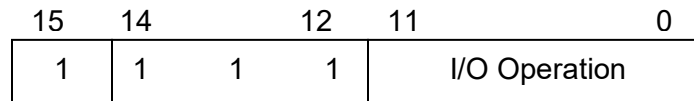


Mnemonic	Description
CME	Complement E bit
INC	Increment AC register
SPA	Skip next instruction if contents of AC register are + ve

Input-Output Instructions - Like register reference instructions, input-output instructions do not need memory reference. The Opcode and I bit for these instructions are 111 and 1, respectively. The remaining 12-bits specify the type of input-output operation or test to be performed.



Example:

Mnemonic	Description
INP	Load a character in AC register from input port
OUT	Send a character to output port from AC register

Microinstruction Format

Figure 1 shows the microinstruction format for the control memory. The microinstruction includes four fields.

1. F1, F2 and F3: These are micro-operation fields. Each field is of three bits. They specify micro-operations for the Computer.
2. CD: This two-bit field selects status bit conditions for branching operation. The condition includes zero value in AC, sign bit of AC equal to 1 or 0, etc.
3. BR: This 2-bit field specifies the type of branch to be used. Branch types include unconditional branch, branch if zero, and branch if negative and so on.
4. AD: This is an address field which contains a branch address. This field is of seven bits since control memory has 128 words. ($128 = 2^7$).

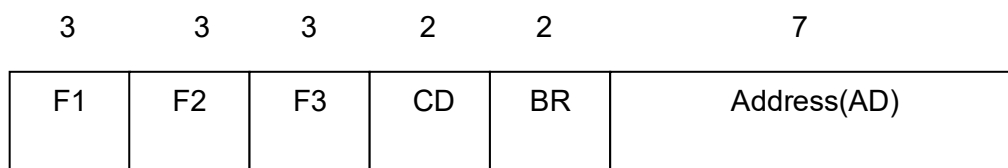


Figure 1: Microinstruction format

- The micro-operations are subdivided into three fields as F1, F2 and F3 each is of 3-bits.
- The three bits in each field are encoded to specify seven distinct micro-operations.
- Thus there are 21 micro-operations.
- A microinstruction includes three micro- operations, one from each field. So, no more than three micro-operations can be selected for a microinstruction.
- If the microinstruction needs micro-operations less than three, one or more of the micro—operation fields will be filled by a binary code 000 for no operation.

Table 1 Micro-instructions with their binary code, micro-operation and symbol Form

F1	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC + DR$	ADD
010	$AC \leftarrow 0$	CLRAC
011	$AC \leftarrow AC + 1$	INCAC
100	$AC \leftarrow DR$	DRTAC
101	$AR \leftarrow DR(0-10)$	DRTAR
110	$AR \leftarrow PC$	PCTAR
111	$M[AR] \leftarrow DR$	WRITE

F2	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC - DR$	SUB
010	$AC \leftarrow AC \vee DR$	OR
011	$AC \leftarrow AC \wedge DR$	AND
100	$DR \leftarrow M[AR]$	READ
101	$DR \leftarrow AC$	ACTDR
110	$DR \leftarrow DR + 1$	INCDR
111	$DR(0-10) \leftarrow PC$	PCTDR

F3	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC \oplus DR$	XOR
010	$AC \leftarrow AC'$	COM
011	$AC \leftarrow \text{shl } AC$	SHL
100	$AC \leftarrow \text{shr } AC$	SHR
101	$PC \leftarrow PC + 1$	INCPC
110	$PC \leftarrow AR$	ARTPC
111	Reserved	

FETCH AND EXECUTION CYCLE

Instruction Cycle

- The simplest model of instruction processing can be a two step process. The CPU reads (fetches) instructions (codes) from the memory one at a time, and executes.
- Instruction fetch involves reading of an instruction from a memory location to the CPU register.
- The execution of this instruction may involve several operations depending on the nature of the instruction.
- Instructions are processed by the control unit in a systematic, step-by-step manner.

The most basic unit of computer processing in the simplest form consists of two parts.

1. **Opcode** (operation code) – A portion of a machine language instruction that specifies the operation to be performed.
2. **Operands** – A part of a machine language instruction that specifies the data to be operated on.

The sequence of steps in which instructions are loaded from memory and executed is called instruction cycle. Each step in the sequence is referred to as a phase. Fundamentally, there are 6 phases.

1. **FETCH (instruction)** - This phase obtains the next instruction from memory and stores it in the IR. The address of the next instruction to be executed is stored in the PC register.

Proceeds in the following manner

$MAR \leftarrow PC$ (memory address register is loaded with the content of PC).

$PC \leftarrow PC + 1$ (value stored in the PC is incremented by one)

$MDR \leftarrow MEM [MAD]$ (interrogate memory, resulting in the instruction being placed in the MDR).

$IR \leftarrow MDR$ (load the instruction from MDR to the instruction register).

For now, we will say that each of these steps proceeds in one *machine cycle*. Note that the instruction to be executed is now stored in IR and the address of the *next* instruction to be executed is stored in PC

2. **DECODE** - In this phase the instruction stored in PC is examined in order to decide what portion of the microarchitecture needs to be involved in the execution of the instruction.
3. **EVALUATE ADDRESS** - This phase Compute the address of the memory location that is needed to process the instruction. Some instructions do not need this phase, e.g., instructions that work directly with the registers and do not require any operands to be loaded or stored form memory.
4. **FETCH OPERANDS** - In this phase, the source operands needed to carry out the instruction are obtained from memory. For some instructions, this phase equals to loading values form the register file. For others, this phase involves loading operands from memory.
5. **EXECUTE** - In this phase instruction is carried out. Some instructions may not require this phase, e.g., data movement instructions for which all the work is actually done in the FETCH OPERANDS phase 6
6. **STORE RESULTS** - In this phase the result is written to its designated destination.

After 6 phases of the instruction cycle are done, the control unit begins the next instruction cycle, starting with the new FETCH (instruction) phase. Since the PC was previous incremented by one, it contains the pointer to the next instruction to be fetched and executed

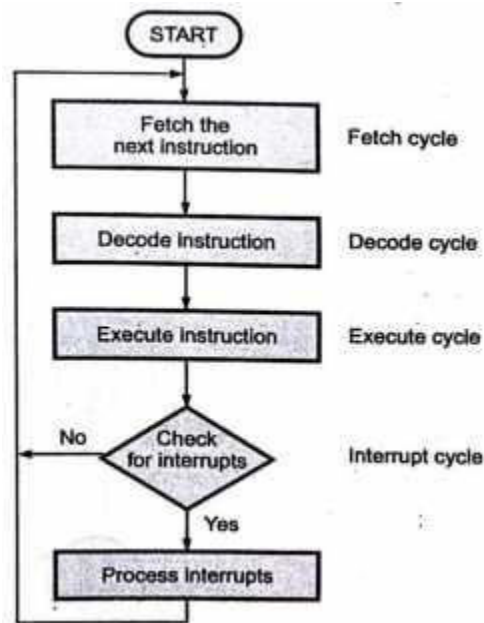


Figure 2: Basic instruction cycle with interrupt cycle

Actually, CPU checks for valid interrupt request after each instruction cycle. If any valid interrupt request is present, CPU saves the current process state, and services the interrupt. Servicing the interrupt means executing interrupt service routine. After completing it, CPU starts the new Instruction cycle from where it has been interrupted. Figure 2 shows this instruction cycle with interrupt cycle.

Determination of type of Instruction

During decoding phase if decoder output $D_7 = 0$, it is memory reference instruction; otherwise, it is as register reference or I/O instruction. According to D_7 and I bits, different instructions are executed listed in Table 2.

Table 2: Instruction Execution

D7	I	Instruction Executed
0	0	Memory Reference Instruction with a Direct Address
0	1	Memory Reference Instruction with an Indirect Address
1	0	Register Reference Instruction
1	1	I/O Instruction

HARDWIRED CONTROL UNIT

- In the hardwired organization, the control logic is implemented with gates, flip-flops, decoders, and other digital circuits. It has the advantage that it can be optimized to produce a fast mode of operation.
- A hardwired control, as the name implies, requires changes in the wiring among the various components if the design has to be modified or changed.
- The control signals are generated in the control unit and provide control inputs for the multiplexers in the common bus, control inputs in processor registers, and micro-operations for the accumulator.
- In the micro-programmed control, any required changes or modifications can be done by updating the micro-program in control memory. A hardwired control for the basic computer is presented in this section.

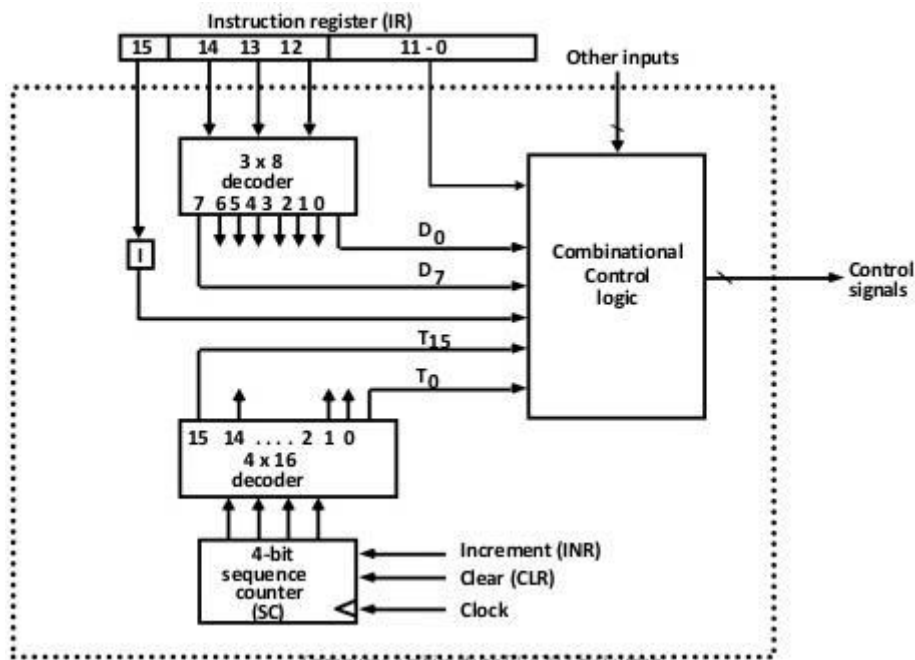


Figure 3: Hardwired Control Unit of Basic Computer

In the hardwired control, the control units use fixed logic circuit to interpret instructions and generate control signals from them.

The block diagram of the control unit is shown in Figure 3.

- It consists of two decoders, a sequence counter, and a number of control logic gates. An instruction read from memory is placed in the instruction register (IR).
- The instruction register is divided into three parts: the I bit, the operation code, and bits 0 through 11.
- The operation code in bits 12 through 14 are decoded with a 3 x 8 decoder.
- The eight outputs of the decoder are designated by the symbols D0 through D7.
- The subscripted decimal number is equivalent to the binary value of the corresponding operation code.
- Bit 15 of the instruction is transferred to a flip-flop designated by the symbol 1.
- Bits 0 through 11 are applied to the control logic gates.
- The 4-bit sequence counter can count in binary from 0 through 15.
- The outputs of the counter are decoded into 16 timing signals T₀ through T₁₅.
- The sequence counter SC can be incremented or cleared synchronously.
- Once in a while, the counter is cleared to 0, causing the next active timing signal to be T₀.

Advantages of Hardwired Control Unit

1. Hardwired control unit is fast because control signals are generated by combinational circuits.
2. The delay in generation of control signals depends upon the number of gates.
3. It has greater chip area efficiency since it uses less area on-chip.

Disadvantages of Hardwired Control Unit

1. More the control signals required by CPU; more complex will be the design of control unit.
2. A modification in control signal is very difficult. That means it requires rearranging of wires in the hardware circuit.
3. It is difficult to correct mistake in original design or adding new feature in existing design of control unit.

MICRO-PROGRAMMED CONTROL UNIT

- A computer that uses a micro-programmed control unit usually has two separate memories - a **main memory and a control memory**.
- The main memory is available to the user for storing their programs.
- The contents of main memory may change when the data are manipulated and every time the program is changed.
- The user's program in main memory consists of machine instructions and data, whereas, the control memory holds a fixed micro-program that cannot be altered by the occasional user.
- The micro-program consists of micro-instructions that specify various internal control signals for execution of register micro-operations.
- Each machine instruction initiates a series of microinstructions in control memory.
- These microinstructions generate the micro-operations to fetch the instruction from main memory; to evaluate the effective address, to execute the operation specified by the instruction, and to return control to the fetch phase in order to repeat the cycle for the next instruction.

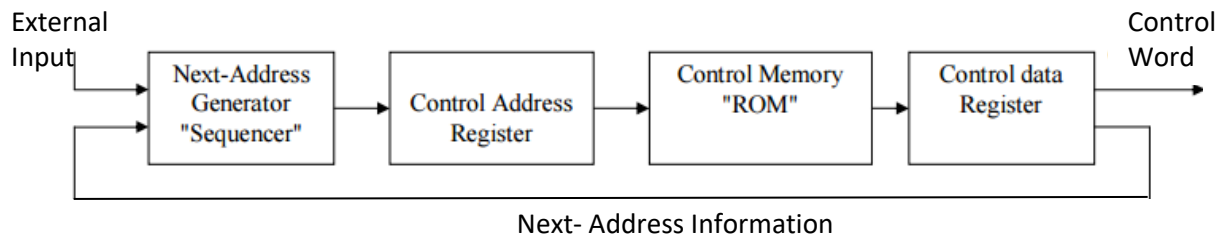


Figure 4: Micro-programmed Control Organization

The general configuration of a micro-programmed control unit is demonstrated in Figure 4 which is consists of Address Sequencer, Control Address Register, Control Memory and Control data Register.

- The next address generator is sometimes called a Micro-program Sequencer, as it determines the address sequence that is read from control memory. The address of the next microinstruction can be specified in several ways, depending on the sequencer inputs. Typical functions of a micro-program sequencer are incrementing the control address register by one, loading into the control address register an address from control memory, transferring an external address, or loading an initial address to start the control.
- The control memory address register specifies the address of the microinstruction, and the control data register holds the microinstruction read from memory.
- The control memory is assumed to be a ROM, within which all control information is permanently stored.

Table 3 Comparison between Hardwired and Micro programmed Control

Attribute	Hardwired Control	Micro programmed Control
Speed	Fast	Slow
Control Functions	Implement in Hardware	Implement in Software
Flexibility	Not flexible, to accommodate new system specifications or new instructions.	More flexible, to accommodate new system specifications or new instructions redesign is required.
Ability to handle large/complex Instruction sets	Somewhat difficult	Easier
Ability to support Operating Systems and diagnostic features	Very difficult	Easy
Design Process	Somewhat complicated	Orderly and Systematic
Application	Mostly RISC Microprocessor	Mainframes Microprocessor
Instruction Set Size	Usually under 100 Instructions	Usually above 100 Instructions
ROM Size	-	2 K to 10 K
Chip Area Efficiency	Uses Least Area	Uses More Area

Advantages of micro programmed control

1. It simplifies the design of control unit, Thus it is both, cheaper and less error prone to implement.

2. Control functions are implemented in software rather than hardware.
3. The design process is orderly and systematic.
4. More flexible and can be changed to accommodate new system specifications or to correct the design errors quickly and cheaply.
5. Complex function Such as floating point arithmetic can be realized efficiently

Disadvantages of micro programmed control

1. A micro programmed control unit is somewhat slower than the hardwired control unit, because time is required to access the microinstructions from CM.
2. The flexibility is achieved at some extra hardware cost due to the control memory and its access circuitry.
3. Besides these disadvantages, the microprogramming is the dominant technique for implementing control.

MICRO-PROGRAM SEQUENCER

- The subunit of the micro programmed control unit which presents an address to the control memory is called micro program sequencer.
- The next-address logic of the sequencer determines the specific address source to be loaded into the control address register.
- The Figure 5 shows the block diagram of commercial micro program sequencer.
- It consists of a multiplexer that selects an address from four sources and routes it into a control address register.
- The output from CAR provides the address for the control memory.
- The contents of CAR are incremented and applied to the multiplexer and to the stack register file.
- The register selected in the stack is determined by the stack pointer.
- Inputs I_2, I_1, I_0 and T derived from the CD and BR fields of microinstruction specify the operation for the sequencer. They specify the input source to the multiplexer also generate push and pop signals required for stack operation.
- The stack pointer is a , three-bit register and it gives the address of stack register file consists of ($2^3 = 8$) eight registers Initially, the stack pointer is cleared and is said to point at address 0 in the stack.
- Using push operation it is possible to write data into the stack at the address specified by the stack pointer. After data is written, stack pointer is incremented by one to get ready for the next push operations.
- In pop operation stack pointer is decremented by one and then the contents of the register specified by the new value' of stack pointer are read. With this mechanism it is possible to implement subroutine calls.
- During subroutine call the incremented address (the address of the next instruction) is stored in the stack. This address also called return address is transferred back into CAR with a subroutine return operation.
- Table 4 gives the function table for micro program sequencer. When $S_1S_0 = 00$, an external address is transferred to CAR. The transfer from address field of microinstruction occurs when $S_1S_0 = 01$ and $T = 1$. When $S_1S_0 = 10$, stack register contents are transferred to CAR and when $S_1S_0 = 11$, incremented contents of CAR are

transferred to the CAR.

Table 4 Function table of micro program sequencer

I2	I1	I0	T	S1	S0	Operation	Description
X	0	0	X	0	0	CAR<-EXA	Transfer external address
1	0	1	1	0	1	CAR<-BRA, SR<-CAR+1	Branch to subroutine and save the next instruction address in stack (Push Operation).
0	0	1	1	0	1	CAR<-BRA	Transfer branch address
X	1	0	X	1	0	CAR<-SR	Transfer from stack register
0	1	1	0	1	1	CAR<-CAR+1	Increment address

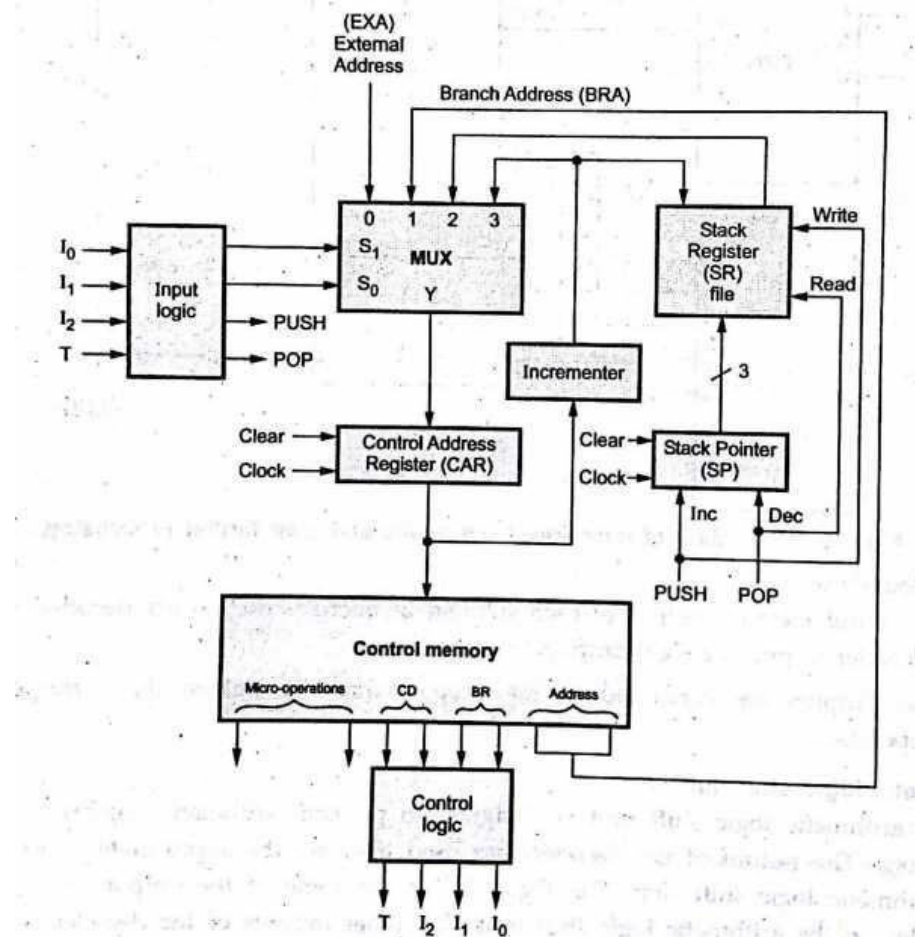


Figure 5: Typical micro program sequencer organization

Control Memory:

Micro-programming is a method of control unit design in which the control signal selection and sequencing information is stored in a ROM or RAM called a control memory (CM). The control signals to be activated at any time are specified by a microinstruction, which is fetched from CM in much similar way an instruction is fetched from main memory. Each microinstruction also explicitly or implicitly Specifies, the next microinstruction to be used, thereby providing the necessary information for sequencing. A sequence of one or more micro-operations designed to control specific operation, such as addition, multiplication is called a

micro program. The micro-programs for all instructions are stored in the control memory.

Sequencing and Execution of Micro Instruction:

A micro-program control unit can be viewed as consisting of two parts:

- The control memory that stores the microinstructions.
- Sequencing circuit that controls the generation of the next address.

A micro-program sequencer attached to a control memory inputs certain bits of the microinstruction, from which it determines the next address for control memory. A typical sequencer provides the following address-sequencing capabilities:

- Increment the present address for control memory.
- Branches to an address as specified by the address field of the micro instruction.
- Branches to a given address if a specified status bit is equal to 1.
- Transfer control to a new address as specified by an external source (Instruction Register).
- Has a facility for subroutine calls and returns.

Depending on the current microinstruction condition flags, and the contents of the instruction register, a control memory address must be generated for the next micro instruction.