1.1. ROM (Read-Only Memory) is a type of non-volatile memory that stores data or instructions that are permanently written during manufacturing. Its primary purpose is to hold firmware or software instructions that are essential for the computer's operation.

1.2. RAM (Random-Access Memory) is volatile memory used for temporary data storage during a computer's operation. It allows for fast read and write operations and is used to store data that can be quickly accessed and modified.

Difference: RAM's contents are not permanent and are erased when the computer is powered off.

1.3. Static RAM (SRAM) and Dynamic RAM (DRAM) differ in how they store data.

* SRAM uses flip-flops to store each bit, making it faster but more expensive and power-hungry.
* DRAM stores data if it’s refreshed frequently enough, making it less expensive and more power-efficient but slower.

1.4. USB thumb drives typically use EEPROM flash memory for storage.

Write bit by injecting electrons through a barrier layer (physically damaging it)

2. 32 bits

3. Von Neumann architecture and Harvard architecture are two different computer system designs.

In Von Neumann architecture, both data and instructions share a single memory space.

In Harvard architecture, data and instructions are stored in separate memory spaces.

Von Neumann:

Data and instructions stored in the same location.

Stack central to handling multiple tasks/interrupts.

Harvard:

Separates data and instructions.

Increased efficiency and security

Reduced generality and versatility.

4. Cache memory is a small, high-speed memory unit located between the CPU and main memory (RAM). Its primary role is to temporarily store frequently used data and instructions to reduce the CPU's access time to the main memory, thereby improving overall system performance.

5. An interrupt is a signal sent to the CPU to request its attention. Four common types of interrupts are:

- Hardware interrupts (I/O devices signaling CPU)

- Software interrupts (system calls)

- External interrupts (external hardware events)

- Internal interrupts (divide-by-zero error)

5.1. Polling is an alternative to interrupts where the CPU periodically checks if a device or condition needs attention. Polling is not commonly used because it is less efficient, consumes CPU resources, and can introduce latency in responding to events.

6. A stack is a data structure that follows the Last-In-First-Out (LIFO) principle. It works by pushing data onto the top of the stack and popping data from the top. The primary purpose of a stack is to manage function calls and store local variables and return addresses in a program's execution.

6.1. Stacks are useful for handling interrupts because they allow the CPU to save its current execution context (registers, program counter, etc.) when an interrupt occurs, so it can later resume the interrupted task.

6.2. Stacks are useful in programming for various tasks, including managing function calls and recursion, storing temporary data, and implementing data structures like the stack data structure itself. They help maintain program control flow and manage memory efficiently.

