18CS44

Model Question Paper-1 with effect from 2019-20 (CBCS Scheme)

USN : □□□□□□□□□□

Fourth Semester B.E. Degree Examination

Microcontroller and Embedded Systems

Time: 03 Hrs **Max. Marks:100**

Note: Answer any **FIVE** full questions, choosing at least **ONE** question from each **MODULE**

Module-1

1. a. Compare and Contrast microprocessor and microcontroller. 4M
   b. Explain ARM core data flow model with a neat diagram. 8M
   c. Along with neat diagram of an ARM based embedded device (Microcontroller), explain the four main hardware components. 8M

OR

2. a. Explain the different processor modes provided by ARM7. 8M
   b. Give the schematic of a Current Program Status Register of ARM7 processor briefing the individual bits. 6M
   c. What s Pipelining. Explain in detail schematically. 6M

Module-2

3. a. Explain the MOV instruction set provided by ARM7 with the example for each. 8M
   b. Explain the ARM swap instruction with an example code. 6M
   c. Brief about the categories of Load-Store instructions used with ARM. 6M

OR

4. a. Explain the ARM Single-Register and Multiple-Register load-store addressing modes with example. 8M
   b. Explain Co-Processor instructions of ARM Processor. 6M
   c. Write a note on Profiling and Cycle Counting. 6M

Module-3

5. a. What are the different types of memories used in Embedded System design? Explain the role of each. 10M
   b. List different purposes of embedded system with examples. 10M

OR

6. a. Briefly Describe the classification of embedded systems 8M

b. Explain the following:

i. I2C

ii.1-Wire Interface

iii. SPI Interface

iv. Reset Circuit 12M

Module-4

7. a. What are the operational and non-operational quality attributes of an embedded systems. 10M

b. Explain the different types of serial interface bus used in Automotive Communication. 4M

c. Design FSM model for tea/coffee vending machine. 6M

OR

8. a. Explain the fundamental issues in hardware software co-design. 6M

b. Explain with a neat block diagram, how source file to object file translation takes place. 8M

c. Explain the different embedded firmware design approaches. 6M

Module-5

9. a. With neat diagram explain operating system architecture. 8M

b. Differentiate between hard real time and soft real time operating system with a example for each. 4M

c. Define process. Explain in detail the structure, memory organization and state transmission of the process. 8M

OR

10. a. Explain the Simulator and Emulator. 8M

b. Write a note on message passing. 8M

c. Explain the concept of deadlock with a neat diagram. 4M

Fourth- Semester B.E Degree Examination.

me:- 03 Hrs
marks - 100.

Microcontroller & Embedded System.

Subcode :- 18CS44.

Module - 1.

1.a. Compare & contrast microprocessor & microcontroller? - 4m.

Microprocessor	Microcontroller.
1) Most of the time, general purpose in design & operation.	1) mostly application oriented or domain specific.
2) Targeted for high end market where performance is important.	2) targetted for embedded market where performance is not so critical.
3) Limited power saving options compared to microcontrollers	3) Includes lot of power Saving features.
4) It is a dependent unit,	4) It is a Self-contained unit.

1.b Explain ARM core data flow model with a neat diagram? 8M.

A programmer can think of an ARM core as functional units connected by data buses, as shown in the following fig.

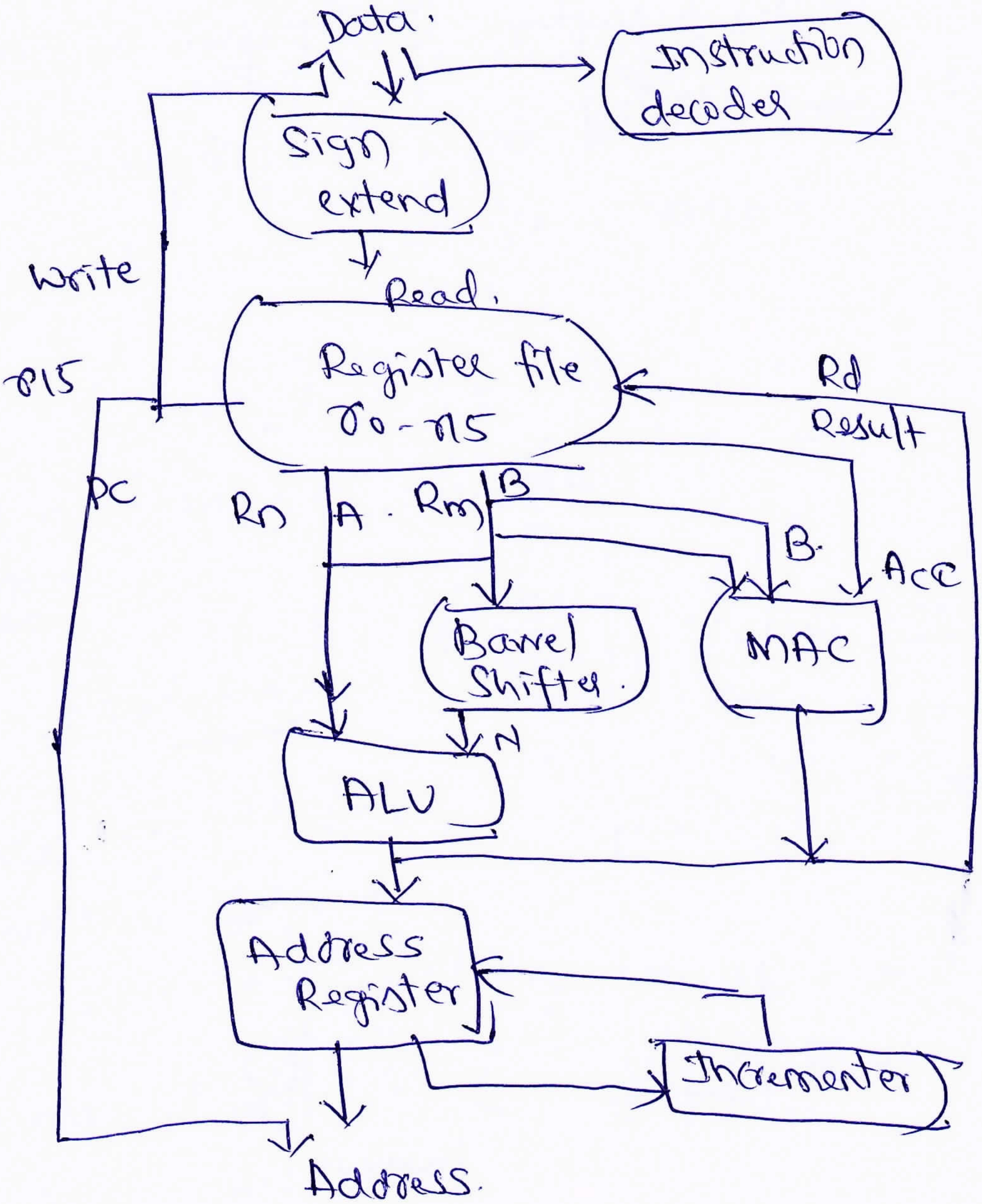


Fig: ARM Core dataflow model.

The arrows represent the flow of data, the lines represent the buses, and the boxes represent either an operation unit or a storage area.

Data enters the processor core through the Data bus. The data may be instruction to execute or a data item.

The instruction decoder translates instructions before they are executed each instructions executed belongs to a particular instruction set.

the ARM Processor, like all RISC Processors, uses load- store architecture.

1. Load instructions copy data from memory to registers in the Core.

2. Store instructions copy data from registers to memory.

* There are no data processing instructions that directly manipulate data in memory, Thus data processing is carried out in registers.

* Data items are placed in the register file.- a storage bank made up of 32-bit registers.

* ARM instructions typically have two source register Rn & Rm & a single destn register rd.

* The ALU or MAC takes the register values Rn & Rm from the A & B buses & computes a result.

* After passing through the functional units, the result in Rd is written back to the register file using the Result bus.

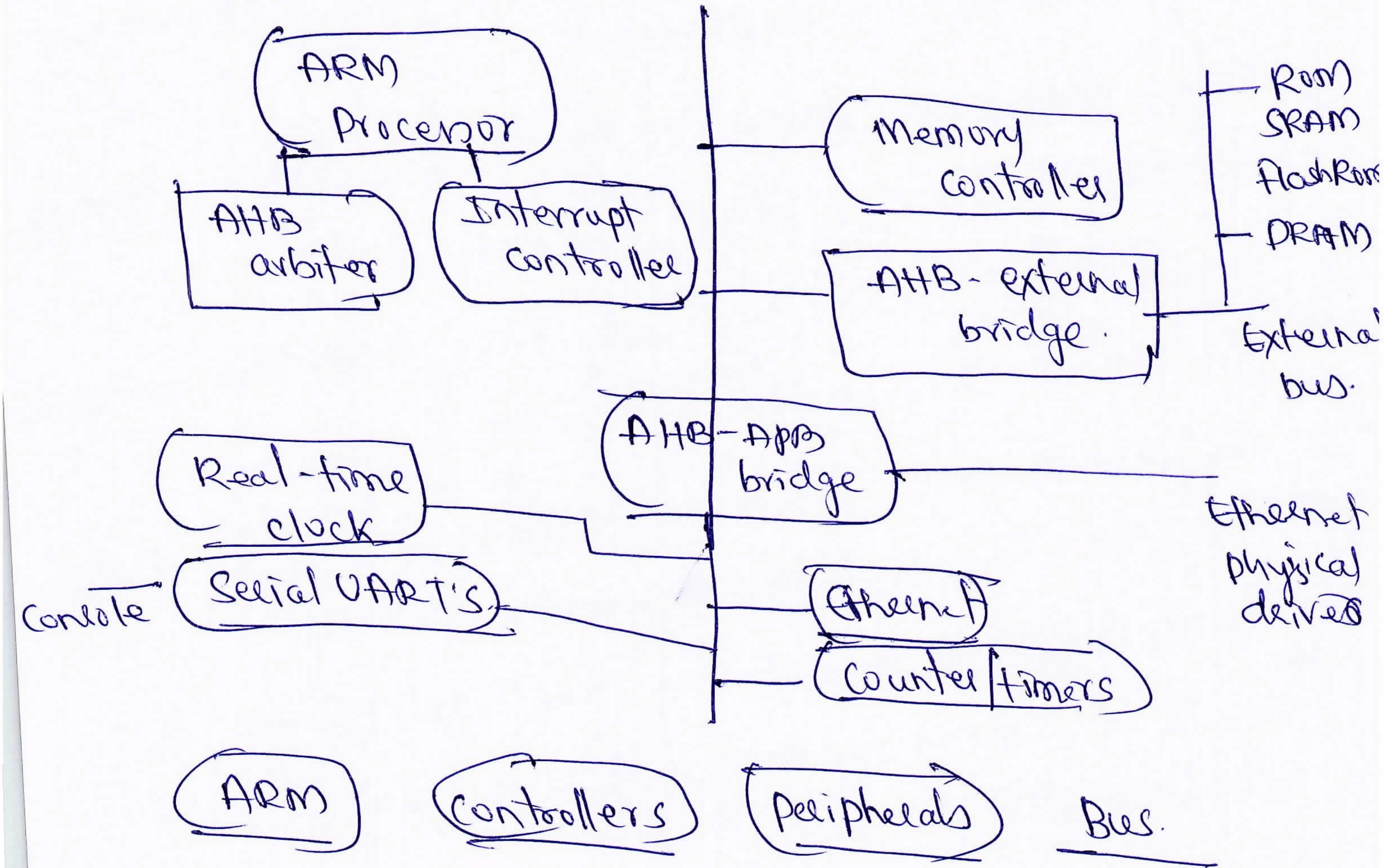
* For load & store instructions the incrementer updates the address register before the core reads or writes the next register value from or to the next sequential memory location.

d The processor continues executing instructions.

C. Along with neat diagram of an ARM based embedded device (Microcontroller). Explain the four main hardware components. 8m

The following figure shows a typical embedded device based on an ARM core.

Each box represents a feature or function. The lines connecting the boxes are the buses carrying data.



An- ARM- Based Embedded Device, a Microcontroller.

we can Seperate the device into four main hardware components.

1. The ARM Processor Controls the embedded device.

2. Contoollers Co-ordinate important functional blucks of the System.

3. The peripeals Provide all the input-output capability external to the Chip and are responsible for the uniqueness of the embedded device.

4. A bus is used to communicate between different parts of the device.

OR.

2a. Explain the different processor modes provided by ARM? 8m

The processor mode determines which registers are active & the access rights to the cpsr register itself. Each processor mode is either Privileged or non-privileged.

1. A privileged mode allows full read-write access to the cpsr.

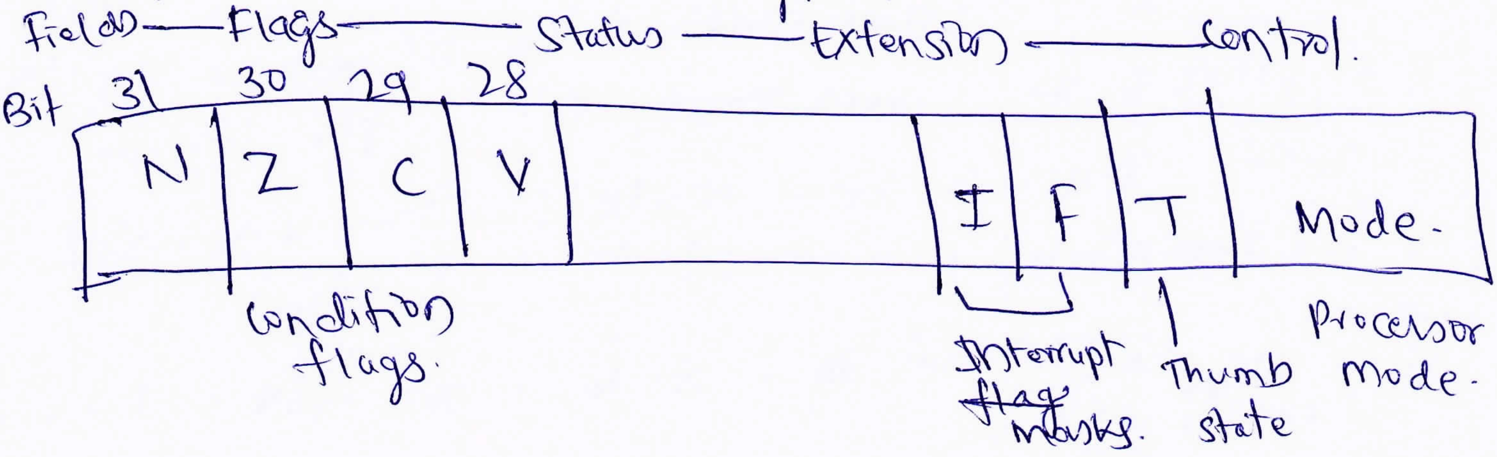
2. A non-privileged mode only allows read access to the control field in the cpsr, but still allows read -write access to the condition flags.

There are 7 processor modes in total.

1. Six privileged modes.
   1. abort
   2. fast interrupt request.
   3. Interupt request.
   4. Supervisor
   5. System
   6. Undefined.
2. one non-privileged mode.
   1. user.

b. Give the schematic of a Current Program Status register of ARM7 processor briefing the individual bits? 6m

The ARM core uses the cpsr to monitor & control internal operations. The cpsr is a dedicated 32-bit register & resides in the register file.



program Status register.

The CPSR is divided into four fields, each 8 bit wide: flags, status, extension & control.

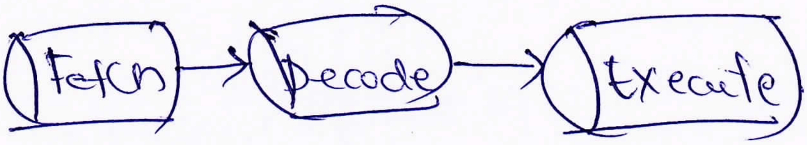
The control field contains the

1) The processor mode, State, Interrupt mask bits

2) The flag field contains the condition flags.

c. What is pipeling. Explain in detain schematically? 6m

A pipeline is the mechanism in a RISC processor, which is used to execte instructions.



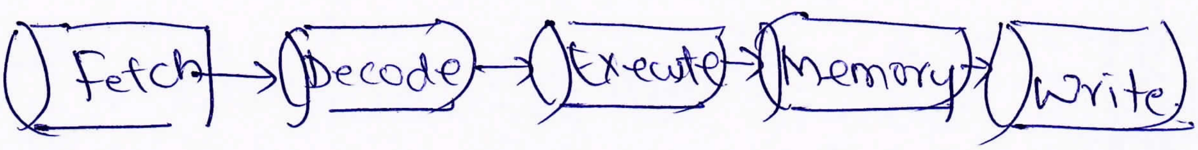
ARM7 Three-stage pipleline.

The above figure shows the three-stage pipeline.

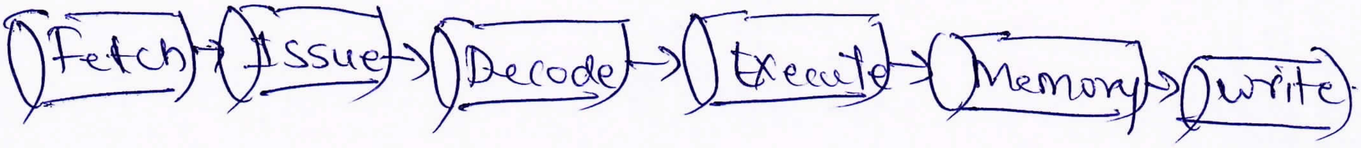
1. fetch loads an instruction from memory.

2. Decode identifies the instruction to be executed.

3. Execute processes the instruction & writes the result back to a register.



ARM9 Five-Stage pipeline.



ARM 10 Six-stage pipeline.

Module-2.

3.a. Explain the MOV instruction set provided by ARM? with a Example for each? 8m

Move instruction: Copies N into a destination registed Rd, where N is a register or a immediate value. This instruction is useful for Setting initial values & transfering data between registors.

Syntax: <instruction> {<cond>} {S} Rd, N.

MOV	Move a 32-bit value into a register	Rd = N.
MVN	move the NOT of the 32-bit value into a register	Rd = ~N.

i) PRE r5 = 5.
r7 = 8.

MOV r7, r5

POST r5 = 5
r7 = 5

ii). MVN r5, r6.

Q. Explain the ARM Swap instruction with an example code? 6m

The Swap instruction is a special case of a load-Store instruction. It swaps the contents of memory with the contents of a register.

Syntax: SWP {B} {<cond>} Rd, Rm, [Rn]

SWP	Swap a word between memory & a register	tmp = mem32[Rn]
		mem32[Rn] = Rm
		Rd = tmp.

SWPB	Swap a byte between memory & a register	tmp = mem8[Rn]
		mem8[Rn] = Rm
		Rd = tmp.

```
PRE   mem32[0x9000] = 0x12345678
          r0 = 0x00000000
          r1 = 0x11112222
          r2 = 0x00009000
      Swp r0, r1, [r2]
Post  mem32[0x9000] = 0x11112222
          r0 = 0x12345678
          r1 = 0x11112222
```

3c. Brief about the categories of Load-Store instructions used with ARM.? 6m

Load- store instructions transfer data between memory & processor Registers

There are 3 types of Load- Store instructions

1) Single- register transfer.

2) multiple - register transfer.

3) Swap.

1) Single- Register transfer.

These instructions are used for moving a single data item in & out of a register.

```
Syntax : <LDR|STR>{<cond>}{B} Rd,addressing
   LDR{<cond>}SB|H|SH  Rd, addressing.
   STR{<cond>}H Rd, addressing
```

2) multiple-register transfer.

```
syntax :- <LDM|STM>{<cond>}<addressing mode>
                Rn{!},<registers>{^}
```

- These instructions can transfer multiple registers between memory & the processor in a single Instruction

c) Swap instruction.

It swaps the contents of memory with the contents of a register.

Syntax: Swp {B} {<cond>} Rd, Rm, (Rn).

4.a). Explain the ARM Single-Register & multiple-Register load-store addressing modes with example? 8m

*. Single-register Transfer.

These instructions are used for moving a single data item in & out of a register.

* the datatypes supported are signed & unsigned words, half words, bytes.

Ex:- LDR r0, [r1] ; = LDR r0, [r1, #0]

STR r0, [r1] ; = STR r0, [r1, #0]

* multiple register transfer

This can transfer multiple registers between memory & the processor in a single instruction.

the transfer occurs from a base register Rn pointing with memory.

multiple register instructions are more efficient from single-register transfer for.

1. moving blocks of data around memory.

2. Saving & restoring context & stacks.

Ex:- mem32[0x80018] = 0x03

mem32[0x80014] = 0x02.

mem32[0x80010] = 0x01

r0 = 0x00080010.

r1 = 0x00000000

r2 = 0x00000000

r3 = 0x00000000

LDMA r0!, {r1-r3}

post r0 = 0x0008001c.

r1 = 0x00000001

r2 = 0x00000002

r3 = 0x00000003.

b. Explain co-processor instruction of ARM processor? 6m

these are used to extend the instructionset.

A co-processor can either provide additional computation capability or be used to control the memory subsystem including caches & memory management.

A co-processor instructions include data processing, register transfer, memory transfer instructions.

Syntax: CDP{<cond>} cp, opcode1, cd, cn, {, opcode2}

<MRC|MCR>{<cond>} cp, opcode1, Rd, cn, cm{, opcode2}

<LDC|STC>{<cond>} cp, cd, addressing.

CDP - co-processor data processing.

MRC MCR - co-processor register transfer.

LDC STC co-processor memory transfer.

Example :-

MRC p15, 0, r10, c0, c0, 0.

4.c. Write a note on profiling & cycle counting? 6m

1- identify critical routines & ~~mea~~ measure their time.

Profiler: Tool to measure the cycle count consumed by any subroutine.

cycle counting : Activity done by profiler.

5 stage pipelining.

1. Fetch : collect operands/ instruction.
2. Decode: understand instruction.
3. ALU : Execute instruction.
4. LS1 : load-store 1 updating result of previous ALU.
5. LS2 : Load-store 2 updating of previous ALU/LS1.

Instruction Address	PC	PC-4	PC-8	PC-12	PC-16.
Action	Fetch	Decode	ALU	LS1	LS2

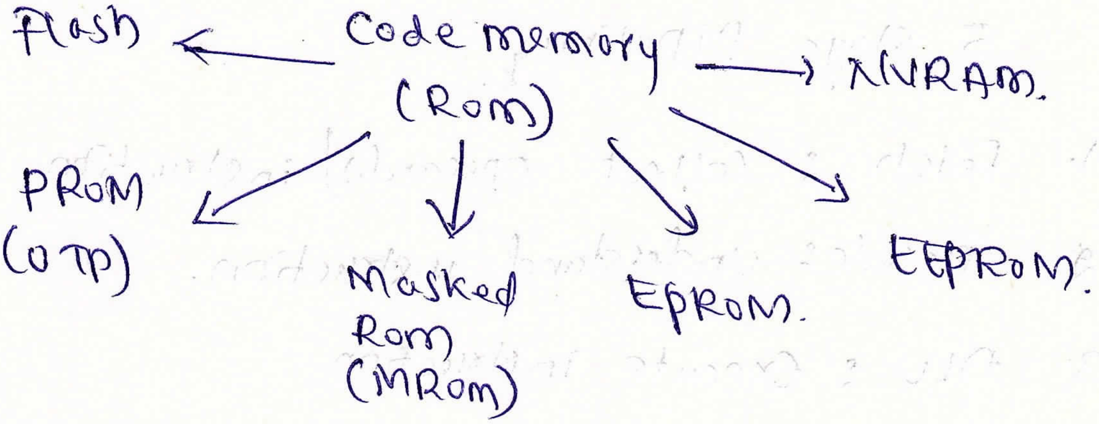
Module - 03.

5a. whats are the different types of memories used in Embedded System design? Explain the role of each.
ROM

Memories/memory is an important part of an Processor/controller based Embedded system.

Program Storage memory (ROM)

It can be classified into dif types as shown.



* Masked-Rom :- It is a one-time programmable device.

It ~~is~~ ~~a~~ makes use of the hardwired technology for storing data.

* programmable Read only memory (OTP) :-

One-time programmable memory (OTP) or PROM is not preprogrammed by the manufacturer.

The end-user is responsible for programming these devices.

*. EpRom (Erasable programmable Read only memory):-

It gives flexibility to re-program the same chip.

It stores the bit information by changing the floating gate of an FET:

* EEpRom (Electrically Erasable programmable Read only memory).

* The information contained in the EEPRom memory can be altered by using Electrical signals at the register/byte level.

* They can be erased & reprogrammed in circuit.

* Flash

Flash memory is a variation of EEpRom technology.

Flash memory is organized as sectors or pages.

* Non-Volatile RAM (NVRAM):

It is a random access memory with battery backup.

The memory & battery are packed in a single package.

The life span of NVRAM is expected to be around 10 years.

5b. List diffeent purposes of embedded system with examples? 10m

1. Data Collection | Storage | Representation.

   Example :- A Digital Camera.

2. Data Communication

   Examplle :- A wireless network router.

3. Data (signal) Processing.

   Example :- A digital hearing aid.

4. Monitoring

   Example :- Digital CRO.

5. Control

   Example :- An Air Conditioner system.

6. Application Specific user interface.

   Example :- Mobile phone.

6a. Briefly Describe the classification of embedded System. 5m

Some of the criteria used in the classification of embedded system are.

1. Based on generation
2. complexity & performance reqts.
3. Based on deterministic behaviour.
4. Based on triggering.

* Classification Based on Generation.

  1. First generation
  2. Second generation
  3. Third generation
  4. Fouth generation
  5. Next generation.

* classification Based on Complexity & performance.

  1. Small-scale Embedded systems
  2. Medium -Scale Embedded Systems.
  3. Large -Scale Embedded systems.

* classification Based on Deterministic Behavior:

  1. Real time Systems.
  2. Deterministic or non -deterministic.
  3. Hard-real time & soft real time.

* Classification Based on Triggering.

  1. Reactive System can be either Event triggered or time triggered.

6b. Explain the following 12m

1. I2C.
2. 1-wire interface.
3. SPI interface.
4. Reset circuit.

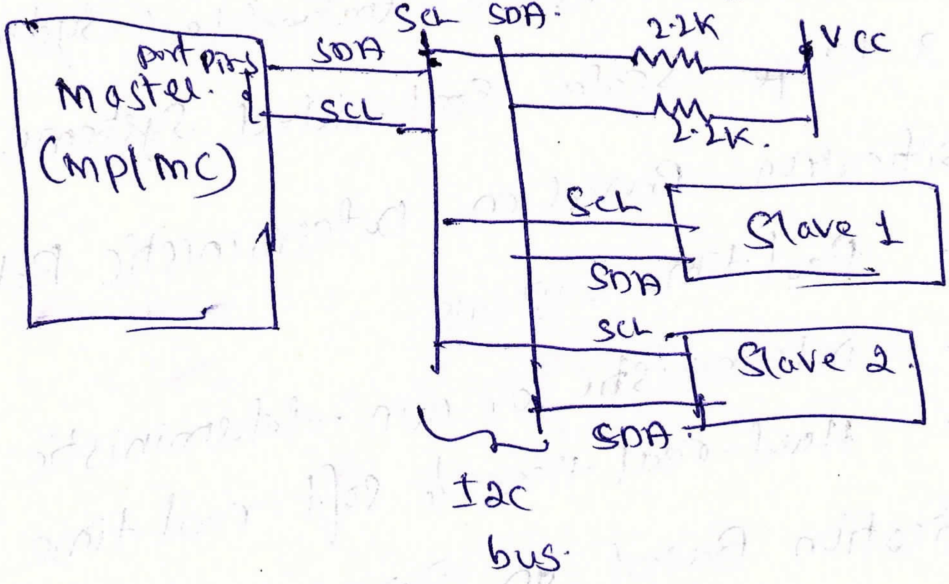
1. Inter integrated Circuit (I2C) Bus.

The concept of I2C bus was developed by Philips Semiconductors in the early 1980's.

The I2C bus Comprise of two bus lines.

1. Serial Clock (SCL line),
2. Serial Data (SDA Line)

I2C supports multimasters on the Same bus.



I2C Bus interfacing

2. Serial peripheral Interface (SPI) Bus:-

IS a Synchronous bi-directional full duplex four-wire Serial interface bus.

The concept of SPI was introduced by Motorola.

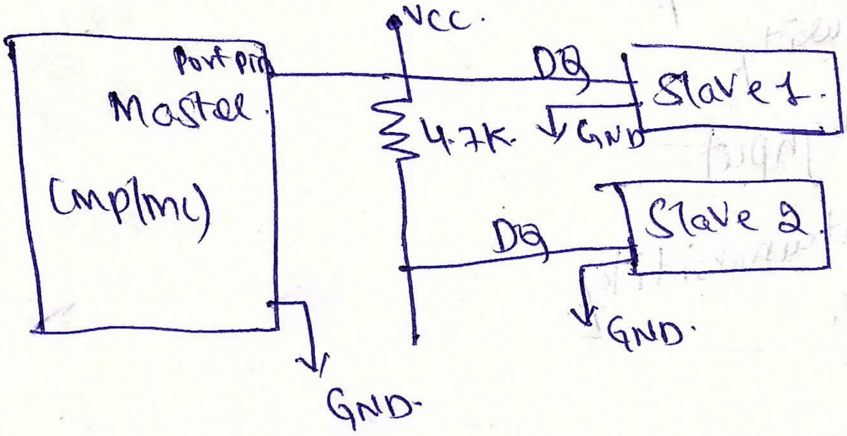
SPI is a single master multi-slave System.

SPI requires four Signal lines for communication

1. MOSI — Master out Slave in
2. MISO — Master in Slave out
3. SCLK — Serial clock.
4. SS — ~~So~~Slave Select.

3. 1-wire Interface.

It is also known as Dallas 1-wire protocol.



1-wire interface.

Every 1-wire device contains a globally unique 64-bit identification number stored within it.

4. Reset Circuit :-

A reset is used at power-on to hold th processor in 'reset' until the power has stabilized.

During 'reset' the clock is stopped, the registers & RAM are cleared & the program counter is set to zero.

Coming out of reset, the clock is star & the program jumps to its starting address.

Module -04.

7a what are the operational & non-operational quality attributes of an embedded systems? 6m

Operational Quality attributes

1. Response.
2. Reliability.
3 Security.
4. Throughput.
5 Maintainability.
6. Safety.

Non-operational quality attributes.

1. Testability & Debug-ability.
2. Evolvability.
3 Portability
4. Time to Prototype & market.
5. Per unit & total cost.

7b. Explain the different types of serial interface bus used in Automotive communication? 4m

1. Controller Area Network (CAN).
2. Local Interconnect Network (LIN)

3. The media oriented system transport (MOST).

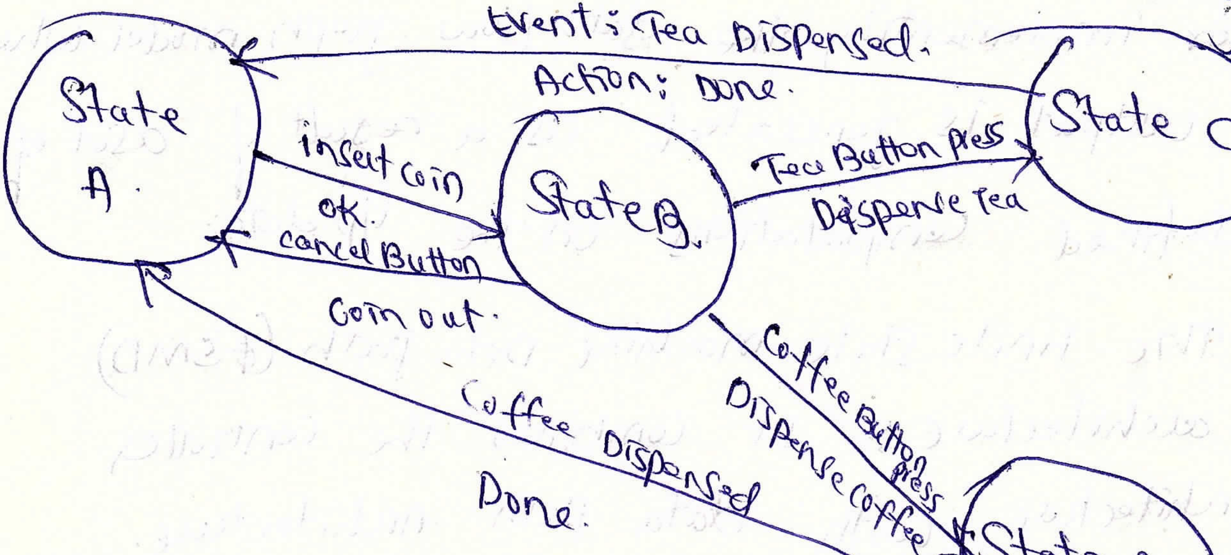
1) CAN :-

It is an event driven serial protocol interface with support for error handling in data transmission.

2) LIN : it is a single master multiple slave communication interface.

3. MOST :- it is targeted for automotive audio vedio equipment interfacing.

7C. Design FSM model for tea / coffee vending machine. 6m



FSM model.

8a. Explain the fundamental issues in hardware software co-design? 6m.

1. Selecting the model :- Here models are used for capturing & describing the s/m characteristics

2. Selecting the Architecture :- A model only captures the system characteristics & does not provide information on 'how the system can be manufactured?

3. The controller Architecture :- It implements the finite State machine model using a State register & two combinational circuits.

4. The Datapath Architecture : it is best suited for implementing the data-flow graph model where the output is generated as a result of a set of predefined computations on the i/p data.

5. The finite state machine Data path (FSMD) architecture :- It combines the controller architecture with data path Architecture.

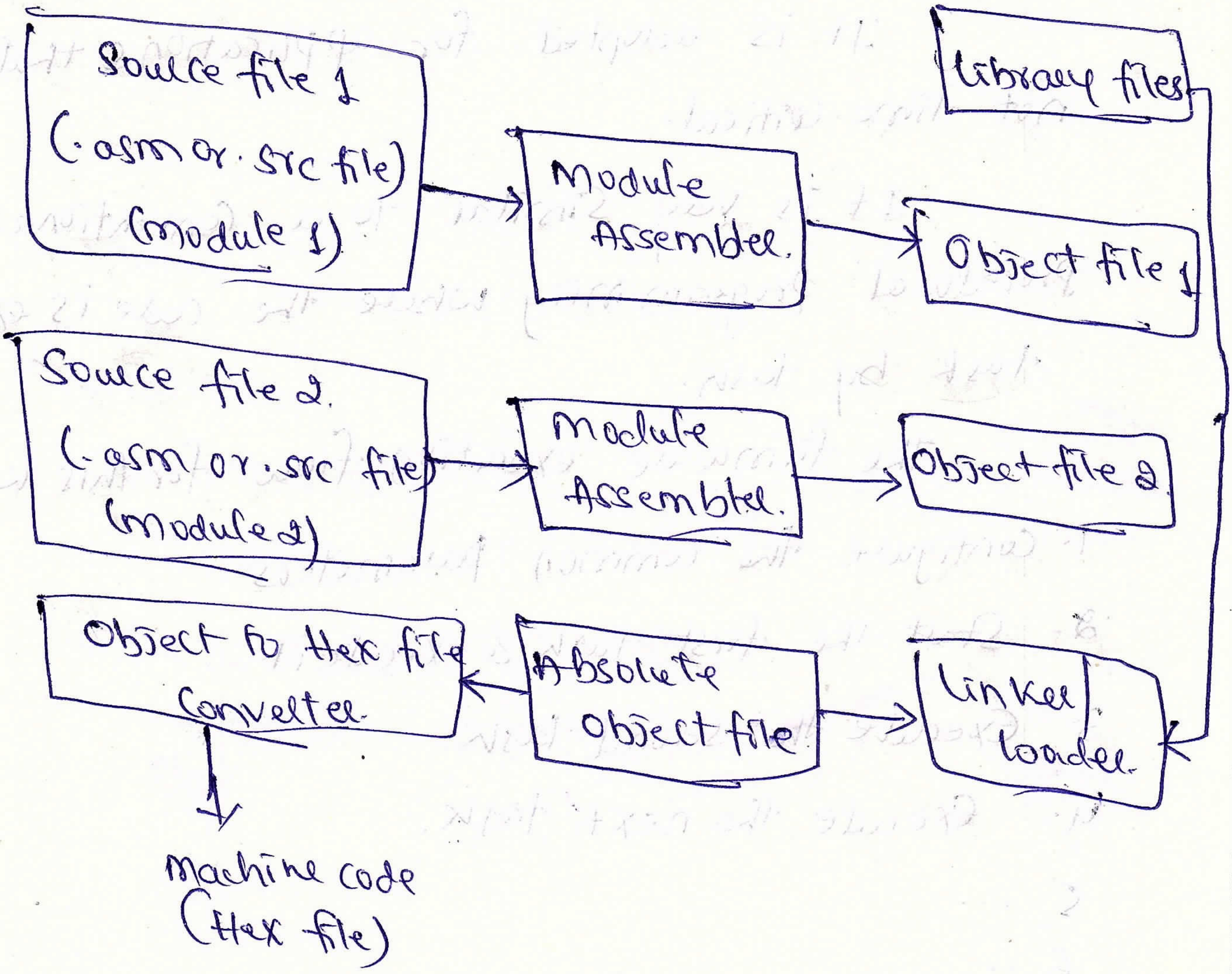
6. The complex instruction Set computing (CISC) Architecture :- uses an instruction set representing complex operations

7. The Reduced instruction set computing (RISC) architecture :- it reuses instruction set representing simple operations.

8. Selecting the language: A programming language Captures a 'computation model' & maps it into architecture.

9. partitioning System Requirements into Hardware & software.

8.b. Explain with a neat block diagram, how Source file to object file translation takes place? 8m



Each Source module is written in Assembly & is stored as .src file or .asm file.

On successful assembly

8.c Explain the different embedded firmware design approaches ? 6m

Two basic approaches are used for embedded firmware design.

1. Conventional Procedural Based firmware design (Super loop model).
2. Embedded operating system (os) Based design.

1. Super loop model:-

It is adopted for applications that are not time critical.

It is very similar to a conventional Procedural programming where the code is executed task by task.

The firmware execution flow for this will be

1. configure the common parameters
2. Start the first task & execute it.
3. Execute the second task.
4. Execute the next task.
5. ...
6. ...
7. Execute the last defined task.
8. Jump back to the first task & follow the

2. The Embedded operating System (OS) Based approach.

It contains operating systems, which can be either a (GPOS) or (RTOS) to host the user written application firmware.

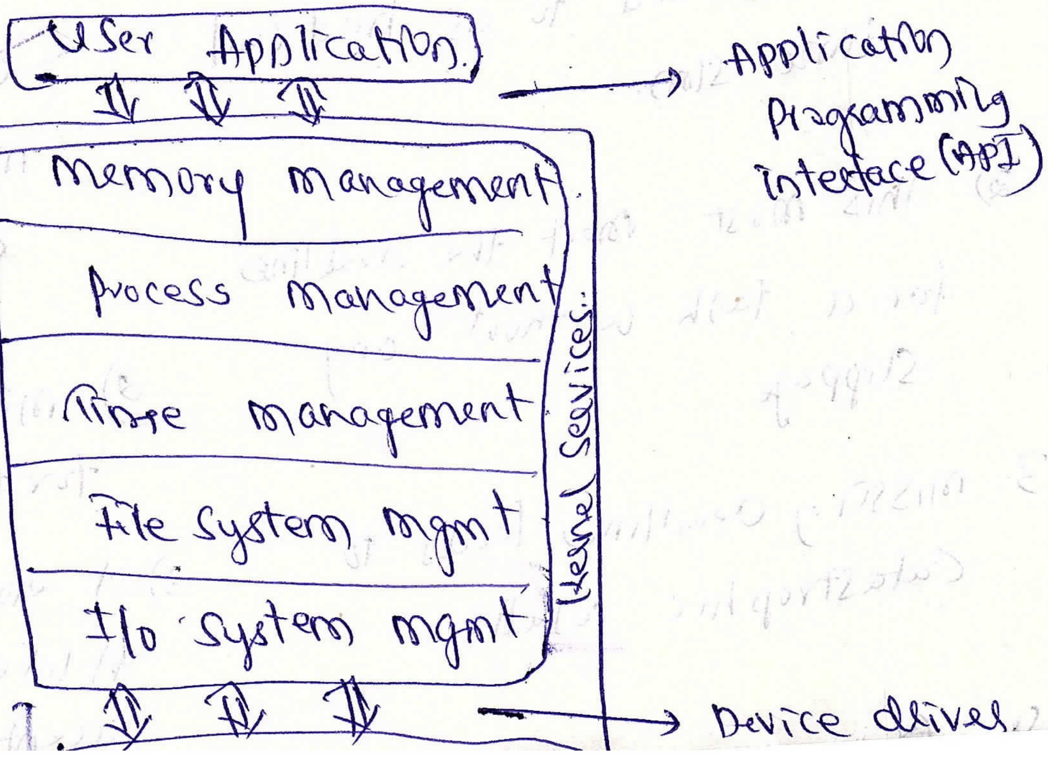
1. The General purpose OS (GPOS)

2. OS Based applications also require Drives Software.

3. The Real time operating System (RTOS). it allows flexible Scheduling of system resources.

Module 05

9.a With neat diagram explain Operating System architecture? 8m



Keenel :- it is the core of opelating System.

It is responsible for Managing the s/m resources & Communication among the h/w & s/w.

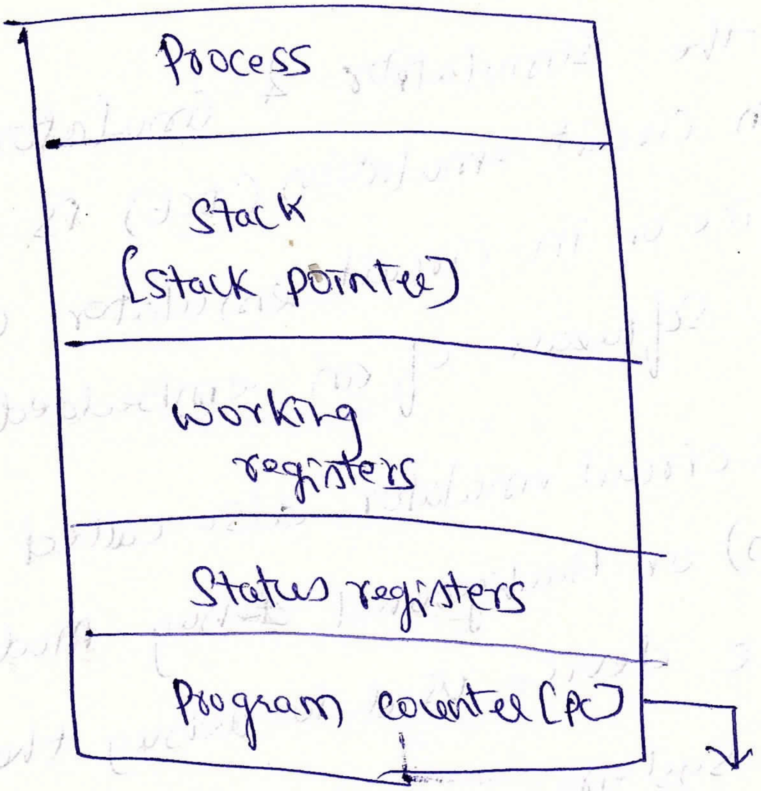
1. Memory mgmt.
2. process mgmt.
3. time mgmt.
4. File S/m mgmt.
5. I/O s/m mgmt.

6. Diffeeentiate between haed real time & soft time operating System with a example for each? 4m

Haed Real Time S/m.	Soft-Real time S/m.
1) It strictly adhele to the timing constraints for task is refeleed to as haed real time s/m.	1) It doesn't quaeantee meeting deadlines but offes best effort to meet the ~~ded~~ deadlin ale refeleed as soft-real time s/m.
2) This must meet the deadlines for a task without any slippage	2) missing deadlines for tasks ale acceptable
3. missing deadlines leads to Catastrophic results.	3) It emphasis the princip 'A late answel is an acceptable answel

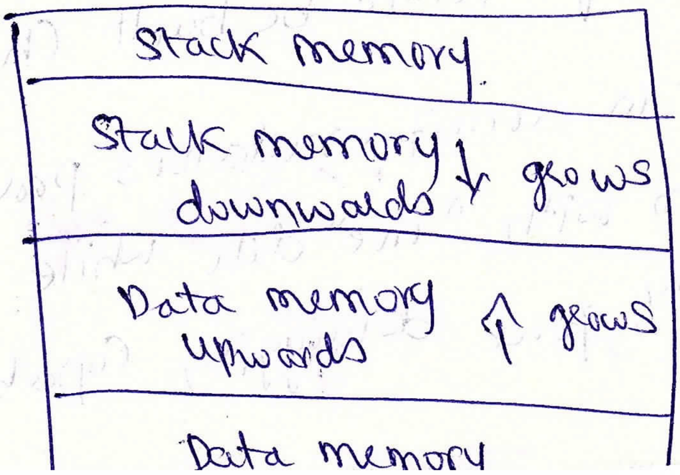
Q.C. Define process. Explain in detail the Structure, memory organization & state transmission of the Process? 8m

A process is program or part of it in execution, it is also known as instance of program in execution



Structure of a process

Memory organisation of a process.



The different States present in process are

1. Created state.
2. Ready State
3. Running State
4. Blocked State.
5. Completed State.

10a. Explain the Simulator & Emulator? 8m

In Circuit emulation (ICE) is the use of a hardware device or in-circuit emulator used to debug the software of an embedded system.

An in-circuit emulator also called on-circuit debugger (OCD) or Background debug module (BDM) is a hardware device used to debug the S/w of an embedded system.

ICE can also refer to the use of hardware emulation, when the emulator is plugged into a system in place of a yet-to-be built Chip.

These in-circuit emulators provide a way to run the System with "live" data while still allowing relatively good debugging Capabilities.

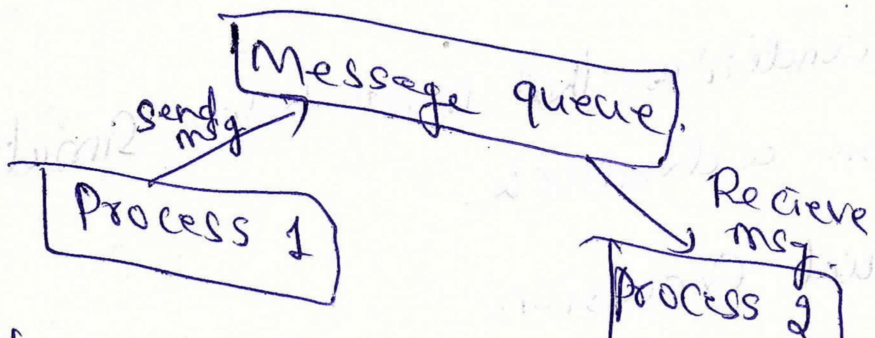
b. Write a note on message passing? 8m

Message passing is an (a)synchronous information exchange mechanism used for inter Process/Thread communication.

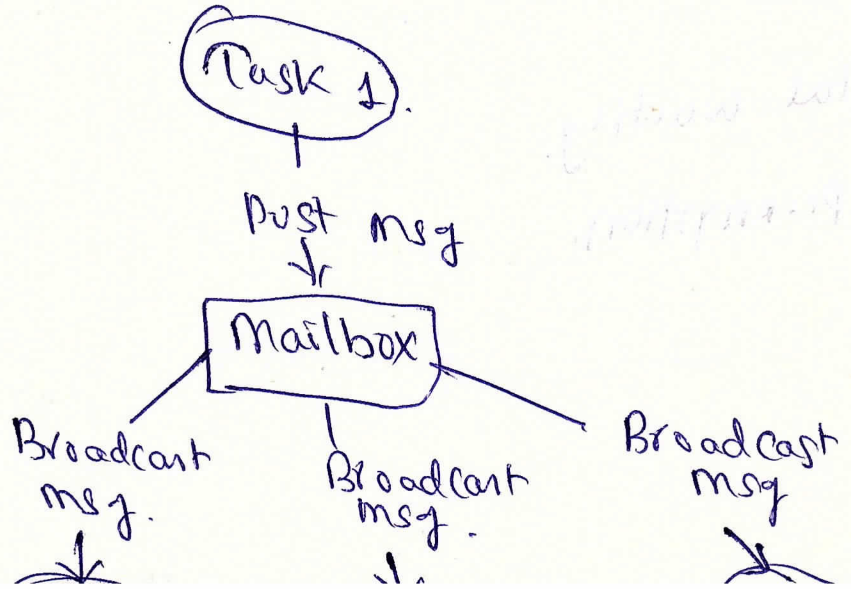
Based on message passing operation between processes, message passing is classified into

1) message queue.
2) Mail Box
3) Signalling.

1) message queue.



2) mail box.



3- Signalling :- it is primitive way for communica-tion. it is used for asynchronous notifications where one process| thread is fires a signal, indicating occurance of a scenario which the other process| threads in waiting.

to.c ⟸ Explain the concept of deadlock with a neat diagram? 4m.

Deadlock condition creates a situation where none of the processor are able to make any progress in their execution, resulting in a set of deadlocked processes.

four conditions that must hold simultaneously for there to be a deadlock.

1. Mutual Exclusion.
2. Hold & wait
3. Circular waiting.
4- No - Preemption.

PREPARED BY,

(Sandeep. P)

HOD.

Dean, Academics