Unit-2

(1) Embedded Networking

- It deals with the network design and topology, hardware devices, and communication or data exchange protocals needed to connect & exchange information between embedded systems.
- Range of wired & wireless communication options available for implementing networking capabilities into embedded system.
- Embedded system form the basis for the IoT.

(2) I/O parts.

- I/O parts in ES are interfaces through which a up or micro-controller can communicate with external devices or other components within the system.
- These parts allow the system to receive input from the ~~output~~ outside world (e.g. sensors) and send output to other devices (e.g. actuators, displays)

Example:- Parts P0, P1, P2 & P3 in 8051
COM1 & COM2 parts in IBM PC
USB, UR UART etc.

Types of Ports.

Serial:-
- i) Synchronous serial input (audio, video, network etc) interprocess data transfer.
- ii) Synchronous serial output (")
- iii) Asynchronous serial UART input
- iv) Asynchronous serial UART output (Both I/P & O/P ex-modem)

Parallel:-
- i) Parallel port one bit input
- ii) Parallel part one bit output
- iii) Parallel part multi-bit input
- iv) Parallel part multi-bit output

③ Modes of Communication

i) **Simplex:-** Simplex communication is a mode of data transmission where communication occurs in only one direction.

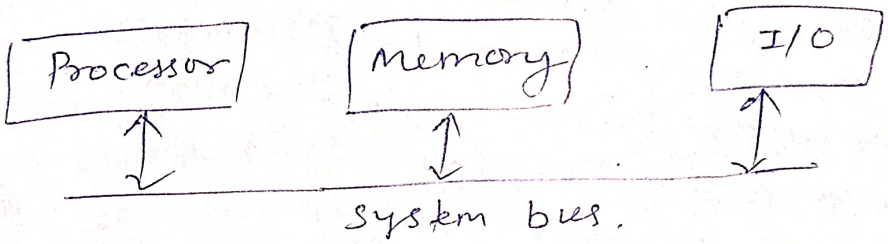
- On a simplex system, one device can only send data while the other can only receive data.

ii) **Half-Duplex :-** Half-duplex communication is a mode of communication where data transmission can occure in both directions, but not simultaneously.

- On half duplex system, a device can either send or receive data at a time.

iii) **Full duplex :-** Full duplex means that at an instant the communication can be both ways.

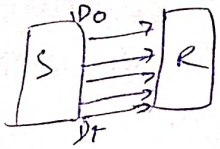
④ Bus System



- Inside an embedded system are numerous component such as the CPU, memory, I/O device etc.
- Inevitably these components need to exchange data, and data is moved from one component to another through interconnections.
- We call these interconnections the bus. (Primus bus)

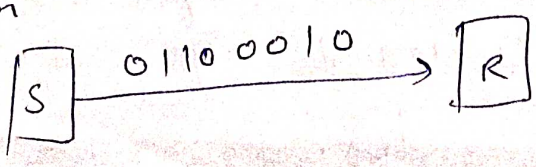
(5) Differences

Serial Communication	Parallel Communication.
1) Data transmitted serially, one bit at a time	1) Data transmitted, all bits at a time.
2) Low speed	2) High speed
3) It has single transmission line	3) It has multiple transmission lines
4) Serial communication do not have any crosstalk problem	4) Parallel communication may have crosstalk problem
5) less expensive	5) more expensive
6) The bandwidth is higher	6) The bandwidth is lower
7) It is not affected with noise problem	7) It may suffer with noise problems.
8) serial communication even works at high frequencies	8) Parallel communication may not work properly
9) It covers long distance when compared to parallel communication	9) It is used for short distance.
Ex - Serial communication b/w a computer & modem	→ Parallel communication b/w motherboard & hard disc.



(6) Serial transmission:-

- Bits of a byte are serially transmitted one after other
- The shift register is used for serial transmission
- The byte to be transmitted is first stored in a shift register. Then these bits are shifted from MSB to LSB bit by bit synchronization with the clock.
- Bits are shifted right by one position per clock cycle.
- As an advantage only one wire is used in serial transmission



Advantage:-

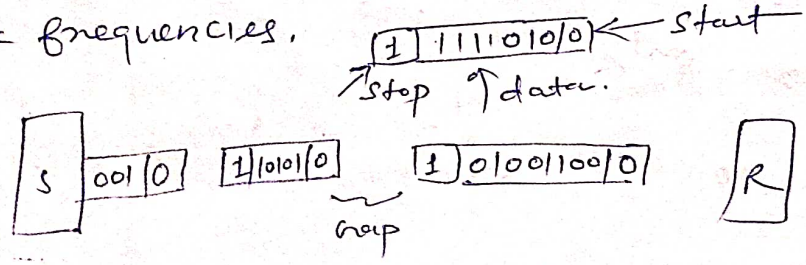
- Only one wire is required.
- Reduction in cost due to less number of conductor wire.
- It is most used method for long distance data transfer.

Disadvantages:-

- Since there is only one line of transmission therefore the speed of transmission is quite slow.
- If we have to increase the speed of data transfer then it is necessary to increase the clock frequency. But we cannot increase frequency beyond limit.

Asynchronous serial transmission:-

- Timing of the signal is not an issue. The information can be sent & receive as per mutual agreement of the sender & Receiver.
- Asynchronous transmission is at byte level only because transmission of bits is always synchronized.
- Asynchronous transmission is eased by two bits, namely start bit as '0' & stop bit as '1'
- We send '0' bit to start the communication & '1' bit to stop the transmission.
- There can be a time delay between communication of two bytes. Only one byte is sent at a time. After a gap of time next byte is transmitted.
- The transmitter & receiver may function at different clock frequencies.



Advantage :-

- It is a flexible data transmission method
- Synchronization between the transmitter & receiver is not necessary.
- It is possible to transmit signals from the sources having different bit rates.
- The transmission can commence as soon as the data byte to be transmitted becomes available
- This mode of transmission is easy to implement.
- It is a cheap scheme in terms of money
- It is an effective scheme

Disadvantages :-

- Additional bits called start and stop bits are required to be used.
- The timing error may take place because it is difficult to determine synchronicity.
- It has slower transmission rate.

(8) Synchronous serial transmission.

- Synchronous transmission is carried out under the control of common master clock - bits which are being transmitted are synchronized to the same reference clock.
- The stream of bits is combined into bigger frames, which may comprise of more than one byte.
- Each byte is transmitted without gap between the next byte.
- No start and stop bits are used. Instead the bytes are transmitted as a block (frame or packets) space by fixed time intervals in a continuous stream of bits.
- The receiver operates at the same clock frequency as that of transmitter.

- It is the duty of the receiver to separate the bits sent in group.
- There is an inter block idle time which also is filled with idle characters.

Advantages:
- The main advantages is speed. The speed of transmission is much higher than that asynchronous transmission
- This is due to the absence of gaps between the data units and absence of start stop bits.
- Timing errors are reduced due to synchronization.

Disadvantage:
- The timing is very important. The accuracy of the received data is dependent entirely on the ability of the receiver to count the received bits accurately.
- The transmitter and receiver have to operate at the same clock frequency.
- Thus requires proper synchronization which makes the system complicated.

(9) Serial communication protocals.
A protocal is a set of rules agreed by both the sender & receiver on.
- How the data is packed
- How many bits constitute a character
- When the data begins & ends

(10) Serial bus communication protocals.
- In embedded systems, the communication means the exchange of data b/w two microcontrollers in the form of bits.

Advantage of SPI:-

- This exchange of data bits in microcontroller is done by some set of defined rules known as communication protocals.
- Now if the data is sent in series i.e. one after the other then the communication protocal is known as serial communication protocal.
- More specifically the data bits are transmitted one at a time in sequential manner over the data bus.

(10) Serial Peripheral Interface (SPI)

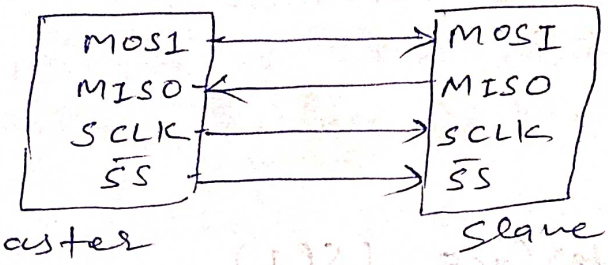
- Synchronous serial communication interface specification used for short distance communication.
- Interface was developed by Motorola in 1980s.
- SPI devices communicate in full duplex mode using a master-slave architecture with a single master.
- The master device originates the frame for reading and writing.
- Multiple slave devices are supported through selection with individual slave select (SS) lines.
- Unique feature of SPI data can be transferred w/o interruption.
- Any number of bits can be sent or received in a continuous stream
- In SPI, only one side generates the clock signal.
- The side that generates the clock signal is called 'master' and other called "slave"
- The master is the controlling device (usually a microcontroller), while the slave (usually sensor, display, memory chip) takes instruction from the master.

Advantages of SPI :-

SPI signals.

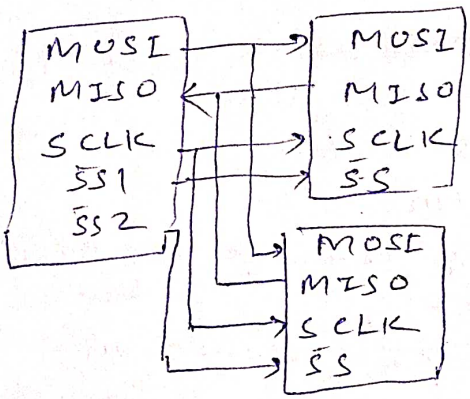
The SPI bus specifies four logic signals.

- SCLK : Serial clock (output from master)
- MOSI : Master output slave input
- MISO : Master input slave output
- $\overline{SS}$: Slave select (often active low)

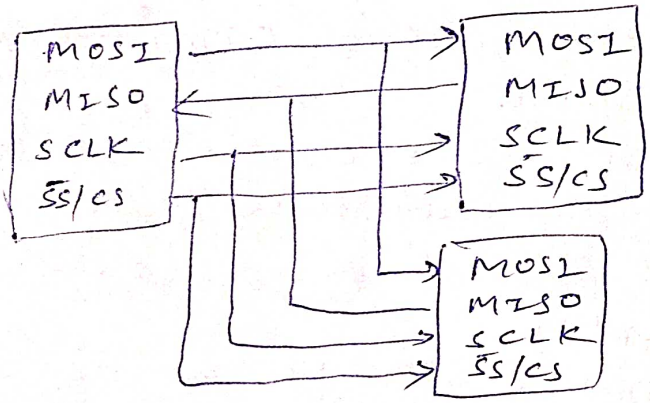


[Process In detail]

- SPI Configuration :-
- Multiple CS/$\overline{SS}$ pins may be available on the master, which allows for multiple slaves to be wired parallel.
- If only on CS/SS pin is present, multiple slaves can be wired to the master by daisy-chaining.



or



The SPI Protocal :-

1. The master outputs the clock signal [SCLK]
2. The master switches the SS pin to a low voltage state, which activates slave
3. The master sends the data one bit at a time to the slave along with MOSI line. The slaves reads the bits as they are receives [MSB-LSB]
4. If a response needed, the slave returns data one bit at a time to the master along the MISO lines. The master reads the bits as they are received [LSB-MSB]

Advantage of SPI:-

- Full duplex
- Push-pull drivers provide good signal integrity and high speed.
- Higher throughput than I²C or SMBus.
- Not limited to any maximum clock speed
- Not limited to 8-bit words.
- Slaves use the master clock and do not need any precision oscillators.

Limitations of SPI

- Typically supports only one master device
- No error-checking protocol is defined
- Require more number of pins
- No flow control, no acknowledgement

Application:-

Sensors: temperature, pressure, touchscreens, video game controller.
Control device: Audio codecs, DAC
Memory: flash, EEPROM
Real time clock

(11) Inter-integrated circuit (I2C)

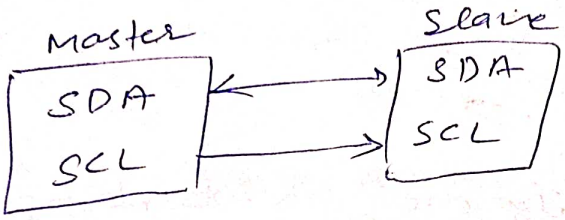
- I2C is a serial communication protocol, so data is transferred bit by bit along a single wire (the SDA line)
- Like SPI, I2C is synchronous, so the o/p of bits is synchronized to the sampling of bits by a clock signal shared between the master & the slave.
- The clock is always control by master.
- I2C combines the best features of SPI & UARTs.
- With I2C, you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single or multiple slaves.
- This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a signal LCD.

Lines used in I2C

Like I2C only uses two wires to transmit data between devices:

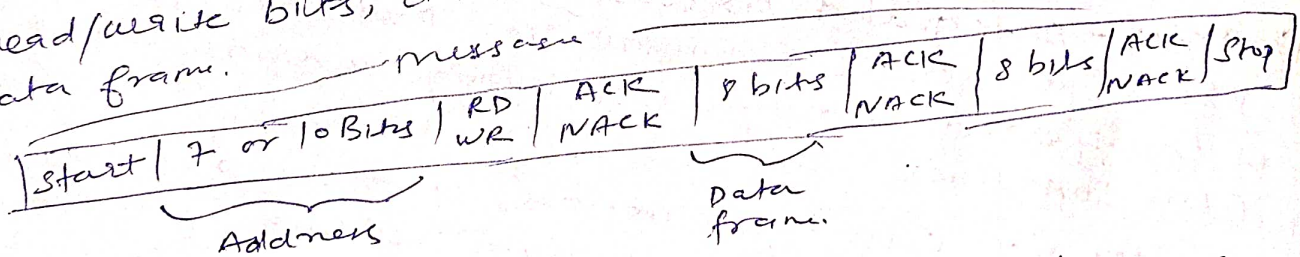
SDA (serial data) :- The line for the Master and slave to send and received data.

SCL (serial clock) :- The line that carries the clock signal



HOW I2C works?.

- On I2C, data is transferred in messages.
- Messages are broken up into frames of data
- Each message has an address frame that contain the binary address of the slave and one or more data frames that contain the data being transmitted
- The message also includes start & stop conditions, read/write bits, and ACK/NACK bits b/w each data frame.



Start condition :- SDA line high to low then SCL high to low

Stop condition :- SCL line low to high then SDA low to high

Address frame :- This contain a unique address of slave to communicate

Read/Write bit :- Sending data to the slave :- RD/WR bit is low
Requesting data from slave :- RD/WR bit is high.

ACK/NACK :- Data is successfully transferred or not

Advantage:-

- Only uses two wires
- Supports multiple masters & multiple slaves
- ACK/NACK is used to tell confirmation.
- Hardware is less complicated than UART.
- Well known & widely used.

Disadvantages:-

- Slower data transfer rate than SPI
- The size of each data frame is limited to 8 bits.
- More complicated hardware needed to implement than S.PI.

(12) CAN bus protocal:-

- Controller area network is the method of communication between electronics devices embedded in a vehicle, such as engine-management systems, active suspension, central locking, AC, air bags etc.
- The idea was initiated by Robert Bosch GmbH in 1983 to improve the quality and safety of automobiles, enhancing automobiles reliability and fuel efficiency.
- First released in 1986, also provided advancements in communication.

Unit-3

① Embedded Product development life cycle (EDLC)

EDLC is an analysis - Design- implementation based problem salving approach for embedded systems development.

- Analysis : It involves understanding what product needs to be developed
- Design ——— what approach used
- Implementation :- Actual development of the product.

② Objective of EDLC

A product is said to be profitable only if the turnover from the selling of the product is more than the overall investment expenditure.

- For this the product should be acceptable by the end user.

i) Ensure high Quality for product.

Quality in any product development is Return on investment (ROI) achieved by the product.

- The expenses for developing the product are-
  1) Initial investment
  2) Developer recruiting
  3) Training
  4) Infra structure requirement

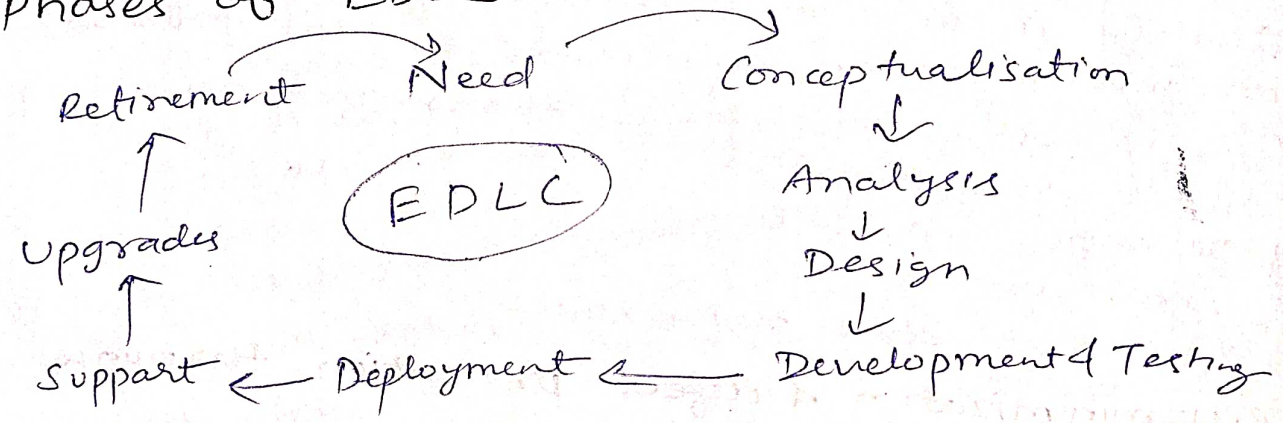
ii) Risk minimization & defect prevention through project management.

- Product development requires "loose" & "tight" project management
- Project management is essential for predictability co-ordination and risk minimization.
- Projects which are complex and requires timelines should have a skilled and dedicated project management part hence called 'highly bounded" to the project management

iii) Maximize the productivity

- Productivity is a measure of efficiency as well as Return on investment (ROI)
- One aspect of productivity covers how many resources are utilized to build product, how much investment required, how much time taken for developing the product.
- Thus productivity measurement is based on total man power efficiency.
- Productivity in terms of returns is said to be increased if the product is capable of yielding maximum returns with reduced investment.
- Saving manpower effort definitely increases productivity
- Usage of automated tools helps

② Phases of EDLC



i) Need phase:-

- The need may come from individual or from public or from company
- Need should initiated Development life cycle
- Need can be visualized in any one of the 3 needs
  1) New or custom product development
  2) Product Re-engineering
  3) Product maintenance.

ii) Conceptualization phase
- It defines the scope of concept.
- It performs cost benifit analysis and prepare project management and risk management plans & feasibility study.

Activities:- i) Feasibility study ii) cost Benifit analysis iii) ~~Proje~~ Product scope iv) planning activities

iii) Analysis phase.
The product is defined in detailed with respect to the inputs, processes outputs and interfaces at a functional level.

Activity:- Analysis & Documentation.

iv) Design phase:-
Identifies application environment and creates an overall architecture for the product such as blueprint of whole system.

v) Development & Testing
- Development means designing the particular product into hardware & software.
- Testing can be divided into independent testing of software & hardwar
  i) Unit Testing ii) integration testing iii) system testing
  iv) User acceptance testing.

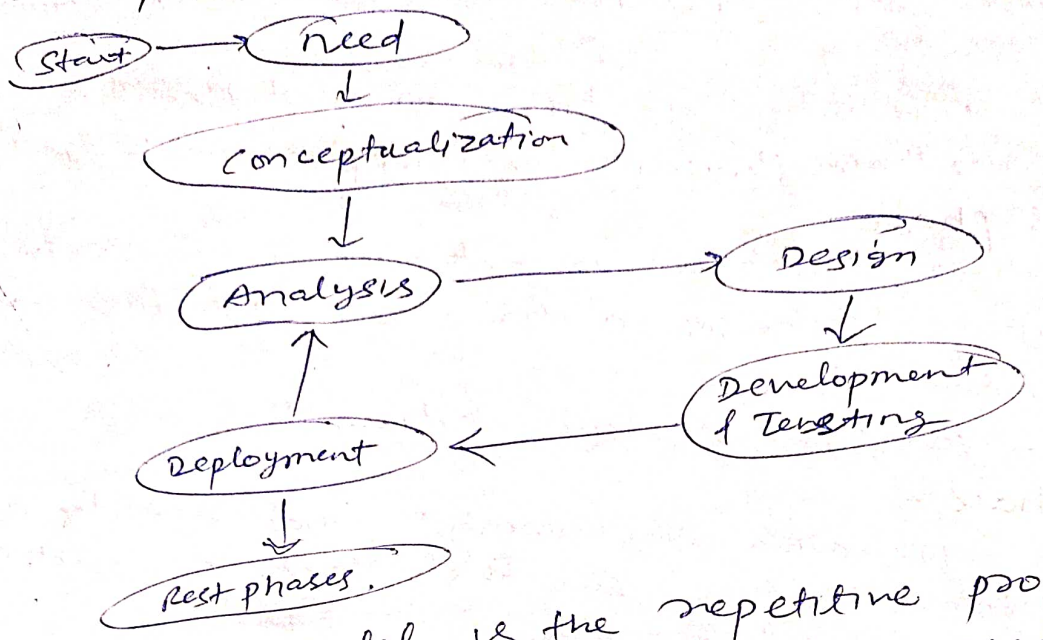
vi) Deployment phase:
- Deployment is the process of launching the first fully functional model of the product in the market
- It is also known as first customer shipping (FCS)

vii) support phase:-
- The support deals with the operational & maintenance of the product in the production environment

viii) Upgrade phase
- Deals with the development of up grades (new version) for the product.

④ Waterfall Model ✓
All phases one after another in a sequence.

⑤ Iterative / Incremental / fountain model.



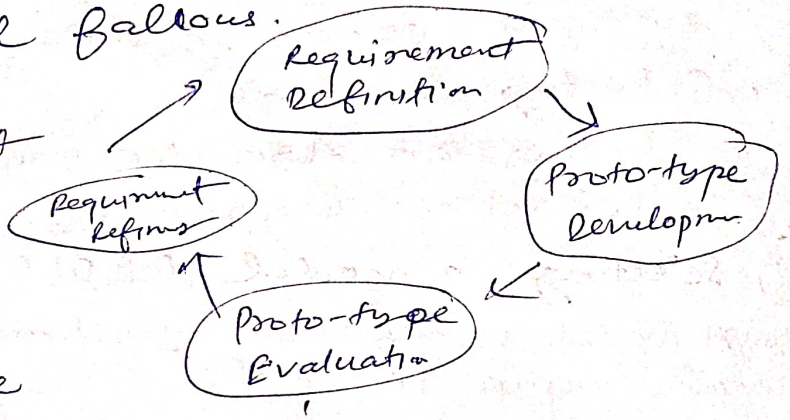
- The Iterative model is the repetitive process in which the waterfall model is repeated over and over to correct the ambiguities observed in each iteration.
- The core set of functions for each group is identified in the first cycle, it is then built deployed and release.
- This release is called first release.
- Bug fixes and modification for the first cycle is carried out in second cycle.
- Process is repeated until all functionalities are implemented meeting the requirements.

Advantage :—
- Good development cycle feedback at each function, feature implementation
- Data can be used as reference for similar product development in future.
- More responsive to changing user needs.
- Provides working product model with at least minimum feature at the first cycle.

Disadvantage:-
- Extensive review requirement each cycle.
- Impact on operations due to new releases.
- Training requirement for each new deployment at the end of each development cycle.

(6) Prototype model :-

The prototyping model follows.
- Requirement definition
- Proto-type development
- Proto-type evaluation
- Requirement refining



- It is similar to iterative model and the product is developed in multiple cycle
- The only differences is that, prototyping model produces a refined prototype of the product at the end of each cycle instead of functionality addition in each cycle as performed by IT model
- There won't be any commercial deployment of the prototype of the product at each cycle
- The short comings of the prototype-model after each cycle are evaluated and it is fixed in the next cycle. (bug)
- On finishing the prototype, it is send to the customer for evaluation.
- The customer evaluate it and give feedback

(7) Spiral model :- (SE) -

(8) Issues in Hardware software co-design

The hardware software co-design is a problem statement and when we try to solve this problem statement in real life we may come across multiple issues in the design.

Some of the fundamental issues in hardware software co-design are -

- Selecting a model
- Selecting an Architecture
- Selecting a language.

1) Selecting a model [EDLC)

- A model is a formal system consisting of objects and composition rules
- In hardware software co-design, models are used for capturing and describing the system characteristics
- It is hard to make a decision which model should be followed in a particular system.
- Most often designers switches b/w various model.

2) Selecting the architecture.

- The architecture specifies how a system is going to implement in terms of the number and types of different components and the interconnection among them.
- Some of commonly used architectures fall into Application Specific architecture class (like controller while other fall into either general purpose architecture class (CISC, RISC etc) or parallel processing class (SIMD, MIMD etc).

3) Selecting a language:

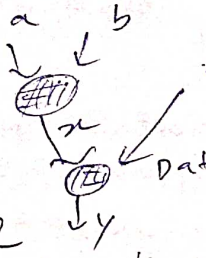
- A programming language captures a computational model and map it into architecture.
- A model can be capture (Implement) using multiple programming language like C, C++, C#, Java etc.
- For hardware implementation :- VHDL, System C, Verilog etc.
- A single language capture whole model :- C++. (The OOPs)

(8) Data flow graph (DFG)

- DFG model translate the data processing requirement into a data flow graph.
- DFG model is a data driven model in which the program execution is determined by data.
- This model emphasizes on the data and operations on the data which transforms the input data to output data.
- DFG is a visual model in which
  - operation on the data is represented using a block and flow is represented using arrows.
  - An inward arrow to the process represents input data and an outward arrow from the process represent output data.
- Embedded applications which are computation intensive and data driven are modeled using the DFG model.
- In a DFG model, a data path is the data flow path from I/P to O/P.
- DFG model is said to be acyclic DFG
- DFG model translates the program as a single sequential process execution.

$x = a+b$

$y = x-c$



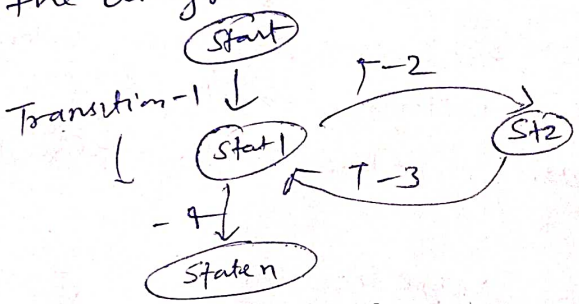
(9) State machine Model

- The state machine model is used for modelling reactive or event-driven embedded systems whose processing behaviour are dependent on state transition.
- Embedded systems used in the control and industrial applications are typical examples for event driven systems.
- The state machine model describes the system behaviour with 'States', 'Events', 'Actions' & 'Transitions'.
- State is a representation of a current situation.
- An event is an input to the state. The event acts as stimuli for state transition.

- Transition is the movement from one state to another
- Actor is an activity to be performed by the state machine.

Finite state machine:- (FSM)

- A finite state machine (FSM) model is a model of discreate behaviors which consists of a finite number of states, transitions between those states and actions.
- Some integrated development environments (IDE) allow a designer to draw a FSM diagram on screen, and then the IDE automatically translate the diagram into a software, or ASIC. implementation



(10) Sequential Program Model.

- In the sequential programming model, the functions or processing requirements are executed in sequence.
- It is same as the conventional procedural programming
- Here the program instructions are iterated and executed conditionally and the data gets transformed through a series of operations.
- FSM are good choice, Another important tools used for modelling sequential program is flow-charts.
- FSM approach represents the states, events, transition and actions,
- Whereas the flow chart models the execution flow

Ex- Sequential program likh do:— koi sa bhi -

(12) Concurrent Model

- The concurrent or communicating process model concurrently executing task processes.
- Sequential execution leads to a single sequential execution of task and thereby leads to poor processor utilization.
- When the task involves I/O waiting, sleeping for specified duration etc.
- If the task is split into multiple subtasks, it is possible to tackle the CPU usage effectively.
- When the subtask under execution goes to a wait or sleep mode, by switching the task execution.
- However, concurrent processing model requires additional overheads in task scheduling, task synchronization & communication.
- It is commonly used for Real time system,

(13) Object Oriented Model

- The object-oriented life cycle model considers 'objects' as the basic of the embedded product development.
- Apart from enhancing the system performance, object-oriented programming offers some advantages such as:
  - → Since it is data-focused and easy to work with problems domains.
  - → It uses encapsulation and data hiding process that allows a developer to build tamper-proof systems.
  - → It enables software modularity, making it easier to manage and maintain complex software.
  - → It allows developers to create new modules using existing model saving time and development cost of organizations.

Objective of Object oriented model

- Object - oriented Analysis.
- Object - oriented Design
- Object - oriented Implementation: