

KATHMANDU UNIVERSITY

Department of Computer Engineering



An

Assignment of
Computer Programming {Comp 102}
Assignment No: 1

Submitted by:
Ashraya Kadel
UNG CE I/E
Rollno: 25

Submitted to:
Sameer Tamraker
Department of Computer Engineering

Submission date: 22/02/2023

Q.1: Discuss briefly about basic computer architecture and its peripherals.

Ans:

Computer architecture is defined as a set of rules and methods that describe the functionality, management and implementation of computers. It is the rules by which a system performs or operates.

Computer architecture is divided into three categories:

- i) Instruction set Architecture / ISA: It reads the instruction given to processor and acts accordingly. It allocates memory to instructions and also acts upon memory address mode.
- ii) Micro-Architecture: It describes how a particular processor will handle and implement instructions from ISA.
- iii) System design: It includes the other entire hardware component within the system such as virtualization, multiprocessing.

The main role of Computer Architecture is to balance the efficiency, cost, performance and reliability of a computer system.

ISA acts as a bridge between computer's software and hardware. It translates high level language to binary language ~~or~~ helping users and computer communicate.

Computer architecture consists of following structures:

- i) Processor
- ii) Memory
- iii) Peripherals.

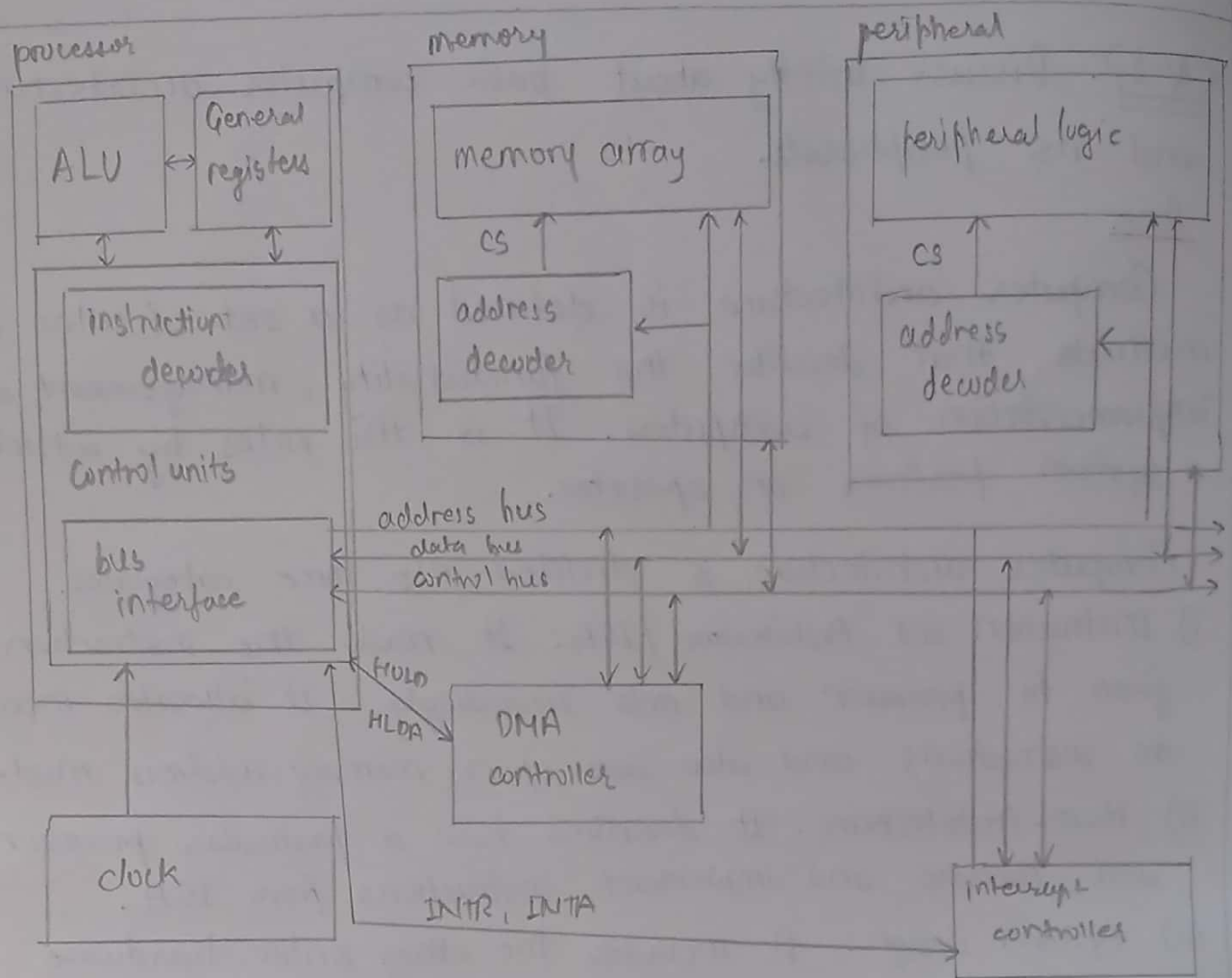


Fig: Computer Architecture

Peripherals of computer architecture are the devices that are connected to a computer system but are not the part of the core computer system architecture. It is classified in three categories.

- i) **Input devices:** Any hardware device that sends data to a computer and allows you to interact and control it.
Eg: Keyboard, mouse, microphone, etc.
- ii) **Output devices:** Any hardware device that gives the result of the entered input once it is processed in human understandable form.
Eg: monitors, headphones, printers, etc.
- iii) **Storage devices:** The devices used to store data in the system which is required for performing any operation in the system.

It makes operation of the system easy.

Q.2: Categorize and explain in brief about the types of software systems.

Ans: Software systems are of two types: Application software and system software.

(i): Application software:

- The type of software system that runs as per user request and runs on the platform provided by system software.
- They are built for specific tasks.
- High level languages are used to write application software.
- It is a software built for specific purpose and a computer system can run without ~~computer~~ application software.

(ii): System software:

- The type of software that is the interface between application software and system is called system software.
- It turns when the system is turned on and turns off when the system is shut down.
- They are general purpose software.
- Low-level languages are used to write the system software.
- A system can't run without system software.

So, System software is more complex to program than application software.

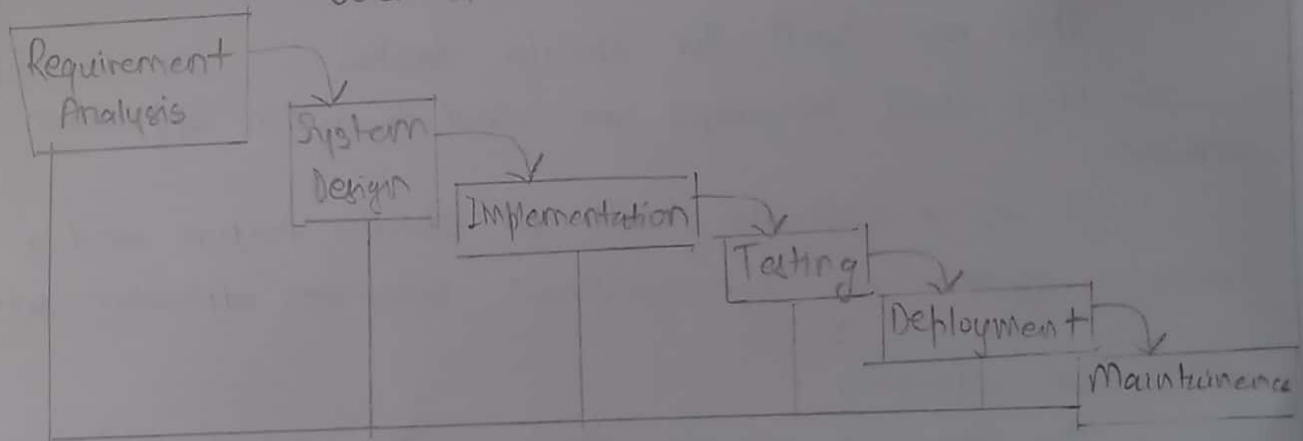
System software runs independently and it doesn't provide interaction with user like application software.

Q-3: Mention in brief about procedures involved in software development life cycle (waterfall model).

Ans:

Waterfall approach was the first SDLC model to be widely used in software engineering to ensure success of the project. In this model, the development process is divided into separate phases.

Waterfall Model



The sequential phase / procedures in SDLC according to waterfall model are as follows:

- (i): Requirement gathering and analysis:
All possible requirements of the system to be developed is gathered and is documented in requirement specification document.
- (ii) System designing: System design is prepared from gathered requirements which helps in defining overall system architecture.
- (iii) Implementation: With inputs from system design, system is developed in units and unit testing is conducted to test its functionality.
- (iv) Integration and testing: All the program units are integrated into a system and post-integration testing is done to check for failures.
- (v) Deployment: Once all testing is completed, the product is deployed for general customer use.
- (vi) Maintenance: Periodic maintenance has to be conducted to enhance product and fix issues.

Q.4: What is meant by operator precedence?
Illustrate with an example.

Ans:

Operator precedence is defined as the order in which operations are performed within an expression. Higher precedence operations are performed

(a) Precedence order in C:

- (i): $() \rightarrow ++ \quad --$ (postfix) Unary
- (ii) $++ \quad --$ (prefix), $!$, \sim , $\&$, sizeof {unary}
- (iii) $* \quad / \quad \%$ {binary}
- (iv) $+ \quad -$ {binary}
- (v) $<< \quad >>$ {bitwise}
- (vi) $< \quad <= \quad > \quad >=$ {relational}
- (vii) $= \quad !=$ {equality}
- (viii) $\&$ {bitwise AND}
- (ix) \wedge {bitwise XOR}
- (x) $|$ {bitwise OR}
- (xi) $\&\&$ {logical AND}
- (xii) $||$ {logical OR}
- (xiii) $?:$ {Ternary}

Eg: (i): $a = 3 * 4 \% 5 / 2 + 6 / 3 * 2$
 $= 2 | 2 \% 5 | 2 + 2 * 2$
 $= 2 | 2 + 4$
 $= 1 + 4 = 5$

(ii) $a = 3 * (4 \% 5) / 2$
 $= 3 * 4 / 2$
 $= 12 / 2 = 6$

Q.5: What is meant by rule of associativity in operations of operators? Illustrate with example.

Ans:

Rule of associativity in operations of operators is a property that determines how operators of the same precedence are grouped in the absence of parentheses.

Associativity of operators:

- i) $() \rightarrow ++ -- (\text{post}) \Rightarrow L \rightarrow R$
- ii) $++ -- (\text{prefix}), !, \sim, *, \& \Rightarrow R \rightarrow L$
- iii) $*, /, \% \Rightarrow L \rightarrow R$
- iv) $+, - \Rightarrow L \rightarrow R$
- v) $<< >>$
- vi) $==, !=, <, <=, >, >= \Rightarrow L \rightarrow R$
- vii) $\&\& ==, ! = \Rightarrow L \rightarrow R$
- viii) $\& \Rightarrow L \rightarrow R$
- ix) $\wedge \Rightarrow L \rightarrow R$
- x) $! \Rightarrow L \rightarrow R$
- xi) $\&\& \Rightarrow L \rightarrow R$
- xii) $|| \Rightarrow L \rightarrow R$
- xiii) $?: \Rightarrow L \rightarrow R$
- xiv) $= \Rightarrow R \rightarrow L$

Eg: i) $a = 5$

$\therefore a = 5$ ie, value of a becomes 5

$$\begin{aligned} \text{(ii)} \quad & 3 * 5 \% 4 + 2 \\ &= 15 \% 4 + 2 \\ &= 3 + 2 = 5 \end{aligned}$$

(iii) $a = 4, b = 6;$

result = $a > b$ $\&\& \text{ printf("5")} || \text{ printf("4")};$
~~Outt~~ $\text{ printf("val", result);}$

Output: ~~printf~~ $\begin{matrix} L \\ 1 \end{matrix}$

Q.67: What are library functions? Why are they important?

Ans:

Library functions are built-in functions that are grouped together and placed in common location called library. Each function in library performs specific operation and it helps us to get pre-defined output.

The important of library functions are as follows:

- i) They work. They have undergone multiple rigorous testing and are easy to use.
- ii) They have been optimized for performance and use the most efficient code for maximum performance.
- (iii) They help us to save considerable development time.
- (iv) They are portable and can be done in any computer.

Q.77: Write in brief about compilers and interpreters.

Ans:

Translators are the programs that convert source code into machine codes. They are of two types:

i) Compiler:

The translators that translate the whole program written in source code to machine codes at once are called compilers. They generate intermediate machine code.

ii) Interpreter:

The translators that translate a line at a time from source code to machine code is called interpreter. They never generate intermediate machine codes.

<Q.8>: Convert:

(a): $(10101011)_2 = (?)_{16} = (?)_8$

Soln:

Given,

$(10101011)_2$

$$\begin{aligned} &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 128 + 0 + 32 + 0 + 8 + 0 + 2 + 1 \\ &= (171)_{10} \end{aligned}$$

Into hexadecimal,

16	171	
16	10	11 (B)
	0	10 (A)

Into octal.

8	171	
8	21	3
8	2	5
	0	2

$\therefore (10101011)_2 = (AB)_{16} = (352)_8$

(b): $768 = (?)_8 = (?)_{16}$

Given,

$(768)_{10}$

Into octal,

8	768	
8	96	0
8	12	0
8	1	4
	0	1

Into hexadecimal.

16	768	
16	48	0
16	3	0
	0	3

$\therefore (768)_{10} = (1400)_8 = ~~(768)~~ (300)_{16}$

<Q9>: List out 10 library functions used in C program.
What are header files in C?

Ans:

The 10 library functions used in C-program are as follows:

- | | |
|-----------------|------------------|
| (i) sqrt() | (vi) scanf() |
| (ii) pow(a, a) | (vii) printf() |
| (iii) toupper() | (viii) putchar() |
| (iv) tolower() | (ix) getch() |
| (v) toascii() | (x) clrscr() |

Header files are the files containing the set of pre-defined standard library functions. It helps us use the contents in the program.

<Q.10>: What are keywords? List out 15 keywords frequently used in C.

Ans:

Keywords are those words whose meaning is pre-defined and it is the basic building block for writing instruction in C-program language.

The 15 keywords frequently used in C are..

- | | |
|-------------|----------------|
| i) const | ix) char |
| ii) float | x) while |
| iii) int | xi) sizeof |
| iv) break | xii) unsigned |
| v) if | xiii) continue |
| vi) else | xiv) default |
| vii) switch | xv) do |
| viii) void | |