Computer

Unit 3
CONTROL UNIT

INSTRUCTION TYPES

Data Movement Instructions

Data movement instructions are used to move data among the different units of the machine.

Data movement operation	Meaning
MOVE	Move data (a word or a block) from a given source (a register or a memory) to a given destination
LOAD	Load data from memory to a register
STORE	Store data into memory from a register
PUSH	Store data from a register to stack
POP	Retrieve data from stack into a register

Arithmetic and Logical Instructions

Arithmetic operations	Meaning
ADD	Perform the arithmetic sum of two operands
SUBTRACT	Perform the arithmetic difference of two operands
MULTIPLY	Perform the product of two operands
DIVIDE	Perform the division of two operands
INCREMENT	Add one to the contents of a register
DECREMENT	Subtract one from the contents of a register

Sequencing Instructions

Logical operation	Meaning
AND	Perform the logical ANDing of two operands
OR	Perform the logical ORing of two operands
EXOR	Perform the XORing of two operands
NOT	Perform the complement of an operand
COMPARE	Perform logical comparison of two operands and set flag accordingly
SHIFT	Perform logical shift (right or left) of the content of a register
ROTATE	Perform logical shift (right or left) with wraparound of the content of a register

Flag name	Meaning
Negative (N)	Set to 1 if the result of the most recent operation is negative, it is 0 otherwise
Zero (Z)	Set to 1 if the result of the most recent operation is 0, it is 0 otherwise
Overflow (V)	Set to 1 if the result of the most recent operation causes an overflow, it is 0 otherwise
Carry (C)	Set to 1 if the most recent operation results in a carry, it is 0 otherwise

Input/output Instructions

Transfer of control operation	Meaning
BRANCH-IF-CONDITION	Transfer of control to a new address if condition is true
JUMP	Unconditional transfer of control
CALL	Transfer of control to a subroutine
RETURN	Transfer of control to the caller routine

Example 1 In this example, we would like to show a program segment that can be used to perform the task of adding 100 numbers stored at consecutive memory locations starting at location 1000. The results should be stored in memory location 2000.

	CLEAR R_0;	$R_0 \leftarrow 0$
	MOVE # *100*, R_1;	$R_1 \leftarrow 100$
	CLEAR R_2;	$R_2 \leftarrow 0$
LOOP:	*ADD 1000*(R_2), R_0;	$R_0 \leftarrow R_0 + M\ (1000 + R_2)$
	INCREMENT R_2;	$R_2 \leftarrow R_2 + 1$
	DECREMENT R_1;	$R_1 \leftarrow R_1 - 1$
	BRANCH-IF > 0 *LOOP*;	*GO TO LOOP if contents of* $R_1 > 0$
	STORE R_0, *2000*;	$M(2000) \leftarrow R_0$

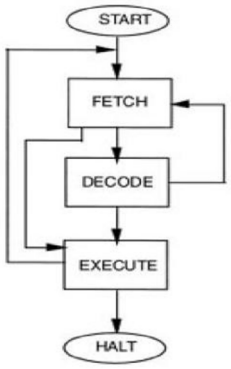
In this example, use has been made of immediate (*MOVE #100*, R_1) and indexed (*ADD* 1000(R_2), R_0) addressing.

Computer

Unit 3
CONTROL UNIT

Instruction cycles and sub cycles

Procedure for an instruction execution one by one sequentially inside CPU is called an instruction cycle.



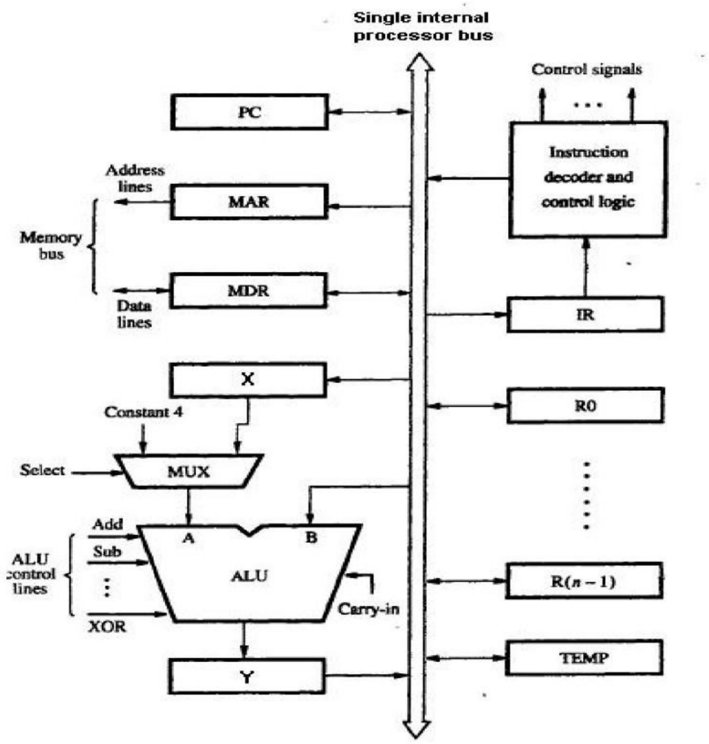
Fetch Cycle:
It includes certain steps like:

1. Fetching Instruction to be executed
2. Decoding of instruction
3. Reading the operands to be

operated Execute Cycle:
It includes steps like:

1. Executing or processing the operands in ALU
2. Storing the result at the destination

Execution of a complete instruction



Computer

Unit 3
CONTROL UNIT

1) Send the content of PC to MAR for memory reading and wait till Memory function is completed. MAR← [PC]
2) Increment PC
   PC←[PC]+1
3) After memory read the instruction resides in MBR. IR← [MBR]
4) [IR] moves to decoder for decoding purpose.
5) Then the operation takes place according to the operand field of IR. In above steps,

Step 1 to 3 is termed as instruction fetch phase. Step 4 is known as decode phase

Step 5 is known as execute phase

CONTROL UNIT

- The function of control unit is to generate relevant timing and control signals to all operations in the computer.
- It controls the flow of data between the processor and memory and peripherals

DESIGN OF CONTROL UNIT

Control unit generates control signals using one of the two organizations:

- Hardwired Control Unit
- Micro-programmed Control Unit

HARDWIRED CONTROL UNIT

- It is implemented as logic circuits (gates, flip-flops, decoders etc.) in the hardware.
- This organization is very complicated if we have a large control unit.
- In this organization, if the design has to be modified or changed, requires changes in the wiring among the various components. Thus the modification of all the combinational circuits may be very difficult.

ADVANTAGES

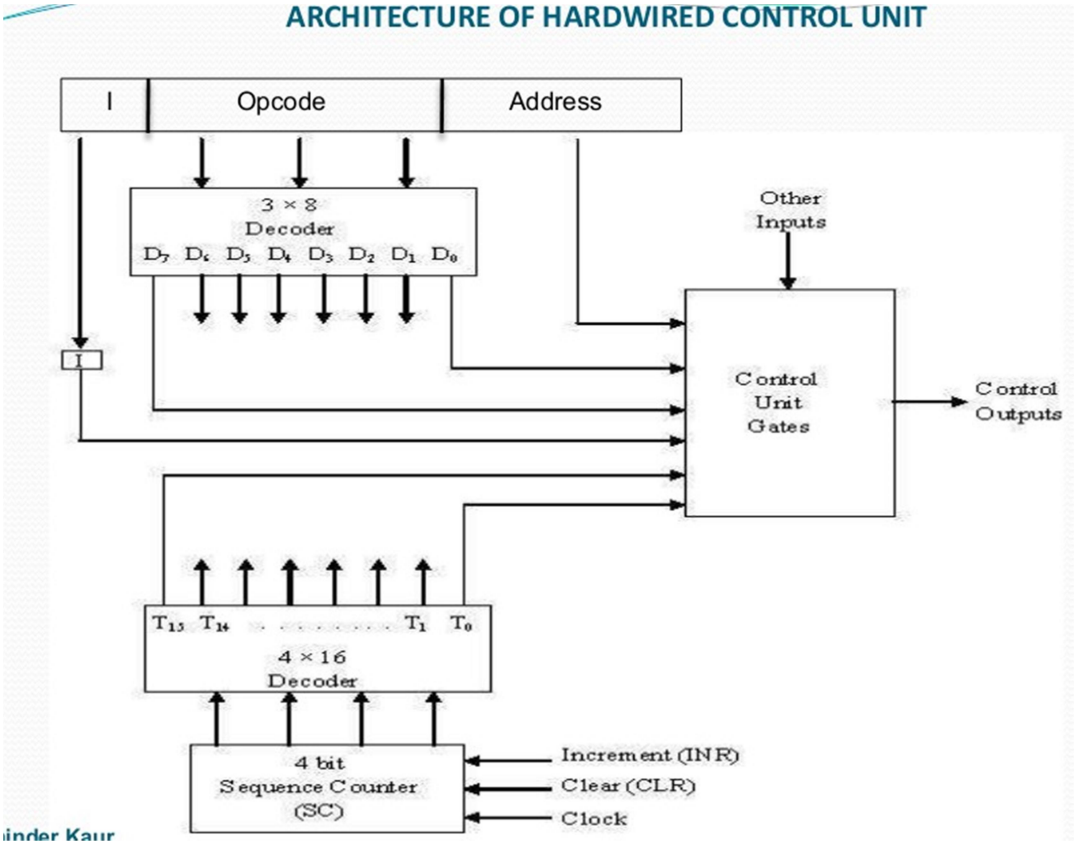
- Hardwired Control Unit is fast because control signals are generated by combinational circuits.
- The delay in generation of control signals depends upon the number of gates.

DISADVANTAGES

- More is the control signals required by CPU; more complex will be the design of control unit.

Unit 3
CONTROL UNIT

- Modifications in control signal are very difficult. That means it requires rearranging of wires in the hardware circuit.
- It is difficult to correct mistake in original design or adding new feature in existing design of control unit.



MICRO-PROGRAMMED CONTROL UNIT

- A micro-programmed control unit is implemented using programming approach. a sequence of micro-operations are carried out by executing a program consisting of micro-instructions.
- Micro-program, consisting of micro-instructions is stored in the control memory of the control unit.
- Execution of a micro-instruction is responsible for generation of a set of control signals.
- A micro-instruction consists of: One or more micro-operations to be executed and address of next microinstruction to be executed.
- **Micro-Operations**: The operations performed on the data stored inside the registers are called micro-operations.
- **Micro-Programs**: Microprogramming is the concept for generating control signals using programs. These programs are called micro-programs.
- **Micro-Instructions**: The instructions that make micro-program are called micro-instructions.
- **Micro-Code**: Micro-program is a group of micro-instructions. The micro-program can also be termed as micro-code.

Unit 3
CONTROL UNIT

- **Control Memory**: Micro-programs are stored in the read only memory (ROM). That memory is called control memory.

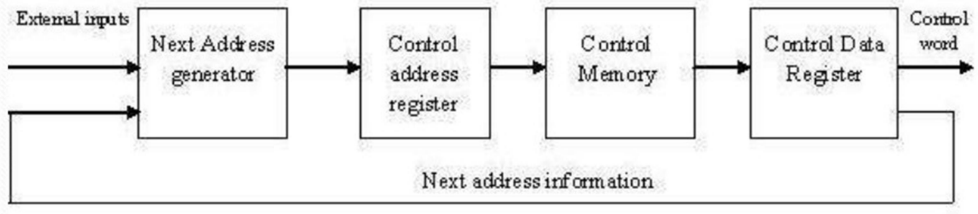
ADVANTAGES

- The design of micro-program control unit is less complex because micro-programs are implemented using software routines.
- The micro-programmed control unit is more flexible because design modifications, correction and enhancement is easily possible.
- The new or modified instruction set of CPU can be easily implemented by simply rewriting or modifying the contents of control memory.
- The fault can be easily diagnosed in the micro-program control unit using diagnostics tools by maintaining the contents of flags, registers and counters.

DISADVANTAGES

- The micro-program control unit is slower than hardwired control unit. That means to execute an instruction in micro-program control unit requires more time.
- The micro-program control unit is expensive than hardwired control unit in case of limited hardware resources.
- The design duration of micro-program control unit is more than hardwired control unit for smaller CPU.

ARCHITECTURE OF MICRO-PROGRAMMED CONTROL UNIT



COMPARISON BETWEEN HARDWIRED AND MICRO-PROGRAMMED CONTROL UNIT

Attributes	Hardwired Control	Micro-programmed Control
Speed	Fast	Slow
Cost of Implementation	More	Cheaper
Flexibility	Not flexible, difficult to modify for new instruction	Flexible, new instructions can easily be added
Ability to Handle Complex Instructions	Difficult	Easier
Decoding	Complex	Easy
Applications	RISC Microprocessor	CISC Microprocessor
Instruction Set Size	Small	Large
Control Memory	Absent	Present
Chip Area Required	Less	More

Computer

Unit 3
CONTROL UNIT

Microinstruction sequencing

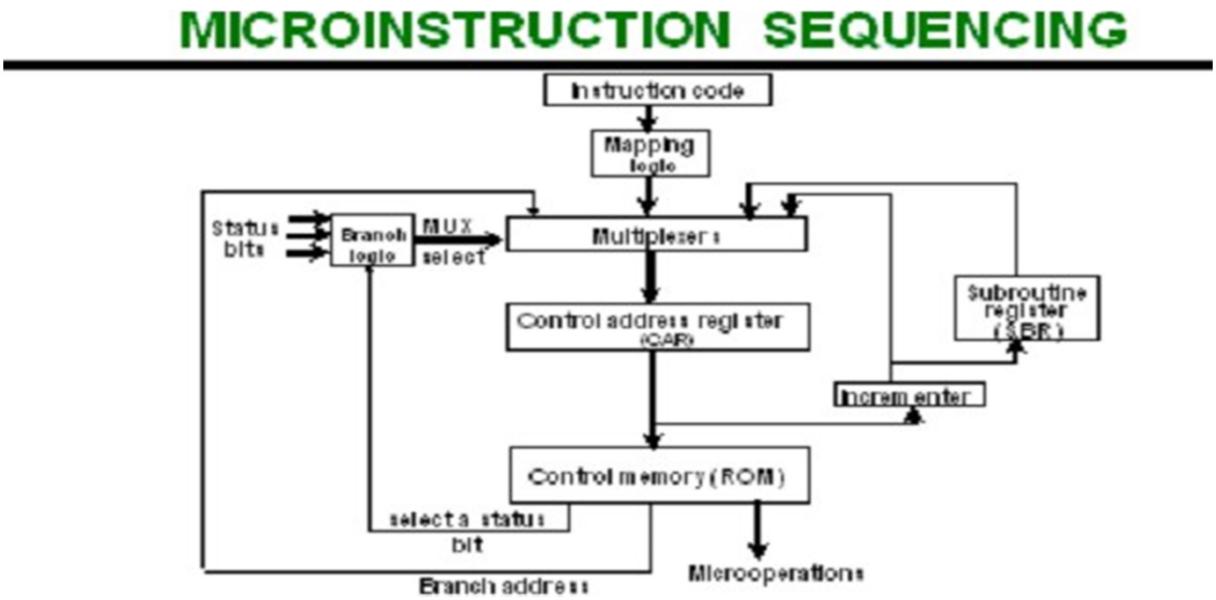
• Microinstructions are usually stored in groups where each group specifies a routine, where each routine specifies how to carry out an instruction.

• Each routine must be able to branch to the next routine in the sequence.

• An initial address is loaded into the CAR when power is turned on; this is usually the address of the first microinstruction in the instruction fetch routine.

• Next, the control unit must determine the effective address of the instruction.
• The next step is to generate the micro operations that executed the instruction.

- This involves taking the instruction's opcode and transforming it into an address for the instruction's micro program in control memory. This process is called mapping.
- While microinstruction sequences are usually determined by incrementing the CAR, this is not always the case. If the processor's control unit can support subroutines in a micro program, it will need an external register for storing return addresses.



Microinstructions with the next-address field

- Several branch microinstructions are required to enable sharing of common code.
- The branch microinstructions do not perform any useful operation related to data.
- They are required to determine the address of the next microinstruction.
- They slow down the execution of the instruction.
- The address field indicates the location of the next microinstruction to be fetched.
- In effect, every microinstruction becomes a branch microinstruction in addition to its other function.

Computer

Unit 3
CONTROL UNIT

Prefetching microinstructions

To achieve faster operation, the next microinstruction can be prefetched while the current one is being executed. Execution time can be overlapped with the fetch time.

- Prefetching microinstructions presents some difficulties:
- Status flags and the results of the current microinstruction that is being executed are necessary to determine the address of the next microinstruction.
- Straightforward Prefetching may occasionally fetch a wrong instruction.
- Fetch must be repeated.

Microinstruction Format

Microinstructions are commonly divided into two types :

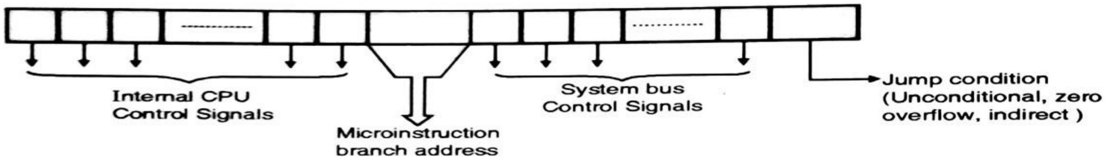
(1) Horizontal microinstructions

(2) Vertical microinstructions

(1) Horizontal microinstructions :

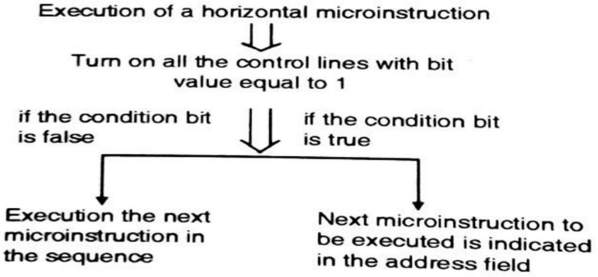
In the horizontal microinstruction each bit of the microinstruction represents a control signal. A Horizontal microinstruction has the following general attributes :

1. long format
2. high degree of parallelism
3. little encoding of control information



The format of a horizontal microinstruction is as follows :

- There is one bit for each internal control line.
- There is one bit for each system bus control line.
- There is condition field for each condition for conditional branching.
- Address field that stores the address of the microinstruction to be executed next when a branch is taken.



(2) Vertical microinstructions :

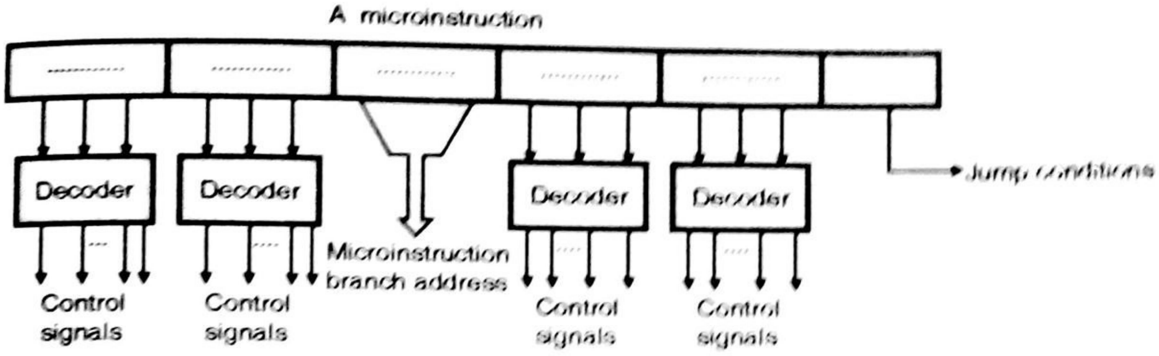
A vertical microinstruction allows encoding of control information. Vertica microinstructions are characterized by :

- Short formats

Computer

Unit 3
CONTROL UNIT

- Limited ability to express parallel micro-operations.
- Considerable encoding of the control information.

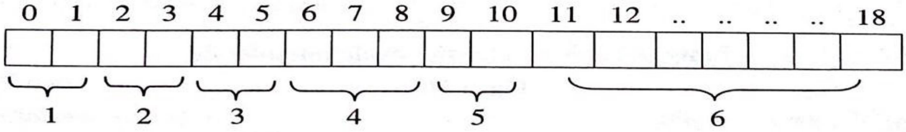


Grouping of control signals

The design of an encoded microinstruction format with distinct groups can be stated as below :

- Organize the format into independent fields. Each field depicts a set of actions (control signals). Control signals from different fields can be activated simultaneously.
- Actions from fields are mutually exclusive. Only one action specified for a given field could occur at a time.

For a hypothetical machine, control signals can be grouped as shown below :



Field definition:

1. Register Transfer
2. Memory operation
3. Sequencing operation
4. ALU operation
5. Register selection
6. Constant

Computer

Unit 3
CONTROL UNIT

Numericals

A computer has 32-bit instructions and 12-bit addresses. If there are 250 two-address instructions, how many one-address instructions can be formulated ?

Ans. : For two address instructions, the instruction format from the given information is as shown in the Fig. 5.56.

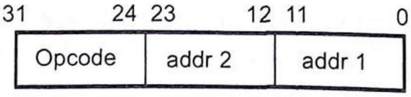


Fig. 5.56

Opcode is 8-bit and remaining 24-bits represent two address fields. If the address is 12-bit, we can have opcode field of 20-bit. Thus we can formulate 2^{20} = 1 M one address instructions with 12-bit address and 32-bit instruction length.

A computer has 16 registers, an ALU with 32 operations and a shifter with eigh operations, all connected to a common bus system.

i) Formulate a Control Word for a micro-operation.

ii) Specify the number of bits in each field of the control word and give a gene encoding scheme.

iii) Show the bits of the control word that specify the micro-operation ***R4 ← R5 + R6.***

ns. : The Control Word is formed with following fields :

A : Register input 1 : 4-bits

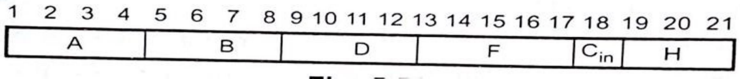
B : Register input 2 : 4-bits

D : Register Destination : 4-bits

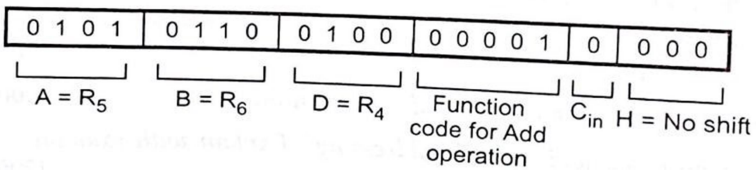
F : ALU Functions : 5-bits

H : Shifter Operations : 3-bits

Control Word



Let us assume function code for addition operation is 00001 and H field with 00 ɔut indicates no shift. Then the Control Word that specify the micro-operatio ← $R_5 + R_6$ is



Unit 3
CONTROL UNIT

Show how a 9-bit micro-operation field in microinstruction can be divided i subfields to specify 46 micro-operation. How many micro-operation can be specified in one micro-instruction ? UPTU 2003-20

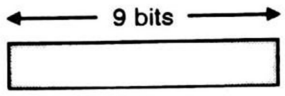
Solution :

A 9-bit microoperation field can generate 2^9 possible control field patterns. This is case when we go for pure vertical format.

In practice, a microoperation field is divided into subfields. This helps in combi functionally unrelated signals for maximum parallism.

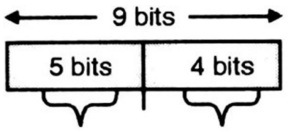
Following divisions are possible to specify 46 microoperation :

(1) Pure vertical format (no division)



46 micro-operation can be used out of $2^9 - 1$.

(2) First subfield of 5 bits and the second subfield of 4 bits.



1	$2^4 - 1$
$2^5 - 1$	= 15
= 31	

The first subfield will generate $2^5 - 1 = 31$ microoperation, whereas the sec subfield will generate $2^4 - 1 = 15$ mcirooperation. Both the subfields together will gene 46 microoperation.

Number of micro – operation in a macro instruction = $2^n - 1$, where n is the lengt the micro-instruction.

Formulate a mapping procedure that provides eight consecutive microinstruction for each routine. The operation code has six bits and the control memory has 2048 words. UPTU 2005-20

Unit 3
CONTROL UNIT

Solution :

In the design given below, two address fields have been provided in e microinstruction.

Number of bits required to address a word = $\log_2 2048 = \log_2 2^{11} = 11$ bits.

∴ Length of the microinstruction

= 6 (Operation code) + 2 × 11 (two address fields of 11 bits each)

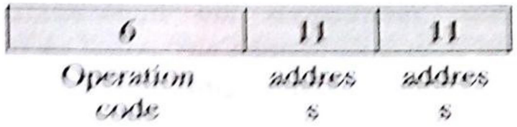
= 28 bits = 4 words.

Number of micro-instructions, that can be stored in the control memory = $\frac{2048}{4} = 512$

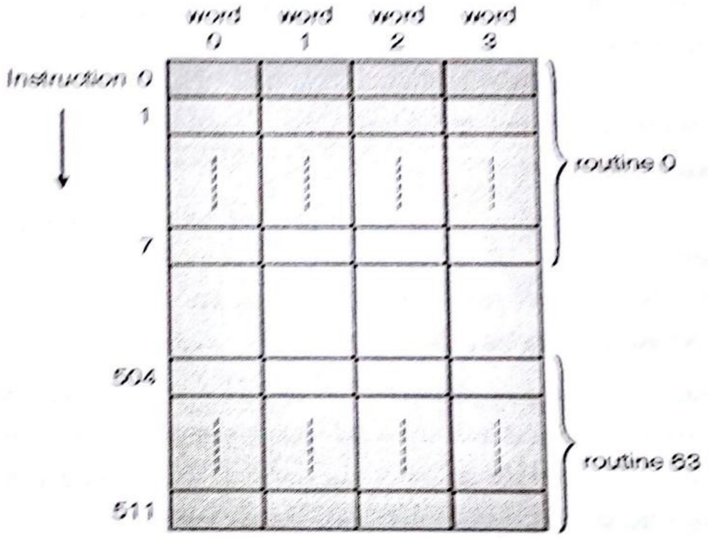
Number of routines, that can be stored in the control memory

(8 instructions per routine) = $\frac{512}{8} = 64 = 2^6$

Microinstruction format :



Control memory organisation :



Instruction size = 4 words = 28 bits

Routine size = 8 instructions = 8 × 4 = 32 words

Number of routines = 64

****End Of Unit 2****