FIRST TERM EXAMINATION [SEPT. 2016]
SEVENTH SEMESTER [B.TECH]
EMBEDDED SYSTEM [ETEC–401]

Time : 1½ hrs. **M.M. : 30**

Note: *Attempt any two more question from the rest. Question No.1 is compulsory.*

Q. 1. All Question are compulsory

Q. 1. (a) What is Embedded System? Explain with the help of suitable examples. What are the characteristics of Embedded System? **(2)**

Ans. An embedded system is a system that has embedded software and computer-hardware, which makes it a system dedicated for an applications or specific part- of an application or product or a part of a larger system.

Applications: Embedded systems have very diversified applications. A few selected application areas of embedded systems are telecommunications, smart cards, missiles and satellites, computer networking, digital consumer electronic and automotives.

Q.1. (b) What is the difference between Microcontroller and Microprocessor? Write any three differences. **(2)**

Ans. Difference between microprocessor and microcontroller:

(*i*) A microprocessor is a single chip CPU whereas a microcontroller has CPU and much of the remaining circuitry to make a microcomputer.

(*ii*) Microprocessor has many opcodes for moving data from external memory to the CPU but microcontroller has one or two.

(*iii*) Microprocessor may have one or two types of bit handling instructions, microcontroller have many bit handling instructions.

(*iv*) Microprocessor is concerned with rapid movement of code and data from external address to the chip but the microcontroller is concerned with rapid movement of bits wi thin the chip.

Q.1. (c) Discuss the difference between General Purpose and Embedded System. Give suitable examples. **(2)**

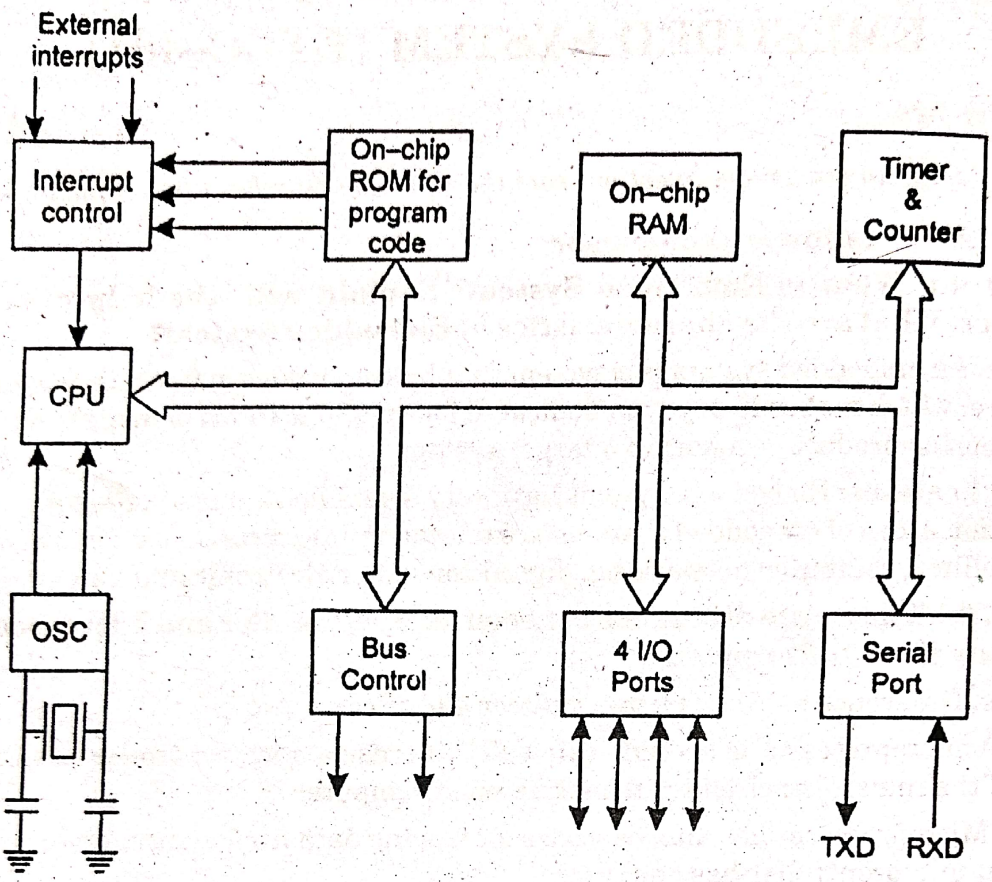
Ans. General Computer System is a computer that is built to be customizable in software, like desktop PCs and laptops, we can make it do many thing, sometimes together, with low or no constraints on power, performance or cost, also a general computing system is contained in itself, it's not a part of a larger system but it is the system itself.

Embedded System, as it appears from its name, is a part of a bigger system, a computer restricted to one function (or a finite set of functions) that controls, monitors or integrate with larger systems like automotive, robotics, home appliances and military applications.

Q.1. (d) Draw the 8051 architecture and discuss its features. **(2)**

Ans. 8051 Architecture: 8051 is an 8-bit microcontroller introduced in 1981. It has 128 bytes of RAM, 4K bytes of on-chip ROM, two timers, one serial port, and four input/output ports (each 8-bits wide) all on a single chip. The 8051 is an 8-bit processor, meaning that the CPU can work on only 8 bits of data at a time. Data larger than 8-bits has to be broken into 8-bit pieces to be processed by the CPU. 8051 can have a maximum

of 64K bytes of on-chip ROM, but many manufacturers have put only 4K bytes on the chip.



8051 can have external memory also. We can connect 64KB of external ROM & 64KB of external ROM with it. RAM of 8051 is divided in four register banks, bit addressable memory and a scratch pad (general purpose storage). It also has SFR's (Special function register) like TMOD, TCON, IE, DPTR etc. Different SFR's have different functionality.

8051 has the capability to communicate serially with IBM PC at different baud rates. Also, there are six different interrupt sources Reset, Timer 0, Timer 1, External interrupt 0 (INTO), External interrupt 1 (INT1) and serial communication interrupt.

8051 microcontroller is different from microprocessor in the sense that it is having CPU along with a fixed amount of RAM, ROM and I/O Ports, timers and counters etc. on a single chip. While in designing a micro—processor based system these peripherals need to be connected externally, which makes the system bulkier and much more expensive. But microprocessor based systems have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/O ports needed to fit the task at hand. This is not the case with microcontrollers.

Q.1. (e) Describe the RAM architecture of 8051. **(2)**

Ans. RAM Structure

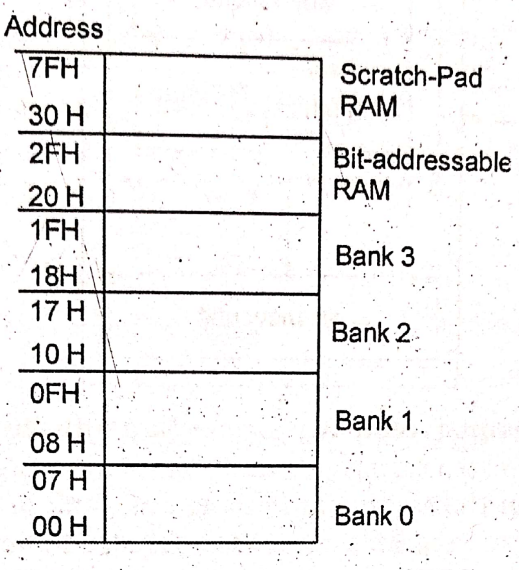
RAM of 8051 is of size 128 Bytes. So its address range is 00-7FH. RAM is containing:

→ 4 Memory banks (Bank 0,1, 2, 3)

→ 16-bytes of bit-addressable memory

→ Scratch Pad RAM of 80 bytes

Bank 0 (Address Range : 00-07H)
Bank 1 (Address Range : 08-0F H)
Bank 2 (Address Range : 10-17 H)
Bank 3 (Address Range : 18-01F H)



Q. 2. (a) What is PIC? What are the features of PIC processor? **(5)**

Ans. The features of PIC microcontroller are given below:

1. PIC uses harvard architecture and they are high peformance RISC processors. Harvard architecture has the program memory and data memory as separate memories and are accessed from separate buses.

2. The register files/data memory can be directly or indirectly addressed. All special fucntion registers, including the program counter, are mapped in the data memory.

3. The program memory bus is 14-bits wide in PIC. The entire instruction is fetched in a single machine cycle. All the required information is contained within a instruction and executed in a single cycle.

4. When PIC is operated at its maximum clock rate, almost each intruction can be executed within 0.2μs or five instructions can be executed per microsecond.

5. The PIC can put itself to sleep to save power during intervals when it has nothing to do. Thus, PIC supports a power saving SLEEP mode. A software command allows the PIC to enter into this mode. The PIC remains in SLEEP mode till it is reset again.

6. A single instruction of PIC can select and drive a single output pin high or low in its instruction execution time (0.2 μs). A load of up to 25 mA can be driven by this pin.

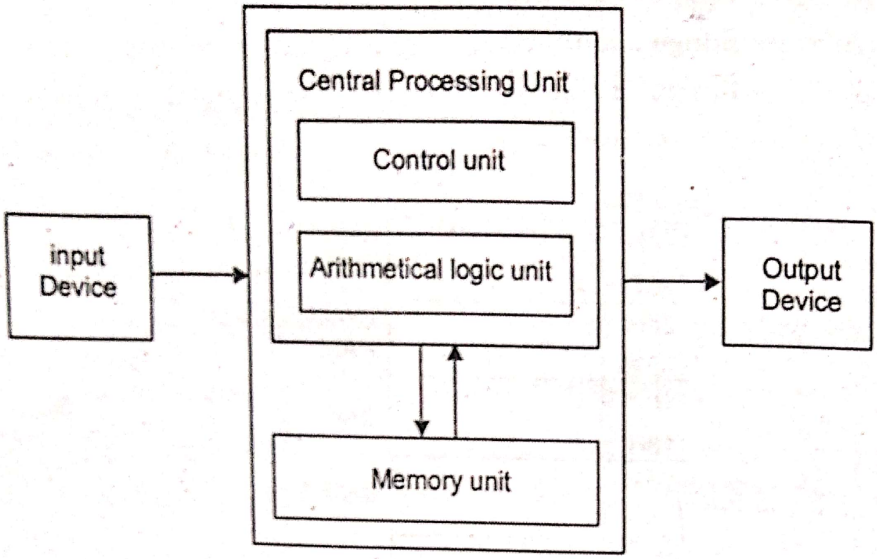
7. There is a built-in serial peripheral interface.

8. The PIC can control up to 12 independent interrupt sources.

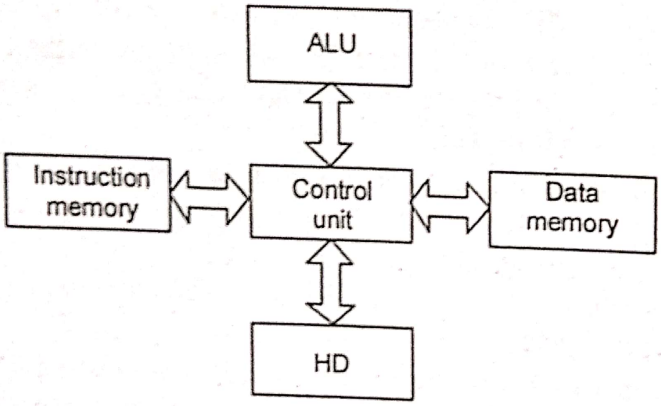
9. Analog to digital conversion function is supported by most of the PICs.

Q. 2. (*b*) Explain Von-Neumann and Harvard architecture. What are the technical reasons that Harvard architecture is better than Von Neumann architecture? **(5)**

Ans. The Von Neumann architecture describes a design architecture for an electronic digital computer with parts consisting of a processing unit containing an arithmetic logic unit and processor registers; a control unit containing an instruction register and program counter; a memory to store both data and instructions; external mass storage; and input and output mechanisms.



The Harvard architecture is computer architecture with physically separate storage and signal pathways for instructions and data. These early machines had data storage entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator; the processor could not initialize itself.



The design of a von Neumann architecture machine is simpler than that of a Harvard architecture machine, which is also a stored-program system but has one dedicated set of address and data buses for reading data from and writing data to memory, and another set of address and data buses for instruction fetching.

Q. 3.(a) What are SFRs of PIC? Explain in details. (5)

Ans. SFR of PIC MICROCONTROLLER

	R/W(0)	R/W(0)	R/W(0)	R(1)	R(1)	R/W(x)	R/W(x)	R/W(x)	
Status	IRP	RP1	RP0	TO	PD	Z	DC	C	Features
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit name

The STATUS register contains: the arithmetic status of the W register, the RESET status and the bank select bits for data memory. One should be careful when writing a value to this register because if you do it wrong, the results may be different than expected. For example, if you try to clear all bits using the CLRF STATUS instruction, the result in the register will be 000xx1xx instead of the expected 00000000. Such errors occur because some of the bits of this register are set or cleared according to the hardware as well as because the bits 3 and 4 are readable only. For these reasons, if it is required to change its content (for example, to change active bank), it is

recommended to use only instructions which do not affect any Status bits (C, DC and Z).

- **IRP** – Bit selects register bank. It is used for indirect addressing.
  - **1** – Banks 0 and 1 are active (memory location 00h-FFh)
  - **0** – Banks 2 and 3 are active (memory location 100h-1FFh)
- **RP1,RP0** – Bits select register bank. They are used for direct addressing.

RP1	RP0	ACTIVE BANK
0	0	Bank0
0	1	Bank1
1	0	Bank2
1	1	Bank3

- **TO – Time-out bit.**

  1 – After power-on or after executing CLRWDT instruction which resets watch-dog timer or SLEEP instruction which sets the microcontroller into low-consumption mode.

  0 – After watch-dog timer time-out has occurred.

- **PD – Power-down bit.**

  1 – After power-on or after executing CLRWDT instruction which resets watch-dog timer.

  0 – After executing SLEEP instruction which sets the microcontroller into low-consumption mode.

- **Z – Zero bit**

  1 – The result of an arithmetic or logic operation is zero.

  0 – The result of an arithmetic or logic operation is different from zero.

- **DC – Digit carry/borrow** bit is changed during addition and subtraction if an "overflow" or a "borrow" of the result occurs.

  1 – A carry-out from the 4th low-order bit of the result has occurred.

  0 – No carry-out from the 4th low-order bit of the result has occurred.

- **C – Carry/Borrow bit** is changed during addition and subtraction if an "overflow" or a "borrow" of the result occurs, i.e. if the result is greater than 255 or less than 0.

  1 – A carry-out from the most significant bit of the result has occurred.

  0 – No carry-out from the most significant bit of the result has occurred.

Q. 3. (b) What is the difference between CISC and RISC? Explain with suitable examples. List the controllers and processor with their classification in terms of CISC and RISC. **(5)**

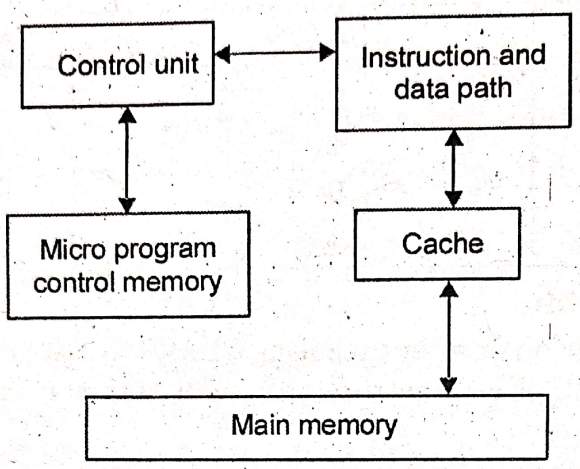
Ans. RISC (Reduced Instruction Set Computing) and CISC (Complex Instruction Set Computing) are two computer architectures that are predominantly used nowadays. The main difference between RISC and CISC is in the number of computing cycles each of their instructions take. With CISC, each instruction may utilize a much greater number of cycles before completion than in RISC.

The differences are:

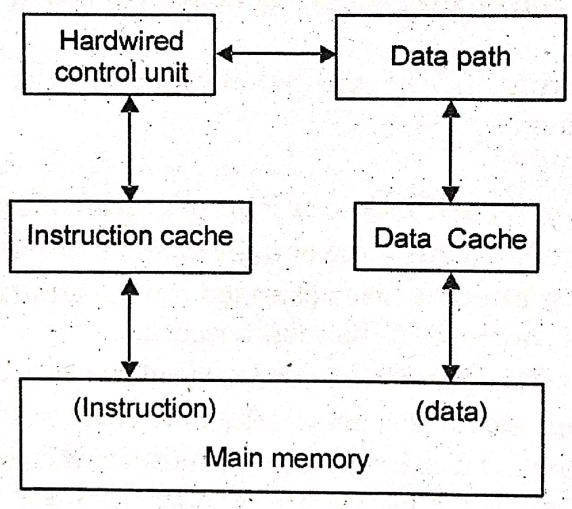
- CISC instructions utilize more cycles than RISC
- CISC has way more complex instructions than RISC

- CISC typically has fewer instructions than RISC
- CISC implementations tend to be slower than RISC implementations
- Computers typically use CISC while tablets, smart phones and other devices use RISC.

The architecture shown below represents the controller and processor with RISC and SICS



CISC ARCHITECTURE



RISC ARCHITECTURE

Q. 4. (a) Explain the features of ARM. **(5)**

Ans. An ARM processor is one of a family of CPUs based on the RISC (reduced instruction set computer) architecture developed by Advanced RISC Machines (ARM). Feature of ARM processor are:

- ARM makes 32-bit and 64-bit RISC multi-core processors.
- RISC processors are designed to perform a smaller number of types of computer instructions so that they can operate at a higher speed, performing more millions of instructions per second (MIPS).
- By stripping out unneeded instructions and optimizing pathways, RISC processors provide outstanding performance at a fraction of the power demand of CISC (complex instruction set computing) devices.
- ARM processors are extensively used in consumer electronic devices such as smart phones, tablets, multimedia players and other mobile devices, such as wearable's

- Because of their reduced instruction set, they require fewer transistors, which enable a smaller die size for the integrated circuitry (IC).
- The ARM processor's smaller size, reduced complexity and lower power consumption makes them suitable for increasingly miniaturized devices.
- Load/store architecture.
- An orthogonal instruction set.
- Mostly single-cycle execution.
- Enhanced power-saving design.
- Hardware virtualization support.

Q. 4. (b) Explain the architecture of ARM processor.

Ans. Refer Q. 3. (a) of End Term Examination 2016.

END TERM EXAMINATION [DEC. 2016]
SEVENTH SEMESTER [B.TECH]
EMBEDDED SYSTEM [ETEC–401]

Time : 3 hrs. **M.M. : 75**

Note: *Attempt any five questions including . Question No.1 which is compulsory.*

Q. 1. (a) What is meant by operating system? **(3)**

Ans. An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs. All computer programs, excluding firmware, require an operating system to function. An operating system performs these services for applications:

- In a multitasking operating system where multiple programs can be running at the same time, the operating system determines which applications should run in what order and how much time should be allowed for each application before giving another application a turn.
- It manages the sharing of internal memory among multiple applications.
- It handles input and output to and from attached hardware devices, such as hard disks, printers, and dial-up ports.
- It sends messages to each application or interactive user (or to a system operator) about the status of operation and any errors that may have occurred.
- It can offload the management of what are called *batch* jobs (for example, printing) so that the initiating application is freed from this work.
- On computers that can provide parallel processing, an operating system can manage how to divide the program so that it runs on more than one processor at a time.

Q. 1. (b) What is embedded system? What are the components of embedded system? **(4)**

Ans. An **embedded system** is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is *embedded* as part of a complete device often including hardware and mechanical parts. Embedded systems control many devices in common use today. Ninety-eight percent of all microprocessors are manufactured as components of embedded systems.

An embedded system has three main components : Hardware, Software and time operating system

(*i*) Hardware

- Power Supply
- Processor
- Memory
- Timers
- Serial communication ports
- Input/Output circuits
- System application specific circuits

(*ii*) Software: The application software is required to perform the series of tasks. An embedded system has software designed to keep in view of three constraints:

- Availability of System Memory
- Availability of processor speed.
- The need to limit power dissipation when running the system continuously in cycles of wait for events, run , stop and wake up.

***(iii)* Real Time Operating System:** (RTOS) It supervises the application software and provides a mechanism to let the processor run a process as per scheduling and do the switching from one process (task) to another process.

Q.1. (c) Name the registers used in ARM processor. **(4)**

Ans. ARM processors provide general-purpose and special-purpose registers. Some additional registers are available in privileged execution modes.

In all ARM processors, the following registers are available and accessible in any processor mode:

- 13 general-purpose registers R0-R12.
- One *Stack Pointer* (SP).
- One *Link Register* (LR).
- One *Program Counter* (PC).
- One *Application Program Status Register* (APSR).

Q. 1.(d) How is a Real time operating system uniquely different than a general purpose OS? **(3)**

Ans. RTOS: An RTOS is a multitasking OS for the applications needing meeting of time deadlines and functioning in real time constraints. Real time constraint means constraint on time interval between occurence of an event and system expected response to the event.

RTOS provides some additional services along with basic OS services. And these are:

- Process priority allocation
- Pre-emption
- Process predictability
- Memory Management
- Asynchronous I/O subsystem.

Q. 1.(e) How to evaluate operating system performance? Explain. **(4)**

Ans. Efficient SQL Server performance monitoring includes monitoring of operating system, SQL Server, and database performance. Operating system performance metrics are related to performance of disk, memory, processor, and network. Some of the most important system performance metrics are available memory, average bytes per read/write, average read/write time, disk reads/writes per second, network utilization, pages input per second, pages per second, processor queue length, and processor usage. The choice of metrics that will be used for monitoring will depend on monitoring goals and performance requirements. It's important that, a DBA is able to improve performance based on the monitoring results, but also to recognize potential issues and bottlenecks on time, and fix them before they affect the system.

Q.1.(f) What are the goals of RTOS? **(4)**

Ans. Goals of RTOS are:

- The RTOS performs few tasks, thus ensuring that the tasks will always be executed before the deadline.
- The RTOS drops or reduces certain functions when they cannot be executed within the time constraints ("load shedding").
- The RTOS monitors input consistently and in a timely manner.
- The RTOS monitors resources and can interrupt background processes as needed to ensure real-time execution
- The RTOS anticipates potential requests and frees enough of the system to allow timely reaction to the user's request

• The RTOS keeps track of how much of each resource (CPU time per timeslice, RAM, communications bandwidth, etc.) might possibly be used in the worst-case by the currently-running tasks, and refuses to accept a new task unless it "fits" in the remaining un-allocated resources.

Q.1. (g) What is multirate system? **(3)**

Ans. Multirate embedded computing systems are very common, including automobile engines, printers, and cell phones. In all these systems, certain operations must be executed periodically, and each operation is executed at its own rate.

Q.2. (a) What are the challenges of embedded system? Give the characteristics of embedded system? **(6)**

Ans. Challenges of embedded system are:

(I). Amount and type of hardware needed

- Optimizing the microprocessors, ASIPs and single purpose processors in the system
- Optimizing according to the performance, power dissipation, cost and other design metrics the system
- Optimizing hardware (memory RAM, ROM or internal and external flash or secondary memory in the system, peripherals and devices internal and external to the system, ports and buses in the system and power source or battery in the system).

(II). Optimizing the Power Dissipation

- Clock Rate Reduction
- Operating Voltage Reduction
- **Wait, Stop and Cache Disable Instructions** – Clever real- time programming. It is by using of 'Wait' and 'Stop' instructions and disabling or controlling certain units when not needed is one method of saving power during program execution
- **Process Deadlines:** Meeting the deadline of all processes in the system while keeping the memory, power dissipation, processor clock rate and cost at minimum is a challenge.
- **Flexibility and Upgradeability:** Ability to offer the different versions of a product for marketing and offering the product in advanced versions later on.
- **Reliability:** Designing reliable product by appropriate design and thorough testing, verification and validation is a challenge.

Following are some characteristics of an embedded system:

- Contains a processing engine, such as a general-purpose microprocessor
- Typically designed for a specific application or purpose
- Includes a simple (or no) user interface, such as an automotive engine ignition controller
- Often is resource-limited. For example, it might have a small memory foot-print and no hard drive
- Might have power limitations, such as a requirement to operate from batteries
- Not typically used as a general-purpose computing platform
- Generally has application software built in, not user-selected
- Ships with all intended application hardware and software pre-integrated
- Often is intended for applications without human intervention

Q. 2.(b) What is a Micro controller? What are the specifications of 8051 Micro controller? **(6.5)**

Ans. A microcontroller (or MCU for *microcontroller unit*) is a small computer on a single integrated circuit. In modern terminology, it is a system on a chip or SoC. A

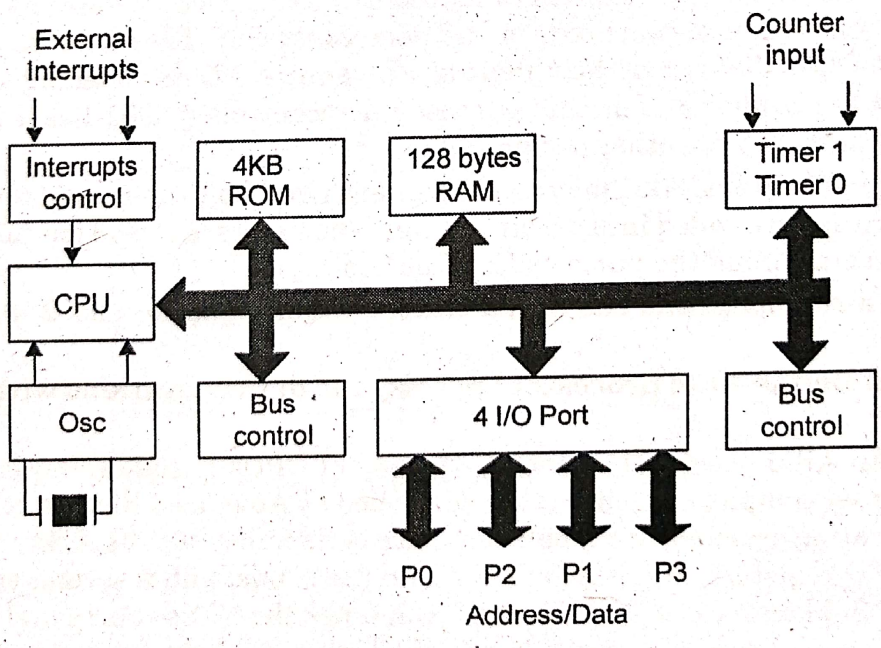
microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems.

Specification of 8051 microcontroller

- 4 KB on chip program memory.
- 128 bytes on chip data memory(RAM).
- 128 user defined software flags.
- 8-bit data bus
- 16-bit address bus
- 32 general purpose registers each of 8 bits
- 16 bit timers (usually 2, but may have more, or less).
- 3 internal and 2 external interrupts.
- Bit as well as byte addressable RAM area of 16 bytes.
- Four 8-bit ports, (short models have two 8-bit ports).
- 16-bit program counter and data pointer.
- 1 Microsecond instruction cycle with 12 MHz Crystal.

Q. 3. Describe the architecture of a typical micro controller with a neat diagram. Draw Von-neumann and Harvard architecture. **(12.5)**

Ans. Central Processor Unit (CPU): As we know that the CPU is the brain of any processing device of the microcontroller. It monitors and controls all operations that are performed on the Microcontroller units. The user has no control over the work of the CPU directly . It reads program written in ROM memory and executes them and do the expected task of that application.



Interrupts: There are 5 vectored interrupts are shown in below

- INTO
- TFO
- INT1
- TF1
- RI/TI

Memory: Microcontroller requires a program which is a collection of instructions. This program tells microcontroller to do specific tasks. These programs require a memory

on which these can be saved and read by Microcontroller to perform specific operations of a particular task. The memory which is used to store the program of the microcontroller is known as code memory or program memory of applications. It is known as ROM memory of microcontroller also requires a memory to store data or operands temporarily of the microcontroller. The data memory of the 8051 is used to store data temporarily for operation is known RAM memory. 8051 microcontroller has 4K of code memory or program memory, that has 4KB ROM and also 128 bytes of data memory of RAM.

BUS: Basically Bus is a collection of wires which work as a communication channel or medium for transfer of Data. These buses consists of 8, 16 or more wires of the microcontroller. Thus, these can carry 8 bits,16 bits simultaneously. Here two types of buses that are shown below

- Address Bus
- Data Bus

Address Bus: Microcontroller 8051 has a 16 bit address bus for transferring the data. It is used to address memory locations and to transfer the address from CPU to memory of the microcontroller. It has four addressing modes that are

- Immediate addressing modes.
- Bank address (or) Register addressing mode.
- Direct Addressing mode.
- Register indirect addressing mode.

Data Bus: Microcontroller 8051 has 8 bits of the data bus, which is used to carry data of particular applications.

Oscillator: Generally, we know that the microcontroller is a device, therefore it requires clock pulses for its operation of microcontroller applications. For this purpose, microcontroller 8051 has an on-chip oscillator which works as a clock source for Central Processing Unit of the microcontroller. The output pulses of oscillator are stable. Therefore, it enables synchronized work of all parts of the 8051 Microcontroller.

Input/Output Port: Normally microcontroller is used in embedded systems to control the operation of machines in the microcontroller. Therefore, to connect it to other machines, devices or peripherals we require I/O interfacing ports in the microcontroller interface. For this purpose microcontroller 8051 has 4 input, output ports to connect it to the other peripherals

Timers/Counters: 8051 microcontroller has two 16 bit timers and counters. These counters are again divided into a 8 bit register. The timers are used for measurement of intervals to determine the pulse width of pulses.

For Von-Neumann and Harvard architecture refer question no. 2.(b) of first term 2016.

Q. 4.(a) Define ARM processor? Specify the data operations with example in ARM. **(6)**

Ans. An ARM processor is one of a family of CPUs based on the RISC (reduced instruction set computer) architecture developed by Advanced RISC Machines (ARM). A data operation executes in a single datapath cycle unless a shift is determined by the contents of a register. A register is read onto the A bus, and a second register or the immediate field onto the B bus. The ALU combines the A bus source and the shifted B bus source according to the operation specified in the instruction, and the result, when required, is written to the destination register. An instruction prefetch occurs at the same time as the data operation, and the program counter is incremented.

When the shift length is specified by a register, an additional datapath cycle occurs during this cycle. The data operation occurs on the next cycle which is an internal cycle

that does not access memory. This internal cycle can be merged with the following sequential access by the memory manager as the address remains stable through both cycles.

The PC can be one or more of the register operands. When it is the destination, external bus activity can be affected. If the result is written to the PC, the contents of the instruction pipeline are invalidated, and the address for the next instruction prefetch is taken from the ALU rather than the address incrementer. The instruction pipeline is refilled before any further execution takes place, and during this time exceptions are ignored.

PSR transfer operations (MSR and MRS) exhibit the same timing characteristics as the data operations except that the PC is never used as a source or destination register.

The cycle timings are listed in Table where:

- pc is the address of the branch instruction
- alu is the destination address calculated by the ARM7TDMI core
- (alu) is the contents of that address.

Data operation instruction cycles

Operation type	Cycle	Address	MAS[1:0]	nRW	Data	nMREQ	SEQ	nOPC
normal	1	pc+2L pc+3L	i	0	(pc+2L)	0	1	0
dest=pc	1	pc+2L	i	0	(pc+2L)	0	0	0
	2	alu	i	0	(alu)	0	1	0
	3	alu+L alu+2L	i	0	(alu+L)	0	1	0
shift(Rs)	1	pc+2L	i	0	(pc+2L)	1	0	0
	2	pc+3L pc+3L	i	0	–	0	1	1
shift(Rs)	1	pc+8	2	0	(pc+8)	1	0	0
dest=pc	2	pc+12	2	0	–	0	0	1
	3	alu	2	0	(alu)	0	1	0
	4	alu+4 alu+8	2	0	(alu+4)	0	1	0

Q. 4.(*b*) Why do we use cache memory? Demonstrate the different types of mapping functions with necessary diagrams. **(6.5)**

Ans. A cache is a smaller, faster memory, closer to a processor core, which stores copies of the data from frequently used main memory locations. Most CPUs have different independent caches, including instruction and data caches, where the data cache is usually organized as a hierarchy of more cache levels (L1, L2, etc.).

- Three techniques can be used:

1. Direct
2. Associative
3. Set Associative.

DIRECT MAPPING

- The simplest technique, known as direct mapping, maps each block of main memory into only one possible cache line. The mapping is expressed as

i = j modulo m

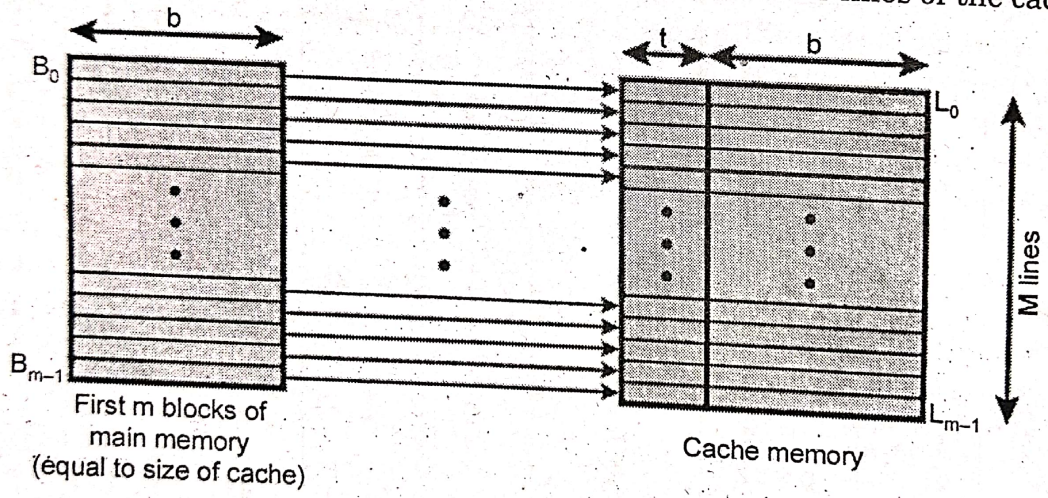
where

i cache line number

j main memory block number

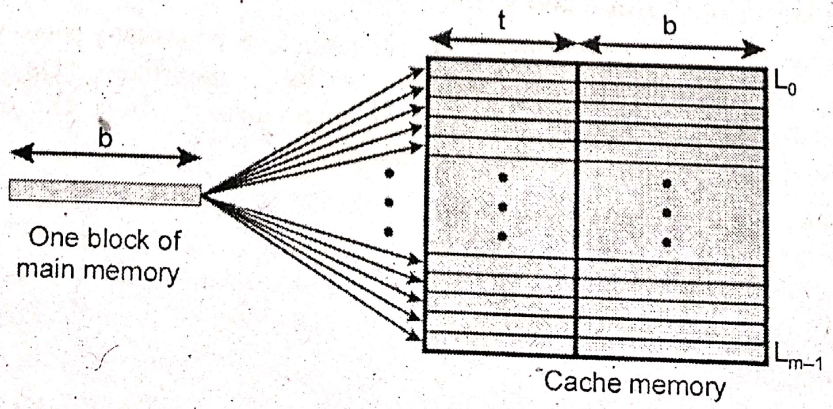
m number of lines in the cache

- Figure shows the mapping for the first m blocks of main memory. Each block of main memory maps into one unique line of the cache.
- The next m blocks of main memory map into the cache in the same fashion; that is, block Bm of main memory maps into line L0 of cache, block Bm_1 maps into line L1, and so on.
- The mapping function is easily implemented using the main memory address. Figure 2 illustrates the general mechanism. For purposes of cache access, each main memory address can be viewed as consisting of three fields.
- The least significant w bits identify a unique word or byte within a block of main memory; in most contemporary machines, the address is at the byte level. The remaining s bits specify one of the 2s blocks of main memory. The cache logic interprets these s bits as a tag of s – r bits (most significant portion) and a line field of r bits. This latter field identifies one of the m = 2r lines of the cache.



ASSOCIATIVE MAPPING

- Associative mapping overcomes the disadvantage of direct mapping by permitting each main memory block to be loaded into any line of the cache Figure.



• In this case, the cache control logic interprets a memory address simply as a Tag and a Word field. The Tag field uniquely identifies a block of main memory.

• To determine whether a block is in the cache, the cache control logic must simultaneously examine every line's tag for a match. Figure 3 illustrates the logic. Note that no field in the address corresponds to the line number, so that the number of lines in the cache is not determined by the address format.

SET-ASSOCIATIVE MAPPING

• Set-associative mapping is a compromise that exhibits the strengths of both the direct and associative approaches while reducing their disadvantages.

• In this case, the cache consists of a number sets, each of which consists of a number of lines. The relationships are

$$m = n \times k$$
$$i = j \text{ modulo n}$$

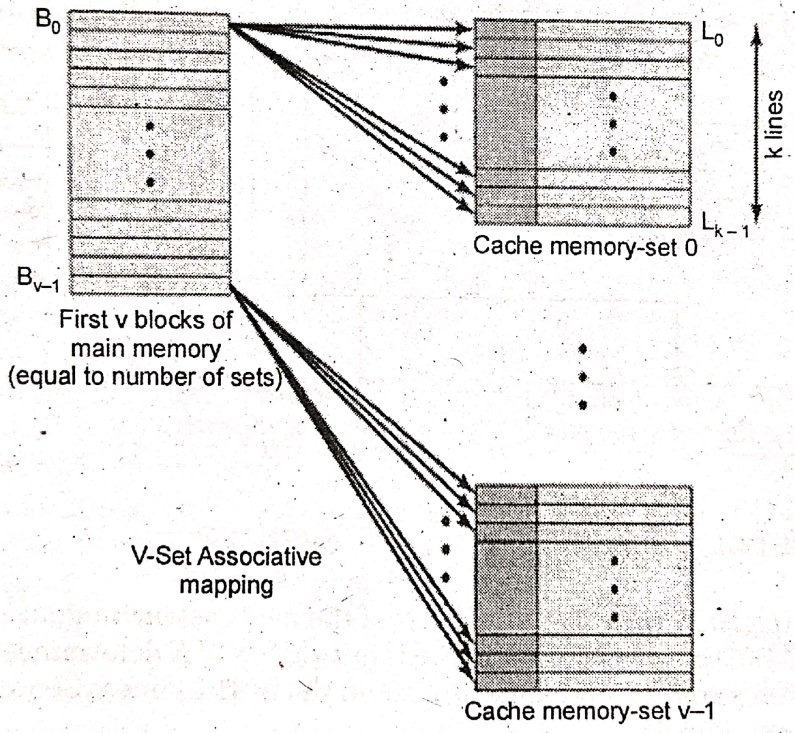
i = Cache set number

j = main memory block number

m = number of lines in the cache

v = number of sets

k = number of lines in each set



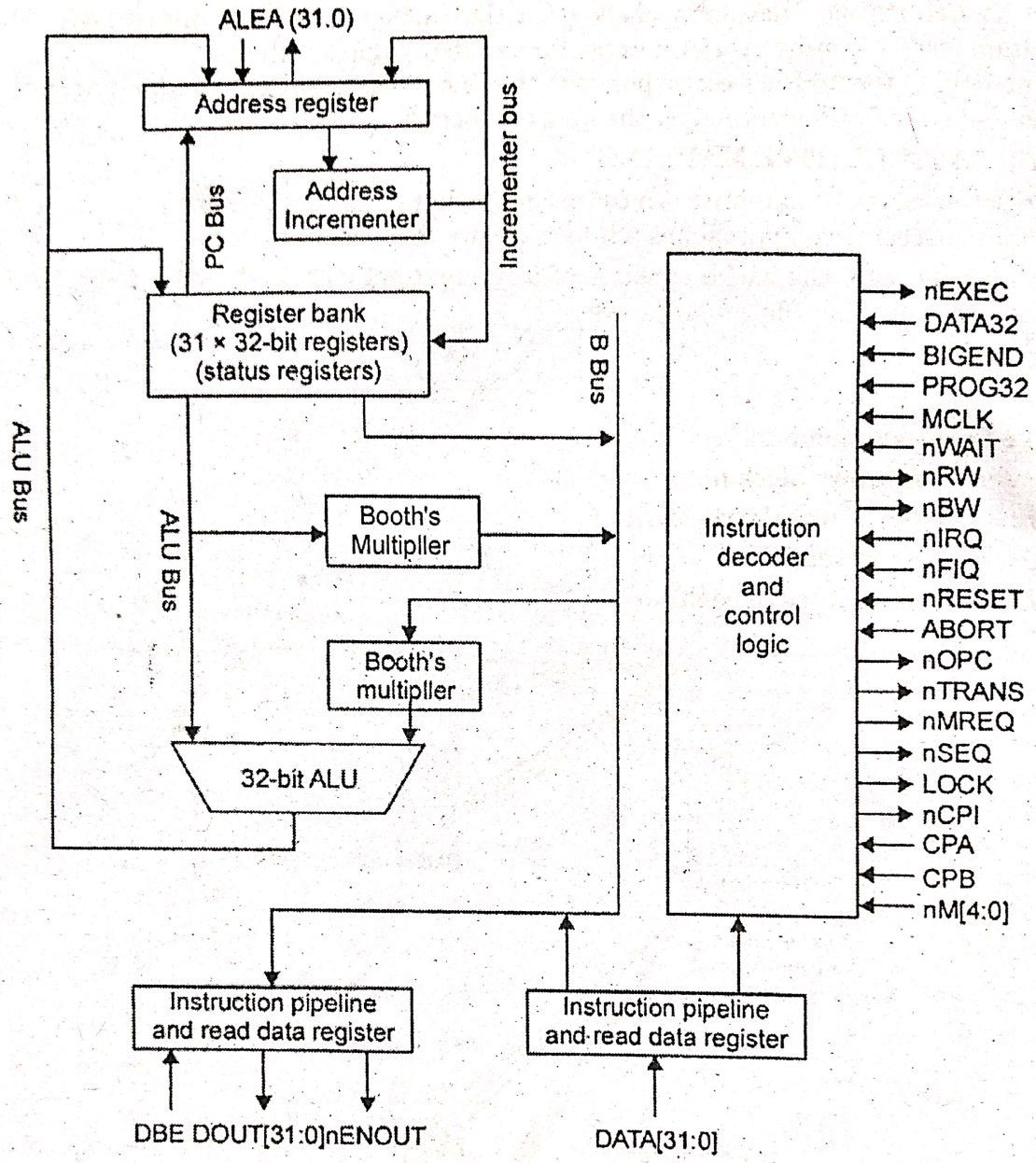
Q. 5. Name the various blocks available in ARM processor and describe in detail with necessary diagram. List out the features of ARM processor. (12.5)

Ans. Structure of ARM7: Figure 1. shows a block of ARM7 processor. The major components of ARM7 processor are described below:

1. *Instruction Pipeline and Read Data Register:* It gets the content of memory location pointed to by the address bus lines A[31:0], of Address Register. The external 32-bit data-in lines DATA[31:0] put the content into this register.

2. *Instruction Decoder and Control Logic:* It has a number of control inputs determining the operation policy of the processor. Also, it outputs *a.* number of control signals useful for interfacing the processor with other perpherals.

Structure of ARM7



3. *Address Register: It* holds the address of the next instruction/data to be fetched. Address bus A[31:0] originates from it. The input *signal ALE* determines the time upto which the register's content remain avaiable on the A[31:0] lines. Content is availabe as long as ALE remains low.

4. *Address Incremented* It increments the Address Register's value by an appropriate amount to point to the next instruction/data.

5. *Register Bank:* It contains 31,32-bit register accessible in different modes of operation of the processor, it also contains 6 status registrs, each of size 32-bits.

6. *Booth's Multiplier:* It is used in the multiplication instructions.

7. *Barrel Shifter:* One of the operands of data processing instruction can be shifted by a few bit positions. The barrel shifter located at the input of ALU performs this function.

8. *ALU:* A 32-bit ALU performs the arithmetic and logic functions.

9. *Write Data Register:* It holds the value to be written into the memory. The 32-bit value is available in the *DOUT* [31:0]

Q.6. (a) How would you explain the various bus structure used in Embedded Systems. **(6)**

Ans. Serial bus communication protocols are:

(1). I2C Bus

- The Bus has two lines that carry its signals—one line is for the clock and one is for bidirectional data.
- There is a standard protocol for the I2C bus.
- Each device has a 7-bit address using which the data transfers take place.
- Master can address 127 other slaves at an instance.
- Master has at a processing element functioning as bus controller or a microcontroller with I2C (Inter Integrated Circuit) bus interface circuit.

Disadvantage of I2C bus

- Time taken by algorithm in the hardware that analyses the bits through I2C in case the slave hardware does not provide for the hardware that supports it.
- Certain ICs support the protocol and certain do not.

(2). CAN Bus

- **CAN bus** (for **controller area network**) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other within a vehicle without a host computer.
- CAN bus is a message-based protocol, designed specifically for automotive applications but now also used in other areas such as aerospace, maritime, industrial automation and medical equipment.

Applications

(I). Automotive: The modern automobile may have as many as 70 electronic control units (ECU) for various subsystems. Typically the biggest processor is the engine control unit. Others are used for transmission, airbags, antilock braking/ABS, cruise control, electric power steering, audio systems, power windows, doors, mirror adjustment, battery and recharging systems for hybrid/electric cars, etc. Some of these form independent subsystems, but communications among others are essential. A subsystem may need to control actuators or receive feedback from sensors. The CAN standard was devised to fill this need.

(II). Industrial: Today the CAN bus is also used as a fieldbus in general automation environments, primarily due to the low cost of some CAN controllers and processors.

(3). Universal Serial Bus (USB)

• USB was designed to standardize the connection of computer peripherals (including keyboards, pointing devices, digital cameras, printers, portable media players, disk drives and network adapters) to personal computers, both to communicate and to supply electric power.

• It has become commonplace on other devices, such as smartphones, PDAs and video game consoles.

• USB has effectively replaced a variety of earlier interfaces, such as serial and parallel ports, as well as separate power chargers for portable devices.

• Variations like USB 1.X, USB 2.X, USB 3.X

Parallel bus communication protocols are:

(1). ISA (Industry Standard Architecture) Bus

• Originally introduced in the IBM PC (1981) as an 8 bit expansion slot

• Runs at 8.3 MHz with data rate of 7.9 Mbytes/s

• Runs at 15.9 MHz with data rate of 15.9 Mbytes/s

• Sometimes just called the "AT bus"

• Used for...

– Just about any peripheral (sound cards, disk drives, etc.)

• PnP ISA

– In 1993, Intel and Microsoft introduced "PnP ISA", for plug-and- play ISA

– Allows the operating system to configure expansion boards automatically

• Form factor

– Large connector in two segments.

– Smaller segment is the 8-bit interface (36 signals)

– Larger segment is for the 16-bit expansion (62 signals)

– 8-bit cards only use the smaller segment

(2). PCI (*Peripheral Component Interconnect*) Bus

• This bus is made by Intel.

• It is used today in all PCs and other computers for connecting adapters, such as network-controllers, graphics cards, sound cards etc.

• The PCI bus is the central I/O bus.

• PCI is a computer bus for attaching hardware devices in a computer.

• These devices can take either the form of an integrated circuit fitted onto the motherboard itself or an expansion card that fits into a slot.

• Typical PCI cards used in PCs include: network cards, sound cards, modems, extra ports such as USB or serial, TV tuner cards and disk controllers.

• Used for...

– Just about any peripheral

– Can support multiple high-performance devices

– Graphics, full-motion video, SCSI, local area networks, etc.

• Specifications

– 64-bit bus capability

– Usually implemented as a 32-bit bus

– Runs at 33 MHz or 66 MHz

– At 33 MHz and a 32-bit bus, data rate is 133 Mbytes/s

3. PCI-X(Peripheral Component Interconnect)

- Extended, is a computer bus and expansion card standard that enhances the 32-bit PCI Local Bus for higher bandwidth demanded by servers.
- It is a double-wide version of PCI, running at up to four times the clock speed, but is otherwise similar in electrical implementation and uses the same protocol.

Q. 6. (b) With a suitable example, explain how debugging is carried out using debuggers & compliers. (6.5)

Ans. The step of Assembly language program are outlines as follows:

1. First we use an editor to type a program, many excellent editors or word processors are available that can be used to create and/or edit the program

• Notice that the editor must be able to produce an ASCII file

• For many assemblers, the file names follow the usual DOS conventions, but the source file has the extension "asm or "src", depending on which assembly you are using

2. The "asm" source file containing the program code created in step 1 is fed to an 8051 assembler

• The assembler converts the instructions into machine code

• The assembler will produce an object file and a list file

• The extension for the object file is "obj" while the extension for the list file is "1st"

3. Assembler require a third step called *linking*

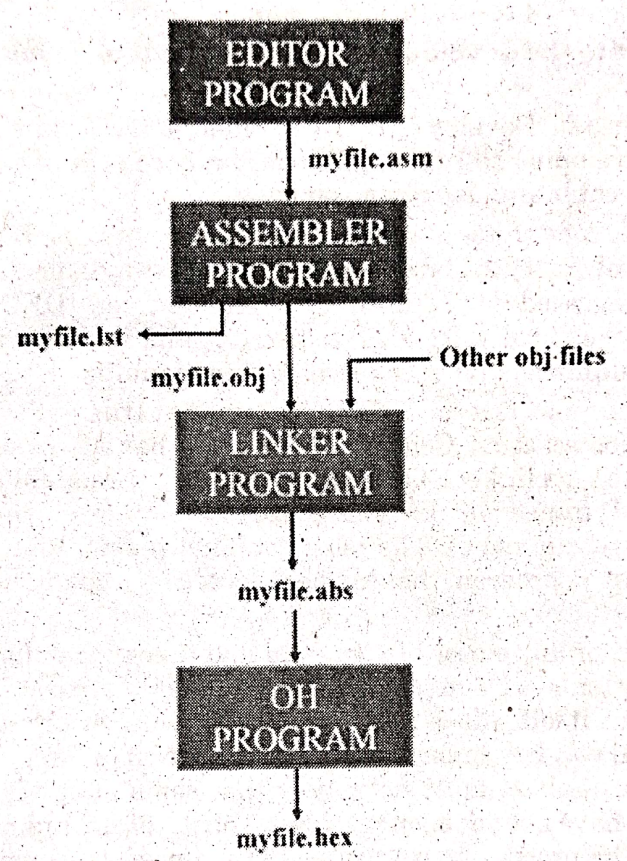
• The linker program takes one or more object code files and produce an absolute object file with the extension "abs"

• This abs file is used by 8051 trainers that have a monitor program

4. Next the "abs" file is fed into a program called "OH" (object to hex converter) which creates a file with extension "hex" that is ready to burn into ROM

• This program comes with all 8051 assemblers

• Recent Windows-based assemblers combine step 2 through 4 into one step.



Q. 7.(a) What are program Validation and testing? **(6)**

Ans. Validation is determining if the system complies with the requirements and performs functions for which it is intended and meets the organization's goals and user needs. Validation is done at the end of the development process and takes place after verifications are completed.

Advantages of Validation:

1. During verification if some defects are missed then during validation process it can be caught as failures.

2. If during verification some specification is misunderstood and development had happened then during validation process while executing that functionality the difference between the actual result and expected result can be understood.

3. Validation is done during testing like feature testing, integration testing, system testing, load testing, compatibility testing, stress testing, etc.

4. Validation helps in building the right product as per the customer's requirement and helps in satisfying their needs.

Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use. Software testing involves the execution of a software component or

system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- is sufficiently usable,
- can be installed and run in its intended environments, and
- achieves the general result its stakeholders desire.

Q. 7.(b) Illustrate the development environment of a embedded system with suitable diagram? . **(6.5)**

Ans. An Integrated Development Environment (IDE) is software that assists programmers in developing software. IDEs normally consist of a source code editor, a compiler, a linker/locater and usually a debugger.

Editor: A source code editor is a text editor program designed specifically for editing source code to control embedded systems. It may be a standalone application or it may be built into an integrated development environment (e.g. IDE). Source code editors may have features specifically designed to simplify and speed up input of source code, such as syntax highlighting and auto complete functionality.

Compiler: A compiler is a computer program that translates the source code into computer language (object code). Commonly the output has a form suitable for processing by other programs (e.g., a linker), but it may be a human-readable text file. A compiler translates source code from a high level language to a lower level language (e.g., assembly language or machine language). The most common reason for wanting to translate source code is to create a program that can be executed on a computer or on an embedded system.

Linker: A linker or link editor is a program that takes one or more objects generated by compilers and assembles them into a single executable program or a library that can later be linked to in itself. All of the object files resulting from compiling must be combined in a special way before the program can be executed. The object files themselves are individually incomplete, most notably is that some of the internal variable and function references have not yet been resolved. The job of the linker is to combine these object files and, in the process, to resolve all of the unresolved symbols.

Debugger: A debugger is a computer program that is used to test and debug other programs. The code to be examined might alternatively be running on an instruction set simulator (ISS), a technique that allows great power in its ability to halt when specific conditions are encountered but which will typically be much slower than executing the code directly on the appropriate processor. When the program crashes, the debugger shows the position in the original code if it is a source-level debugger or symbolic debugger, commonly seen in integrated development environments.

Q. 8.(a) What are the services of operating system in handling multi process scheduling and communication? Discuss. **(6.5)**

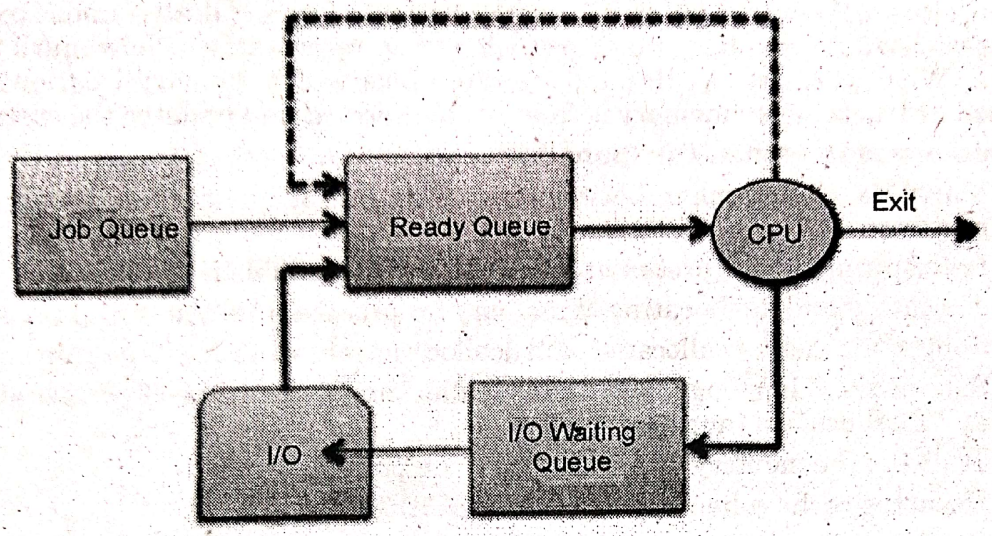
Ans. The process scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy.

Process scheduling is an essential part of a Multiprogramming operating systems. Such operating systems allow more than one process to be loaded into the executable memory at a time and the loaded process shares the CPU using time multiplexing.

The Operating System maintains the following important process scheduling queues

- **Job queue:** This queue keeps all the processes in the system.
- **Ready queue:** This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.

Device queues " The processes which are blocked due to unavailability of an I/O device constitute this queue.



Q.8. (b) What are the types of Scheduling? Discuss earliest deadline first (EDF) scheduling. **(6)**

Ans.

- Cooperative Scheduling of ready tasks in a circular queue. It closely relates to function queue scheduling.
- Cooperative Scheduling with Precedence Constraints
- Cyclic scheduling of periodic tasks and Round Robin Time Slicing Scheduling of equal priority tasks
- Preemptive Scheduling
- Scheduling using Earliest Deadline First (EDF) precedence.
- Rate Monotonic Scheduling using higher rate of events occurrence First precedence
- Fixed Times Scheduling
- Scheduling of Periodic, sporadic and aperiodic Tasks
- Advanced scheduling algorithms using the probabilistic Timed Petri nets (Stochastic) or Multi Thread Graph for the multiprocessors and complex distributed systems.

Earlies Deadline First Scheduling

- When a task becomes ready, its will be considered at a scheduling point.
- The scheduler does not assign any priority.
- It computes the deadline left at a scheduling point.
- Scheduling point is an instance at which the scheduler blocks the running task and re-computes the deadlines and runs the EDF algorithm and finds the task which is to be run

  An EDF algorithm can also maintain a priority queue based on the computation when the new task inserts

Precedence Assignment in the Scheduling Algorithms:

- Best strategy is one, which is based on EDF (Earliest Deadline First) precedence.
- Precedence is made the highest for a task that corresponds to an interrupt source, which occurs at the earliest a succeeding times, and which deadline will finish earliest at the earliest
- We assign precedence by appropriate strategy in the case of the variable CPU loads for the different tasks and variable EDFs.

Q. 9.(a) Explain process management and memory management in embedded system. **(6)**

Ans. The memory management function keeps track of the status of each memory location, either *allocated* or *free*. It determines how memory is allocated among competing processes, deciding which gets memory, when they receive it, and how much they are allowed. When memory is allocated it determines which memory locations will be assigned. It tracks when memory is freed or *unallocated* and updates the status.

Memory management responsibilities include:

- Managing the mapping between logical (physical) memory and task memory references.
- Determining which processes to load into the available memory space.
- Allocating and deallocating of memory for processes that make up the system.
- Supporting memory allocation and deallocation of code requests (within a process),
- Such as the C language "alloc" and "dealloc" functions, or specific buffer allocation and Deallocation routines.
- Tracking the memory usage of system components.
- Ensuring cache coherency (for systems with cache).
- Ensuring process memory protection.

Process management is an integral part of any modern-day operating system(OS). The OS must allocate resources to processes, enable processes to share and exchange information, protect the resources of each process from other processes and enable synchronization among processes. To meet these requirements, the OS must maintain a data structure for each process, which describes the state and resource ownership of that process, and which enables the OS to exert control over each process.

Q. 9.(b) Define Process? What are the states of a process. **(6.5)**

Ans. A process is basically a program in execution. The execution of a process must progress in a sequential fashion.

In general, a process can have one of the following five states at a time.

S.N.	State & Description
1.	**Start:** This is the initial state when a process is first started/created.
2.	**Ready:** The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run. Process may come into this state after start state or while running it by but interrupted by the scheduler to assign CPU to some other process.
3.	**Running:** Once the process has been assigned to a processor by the OS scheduler, the process state is set to running and the processor executes its instructions.
4.	**Waiting:** Process moves into the waiting state if it needs to wait for a resource, such as waiting for user input, or waiting for a file to become available.
5.	**Terminated or Exit:** Once the process finishes its execution, or it is terminated by the operating system, it is moved to the terminated state where it waits to be removed from main memory.