UNIT I
INTRODUCTION TO EMBEDDED SYSTEM

   An embedded system is one that has computer hardware with software embedded on it as one of its most important component.

2. What are the hardware components of embedded systems?
   Input devices, Interfacing/Driver circuits
   Processor
   Timers
   Interrupt Controller
   Memory
   Serial/Parallel Ports
   Output Interfacing/Driver circuits
   Power supply, Reset & Oscillator circuits.

3. What are the classifications of Embedded system?
   Small scale Embedded System
   Medium scale Embedded System
   Sophisticated Embedded System.

4. Define processor.
   A processor is a device, which implements a process or processes as per the command given to it.

5. Define Embedded system processor.
   An embedded system processor has two essential units: Program Flow Control Unit and Execution Unit. The control unit includes a fetch unit for fetching instructions from the memory. Execution unit has circuits that implement the instructions.

6. What is microcontroller?
   A microcontroller is a unit with a processor. It is a single chip VLSI unit which, thought having limited computational capabilities, possesses enhanced input-output capabilities and a number of on-chip functional units.

7. What are the functional circuits of a microcontroller?
   The functional circuits of a microcontroller are,
   Processor
   Data & stack in internal RAM
   Timers & Watchdog timer
   ROM / PROM / EPROM
   External Memories Interfaces
   Interrupt Controller
   I/O Ports Control & Interfaces / Drivers
Serial UART communication port
Serial Synchronous communication port.

8. What is GPP & ASIP?
   GPP: A processor from a number of families of processors, microcontroller, embedded processors and DSPs having a general purpose instruction set and readily available compilers to enable programming in a high level language is called a General Purpose Processor (GPP).
   ASIP: A processor designed for specific application on a VLSI chip is called an Application Specific Instruction Processor.

9. What are the various forms of system memories?
   Internal RAM
   Internal ROM / PROM /EPROM
   External RAM
   Internal Caches
   E²PROM or flash memory
   External ROM / PROM
   RAM Memory buffers
   Caches

10. What are the steps required for converting assembly language into a ROM image?
    Assembler
    Linker
    Loader
    Locator
    Device Programmer / Foundry.

11. What are the steps required for converting high level language into a ROM image?
    Compiler
    Linker
    Loader
    Locator
    Device Programmer / Foundry.

12. What is Kernel?
    Kernel is a program with functions for memory allocation and deallocation, task scheduling, inter process communication, effective management of shared memory access by using the signals, exception handling signals, Semaphores, queues, mailboxes etc.

13. What are the features of DSP?
    A DSP provides fast, discrete-time, signal processing instructions. It has
very large instruction word (VLIW) processing capabilities, it processes Single Instruction Multiple Data (SIMD) instructions fast, it processes Discrete Cosine Transformations (DCTs) and inverse DCTs (IDTs) fast.

14. What is ASSP?
   Application Specific System Processor is a processing unit for specific tasks, for e.g. Image compression and that is integrated through the buses with the main processor in an embedded system.

15. Write short notes on multiplexer and demultiplexer.
   **Multiplexer**: A digital circuit that has digital inputs from multiple channels. It sends only one channel output at a time. The channel at the output has the same address as the channel address bits in its input.
   **Demultiplexer**: A digital circuit that has digital outputs at any instance in multiple channels. The channel that is connected is the one that has the same address as the channel address bits in its input.

16. What is the use of pipe & file?
   A pipe is used for buffering a stream of bytes.
   A file is used for reading and writing the stream of bytes or words.

17. Define RTOS.
   Real Time Operating System is software for real-time programming and scheduling, process and memory manager device drivers, device management and multitasking.

18. What is watchdog timer?
   It is a timer that times out from which resets the processor in case the program gets stuck for an unexpected time.

19. What is System On Chip?
   A system on a VLSI chip that has all of needed analog as well as digital circuits for e.g. in mobile phone.

20. What is the use of code optimizer?
   Code optimizer is used in the conversion of high level language into a ROM image for optimizing the code before linking.

21. Write some applications of Embedded Systems.
   - Smart card
   - Missiles & Satellites
   - Computer Networking
   - Automotives

22. What is ROM image?
   The final stage software is also called ROM image, because just as an
image is a unique sequence and arrangement of pixels, embedded software is also a unique placement and arrangement of bytes for instructions and data.

23. What is the use of glue logic circuit?
   It is a circuit for interconnecting the processor to external memory so that the appropriate chip-select signals, according to the system memory, map each of the memory chips.

24. Define interrupt handler.
   It is a unit that handles the processor operations arising out of an interrupt from a source.

25. Write briefly about reset circuit and power up reset.
   The reset circuit activates for a fixed period and then deactivates. It helps the processor to start the processing of instructions from a starting address.
   Powerup reset vector also provides a starting address which is different from that provided by a reset circuit.

26. What are various models used in the design of an embedded system?
   The various models used in the design of an embedded system are,
   Finite state machine
   Petri Net model
   Control & Data flow graph
   Activity diagram based on UML model
   The additional models are,
   Synchronous data flow graph.
   Timed petri nets & extended predicate / transition net.
   Multi threaded graph system.

27. Give some examples for Small, Medium & sophisticated scale embedded systems.
    Small Scale Embedded System
      Automatic chocolate vending machine
      Stepper motor controller for a robotics system
      Washing or cooking system
      Multitasking toys
    Medium Scale Embedded System
      Computer Networking Systems
      Entertainment systems
      Embedded firewall / Router
      Signal tracking system
    Sophisticated Scale Embedded System
      Embedded systems for wireless LAN & for convergent technology devices.
      Security products & high speed network security, gigabit rate encryption rate products
      Embedded systems for real time video & speech
UNIT II
DEVICES AND BUSES FOR DEVICES NETWORK

1. Define handshaking signals.
   The signals before storing the bits at the port buffer or before accepting
   the bits from the port buffer are called handshaking signals.

2. What are I/O device types?
   The types of I/O devices are,
   Serial Input
   Serial Output
   Serial UART Input
   Serial UART Output
   Parallel Port Single bit input
   Parallel Port Single bit output
   Parallel Port Input
   Parallel Port Output

3. What is Synchronous and Asynchronous Communication?
   Synchronous Communication:
   Communication in which a constant phase difference is maintained
   between the clocks that guide the transmitter and receiver. A maximum
   time interval is pre-fixed between which a frame of byte transmits.
   Asynchronous Communication:
   A communication in which a constant phase difference is not
   maintained and the clocks that guide the transmitter and receiver are
   separate. Time interval between which a frame of byte transmits is not
   pre-fixed and is indeterminate.

4. What is Timer and counting devices?
   A time gets the inputs from the internal clock with the processor or system
   clock. Counting device is a unit for getting the count-inputs on the occurrence of
   events that may be at irregular intervals.

5. What are the types of Timer?
   There are ten types of timer. They are,
   Hardware internal timer
   Software internal timer
   User software-controlled hardware timer
   RTOS controlled Hardware Timer
   Timer with periodic time out events
   One shot Timer
   Up count action timer
   Down count action timer
   Timer with its overflow flag – auto reset
   Timer with its overflow flag – no auto reset
6. Write the frame format for I\(^2\)C bus & CAN bus?

<table>
<thead>
<tr>
<th>Frame Format for I(^2)C bus</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Bit (1 bit)</td>
<td>Start Address Bit (7 bits)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frame Format for CAN bus</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Bit (1 bit)</td>
<td>Address of Destination (12 bits)</td>
</tr>
</tbody>
</table>

7. What is serial and parallel communication?
   In parallel communication, read and write operation take place on multiple bits at an instance.
   In serial communication, read and write operation take place with one bit at an instance and each bit of the message is separated by constant time intervals.

8. Give some advanced Parallel and Serial high speed buses.
   - Advanced Parallel high speed buses
     - GMII
     - XGML
     - CSIX – 1.6.6
     - Rapid IO
   - Advanced Serial high speed buses
     - XAUI
     - XSBI
     - SONET OC – 48
     - SONET OC – 192

   - It provides a superior throughput than EISA.
   - It is platform-independent.
   - Its clock rate is nearest to the sub-multiple of system clock.
   - It can be initialized at booting time.

10. What is Timer Overflow or Time-Out?
   A state in which the number of count inputs exceeded the last acquirable value and on reaching that state, an interrupt can be generated. This state is called Time-Out or Timer Overflow.

11. What is the difference between hardware timer and software timer?
    A hardware timer gets the inputs from the internal clock with the
processor or system clock. A device driver program programs it like any other physical device.

Software timer executes and increases or decreases a count variable on an interrupt from a timer output or from a real-time clock interrupt.

12. What is isosynchronous communication?
Isosynchronous communication is one in which a constant phase difference is not maintained between the frames but maintained between the frame clocks that guide the transmitter and receiver are not separate.

It is a gate with internally a missing connection between its drain and supply. It pulls up circuit voltage and current levels, which are required when interfacing it.

14. What is Quasi Bi-directional Port?
It is port with a dual advantage of using pull up circuit as per the voltage and current level required when interfacing it and using no pull up circuit for a short period sufficient to drive a LSTTL circuit.

15. What are the different states in a timer?
The different states in a timer are,
- Reset State
- Idle State
- Present State
- Overflow State
- Overrun State
- Active or Blocked State
- Done State
- Reset enabled / Enabled State
- Load enabled / Enabled State
- Auto Re-Load enabled / Disabled State

16. What are the three ways of communication from a device?
The three ways of communication from a device
- Synchronous Communication
- Asynchronous Communication
- Iso-synchronous Communication

17. What are the two ways of operation of the serial interface device?
The two ways of operation of the serial interface device are,
- Half duplex synchronous mode of operation called mode 0.
- Full duplex synchronous mode of operation called mode 1,2 or 3.
18. Name any two characteristics to be taken into account while interfacing a device port.

The two characteristics are,
- A port device may have multi-byte data input buffer(s) and data output buffers
- A port device may have a DDR (Data Direction Register).

19. Name some of the sophisticated features used in I/O devices.

Some of the sophisticated features used in I/O devices are,
- Schmitt trigger inputs are for noise elimination.
- Devices with low voltage gates and devices are using power management by preventing unnecessary toggling at the inputs are used for sophisticated application

20. Is it necessary to have a hardware timer device in a system? How do it work?

Yes, it is necessary to have at least one hardware timer device in a system. It is used as a system clock. The hardware timer gets the input from a clock out signal from the processor and activates the system clock as per the number of ticks preset at the hardware timer.


A software timer is software that executes and increases or decreases a Count variable as an interrupt on a timer output or on a real-time clock interrupt.

22. List some features of the CAN bus.

It has a serial line, which is bi-directional. It receives or sends a bit at an instance by operating at the maximum rate of 1 Mbps. It employs a twisted pair connection to each node, which runs to a maximum of 40m.

23. Explain briefly about USB.

USB is the Universal Serial Bus, used between a host system and a number of interconnected peripherals.

The two standards of USB available are,
- USB 1.1 – A low-speed 1.5 Mbps 3 meter channel along with a high-speed 12 Mbps 25 meter channel.
- USB 2.0 – High-speed 480 Mbps 25 meter channel.

24. Explain briefly the features of ISA and EISA buses.

ISA and EISA buses are compatible with IBM architecture. They are used for connecting devices following 10 addresses and interrupt vectors as per IBM PC architecture. EISA is a 32-bit extension of ISA. It also supports software interrupt functions and Ethernet devices.
25. Define protocol.
   A way of transmitting messages on a network by using software for adding the additional bits like starting bits, headers addresses of source and destination.

26. Define half-duplex communication.
   Serial port has one common I/O lines. For example, a telephone line message flows one way at an instance.

27. Define full duplex communication.
   Serial port has two distinct I/O lines. For example, a modem connection to the computer can port. There are two lines TXD and RXSD at 9 pins or 25 pins connector. Message flows both ways at an instance.

28. What are the features of the USB protocol?
   Using USB, a device can be attached configured and used, reset configured and used, share the bandwidth with other devices detached and reattached.

29. Explain briefly about PCI and PCI/X buses.
   PCI and PCI/X buses are used and these are independent from the IBM architecture. PCI/X is an extension of PCI and supports 64/100 MHz transfer. Lately, new versions have been introduced for the PCI bus architecture.
UNIT III
PROGRAMMING CONCEPTS AND EMBEDDED PROGRAMMING IN C, C++

1. What are advantages of writing embedded software in Assembly Language?
   It gives a precise control of the processor internal devices.
   The machine codes are compact.
   Device driver codes may need only a few assembly instructions

2. What are advantages of writing embedded software in C Language?
   The development cycle is short for complex systems.
   Type checking makes the program less prone to error.
   Control Structures make the program-flow path design tasks simple.
   Portability.

3. What is the use of type checking?
   Type checking makes the program less prone to error. For eg. It does not
   provide subtraction, multiplication and division on the ‘char’ data types.

4. Define Configuration files.
   Configuration files are the files for the configuration of the system. Device
   configuration codes can be put in a file of basic variables and included when
   needed.

5. What is difference between function and macrofunction?
   A macro function is a collection of codes that is defined in a program by a
   name. It differs from a function in the sense that once a macro is defined by a
   name, the compiler puts the corresponding codes for it at every place where that
   macro name appears. But the codes for a function are compiled once only.

6. What is recursive and reentrant function?
   Recursive Function:
   It is a function that calls itself. It must be a reentrant function also.
   Most often its use is avoided in embedded systems due to memory constraints.
   Reentrant Function:
   It is a function that is usable by the several tasks and routines at the
   same time. All its argument values are retrievable from the stack.

7. What is the use of modifier register?
   When a modifier register is inside a function block, a CPU register is
   temporarily allocated when needed. There is no ROM or RAM allocation.

8. Define Queue.
   It is a data structure into which elements can be sequentially inserted and
   retrieved in a FIFO mode. It needs two pointers, one for the queue tail for
   insertion and other for the queue head for deletion.
   A stack is a data structure in which elements can be pushed in or pulled out. It works on the principle of Last-In-First-Out (LIFO).

10. Explain briefly First In Provisionally Out (FIPO).
    FIPO has three pointers; one for the front (*QHEAD) a second for the back (*QTAIL) and a third pointer is tempfront (*QACK). Two pointers are the same as in every queue. The pointer defines a point upto, which an acknowledgement has been received.

11. How interrupts are handled in Queue?
    Queuing of pointers to the function on interrupts and later on calling the functions from this queue is a better approach as it provides the use of short execution time interrupt-service routines.

12. Define Compiler.
    Compiler is a program in the host computer, which does the development, and design testing and debugging. It converts high-level language into a machine language. The compiler can be turbo C, turbo C++ or Borland C, Borland C++.

    The cross compiler runs on a host but develops the machine codes for a targeted system.

14. Explain briefly stack overhead.
    The repeated call to recursive function may cause the stack to full. This leads to insufficient memory. Hence the stack overhead may occur which is primarily due to overflow of the stack.

15. What is meant by inline assembly?
    Inserting the assembly language codes in between the high-level language codes are called inline assembly. It gives the benefits of processor specific instructions and addressing modes.

16. What is Optimization of memory?
    When codes are made compact and fitted in small memory without affecting the code performance, it is called memory optimization.

17. Define scalar data types?
    The character, integer, unsigned integer, floating point number, long and double are called scalar data types. Unlike an array data consists of one single element.

18. Give some examples for reference data types.
    Arrays and strings are examples of reference data types.
19. What is meant by platform independence?
   A code that can port on different machine and operating systems is said to be platform independent.

20. What are preprocessor directives?
   Program statements and directives for the compiler before the main function to define global variable, global macro, new data type and global constants.

21. What are the differences between including a header file and a text file or data file?
   The differences between including a header file and a text file or data file are,
   - The header files are well tested and debugged modules.
   - They provide access to standard libraries.
   - The header file can include several text file or C files.
   - A text file is description of the text that contains specific information.

22. What is the use of void pointers?
   Void pointers can be used to point to the memory location of any data type.

23. How can optimization be used to eliminate the disadvantages in embedded C++ programs?
   Optimization can be used as follows.
   - Declare private as many classes as possible.
   - Use of char, int and boolean in place of the objects as arguments
   - Use local variables as much as feasible.
   - Recover memory already used by changing the reference to an object to NULL.

24. Name some features of source code engineering tools for embedded C++.
   Some features of source code engineering tools for embedded C++ are,
   - It searches and lists the dependencies and hierarchies of included header files.
   - It browses through object component relationship.

25. What is embedded C++?
   Embedded C++ is a version of C++ that provides for a selective disabling which is the disadvantage in C++. So, there is a less run-time overhead and less run-time library.
1. Define RTOS.
   Operating System with real time task scheduling, interrupt-latency control, Synchronization of tasks with IPCs, predictable timing and synchronization behavior of the system.

2. What is Round Robin or cyclic scheduling?
   A scheduling algorithm in which the tasks are scheduled in sequence from a list of ready tasks.

3. Explain briefly about Preemptive scheduling.
   A scheduling algorithm in which a higher priority task is forced (Preempted) to block by the scheduler.

4. What is Time Slicing and Fixed real time scheduling?
   **Time Slicing Scheduling**
   A scheduling algorithm in which each task is allotted a time slice after which it is blocked and waits for its turn on the next cycle.
   **Fixed Real Time Scheduling**
   A scheduling strategy in which the time for each task is fixed

5. Explain briefly the term Process.
   A code that has its independent program counter values and an independent stack. A single CPU system runs one process (or one thread of a process) at a time. It defines sequentially executing (running) program and its state. A state, during the running of a process, is represented by its status (running, blocked or finished), its control block, called process control block (PCB) or process structure, its data, objects and resources.

6. What is Task?
   A task is for the service of specific actions and may also correspond to the codes, which execute for an interrupt. A task is an independent process that takes control of the CPU when scheduled at an OS. Every task has a TCB (Task Control Block).

7. What is Task State?
   A state of a task that changes on scheduler directions. A task at an instance can be in one of the four states, idle, ready, blocked ad running that are controlled by the scheduler.

8. Define inter process communication.
   An output from one task (or process) passed to another task through the scheduler and use of signals, exceptions, semaphores, queues, mailboxes, pipes, sockets, and remote procedure call is known as inter process communication.
   Semaphore is a special variable or function that is used to take note of
certain actions to prevent another task or process from proceeding.

10. What is shared data problem?
    If a variable is used in two different processes (tasks) and another task if
interrupts without before the operation on that is completed, then the shared data
problem arises.

11. What is priority inversion problem? How it can be solved?
    A problem in which a low priority task inadvertently does not release the
process for a higher priority task. An operating system can take care of this it by
appropriate provisions.
    This problem can be solved by temporarily boosting the low priority task
to higher priority task which is called as priority inheritance.

12. Explain briefly Deadlock situation.
    A task waiting for some semaphore the release of a semaphore from a task
and another a different task waiting for another semaphore release to run. None of
these is able to proceed further. An operating system can take care of this it by
appropriate provisions.

13. Explain the term (i) Message Queue (ii) Mailbox (iii) Pipe (iv) Socket.
    Message Queue
    A task sending the multiple FIFO or priority message into a queue for use
by another task(s) using queue message as an input.
    Mailbox
    A message(s) or message pointer from a task that is addressed to another
task.
    Pipe
    A task sending the messages used by and another task using these as input.
A pipe can be a device like file which is also a virtual device.
    Socket
    It provides the logical link using a protocol between the tasks in a client-
server or peer-to-peer environment.

14. What is the use of Remote Procedure Call?
    A method used for connecting two remotely placed methods by first using
a protocol for connecting the processes. It is used in the cases of distributed tasks.

15. Define Thread.
    A minimum unit for a scheduler to schedule the CPU and other system
resources. A process may consist of multiple threads. A thread has an independent
process control block like a task control block and a thread also executes codes
under the control of a scheduler.

<table>
<thead>
<tr>
<th>Mutex Semaphore</th>
<th>Counting Semaphore</th>
</tr>
</thead>
<tbody>
<tr>
<td>A special variable used to take note of Certain actions to prevent any task or Process from proceeding further and at the same time let another task exclusively proceed further</td>
<td>A semaphore in which the value of which can be incremented and decremented and which is not a Boolean variable</td>
</tr>
</tbody>
</table>

17. Explain the uses of Task Control Block (TCB)?
   A memory block holds information of program counter, memory map, the signal (message) dispatch table, signal mask, task ID, CPU state (registers etc) and a kernel stack (for executing system calls, etc.).

18. What are the problems that may arise while using semaphores?
   The problems that may while using semaphores are,
   (i) Sharing of two semaphores creates a deadlock problem.
   (ii) Without a timeout an ISR worst-case latency may exceed the deadline.
   (iii) A semaphore not taken, and another task uses a shared variable.
   (iv) When using multiple semaphores, if an unintended task takes the semaphore, it creates a problem.
   (v) It may introduce priority inversion problem.

19. Explain any two features of RTOS.
   The two features of RTOS are,
   (i) Each queue for a message may need initialization before using the functions in the scheduler for the message queue.
   (ii) There may be a provision for multiple queues for the multiple types or destination of messages. Each queue may have an id.

20. What are the strategies used by RTOS on interrupt source calls?
   The strategies used by RTOS on interrupt source calls are,
   (i) Direct call to ISR by an interrupting source.
   (ii) Direct call to RTOS by an interrupting source and temporary suspension of a scheduled task
   (iii) Direct call to RTOS by an interrupting source and scheduling of tasks as well as ISRs by RTOS.

21. Name some RTOS services.
   (i) Basic OS functions
   (ii) RTOS Main functions
   (iii) Time Management
   (iv) Predictability
UNIT V
REAL TIME OPERATING SYSTEMS – PART II

1. Name any two important RTOS
   MUCOS and Vx works are the two important RTOS.

2. Name any two mailbox related functions.
   The two mailbox related functions are,
   (i) Creating a mailbox for an IPC.
   (ii) Checking for availability of an IPC after a message at mailbox.

3. Name any two queue functions for the inter-task communications.
   The two queue functions for the inter-task communications are,
   (i) Creating a queue for an IPC.
   (ii) Waiting for an IPC message at a queue.

4. Name some applications for the Vx Works RTOS.
   Some applications for the Vx Works RTOS are,
   (i) Multitasking environment.
   (ii) Inter-process communications.
   (iii) Synchronization using
        (a) Event flag
        (b) Mutually exclusive access using resource key
        (c) Counting mechanism using three types of semaphores
   (iv) Synchronization using POSIX.

5. What are the various features supported by Vx works?
   The various features supported by Vx works are,
   (i) Pipe drivers for inter-process communications as an virtual device.
   (ii) Network transparent sockets.
   (iii) Network drivers for shared memory and ethernet.
   (iv) RAM disk drivers for memory resident files.
   (v) Drivers for SCSI, keyboard, VGA display, disk and parallel port of a
       computer system, HDD, diskette, tapes, keyboard and displays.

6. What is an active task in the context of Vx works?
   An active task is one that is in one of the three states, ready, running or
   waiting.

7. What are the task service functions supported by Vx works?
   Vx works support the following task service functions.
   (i) Task creating
   (ii) Task running
   (iii) Task waiting
   (iv) Task pending
8. Name some interrupt service functions supported by Vx works?
   The four interrupt service functions supported by Vx works are,
   (i) `intLock()` - Disables interrupts.
   (ii) `intVectSet()` - Set the interrupt vector.
   (iii) `intVectGet()` - Get interrupt vector.
   (iv) `intVecBaseGet()` - Get interrupt vector base address.
   (v) `intContext()` - Returns true when calling functions is an ISR.

9. Name some of the inter-process communication functions.
   Some of the inter-process communication functions are,
   (i) `semBCreate()` - Creates a binary semaphore.
   (ii) `semMCreate()` - Creates a mutex semaphore.
   (iii) `semCCreate()` - Creates a counting semaphore.
   (iv) `semDelete()` - Delete a semaphore.

10. Name some of the inter-process communication functions used for messaging.
    Some of the inter-process communication functions used for messaging are,
    (i) `msgQCreate()` - Allocation and initializes a queue for the message.
    (ii) `msgQDelete()` - Eliminates the message queue by freeing the memory.
    (iii) `msgQSend()` - Sends into a queue.
    (iv) `msgQReceiver()` - Receives a message into the queue.

11. What is a signal in the RTOS environment?
    A signal is a flag like intimation to RTOS for development of certain situations during a run that need urgent attention by executing as ISR.

12. What is meant by task delay?
    Time delay is the minimum number of system ticks that a task must wait.

13. What are Vx works pipes?
    Vx works pipes are queues that can be opened and closed like a file. Pipes are like virtual IO devices that store the messages as FIFO.

14. What are the different types of scheduling supported by Vx works?
    The different types of scheduling supported by Vx works are,
    (i) Round robin time sliced scheduling.
    (ii) Preemptive scheduling.
15. How are pending tasks handled in Vx works?
   The pending tasks handled in Vx works are,
   (i) Using task priority.
   (ii) As a FIFO when accepting an IPC.

16. Explain briefly about the system timer in MUCOS.
   There are functions for initiating the system time in MUCOS. Starting a
   multitasking system by a first and later suspending if forever is shown as a
   technique in programming for a multitasking system.

17. How are pending tasks handled in Vx works?
   The pending tasks handled in Vx works are,
   (i) Using task priority.
   (ii) As a FIFO when accepting an IPC.

18. Briefly explain the features supported for mailbox by MUCOS.
   MUCOS has mailbox functions and a simple feature that a mailbox has
   one message pointer per mailbox. There can be any number of messages or bytes,
   provided the same pointer accesses them.

19. What is meant by task resumption?
   Task can be scheduled now when the turn comes, which was delayed or
   suspended.

20. Explain briefly the queue functions adapted in MUCOS.
   A queue in MUCOS receives from a sender task and array of message
   pointers. Message pointers insertion can be such that later on it can retrieve in FIFO
   method as well as in LIFO method from a queue. It depends on whether the post was
   used or post front function was used, respectively. This helps in taking notice of a
   high priority message at the queue.
1. Explain the classification of Embedded Systems?
   - Small scale Embedded Systems.
   - Medium scale Embedded Systems.
   - Sophisticated Embedded Systems.

2. Discuss the skills required for an Embedded system designer.
   - Skills for small scale Embedded System Designer.
   - Skills for medium scale Embedded System Designer.
   - Skills for sophisticated Embedded System Designer.

3. Explain the Hardware units of an Embedded System?
   - Processor in a System
   - Microcontroller / Microprocessor
   - Digital Signal Processor (DSP)
   - Application Specific System Processors (ASSPs) in Embedded Systems
   - Multi-processor Systems using General Purpose Processors (GPP).
   - Power Source and managing the power Dissipation & Consumption.
   - Clock Oscillator Circuit and Clocking Unit.
   - Real Time Clock and Timers for various Timing and Counting.
   - Reset Circuit, Power-up Reset and Watchdog-Timer Reset.
   - Memories, Input, Output and I/O Ports, IO Buses and IO Interfaces.
   - Interrupt Handler, DAC, LCD and LED Displays.
   - Keypad / Keyboard.
   - Pulse Dialer, Modem and Transceiver, GPIB Link.
   - Linking & Interfacing Buses

4. Write short notes on
   (i) GPP
       Explanation
       Examples
   (ii) ASSP
       Explanation
       Examples
   (iii) DSP
       Explanation
       Examples
   (iv) ASIC
       Explanation
       Examples
   (v) SoC
       Explanation
       Examples
5. Explain the steps to convert an assembly language & High-level program program into a ROM image?

Assembly Language to ROM Image
- Explanation

Machine Specific Assembly Language Program

Assembler

Machine codes for the programs at Various Addresses

Re-allocating Addresses

Machine codes ready for Loading at Various Addresses

Needed Machine codes from library

Linker

Bytes for Linked Programs

Loader

Device (ROM) Programmer (Burner)

Data Bytes

Embedded System ROM Memory

High-level Language to ROM Image
- Explanation

C Program Functions

Compiler

Machine Codes in Object File

Code Optimizer

Needed Machine Codes from Library

Linker

Bytes for Linked Programs

Same Steps As in Above Fig

Embedded System ROM Memory
6. Explain the software tools required for designing an embedded system

- Editor
- Interpreter
- Compiler
- Assembler
- Cross Assembler
- Simulator
- RTOS
- Source Code
- Engineering Software
- Stethoscope
- Trace Scope
- Prototyper
- Locator

UNIT II

1. Explain parallel port devices?
   - Diagram
   - Explanation

2. Write short notes on.
   - (i) ISA Bus
     - Explanation
   - (ii) USB Bus
     - Explanation
   - (iii) PCI/X Bus
     - Explanation
   - (iv) I2C Bus
     - Frame Format
     - Explanation
   - (v) CAN Bus
     - Frame Format
     - Explanation

3. Explain the three modes of serial communication, ‘Synchronous’, ‘Iso-synchronous’ & ‘Asynchronous’ from the serial devices with one example?
   - Characteristics
   - Explanation
   - Examples

4. Explain the types and examples of I/O devices?
   - Types – Explanation
   - Examples
5. Discuss the sophisticated interfacing features.
   Explanation

   **UNIT III**

1. Discuss the uses of the various data structures in a program elements.
   Queue
   Stack
   Array
   List
   Tree

2. Explain the use of Modifiers, Pointers?
   Explanation

3. Discuss the uses of Conditions, Loops & Infinite Loops.
   Explanation
   Examples

4. Explain Recursive & Reentrant function?
   Recursive Function
   Example
   Explanation
   Reentrant Function
   Example
   Explanation

5. How interrupts are handled in Queues?
   Explanation
   Examples

   **UNIT IV**

1. Explain Co-operative Round Robin Scheduling algorithm in detail.
   Explanation
   Algorithm

2. Explain Cyclic Scheduling with Time Slicing algorithm in detail.
   Explanation
   Algorithm

3. Explain Preemptive Scheduling algorithms in detail.
   Explanation
   Algorithm

4. How Critical section is handled by a preemptive scheduler? Give the difference between Co-operative & Preemptive.
   Explanation
   Algorithm, Difference
5. Explain the inter process communications in detail.
   - Signals
   - Message Queues
   - Mailboxes
   - Pipes
   - Virtual Sockets
   - Remote Procedure Calls (RPCs)

UNIT V

1. Explain Micro C/OS-II in detail.
   - Introduction
   - Explanation

2. Explain Vx works RTOS in detail.
   - Introduction
   - Explanation

3. Explain the functions that are used in MUCOS.
   - System Level functions
   - Task service functions
   - Semaphore Related functions
   - Mailbox Related functions
   - Queue Related functions

4. (a) Explain the difference between MUCOS and Vx works.
    - Difference

   (b) Explain multi tasks and their functions.
    - Introduction
    - Explanation
    - Functions

5. Design and explain any one embedded system.
   - Design
   - Example
   - Coding
   - Explanation