The Islamia University of Bahawalpur Pakistan

Introduction to Computer

Computer System Architecture

Course Instructor: Hafiz Jawad Ahmad

بِسۡمِ اللّٰہِ الرَّحۡمٰنِ الرَّحِیۡمِ

ترجمہ: شُروع اَللہ کے پاک نام سے جو بڑا مہربان نہایت رحم والا ہے۔

رَّبِّ زِدۡنِیۡ عِلۡمًا

ترجمہ: اے میرے رب! میرے علم میں اضافہ فرما۔

سورہ طٰہٰ: ﴿۱۱۴﴾ پارہ نمبر-(16)

Processor & Memory Interaction

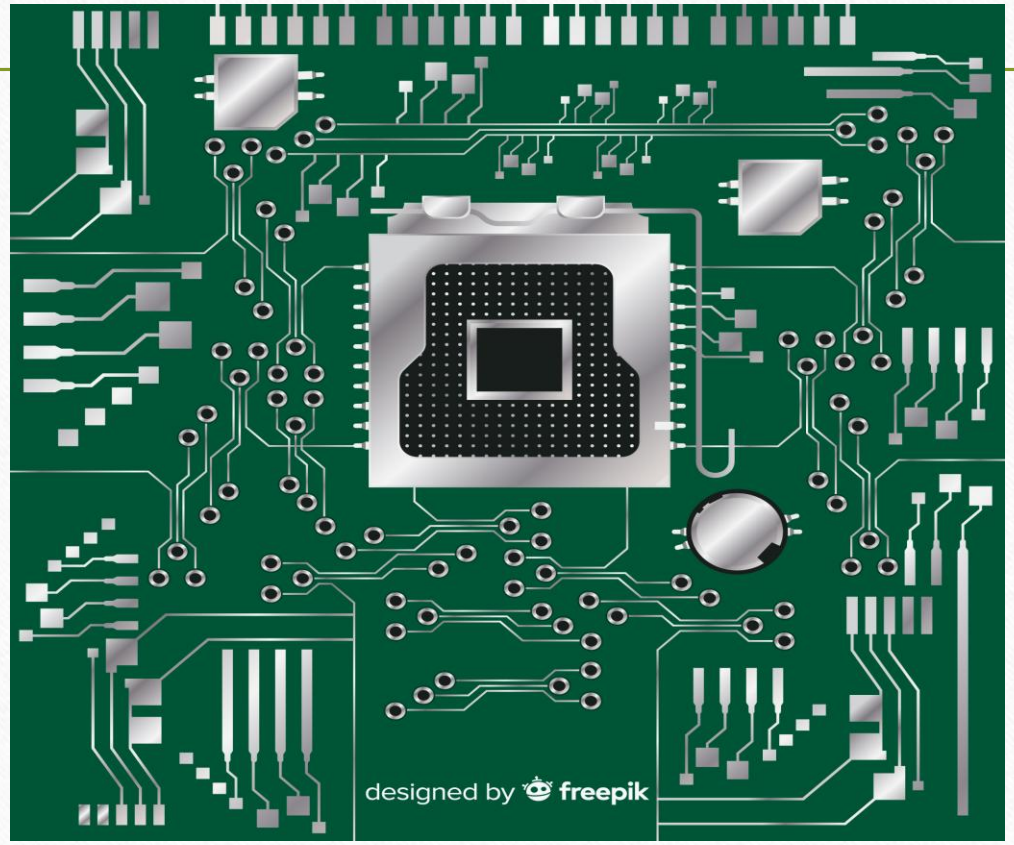
Processor and Memory Interaction

- The **interaction between a processor and memory** is fundamental to the functioning of a computer system. The processor (CPU) executes instructions and processes data, while memory stores the instructions and data temporarily or permanently. Their seamless collaboration ensures smooth operation.

Components Involved

- **Processor (CPU)**
- **Memory**

Bus System



Bus System

- **Bus System:** A bus consists of parallel electrical wire lines. It is used to transfer data between different components of computer system. One line can transfer one bit at a time

Types Of Busses

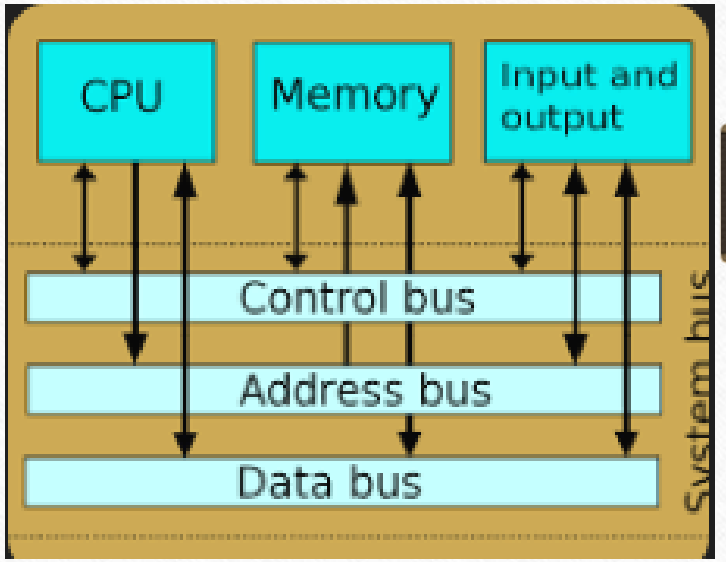
1) Data Bus **2) System Bus**

3) Address Bus **4) Control Bus**

5) Expansion Bus **6) Memory Bus**

7) Video Bus **8)USB (Universal Serial Bus)**

9) Serial Bus **10)Parallel Bus**



System Bus

- **System Bus:** The system bus is used to connect main components of computer system such as cpu & main memory. System Buses are the part of main mother board.
- **Data Bus:** The data bus is used to transfer data between different components of computer systems.

Types Of System Bus

- **Address Bus:** Many Components are connected with each other through buses. Each component has a unique id. This id is called address of that component. These components use address bus to communicate with each other. It unidirectional as it carries address from source to destination
- **Control Bus:** Control bus is used to transmit different commands or signals from one component to another.
- **Expansion Bus:** The expansion bus is used to connect cpu with external devices such as mouse,keyboard, printer, scanner & modems etc.

Types Of System Bus

- **Memory Bus:** Special bus used to connect the CPU directly to the main memory (RAM). Characteristics: Bandwidth and speed can significantly affect system performance.
- **Video Bus:** Specifically designed for graphics data transfer.

Example: AGP, used to connect video cards to a system for improved graphics performance.

- **USB Bus:** Transfers data one bit at a time over a single channel.
- **Examples**: USB, SATA (Serial ATA), and RS-232.

Types Of System Bus

- **Parallel Bus:** Transfers multiple bits simultaneously across multiple channels.
- Examples: Older connections like the PCI bus and various printer interfaces.

CPU (Central Processing Unit)

- **Control Unit (CU):** The Control Unit coordinates and manages the flow of data and instructions within the computer. It fetches instructions from memory, decodes them, and directs the CPU and other components to execute the required operations.
- **Arithmetic Logic Unit (ALU):** Performs arithmetic and logical operations. ALU handles arithmetic and logical operations, while the CU manages instructions and coordination.

What is Memory ?

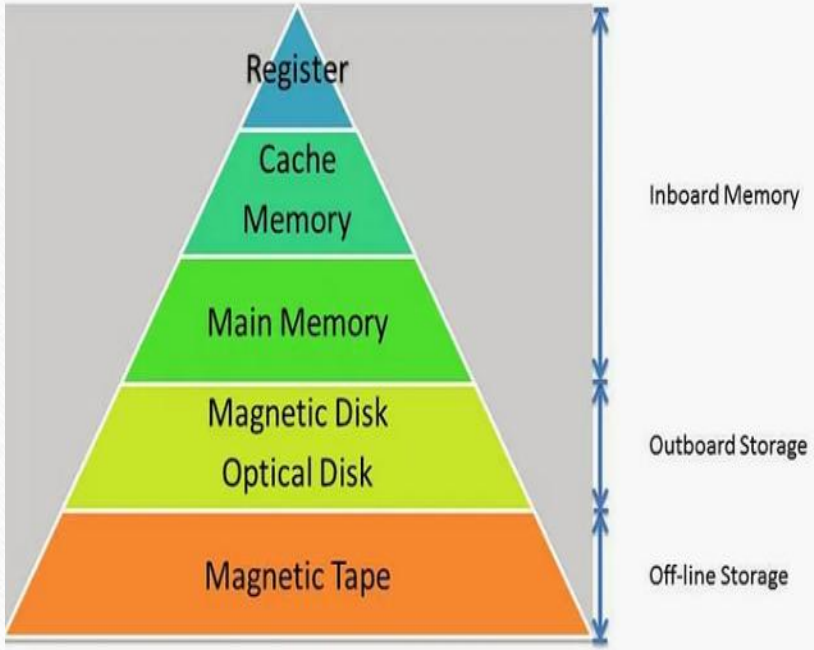
Definition: Memory is a component that is used to store data temporarily or permanently.

- Examples include

RAM (Random Access Memory),

ROM (Read-Only Memory),

and storage memory like hard drives or SSDs.



Memory Types

Volatile Memory/Primary memory/:

- **Definition:** Volatile memory is a type of computer memory that requires a constant power supply to retain stored data. When the power is turned off, all data is lost

Non-Volatile Memory:

- **Definition:** Non-volatile memory is a type of computer memory that retains data even when the power supply is turned off

Memory Types

Volatile Memory Examples

- Flip Flop
- RAM (Random Access Memory)
- Cache Memory
- Registers

Non-Volatile Examples

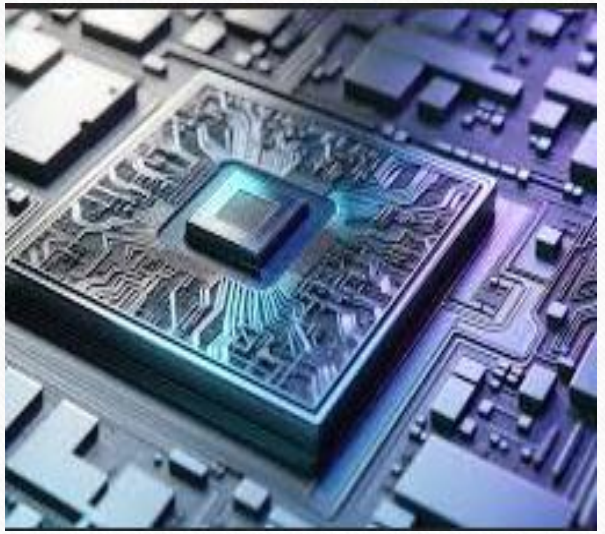
- ROM (Read-Only Memory)
- Hard Disk Drives (HDDs)
- Solid-State Drives (SSDs)
- Flash Memory
- CMOS Memory

Registers

- **Registers:** It is a small high speed memory inside CPU, it is used to store data temporary. Data is stored in main memory for execution.

Types of Names

- **1) Program Counter (PC)**
- **2) Instruction Register (IR)**
- **3) Memory Address Register (MAR)**
- **4) Memory Buffer Register (MBR)**
- **5) Stack Pointer Register**



Types of Registers

1) Program Counter (PC): Stores Next Instruction to be executed

2) Instruction Register (IR): Used to Store Fetched Instructions

3) Memory Address Register (MAR): Used to Store Memory Address Used By CPU

4) Memory Buffer Register (MBR): Used to Store Data that is coming or outgoing from memory

5) Stack Pointer Register: Use LIFO order to access the address stored in it

Types of Registers

- Accumulator Register
- Pointer Register
- Index Register
- Base Register
- Flag Register
- Instruction Pointer Register

Types of Registers

- **Accumulator Register (AC)**: Stores results of arithmetic and logic operations. It's heavily used in calculations.
- **Pointer Register**: Holds memory addresses to locate data or instructions. Often used for dynamic memory access.
- **Index Register**: Stores offsets for accessing array elements or data in loops. It simplifies repetitive memory tasks.
- **Base Register**: Holds a base address, combined with an offset for memory access. Useful in segmented memory systems.
- **Flag Register**: Tracks operation status (e.g., zero, carry, overflow). It helps in decision-making during execution.
- **Instruction Pointer (IP)**: Points to the next instruction for execution. It ensures sequential program flow.

INSTRUCTION GROUPS

Instruction groups in registers refer to categories of instructions based on their function in relation to registers.

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Logical Instructions
4. Program Control Instructions
5. Bit Manipulation Instructions
6. Processor Control Instructions

INSTRUCTION GROUPS

Instruction groups in registers refer to categories of instructions based on their function in relation to registers.

Here's a simple Explanation:

1. **Data Transfer Instructions**: Move data between registers, memory, or I/O (e.g., MOV, PUSH, POP).
2. **Arithmetic Instructions**: Perform calculations like addition, subtraction, multiplication, and division (e.g., ADD, SUB, MUL).
3. **Logical Instructions**: Handle bitwise operations such as AND, OR, XOR, and NOT (e.g., AND, OR, XOR).
4. **Program Control Instructions**: Alter the program flow, like jumps or calls (e.g., JMP, CALL, RET).
5. **Bit Manipulation Instructions**: Operate on individual bits within a register (e.g., SHL, SHR, ROL).
6. **Processor Control Instructions**: Manage processor-specific tasks like interrupts or flags (e.g., STC, CLC, HLT).

These groups allow efficient organization and execution of operations using registers.

ADDRESSING MODES

The iAPX88 processor supports seven modes of memory access.

- Direct
- Based Register Indirect
- Indexed Register Indirect
- Based Register Indirect + Offset
- Indexed Register Indirect + Offset
- Base + Index
- Base + Index + Offset

ADDRESSING MODES

•**Direct Addressing**: The operand (data) is directly specified by the register's value. The register holds the actual address of the data.
•**Example**: MOV AX, [1000h] — Loads data from memory location 1000h into AX.

•**Based Register Indirect**: The operand's address is stored in a register, and the actual data is fetched from that address.
•**Example**: MOV AX, [BX] — Loads data from the address stored in register BX into AX.

•**Indexed Register Indirect**: The operand's address is the sum of a register (index) and a fixed value, and the data is fetched from that computed address.
•**Example**: MOV AX, [SI + 5] — Loads data from the address SI + 5 into AX.

•**Based Register Indirect + Offset**: The operand's address is computed by adding an offset to the address stored in a base register, and the data is fetched from the resulting address.
•**Example**: MOV AX, [BX + 10] — Loads data from the address BX + 10 into AX.

ADDRESSING MODES

•**Indexed Register Indirect + Offset**: The operand's address is computed by adding both an index register and an offset, then the data is fetched from that address.
Example: MOV AX, [SI + DI + 4] — Loads data from the address SI + DI + 4 into AX.

•**Base + Index**: The operand's address is calculated by adding the contents of two registers (base and index), without any additional offset.
Example: MOV AX, [BX + SI] — Loads data from the address BX + SI into AX.

•**Base + Index + Offset**: The operand's address is computed by adding the base register, index register, and an offset value.
Example: MOV AX, [BX + SI + 8] — Loads data from the address BX + SI + 8 into AX.

SIZE MISMATCH ERRORS

- **Size Mismatch Errors in Registers** occur when there is a discrepancy between the size (bit width) of the operands or registers involved in an operation. This can lead to unexpected behavior, data corruption, or crashes
  - **Register Size Mismatch**
  - **Immediate Value Mismatch**:
  - **Memory-to-Register Mismatch**
  - **Pointer Size Mismatch**
  - **Arithmetic Size Mismatch**

SIZE MISMATCH ERRORS

•**Register Size Mismatch**: When trying to move data between registers of different sizes (e.g., 16-bit data in a 32-bit register), it can cause errors if not handled properly.
Example: Attempting to move a 32-bit value into a 16-bit register.

•**Immediate Value Mismatch**: Using an immediate value (constant) that does not match the size of the destination register.
Example: MOV AX, 0x12345678 where AX is a 16-bit register, but the value is 32 bits.

•**Memory-to-Register Mismatch**: When accessing memory with a size that does not match the register, like loading a 16-bit value into a 32-bit register.
Example: MOV EAX, [BX] where [BX] points to a 16-bit value but EAX is 32-bit.

SIZE MISMATCH ERRORS

•**Pointer Size Mismatch**: In some architectures, registers for addresses (pointers) may differ in size from general-purpose data registers, leading to issues when trying to use them together.
Example: Using a 16-bit pointer in a 32-bit register.

•**Arithmetic Size Mismatch**: Performing arithmetic on registers of different sizes without proper extension or truncation, leading to incorrect results.

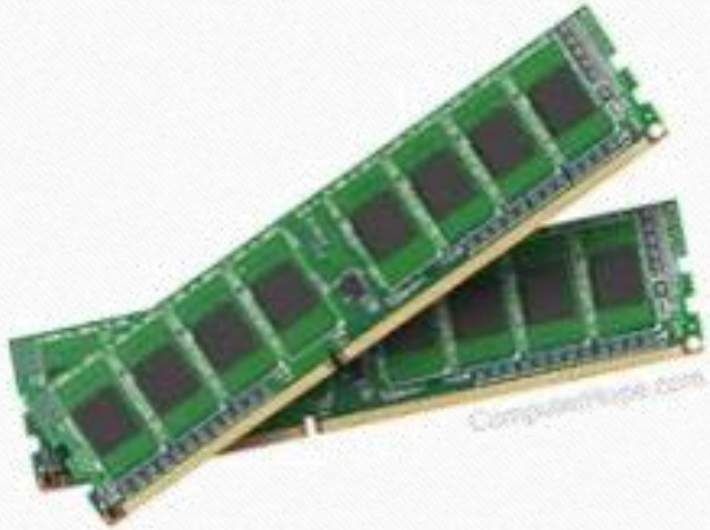
Example: Adding a 16-bit value to a 32-bit register without sign extension

Cache Memory

- **Cache Memory:** A Cache memory is a high-speed memory located near the CPU for frequently accessed data.

Main Memory/ Volatile Memory

- **RAM:** Ram is a primary memory that Holds data and instructions currently being used by the CPU. It's volatile, meaning data is lost when the computer is turned off.
- **Examples are:** DDR1, DDR2, DDR3 & DDR4
  Types
- **1) S-(Static RAM):** don't need proper refresh, faster, more expensive, used as cache memory.
- **2) D- (Dynamic Ram):** need proper refresh, slower than D-Ram , cheap price, used as system memory.



Floppy Drive

- A **Floppy Drive** is an older type of disk drive used to read and write data on **floppy disks**, which are thin, flexible magnetic storage media enclosed in a square plastic case. It was widely used from the late 1970s to the early 2000s for data storage and transfer.
- **Storage Capacity:** Early models: **80 KB to 360 KB**.
- Standard 3.5-inch floppy disks: **1.44 MB**.

Rom(Read Only Memory)

- **ROM (Read-Only Memory)** is a type of non-volatile memory in computers and other electronic devices. It is primarily used to store firmware or permanent data that does not change during the operation of the system. Examples include hard disk drives (HDDs), solid-state drives (SSDs), and optical drives.
- **Key Features of ROM:**
- **Non-Volatile:**
  - Retains data even when the power is turned off.
- **Pre-Programmed:**
  - Data is written during manufacturing and cannot be modified easily (in most cases).
- **Permanent Storage:**
  - Stores essential data such as firmware, BIOS, or bootloader instructions.

Types of ROM

- **Programmable ROM (PROM):**
  - Can be programmed once after manufacturing.
  - Uses a special device to "burn" data.
- **Erasable Programmable ROM (EPROM):**
  - Can be erased and reprogrammed using ultraviolet (UV) light.
  - Transparent window on the chip allows UV light exposure.
- **Electrically Erasable Programmable ROM (EEPROM):**
  - Can be erased and reprogrammed electrically without removing it from the device.
  - Commonly used in devices for firmware updates.
- **Flash Memory:**
  - A modern type of EEPROM that is faster and used in USB drives, SSDs, and memory cards.

CD(Compact Disc)

- **CD (Compact Disc)**
- A Compact Disc (CD) is an optical disc used for storing data. It was initially developed for audio storage but later adapted to hold various types of digital data, such as software, video, and documents. CDs were widely used in the late 20th and early 21st centuries.

USB Storage (Flash Drives)

- **USB Storage (Flash Drives)**
- USB storage refers to portable data storage devices that use flash memory and connect to computers or other devices via a USB (Universal Serial Bus) port. Commonly known as USB flash drives, thumb drives, or pen drives, these devices are widely used for storing and transferring digital data.

Memory Card

Memory Card

- A **memory card** is a small, portable storage device used to store digital data in electronic
- devices. It uses flash memory technology to save and retain data without requiring power.
- Memory cards are widely used in devices such as cameras, smartphones, tablets, gaming consoles, and more.

Cloud Storage

Cloud Storage is a service model that allows users to store data on remote servers accessed over the internet, rather than on local devices like hard drives.

- The data stored in cloud storage can be accessed anytime, anywhere, provided there is an internet connection.
- Cloud storage is scalable, secure, and typically managed by third-party providers.

Memory Hierarchy

- Registers, Cache (L1, L2, L3), Main Memory
- Virtual Memory refers to cloud computing
- Tables: Comparison of different types of memory (speed, size, cost).
- **Primary Memory (RAM):** Primary memory stores temporary data for quick access. Temporary storage for data and programs currently in use.

Memory Hierarchy

- **Secondary Memory (HDD/SSD):** Secondary memory provides long-term storage capabilities.
- **Storage:** Holds data and programs permanently. Examples include hard disk drives (HDDs), solid-state drives (SSDs), and optical drives.

Types:

- **Internal Memory:** HDD, SSD, M2, NVME
- **External Memory:** USB, Memory Card, other portable storage devices
- **Cloud Memory:** Stores Large amount of data on internet on basis of cloud server and fetches data at one click