Hardwired Control Design

- To execute instructions, the processor must have some means of generating the required **control signals** in **proper sequence**.
- The two categories to do this are **hardwired control design/unit** and **micro programmed control design/unit**.
- A **hardwired control unit** is implemented using **hardware**; **decoder**, **counter**, and **logic gates**.
- The **components** of the circuit are **physically wired**/connected, hence called **hardwired**.
- It is called a **sequence counter** used to develop control signals in a sequential order.
- It is a **state machine** that changes from one state to another in every clock cycle.

See Zaky (Chapter 7: Basic Processing Unit)

Hardwired Control Design

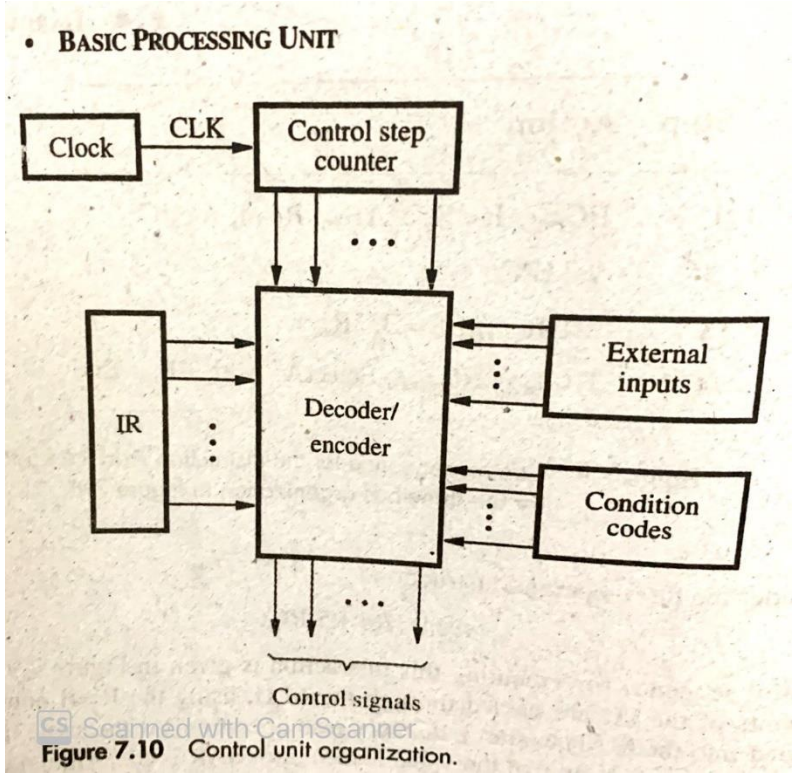

Figure 7.10 Control unit organization.

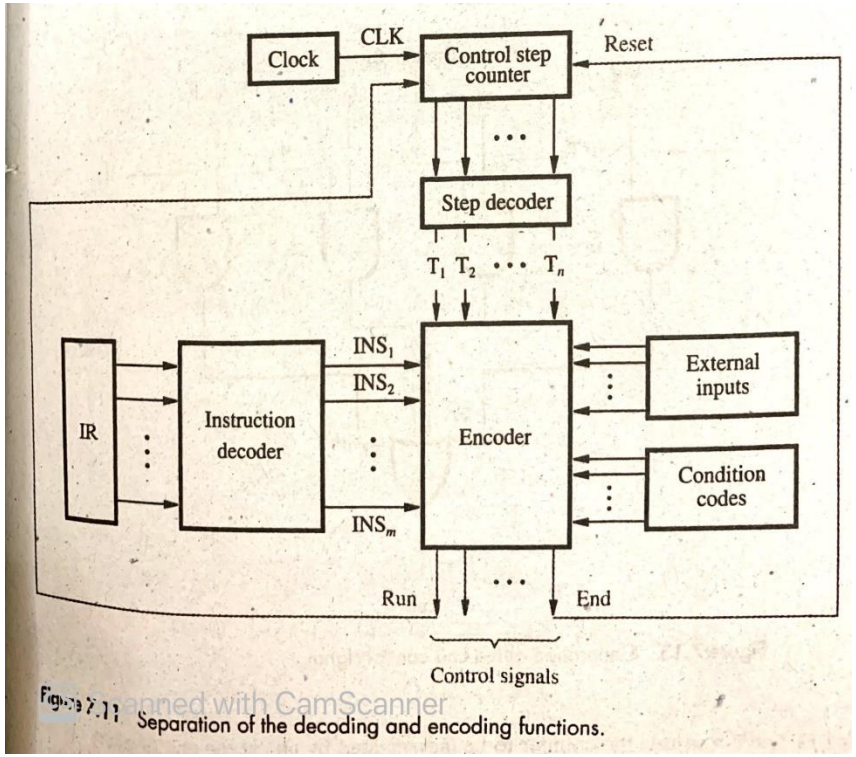

Figure 7.11 Separation of the decoding and encoding functions.

Hardwired Control Design

- A sequence of control signals is generated to execute an instruction. The working procedure of the hardwired control unit, as shown in the figure, is as follows:
  - The **decoder/encoder block** is a **combinational circuit** that generates the required control outputs, depending on the state of all its inputs.
  - The **counter** is used to keep track of the control steps.
  - The **step decoder** provides a separate signal line for each time slot in the control sequence.
  - The instruction that is loaded in the IR is decoded by the instruction decoder. If the IR is an 8-bit register, then the **instruction decoder** generates 2^8 (256) output lines, separate line for each machine instruction.
  - For any instruction loaded in the **IR**, one of the output lines INS_1 through INS_m is set to 1, and all other lines are set to 0.
  - The inputs to the **encoder** are combined (from the instruction step decoder, external inputs, and condition codes) to generate individual control signals.
  - **External input signals** can be from an external interrupting source or a device requesting access to an external bus. **Conditions codes** are as per condition and status flags set in earlier instructions. Condition codes are extra bits kept by a processor that summarize the results of an operation and that affect the execution of later instructions.
  - The **end signal** is generated after all the instructions get executed. This signal starts a new instruction fetch cycle by resetting the control step counter to its starting value.

Hardwired Control Design

- A control unit that uses this approach can operate at **high speed**.
- But, it has **little flexibility**, and the **complexity of instruction** set it can **implement** is **limited**.

Micro Programmed Control Design

- In **micro programmed control design/unit**, **control signals** are **generated** by a **program** similar to machine language program.
- A few basic terms
  - **Control word (CW)**: A word whose individual bits represent the various control signals. Each of the control steps in the control sequence of an instruction defines a unique combination of 1s and 0s in the CW.
  - **Micro routine**: A sequence of control words corresponding to the control sequence of a machine instruction constitutes the micro routine for that instruction.
  - **Micro instruction**: Individual control words in a micro routine are referred to as micro instructions.
  - **Control store**: The micro routines for all instructions in the instruction set of a computer are stored in a special memory called the control store.

Micro Programmed Control Design

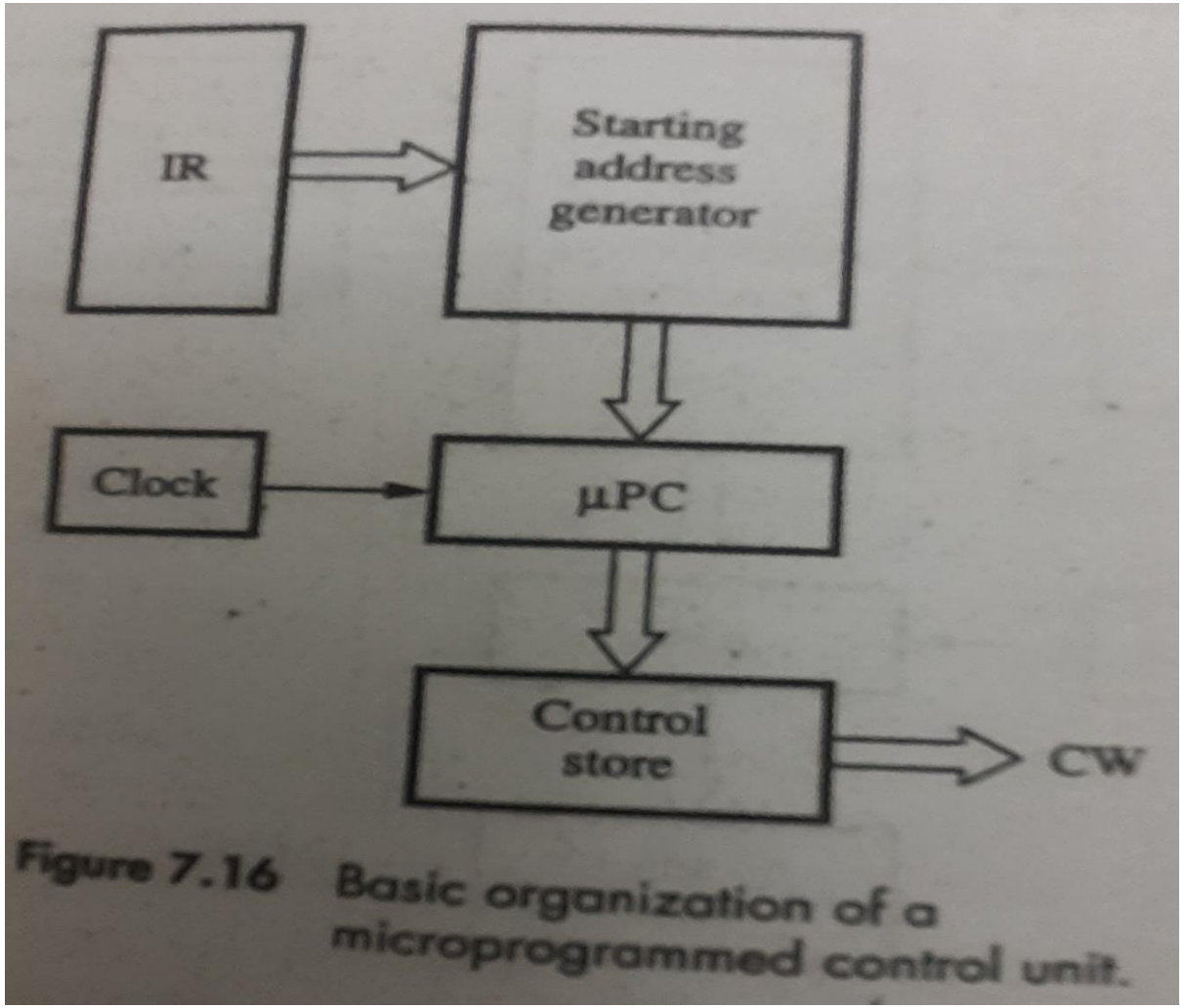

Figure 7.16 Basic organization of a microprogrammed control unit.

Micro Programmed Control Design

- The **micro programmed control design**, as shown in the figure, **approaches** as follows:
  - The control unit can generate the control signals for any instruction by sequentially reading the CWs of the corresponding micro routine from the control store.
  - To read the control words sequentially from the control store, micro program counter (µPC) is used.
  - Every time a new instruction is loaded into the IR, the output of the block labeled “Starting address generator” is loaded into the µPC.
  - The µPC is then automatically incremented by the clock, causing successive micro instructions to be read from the control store.
  - Hence, the control signals are delivered to the various parts of the processor in the correct sequence.

Micro Programmed Control Design

- When the micro programmed control unit requires to check the status of the condition codes or external inputs to choose between alternative action (Figure 7.18), it uses conditional branch micro instructions.
- In addition to the branch address, these micro instructions specify which of the external inputs, condition codes, or possibly, bits of the instruction should be checked as a condition for branching to take place.

I/O Control

- One of the basic features of a computer is its ability to exchange data with other devices.
- I/O control can be defined as any set of programs, operations, and/or devices that transmit data between computer systems or other I/O devices.
- I/O controllers communicate with external devices connected to the computer. They assist with data transfer and exchange data between the processor and other components within a computer.
- I/O controllers play a bridging role between CPU, Memory and I/O Device by taking care of all kinds of communication.
- The main purpose of this system is to help in the interaction of peripheral devices with the control units (CUs).
- There are major three types of I/O control:
  - Programmed I/O
  - Interrupt based I/O
  - Direct Memory Access (DMA)

See Stallings (Chapter: Input/Output)

Programmed I/O

- The **programmed I/O method** controls the transfer of data between the connected devices/peripherals and the computer.
  - The programmed I/O is the most simple type of I/O technique for the exchanges of data or any types of communication between the processor and the external devices.
  - Each data item transfer is initiated by an instruction in the program, involving the CPU for every transaction.
  - When the processor is executing a program and encounters an instruction relating to input/output, it executes that instruction by issuing a command to the appropriate input/output module. With the programmed input/output, the input/output module will perform the required action and then set the appropriate bits in the input/output status register. **The input/output module takes no further action to alert the processor. In particular it doesn't interrupt the processor. Thus, it is the responsibility of the processor to check the status of the input/output module periodically, until it finds that the operation is complete.**
  - Data transfer through this mode requires constant monitoring of the peripheral device by the CPU and also monitor the possibility of new transfer once the transfer has been initiated. Thus CPU stays in a loop until the I/O device indicates that it is ready for data transfer.
  - Thus programmed I/O is a time consuming process that keeps the processor busy needlessly and leads to wastage of the CPU cycles.
  - This situation can be avoided by using an interrupt facility.

Programmed I/O

- To execute an I/O related instruction, the processor issues an address, specifying the particular I/O module and external device, and an I/O command. The basic four types of **I/O commands** are
  - **Control**: used to activate a peripheral and tell it what to do.
  - **Test**: used to test various test status conditions associated with an I/O module and its peripherals.
  - **Read**: causes the I/O module to obtain an item of data from the peripheral and place it in an internal buffer (data register). The processor then can obtain the data item by requesting that the I/O module place it on the data bus.
  - **Write**: causes the I/O module to take an item of data from the data bus and subsequently transmit that data item to the peripheral.

Programmed I/O

- **Memory mapped I/O**
  - A programmed I/O.
  - Memory-mapped I/O **uses the same address space to address both memory and I/O devices**. The memory and registers of the I/O devices are mapped to (associated with) address values.
  - Thus,. **the CPU instructions used to access the memory can also be used for accessing devices**
  - Each I/O device monitors the CPU's address bus and responds to any CPU access of an address assigned to that device, connecting the data bus to the desired device's hardware register.
  - To accommodate the I/O devices, areas of the addresses used by the CPU must be reserved for I/O and must not be available for normal physical memory. The reservation may be permanent, or temporary.
  - For example, with 10 address lines, a combined total of 2^{10}=1024 memory locations and I/O addresses can be supported.
  - A single read line and a single write line are needed on the bus.

Programmed I/O

- **Isolated I/O (also called Port-mapped I/O)**
  - Here, **the bus is equipped with memory read and write lines plus I/O lines**.
  - So, now the line specifies whether the address refers to a memory location or an I/O device.
  - Hence, with 10 address lines, the system may now support **1024 memory locations and 1024 I/O addresses**.
  - Because **the address space for I/O is isolated from that for main memory, this is referred to as isolated I/O**.