

The Stored Program

Shaan Fulton

July 4, 2025

The first general-purpose digital computer was the ENIAC. To program the ENIAC, programmers would rewire patch cables to link up different logical switches and augment execution order. The ENIAC did have memory, but it wasn't used to store instructions, only data. It had 20 accumulators, which consisted of binary vacuum tubes. Each represented one 10-digit decimal number.

Von Neumann Architecture

John von Neumann (among others) recognized that operations could be minimized to a select few composable options whose execution order could be stored in memory, thus creating digitally reprogrammable computers whose instructions were written in bits.

Von Neumann architecture is the modern standard. **It is an excellent example of early computer abstraction:** The programmer should not have to worry about timing, only execution order. By abstracting away the manual rewiring of logical operations, you not only made it faster to program a computer—you removed a layer of unnecessary detail, allowing for more mental bandwidth to be reserved for the actual computation design.

The Stored Program

Before stored programs, memory was only used to store data. The processor would have a minimal instruction set, but these instructions and their execution order would be manually wired in. The key difference with a von Neumann architecture is that instructions were now stored in memory, and

their execution was facilitated by a control unit and timed by a program counter and clock.

Component	Role
Memory	Stores both instructions and data in a single address space.
ALU	Performs arithmetic and logic operations.
Control Unit	Fetches, decodes, and directs execution of instructions.
Registers	Fast, temporary storage for immediate values and operands.
Program Counter	Tracks the address of the next instruction to execute.
Clock	Provides timing signals to coordinate operations.

Table 1: Core Components of the von Neumann Architecture