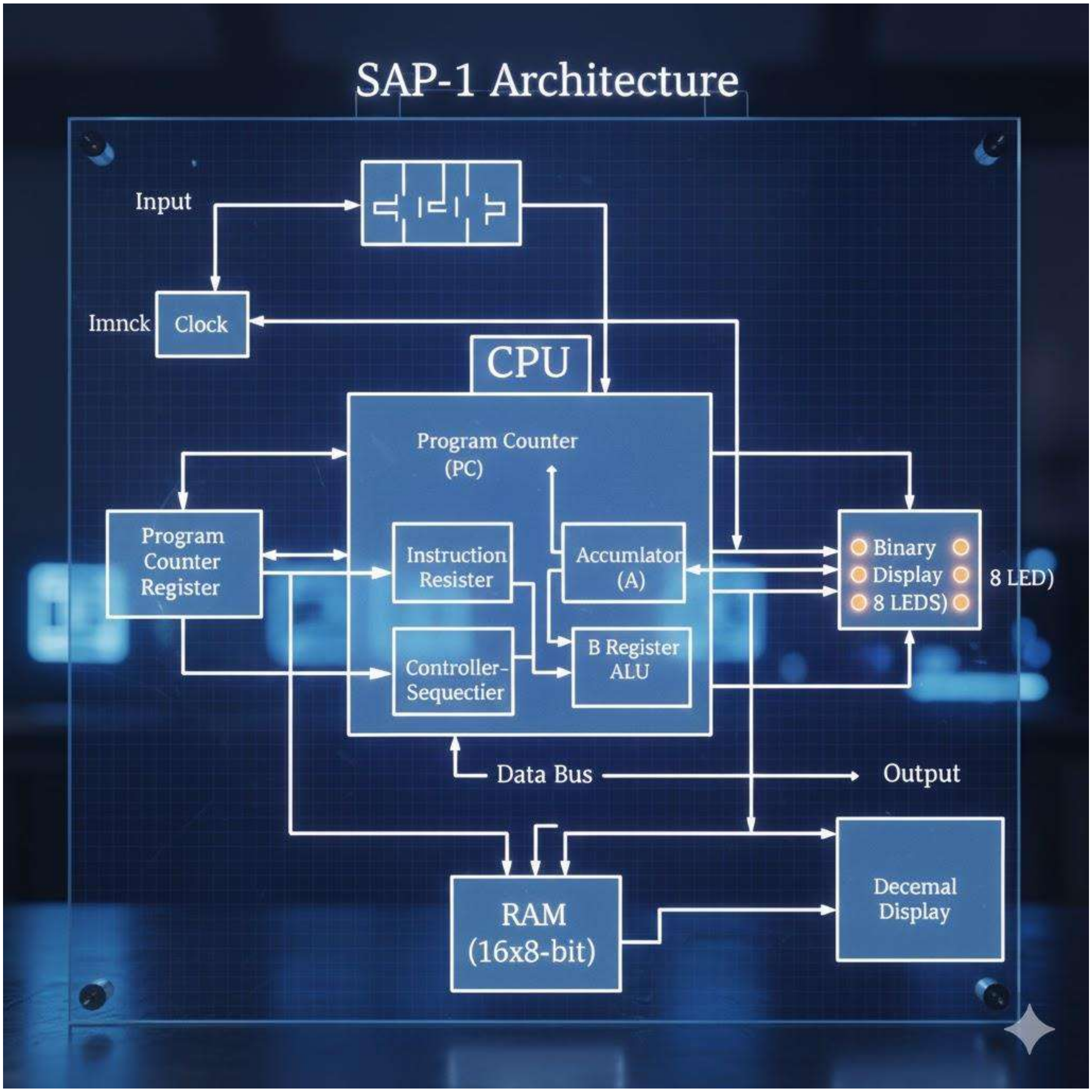
The SAP-1 Architecture



- Prepared By: Mahmoud Saleh Elsayed

1. Introduction

The Simple-As-Possible computer, Model 1 (SAP-1), is an educational, bus-organized processor designed to illustrate the Von-Neumann architecture and the fundamental fetch-execute cycle. It operates using an 8-bit data bus and a limited instruction set, providing a clear, step-by-step model of how a Central Processing Unit (CPU) interacts with memory and performs arithmetic logic operations.

The SAP-1 serves as a conceptual blueprint, simplifying the operation of a full-scale computer into ten main components interconnected by the central bus. Its simplicity ensures an in-depth understanding of the control logic and timing signals required for the stored-program concept.

2. Functional Components and Their Roles

The architecture relies on an 8-bit **W-Bus** for inter-component data transfer, regulated by tri-state buffers to ensure only one component drives the bus at any given clock cycle.

The ten main components and their primary roles are:

- **Program Counter (PC)**: A 4-bit counter (`Counter2`) that holds the memory address of the next instruction to be fetched. It is incremented ($₩text{C}_₩text{p}$ signal) after each instruction fetch.
- **Memory Address Register (MAR)**: A 4-bit register (`Register - 4 bits`) that stores the memory address (either from the PC or the Instruction Register) required to access the RAM.
- **RAM**: The memory unit (`RAM` 16x8) that stores both program instructions and data (emulating the Von-Neumann model).
- **Instruction Register (IR)**: An 8-bit register (`Register`) that stores the fetched instruction. The upper 4 bits are the **Opcode** (for the Control Unit), and the lower 4 bits are the **Operand** (a memory address).
- **Controller/Sequencer (CU)**: The control logic (`Controller`) that generates the necessary 12 control signals (micro-instructions, like $₩text{L}_₩text{a}$, $₩text{E}_₩text{p}$) to coordinate the entire operation in sync with the clock.
- **Accumulator (Reg A)**: An 8-bit register (`Register`) that acts as the primary operand for the ALU and stores intermediate results.
- **B Register (Reg B)**: An 8-bit buffer register (`Register`) that holds the second operand for ALU operations.
- **Adder/Subtractor (ALU)**: The arithmetic logic unit (`ALU`, constructed from `FA`/`One`/`ZeroCircuit` sub-circuits) that performs 8-bit addition and subtraction (using two's complement) on the contents of Reg A and Reg B.
- **Output Register (OUT)**: An 8-bit register (`Register`) that latches the final result from the bus for display.

3. Step-by-Step Operation

The SAP-1 executes a program through a repetitive **Instruction Cycle**, which is divided into a **Fetch Phase** and an **Execute Phase**, coordinated by the Control Unit's internal ring counter (T0-T5).

1. The Fetch Phase (T0, T1, T2)

This phase retrieves the next instruction from memory and loads it into the Instruction Register.

- **T0 (Address Transfer):** The Program Counter ($₩text{PC}$) places the address onto the W-Bus using the $₩text{E}_₩text{p}$ signal. The Memory Address Register ($₩text{MAR}$) simultaneously loads this address using the $₩text{L}_₩text{m}$ signal.
- **T1 (PC Increment):** The PC is synchronously incremented using the $₩text{C}_₩text{p}$ signal, preparing for the next instruction address.
- **T2 (Instruction Load):** The 8-bit instruction is fetched from RAM (addressed by MAR) and placed onto the W-Bus via the $₩text{C}_₩text{E}$ signal. The Instruction Register ($₩text{IR}$) loads the instruction using the $₩text{L}_₩text{i}$ signal.

2. The Decode and Execute Phase (T3, T4, T5)

The Opcode (upper 4 bits) of the instruction in the IR is sent to the Control Unit, which decodes it to determine the required micro-instructions for the execution phase.