

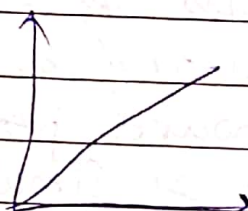
CO CHP-1

- Personal Computers (PC's) → delivery of good performance
 - Low cost
 - execute third-party software
- Servers → usually accessed via network
 - oriented to carry large workload
 - which consists of single complex application or handling many small jobs.
 - based on software from another source:
 - Data Base or Simulation System
 - greater emphasis on DEPENDABILITY
 - widest range in cost & capability
- Supercomputers →
 - Has tens of thousands of processors
 - Many terabytes of memory
 - Used for high-end scientific calculation
 - represent small fraction of computer market
- Embedded Comp.
 - Used to run one set of application
 - Integrated with hardware and delivered as a single system.
 - low tolerance towards failure
- PMD (Personal Mobile Device)
 - Battery operated, wireless connectivity
 - rely on touch-sensitive screen

- Cloud Computing
→ depends on giant data centers \Rightarrow WSCs
(Warehouse Scale Computers)

Eight Ideas

- 1) Moore's law \rightarrow Integrated circuit resources double every 18 - 24 months.



- 2) Make the Common Case Fast:
It will enhance the performance better than optimizing the rare case.

- 3) Performance via Parallelism:
→ More performance on performing operations in parallel \rightarrow PARALLEL PERFORMANCE

- 4) Pipelining \rightarrow A particular pattern of parallelism

- 5) Performance via Prediction:

- 6) Hierarchy of memories

- 7) Dependability via redundancy

Operating System - Interface b/w user's program and the hardware

- Handling basic input/output operation
- Allocating storage & memory
- Providing for protected sharing of comp. among multiple applications using it simultaneously

Compiler: Translation of program from high level lang such as C, C++ etc into basic instructions that hardware can execute.

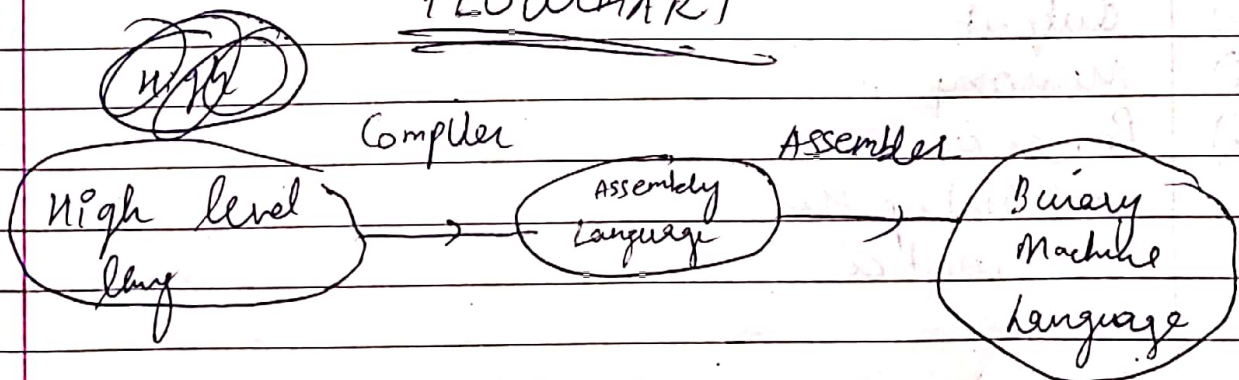
Assembler: First program that translates a symbolic version of an instruction to binary version

Finally:

ASSEMBLY LANGUAGE \rightarrow Symbolic Language

MACHINE LANGUAGE \rightarrow Binary language

FLOWCHART



- Fortran → designed for scientific computation
- Cobol → Business data processing
- Lisp → Symbol manipulation

WHY HIGH-LEVEL PROGRAMMING ^{LANG} ~~LANG~~ ?

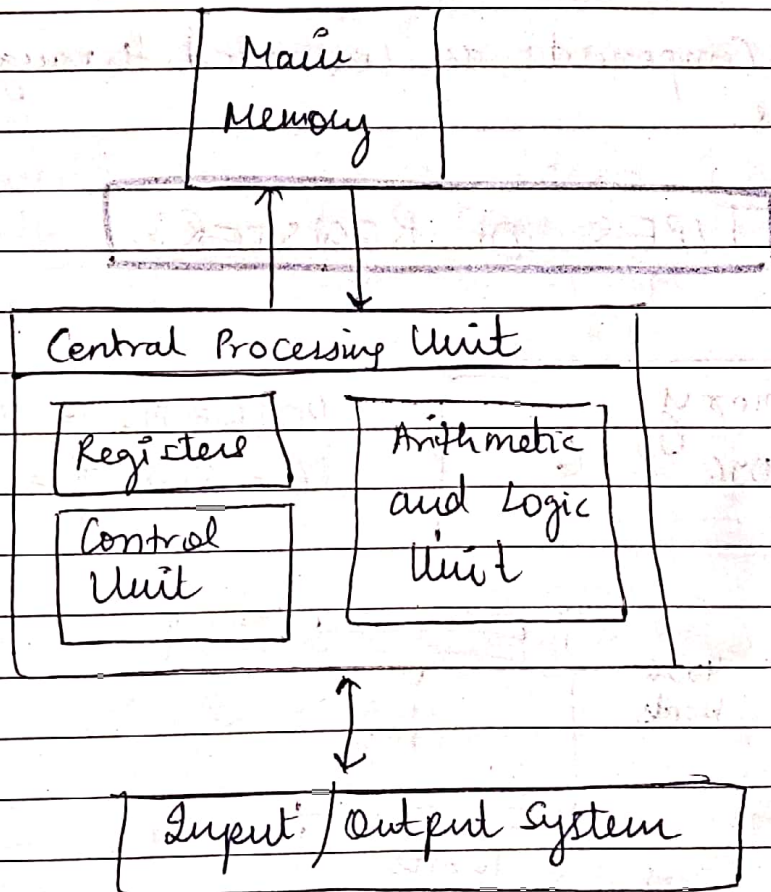
- 1) Allow programmers to think in a more natural language
- 2) Improved programmer productivity → less time when programs are written in languages that require few lines. Conciseness is a big advantage
- 3) Allow programs to be independent of the computer on which they were developed

• FIVE COMPONENTS OF COMPUTER :

- 1) Input
- 2) Output
- 3) Memory
- 4) Processor
 - Datapath
 - Control

VON NEUMANN ARCHITECTURE (Stored Memory Program)

Data & Program (Set of Instructions) store in Main memory



- Register → to store temporary data, fastest memory, size of register is smallest, means has smallest memory, ~~size~~
- It is sequence of bits (when we combine multiple flip-flops)
- There is a speed mismatch in the working of CPU (very fast) and main memory (relatively slower). So to reduce this burden from memory, registers store data temporarily as it is closer to ALU.

Control Unit -

- Timing signal → generate timing .. every time particular task is done
- control register → control & coordinate register.

→ These components are connected through Buses.