❖ 1-Bit ALU

➢ Definition:

A 1-bit ALU (Arithmetic Logic Unit) is a fundamental digital circuit that performs arithmetic and logical operations on single-bit inputs. It serves as the basic building block for multi-bit ALUs in processors.

➢ Components of a 1-Bit ALU:

1. Inputs:

- Two operands: A and B (1-bit each).
- Carry-in (Cin) for addition.
- Control signals to select the operation.

2. Outputs:

- Result (R) (1-bit).
- Carry-out (Cout) (for addition).

3. Logic Circuits:

- AND, OR, XOR, and Adder circuits.

➢ Operations Performed:

- Adder circuit uses a Full Adder for addition.
- Logical operations use logic gates.

➢ Importance of a 1-Bit ALU:

- Forms the basic unit of an n-bit ALU (e.g., 8-bit, 16-bit, etc.).
- Used in microprocessors for arithmetic and logic functions.
- Essential for CPU operations, embedded systems, and digital circuits.

➢ Block Diagram:

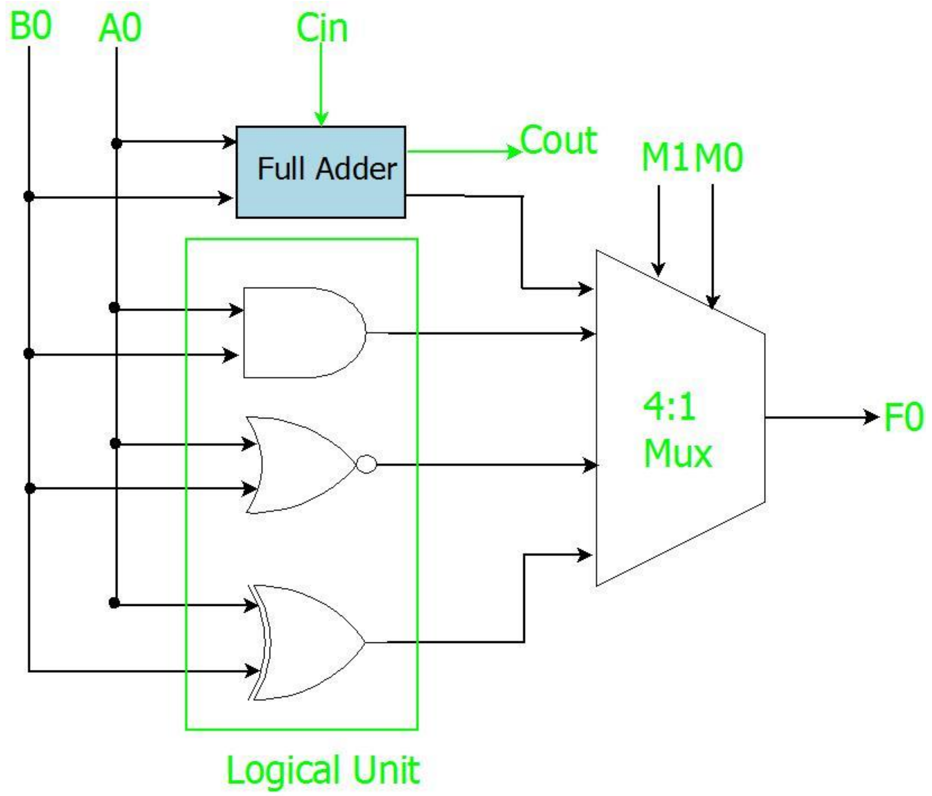

Figure: 1 bit ALU

❖ Memory Hierarchy

Memory hierarchy represents different levels of memory arranged according to speed, size, and cost.

Levels of Memory Hierarchy

1. Registers – Fastest, smallest, used by CPU for immediate execution.
2. Cache Memory – High-speed memory between CPU and RAM.
3. Main Memory (RAM) – Primary volatile memory for active processes.
4. Secondary Storage (HDD/SSD) – Non-volatile, slower than RAM, used for permanent storage.
5. Tertiary Storage (Optical Disks, Magnetic Tape) – Used for backups and long-term storage.

➢ Key Characteristics

- Speed: Registers > Cache > RAM > SSD > HDD > Magnetic Tape.
- Size: Registers < Cache < RAM < SSD < HDD < Tape.
- Cost: Faster memory is more expensive.

❖ Stack Memory

➢ Definition:

Stack memory is a part of RAM that stores temporary data, such as function calls, local variables, and return addresses.

➢ Operations:

- Push: Adds data to the top of the stack.
- Pop: Removes data from the top of the stack.

➢ Features:

- Follows LIFO (Last In, First Out) principle.
- Grows and shrinks dynamically.
- Faster access compared to heap memory.

➢ Usage:

- Function calls and recursion.
- Memory allocation in programming.

❖ **Virtual Memory**

➢ Definition:
Virtual memory is a memory management technique that extends physical RAM using disk storage (swap space) to run large applications efficiently.

➢ Working:

1. When RAM is full, the OS moves inactive data to virtual memory (on the hard disk).

2. When required, the data is swapped back into RAM.

- Advantages:
  - Allows execution of large programs.
  - Provides memory isolation for processes.
- Prevents system crashes due to low RAM.

- Disadvantages:

- Slower than RAM.
- Excessive swapping can reduce performance (thrashing).

❖ **Cache Memory**

Cache memory is a small, high-speed memory located between the CPU and main memory to store frequently used data and instructions.

- Types of Cache Memory

1. L1 Cache (Primary Cache) – Fastest, smallest, located inside CPU.
2. L2 Cache (Secondary Cache) – Larger than L1 but slower.
3. L3 Cache – Shared among multiple cores, larger but slower than L2.

- Mapping Techniques

- Direct Mapping – Each block has a fixed location in the cache.
- Associative Mapping – Data can be placed anywhere in the cache.
- Set-Associative Mapping – Hybrid of direct and associative mapping.

➢ Importance of Cache Memory

Reduces latency and improves CPU performance.
Minimizes the gap between fast CPU and slow RAM.

❖ Direct Memory Access (DMA)

➢ Definition: DMA is a technique that allows peripherals (like hard disks, sound cards) to transfer data directly to/from memory without involving the CPU.

➢ Working of DMA

1. CPU initiates the transfer by sending a request to the DMA controller.
2. DMA controller takes control of the system bus.
3. Data is transferred directly between memory and the device.
4. After completion, DMA informs the CPU via an interrupt.

➢ Advantages of DMA

- Reduces CPU overhead.

- Increases data transfer speed.
- Allows multitasking.

➢ Disadvantages of DMA

- Adds hardware complexity.
- Requires additional control logic.

Block Diagram: