Computer Architecture and Organization

Module -1

- Introduction and overview of computer architecture
  - Introduction to computer systems
  - Over view of organization and architecture
  - Functional components of a computer
  - Registers and register files
  - Interconnection of components
  - Organization of Von Neumann machine
  - Harvard architecture
  - Performance of processor

Why this subject?

- To acquire some understanding and appreciation of a computer systems functional components, their characteristics, performance and interactions
- Need to understand computer architecture in order to structure a program so that it runs more efficiently on a real machine
- In selecting a system to use, they should be able to understand the trade of among various components such as CPU clock speed vs memory

Module 1

Introduction and Overview of Computer Architecture

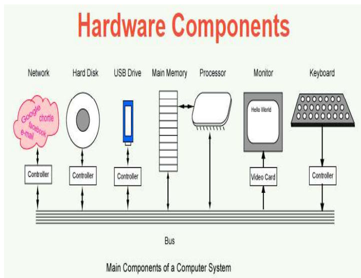
Introduction to Computer Systems

A computer is an electronic data processing machine that accepts data and instructions through input unit (or from memory) and processes data as per the instruction, to get desired output.

A computer system is a complex system consists of both hardware and software.

Hardware and Software

- The **hardware** components of a computer system are the electronic and mechanical parts
- The **software** components of a computer system are the data and the computer programs.

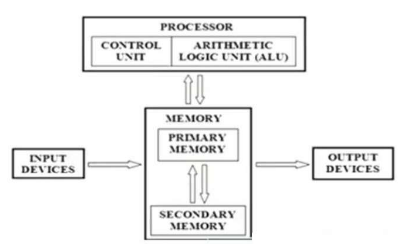


- Computer is a digital device, which works on two levels of signal: High or Low
- High level signal (5v or 12v) and low level signal (0v)
- To make it convenient, 0v represented by L (Low) or 0
  1v represented by H(High) or 1
- All the functionalities of the computer system are represented with 0 and 1
- Binary number system is used to represent information and manipulation of information in computer

Overview of Computer Organization and Architecture

- Organization
  - Design of the components and functional blocks using which computers are built
  - Analogy: Civil Engineer task during building construction (cement, bricks, iron rods and other building materials during construction)
- Architecture
  - How to integrate the components to build a computer system to achieve a desired level of performance
  - Analogy: Architect task during the planning of the building (overall layout, floor plan, etc.)

Functional Components of a Computer



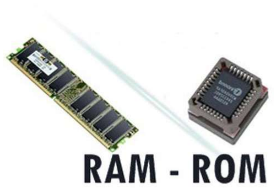
Input Unit

Memory

- Physical device to store programs or data
- Two types: Main memory (Physical memory) and Secondary memory
- Memory vs storage
- Main memory vs Secondary memory

- **Main memory:**
  - closely connected to the processor.
  - stored data are quickly and easily changed.
  - holds the programs and data that the processor is actively working with.
  - interacts with the processor millions of times per second.
  - needs constant electric power to keep its information.
- **Secondary memory:**
  - connected to main memory through the bus and a controller.
  - stored data are easily changed, but changes are slow compared to main memory.
  - used for long-term storage of programs and data.
  - before data and programs can be used, they must be copied from secondary memory into main memory.
  - does not need electric power to keep its information.

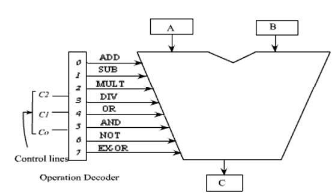
Main memory vs Secondary memory



Central Processing Unit (CPU)

Arithmetic and Logic Unit (ALU)

- Most computer operations are performed at ALU
- Example



Block Diagram of the ALU

Control Unit

- Operations of all the units are coordinated by control unit
- Sends control signal to all the units
- Timing signals that govern the data transfer, are generated by control circuits
- Timing signals are signals that determine when a given action takes place

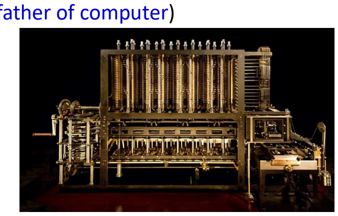
Output Unit

Evolution of computer system

- The beginning of computing – Abacus 3000 BC
- Calculating tool that was in use centuries before the adoption of the written modern numeral system
- Still widely used by traders, merchants and clerks in Asia and Africa

Babbage's Difference Engine (1823)

- He was a mathematician, philosopher and a mechanical engineer
- He is best remembered now for originating the concept of a programmable computer
- He is credited with inventing the first mechanical computer (So considered as father of computer)



ENIAC (1943 -1946)

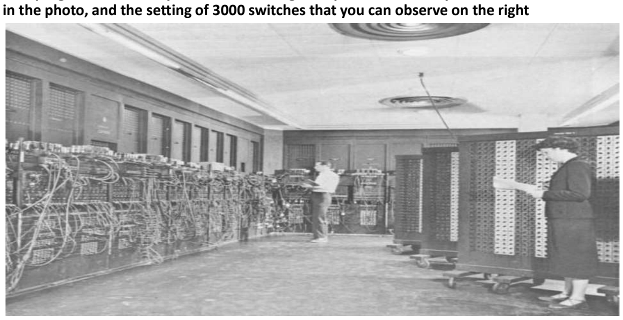
- ENIAC (Electronic Numeric Integrator And Computer)
- Designed by Mauchly and Eckert (University of Pennsylvaniya)
- First general purpose electronic computer
- Response to WW2 need to calculate trajectory tables for weapons
- ENIAC details in decimal (not binary)
- 20 accumulators for 10 digits
- Programmed manually by switches
- 18,000 vacuum tubes, 30 tons
- 15,000 sft, 140 KW power consumption
- 5000 additions per second

ENIAC (1943 -1946)

- Vacuum tubes

ENIAC (1943 -1946)

- **To reprogram the ENIAC you had to rearrange the patch cards that you can observe on the left in the photo, and the setting of 3000 switches that you can observe on the right**

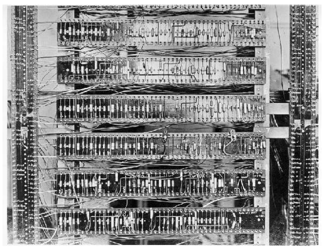


Transistor Based Computers

- Transistors replaced vacuum tubes
- Smaller, cheaper and less heat dissipation
- Made from silicon (sand)
- Invented at Bell labs, 1947
- Commercial Transistor based computers: NCR & RCA produced small transistor machines, IBM 7000, DEC – 1957 (PDP -1)

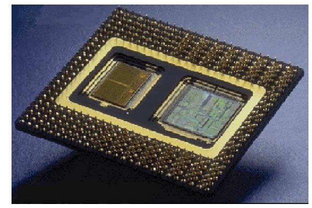
Transistor Based Computers

- First transistor based computer – Manchester University (1953)



Integrated Circuits

- A single self contained transistor is called a discrete component
- Transistor based computers – discrete components manufactured separately, packed in their own containers, and soldered or wired together onto circuit boards
- The transistor based computers contained about 10000 transistors – but grew up to hundreds of thousands
- Integrated circuits revolutionized electronics
- Silicon chip - collection of tiny transistors



Generation of computers

- Vacuum tubes – 1946 – 1957 (one bit – size of hand)
- Transistors - 1958 -1964 (one bit – size of a finger nail)
- Small Scale Integration (SSI) – 1965 onwards (up to 100 of devices on a single chip)
- Medium Scale Integration (MSI) –up to 1971 (100 to 3000 devices on chip
- Large Scale Integration (LSI) – 1971 -1977 (3000 to 100,000 devices on a chip
- Very Large Scale Integration (VLSI) – 1978 – Till date (100,000 to 100,000,000 devices on a chip
- Ultra Large Scale Integration (ULSI) (over 100,000,000 devices on a chip

Computer Generations

Generation	Period	Technology
First	1945 - 1954	Vacuum Tubes
Second	1955 - 1964	Transistors
Third	1965 -1974	Integrated Circuits (SSI, MSI)
Fourth	1975 - ?	Integrated Circuits (LSI, VLSI)
FIFTH	?	AI, Neural Network, Web Computing etc.

Registers

- To speed up the processor operations, the processor includes some internal memory storage locations called registers
- Memory hierarchy
  - Registers
  - Cache memory
  - Main memory
  - Secondary memory
- At higher level of hierarchy, memory is faster, smaller and more expensive
- Registers top level of hierarchy
- Two roles
  - User visible registers – referenced by machine instruction
  - Control and Status registers – employed to control the operation

User visible registers

- General purpose register
- Data registers
- Address registers
- Condition code registers

User visible registers

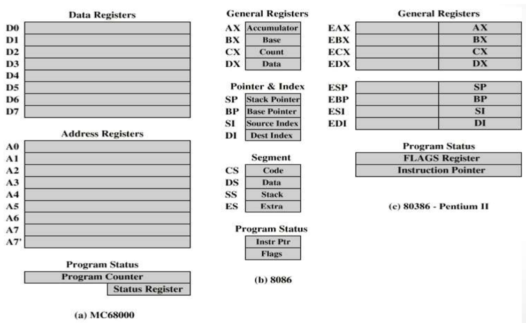
- General-purpose register ($R_0 – R_{n-1}$)
- Data register – used only to hold data and not for any other purpose
- Address Register
  - Segment register – holds the address of the base of the segment
  - Index register – Used for indexed addressing
  - Stack Pointers – points to the top of the stack
- Condition Codes (Flags) – are bits set by the processor as a result of an operation

Control and Status registers

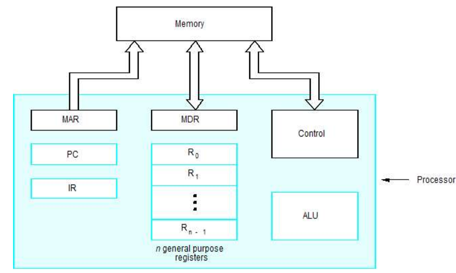
- Not visible under user mode

1. Instruction Register (IR) – used to store the instructions
2. Program Counter (PC) – gives the address of next instruction to be executed
3. Memory address register (MAR) – holds the address of the memory to read/write data
4. Memory data register (MDR) – holds the data/instruction fetched from the memory

Examples of Register Organization



Connection Between the Processor and the Memory

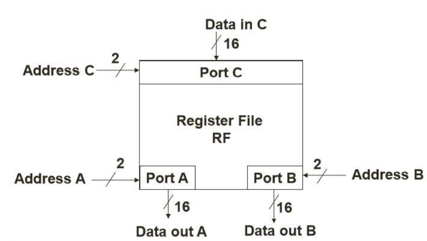


Register Files

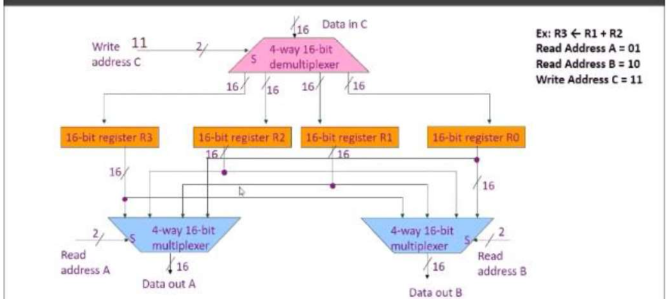
- The register file is the component that contains all the general purpose registers of the microprocessor.
- It is an array of processor register in a CPU
- It can be implemented using SRAM with multiple ports
- SRAMs are distinguished by having dedicated read and write ports, whereas ordinary multiported SRAMs will usually read and write through the same ports.

Register Files

- A simple register file is a set of registers and a decoder. The register file requires an address and a data input
- To read two values at once and write one value back in a single cycle. Consider the following equation: C = A + B



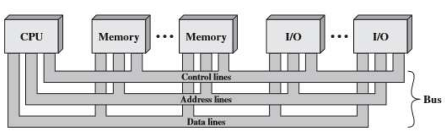
A Register File with three access ports – logic diagram

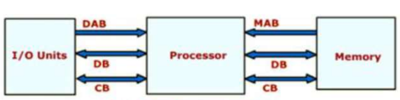


Interconnection of Components

- Functional components of the computer needs to be interconnected
- A group of lines/wires that serves as a connecting path for several devices is called a Bus
- The collection of paths connecting the various modules is called the interconnection structure
- Bus that connects the major components are called System Bus
- Bus is a shared transmission medium

Bus Structures





Two Bus Organization

Speed Issue

- Different devices have different transfer/operate speed.
- If the speed of bus is bounded by the slowest device connected to it, the efficiency will be very low.
- How to solve this?
- A common approach – use buffers.

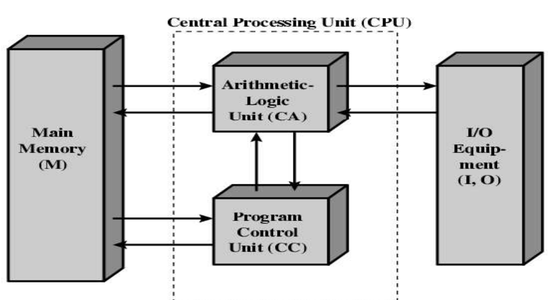
Computer Architecture

- Broadly can be classified into two types
  - Von Neumann architecture
  - Harvard architecture

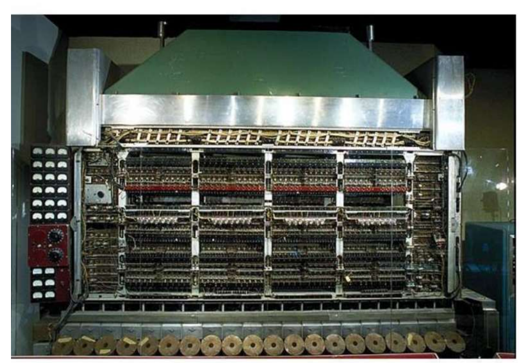
Von Neumann Architecture

- The stored program concept is given by a mathematician Von Neumann
- In 1946, Von Neumann and his colleagues began the design of a new stored program computer, referred as IAS computer at the Princeton Institute for Advanced Studies
- General structure of IAS computer
  - Instructions and data are stored in the same memory module
  - More flexible and easier to implement
  - Suitable for most of the general purpose processors
  - Disadvantage: All instructions and data are moved back and forth through the same pipe

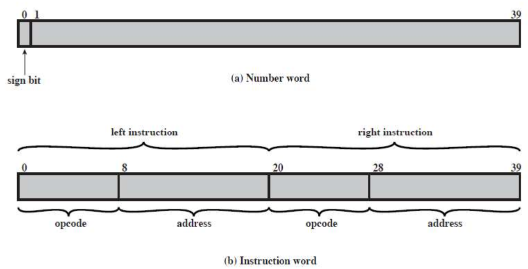
Structure of Von Neumann Machine



The IAS Computer, 1952



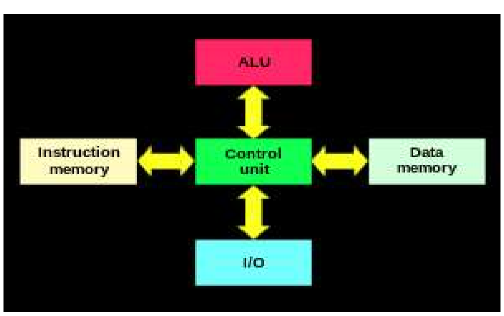
IAS Memory Format



IAS Memory Format

- 1000 x 40 bit words (1000 storage locations of 40 binary bits each)
  - Both instructions and data stored here
- Number format
  - Each number is represented by a sign bit and a 39 bit value
- Instruction format
  - A word may contain two 20 bit instructions
  - 8 bit operation code (opcode) : specifying the operation to be performed and
  - 12 bit address : designating one of the word in memory (0 to 999)

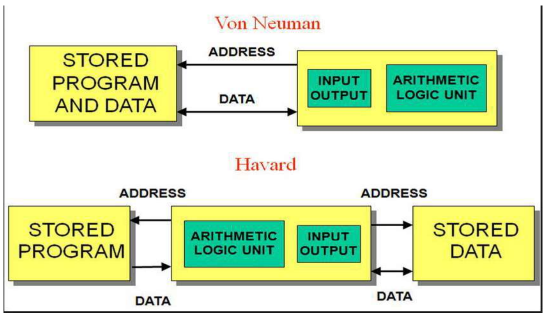
Harvard Architecture



Harvard Architecture

- The Harvard architecture is a computer architecture with physically separate memory
- Separate memory for program and data
  - Instruction are stored in program memory
  - Data stored in data memory
- In Harvard architecture, there is no need to make the two memories shared characteristics
- In particular, word width and memory address structure can differ
- In some systems, instructions can be stored in read-only memory whereas the data stored in read-write memory
- In some systems, there is much more instruction memory than data memory so instruction addresses is wider than data addresses

Von Neumann vs Harvard architecture



Pipeline in Executing Instructions

- Pipeline is the concept to speedup the execution
- Pipeline means overlap execution
- Instruction execution is typically divided into 5 stages
  - Instruction Fetch (IF)
  - Instruction Decode (ID)
  - ALU operation (EX)
  - Memory Access (MEM)
  - Write Back result to register file (WB)
- These five stages can be executed in an overlapped fashion in pipeline architecture

Basic 5-stage Pipelining Diagram

Instruction	Pipeline Stage						
1	IF	ID	EX	MEM	WB		
2		IF	ID	EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX
Clock Cycle	1	2	3	4	5	6	7

Performance

- The most important measure of a computer is how quickly it can execute programs.
- Three factors affect performance:
  - Hardware design
  - Instruction set
  - Compiler

Performance

- Processor time to execute a program depends on the hardware involved in the execution of individual machine instructions.
- The processor and a relatively small cache memory can be fabricated on a single integrated circuit chip.
- Speed
  - The internal speed of performing the basic steps of instruction processing on cache is very high
- Cost
- Memory management

Processor Clock

- Clock, clock cycle, and clock rate
- Processor circuits are controlled by a timing signal called Clock
- The clock defines regular time intervals called clock cycles
- To execute a machine instruction, the processor divides the action to be performed into sequence of basic steps such that each step can be completed in one clock cycle
- Let P be the length of one clock cycle. Its inverse R=1/P is called clock rate, which is measured in cycles per second.

Basic Performance Equation

- T – processor time required to execute a program that has been prepared in high-level language
- N – number of actual machine language instructions needed to complete the execution (note: loop)
- S – average number of basic steps needed to execute one machine instruction. Each step completes in one clock cycle
- R – clock rate

$$T = \frac{N \times S}{R}$$

How to improve T?

- Instructions are not necessarily executed one after another.
- The value of S doesn't have to be the number of clock cycles to execute one instruction.
- Pipelining – overlapping the execution of successive instructions.
- Add R1, R2, R3
- Superscalar operation – multiple instruction pipelines are implemented in the processor.
- Goal – reduce S (could become <1!)

Clock Rate

- Increase clock rate
  - Improve the integrated-circuit (IC) technology to make the circuits faster
  - Reduce the amount of processing done in one basic step (however, this may increase the number of basic steps needed)
- Increases in R that are entirely caused by improvements in IC technology affect all aspects of the processor's operation equally except the time to access the main memory.

Compiler

- A compiler translates a high-level language program into a sequence of machine instructions.
- To reduce N, we need a suitable machine instruction set and a compiler that makes good use of it.
- Goal – reduce N×S
- A compiler may not be designed for a specific processor; however, a high-quality compiler is usually designed for, and with, a specific processor.

Performance Measurement

- T is difficult to compute.
- Measure computer performance using benchmark programs.
- System Performance Evaluation Corporation (SPEC) selects and publishes representative application programs for different application domains, together with test results for many commercially available computers.
- Compile and run (no simulation)
- Reference computer

$$SPEC\ rating = \frac{\text{Running time on the reference computer}}{\text{Running time on the computer under test}}$$

$$SPEC\ rating = \left(\prod_{i=1}^{n} SPEC_i\right)^{\frac{1}{n}}$$