Q. - What do you mean by single purpose processor & general purpose processor. Write difference b/w them.

Ans - A single purpose processor is designed to execute only one program. It performs particular computation task. It is also known as co-processor, accelerator or peripherals.

A general purpose processor is a macroprocessor that is not tied to or integrated with a particular language or piece of software.

Single purpose processor	General purpose processor
(i) Run a single or few specialized application often known at system design time.	(i) Used for general purpose software: Intended to run a fully general set of applications that may not be known at design time.
(ii) May require application-specific capability.	(ii) No application specific capability required.
(iii) No end-user programmable	(iii) End-user programmable.
(iv) Minimum code size is highly desirable.	(iv) Minimizing code size is not an issue.
(v) Low power & cost requirements	(v) Heavy weight, multi-tasking OS. High power & cost constraints.

Q. What is NRE (Non-Recurring Engineering) cost.

Ans- Non-recurring engineering (NRE) cost refers to the one-time cost to research, design, develop and test a new product or project enhancement. When budgeting for a new product, NRE must be considered to analyze if a new product will be profitable. Even though a company will pay for NRE on a project only once, NRE costs can be prohibitively high & the product will need to sell well enough to produce a return on the initial investment.

NRE is unlike production costs, which must be paid constantly to maintain production of a product. It is a form of fixed cost in economic terms.

Once a system is designed any number of units can be manufactured without increasing NRE cost.

NRE can be also formulated & paid via another commercial term called Royalty Fee. The Royalty Fee could be a percentage of sales revenue or ~~per~~ profit or combination of these two.

In a ~~product~~ project type company, large parts of the project represent NRE. In this case the NRE costs are likely to be included in the first projects' costs, this can also be called research & development (R&D).

Q. What do you mean by FSM (Finite State Machine) & FSMD (Finite state machine with datapath).

Ans- FSM → A finite state machine is a sequential circuit with random next-level logic.

The derivation of an FSM starts with a more abstract model, such as a state digram or an algorithm state machine (ASM) chart. Both show the interaction & transitions b/w the internal states in graphical formats.

- Formally an FSM is specified by five entities —
  (i) symbolic states
  (ii) input signal
  (iii) output signal
  (iv) Next-state function
  (v) output function.

FSMD → A FSMD is a mathematical abstraction that is sometimes used to design complex digital logic devices or computer programs.

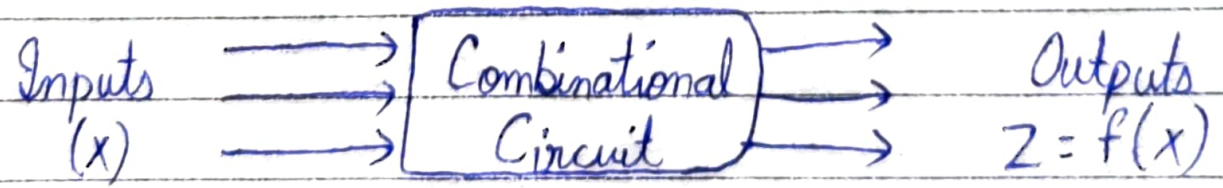
An FSMD combines a controller, modelled as a FSM & a datapath. The datapath receives commands from the controller & performs operations as a result of executing those commands.

The FSMD model will be used throughout the remainder of the book as a reference model for the 'hardware' part of HW/SW codesign.

Q. Discuss combinational & sequential machines.

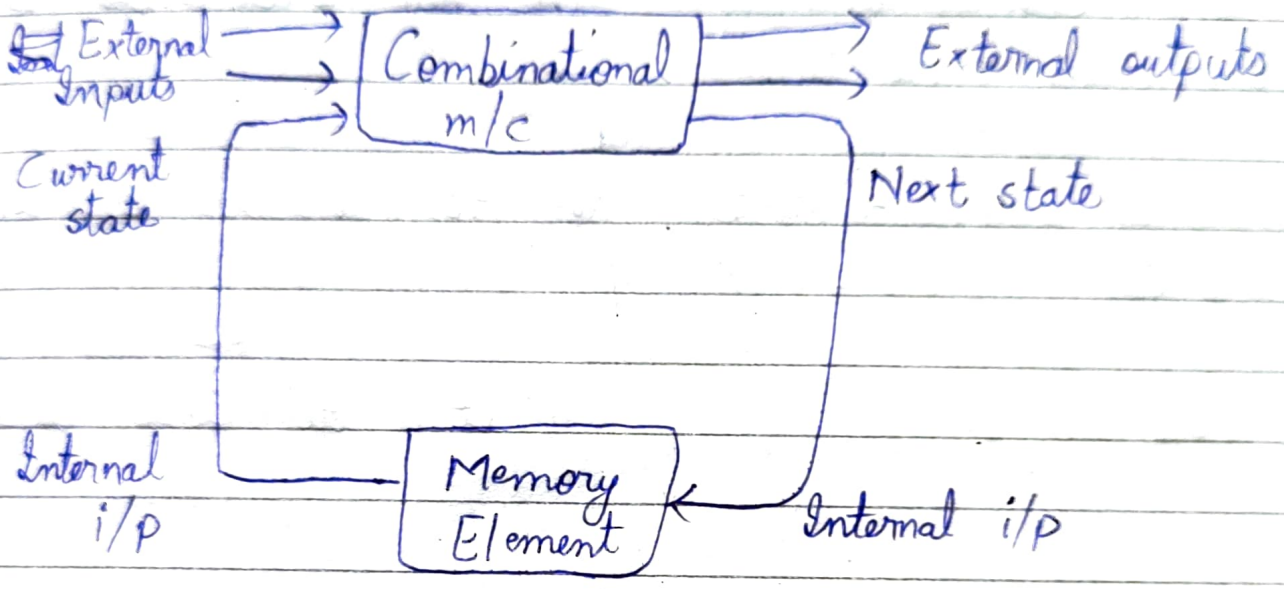
Ans- Combinational machines are defined as the time independent circuits which do not depends upon previous inputs to generate any output are termed as combinational machines.

(i) In this output depends only upon present input.
(ii) Speed is fast
(iii) There is no feedback b/w input & output.
(iv) Used for arithmetic as well as boolean operations.
(v) They don't have capability to store any state.
(vi) As they don't have clock, they don't require triggering



Sequential Machine : They are those which are dependent on clock cycles & depends on present as well as past inputs to generate any output.

(i) In this output depends upon present as well as past i/p.
(ii) Speed is slow
(iii) There exists a feedback path b/w input & output.
(iv) Mainly used for storing data.
(v) Elementary building blocks : flip-flops.



Q. What is the basic difference & similarities b/w Mealy and Moore model of machine.

Ans-

Mealy m/c	Moore machine
(i) It changes its output on the basis of its present state & current i/p	(i) It depends only on the current state. It does not depend on current i/p.
(ii) It places its output on the transition	(ii) It also places its output on ~~transition~~ states.

Moore	Mealy
(iii) If input changes, output does changes.	(iii) If input changes, output also changes.
(iv) More no. of states are required	(iv) Less no. of states are required.
(v) There is less h/w requirement for circuit implementation	(v) There is more h/w requirement.
(vi) They react slower to input (One clock cycle later)	(vi) They react faster to inputs.
(vii) Synchronous output & state generation	(vii) Asynchronous o/p generation
(viii) Easy to design.	(viii) It is difficult to design.

Q. What do you mean by state diagram & ASM representation of a FSM. How ASM chart representation is more advantageous than state diagram representation.

Ans- State digram : A state diagram consists of nodes, which are draws as circles (also known as bubbles) and one-direction transition arc.

- A node represents a unique state of the FSM and it has a unique symbolic name.

- An arc represents a transition from one state to another and is labelled with the condition that will cause the transition. The condition is expressed as a logic expression composed of input signals.

- The output values are also specified on the state digram.

• ASM chart : An algorithmic state machine (ASM) chart is an alternative method for representing an FSM. Although an ASM chart contains the same amount of information as a state diagram, it is more descriptive.

We can use an ASM chart to specify the complex sequencing of events involving commands (input) & action (output).

- An ASM chart representation can easily be transformed to VHDL code.
- An ASM block consists of one state box & an optimal network of decision boxes & conditional output boxes.

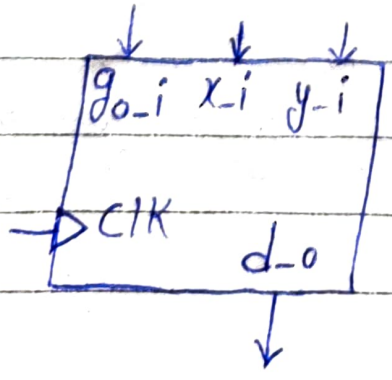
** An ASM diagram offers several advantages over state diagram
(i) For larger state diagrams, ASM diagrams are easier to interpret.
(ii) Conditions for a proper state diagram are automatically satisfied
(iii) ASM diagrams are easily to other forms.

Q. Design a processor to compute the LCM of two numbers.

Ans- Algo:

```
int x, y, z
while (1) {
    while (!go_i);
    x = x_i;
    y = y_i;
    z = x * y;
    while (x != y) {
        if (x < y)
            y = y - x;
        else
            x = x - y;
    }
    d_o = x;
}
```

```
int x, y, max, lcm;
while (1)
{ while (!go_i);
    x = x_i;
    y = y_i;
    if (x > y)
        max = x;
    else
        max = y;
    while (1) {
        if ((max % x == 0)
            && (max % y == 0))
        { lcm = max;
            break;
        }
        max ++;
    }
    d_o = lcm;
}
```



Q. Design a processor to compute the HCF of two numbers.

```
int x, y;
while (1)
{   while (! go_i) ;
    if (x_i >= y_i)
    { x = x_i;
      y = y_i;
    }
    else {
        x = y_i;
        y = x_i;
    }
    while (y != 0)
    { r = x % y;
      x = y;
      y = r;
    }
    d_o = x;
}
```