

**K.S.R. COLLEGE OF ENGINEERING (Autonomous), TIRUCHENGODE - 637 215****DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING****COURSE / LESSON PLAN SCHEDULE**

IV YEAR (EEE)

**Subject : 12EE712 - Embedded Systems      Year/ Sem : IV/VII****Regulation : 2016****Staff Name: M.VIJAYAKUMAR**

<b>Lecture</b>	<b>Topic</b>	<b>Hours</b>	<b>Teaching Aid</b>	<b>Source</b>	<b>Page No</b>
<b>UNIT – I INTRODUCTION TO EMBEDDED SYSTEMS</b>					
L01	Definition of embedded systems	1	BB	TB1	3
L02	Embedded system Vs General computing system	1	BB	TB1	4
L03	Classification of embedded systems	1	BB	TB1	52-53
L04	Functional building blocks of embedded systems	1	BB	TB1	5
L05	Processor in the system	1	BB	TB1	5-10
L06	Embedded hardware units	1	BB	TB1	10-18
L07	PIC 16F877A- Register, memory devices,	1	BB	TB2	35-38 28-32
L08	ports, timer	1	BB	TB2	139, 114-116
L09	Instruction Set and Addressing Modes	1	BB	TB2	38-40 32-35
<b>TOTAL</b>		<b>09</b>			
<b>UNIT - II PROCESSOR AND MEMORY ORGANIZATION</b>					
L10 , L11	Structural units in a processor	1	BB	TB1	84-88
L12, L13	Selection of processor	1	BB	TB1	113-117
L14	Memory selection	1	BB	TB1	118-119
L15	Processor interfacing with memory and I/O units	1	BB	TB1	72-75
L16	Shared Memory	1	BB	TB1	326-330
L17	DMA	1	BB	TB1	218-219
L18	memory management	1	BB	TB1	359-361
<b>TOTAL</b>		<b>09</b>			

UNIT - III EMBEDDED NETWORKS					
L19	Serial communication using I <sup>2</sup> C	1	BB	TB1	160-161
L20	Serial communication using CAN, SPI,	1	BB	TB1	161-163
L21	Serial communication using USB & PROFI buses	1	BB	TB1	163-165
L22	Parallel communication using PCI,PCI-X buses	1	BB	TB1	167-169
L23	Parallel communication using Arm bus	1	BB	TB1	169-170
L24	Networks protocols introduction: HTTP	1	BB	TB1	170-173
L25	TCP	1	BB	TB1	173
L26	UDP	1	BB	TB1	174
L27	IP	1	BB	TB1	174-175
TOTAL		09			
UNIT - IV I/O PROGRAMMING SCHEDULE MECHANISM					
L28	Transfer rate, latency, interrupt driven I/O				211-214
L29	Non-maskable interrupts, software interrupts				207,195
L30	writing interrupt service routine in C & assembly languages				192-198
L31	preventing interrupt overrun, disability interrupts				207-211
L32	Multi threaded programming ,Context switching				306-308 211-212
L33	premature & non premature multitasking				392-393
L34	Scheduling - Thread states				385-389
L35	round robin scheduling and priority based scheduling				385-389
L36	assigning priorities, Deadlock, watch dog timers				209-211 329-330 157
TOTAL		09			
UNIT - V REAL TIME OPERATING SYSTEM (RTOS)					
L37	Introduction	1	BB	TB1	352-354
L38	Basic function of kernel	1	BB	TB1	352-354
L39	Basic function of RTOS	1	BB	TB1	370-371
L40	OS Vs RTOS	1	BB	TB1	354, 371
L41	Types of RTOS	1	BB	TB1	408-410
L42	Interrupt routines in RTOS	1	BB	TB1	366-370

L43	Embedded system design process	1	BB	TB1	37-42
L44,L45	Embedded software development process and tools.	1	BB	TB1	620-643
		<b>09</b>			

## TEACHING AID

1. BB : Black Board
2. OHP : Overhead Projector
3. LCD : LCD Projector
4. MM : Multimedia

## Text Books:

### Text Books :

1. Rajkamal.P Embedded System – Architecture, Programming, Design, Tata McGraw Hill, Third Edition, 2016.
2. John B. Postman, Design with PIC Microcontrollers, McGraw Hill International Limited, Singapore, 2009.

### Reference Books :

1. Frank Vahid and Tony Givargi, Embedded System Design – A Unified Hardware & Software Introduction, John Wiley, 2011.
2. Steve Furbe, ARM System-on-chip Architecture, Addison-Wesley Professional, Second Edition, 2000.
3. Steve Heath, Embedded System Design, Second Edition, Elsevier, 2003.
4. Wayne wolf, Computers as components: Principles of embedded computing system design, Morgan Kaufmann publishers, Third Edition, 2012.

**Staff in charge**  
(M.Vijayakumar)

**HOD-EEE**

**UNIT-I (CO1)**  
**INTRODUCTION TO EMBEDDED SYSTEM**

**1. Define a system. (Remembering)**

A system is a way of working, organizing or doing one or many tasks according to a fixed plan, program, or set of rules.

**2. Define an Embedded System. (Remembering)**

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

**3. Name the hardware components of an embedded system. (Remembering)**

Processor  
Program memory & Data memory  
Input & output ports  
Timer  
Power supply, Clock oscillator circuit

**4. Name the different types processor used in an embedded system. (Remembering)**

General Purpose Processor (GPP)  
Microprocessor, Microcontroller, Embedded processor, Digital signal processor,  
Media processor  
Application Specific Instruction-set Processor (ASIP)  
Single Purpose Processor  
Application Specific System Processor (ASSP)  
Multiprocessor.

**5. What is General Purpose Processor (GPP)? Give example. (Remembering)**

A processor from a number of families of processors, microcontrollers, embedded processors and digital signal processors (DSPs) having a general- purpose instruction set and readily available compilers to enable programming in a high level language.  
Example: Microcontroller, Microprocessor.

**6. What is Application Specific Instruction-set Processor (ASIP)? (Remembering)**

A processor with an instruction set designed for specific application on a VLSI chip, for example Microcontroller, DSP etc.

**7. What is Application Specific System Processor (ASSP)? (Remembering)**

A processing unit for system specific tasks, for example, image processing, compression and decompression etc.

**8. Give the classification of embedded systems. (Remembering)**

Small scale embedded systems, Medium scale embedded systems,  
Sophisticated embedded systems

**9. Give example for embedded systems. (Remembering)**

Washing or cooking system, Multitasking toys, Keyboard controller, Fax or Photocopy or printer or scanner machine, Mobile smart phones, Video games

**10. What is System-on-Chip (SOC)? (Remembering)**

Embedded systems are being designed on a single silicon chip called System on chip. SOC is a new design innovation for embedded system Eg. Mobile phone.

**11. What is small scale embedded systems? Give example. (Remembering)**

Small scale embedded systems are designed with a single 8-bit or 16-bit microcontroller, they have little hardware and software complexities, involves board level design.

E.g. Stepper motor controllers for a robotic system

Washing or cooking system and Multitasking toys

**12. What is medium scale embedded systems? Give example. (Remembering)**

Medium scale embedded systems are designed with a single or a few 16-bit or 32-bit microcontrollers, DSPs or RISCs.

E.g. Router, a hub and a gateway, Entertainment systems

**13. What is a sophisticated embedded system? Give example. (Remembering)**

Sophisticated embedded systems having enormous hardware and software complexities and may need several IPs, ASIPs, scalable processors or configurable processors and programmable logic arrays. Example: Embedded system for wireless LAN

Embedded systems for real time video

**14. List out the various types of memory used in embedded systems? (Remembering)**

RAM (internal External)

ROM/PROM/EEPROM/Flash

Cache memory

**15. List out the uses of Real Time Clock. (Remembering)**

Schedulers, Real time programming, To obtain the timing and counting needs of the system, To obtain Software controlled delays and time outs.

**16. What is watch dog timer? (Remembering)**

A timer that resets the processor in case the program gets stuck for an unexpected length of time.

**17. State the purpose of Input Output (IO) ports. (Remembering)**

The system gets the inputs and outputs from these. Through these, the keyboard or LCD units or touch-screen or peripherals and external systems connect to the system.

**18. What is interrupt controller? (Remembering)**

A unit that controls (handles) processor operations arising out of an interrupt from a source.

**19. What is CISC processor? (Remembering)**

Complex instruction-set computing processors are usually represented as CISC processor, example 8085 microprocessor.

**20. What is RISC processor? (Remembering)**

Reduced instruction-set computing processor is usually represented as RISC processor, example 16F877 microcontroller.

**21. What does the execution unit of a processor in an embedded system do? (Remembering)**

The EU includes the ALU and also the circuits that execute instructions for a program control task. The EU has circuits that implement the instructions pertaining to data transfer operations and data conversion from one form to another.

**22. What are the important embedded processor chips? (Remembering)**

ARM 7 and ARM 9  
i 960  
AMD 29050

**23. Name some DSP used in embedded systems? (Remembering)**

TMS320Cxx, SHARC, 5600xx

**24. Define ROM image. (Remembering)**

Final stage software is also called as ROM image .The final implement able software for a product embeds in the ROM as an image at a frame. Bytes at each address must be defined for creating the image.

**25. Define device driver. (Remembering)**

A device driver is software for controlling, receiving and sending byte or a stream of bytes from or to a device.

**26. Name some of the software's used for the detailed designing of an embedded system. (Remembering)**

Final machine implement able software for a product  
Assembly language  
High level language  
Machine codes  
Software for device drivers and device management.

**27. What are the various models used in the design of an embedded system? (Remembering)**

Finite state machine  
Petri net  
Control and dataflow graph  
Activity diagram based UML model  
Synchronous data flow graph  
Timed Petri net and extended predicate/transition net  
Multithreaded graph

**28. What are the requirements of embedded system? (Remembering)**

Reliability, Low power consumption, Cost effectiveness, Efficient use of processing power

**29. What are the design metrics? (Remembering)**

Power, Size, NRE cost, Performance

**30. What are the challenges of embedded systems? (Remembering)**

Hardware needed

Meeting the deadlines

Minimizing the power consumption

Design for upgradeability

**31. Give the steps in embedded system design? (Remembering)**

Requirements, Specifications, Architecture, Components, System integration

**32. What are the requirements? (Remembering)**

Before designing a system, it must to understand what has to be designed. This can be known from the starting steps of a design process.

**33. Give the types of requirements? (Remembering)**

Functional requirements

Non functional requirements

**34. Define functional requirements? (Remembering)**

It says the fundamental functions of an embedded system.

**35. Give some examples of functional requirements? (Remembering)**

Performance, Cost, Physical size and weight, Power

**36. What is the use of requirements form? (Remembering)**

It is used as a checklist in the requirements analysis. From this the fundamental properties of a system came to be known.

**37. What are the entries of a requirement form? (Remembering)**

Name, Purpose, Inputs and outputs, Functions, Performance

Manufacturing cost, Power, Physical size and weight

**38. What is meant by specification? (Remembering)**

This is a bridge between Customer and Architect. It conveys the customer's needs. These needs are properly used in the design process.

**39. What is architecture design? (Remembering)**

It says the way of implementing functions by a system. Actually architecture is a plan for whole structure of a system. While will bring the design of components later.

**40. Define system integration? (Remembering)**

It is a processor of combining the components into one system.

**41. What are the functions of memory? (Remembering)**

The memory functions are

To provide storage for the software that it will run. To store program variables and the intermediate results

Used for storage of information

## UNIT II (CO2)

### PROCESSOR AND MEMORY ORGANIZATION

**1. Name the essential units of a processor and state their functions. (Remembering)**

- i. Program flow control unit (CU).
- ii. Execution unit (EU)

The CU includes a fetch unit for fetching instruction from the memory.

The EU has circuits that implement the instructions pertaining to data transfer operations and data conversion from one from to another. The EU includes the Arithmetic and Logical Unit (ALU) and the circuits that execute instructions for a program control task.

**2. Name the various structural units of a processor. (Remembering)**

- Arithmetic and logical subunit
- Processing unit
- Basic units
- Instruction dispatch
- Control register
- Pre-fetch unit.

**3. State the function of Arithmetic and logical unit. (Remembering)**

To execute arithmetical or logical instruction according to current instruction fetched at IR.

**4. What is pipelined architecture? (Remembering)**

Pipelining is an implementation technique where multiple instructions are overlapped in execution. The computer pipeline is divided in stages. Each stage completes a part of an instruction in parallel.

**5. What is superscalar architecture? (Remembering)**

In super scalar architecture the performance of the processor enhanced by executing two or more instruction in parallel with help of parallel execution units.

**6. What is Flash memory? (Remembering)**

Flash memory is a form of EEPROM in which a sector of bytes can be erased in flash (very short duration corresponding to a signal clock cycle).

**7. List out a few important parameters while selecting a processor. (Remembering)**

- i. clock speed
- ii. execution of number of instruction per second
- iii. type of processor architecture (Pipeline and/or superscalar architecture)
- iv. memory size (internal and external)

**8. What is MIPS? (Remembering)**

“Million instruction set per second”. This system performance index shows that how long a system takes to execute the desired system function.

**9. What is MFLOPS? (Remembering)**

Million floating point instructions per second. This system performance index shows that how long a system takes to execute the desired system function.



**10. Distinguish parallel and serial port.(Understanding)**

A port for read and write operations on multiple bits at an instance is a parallel port. A port read and writes operations with one bit at an instance and where each bit of the message is separated by constant time interval.

**11. Difference between memory mapped I/O and peripheral I/O? (Understanding)**

MEMORY MAPPED I/O	PERIPHERAL I/O
16-bit device address	8-bit device address
The data transfer between any general-purpose register and I/O port	The data transfer only between accumulator and I/O port
The memory map (64kb) is shared between I/O Device and system memory	The I/O map is independent of the memory
More hardware is required to decode 16-bit address	Less hardware is required to decode 8-bit address

**12. Why interfacing is needed for I/O devices? (Remembering)**

Generally I/O devices are slow devices. Therefore the speed of I/O devices does not match with the speed of microprocessor. And so an interface is provided between system bus and I/O devices.

**13. What is the difference between CPU bus and system bus? (Remembering)**

The CPU bus has multiplexed lines but the system bus has separate lines for each signal. (The multiplexed CPU lines are demultiplexed by the CPU interface circuit to form system bus).

**14. Write the various form of memories present in a system.(Remembering)**

**Various forms of memories are**

- RAM (internal External)
- ROM
- PROM
- EEPROM
- Flash
- Cache memory
- EEPROM or flash
- External ROM or PROM for embedding software
- RAM Memory buffers
- Caches (in superscalar microprocessors)

**15. What does memory-mapping mean? (Remembering)**

The memory mapping is the process of interfacing memories to microprocessor and allocating addresses to each memory locations.

**16. Why EPROM is mapped at the beginning of memory space in 8085 system? (Remembering)**

In 8085 microprocessor, after a reset, the program counter will have 0000H address. If the monitor program is stored from this address then after a reset, it will be executed

automatically. The monitor program is a permanent program and stored in EPROM memory. If EPROM memory is mapped at the beginning of memory space, i.e., at 0000H, then the monitor program will be executed automatically after a reset.

**17. Mention the features of RISC which are used and rejected in ARM processors. (Understanding)**

**Features used:**

1. Load store architecture
2. Fixed-length 32-bit instructions.
- 3-address instruction formats.

**Features rejected:**

1. Register windows
2. Delayed branches
3. Single cycle execution of all instructions.

**18. Explain the ARM CPSR format. (Understanding)**

CPSR—Current Program Status Register is used to store the status bits.



- Mode [4:0] (lower 5 bits) – represents the processor operating mode.
- T – 5th bit: represents currently ARM or Thumb instruction is executed.
- I, F – 6th, 7th bits: interrupt flag and fast interrupt flag.
- N, Z, C, V – Negative, Zero, Carry, overflow flags.

**19. How data items are arranged in memory system? (Remembering)**

Memory may be viewed as a linear array of bytes numbered from zero up to  $2^{32}-1$ . Data items may be 8-bit bytes, 16-bit half-words or 32-bit words. A word-sized data item must occupy a group of four byte locations starting at a byte address which is a multiple of four. Half-words occupy two byte locations starting at an even byte address.

**20. Define Load-store architecture. (Remembering)**

This means that the instruction set will only process (add, subtract, and soon) values which are in registers (or specified directly within the instruction itself), and will always place the results of such processing into a register. The only operations which apply to memory state are ones which copy memory values into registers (Load instructions) or copy register values into memory (store instructions).

**UNIT-III (CO3)**  
**EMBEDDED NETWORKS**

**1. What are the modes serial communications? (Remembering)**

Synchronous communication  
Asynchronous communication

**2. Give the general classifications of I/O devices? (Remembering)**

- i. Synchronous serial input and output
- ii. Asynchronous serial (UART) input and output
- iii. Parallel one bit input and output
- iv. Parallel port input and output.

**3. What is meant by serial port? (Remembering)**

A serial port is a port for serial communication. Serial communication means that over a given line or channel one bit can communicate and the bits transmit at periodic intervals generated by a clock.

**4. Distinguish control register and status register of IO port. (Understanding)**

A register bit(s) that controls the actions of a device is represented control register. It is for a write operation only.

A register bit(s) that reflects the current status at the port buffer is represented statue register. It is for a read operation only.

**5. State the purpose parallel port. (Remembering)**

A parallel port is a port for parallel communication. Parallel communication means that multiple bits can communicate over a set of parallel lines at any given instance.

**6. What is bus? (Remembering)**

In a computer or on a network, a bus is a transmission path on which signals are dropped off or picked up at every device attached to the line. Only devices addressed by the signals pay attention to them; the others discard the signals.

**7. What is meant by UART? (Remembering)**

UART stands for universal Asynchronous Receiver/Transmitter.

UART is a hardware component for translating the data between parallel and serial interfaces.

ART does convert bytes of data to and from asynchronous start stop bit.

UART is normally used in MODEM.

**8. What does UART contains? (Remembering)**

A clock generator. Input and Output control Registers, Buffers and Transmitter/Receiver control.

**9. Distinguish half-duplex and full duplex communication. (Understanding)**

In half-duplex communication, transmission occurs in both the direction, but not simultaneously. In full-duplex communication, transmission occurs in both the direction, simultaneously.

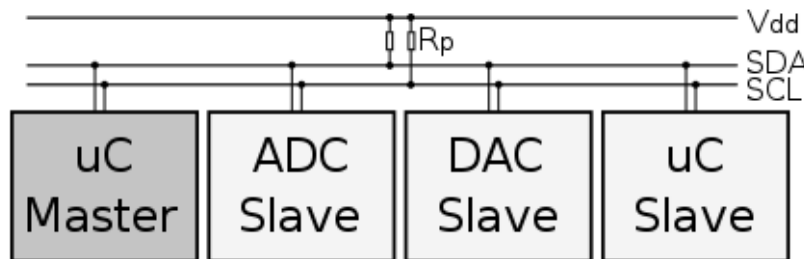
**10. What is I<sup>2</sup>C bus? (Remembering)**

Inter-Integrated Circuit (I<sup>2</sup>C), generically referred to as "two-wire interface serial bus for interconnecting ICs. It has a start bit and a stop bit like an UART. It has seven fields for start, 7 bit address, defining a read or a write, defining byte as acknowledging byte, data byte, NACK and end.

**11. State the special features on I<sup>2</sup>C? (Remembering)**

- Low cost
- Easy implementation
- Moderate speed (up to 100 kbps).

**12. Draw a typical schematic diagram of I<sup>2</sup>C bus. (Remembering)**



One master (a microcontroller), three slave nodes (an ADC, a DAC, and a microcontroller), and pull-up resistors R<sub>p</sub>.

**13. What is CAN bus? (Remembering)**

Controller-area network (CAN or CAN-bus) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other within a vehicle without a host computer.

**14. State the advantages of CAN bus. (Remembering)**

The wiring complexity is low. This factor plays an especially large role in automobiles. An economical and easy to manage twisted wire pair serves as the transmission medium. CAN stations can be subsequently added to and removed from the existing CAN bus relatively easily.

**15. State the disadvantages of CAN bus. (Remembering)**

The bus line cannot be designed arbitrarily long since the electrical properties (e.g. signal reflections) set physical limits in combination with the transmission speed. The same is also valid for the branch lines to the control units in the automobile. Depending on the transmission speed they may not exceed a certain length. In order to optimise the signal quality the ends of the bus line must be "terminated" with load resistors.

**16. What is USB bus? (Remembering)**

USB is a serial bus for interconnecting a system. It attaches and detaches a device from the network. It uses a root hub. Nodes containing the devices can be organized like a tree structure. It is mostly used in networking the IO devices like scanner in a computer system.

**17. What are the features of the USB protocol? (Remembering)**

A device can be attached, configured and used, reset, reconfigured and used, share the bandwidth with other devices detached and reattached.

**18. Draw the data frame format of CAN? (Remembering)**

Start	Arbitration field	Control field	Data field	CRC field	Acknowledgement field	End of frame
1	12	6	0-64	16	2	7

**19. What is meant by ISA? (Remembering)**

ISA stands for Industry Standard Architecture. Used for connecting devices following IO addresses and interrupts vectors as per IBM pc architecture.

**20. Name the four types of data transfer used in USB? (Remembering)**

Controlled transfer  
Bulk transfer  
Interrupt driven data transfer  
Is synchronous transfer

**21. What is PCI bus? (Remembering)**

Peripheral Component Interconnect (PCI) is a computer bus for attaching hardware devices in a computer.

**22. What is the difference between ISA and PCI? (Remembering)**

Industry Standard Architecture (ISA) is a data transfer standard introduced in the IBM PC in 1982. An ISA bus, or data path way, operates at 8 MHz clock rate and has a maximum data rate of 8 MBps. In newer systems ISA buses are used only for slower devices. Peripheral Component Interconnect (PCI) was introduced in 1992. It operates at clock speeds of 33 or 66

MHz At 32 bits and 33 MHz, a PCI bus has a maximum data rate of 132 MBps. Sockets, slots, for PCI cards are shorter than those for ISA plug-in cards.

**23. Distinguish PCI and PCI-X bus.(Remembering)**

PCI supports 33MHz, 66MHz clock speeds and operates in 64/32bit mode. PCI supports 5V as well as 3.3 volt signaling. PCI-X supports 66MHz, 133MHz, 266MHz, and 533MHz clock speeds and is 64-bits wide. PCI-X 66MHz, 133MHz, 266MHz and 533MHz clock speeds provide the following throughput respectively: 4Gbit/sec, 8Gbit/sec, 17Gbit/sec, 34Gbit/sec.

**24. What is ARM bus? (Remembering)**

The Advanced Microcontroller Bus Architecture (AMBA) was introduced by ARM Ltd in 1996 and used as the on-chip bus in system-on-a-chip (SoC) designs. The first AMBA buses were Advanced System Bus (ASB) and Advanced Peripheral Bus (APB).

**25. What is device driver? (Remembering)**

In computing, a **device driver** or **software driver** is a computer program allowing higher-level computer programs to interact with a hardware device.

**26. What do you mean by bus Arbitration? (Remembering)**

In single bus architecture when more than one device requests the bus, a controller called bus Arbitrator decides who gets the bus, this is called the bus arbitration. Arbitration is mostly done in favour of a master microprocessor with the highest priority.

**27. Give the summary of I/O devices used in embedded system.(Remembering)**

Program, data and stack memories occupy the same memory space. The total addressable memory size is 64 KB. Program memory program can be located anywhere in memory. Jump, branch and call instructions use 16bit addresses, i.e. they can be used to jump/branch anywhere within 64 KB. All jump/branch instructions use absolute addressing. Data memory the processor always uses 16bit addresses so that data can be placed anywhere. Stack memory is limited only by the size of memory. Stack grows downward. First 64 bytes in a zero memory page should be reserved for vectors used by RST instructions. I/O ports 256 Input ports 256 Output ports Registers Accumulator or A register is an 8bit register used for arithmetic, logic, I/O and load/store operations.

**28. What is Protocol? (Remembering)**

A protocol is a set of rules that governs data communication; the key elements of a protocol are syntax, semantics, and timing.

**29. What is HTTP? (Remembering)**

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. The protocol transfers data in the form of plain text, hyper text, audio, video and so on.

**30. What is UDP? (Remembering)**

The User Datagram Protocol (UDP) is one of the core members of the Internet protocol. UDP uses a simple connectionless transmission model with a minimum of protocol mechanism. It has no handshaking dialogues, and thus exposes any unreliability of the underlying network protocol to the user's program. There is no guarantee of delivery, ordering, or duplicate protection. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram.

**31. What is TCP/IP? (Remembering)**

The Transmission Control Protocol (TCP) is a core protocol of the Internet Protocol. It originated in the initial network implementation in which it complemented the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as TCP/IP. The Transmission Control Protocol provides a communication service at an intermediate level between an application program and the Internet Protocol. It provides host-to-host connectivity at the Transport Layer of the Internet model. An application does not need to know the particular mechanisms for sending data via a link to another host, such as the required packet fragmentation on the transmission medium. At the transport layer, the protocol handles all handshaking and transmission details and presents an abstraction of the network connection to the application.

**32. List the features of HTTP? (Remembering)**

Persistent Connection and Non persistent Connection.

**33. Explain the three types of addresses in TCP/IP? (Remembering)**

Three types of addresses are used by systems using the TCP/IP protocol: the physical address, the internet work address (IP address), and the port address. The physical address also known as the link address, is the address of a node as defined by its LAN or WAN. The IP address uniquely defines a host on the Internet. The port address identifies a process on a host.

**34. What is IP address? (Remembering)**

The Internet address (or IP address) is 32 bits (for IPv4) that uniquely and universally defines a host or router on the Internet. The portion of the IP address that identifies the host or router on the network is called the host id. An IP address defines a device's connection to a network.

**UNIT IV (CO4)****I/O PROGRAMMING SCHEDULE MECHANISM****1. Define scheduling. (Remembering)**

This is defined as a process of selection which says that a process has the right to use the processor at given time.

**2. What is scheduling policy? (Remembering)**

It says the way in which processes are chosen to get promotion from ready state to running state.

**3. What is schedulability? (Remembering)**

It indicates any execution schedule is there for a collection of process in the system's functionality.

**4. What are the types of scheduling? (Remembering)**

- Clock Driven Scheduling
- Weighted Round Robin Scheduling
- Priority Scheduling
  - Earliest Deadline First (EDF)

- Least Slack Time First (LSF)
- Rate Monotonic Scheduling (RMS)

**5. Define round robin scheduling? (Remembering)**

This type of scheduling also employs the hyper period as an interval. The processes are run in the given order.

**6. What is scheduling overhead? (Remembering)**

It is defined as time of execution needed to select the next execution process.

**7. What is meant by context switching? (Remembering)**

The actual process of changing from one task to another is called a context switch.

**8. Define priority scheduling? (Remembering)**

A simple scheduler maintains a priority queue of processes that are in there unable state.

**9. What is rate monotonic scheduling? (Remembering)**

Rate monotonic scheduling is an approach that is used to assign task priority for a preemptive system.

**10. What is critical instant? (Remembering)**

It is the situation in which the processor task posses' highest response time.

**11. What is critical instant analysis? (Remembering)**

It is used to know about the schedule of a system. It says that based on the periods given, the priorities to the processes has to be assigned.

**12. Define earliest deadline first scheduling? (Remembering)**

Earliest deadline first (EDF) is a scheduling policy which uses a dynamic priority scheme it changes process priorities during execution based on initiation times. As a result, it can achieve higher CPU utilizations than RMS. The EDF policy is very simple: It assigns priorities in order of deadline. The highest-priority process is the one whose deadline is nearest in time, and the lowest priority process is the one whose deadline is farthest away.

**13. Define task and Task state. (Remembering)**

A task is a set of computations or actions that processes on a CPU under the control of as scheduling kernel. It also has a process control structure called a task control block that saves at the memory. It has a unique ID. It has states in the system as follows: idle, ready, running, blocked and finished.

**14. Define Task Control Block (TCB) (Remembering)**

A memory block that holds information of program counter, memory map, the signal dispatch table, signal mask, task ID, CPU state and a kernel stack.



**15. What is preemptive and non-preemptive scheduling? (Remembering)**

- Under non-preemptive scheduling once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either by terminating or switching to the waiting state.
- Preemptive scheduling can preempt a process which is utilizing the CPU in between its execution and give the CPU to another process.

**16. What is a Dispatcher? (Remembering)**

The dispatcher is the module that gives control of the CPU to the process selected by the short-term scheduler. This function involves:

- Switching context
- switching to user mode
- jumping to the proper location in the user program to restart that program.

**17. Define context switching. (Remembering)**

A context switch (also sometimes referred to as a process switch or a task switch) is the switching of the CPU from one process or thread to another.

A context is the contents of a CPU's registers and program counter at any point in time.

**18. What is dispatch latency? (Remembering)**

The time taken by the dispatcher to stop one process and start another running is known as dispatch latency.

**19. What are the various scheduling criteria for CPU scheduling? (Remembering)**

The various scheduling criteria are

- CPU utilization, \* Throughput,
- Turnaround time, \* Waiting time
- Response time,

**20. Define throughput? (Remembering)**

Throughput in CPU scheduling is the number of processes that are completed per unit time. For long processes, this rate may be one process per hour; for short transactions, throughput might be 10 processes per second.

**21. What is turnaround time? (Remembering)**

Turnaround time is the interval from the time of submission to the time of completion of a process. It is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O.

**22. Define race condition. (Remembering)**

When several process access and manipulate same data concurrently, then the outcome of the execution depends on particular order in which the access takes place is called race condition. To avoid race condition, only one process at a time can manipulate the shared variable.

**23. What is critical section problem? (Remembering)**

Consider a system consists of 'n' processes. Each process has segment of code called a critical section, in which the process may be changing common variables, updating a table, writing a file. When one process is executing in its critical section, no other process can allowed executing in its critical section.

**24. What are the requirements that a solution to the critical section problem must satisfy? (Remembering)**

The three requirements are

- ¬ Mutual exclusion
- ¬ Progress
- ¬ Bounded waiting

**25. Define deadlock. (Remembering)**

Process requests resources; if the resources are not available at that time, the process enters a wait state. Waiting processes may never again change state, because the resources they have requested are held by other waiting processes. This situation is called a deadlock.

**26. What are conditions under which a deadlock situation may arise? (Remembering)**

A deadlock situation can arise if the following four conditions hold simultaneously in a system:

1. Mutual exclusion
2. Hold and wait
3. No pre-emption
4. Circular wait

**27. Explain multi task and their functions in embedded system. (Remembering)**

This system implements cooperative and time-sliced multitasking, provides resource locking and mailbox services, implements an efficient paged memory manager, traps and reports errors, handles interrupts, and auto starts your application at system startup. By following some simple coding practices as shown in the documented coding examples, you can take advantage of these sophisticated features without having to worry about the implementation details.

**28. What is a thread? (Remembering)**

Thread is a concept in Java and UNIX and it is a light weight sub process or process in an application program. It is controlled by the OS kernel. It has a process structure, called thread stack, at the memory. It has a unique ID. It have states in the system as follows: stating, running, blocked and finished.

**29. What is shared data problem? (Remembering)**

If a variable is used in two different processes and another task if interrupts before the operation on that data is completed then the value of the variable may differ from the one expected if the earlier operation had been completed. This is known as shared data problem.

**30. Define Semaphore. (Remembering)**

Semaphore provides a mechanism to let a task wait till another finishes. It is a way of synchronizing concurrent processing operations. When a semaphore is taken by a task then that task has access to the necessary resources. When given the resources unlock. Semaphore can be used as an event flag or as a resource key.

**31. Define Mutex. (Remembering)**

A phenomenon for solving the shared data problem is known as semaphore. Mutex is a semaphore that gives at an instance two tasks mutually exclusive access to resources.

**32. Differentiate counting semaphore and binary semaphore. (Understanding)**

**Binary semaphore**

When the value of binary semaphore is one it is assumed that no task has taken it and that it has been released. When the value is 0 it is assumed that it has been taken.

**Counting semaphore**

Counting semaphore is a semaphore which can be taken and given number of times. Counting semaphores are unsigned integers.

**33. What is Priority inversion? (Remembering)**

A problem in which allow priority task in advertently does not release the process for a higher priority task.

**34. What is Deadlock situation? (Remembering)**

A set of processes or threads is deadlocked when each processor thread is waiting for a resource to be freed which is controlled by another process.

**35. Define Message Queue. (Remembering)**

A task sending the multiple FIFO or priority messages in to a queue for use by another task using queue messages as an input.

**36. What are the two ways of process communication? (Remembering)**

A process can send a communication in one of two ways: blocking or non-blocking. After sending a blocking communication, the process goes into the waiting state until it receives a response. On non-blocking communication allows the process to continue execution after sending the communication.

**37. Define CPU scheduling. (Remembering)**

CPU scheduling is the process of switching the CPU among various processes. CPU scheduling is the basis of multi-programmed operating systems. By switching the CPU among processes, the operating system can make the computer more productive.

**38. Define Synchronization. (Remembering)**

Message passing can be either blocking or non-blocking. Blocking is considered to be synchronous and non-blocking is considered to be asynchronous.

**39. Define Inter process communication. (Remembering)**

Inter-process communication (IPC) is a set of techniques for the exchange of data among multiple threads in one or more processes. Processes may be running on one or more computers connected by a network. IPC techniques are divided into methods for message passing, synchronization, shared memory, and remote procedure calls (RPC). The method of IPC used may vary based on the bandwidth and latency of communication between the threads, and the type of data being communicated.

**40. Define Mailbox. (Remembering)**

Mailboxes are software-engineering components used for inter process communication, or for inter-thread communication within the same process. A mailbox is a combination of a semaphore and a message queue (or pipe).

**41. What is meant by lock? (Remembering)**

There is a function in kernel called lock (). It locks a process to the resources till that process executes unlock.

**42. What is meant by spin lock? (Remembering)**

A Spin lock does not let a running task to be blocked instantly, but first successively tries with or without decreasing the trial periods before finally blocking a task. A spin-lock obviates need of context-switching by pre-emption and use of mutex function-calls to OS.

**UNIT V (CO5)**  
**REAL TIME OPERATING SYSTEM**

**1. Name the important terms of RTOS? (Remembering)**

Task, State, Scheduler, Shared data, Reentrancy

**2. Define process. (Remembering)**

Process is a computational unit that processes on a CPU under the control of a scheduling Kernel of an OS. It has a process structure, called Process control block. A process defines a sequentially executing program and its state.

**3. What is meant by PCB? (Remembering)**

Process Control Block' is abbreviated as PCB. PCB is a data structure which contains all the information and components regarding with the process.

**4. What is RTOS? (Remembering)**

An RTOS is an OS for response time controlled and event controlled processes. RTOS is an OS for embedded systems, as these have real time programming issues to solve.

**5. What are the goals of RTOS? (Remembering)**

- Facilitating easy sharing of resources
- Facilitating easy implantation of the application software
- Maximizing system performance
- Providing management functions for the processes, memory, and I/Os and for other functions for which it is designed.
- Providing management and organization functions for the devices and files and file like devices.
- Portability
- Interoperability
- Providing common set of interfaces.

**6. What are soft real time systems? (Remembering)**

Systems that demand good response but that allows some fudge in the deadlines are called soft real time systems.

**7. Distinguish hard-real time and soft-real time system? (Understanding)**

A hard real-time system guarantees that critical tasks complete on time. This goal requires that all delays in the system be bounded from the retrieval of the stored data to the time that it takes the operating system to finish any request made of it. A soft real time system where a critical real-time task gets priority over other tasks and retains that priority until it completes. As in hard real time systems kernel delays need to be bounded

**8. Define preemption. (Remembering)**

Preemption is the act of forcing a process out of execution i.e. making a context switch.

**9. List the functions of a kernel.**

**(Remembering)**

- Process management
- Scheduling
- IPC
- Memory management
- I/O management
- Device management

**10. What are the two methods by which a running requests resources? (Remembering)**

- Message,
- System call

**11. What are the functions of device manager? (Remembering)**

- Device detection and addition
- Device deletion
- Device allocation and registration
- Detaching and deregistration
- Device sharing

**12. List the set of OS command functions for a device. (Remembering)**

- Create and open
- Read and Write
- Close and delete

**13. What are the three methods by which an RTOS responds to a hardware source call on interrupt? (Remembering)**

- Direct call to ISR by an interrupt source
- Direct call to RTOS by an interrupt source and temporary suspension of a Scheduled task.
- Direct call to RTOS by an interrupt source and scheduling of tasks as well as ISRs by the RTOS.

**14. Name any two important RTOS. (Remembering)**

- MUCOS, • VxWorks

**15. What is meant by well tested and debugged RTOS? (Remembering)**

An RTOS which is thoroughly tested and debugged in a number of situations.

**16. What is sophisticated multitasking embedded system? (Remembering)**

A system that has multitasking needs with multiple features and in which the tasks have deadlines that must be adhered.

**17. List the types of RTOS. (Remembering)**

- In-House developed RTOS
- Broad-based Commercial RTOS
- General Purposes OS with RTOS
- Special Focus RTOS
-

**18. What are the features of MC/OS II? (Remembering)**

- Preemptive • Multitasking • Scalable • Portable

**19. What is MICRO C/OS II? (Remembering)**

- It stands for micro-controller operating system (UC/OS II).
- It is a real time kernel
- The other names of MICRO C/OS II are MUCOS and UCOS.
- The codes are in 'C' and Assembly language.

**20. What is meant by supervisory mode? (Remembering)**

In the supervisory mode the kernel codes run in protected mode. Only a system call is permitted to read and write into the protected memory allotted to the OS functions, data, stack and heap. The kernel space functions execute faster than user-space functions.

**21. Name the RTOS functions of mailboxes. (Remembering)**

The RTOS functions are

1. Create
2. Write
3. Read
4. Check
5. Destroy

**22. Is there any difference between mailboxes in different RTOSs? (understanding)**

Yes, the variations are

1. Some RTOSs allow a certain number of messages in each mailbox that is chosen by the user while creating a mailbox. Others allow only one message in a mailbox at a time.
2. In some RTOSs the number of messages in each mailbox is limited. There is a limit to the total number of messages that can be in all of the mailboxes in the system but these messages will be distributed into the individual mailbox, as they are needed.

**23. List the variations between pipes in different RTOSs. (Remembering)**

The variations are

1. Some RTOSs allow writing message of varying lengths onto pipes.
2. Pipes in some RTOSs are entirely byte oriented.
3. Some RTOSs use standard C library functions to read from and write to pipes.

**24. What are the two rules that the interrupt routines must follow? (Remembering)**

The two rules used by interrupt routines are,

RULE 1: An interrupt routine must not call any RTOS function that might block the caller.

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

**25. List out the most common RTOS timing services. (Remembering)**

The most common RTOS timing services are,

- a. The task can block itself for specified number of system ticks.
- b. A task can limit how many system ticks it will wait for a semaphore, a queue etc.
- c. Your code can tell the RTOS to call specified function after a specified number of system ticks.

**26. What is the advantage of time-slicing? (Remembering)**

The advantage of time slicing is that, when several human users have computed intensive programs running on a single system, each program gets some microprocessor time and each user sees programs.

**27. What should be done to guarantee that a hard real time system meets its deadlines? (Remembering)**

In order to guarantee that a hard real time system meets its deadlines, we must ensure that each of the tasks has a predictable worst case execution time.

**28. What are the disadvantages of using certain memory saving techniques? (Remembering)**

The disadvantages are,

1. Two or more tasks perform same functions
2. Development tools may sabotage the user.
3. RTOS contains many functions that are not needed to the task.
4. Certain C statements are translated into huge number of assembly language instructions.

**29. What are the different methods of power saving? (Remembering)**

A common power saving method is one in which the microprocessor stops executing instructions, stops any built in peripherals and stops its clock circuit.

Another method is one in which the microprocessor stops executing instructions but the on-board peripherals continue to operate.

Another common method for saving power is to turn off the entire system and have the user turn it back on when it is needed.

**30. Give the needs for memory management.(Remembering)**

Each new model of computer seems to come with more main memory than the last, but, since the memory requirements of the software rise just as fast, memory is always a precious commodity, hence the need for memory management .

- Memory is allocated to a process when needed and it
- Memory is de allocated when no longer in use
- Swapping allows the total memory used by all the running processes to exceed main memory
- Virtual memory makes it possible to run a single program that uses more memory than the main memory (normally RAM) available on the system. Virtual memory is normally divided into pages.
- Programs refer to parts of memory using addresses. In a virtual memory system, these are virtual addresses. The virtual address is mapped onto physical addresses by a memory management unit (MMU)



## **PART B QUESTIONS**

### **UNIT-I (CO1)**

#### **INTRODUCTION TO EMBEDDED SYSTEMS**

1. Explain the basic building blocks of embedded systems. **(Remembering)**
2. List the hardware units that must be present in the embedded systems. **(Remembering)**
3. Compare embedded system with general computing system. **(understanding)**
4. Write the classification of embedded system and explain the Exemplary applications of each type of embedded system. **(understanding)**
5. Explain the Embedded System on Chip (SoC) in VLSI circuit. **(Remembering)**
6. i) Explain the various form of memories present in a system **(Remembering)**
7. ii) Explain the software tools in designing of an embedded system.  
**(Remembering)**
8. What is the need for IDE in an Embedded Architecture? Discuss. **(Remembering)**
9. Explain the various forms of memory and the functions assigned to them.  
**(Remembering)**
10. Describe the architecture of a typical micro controller with a neat diagram.  
**(Understanding)**
11. Explain the basic processors and hardware units in the embedded system.  
**(Remembering)**
12. Explain how software is embedded into a system. **(Remembering)**
13. Explain the methods used in the embedded system on a chip. **(Remembering)**
14. Explain embedded system design process. **(Remembering)**
15. Explain embedded system design process of automatic chocolate vending machine.  
**(Remembering)**
16. Explain embedded system design process of digital camera. **(Remembering)**
17. Explain the case study of an embedded system for a smart card. **(Remembering)**

### **UNIT – II (CO2)**

#### **PROCESSOR AND MEMORY ORGANIZATION**

1. Explain selection of processor and memory devices? **(Remembering)**
2. Draw and explain the structural units in a processor? **(Remembering)**
3. Explain memory types and memory mapping with neat diagram? **(Remembering)**

4. Explain different types of I/O device with neat diagram? **(Remembering)**
5. Draw the circuit for interfacing processor, memory and I/O devices. **(Remembering)**
6. Design memory the devices needed for a low resolution uncolored digital camera.  
**(Remembering)**
7. Explain Automatic operation unit in advanced processors. **(Remembering)**
8. What is the advanced processing unit found in the processor Architecture?  
**(Remembering)**
9. Explain the architecture and memory organization of ARM processor. **(Remembering)**
10. Explain interfacing of memory with processor. **(Remembering)**

### **UNIT-III (CO3)**

#### **EMBEDDED NETWORKS**

1. Explain the parallel port devices. **(Remembering)**
2. Explain the sophisticated interfacing features in device ports. **(Remembering)**
3. Explain the signal using a transfer of byte when using the I<sup>2</sup>C bus and also the format of Bits at the I<sup>2</sup>C bus with diagram. **(Remembering)**
4. Explain the internal serial communication devices. **(Remembering)**
5. Explain the following parallel communication buses. **(Remembering)**
  - i) ISA bus
  - ii) PCI and PCI/X
6. Write short notes on **(Remembering)**
  - (i) Analog to digital converter
  - (ii) UART
7. Explain the synchronous and asynchronous communications from serial devices.  
**(Remembering)**
8. Explain the various bus structures used in embedded systems. **(Remembering)**
9. Explain the serial communication using I2C, CAN, and USB in detail. **(Remembering)**
10. Explain the signal using a transfer of byte when using the I2C bus and also the format of bits at the I2C bus with diagram. **(Remembering)**
11. Explain serial communication using SPI. **(Remembering)**
12. Explain with neat diagram about ARM bus. **(Remembering)**
13. Explain different types of network protocols. **(Remembering)**

## UNIT – IV (CO4)

### I/O PROGRAMMING SCHEDULE MECHANISM

1. Explain the use of Semaphores for a Task or for the Critical Sections of a Task.  
(Remembering)
2. Explain about task and different task states. (Remembering)
3. Explain the strategy for synchronization between the processes, ISRs, OS functions and tasks for resource management. (Remembering)
4. Explain how thread and process are used in embedded system. (Remembering)
5. List the mechanism available to solve the shared data problem? (Remembering)
6. Briefly explain different types of inter process communication features.  
(Remembering)
7. Explain in detail about message mailbox and queue. (Remembering)
8. Write about Semaphores with types in detail. (Remembering)
9. Explain in detail about mutex, lock and spin lock. (Remembering)
10. Explain inter process communication and synchronization. (Remembering)
10. What is the need of scheduling? Explain different types of scheduling mechanisms. (Remembering)
11. Explain the scheduler in which RTOS insert into the list and the ready task for Sequential execution in a co-operative round robin model. (Remembering)
12. (i) Explain the critical section service by a preemptive scheduler.  
ii) Explain the Rate Monotonic Co-operative scheduling. (Remembering)
13. Discuss with the diagram task synchronization model for a specific application.  
(Remembering)
14. Explain cyclic scheduling with time slicing. (Remembering)

## UNIT –V (CO5)

### REAL TIME OPERATING SYSTEMS

1. Explain the goals of operating system services. (Remembering)
2. Explain basic kernel services in operating system. (Remembering)
3. Write basic functions of RTOS. (Remembering)
4. Write about different types of RTOS. (Remembering)
5. Explain the three alternative systems in RTOS for responding a hardware source call with the diagram. (Remembering)

6. Explain process management in RTOS. (**Remembering**)
7. Explain I/O subsystems management in RTOS. (**Remembering**)
8. Briefly explain Event functions in RTOS. (**Remembering**)
9. Explain file system organization and implementation. (**Remembering**)
10. Explain the real time operating system features. (**Remembering**)
11. Compare OS Vs RTOS. (**Understanding**)
12. Explain Memory management in RTOS. (**Remembering**)
13. Explain in detail about device management functions in RTOS. (**Remembering**)
14. Explain how critical section is handled by a pre-emptive scheduler. (**Remembering**)
15. Explain in detail about Interrupt Service Handling in RTOS. (**Remembering**)
16. i) Explain the goals of operating system services. (**Remembering**)  
ii) Explain the three alternative systems in three RTOS for responding a hardware source call with the diagram. (**Remembering**)
17. Explain in detail about embedded software development process and tools. (**Remembering**)