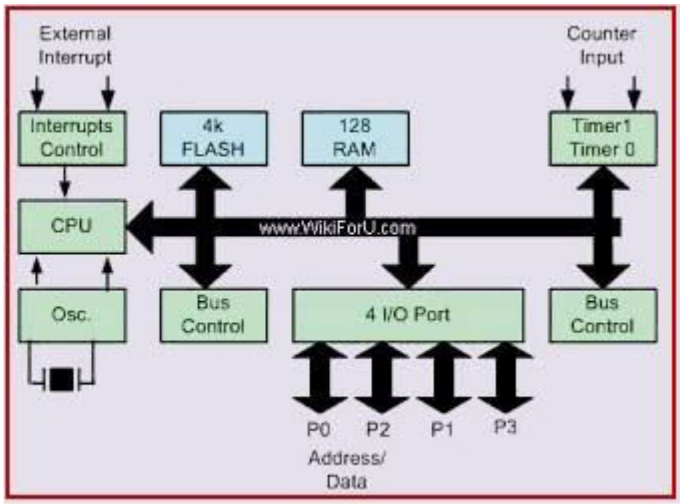
Theory Questions(Units-1,2)

1)Draw the block diagram of 8051 and explain

The 8051 Microcontroller is one of the basic types of microcontrollers, designed by Intel in 1980s. This microcontroller was based on Harvard Architecture



Central Processor Unit (CPU)

CPU is the brain of any processing device of the microcontroller. It monitors and controls all operations that are performed on the Microcontroller units.

Memory

Microcontroller requires a program which is a collection of instructions. This program tells microcontroller to do specific tasks.

BUS

Bus is a collection of wires which work as a communication channel for transfer of Data.

- Address Bus
- Data Bus

Input/Output Port

8051 has 4 input, output ports to connect it to the other peripherals

Timers/Counters

8051 microcontroller has two 16 bit timers and counters. The timers are used for measurement of intervals to determine the pulse width of pulses.

....

3)List out modes of operations of timers in 8051 and explain

8051 contains to 16-bit timers

- T0 (timer 0)
- T1 (timer 1)

Mode	**Description**
Mode 0	8 or 13 bit timer
Mode 1	16 bit timer
Mode 2	8 bit auto reload
Mode 3	split timer mode

Mode 0 (13-bit timer mode)

Mode 0 is a 13-bit timer mode for which 8-bit of THx and 5-bit of TLx (as Prescaler) are used. It is mostly used for interfacing possible with old MCS-48 family microcontrollers.

Mode1 (16-bit timer mode)

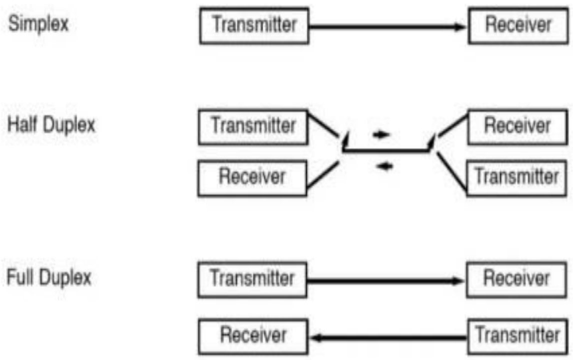
Mode 1 is a 16-bit timer mode used to generate a delay, it uses 8-bit of THx and 8-bit of TLx to form a total 16-bit register.

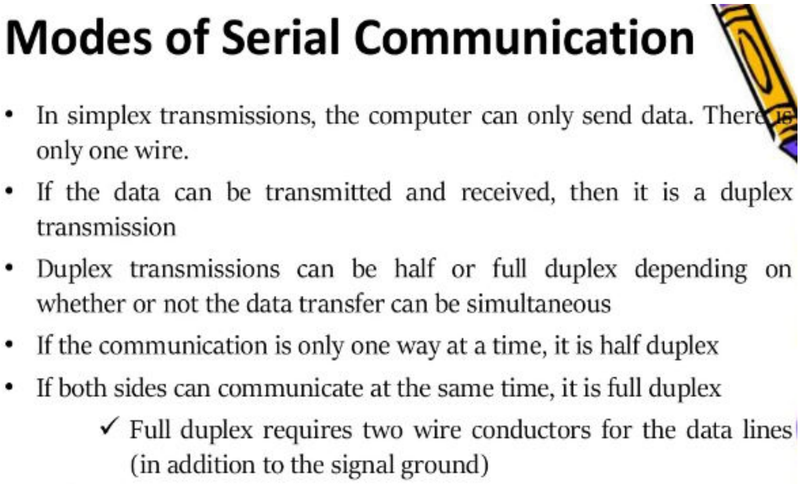
Mode2 (8-bit auto-reload timer mode)

Mode 2 is an 8-bit auto-reload timer mode. In this mode, we have to load the THx-8 bit value only. when the Timer gets started, the THx value gets automatically loaded into the TLx and TLx starts counting from that value.

....

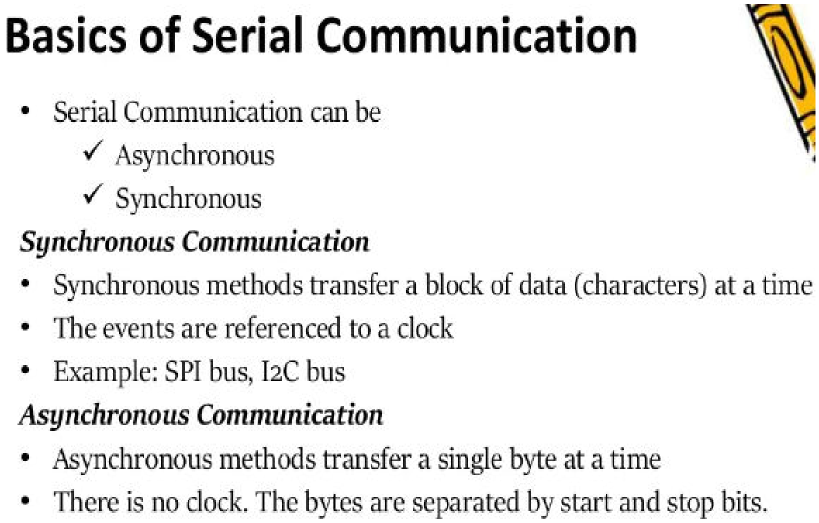
4)Explain serial communication modes of 8051





Modes of Serial Communication

- In simplex transmissions, the computer can only send data. There is only one wire.
- If the data can be transmitted and received, then it is a duplex transmission
- Duplex transmissions can be half or full duplex depending on whether or not the data transfer can be simultaneous
- If the communication is only one way at a time, it is half duplex
- If both sides can communicate at the same time, it is full duplex
  - ✓ Full duplex requires two wire conductors for the data lines (in addition to the signal ground)



Basics of Serial Communication

- Serial Communication can be
  - ✓ Asynchronous
  - ✓ Synchronous

Synchronous Communication

- Synchronous methods transfer a block of data (characters) at a time
- The events are referenced to a clock
- Example: SPI bus, I2C bus

Asynchronous Communication

- Asynchronous methods transfer a single byte at a time
- There is no clock. The bytes are separated by start and stop bits.

....

5)Explain about interrupts in 8051

Interrupts are the events that temporarily suspend the main program, pass the control to the external sources and execute their task. It then passes the control to the main program where it had left off.

8051 has 5 interrupt signals, i.e. INT0, TFO, INT1, TF1, RI/TI. Each interrupt can be enabled or disabled by setting bits of the IE register and the whole interrupt system can be disabled by clearing the EA bit of the same register.

IE (Interrupt Enable) Register

This register is responsible for enabling and disabling the interrupt. EA register is set to one for enabling interrupts and set to 0 for disabling the interrupts.

IP (Interrupt Priority) Register

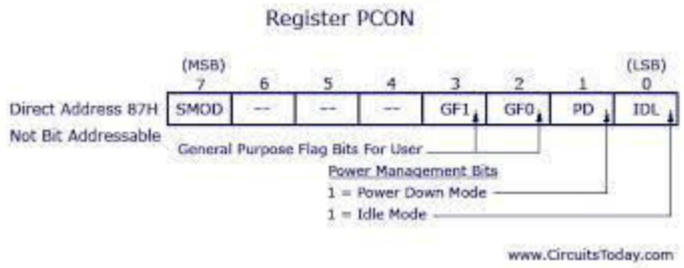
If two interrupts of different priority levels are received simultaneously, the request of higher priority level is served.

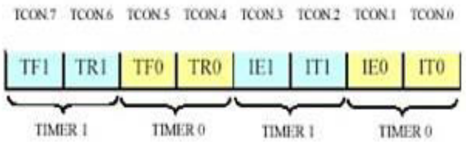
TCON Register

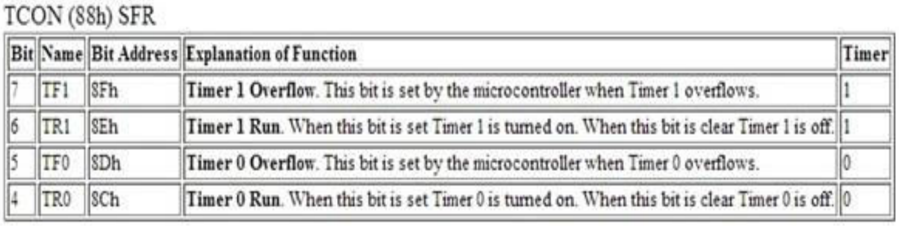
TCON register specifies the type of external interrupt to the microcontroller.

..

6)Draw the formats of TCON,SCON,PCON and explain

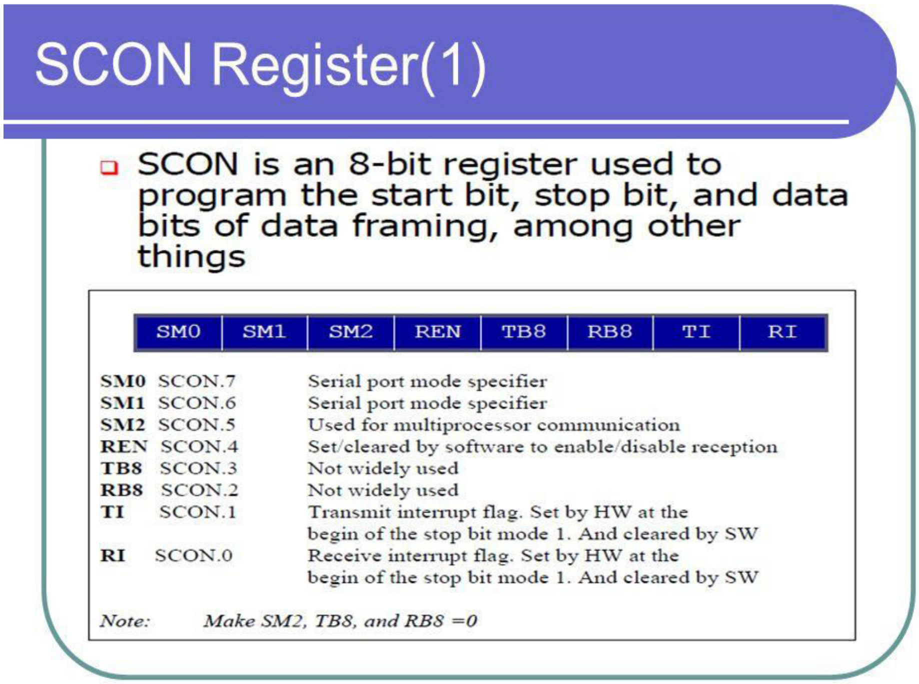





TCON (88h) SFR

Bit	Name	Bit Address	Explanation of Function	Timer
7	TF1	8Fh	**Timer 1 Overflow**. This bit is set by the microcontroller when Timer 1 overflows.	1
6	TR1	8Eh	**Timer 1 Run**. When this bit is set Timer 1 is turned on. When this bit is clear Timer 1 is off.	1
5	TF0	8Dh	**Timer 0 Overflow**. This bit is set by the microcontroller when Timer 0 overflows.	0
4	TR0	8Ch	**Timer 0 Run**. When this bit is set Timer 0 is turned on. When this bit is clear Timer 0 is off.	0

lower 4 bytes are IE0/IE1 & IT0/IT1



SCON Register(1)

- SCON is an 8-bit register used to program the start bit, stop bit, and data bits of data framing, among other things

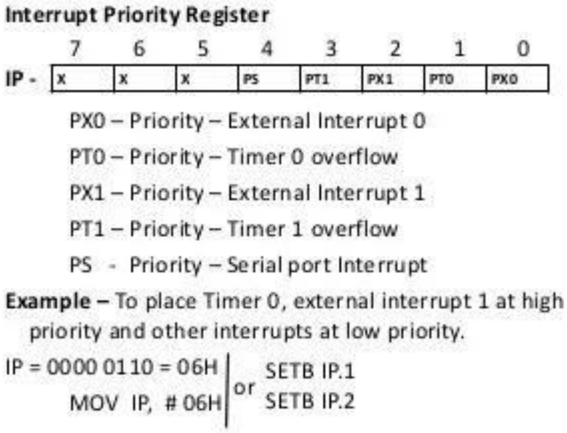
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial port mode specifier
SM1	SCON.6	Serial port mode specifier
SM2	SCON.5	Used for multiprocessor communication
REN	SCON.4	Set/cleared by software to enable/disable reception
TB8	SCON.3	Not widely used
RB8	SCON.2	Not widely used
TI	SCON.1	Transmit interrupt flag. Set by HW at the begin of the stop bit mode 1. And cleared by SW
RI	SCON.0	Receive interrupt flag. Set by HW at the begin of the stop bit mode 1. And cleared by SW

Note: Make SM2, TB8, and RB8 =0

....

7)Draw the formats of IP,IE ,TMOD and explain



Interrupt Priority Register

	7	6	5	4	3	2	1	0
IP -	X	X	X	PS	PT1	PX1	PT0	PX0

PX0 – Priority – External Interrupt 0

PT0 – Priority – Timer 0 overflow

PX1 – Priority – External Interrupt 1

PT1 – Priority – Timer 1 overflow

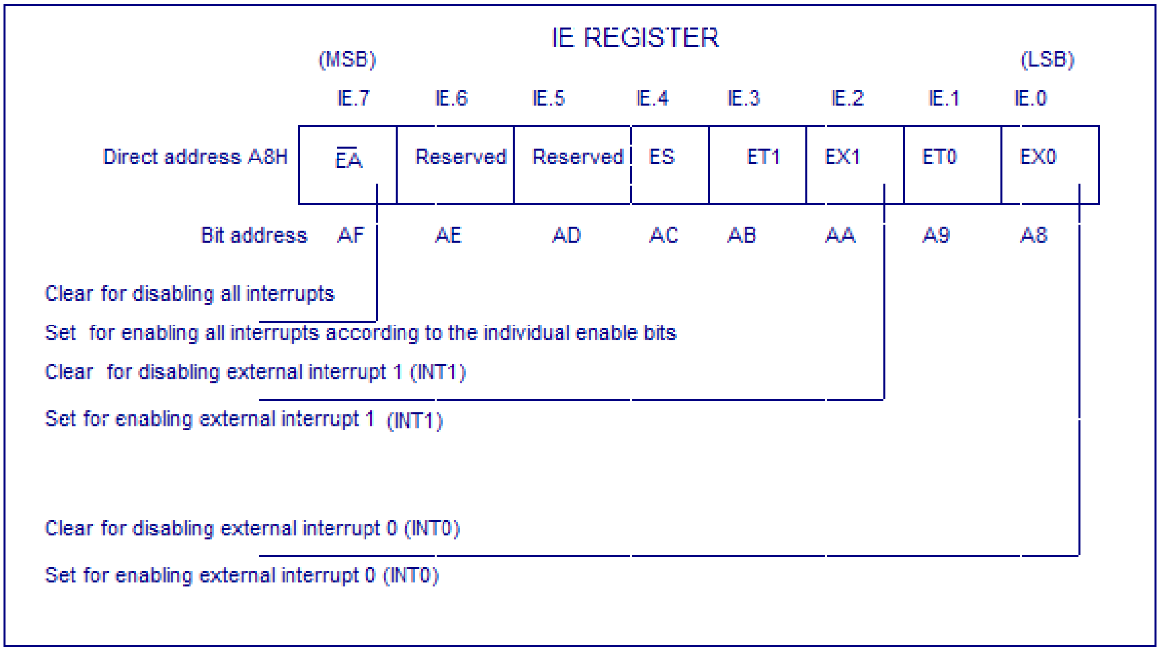
PS - Priority – Serial port Interrupt

Example – To place Timer 0, external interrupt 1 at high priority and other interrupts at low priority.

IP = 0000 0110 = 06H
MOV IP, # 06H

or

SETB IP.1
SETB IP.2



IE REGISTER

	(MSB)							(LSB)
	IE.7	IE.6	IE.5	IE.4	IE.3	IE.2	IE.1	IE.0
Direct address A8H	$\overline{\text{EA}}$	Reserved	Reserved	ES	ET1	EX1	ET0	EX0
Bit address	AF	AE	AD	AC	AB	AA	A9	A8

Clear for disabling all interrupts

Set for enabling all interrupts according to the individual enable bits

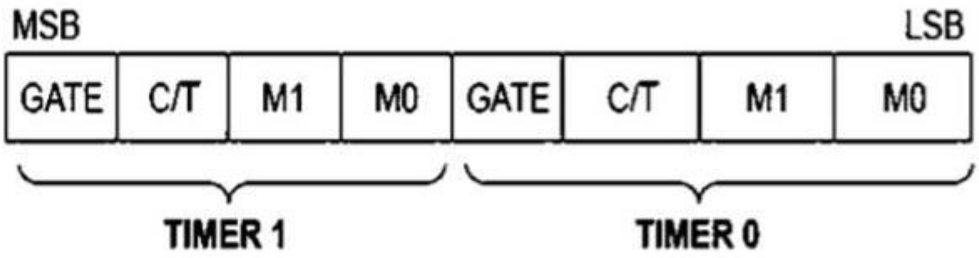
Clear for disabling external interrupt 1 (INT1)

Set for enabling external interrupt 1 (INT1)

Clear for disabling external interrupt 0 (INT0)

Set for enabling external interrupt 0 (INT0)

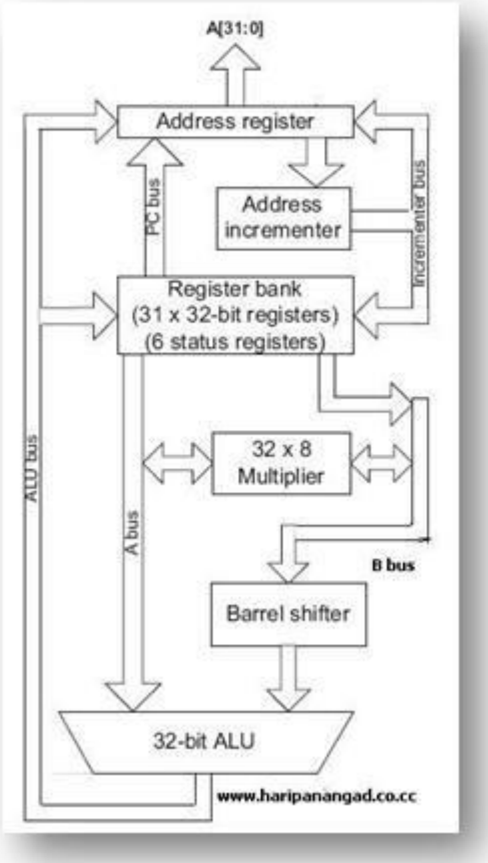
www.cselectricalandelectronics.com



MSB							LSB
GATE	C/T	M1	M0	GATE	C/T	M1	M0
TIMER 1				TIMER 0			

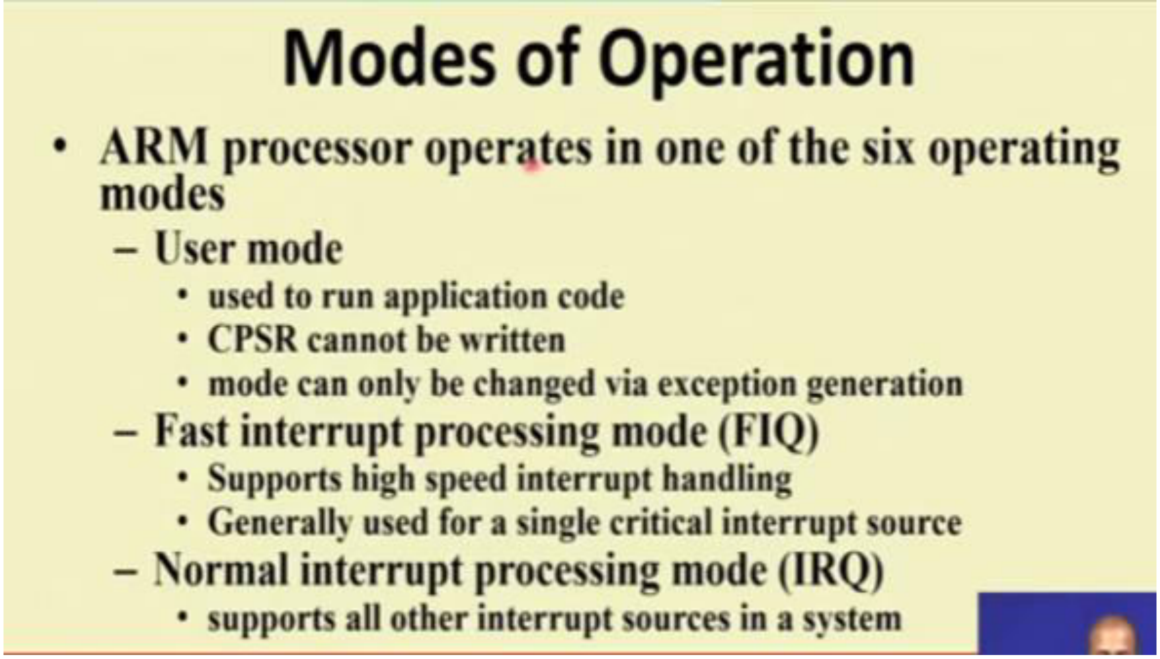
....

8) Block diagram of ARM7



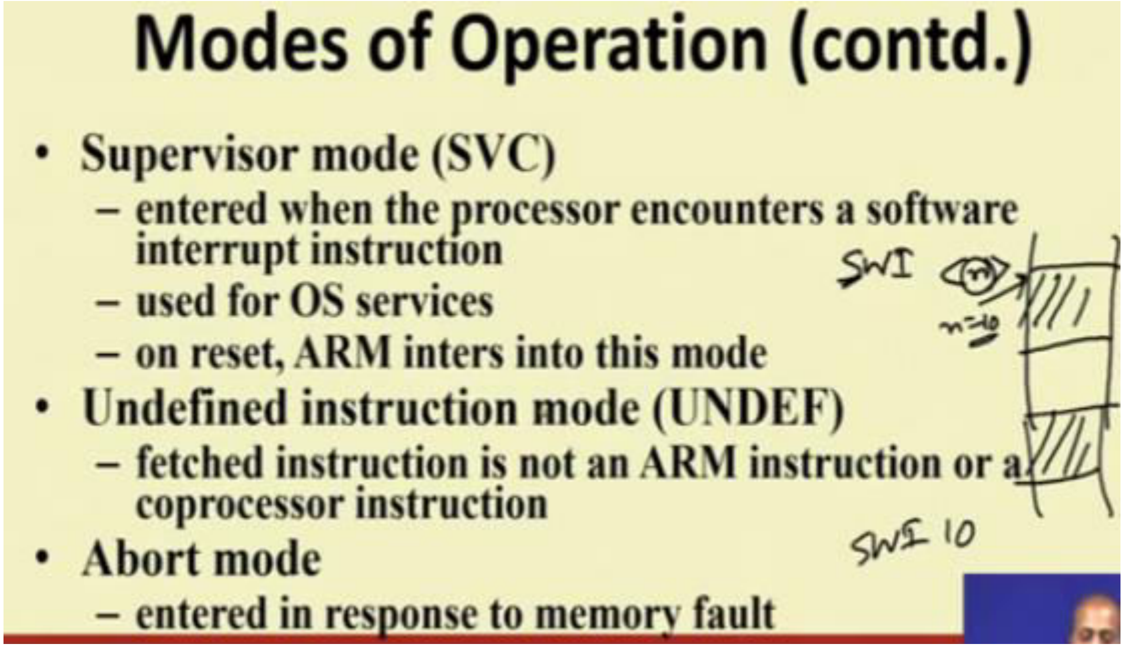
....

9)Explain the modes of operation of ARM7



Modes of Operation

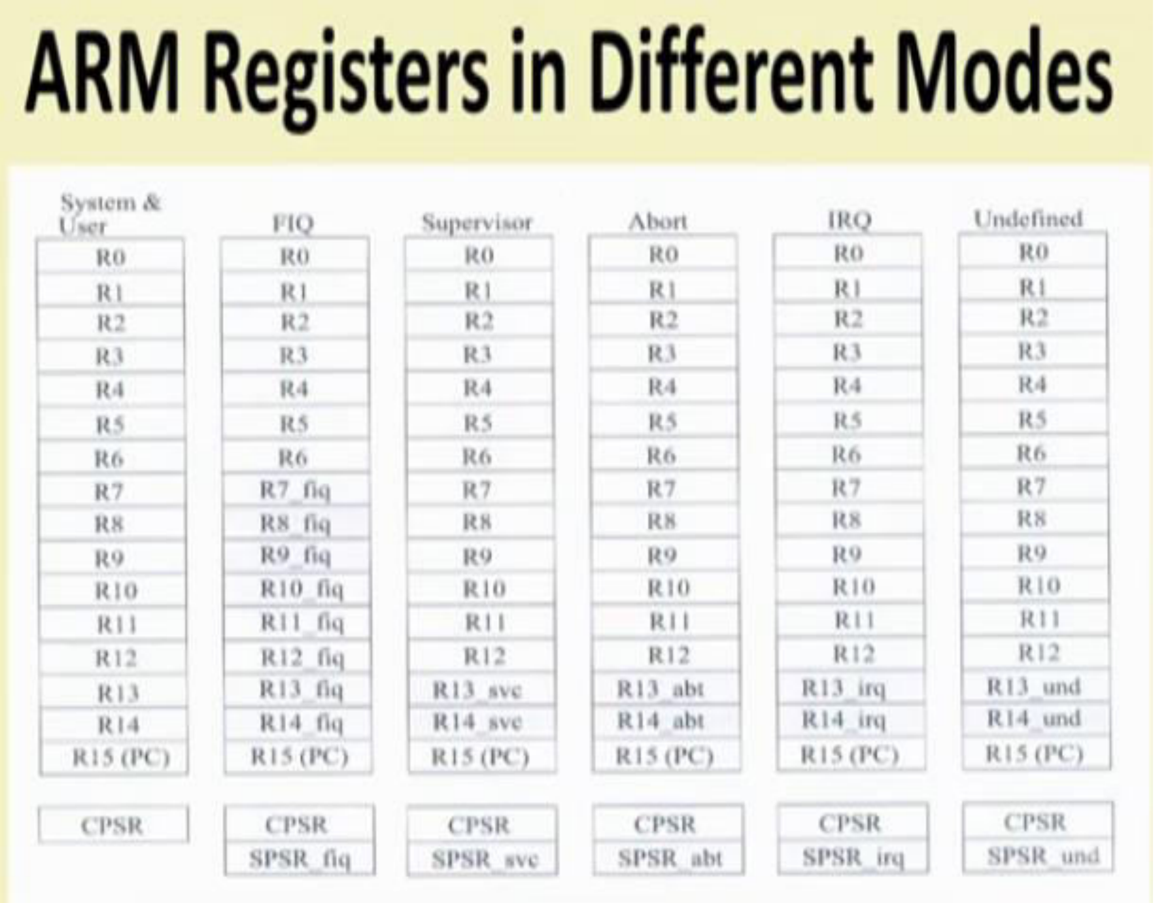
- ARM processor operates in one of the six operating modes
  - User mode
    - used to run application code
    - CPSR cannot be written
    - mode can only be changed via exception generation
  - Fast interrupt processing mode (FIQ)
    - Supports high speed interrupt handling
    - Generally used for a single critical interrupt source
  - Normal interrupt processing mode (IRQ)
    - supports all other interrupt sources in a system



Modes of Operation (contd.)

- Supervisor mode (SVC)
  - entered when the processor encounters a software interrupt instruction
  - used for OS services
  - on reset, ARM inters into this mode
- Undefined instruction mode (UNDEF)
  - fetched instruction is not an ARM instruction or a coprocessor instruction
- Abort mode
  - entered in response to memory fault

10)Explain Register organization on ARM7



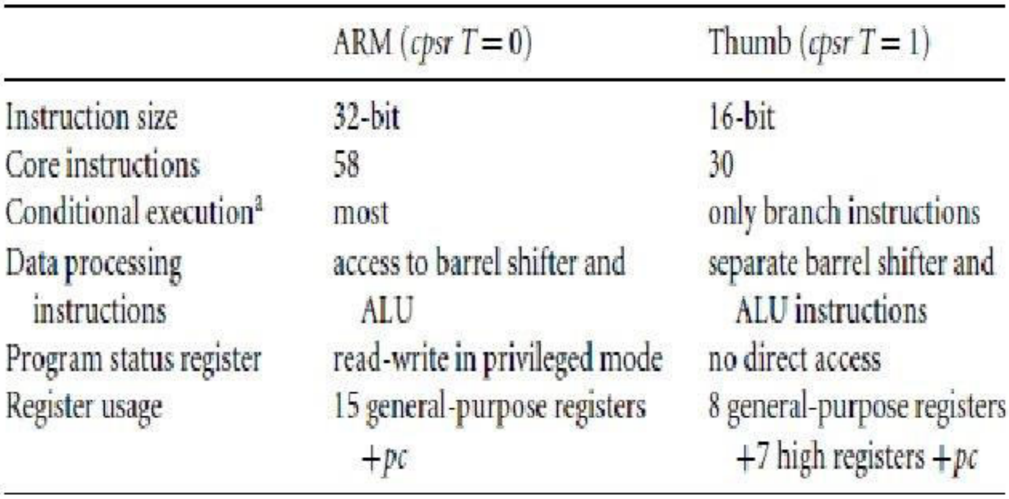
ARM Registers in Different Modes

System & User	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7_fiq	R7	R7	R7	R7
R8	R8_fiq	R8	R8	R8	R8
R9	R9_fiq	R9	R9	R9	R9
R10	R10_fiq	R10	R10	R10	R10
R11	R11_fiq	R11	R11	R11	R11
R12	R12_fiq	R12	R12	R12	R12
R13	R13_fiq	R13_svc	R13_abt	R13_irq	R13_und
R14	R14_fiq	R14_svc	R14_abt	R14_irq	R14_und
R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

....

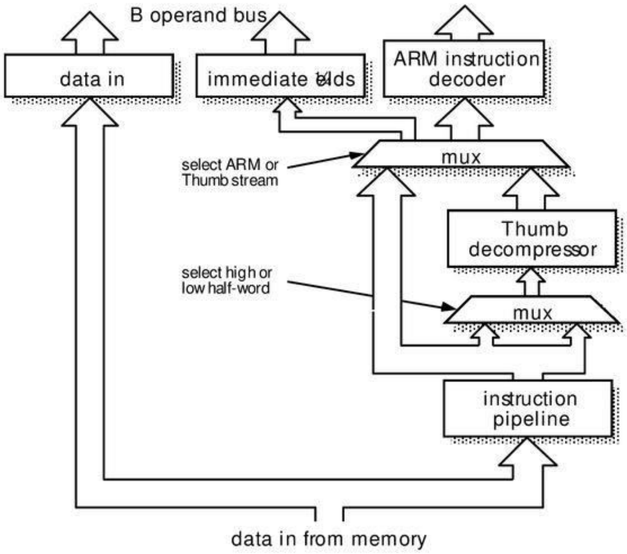
11)Write the differences between ARM and Thumb instruction and also draw Thumb to Arm decompressor

ARM and Thumb instruction set features



	ARM (*cpsr* $T = 0$)	Thumb (*cpsr* $T = 1$)
Instruction size	32-bit	16-bit
Core instructions	58	30
Conditional execution[a]	most	only branch instructions
Data processing instructions	access to barrel shifter and ALU	separate barrel shifter and ALU instructions
Program status register	read-write in privileged mode	no direct access
Register usage	15 general-purpose registers $+pc$	8 general-purpose registers $+7$ high registers $+pc$

The Thumb instruction decompressor organization



©2000 Addison Wesley

13. **Explain about Characteristics of good RTOS**

The characteristics of good RTOS are:

Multitasking Capabilities: An RT application is divided into multiple tasks. This separation keeps CPU busy

Short Interrupt latency: interrupt latency is the time required for transferring execution to the interrupt handler.

Fast Context switching: The time between the OS recognizing that awaited event and the beginning of the waiting task is called context switching time. This switching time must be small.

Control of memory management: A very less memory provided for Embedded systems ,so memory should be utilized by tasks efficiently.

Proper Scheduling: OS must provide facility to schedule properly time constrained tasks.

Rich set of inter task communication mechanisms: message queues, shared memory , semaphores ,event flags

14. **Explain how semaphores used in Shared data problems with a µc Os program example**

- Shared data problem can occur when another higher priority task finishes an operation and modifies the content befor the completion of previous task operations.
- Semaphores is one such tool to deal with these types of problems:

Example Programs:

```
#include "vxworks.h"
#include "tasklib.h"
#include "semlib.h"
#include "stdio.h"
void taskOne(void) {
      For(int i=0; i<ITER; i++) {
            semTake(semBinary, WAIT_FOREVER);
            printf("I am Task-1 and global = %d....\n", ++global);
            semGive(semBinary);
      }
}
#define ITER 10
int global = 0;
SEM_ID semBinary;
```

```
void taskTwo() {
      semGive(semBinary);
      for(int i=0; i<ITER; i++) {
            semTake(semBinary, WAIT_FOREVER);
            printf("I am Task-2 and global = %d", --global);
            semGive(semBinary);
      }
}
void main(void) {
      int taskIdone, taskIdtwo;
      semBinary = semBCreate(SEM_Q_FIFO, SEM_FULL); } Semaphore Creation
      semTake(semBinary, WAIT_FOREVER);
      taskIdone = taskSpawn("t1", 90, 0x100, 2000, (FUNCPTR)taskOne, 0, 0, 0, 0, 0, 0, 0, 0, 0);
      taskIdtwo = taskSpawn("t2", 90, 0x100, 2000, (FUNCPTR)taskTwo, 0, 0, 0, 0, 0, 0, 0, 0, 0);
}
```

- semBinary - semaphore accessed by both task1 and task 2.
- Taking semaphore is the most targeted way to protect data, because it affects only those tasks that need to take the same semaphore.

15. **Explain about rules to be followed in RTOS by interrupt procedures.**

Interrupt routines in RTOS must follow
two rules:

Rule1: An interrupt routine must not call any RTOS function ,that might block the interrupt

- Interrupt routines must not get semaphores ,read from queues, mail boxes that might weight for events.
- If an interrupt routine calls an RTOS function and gets blocked , then in addition to the interrupt routine , tasks which calls ISR also blocked (though task having highest priority).

Rule 2: An interrupt routine may not call any RTOS function that cause the RTOS to switch to the tasks unless the RTOS knows that an interrupt routine , and not task is executing.

- It means that interrupt routine may not write to mailboxes or queues on which tasks may be waiting , set events , for releasing semaphores , Unless RTOS knows that interrupt routine that is doing these.
- If interrupt routine breaks this rule , the RTOS might switch control away and run different tasks and interrupt routine may not completed at right time.

16. **Describe the following and also pitfalls**

a) mailboxes b) pipes c) message queues

Ans :

Mail boxes:

Mailboxes are like queues

RTOS has functions to create ,to write ,and read from mailboxes.

Pipes:

Pipes are like queues.

RTOS can create them ,write to them ,read from them , and so on.

Msg Queue:

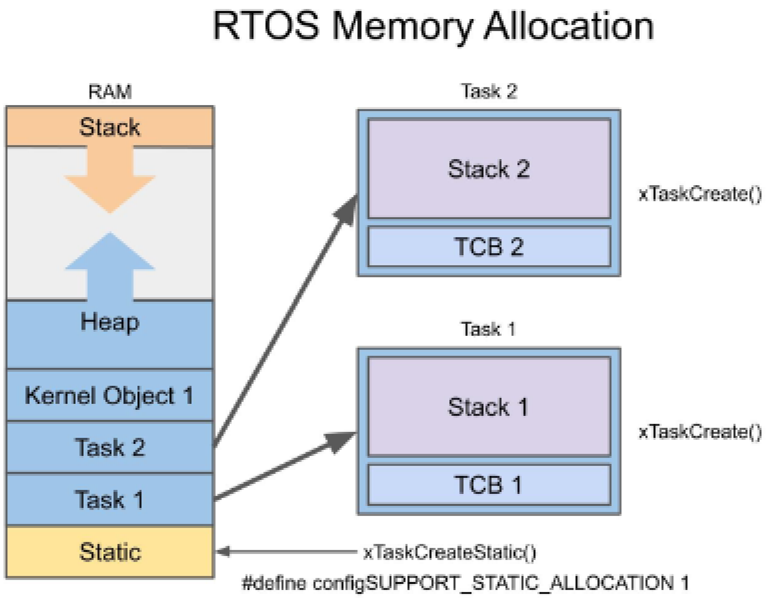
- Tasks must communicate with each other to coordinate their activities or share data.
- using shared data and semaphores each task can share data to other.
- Each task can also communicate with other using, Message queues

Pitfalls:

Although the above 3 mechanisms simplify sharing data they can also introduce several bugs.

- User should ensure that tasks write to the correct mailboxes otherwise there might be errors.
- If a task writes an integer to a queue and the second task is expecting a character this will cause errors. Sometimes the compiler might not find the bug.
- Running out of space when a task needs to write data is another problem.

17. **Describe memory management in RTOS**



Static memory is used for storing global variables and variables designated as “static” in code (they persist between function calls).

The stack is used for automatic allocation of local variables. Stack memory is organized as a last-in-first-out (LIFO) system

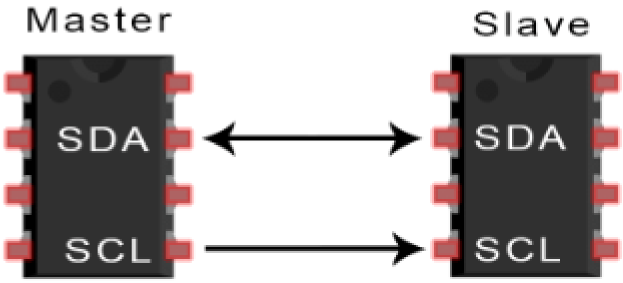
One part of that allocated memory is the Task Control Block (TCB), which is used to store information about the task, such as its priority and local stack pointer. The other section is reserved as a local stack that operates just like the global stack (but on a smaller scale for just that task).

18. Describe the I2C bus protocol

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 output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. The clock signal is always controlled by the master.

I2C combines the best features of SPI and 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.



SDA (Serial Data) – The line for the master and slave to send and receive data.

SCL (Serial Clock) – The line that carries the clock signal.

19. Explain CAN protocol and applications

CAN represents the Controller Area Network protocol.

The CAN is a message-based protocol

Node identification in the CAN network is not required, so inserting or removing it from the network is very easy.

The CAN protocol is a set of rules for transmitting and receiving messages in a network of electronic devices. It defines how data is transferred from one device to another in a network.

Applications of CAN Protocol

The CAN protocol was initially developed to fix the communication issue in the vehicles. Nonetheless, it would also be used in many other fields due to its functionality.

The specification of the CAN protocol is as follows:

- Automation Building
- Escalators and Elevator
- Mechanical Control and Industrial Automation
- For navigation and aviator Electronic equipment

20. Define IoT and explain characteristics of IoT

A:

The Internet of Things (IoT) is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment.

Characteristics of IOT:

Efficient: It has to serve efficiently the requirements of the application for which they have deployed.

Scalable: They have to scalable because in IOT , there are large number of things , they are millions of things , must be expandable. (trillions of things) . Even we increase the nodes , performance of the system should not be degraded.

Unambiguous naming and addressing: we need a different mechanism for naming and addressing of different nodes.

Abundance of sleeping nodes: Most of the devices are battery operated so power saving by operating the devices in sleeping mode , when ever they not operating is required.

Mobile devices: mobility of the devices must be possible .

Dynamic and self-Adapting: IOT devices and systems may have the capability to dynamically adapt with the changing contexts and take actions based on their operating conditions

21. Define Sensor and explain operation thermocouple and ultrasonic sensor

Sensor : A device which detects or measures a physical property and records, indicates, or otherwise responds to it

A Thermocouple is a sensor used to measure temperature. Thermocouples consist of two wire legs made from different metals. The wires legs are welded together at one end, creating a junction. This junction is where the temperature is measured. When the junction experiences a change in temperature, a voltage is created.

Ultrasonic sensor, this basically detects that how far that obstacle is. there are two eyes kind of things. Ultrasonic sensors may send ultrasound waves. These ultrasound waves are sent and then, that sound wave is going to get reflected back.
We know velocity and then, depending on how much time has elapsed from the point sound wave was sensed and the reflection is received back, based on that the distance is calculated.

22. Define Sensor and explain operation PIR and LDR

Sensor : A device which detects or measures a physical property and records, indicates, or otherwise responds to it

Passive infrared (PIR) sensors **use a pair of pyroelectric sensors to detect heat energy in the surrounding environment**. These two sensors sit beside each other, and when the signal differential between the two sensors changes (if a person enters the room, for example), the sensor will engage.

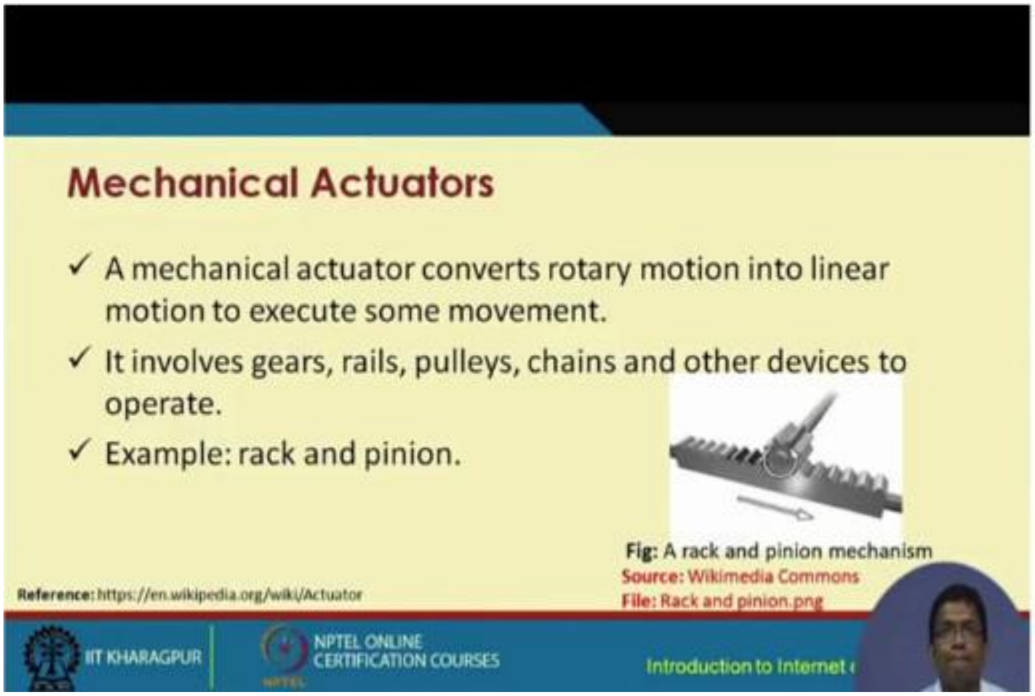
LDR's are **light dependent devices whose resistance is decreased when light falls on them and that is increased in the dark**.

The working principle of an **LDR** is photoconductivity, which is nothing but an optical phenomenon. When the light is absorbed by the material then the conductivity of the material enhances.

23. Define actuator and explain about Solenoid , Mechanical actuator

An **actuator** is a component of a machine that is responsible for moving and controlling a mechanism or system, for example by opening a valve.

Solenoid can be used for controlling the flow of liquid. Valve involved, used for controlling the flow liquid.
Actuator is control system requires a control signal , to act on the environment.



Mechanical Actuators

- ✓ A mechanical actuator converts rotary motion into linear motion to execute some movement.
- ✓ It involves gears, rails, pulleys, chains and other devices to operate.
- ✓ Example: rack and pinion.

Fig: A rack and pinion mechanism
Source: Wikimedia Commons
File: Rack and pinion.png

Reference: https://en.wikipedia.org/wiki/Actuator

24. Define actuator and explain about Hydraulic , Mechanical actuator

An **actuator** is a component of a machine that is responsible for moving and controlling a mechanism or system, for example by opening a valve.

✓ A hydraulic actuator consists of a cylinder or fluid motor that uses hydraulic power to facilitate mechanical operation.

✓ The mechanical motion is converted to linear, rotary or oscillatory motion.

✓ Since liquids are nearly impossible to compress, a hydraulic actuator exerts considerable force.

✓ The actuator's limited acceleration restricts its usage.

25. **Define actuator and explain about temperature and magnetic actuator**

An **actuator** is a component of a machine that is responsible for moving and controlling a mechanism or system, for example by opening a valve.

Magnetic Actuator:

These can be actuated by applying thermal or magnetic energy.

✓ They tend to be compact, lightweight, economical and with high power density.

✓ These actuators use shape memory materials (SMMs), such as shape memory alloys (SMAs) or magnetic shape-memory alloys (MSMAs).

✓ Some popular manufacturers of these devices are Finnish Modti Inc.

and American Dynalloy.

Temperature Actuator:

A temperature **actuator** is any device that turns equipment on and off in response to temperature changes. They may use a variety of methods to measure temperature changes, including metals, chemicals or gases. Devices to control temperature can vary from a simple home thermostat for controlling heaters or air conditioners, to complex systems that control chemical reactions in industrial plants.

26. Describe different communication models used in IoT

Types of Communication Model :

1. Request & Response Model –

This model follows a client-server architecture.

- The **client**, when required, requests the information from the server. This request is usually in the encoded format.
- This model is stateless since the data between the requests is not retained and each request is independently handled.

2. Publisher-Subscriber Model –

This model comprises three entities: Publishers, Brokers, and Consumers.

- **Publishers** are the source of data. It sends the data to the topic which are managed by the broker. They are not aware of consumers.
- **Consumers** subscribe to the topics which are managed by the broker.
- Hence, **Brokers** responsibility is to accept data from publishers and send it to the appropriate consumers.

3. Push-Pull Model –

The push-pull model constitutes data publishers, data consumers, and data queues.

- **Publishers** and **Consumers** are not aware of each other.
- Publishers publish the message/data and push it into the queue. The consumers, present on the other side, pull the data out of the queue. Thus, the queue acts as the buffer for the message when the difference occurs in the rate of push or pull of data on the side of a publisher and consumer.

4. Exclusive Pair –

- **Exclusive Pair** is the bi-directional model, including full-duplex communication among client and server. The connection is constant and remains open till the client sends a request to close the connection.
- The **Server** has the record of all the connections which has been opened.
- This is a state-full connection model and the server is aware of all open connections.

27. Explain any 4 application layer protocols of IoT

1. **HTTP:** Hypertext Transfer protocol is the application layer protocol ,forms the foundation of the World Wide Web(WWW)

Enables communication between client and servers.

The protocol follows a request response model where a client sends requests to a server using HTTP command

HTTP includes commands such as GET,PUT,POST, DELETE,HEAD

2. **CoAP:** Constrained Application Protocol is an application layer protocol for machine to machine application , meant for constrained environments with constrained devices and constrained networks.(used where resource constraints are more ,like power, bandwidth etc)

   Like HTTP,COAP is a web transfer protocol and uses request response model, how ever it runs on top of UDP instead of TCP.

3. **MQTT:** Message Queue Telemetry Transport (MQTT) is a messaging protocol based on publish –subscribe model.

   MQTT   uses a client –server architecture where a client(an IOT device) connects to the server (called MQTT broker) and publishes messages to topics on the server .

   The broker forwards the messages to the clients subscribed to topics.

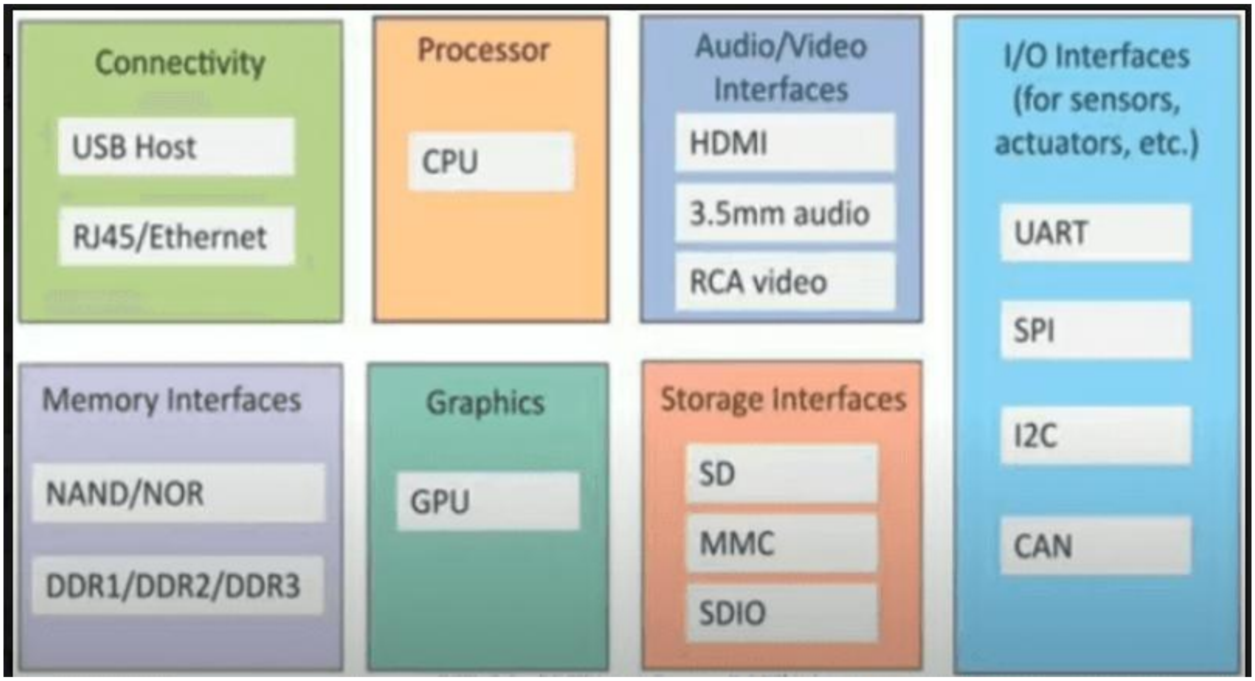
4. **AMQP:** Advanced Message Queuing protocol is an open application layer for business messaging.

   AMQP supports both point –to-point and publisher/subscriber models , routing and queuing .

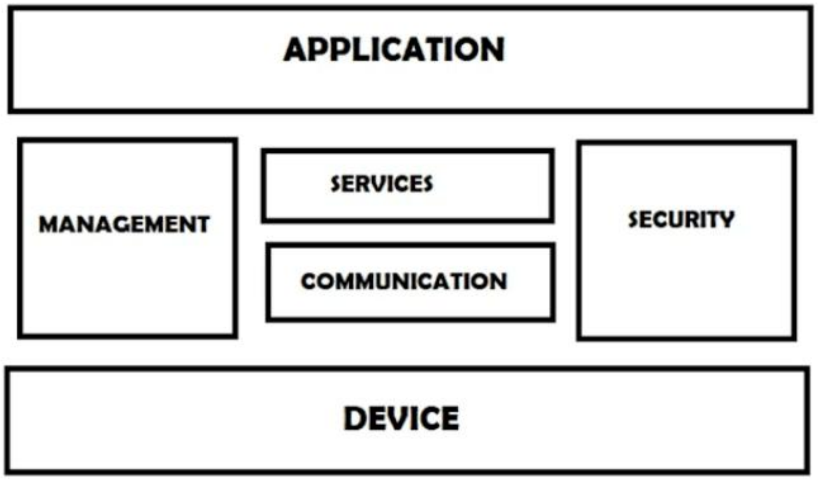
   AMQP brokers receive messages from publishers and route them over connections to consumers .

   Publishers publish the messages to exchanges which then distribute message copies to queues.

28. Describe the physical design of IoT



29. Explain Logic design of IoT



Application
It is an interface that provides a control system that use by users to view the status and analyze of system.

Management
This functional block provides various functions that are used to manage an IoT system.

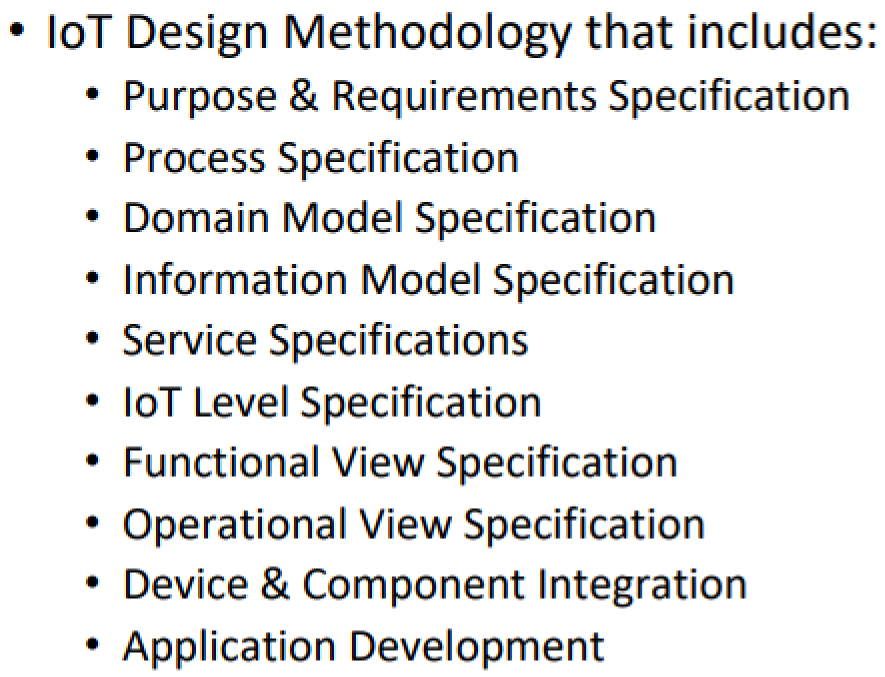
Services
This functional block provides some services like monitoring and controlling a device and publishing and deleting the data and restoring the system.

Communication
This block handles the communication between the client and the cloud-based server and sends/receives the data using protocols.

Security
This block is used to secure an IoT system using some functions like authorization, data security, authentication, 2-step verification, etc.

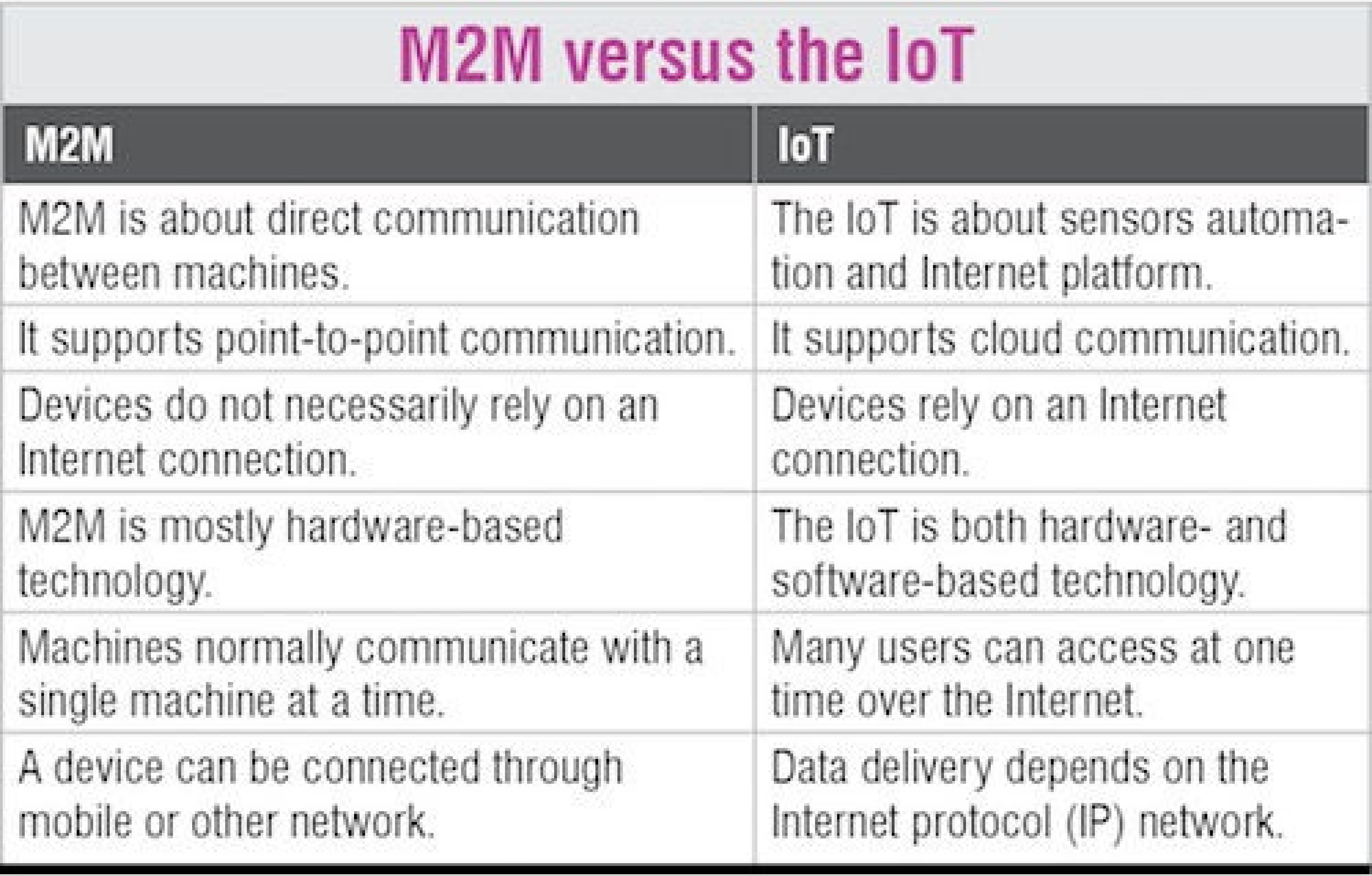
Device
These devices are used to provide sensing and monitoring control functions that collect data from the outer environment.

30. Describe IoT design methodology for controlling the light in the room using web application



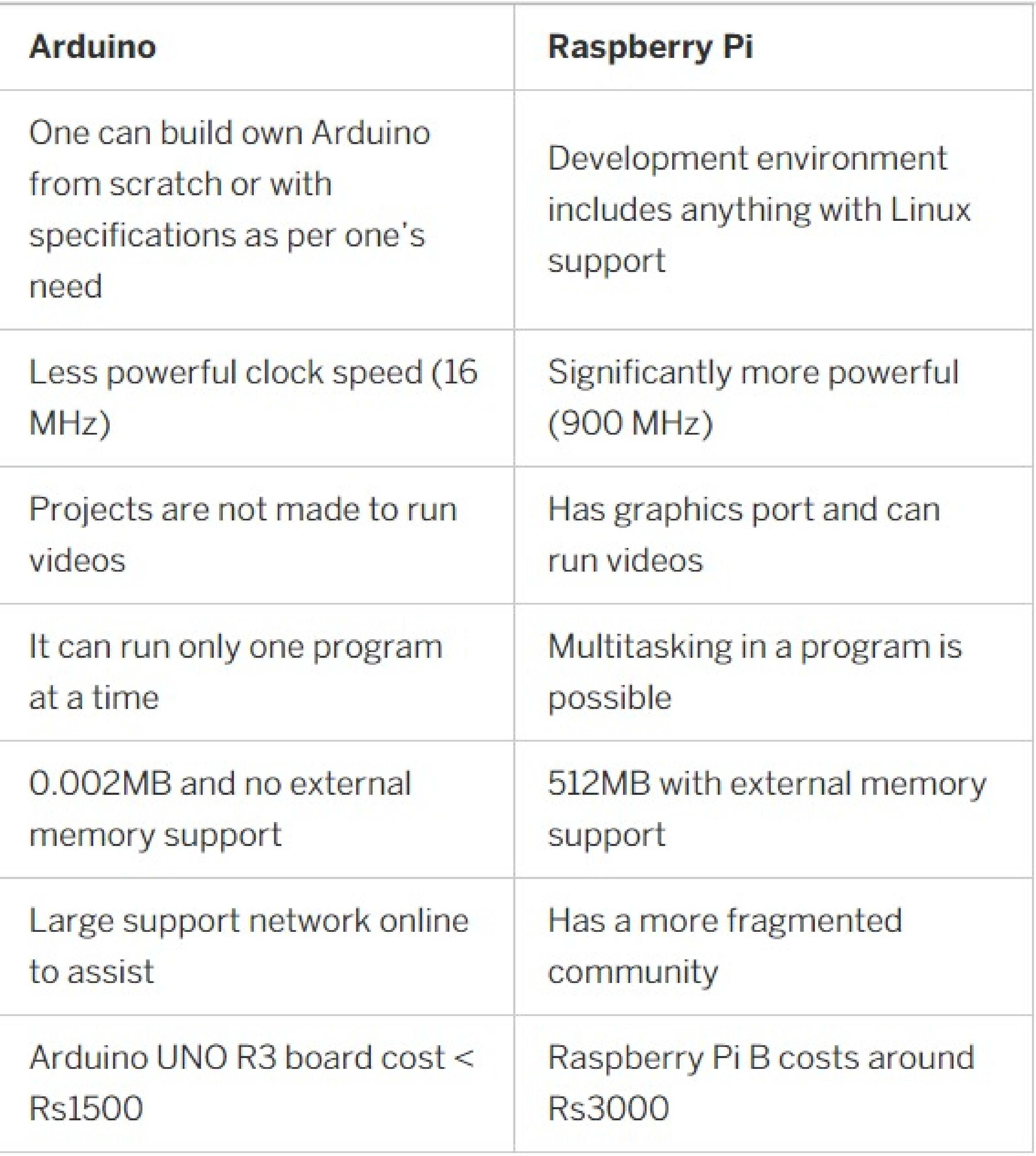
- IoT Design Methodology that includes:
  - Purpose & Requirements Specification
  - Process Specification
  - Domain Model Specification
  - Information Model Specification
  - Service Specifications
  - IoT Level Specification
  - Functional View Specification
  - Operational View Specification
  - Device & Component Integration
  - Application Development

Note: Learn Case study from PPT IOT Deployment templates.



M2M versus the IoT

M2M	IoT
M2M is about direct communication between machines.	The IoT is about sensors automation and Internet platform.
It supports point-to-point communication.	It supports cloud communication.
Devices do not necessarily rely on an Internet connection.	Devices rely on an Internet connection.
M2M is mostly hardware-based technology.	The IoT is both hardware- and software-based technology.
Machines normally communicate with a single machine at a time.	Many users can access at one time over the Internet.
A device can be connected through mobile or other network.	Data delivery depends on the Internet protocol (IP) network.



Arduino	Raspberry Pi
One can build own Arduino from scratch or with specifications as per one's need	Development environment includes anything with Linux support
Less powerful clock speed (16 MHz)	Significantly more powerful (900 MHz)
Projects are not made to run videos	Has graphics port and can run videos
It can run only one program at a time	Multitasking in a program is possible
0.002MB and no external memory support	512MB with external memory support
Large support network online to assist	Has a more fragmented community
Arduino UNO R3 board cost < Rs1500	Raspberry Pi B costs around Rs3000

I2C	SPI	CAN
1. I2C stands for inter Integrated circuit.	**1.** SPI stands for serial peripheral Interface.	**1.** CAN stands for controller Area network.
2. It is developed by the Philips.	**2.** It is developed by the Motorola.	**2.** It is developed by the Robert Bosch.
3. It is a Half Duplex protocol	**3.**It is a full duplex protocol	**3.** It is a full duplex protocol
4. Syncronization	**4.** Synchronization	**4.** Synchronization
5. It is two wire protocols SCL and SDL.	**5.** It is a four wire protocol SCL and MISO/MOSI, SS	**5.** It is a two wire protocol CAN H+ and CAN H-.
6. It is multi master protocol	**6.** It is single master protocol	**6.** It is multi master protocol
7. With in the circuit board	**7.** With in the circuit board	**7.** With in two circuit board.