(c)

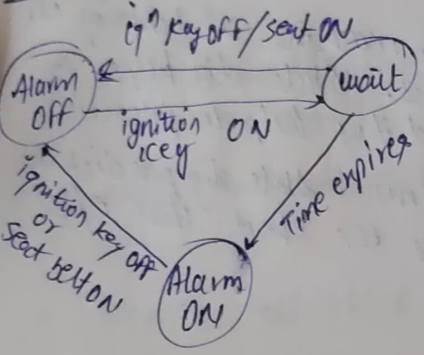
Q what do you mean by FSM & FSMD.

FSM (final-state Machine) - It is a mathematical model of sequence of events & actions describing a certain (logical process) which are finites.

formally FSM is specified by 5 entities.

- input signal
- output signal
- next state function
- output function
- Symbolic States

Example:



- The derivation of FSM starts with a more abstract model, such as state diagram & ASM (algorithm state machine)
- Both shows the interaction and transitions between the states.
- Both utilizes symbolic notations, shows the transitions among state and indicate the output value under various cond^n.
- A state diag & ASM chart can capture all the needed info like (input signal, output signal, next-state function, output function) in a single graph.

FSMD (final state Machine with datapath) :. It is a mathematical model. sometimes used to design digital logic or computer programms.

- FSMD is a digital system composed of final state machine, which control the programs flow and a datapath which performs data processing opr^n.
- FSMDs are essentially sequential programs in which statements has been scheduled into states that results is more complex state diagram.

Datapath:

- Contains ALU for transforming data through opr^n such as +, -, logical, AND, logical OR, invertig & shifting.
- also contains registers capable of storing data which are temporary.
- it also generated status signal with help of ALU.
- within datapath the internal data bus where the data travel.
- External data bus is used to brought data to and from data memory.

- FSMD is more powerful that FSM because it uses variables or arithmetic opr^n / cond^n.
- FSMD abstraction is equivalent to Turing Machine.

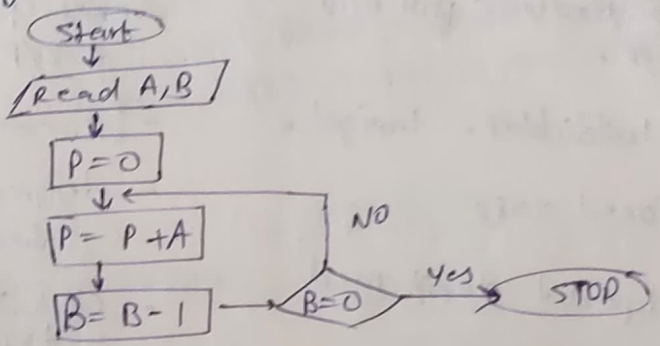
8 Design System using Datapath & Control path: (6)

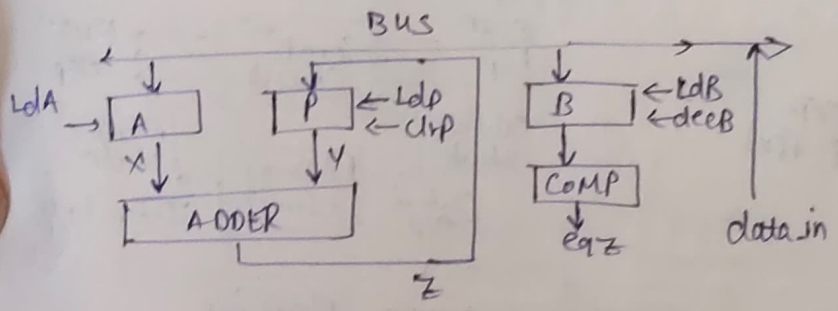
In a Complex digital sys. the hardware is typically partitioned into two parts.

(i) Datapath:
- which consist of functional units where all computations are carried out.
- Typically consist of registers, multiplexers, bus, adders, multipliers, counters and other functional blocks.

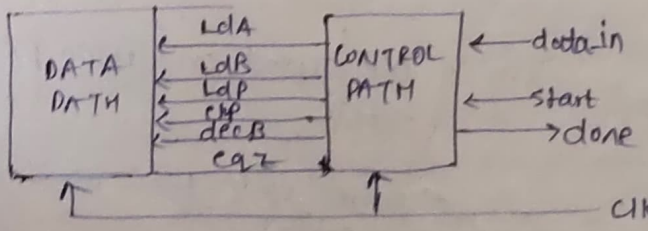
(ii) Control path:
- which implements a FSM and provide control signals to the data path in proper sequence.
- in response to control signal, various opn are carried out by data path.
- Also takes input from data path.

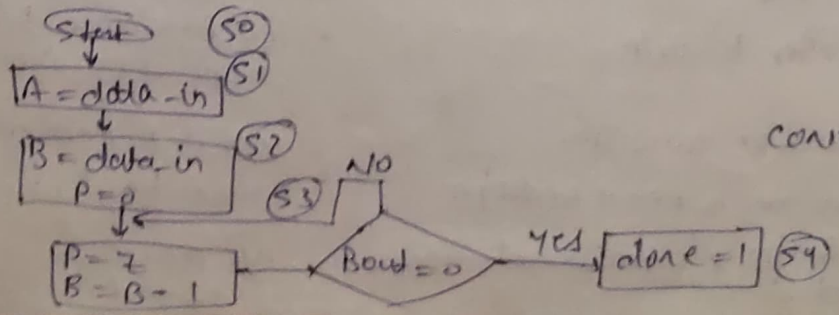
Example: Multiplication by Repeated add:





DATA PATH





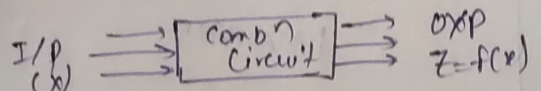
CONTROL PATH

Ques 6

Combinational circuit

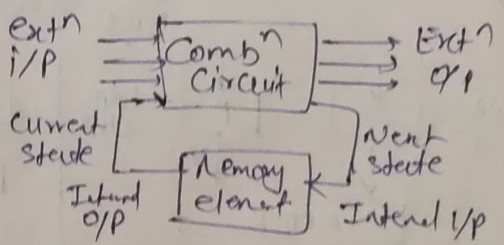
- defined as the time independent circuit which do not depend upon previous input to generate any output.
- speed is fast
- it is easy to use & handle
- Do not have memory element
- used for arithmetic as well as boolean opr^n.
- exist no feedback path b/w i/p & o/p.
- Elementary building block - Logic gates.
- It is designed easy
- clock independent do not need triggering

Block dig.



Sequential Circuit

- which are dependent on clock cycles and depend on present as well as past input to generate any output.
- speed is slow
- not easy to use & handle
- Circuits have memory element.
- Mainly used for storing data
- exist feedback path b/w i/p & O/P.
- Elementary building block - flipflops.
- designed tough as compare to cc.
- clock dependent so need triggering



Mealy Machine

- Output depends on both upon present state & present i/p
- has less state than Moore machine.
- reacts faster to in/p
- o/p is placed on transition
- less hardware is need to design
- change in i/p, o/p also get changed.

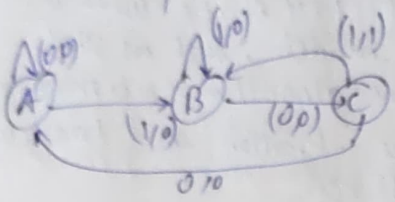
Moore Machine

- o/p depend only upon present state.
- more state than Mealy machine.
- react slower b/c more Logic is required.
- o/p is placed on states.
- more hardware required
- No change in o/p if i/p changed.

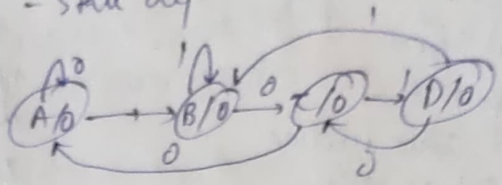
- very difficult to design
- represented by 6 tuples
  $(Q, \Sigma, O, \delta, X, q_0)$

- easy to design.
- " —
  $(Q, \Sigma, O, \delta, X, q_0)$

- State dig.

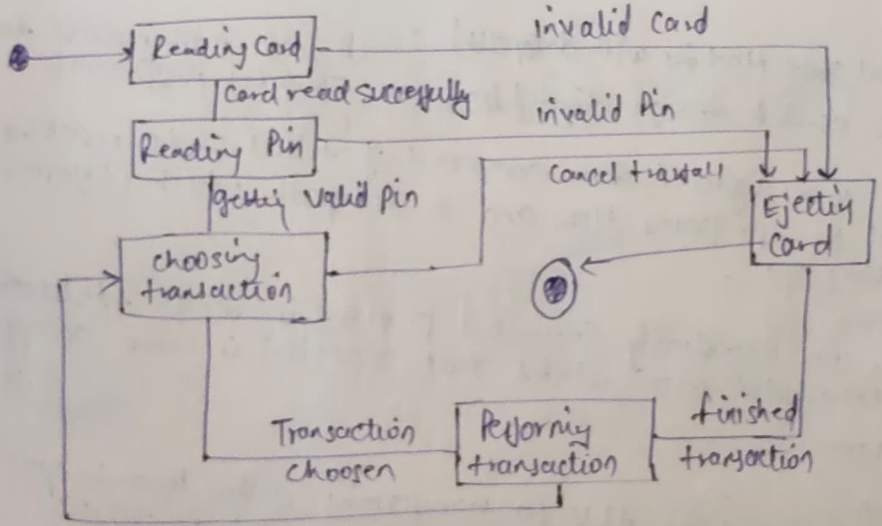


- state dig

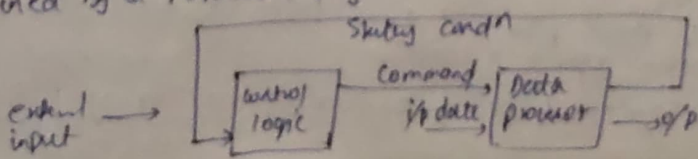


→ State diagram:-

It is the graphical representation of a state machine and one of the 14 UML dig types for software and systems. State dig helps to visualize the entire life cycle of object and thus help to provide a better understanding of state based systems. State dig predict the permitted states, transition states as well as the event that effect these transition.



→ Algorithmic state machine - It is a method for designing finite state machine
- In digital sys. binary info. is divided into data and control info.
- Data info manipulated by various opr^n like arithmetic, shift, logic etc. these opn^ are implemented by Multiplexers.
- Control info gives various command signal that help to perform various data opr^.
- The control sequence and data processing of digi system can be determined by a hardware algo.



→ ASM Method

Step-1 create an algori. using pseudocode

Step-2 Convert pseudocode into an ASM chart.

Step-3 Design the datapath based on ASM chart.

Step-4 Create a detailed ASM Chart based on datapath.

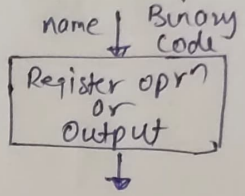
Step-5 design a control logic based on the detailed ASM Chart.

→ ASM Chart

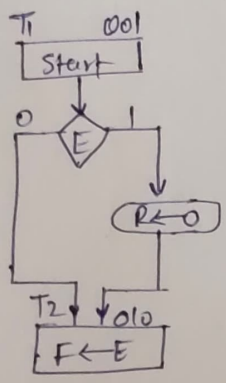
It is a special type of flow chart that is used to describe the sequential oprn of a digital circuit. The ASM chart determines the sequence of events, timing relationship between states of sequential controller and the event that happen while going from one state to another.

The ASM chart is composed of 3 basic elements.

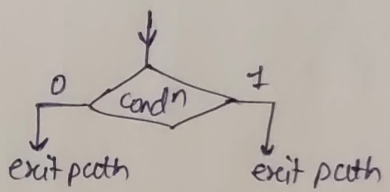
- State Box



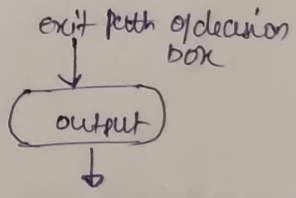
example



- Decesion Box



- condn box



Q

CISC	RISC
- Complex instruction Set computer	- Reduced instruction set Computer.
- Generally the total no. of instruction is large. for CPU	- total no. of instruction for CPU is small.
- It emphasize hardware to optimize instruction set	- It emphasize software to optimize instruction set.
- require single register set to store instruction	- require multiple set of registers.
- use of PIPELINE are difficult	- use of pipeline is simple.
- uses limited no. of instruction so require less time.	- uses large no. of instruction so take more time to execute
- has fixed formed instruction	- It has variable format instruction
- take more memory	- takes less memory
- ARM, AVR, SPARC.	- exc: Intel x86 CPUs. AMD.

→ ASM Method

Step-1 create an algori. usig pseudocode

Step-2 Convert pseudocode into an ASM chart.

Step-3 Design the datapath based on ASM chart.

Step-4 Create a detailed ASM Chart based on datapath.

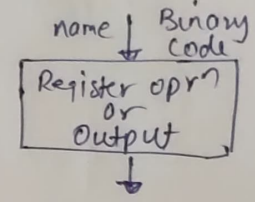
Step-5 design a control logic based on the detailed ASM Chart.

→ ASM Chart

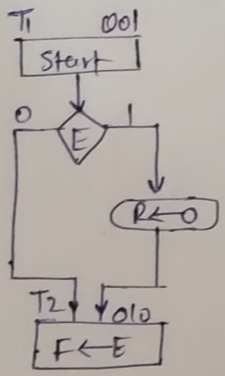
It is a special type of flow chart that is used to describe the sequential oprn of a digital circuit. The ASM chart determines the sequence of events, timing relationship between states of sequential controller and the event that happen while going from one state to another.

The ASM chart is composed of 3 basic elements.

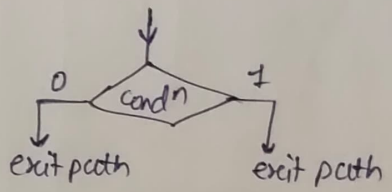
- State Box



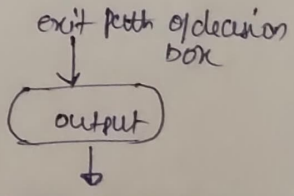
example



- Decesion Box



- condn box

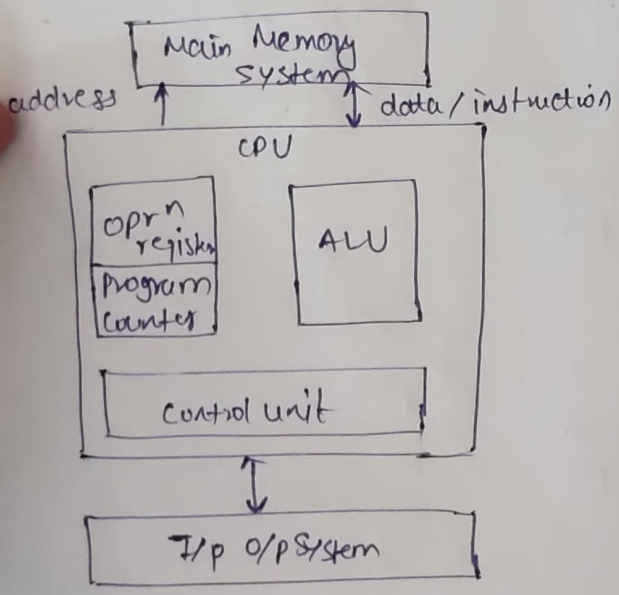


Q

CISC	RISC
- Complex instruction Set computer	- Reduced instruction set computer.
- Generally the total no. of instruction is large. for CPU	- total no. of instruction for CPU is small.
- It emphasize hardware to optimize instruction set	- It emphasize software to optimize instruction set.
- require single register set to store instruction	- require multiple set of registers.
- use of PIPELINE are difficult	- use of pipeline is simple.
- uses limited no. of instruction so require less time.	- uses large no. of instruction so take more time to execute
- has fixed format instruction	- It has variable format instruction
- take more memory	- takes less memory
- ARM, AVR, SPARC.	- exc: Intel x86 CPUs. AMD.

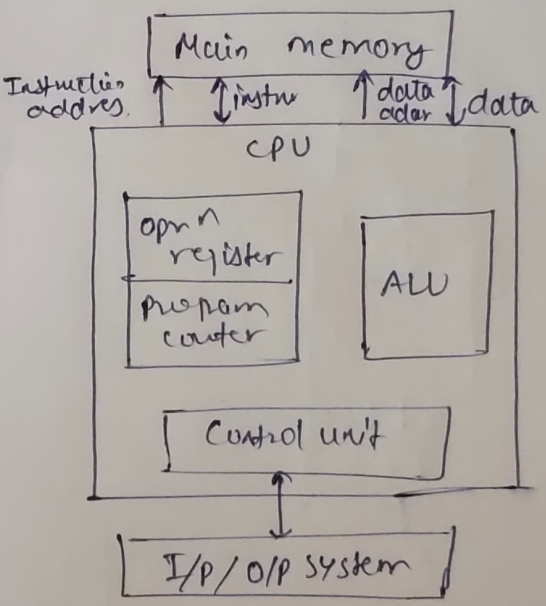
Q Von-neumann Architecture

- Single memory to be shared by both code & data
- requires two clock cycle Processor need to fetch code in diffent clock cycle & data in other
- higher speed thus less time consuming
- simple in design
- cheaper in cost



Q Harvard Architecture

- Seperate memory to for code & data.
- single clock cycle is sufficient as seperate bues are used to acces code & data.
- slower in speed, thus more time consuming
- complex in design.
- costly in cost as compare to von-neumann.



Q write a program to add a 16 bit no. the 3CE7H & 3B8DH Place the sum in R7, R6 rejistr, where R6 register should have lower byte

```
CLR C          : Make carry (C4) = 0
MOV A, #E7H    : lower byte of operand 1 in A.
ADD A, #8DH    : stores LSB (least Significat Bit) of Result in R6
MOV R6, A      : Add lower byte of operand 2 with A
MOV A, #3CH    : higher byte of operand 2 in A.
ADD C, A #3BH  : Add higher byte of operand 1.
MOV R7, A      : stores two MSB (Most Significod bit) in R7.
End
```

E7	1110	0111
80	1000	1101
$C_4 \to 1$ (1)	0111	0100
	7	4H

3C	0011	1100
3B	0011	1011
	0111	1000
	7	8H

8 4 2 1

10 – A
11 – B
12 – C
13 – D
14 – E
15 – F
16 – G
H – 17

---

automata ... r (theory)
... derived string (numerical)
... Ambi... (numerical)
... (tree)
CF... ... CFG (numerical)
... CFA to CFG.