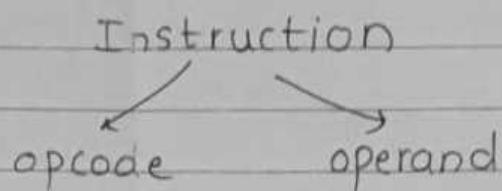


Module 2

Processor Organization and Architecture

Instruction:-

An instruction (word of processor) defines a task to be performed by the processor and it is stored in memory.



Types of instructions:-

① Data Transfer Instruction-

These instructions transfer data from memory locations to registers or vice-versa.

~~There is no~~ arithmetic or logical operations are performed on data. Hence flags (^{No}status register) are not affected.

eg. MOVE B

② Data processing Instruction-

These instructions perform all arithmetic and logical operations on the data hence flags are affected.

eg. ADD B

③ Control instructions-

They control the flow of the program and hence are called branch control instruction.

Their main job is to change the value of PC thereby causing a branch.

eg. JMP 2000H

Instruction format:-

Every instruction has opcode. Additionally it may have one or more operands.

Op indicates the operation to be performed.

Every instruction has unique opcode.

Operands indicates on which data operations are to be performed.

opcode	operand 1	operand 2	...n	Address of Result	Address of next instruction
--------	-----------	-----------	------	-------------------	-----------------------------

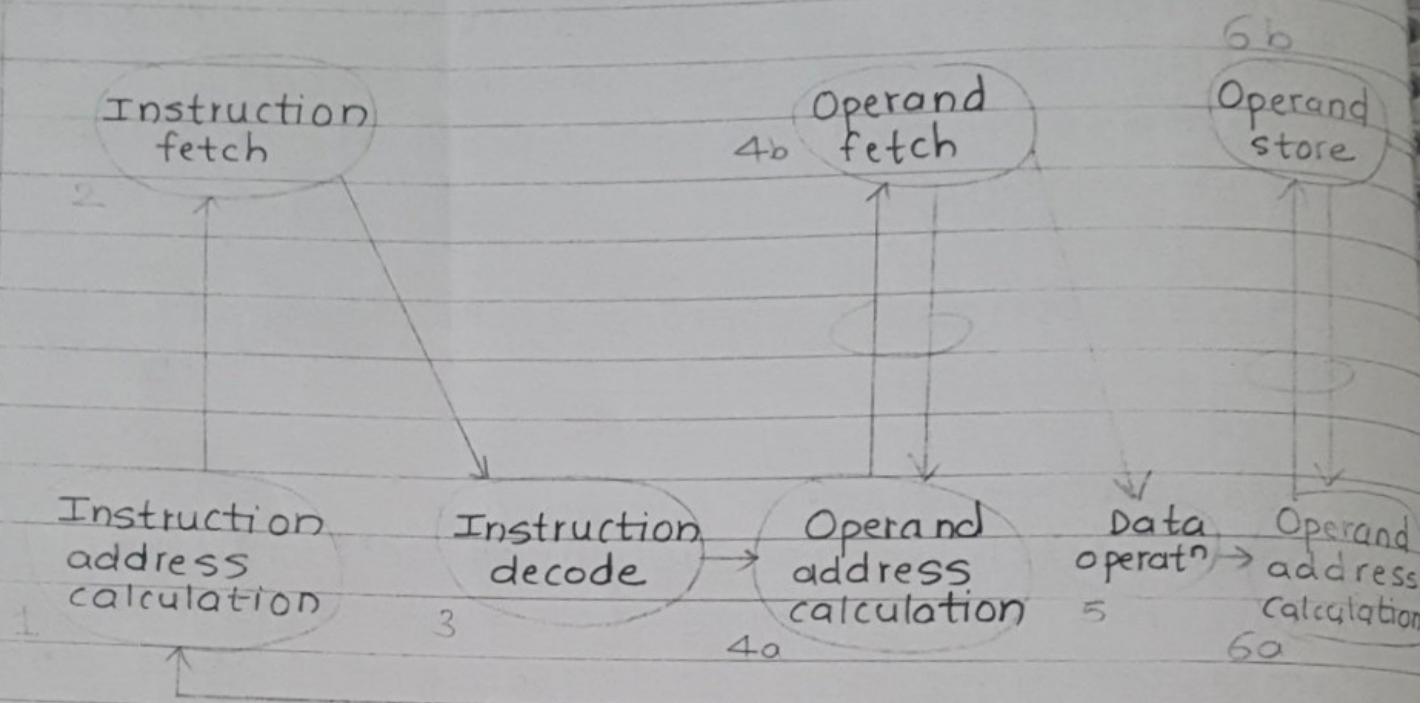
Drawback-

~~After~~ To fetching such instructions we require more no. of machine cycles, which increases the processor time and decreases the speed and performance of the processor.

Solution:-

- Address of next instructⁿ is replaced by program counter which is a separate register. After fetching of each instruction it is incremented.
- Address of result is replaced by accumulator. It is used to store the current arithmetic or logical operation result.

Instruction cycle (state diagram)

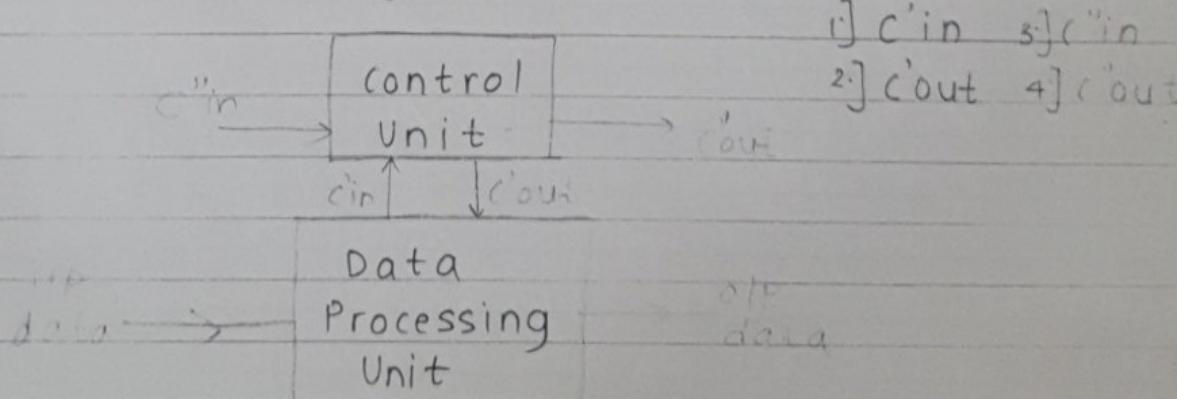


Instruction Sequencing :-

- Instruction sequencing is the method by which instructions are selected for execution.
i.e. manner in which the control of the processor is transferred from one instruction to another.
- The simplest method of controlling the sequence in which instructions are executed is to have each instruction explicitly specify the address of the next instruction.
- But inclusion of instruction address in all instructions has disadvantage that it increases the instruction length which inturn increases the cost of the memory where the instructions are stored.
- PC contains address of instruction ~~of~~ i.
The address of $i+1$ instruction can be determined by incrementing PC.

Instruction Interpretation -

- A control unit interprets an instruction in order to determine the control signals to be issued
- The control signals are transmitted from the CU to the outside world via control lines.
- The given figure shows main control lines connected to the typical control unit.



Four types of control -

- 1) C^{out} - These signals directly control the operation of the data processing unit.
The main function of CU is to generate C^{out} .
- 2) C^{in} - These signals enable data being processed to influence the control unit, allowing the data dependent decisions to be made.
A frequent of C^{in} indicates the occurrence of unusual conditions such as error.
(Underflow, overflow error).
- 3) C^{out} - These signals are transmitted to another control unit and may indicate status conditions such as 'busy' or operation completed.
- 4) C^{in} - These signals are received from other control units. They typically indicate start signal.

C^{in} and C^{out} primarily used to synchronize with other control units.

Module 3 Control Unit Design

Page No.:
Date: / / youva

ISA (Instruction Set Architecture) -
(Vocabulary of processor)

- ISA is the structure of the computer that a machine language programmer (or compiler) must understand to write the correct program for that machine.

- Instruction can be of fixed or variable length.

- The more restriction we put on the size and format of the instruction, lesser is the flexibility to the programmer. But processor design becomes simpler.

This gives rise to two types of processor design

- ① RISC - Reduced Instruction Set Computer
- ② CISC - Complex Instruction Set Computer.

RISC**CISC**

- Reduced Instruction set computer
 - Size of the instruction is fixed
 - Format of the instruction. Format is variable.
 - Limited no. of instruction - Large no. of instruction (Generally below 100)
 - Instructions are more register oriented.
 - Simple addressing modes
 - More user registers are required.
 - It uses hardwired control unit.
 - One machine cycle per instruction.
 - Supports more higher degree of pipelining
 - Complexity in compiler designing.
- complex instruction set computer.
 - size of instruction is variable.
 - More memory based instructions.
 - Very advance level addressing mode.
 - Less no. of registers are required.
 - Microprogram control unit is used.
 - Complex instruction will require more machine cycles to complete the task.
 - Supports low degree of pipelining.
 - Complexity in micro-program designing

Microprogram:-

Set of micro instructions

^{Operation} Micro Instruction:-

- Every instruction is further divided into a set of very basic micro operations.
- A micro operation takes one clk pulse i.e. one 'T' state.

Micro Program for fetch cycle.

- T₁ : MAR \leftarrow PC (load the address of inst. to be fetched)
- T₂ : MDR \leftarrow memory (Get inst. from memory)
- T₃ : IR \leftarrow MDR (Load inst. into inst. register)
- PC \leftarrow PC + 1 (Increment PC to point to next instruction)

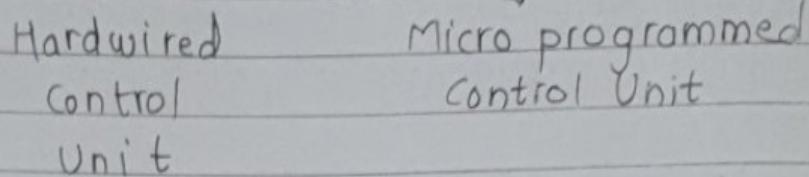
Note: when 2 or more micro-operations are independent of each other, they can be performed simultaneously in 1 clk cycle.

Control Unit Designing-

- The main task of CU is to generate internal control signals to all internal registers and also enable activities on internal bus of the processor.
- All the micro-operations are performed by the control unit.
- If the control signals are generated using combinational logic circuits then it is called as hardwired control unit.
- If these control signals are generated by software

then it is called as micro-program C.U.

Control Unit design methods



Hardwired control Unit design methods:-

- 1] state table method
- 2] Delay element method
- 3] sequence counter method.

1] State table method :-

		I/P					
		States	I ₁	I ₂	I ₃	I _m
next state	S ₁	S ₁₁ Z ₁₁	S ₁₂ Z ₁₂				S _{1m} Z _{1m}
	S ₂	S ₂₁ Z ₂₁					
S ₃							
:							
:							
S _n	S _{n1} Z _{n1}						

o/p control signals
to be generated (Cout)

- The behaviour of the control unit is represented in the form of state table as shown.
- The table shows what corresponding o/p should be generated when different inputs

various states.

- The table
- Let C_{in} and C_{out} denote the i/p and o/p variables of the control unit.
- Each row in state table corresponds to an internal state ' S_i ' of control unit.
- A state is determined by the information stored in the processor at that unit of time.
- Each column denotes the different set of external i/p signals applied to the control unit.
- The intersection of row S_i and column I_j means → when i/p i_j is applied to state S_i we get $S_{ij} Z_{ij}$ where Z_{ij} is the set of o/p signals to be activated. S_{ij} indicates that S_{ij} should become the next state of CU. A circuit is then constructed based on the state table.

Advantage -

- This is the simplest method to implement a hardwired control unit.

Disadvantages -

- If the processor has large no. of states and i/p combinations then this method cannot be used as the size of table will become too large and the circuit will become very difficult to implement.
- A state table method tends to hide useful information like existence of loops and repeated patterns. Even if 2 different states have the same behaviour, we will require separate hardware for both.
- Circuit design using this method tends to have a random structure and hence difficult to debug and maintain.

Delay Element Method -

Here the behaviour of the control unit is represented in the form of flow chart.

Every step of flow chart at time t_i will activate C_{ij} control signals.

eg. at $t_1 \rightarrow C_{1j}$ } j is instruction
at $t_2 \rightarrow C_{2j}$

where C_j is the control signal at time t_i for the execution of instruction j .

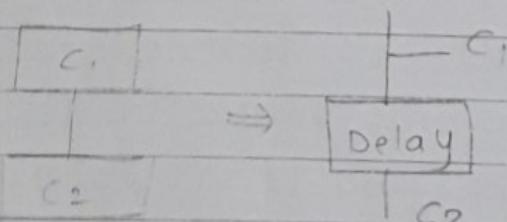
Once the flowchart is complete then individual circuit for each instructions are formed.

Instruction j will be executed when all the steps of C_{ij} are performed but all the steps should not be performed together. Instead there should be a finite time gap in between every 2 steps.

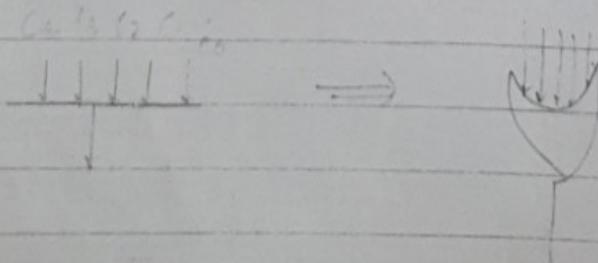
The delay circuit is introduced by D flip-flop.

Rules:-

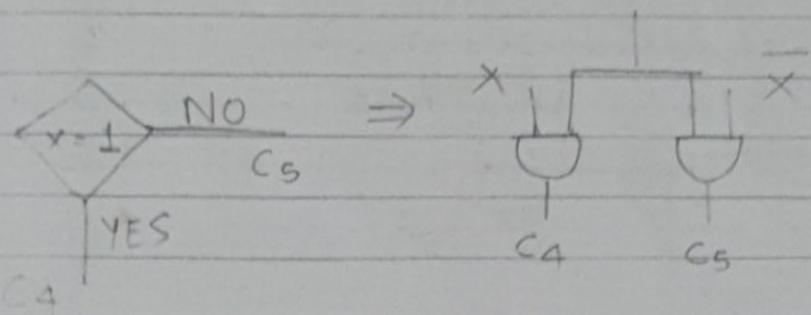
- Between 2 successive step 'D f/f'



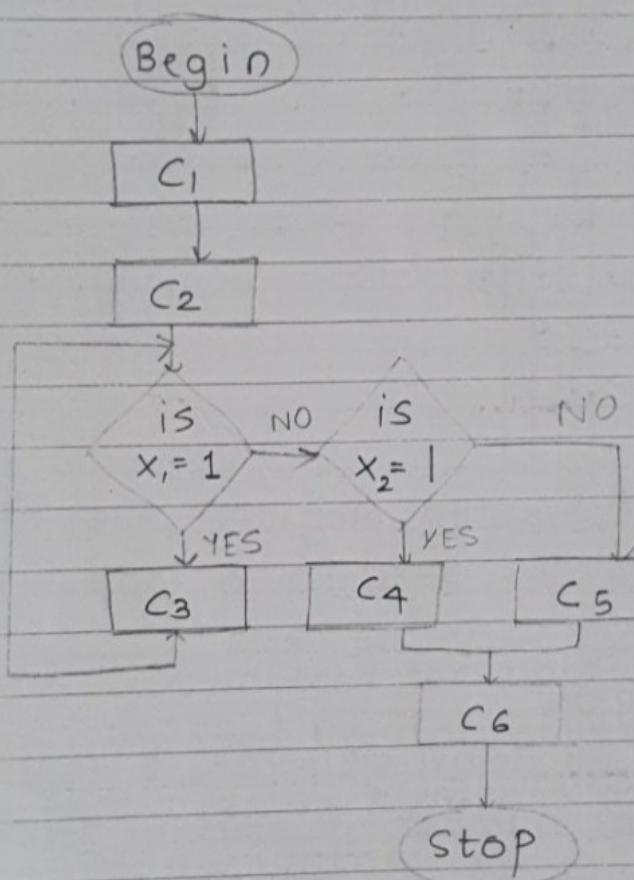
- Multiple i/p are combined by 'OR' gate



3.) Conditional branch (Decision box)

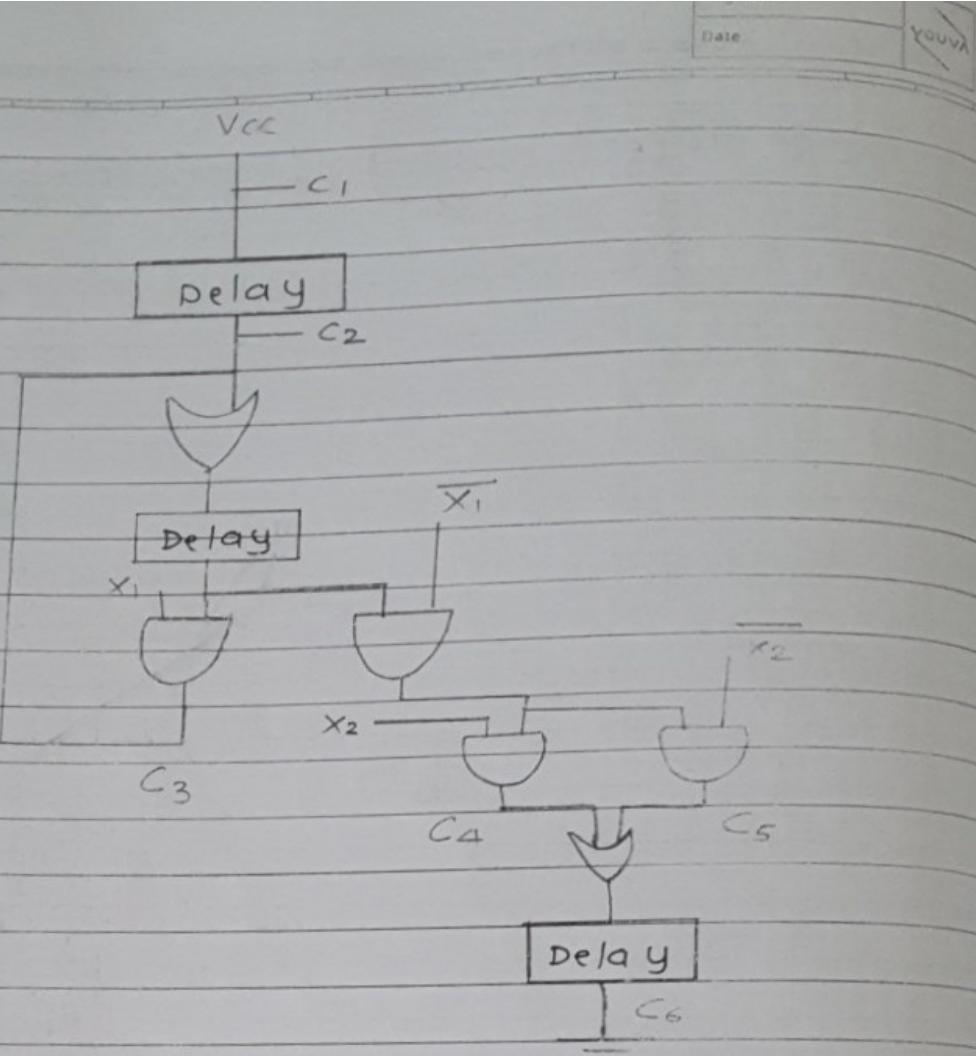


eg:-



If control signals are coming after transmission box then delay before states

If control signals are coming after start box then delay after states



Advantages-

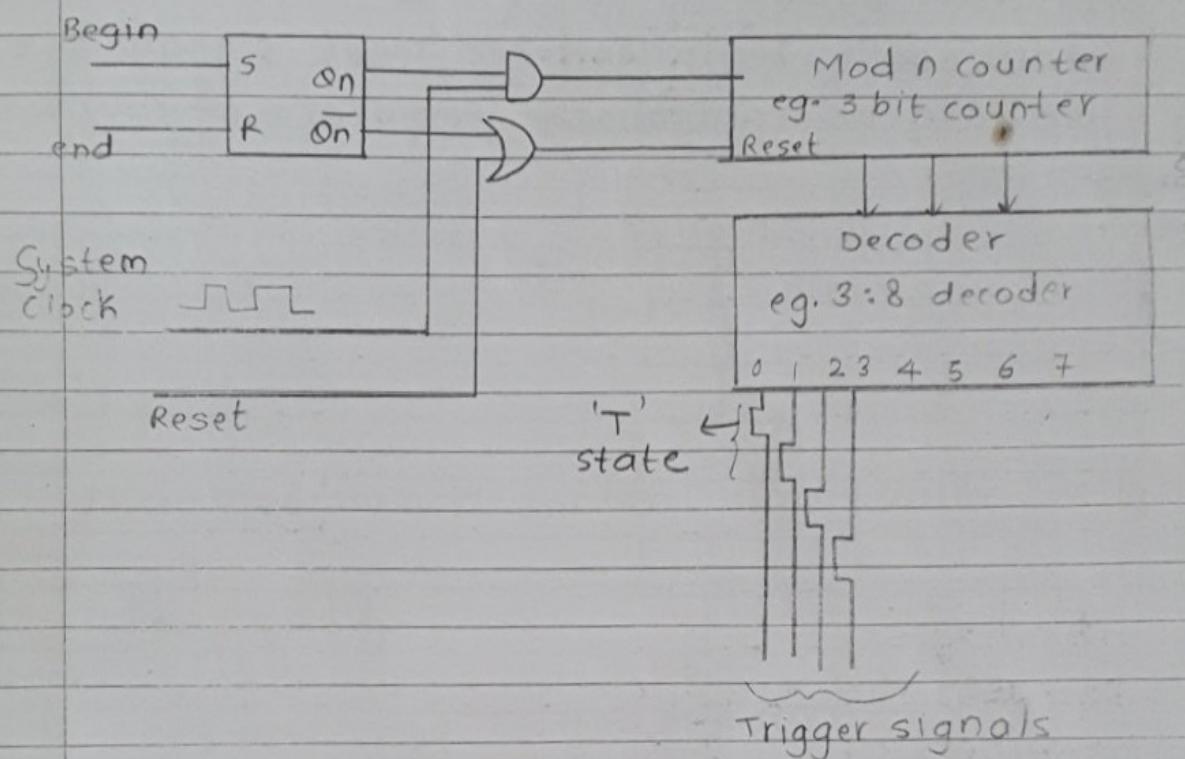
In case of looping programs, the same hardware can be used in a loop.

Disadvantages -

The number of delay elements is approximately equal to the number of states. Hence if there are lot of states, circuit can be too large. There are lot of D/l/f then synchronization of delay becomes very difficult.

Note: Since out of all D f/f only one is activated or enabled (hot) this method is also called 'one hot method'.

3] Sequence Counter Method:-



- This is the best method for hardware design.
- It is an improved version of the delay element method where n delay elements are replaced by a single modulo n counter.
- In delay element, n control signals should be produced one after the other with fixed time interval of one clock cycle.
- For this we have ' n ' D f/f each giving one clock pulse delay.
- Now in this method we use single modulo n counter and provide the same clock input to it.

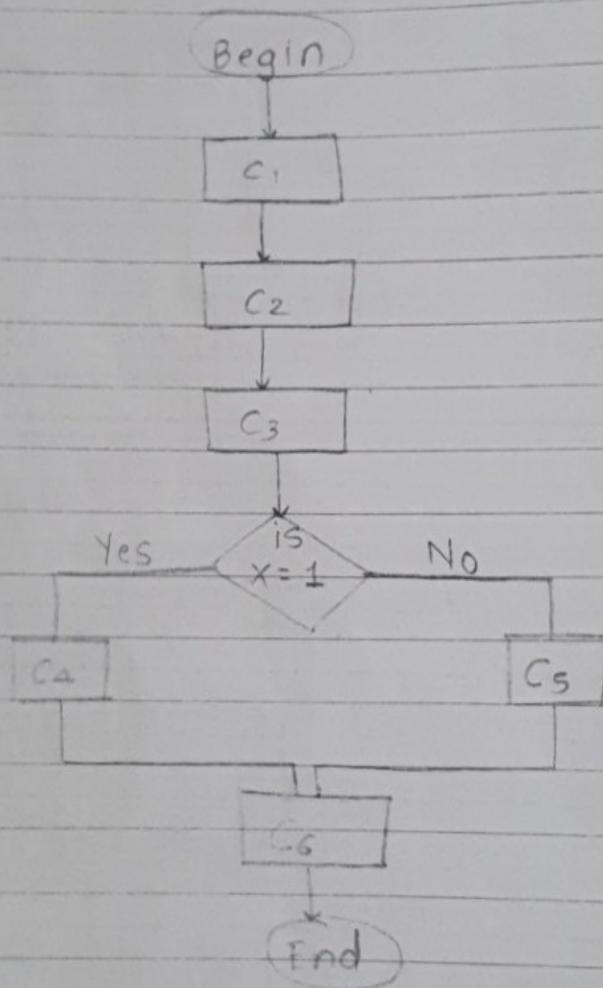
The output of this counter is connected to $k:n$ decoder.

- Here we will get n different o/p from decoder each after one clock pulse.

- As seen in above diag. the time gap between each of the n control signals produced is one clock pulse.

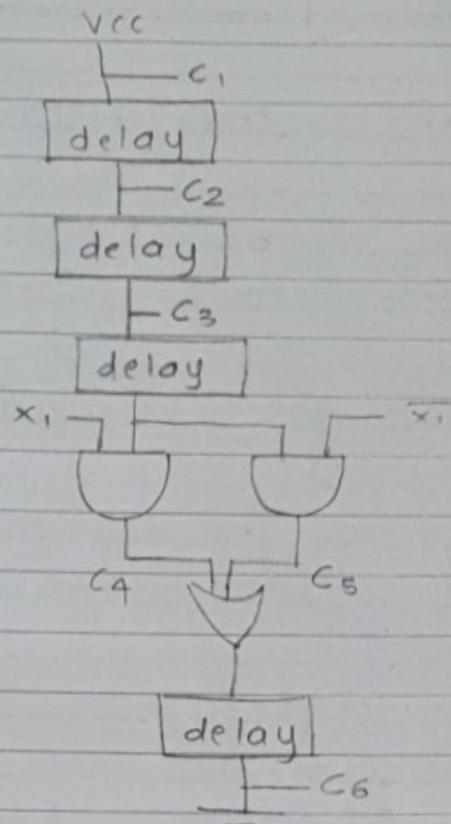
- Additionally begin, end and reset signals are provided to control the sequence of counting.

e.g.

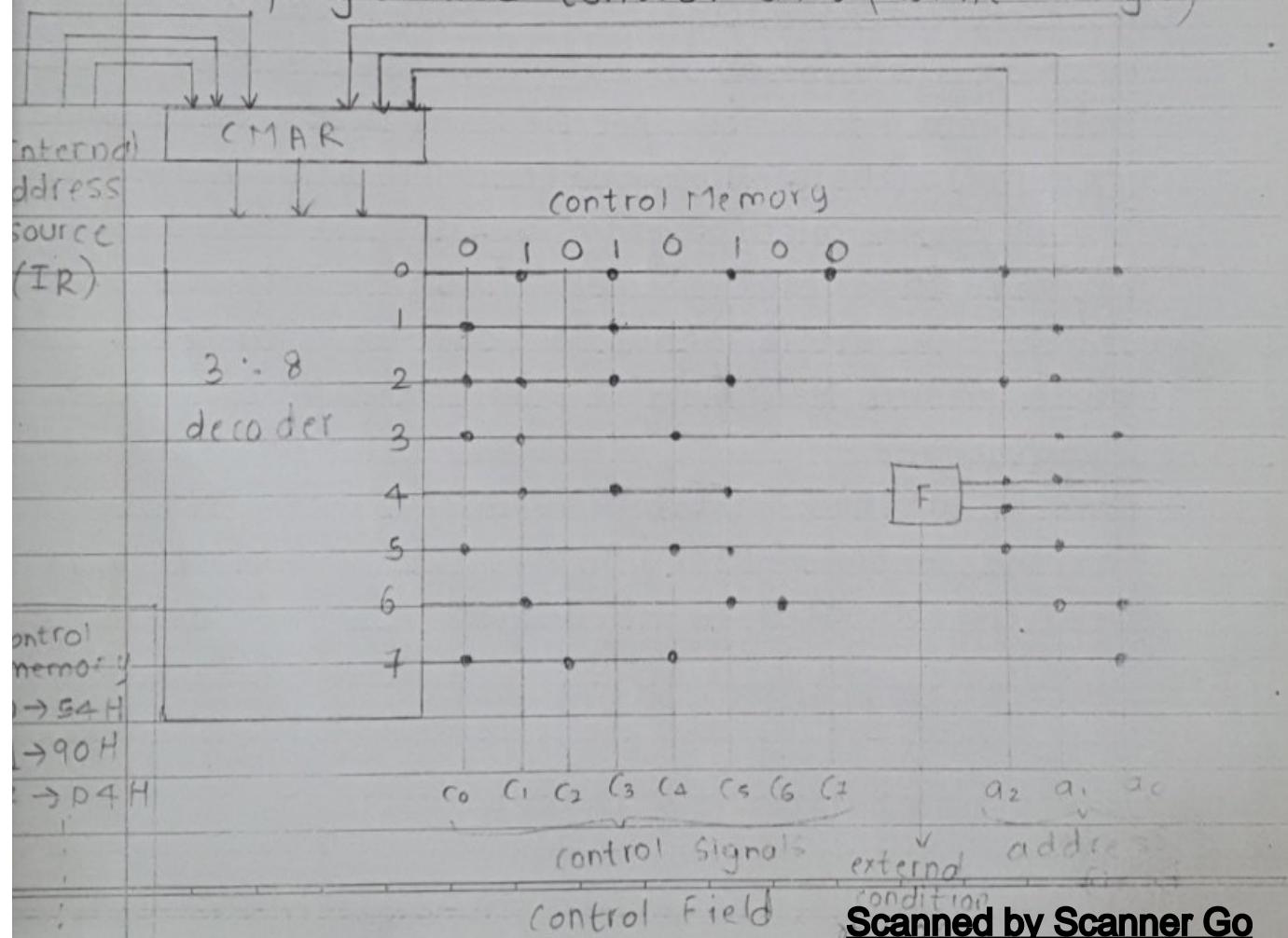


CMAR - Control memory address register

Page No. _____
Date. _____



Micro programmed control unit (Wilke's Design)



* Definition of
 ↳ micro instruction
 ↳ micro program
* Control memory concept

Book

Page No.:

Date:

YUVRAJ

Micro-programmed Control Unit -

- With the large no. of instructions the hardwired control unit becomes very complicated and not flexible.
- Each time a change has to be made , the entire CU has to be re-wired.
- This gives rise to new approach for CU design which was more software based and is called micro-programmed CU.
- In this method for every instruction there is a set of micro-instruction and these micro-instruction indicate the control signals to be activated.
- Micro instructions are stored in ROM called as control memory.
- A set of related micro-instructions used for a single machine instruction is called as micro program.
- Each micro instruction specifies the address of the next micro-instruction either explicitly or implicitly.
- This is called micro program instruction sequencing.
- The main advantage of micro-programmed CU is that the micro instruction can be changed quite easily. Hence it is more flexible.

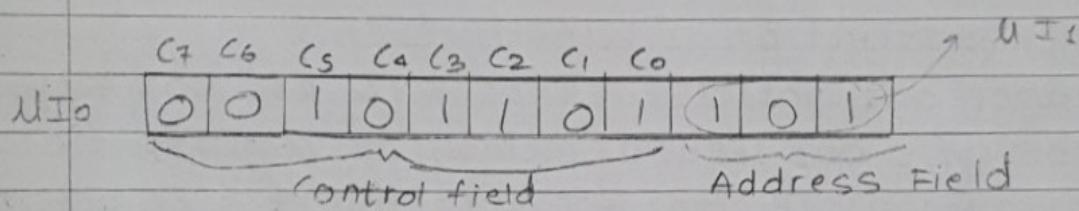
Disadvantage -

- Time is wasted in fetching micro instructions from the control memory hence it is slower in speed as compared to hardwired CU .

Wilkes's Design -

- Here the CPU has on-chip control memory.
- Control memory contains various micro-instructions, which will specify the control signals to be activated.
- Each micro-instruction has 2 parts-
 - ① Control Field - specifies control signals to be activated
 - ② Address Field - address of next micro-instruction
- Each bit of the control field directly corresponds to a control signal, hence for any micro-instruction bits which are high will be the control signals to be generated.

eg. M instruction format



- This allows us to change the control unit very easily (simply by adjusting the bits)
- Starting address ^{at} of the micro-program is produced by internal source (IR) into CMAR (control memory address register).
- Based on this ^{starting} address of the micro-instruction is selected.
- Using this micro instruction the corresponding control signals are generated and also the address of next micro instruction is placed in CMAR.
- Switch S is useful for conditional instruction (F).

Hardwired Control Unit Microprogrammed Control Unit

- Comparitively very fast - Comparitively slow as extra time is required as control memory is not present no time is wasted in instruction fetching.
- Implementation using hardware like combinational and sequential circuit.
- Debugging such circuits - It is software based so is difficult.
- Difficult to handle complex instruction.
- Cheaper as control memory is not required.
- Instruction set size is less than 100.
- Doesn't support emulation.
- Due to their rigidness they are more suited for RISC computers.
- Uses less chip area.
- Design procedure is some what complicated.
- Comparitively slow as extra time is required to fetch the micro instruction.
- Implementation using software.
- Easier to handle complex instruction.
- Expensive as control memory is required.
- Instruction set size does not have any restriction.
- Emulation is possible by simply coping the control memory of one CPU to another.
- Due to their flexibility, they are more suited for CISC computers.
- Uses more chip area due to control memory.
- Design procedure is systematic.

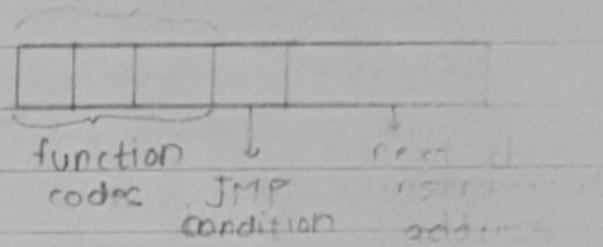
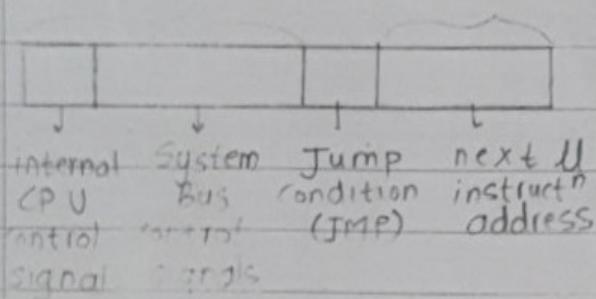
Micro instruction designing types:-

- The main task of the U instruction is to generate the control signals.
- The control field of U instruction indicates the control signals to be activated.
- There are 2 ways to design U instructions.

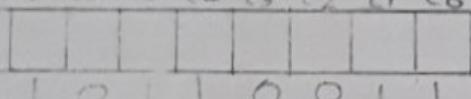
Horizontal U instruction Vertical U instruction

1) Format

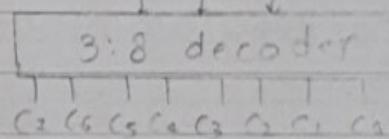
Format -



2) e.g. a₀ a₁ a₂ a₃ a₄ a₅ a₆ a₇ 8 bit



e.g. 1 0 0 1 3 bit



Here each bit of U instruction (control field) represents control signals to be produced. Hence whichever bits are logic 1 those signals are produced.

Here bits of the U instruction (control field) are decoded through a decoder. Hence less no. of signals the instruction will have to contain more no. control field

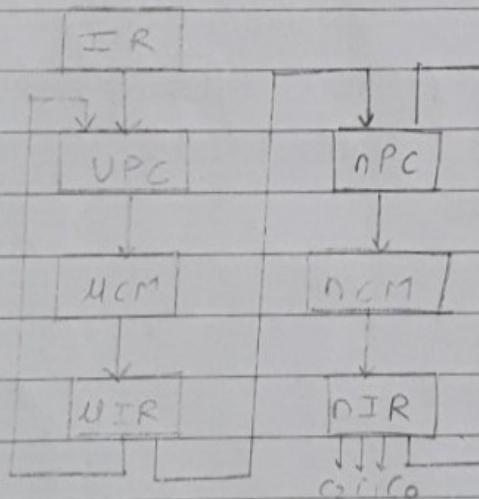
3) n bits will produce only n bits will produce 2^n control signals hence control signals through if we want more control signals the instruction will have to contain more no. control field

bits in control field.

Hence it grows horizontally.

- 4) No external decoders are required. Hence circuit is simple Since decoders are required cost and complexity increases.
- 5) Multiple control signals can be produced simultaneously. only 1 signal can be produced at a time.
- 6) As it is showing multiple control signals at a time so high operating speed. low operating speed.

Nano Programming -



- For large CU requiring many control signals neither vertical nor horizontal designs are optimal.

- Horizontal design makes operation fast but tremendously increases the size of

- Vertical design will keep μ instruction size small but can produce only one control signal at a time resulting in slow operation.
- Hence combination of both designs used gives rise to a new technique called as nano programming.
- Here 2 level control memory storage is used.
- The higher level micro control memory (UCM) uses vertical format micro-instructions.
- The lower level nano control memory (ncM) uses horizontal format micro-instructions.
- The micro instruction from the micro control memory is decoded to select nano instruction.
- The nano instruction from nano control memory actually produces the required control signals.
- Hence various control signals can be produced simultaneously without increasing the size of μ instruction.

Disadvantage -

- However operation becomes slower as 2 memories have to be accessed.

eg.

processor has

Arithematic instruction - 8

logical instruction - 8

data transfer - 8

floating point - 8

32

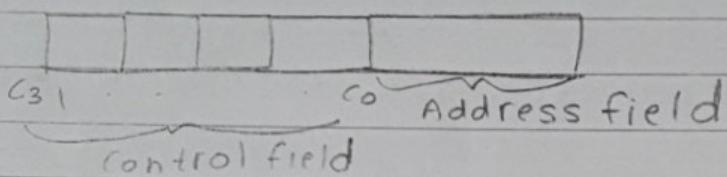
00 → Arithmetic

01 → logical

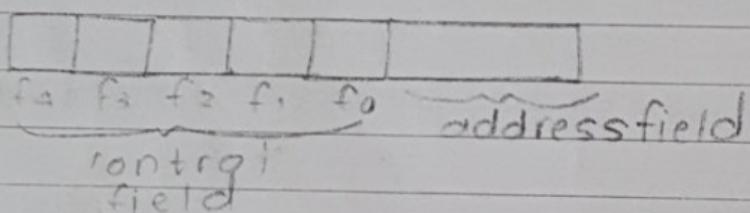
10 → data

11 → floating

Horizontal -



Vertical -



- If we use only horizontal, then we will need 32 bits.
- This will need a lot of memory.
- Hence we use vertical design to generate the type of instruction.
- And then the horizontal 8 bit design is used to generate the actual control signals.