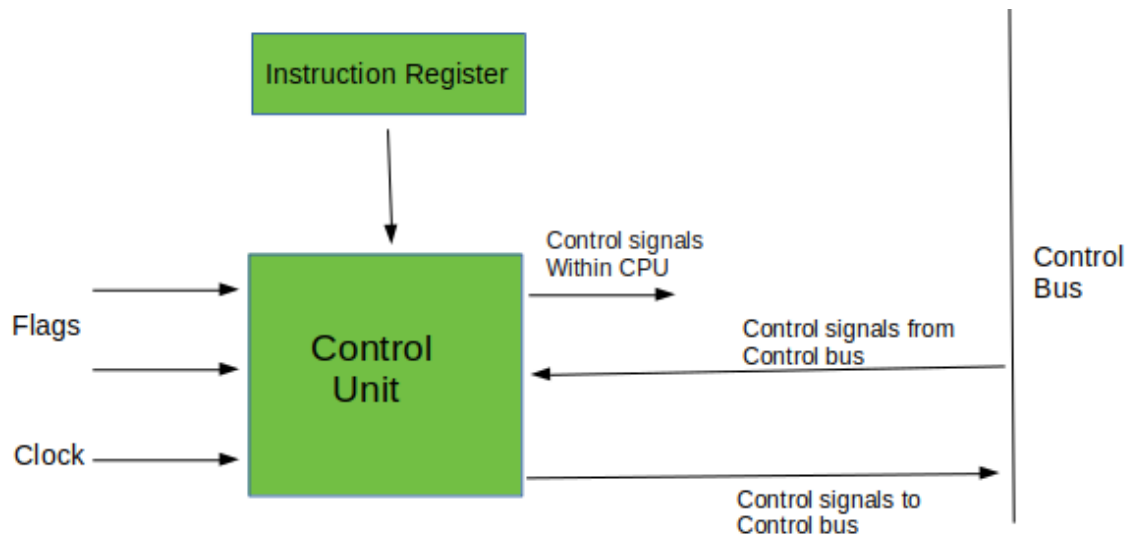


## Control Unit (CU) Architecture of a Computer:

The Control Unit (CU) is responsible for managing and coordinating the activities of the entire CPU. It does this by interpreting the instructions from memory and generating the necessary control signals to direct the operation of other CPU components like the Arithmetic Logic Unit (ALU), registers, and memory. The CU is a key part of the CPU that ensures the proper execution of the Fetch-Decode-Execute cycle. The Control Unit is the part of the computer's central processing unit (CPU), which directs the operation of the processor. **It was included as part of the Von Neumann Architecture by John von Neumann.** It is the responsibility of the control unit to tell the computer's memory, arithmetic/logic unit, and input and output devices how to respond to the instructions that have been sent to the processor. It fetches internal instructions of the programs from the main memory to the processor instruction register, and based on this register contents, the control unit generates a control signal that supervises the execution of these instructions. A control unit works by receiving input information which it converts into control signals, which are then sent to the central processor. The computer's processor then tells the attached hardware what operations to perform. The functions that a control unit performs are dependent on the type of CPU because the architecture of the CPU varies from manufacturer to manufacturer.

### Main Components of the Control Unit:

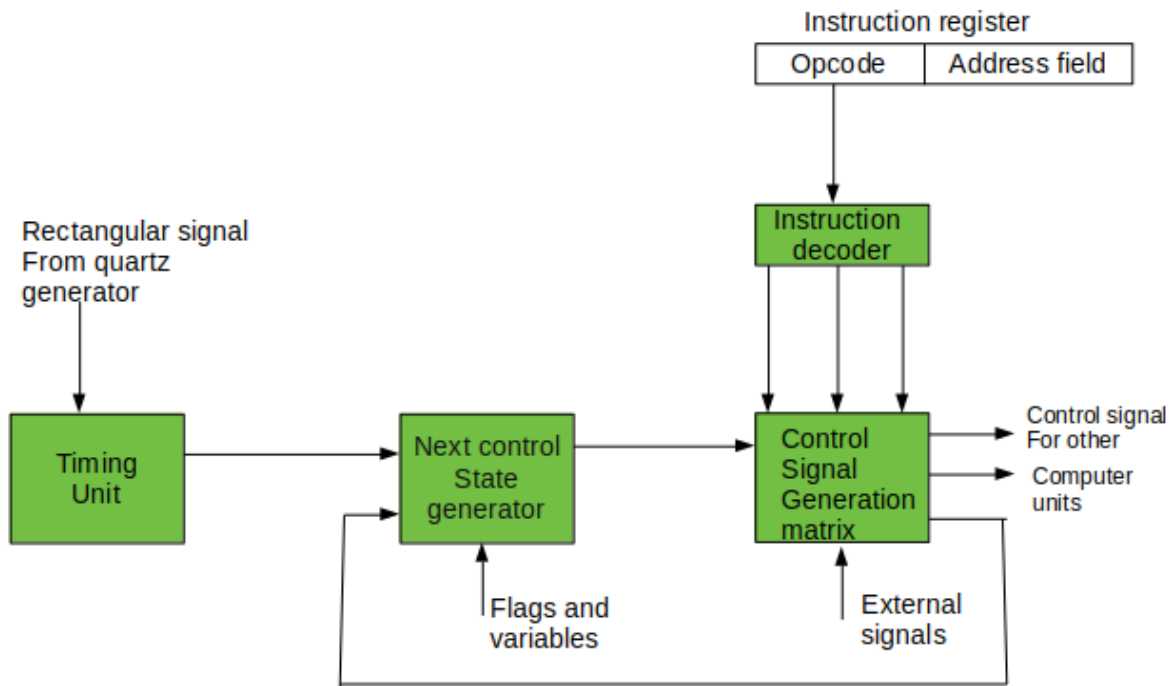
1. **Instruction Register (IR):**
  - Holds the current instruction that is being decoded and executed by the CU.
2. **Decoder:**
  - Decodes the opcode (operation code) from the instruction in the IR to determine what action is needed.
3. **Control Logic:**
  - Generates control signals based on the decoded instruction. These signals direct other CPU components like the ALU, memory, and registers.
4. **Program Counter (PC):**
  - Keeps track of the address of the next instruction to be fetched from memory.
5. **Sequencer:**
  - Manages the order of execution, ensuring the proper flow of the instruction cycle (fetch, decode, execute).
6. **Control Register (Control Word Register):**
  - Stores the control signals generated by the Control Logic, which govern the operation of the CPU components.
7. **ALU (Arithmetic Logic Unit):**
  - Although not strictly part of the CU, the ALU performs arithmetic and logical operations based on the control signals it receives from the CU.



Block Diagram of the Control Unit

**In the Hardwired control unit**, the control signals that are important for instruction execution control are generated by specially designed hardware logical circuits, in which we can not modify the signal generation method without physical change of the circuit structure. The **operation code of an instruction contains the basic data for control signal generation**. In the instruction decoder, the operation code is decoded. The instruction decoder constitutes a set of many decoders that decode different fields of the instruction opcode.

As a result, **few output lines going out from the instruction decoder obtains active signal values**. These output lines are connected to the inputs of the matrix that generates control signals for execution units of the computer. **This matrix implements logical combinations of the decoded signals from the instruction opcode with the outputs from the matrix that generate signals representing consecutive control unit states and with signals coming from the outside of the processor, e.g. interrupt signals**. The matrices are built in a similar way as programmable logic arrays.



A **Microprogrammed Control Unit** (MCU) is a type of control unit used in computer systems to generate the necessary control signals for various parts of the processor. Unlike a hardwired control unit, which uses a fixed set of logic circuits to produce control signals, a microprogrammed control unit uses a set of instructions, called **microinstructions**, stored in memory. These microinstructions define the steps for controlling the operations of the CPU.

### Key Concepts:

1. **Microinstructions:** Microinstructions are small, low-level instructions that directly control the various components of the processor, such as the ALU (Arithmetic Logic Unit), registers, buses, and memory. A microinstruction typically represents a single step in executing an instruction (like moving data, performing arithmetic operations, or fetching instructions).
2. **Control Memory:** Microprogrammed control uses a dedicated memory (often called **control memory**) to store the microinstructions. This memory could be either **ROM (Read-Only Memory)** or **RAM**, depending on the system's needs. The control memory is addressed sequentially, and the microinstructions are fetched and executed in sequence to generate the required control signals.
3. **Microprogram:** A sequence of microinstructions that defines how a higher-level machine instruction is executed is called a **microprogram**. When the processor executes an instruction, the control unit fetches the corresponding microprogram from control memory and carries out the necessary steps.

4. **Control Address Register (CAR):** The Control Address Register holds the address of the current microinstruction in the control memory. After fetching a microinstruction, the CAR is updated to the address of the next microinstruction.
5. **Micro-Operation:** Each microinstruction corresponds to a specific micro-operation, which is a fundamental operation on registers or memory. These micro-operations could involve loading data into registers, performing an arithmetic operation, or moving data between the processor and memory.

### Structure of a Microprogrammed Control Unit:

1. **Sequencer:** The sequencer is responsible for selecting the correct microinstruction based on the current state of the processor. The address of the next microinstruction can either be fixed (sequential execution) or based on a condition (branching).
2. **Control Memory (ROM):** This memory stores all the microinstructions. Each instruction in the machine language corresponds to a sequence of microinstructions stored in control memory.
3. **Microinstruction Register (MIR):** The MIR holds the current microinstruction that is being executed. It decodes the microinstruction and issues control signals to the various components of the CPU.
4. **Decoder:** The decoder interprets the microinstruction stored in the MIR and generates the specific control signals needed to perform the operation.

### Advantages of Microprogrammed Control:

1. **Flexibility:** Microprogramming allows the control unit's behavior to be easily changed by altering the microprogram in memory. This can be useful for modifying instruction sets or adding new instructions.
2. **Simplicity in Design:** It is simpler to design a microprogrammed control unit than a hardwired control unit, especially for complex instruction sets.
3. **Easier to Modify:** Updating the control unit to support new features or to fix bugs can be done by modifying the microprogram, rather than redesigning the entire logic circuit.
4. **Reduced Hardware Complexity:** Instead of a complex network of logic gates, the microprogrammed control unit relies on memory and simpler hardware, which can reduce the overall complexity of the control unit.

### Disadvantages of Microprogrammed Control:

1. **Slower Execution:** Since the control unit fetches microinstructions from memory (often slower than directly generated hardwired control signals), the execution speed may be slower compared to hardwired control units.
2. **Memory Overhead:** Storing microinstructions in memory requires additional memory space, which can increase the cost and complexity of the system.
3. **Complexity in Design of Microprogram:** While the design is flexible, writing and organizing the microprogram for complex instructions can become quite tricky.

