal number $(33.12)_{10}$ to base 6, you can separate the integer part from the fractional part and then calculate their equivalents:

Integer part:
33 divided by 6 gives a quotient of 5 and a remainder of 3.
So, 33 in base 6 is represented as 53.

Fractional part:
For the fractional part, multiply the fractional part by 6 until it becomes zero or you reach your desired precision.
0.12 * 6 = 0.72, which is 0 in base 6.
So, 0.12 in base 6 is represented as 0.0.

Combining the integer and fractional parts:

$(33.12)_{10} = (53.0)_{6}$ in base 6.

2.

To convert the base 6 number $(54.22)_6$ to base 10, you can separate the integer part from the fractional part and then calculate their decimal equivalents:

Integer part:
$(54)_6$ = 5 * 6^1 + 4 * 6^0 = 30 + 4 = 34

Fractional part:
For the fractional part, multiply the fractional part by powers of 6 starting from -1, -2, and so on.
$(0.22)_6$ = 2 * 6^-1 + 2 * 6^-2 = 2/6 + 2/36 = 0.3333333333333333 + 0.05555555555555555 ≈ 0.3888888888888889

Combining the integer and fractional parts:

$(54.22)_6$ ≈ 34.38888888888889 in base 10.