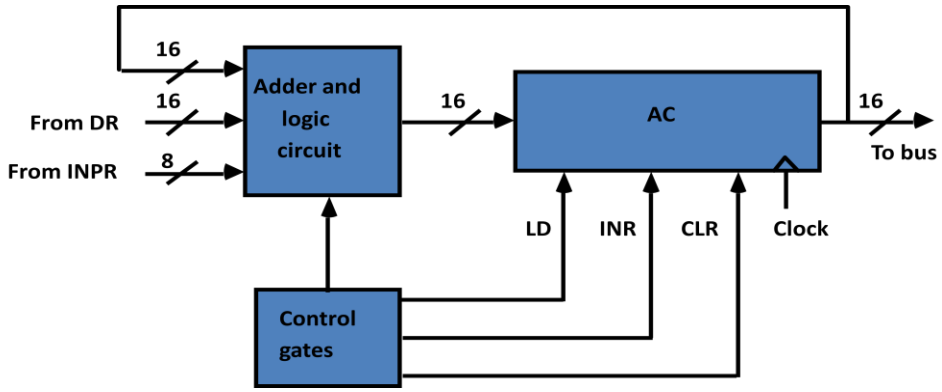
Chapter 4

Some Of Important Terms :

- **Control Memory**: Control Memory is the storage in the microprogrammed control unit to store the microprogram.
- **Control Word**: The control variables at any given time can be represented by a control word string of 1 's and 0's called a control word
- **Microoperations:** In computer central processing units, micro-operations (also known as a micro-ops or μops) are detailed low-level instructions used in some designs to implement complex machine instructions (sometimes termed macro-instructions in this context).
- **Micro instruction**: · A symbolic microprogram can be translated into its binary equivalent by means of an assembler. · Each line of the assembly language microprogram defines a symbolic microinstruction. · Each symbolic microinstruction is divided into five fields: label, microoperations, CD, BR, and AD.
- **Micro program:** A sequence of microinstructions constitutes a microprogram. · Since alterations of the microprogram are not needed once the control unit is in operation, the control memory can be a read-only memory (ROM).

BackGround: Every Instruction that is executed by the microprocessor have their own microoperations involved. The microoperation is determined by the microinstruction subroutines. There is subroutine of each instruction. The collection of microinstruction subroutine is called microprogram. MicroProgram is stored in the control memory. Every instruction is defined by the 3 bits and corresponding instruction's microoperation subroutine is stored in the contol memory of 7 bit . So using 3 bit of opcode requires to generate 7 bit address. This is called as mapping. The microinstruction stored in control memory is executed usually sequentially. But the sequence may be break or can be called as branching with the condition of status bit. The next address of the microinstruction to be executed in case of branching is determined by address sequencing circuit or address Sequencer.

1. **Explain the design procedure of Accumulator logic.**
   - The circuit associated with the Accumulator register is shown in the figure. The adder and logic circuit has three set of input. One set of 16 inputs comes from output of AC. Another set of 16 inputs come from the data register DR. A third set of eight inputs comes from the input register INPR. The output of the adder and logic circuit provides the data inputs for the Accumulator register (AC). In addition it is necessary to include logic gates for controlling the LD, INR, and CLR in the register and for controlling the operation of the adder and logic circuit. The control function of the different microoperation is shown below and accordingly circuits are designed. The 9 different microoperation are generated in the adder and logic circuit and loaded in AC at proper time.



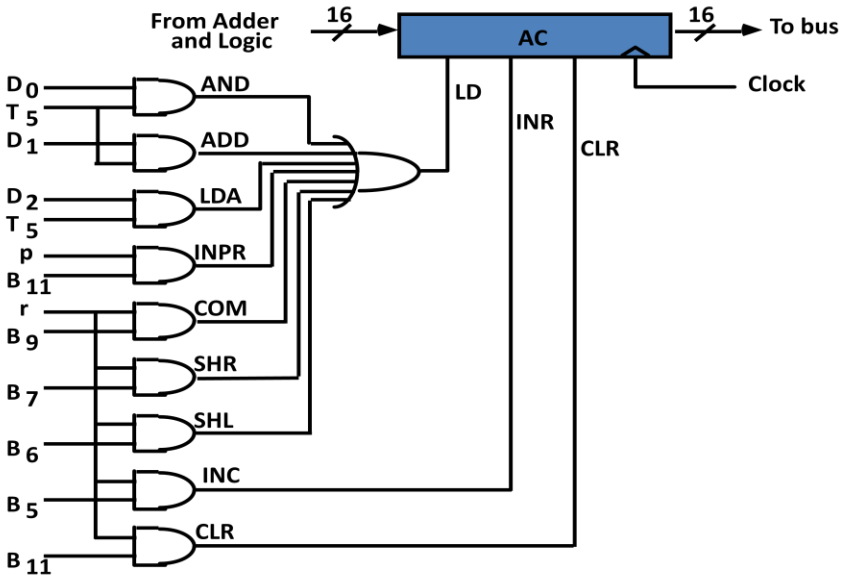
All the statements that change the content of AC

D_0T_5:	$AC \leftarrow AC \wedge DR$	AND with DR
D_1T_5:	$AC \leftarrow AC + DR$	Add with DR
D_2T_5:	$AC \leftarrow DR$	Transfer from DR
pB_{11}:	$AC(0\text{-}7) \leftarrow INPR$	Transfer from INPR
rB_9:	$AC \leftarrow AC'$	Complement
rB_7 :	$AC \leftarrow shr\ AC, AC(15) \leftarrow E$	Shift right
rB_6 :	$AC \leftarrow shl\ AC, AC(0) \leftarrow E$	Shift left
rB_{11} :	$AC \leftarrow 0$	Clear
rB_5 :	$AC \leftarrow AC + 1$	Increment

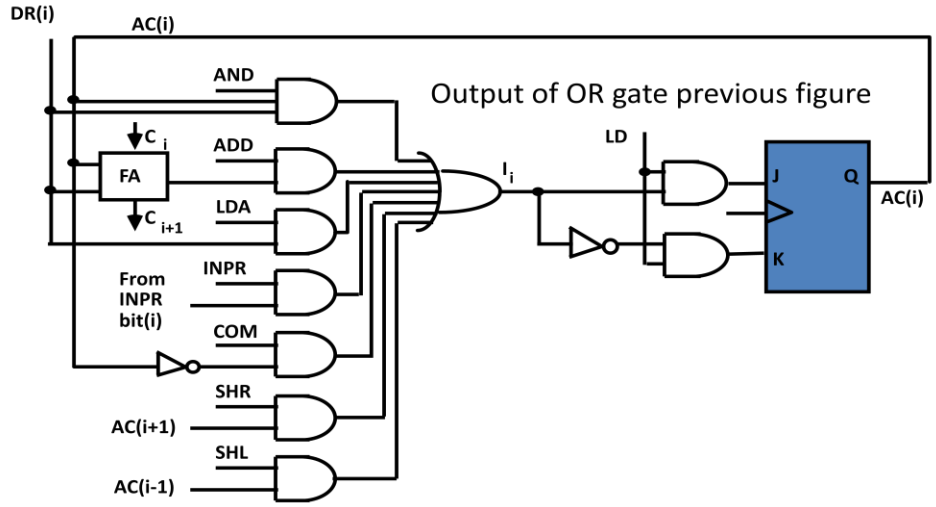
- D0 D1 … are same signals used in common bus design
- $D_7I'T_3=r$
- IR(i)=Bi(IR(0-11)) eg. B11=IR(11)

2. Explain the gate structure for controlling the LD INR and CLR of Accumulator.

- The gate structure that control the LD, INR and CLR inputs of AC is shown in figure .The gate configuration is derived from the control functions in the above list. The control function for the clear micro operation is rB_{11} where $r=D_7I'T_3$ and $B_{11}=IR(11)$.The output of the AND gate that generates the control function is connected to the CLR input of the register. Similarly the output of the gate that implements the increment micro operation is connected to the INR input of the register. The other 7 micro operations are generated in the adder and logic circuit and are loaded into AC at the proper time. The outputs of the gates for each control function are marked with the symbolic name .The outputs are used in the design of adder and logic circuit.



3. **Explain the component of ALU with their functions.**
   - The adder and logic circuit can be subdivided into 16 stages with each stage corresponding to one bit of AC.The internal construction of the register is shown in figure.Looking back at the figures we note that each stage has a JK flip-flop , two OR gates , and two AND gates. The LD(load) input is connected to the inputs of the AND gates .Figure shows one such AC register stage with OR gate removed. The input is labeled and the output AC(i).When the LD input is enabled the 16 inputs I_i for i=0,1,2,..15 are transferred to AC(0-15). One stage of the adder and subtractor consists of seven AND Gates one Or gates and a Full Adder (FA).The input of the gates with symbolic name comes from the output of the gates marked with the same symbolic name from previous figure .For Example , the input marked with ADD in Figure is connected to the output marked ADD.
   - The AND operation is achieved by ANDing AC(i) with the corresponding bit in the data register DR(i) . The ADD operation is obtained using a binary adder.One stage of the adder uses a full-adder with the corresponding input and output carries .The transfer from INPR to AC is only from bits 0 through 7 . The complement microoperation is obtained by inverting the bit value in AC . The shift –right operation transfers the bit from AC(i+1) and the shift-left operation transfer the bit from AC(i-1).The complete adder and logic circuit consists of 16 stages connected together .



4. Differentiate between Hardwired and Microprogrammed Control unit.

Hardwired Control unit	Micro-programmed Control unit
Hardwired control unit generates the control signals needed for the processor using logic circuits	Micrprogrammed control unit generates the control signals with the help of micro instructions stored in control memory
Hardwired control unit is faster when compared to microprogrammed control unit as the required control signals are generated with the help of hardwares.	This is slower than the other as micro instructions are used for generating signals here.
Difficult to modify as the control signals that need to be generated are hard wired	Easy to modify as the modification need to be done only at the instruction level
More costlier as everything has to be realized in terms of logic gates	Less costlier than hardwired control as only micro instructions are used for generating control signals
It cannot handle complex instructions as the circuit design for it becomes complex	It can handle complex instructions
Only limited number of instructions are used due to the hardware implementation	Control signals for many instructions can be generated
Used in computer that makes use of Reduced Instruction Set Computers(RISC)	Used in computer that makes use of Complex Instruction Set Computers(CISC)

5. **Define control unit. Explain the organization of Microprogrammed Control Unit.**

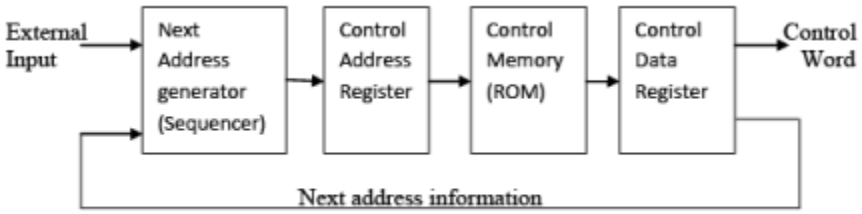


Fig 3-1: Microprogrammed Control Organization

- The general configuration of a micro-programmed control unit is demonstrated in the block diagram above. **The control memory** is usually a ROM, which stores the microinstruction with control function. The **control address register** specifies the address of the microinstruction, and the **control data register** holds the microinstruction read from memory currently. The microinstruction contains a control word that specifies one or more microoperations for the data processor. Once current operations are executed, the control must determine the next address. The location of the next microinstruction may be the one next in sequence, or branch.
- While the microoperations are being executed, the next address is computed in the **next address generator** circuit and then transferred into the control address register to read the next microinstruction. Thus a microinstruction contains bits for initiating microoperations in the data processor part and bits that determine the address sequence for the control memory. The **next address generator** is sometimes called a micro-program sequencer, as it determines the address sequence to be read from control memory. 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 operations.

.

6. **Explain the address Sequencing Process.**

- Address Sequence is the process of determining the next address of the microinstruction tobe executed after executing current microinstruction. Microinstructions are stored in control memory in groups, with each group specifying a routine. Each computer instruction has its own microprogram routine in control memory to generate the microoperations that execute the instruction To appreciate the address sequencing in a microprogram control unit. The steps during the execution of a single computer instruction are as follows:

Step 1:

An initial address is loaded into the control address register when power is turned on in the computer that activates the instruction fetch routine.

Step 2:

The control memory next must go through the routine that determines the effective address of the operand. When the effective address computation routine is completed, the address of the operand is available in the memory for address register.

Step 3:

The next step is to generate the microoperations that execute the instruction fetched from memory. Each instruction has its own microprogram routine stored in memory location that is found with the mapping process.

Step 4:

Once the required routine is reached the micoinstruction that execute the instruction may be sequenced by the incrementing the control address register.

7. **Explain the conditional branching mechanism.**

- As we know the microinstruction are executed sequentially from the control memory .But it is not the case always sometime the branching of instruction occurs .The branch logic of Figure provides decision-making capabilities in the control unit. The status conditions are special bits in the system that provides parameter information such as the carry-out, the sign bit, the mode bits, and input or output status condition. Information in these bits can be tested as whether value is 1 or 0.The status bits, together with the field in the microinstruction that specifies a branch address, control the conditional branch decisions generated in the branch logic. The simple way is to test the specified condition and branch to the indicated address if the condition is met otherwise the address register is incremented. The branch logic hardware may be implemented by multiplexer. An unconditional branch microinstruction can be implemented by loading the branch address from control memory into the control address register. This can be accomplished by fixing the value of one status bit at the time input of the multiplexer so ,it is always equal to one .If Condition is true, then Branch (address from the next address field of the current microinstruction) else Fall Through

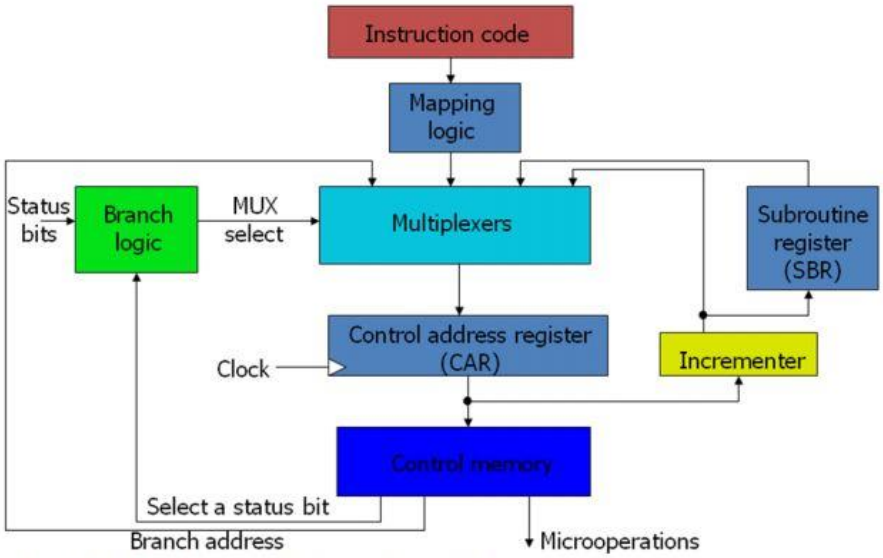


Fig 3-2: Selection of address for control memory

8. **What do you mean by mapping of instructions? Explain the procedure for mapping from instruction code to microinstruction address.**

- Mapping of instruction is the process of finding the microinstruction address from the instruction code. As we know every microprocessor instruction opcode has their corresponding microprogram. Mapping helps to get the starting address of microprogram as a input of opcode bits.
- A special type of branch exists when a microinstruction specifies a branch to the first word in control memory where a microprogram routine for an instruction is located.The status bits for this type of branch are the bits in the operation code part of the instruction. Let's take a example instruction with 4 bit opcode whose microprogram routine in the control memory needs 7 bits to address. We know for each opcode there exist a microprogram

routine in control memory that executes the instruction. One simple mapping process that converts the 4-bit operation code to a 7-bit address for control memory is shown in Figure.

→Placing a 0 in the most significant bit of the address

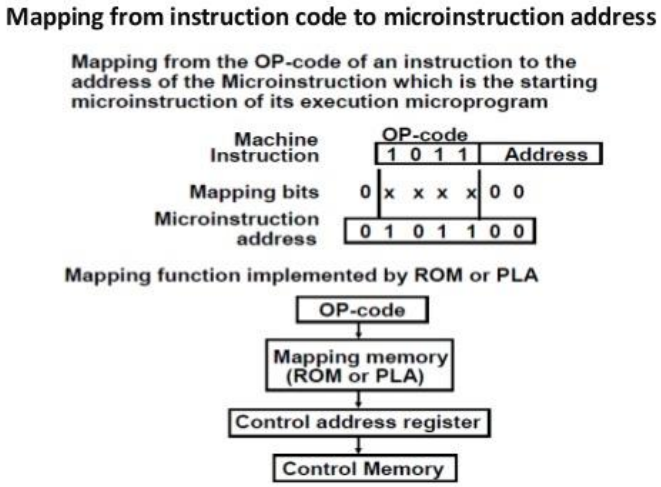
→Transferring the four operation code bits

→Clearing the two least significant bits of the control address register

- This provides for each computer instruction a microprogram routine with a capacity of four microinstructions.

→ If the routine needs more than four microinstructions, it can use addresses 1000000 through 1111111.

→If it uses fewer than four microinstructions, the unused memory locations would be available for other routines.



9. What do you mean by microprogram. Differentiate between Symbolic and Binary microprogram with example?

➢ A sequence of microinstructions constitutes a microprogram. Microprogram contains different subroutines of microinstruction. Microoperation performs one or more microoperation. Since alterations of the microprogram are not needed once the control unit is in operation, the control memory can be a read-only memory (ROM).

Symbolic Microprogram	Binary MicroProgram
The symbolic microprogram is a convenient from for writing microprogram in a way that people can read and understand	The binary microprogram is a quiet difficult from for writing microprogram in as symbolic equivalent binary must be memorized.
This is not the way that the microprogram is stored in memory .	This is the way that the microprogram is stored in memory .
The symbolic microprogram must be translated to binary either by means of assembler program or by the user	No any translation is required for binary microprogram.

The execution process is slower due to conversion.	The execution process is faster as no conversion.

Give example of ADD from below:

Label	Microoperations	CD	BR	AD
	ORG 0			
ADD:	NOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
	ADD	U	JMP	FETCH
	ORG 4			
BRANCH:	NOP	S	JMP	OVER
	NOP	U	JMP	FETCH
OVER:	NOP	I	CALL	INDRCT
	ARTPC	U	JMP	FETCH

Micro Routine	Address		Binary microinstruction					
	Decimal	Binary	F1	F2	F3	CD	BR	AD
ADD	0	0000000	000	000	000	01	01	1000011
	1	0000001	000	100	000	00	00	0000010
	2	0000010	001	000	000	00	00	1000000
	3	0000011	000	000	000	00	00	1000000
BRANCH	4	0000100	000	000	000	10	00	0000110
	5	0000101	000	000	000	00	00	1000000
	6	0000110	000	000	000	01	01	1000011
	7	0000111	000	000	110	00	00	1000000
STORE	8	0001000	000	000	000	01	01	1000011
	9	0001001	000	101	000	00	00	0001010
	10	0001010	111	000	000	00	00	1000000
	11	0001011	000	000	000	00	00	1000000

10. Explain the basic requirement for designing control unit.

The basic components of the microprogrammed control unit are the control memory and the circuits that select the next address . to be completed…

11. Explain the structure of Control unit.(MicroProgrammed)

- The bit of the microinstruction are divided into fields (F1,F2,F3) with each field defining a distinct separate function .The various fields encountered in the instruction format provides control bit to initiate microoperation in the system , special bits to specify the way that the next address is to be evaluated , and an address field for branching.
- Since, there are three microoperation field we need 3 decoder. Only some of the output of decoder are shown to be connected to their output .
- Each of the output of decoder must be connected to the proper circuit to initiate the corresponding microoperation .
- For Example when F1=001 the next clock pulse transition the content of AC is Added with DR and stored in AC.
- For Example when F1=101 the next clock pulse transition transfer content of DR(0-10) to AR(symbolized by DRTAC)
- Similarly other operation is performed.

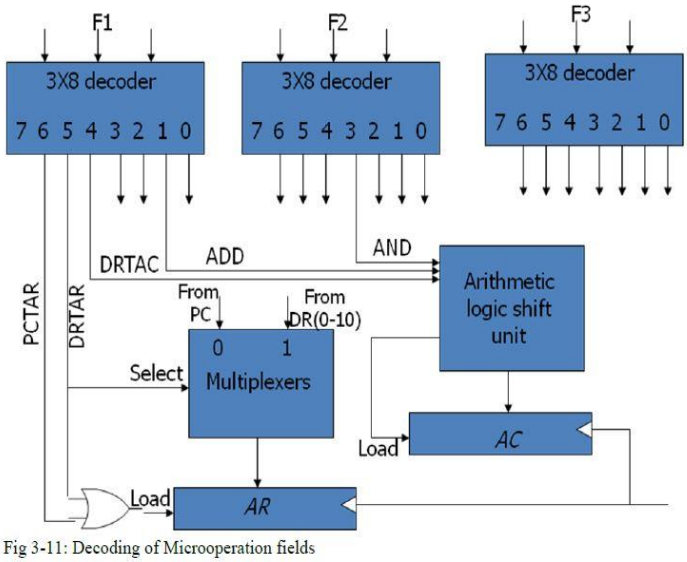


Fig 3-11: Decoding of Microoperation fields

12. Write the role of microprogram sequencer in microprogrammed control unit.

- The basic components of the microprogrammed control unit are the control memory and the circuits that select the next address . The address selection part is called a microprogram sequencer .The purpose of a microprogram sequencer is to present an address to the control memory so that a microinstruction may be read and executed .*Typical sequencer operation are increment, branch or jump, call and return from subroutine , load an external address , push or pop of stack and other address sequencing operations.*A microprogram sequencer can be constructed with digital function to suite a particular application .To guarantee a wide range of acceptability, an *integrated circuit sequencer must provide an internal organization that can be adapted to a wide range of application.* Commercial sequencer include within unit an internal register stack used for temporary storage of addresses during microprogram looping and subroutine calls.Some sequencer provide an output register which can function as the address register for the control memory.

13. Explain the basic Comfiguration of MicroProgram based computer.

- The block diagram of computer is shown in the figure.
  It consists of two memory :
  → A main memory for storing instructions and data
  → A control memory for storing the microprogram
- Four registers Program Counter (PC), Address Register (AR), Data Register (DR) and Accumulator (AC) are associated with the processor unit and two register Control Address Register (CAR) and SBR (Subroutine Register) with the control unit. PC stores the address of next instruction to be executed, AR contains …….., AC is Accumulator and DR …….. The control memory and its register are organized as a microprogrammed control unit, as shown. The transfer of information among the registers in the processor is done through multiplexers rather than a common bus. DR can receive information from AC , PC or memory.AR can receive information from PC or DR.PC can get information only from AR.The arithmetic , logic and shift unit performs microoperation with data from AC and DR and places the result in AC. Memory receives its address from AR.

Prepared By: Er. Santosh Dahal