EC6703 EMBEDDED AND REALTIME SYSTEMS

UNIT I –INTRODUCTION TO EMBEDDED COMPUTING AND ARM PROCESSORS

PART A

1. What is an embedded computer system?

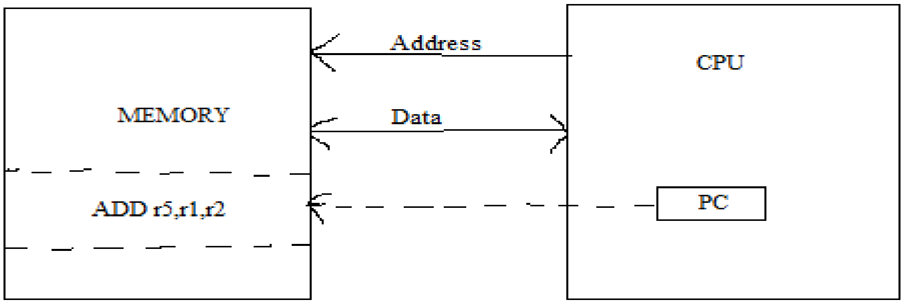
   It is any device that includes a programmable computer but is not itself intended to be a general-purpose computer. Thus, a PC is not itself an embedded computing system. But a fax machine or a clock built from a microprocessor is an embedded computing system. This means that embedded computing system design is a useful skill for many types of product design. Automobiles, cell phones, and even household appliances make extensive use of microprocessors.
2. Why microprocessor is used in embedded system?
   - ✓ Microprocessors are a very efficient way to implement digital systems.
   - ✓ Microprocessors make it easier to design families of products that can be built to provide various feature sets at different price points and can be extended to provide new features to keep up with rapidly changing markets.
3. Mention the challenges in embedded computing system design.
   - ✓ Hardware
   - ✓ Deadlines
   - ✓ Power Consumption
   - ✓ Upgradability
   - ✓ Reliability
4. Mention the reasons that makes embedded computing machines design difficult.
   - ✓ Complex testing
   - ✓ Limited controllability and observability
   - ✓ Restricted development environment
5. Illustrate the importance of design methodology.

   A design methodology is important for three reasons.
   - ✓ First, it allows us to keep a scorecard on a design to ensure that we have done everything we need to do, such as optimizing performance or performing functional tests.
   - ✓ Second, it allows us to develop computer-aided design tools. Developing a single program that takes in a concept for an embedded system and emits a completed design would be a daunting task, but by first breaking the process into manageable steps, we can work on automating the steps one at a time.
   - ✓ Third, a design methodology makes it much easier for members of a design team to communicate.
6. Describe the major steps in embedded system design process.
   - ✓ Requirements

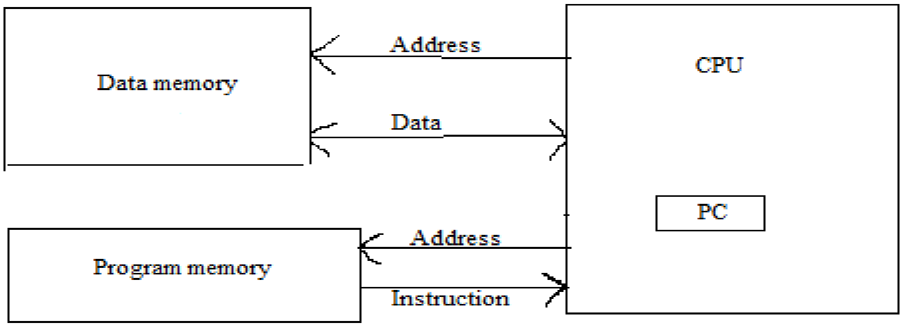
- ✓ Specification
- ✓ Architecture
- ✓ Components
- ✓ System integration

7. State the major goals of embedded system design.
   i. Manufacturing cost
   ii. Performance
   iii. Power consumption
8. List the non-functional requirements of an Embedded Architecture.
   - ✓ Performance
   - ✓ Cost
   - ✓ Physical size and weight
   - ✓ Power consumption
9. Assess the characteristics of embedded computing.
   - ✓ Embedded computing systems have to provide sophisticated functionality such as complex algorithms and user interface.
   - ✓ Embedded computing operations must often be performed to meet deadline are realtime and multirate.
   - ✓ Cost of various sorts are important based on manufacturing cost and power and energy consumption.
10. Compare the difference between the Harvard & Von-Neumann architecture?

von Neumann architecture:



Harvard architecture:



11. Evaluate the different types of relationships while designing an embedded system.
    - ✓ Requirements

- ✓ Specification
- ✓ Architecture
- ✓ Components
- ✓ System integration

12. List the functions of ARM processor in supervisor mode.

    The supervisor mode had privilege that user modes do not control of memory management unit is reserved for supervisor mode to avoid the obvious problems that could occur when programs bug cause inadvertent changes in the memory management registers. This is the function of ARM process in supervisor mode.

13. Investigate the significance of ARM Processor compared to other processors.
    - ✓ Enhanced power saving design
    - ✓ Mostly single cycle execution
    - ✓ An orthogonal instruction set
    - ✓ Load/store architecture

14. Identify the various issues in real time computing.
    - ✓ Complex testing
    - ✓ Limited controllability and observability
    - ✓ Restricted development environment

15. Evaluate the purpose of current program status register (CPSR) and Z-bit.

    The important basic register in the programming model is the current program status register (CPSR). This register is set automatically during every arithmetic, logical, or shifting operation. The top four bits of the CPSR hold the following useful information about the results of that arithmetic/logical operation:

    - The negative (N) bit is set when the result is negative in two's-complement arithmetic.
    - The zero (Z) bit is set when every bit of the result is zero.
    - The carry (C) bit is set when there is a carry out of the operation.
    - The overflow (V) bit is set when an arithmetic operation results in an overflow. These bits can be used to easily check the results of an arithmetic operation.

16. Discuss the addressing modes of C55x DSP.

    The C55x has three addressing modes:

    - Absolute addressing supplies an address in the instruction.
    - Direct addressing supplies an offset.
    - Indirect addressing uses a register as a pointer.

17. Analyze the term procedure linkage.

    A C function returns a value (unless its return type is void); subroutine or procedures are the common names for such a construct when it does not return a value. Consider this simple use of a function in C:

    ```
    x=a+b;
    foo(x);
    y=c-d;
    ```

A function returns to the code immediately after the function call, in this case the assignment to y. A simple branch is insufficient because we would not know where to return. To properly return, we must save the PC value when the procedure/function is called and, when the procedure is finished, set the PC to the address of the instruction just after the call to the procedure. The branch-and-link instruction is used in the ARM for procedure calls. For instance, BL foo;

18. Demonstrate the key parameters of CISC and RISC.

CISC	**RISC**
Emphasis on hardware	Emphasis on software
Includes multi clock	Single clock
Complex instructions	Reduced instructions
Memory to memory: Load and Store incorporated in instructions	Register to Register: Load and Store are independent in instructions
High cycles per second, small code size	Low cycles per second, large code size
Transistors used for storing complex instructions	Spends more transistors on memory registers

19. Frame the parameters used to evaluate the CPU performance.
    - ✓ Pipelining
    - ✓ Caching
20. Interpret the functions of Co-Processor.

One way to provide such flexibility at the instruction set level is to allow coprocessors, which are attached to the CPU and implement some of the instructions.
To support co-processors, certain opcodes must be reserved in the instruction set for co-processor operations. Because it executes instructions, a co-processor must be tightly coupled to the CPU. When the CPU receives a co-processor instruction, the CPU must activate the co-processor and pass it the relevant instruction. Coprocessor instructions can load and store co-processor registers or can perform internal operations. The CPU can suspend execution to wait for the co-processor instruction to finish; it can also take a more superscalar approach and continue executing instructions while waiting for the co-processor to finish.

PART B

1. Analyze in detail about the challenges in embedded computing system design.(8)
2. What are the parameters to be considered while designing an Embedded System Process? (8)
3. State the importance of Structural and Behavioral description in detail. (8)
4. Draw the architecture of ARM processor. (8)
5. List the various blocks of an Embedded System in detail (8)
6. Mention the major levels of abstraction in design process for GPS moving map.(8)
7. Write down the major operations and data flows of a GPS moving map and draw its Architecture. (8)

8. Evaluate CPU performance. (6)
9. Determine various instruction set preliminaries. (10)
10. Explain in detail about the characteristics of embedded computing applications.(8)
11. Explain in detail about supervisor mode, exception and traps. (8)
12. Explain about caches and memory management units. (10)
13. Analyze the concept of Pipelining. (6)
14. Write the data operations of an ARM processor (8)
15. Sketch the advanced ARM Processor features in detail (8)
16. Discuss with a simple system namely, a model train controller, how will you use the UML to model systems? (8)
17. Summarize the operation of BL instruction; including the state of ARM registers before and after its operations. (4)
18. Discuss how do you return an ARM procedure? (4)
19. Discuss about the CPU performance. (4)
20. Discuss in detail about Coprocessors. (12)
21. Demonstrate the operation of ARM processor. (8)
22. Illustrate the advanced features of ARM Processor and explain (8)
23. Design a Model Train Controller with suitable diagrams and formulate its design steps. (16)
24. Describe the performance of embedded computing systems. (8)
25. Discuss about the special functions of VLIW processors. (8)
26. Illustrate the special features of CPU power consumption. (8)
27. Demonstrate CPU Power consumption with the help of real time application (8).

UNIT II EMBEDDED COMPUTING PLATFORM DESIGN

PART A

1. Illustrate the bus protocol especially, the four-cycle handshake?

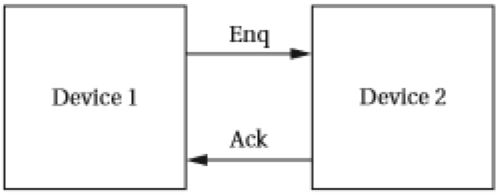
The basic building block of most bus protocols is the four-cycle handshake. The handshake ensures that when two devices want to communicate, one is ready to transmit and the other is ready to receive. The handshake uses a pair of wires dedicated to the handshake: enq (meaning enquiry) and ack (meaning acknowledge). Extra wires are used for the data transmitted during the handshake. Each step in the handshake is identified by a transition on enq or ack:

1. Device 1 raises its output to signal an enquiry, which tells device 2 that it should get ready to listen for data.

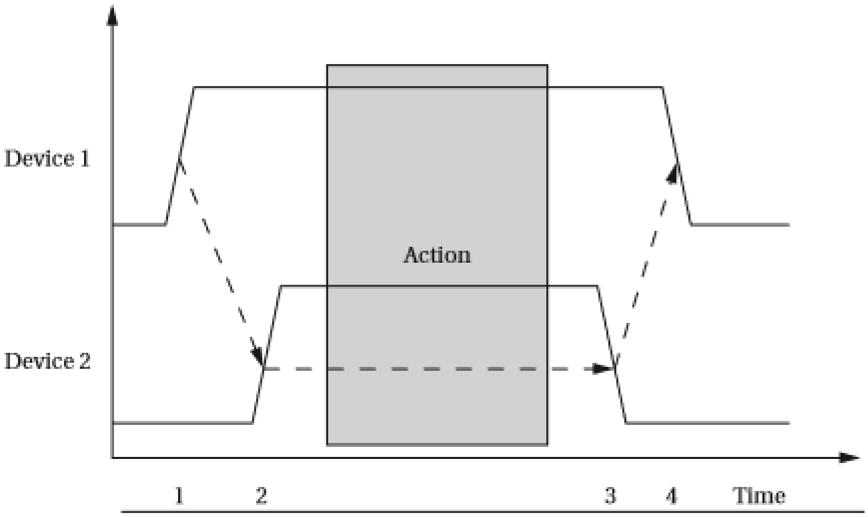
2. When device 2 is ready to receive, it raises its output to signal an acknowledgment. At this point, devices 1 and 2 can transmit or receive.

3. Once the data transfer is complete, device 2 lowers its output, signaling that it has received the data.

4. After seeing that ack has been released, device 1 lowers its output.



Structure



2. Define data flow graph.

A data flow graph is a model of a program with no conditionals. In a high-level programming language, a code segment with no conditionals more precisely, with only one entry and exit point is known as a basic block.

3. Formulate the theory of CPU buses.

   The bus is the mechanism by which the CPU communicates with memory and devices. A bus is, at a minimum, a collection of wires but it also defines a protocol by which the CPU, memory, and devices communicate. One of the major roles of the bus is to pro- vide an interface to memory.
4. List out the various compilation techniques.
   - ✓ Analysis and optimization of execution time
   - ✓ Power energy and program size
   - ✓ Program validation and testing
5. Validate the basic principles of basic compilation techniques.

   Compilation = Translation + Optimization

   The high level language program is translated into the lower level form of instructions, optimizations try to generate better instruction sequence.
6. Name any two techniques used to optimize execution time of program.
   - ✓ Loop optimization
   - ✓ Cache optimization
   - ✓ Performance optimization strategies
7. What does a linker do?

   A linker allows a program to be stitched together out of several smaller pieces. The linker operates on the object files created by the assembler and modifies the assembled code to make the necessary link between them.
8. Mention the different types of data transfer in USB.
   - ✓ Control transfer
   - ✓ Interrupt transfer
   - ✓ Bulk transfer
   - ✓ Isochronous transfer(Sequence of data)
9. Find the limitation of polling techniques.
   - ✓ It is wasteful of the processor time, as it needlessly checks the status of all devices all the time.
   - ✓ It is inherently slow, as it checks the status of all input/output devices before it comes back to check any given one again.
   - ✓ Priority of the devices cannot be determined frequently.
10. What do you mean by Control Bus in CPU?

    A control bus is a part of computer bus, used by CPU for communicating with other devices within the computer. The control bus carries commands from the CPU and returns status signal from the devices.
11. Discuss the reason to use latches for constructing input ports.

    The system’s data signals are sampled at a latch within the logic analyzer; the latch is controlled by either the system clock or the internal logic analyzer sampling clock, depending on whether the analyzer is being used in state or timing mode. Each sample is copied into a vector memory under the control of a state machine.

12. Summarize the two ways used for performing input and output operations

A data flow graph is a model of a program with no conditionals. In a high-level programming language, a code segment with no conditionals more precisely, with only one entry and exit point is known as a basic block.

A CDFG uses a data flow graph as an element, adding constructs to describe control. In a basic CDFG, we have two types of nodes: decision nodes and data flow nodes. A data flow node encapsulates a complete data flow graph to represent a basic block. We can use one type of decision node to describe all the types of control in a sequential program.

13. Discuss about Busy Wait I/O Concept in polling.

The simplest way to communicate with devices in a program is busy-wait I/O. Devices are typically slower than the CPU and may require many cycles to complete an operation. If the CPU is performing multiple operations on a single device, such as writing several characters to an output device, then it must wait for one operation to complete before starting the next one. Asking an I/O device whether it is finished by reading its status register is often called polling.

14. Explain the important stages of DMA.

Direct memory access (DMA) is a bus operation that allows reads and writes not controlled by the CPU. A DMA transfer is controlled by a DMA controller, which requests control of the bus from the CPU. After gaining control, the DMA controller performs read and write operations directly between devices and memory. The DMA requires the CPU to provide two additional bus signals:

- The bus request is an input to the CPU through which DMA controllers ask for ownership of the bus.
- The bus grant signals that the bus has been granted to the DMA controller.

15. Evaluate the importance of registers used in the DMA controller.

A typical DMA controller includes the following three registers:

- A starting address register specifies where the transfer is to begin.
- A length register specifies the number of words to be transferred.
- A status register allows the DMA controller to be operated by the CPU.

16. Differentiate compiler and cross compiler.

- ✓ Compiler translates highlevel language into machine language one by one translation.
- ✓ A cross-compiler is a compiler that runs on one type of machine but generates code for another. After compilation, the executable code is typically downloaded to the embedded system by USB.

17. List the difference between program location counter and program counter.

During scanning, the current location in memory is kept in a program location counter (PLC). Despite the similarity in name to a program counter, the PLC is not used to execute the program, only to assign memory locations to labels. For example, the PLC always makes exactly one pass through the program, whereas the program counter makes many passes over code in a loop. Thus, at the start of the first pass, the PLC is set to the program's starting address and the assembler looks at the first line. After examining the line, the assembler updates the PLC to the next location and looks at the next instruction.

18. Interpret the importance of Boot-block flash.

Most flash memories today allow certain blocks to be protected. A common application is to keep the boot-up code in a protected block but allow updates to other memory blocks on the device. As a result, this form of flash is commonly known as boot- block flash.

19. Write note on assembler/interpreters for embedded systems.

    - ✓ The assembler's job is to translate symbolic assembly language statements into bit-level representations of instructions known as object code. The assembler takes care of instruction formats and does part of the job of translating labels into addresses.
    - ✓ Interpreter constantly runs and interprets source code as a set of directives.

20. Elaborate the function of I/O device.

    Each I/O device monitors the CPU address bus and responds to any of the CPU access of device assigned address space, connecting the data bus to a desirable devices hardware register.

PART-B

1. Explain on how on chip memory management schemes can improve higher speed process. (12)
2. Describe about Memory devices with suitable examples. (16)
3. List any two factors which may be the cause for delay in peripheral interface (4)
4. Briefly explain with neat diagrams on how DMA based processor can remove delay for higher speed process. (12)
5. Discuss how component interfacing is done in embedded system in detail. (6)
6. Describe the development environment of an embedded system with suitable diagram? (10)
7. With a suitable example, explain how debugging is carried out using debuggers & compliers? (16)
8. Evaluate how Logic analyzer, In circuit Emulator and Co simulator are used as debugging tools with examples. (12)
9. Elaborate briefly about Assembly and Linking. (10)
10. Investigate the importance of program Validation and testing? (10)
11. Write about program level performance and software performance optimization. (10)
12. Describe briefly on the memory management & mapping techniques that enhance the efficiency of the processor. (16)

13. Summarize the advantage of vectored addressing of stack? (4)
14. Demonstrate the Model of Programs in detail. (8)
15. Illustrate about basic compilation techniques. (8)
16. What do you mean by memory system interface with CPU? Explain with examples. (16)
17. Illustrate with necessary diagrams about the design pattern, loop transformation and scheduling. (10)
18. Frame the key features of clear box testing. (6)

UNIT III PROCESSES AND OPERATING SYSTEMS

PART A

1. Define process in RTOS.
2. Compare between a process and thread.
3. Differentiate between initiation time and completion time
4. Summarize the essential criteria's of rate monolithic scheduling.
5. Describe context switching briefly.
6. What do you mean by time quantum?
7. Summarize the various scheduling states of a process.
8. Investigate the organization of scheduling policy.
9. How to compute the CPU utilization of the system.
10. Define rate monolithic scheduling
11. State the two ways of assigning priority to a process.
12. Evaluate the communication among processes which runs at different rates.
13. Define power management policy.
14. Explain multi-processing systems.
15. Determine the important characteristics of Multitasking.
16. Design a hard real-time operating system with an example.
17. Illustrate the principle of multi rate embedded system by quoting three examples
18. Give examples of blocking and Non-blocking inter process communication
19. What is response time?
20. Frame the two different styles used for inter process communication.

PART –B

1. Compare RMS versus EDF. (8)
2. Explain about Windows CE with a neat diagram. (8)
3. Discuss the strategies behind running periodic processes. (8)
4. Describe process state and scheduling. (8)
5. Explain preemptive real time operating systems in detail. (8)
6. Analyze the special characteristics of Processes and Internet with the help of a suitable diagrams. (8)
7. Outline about priority based scheduling in detail. (8)
8. Explain with the help of an example that the knowledge of data dependencies can help use the CPU more efficiently. (8)
9. Describe in detail about the inter process communication mechanism
   (i) Shared Memory communication (4)
   (ii) Message passing (4)
   (iii)Signals (4)
   (iv)Mailboxes (4)

10. Enumerate the context switch mechanism for moving the CPU from one executing process to another (8)
11. State how the Kernel determines the order of the processes which has to be executed. (8)
12. Evaluate operating system performance in detail. (8)
13. Justify the statement “Many Real Time Systems have been designed based on the assumption that there is no cache present, even though one actually exists.” (8)
14. Demonstrate in detail about power optimization strategies for CPU operation. (8)
15. Illustrate how the Predictive shut down technique proved itself as more sophisticated. (8)
16. Enumerate why an automobile engine requires multirate control (4)
17. Recall the performance of the Earliest – Deadline – First scheduling with other scheduling algorithms with suitable example. (12)
18. What is Real time operating system? (4)
19. Mention the special features of POSIX with neat diagram. (12)
20. Summarize the services of operating system in handling multiple tasks and multiple processes. (8)
21. Outline the features of preemptive execution with the help of a sequence diagram. (8)
22. Illustrate an approach to cooperative multitasking in the PIC16F with the help of a program. What would happen if we put the tasks into Time Handler? (12)
23. Examine about CPU usage metrics. (4)
24. Mention in detail about Shared Resources. (8)
25. Recognize the ARM atomic read/write operation in detail with the help of an example (8)

UNIT-4 SYSTEM DESIGN TECHNIQUES AND NETWORKS

PART A

1. List the OSI layers from lowest to highest level of abstraction.
2. What is a distributed embedded architecture?
3. Assess the 2 different approaches for designing an embedded system?
4. State the advantages of network based design.
5. Evaluate the issues in hardware and software design for an embedded system.
6. Describe the merits of embedded distributed architecture?
7. Illustrate the main stages of a design flow of an Embedded System.
8. Define I2C bus.
9. Describe the special characteristics of a CRC card
10. Mention the networks for distributed embedded systems.
11. Distinguish multistage network from direct network.
12. Determine the use of exponential back off technique.
13. Define message delay.
14. Discriminate single hop network from multi hop network.
15. Illustrate some internet enabled embedded systems.
16. Formulate various design methodologies.
17. Summarize the issues in hardware and software design for an embedded system.
18. Explain the advantages of hardware implementations.
19. Illustrate the specifications of a good requirement analysis.
20. Why we Prefer shared memory multiprocessor?

PART B

1. Explain in detail about

   i)MPSoCs (8)

   ii)Shared memory multiprocessor. (8)
2. Explain the design methodology of an embedded computing system in detail. (8)
3. Analyze the features of SDL Specification language with suitable diagrams. (8)
4. Give the examples of the component networks in a federated network for an automobile. (8)
5. Discuss a problem that led to the loss of Unmanned Martian space probe. (8)
6. Discuss in detail about

   i)CAN Bus. (8)

   ii)I2C Bus. (8)
7. Enumerate about Internet enabled Systems in detail. (8)
8. Mention the requirements needed to design an embedded system and how to determine them (8)
9. With a neat diagram, describe the typical bus transactions on the I2C Protocol. (8)

10. Discuss the role of distributed embedded architecture available for embedded systems. (8)
11. Discuss in detail about
    i)Characteristics of distributed embedded System. (8)
    ii)Architecture of Distributed Embedded System with neat sketches. (8)
12. State in detail about CAN bus protocol and Ethernet with necessary diagrams? (8)
13. List the characteristics of high performance embedded platforms which act as heterogeneous multiprocessors.
14. Demonstrate the operation of Ethernet enabled system. With a suitable example.(8)
15. ii)Illustrate scheduling and allocation in an accelerated embedded system with an help of suitable diagrams. (8)
16. Demonstrate the single threaded and multithreaded control of an accelerator in the embedded system design. (8)
17. Illustrate about the cache problem in a system involving an accelerator and suggest a method to overcome it. (8)
18. List
    i)The key features of Accelerated based embedded system. (8)
    ii)Characteristics of network based Embedded Systems in detail. (8)
19. Justify how Poor Specifications can lead to Poor Quality Code. (8)
20. Evaluate whether the aspects of a Poorly constructed specification necessarily lead to Bad Software. (8)
21. Propose a method for understanding the architectural design of a complex systems by using CRC Cards. (12)
22. Elaborate the important criterions that can be considered for design reviews in Quality Assurance process. (4)
23. Observe in detail about Quality Assurance Process using the following
    i) Quality Assurance Techniques (8)
    ii)Verifying the Specifications (8)

UNIT V CASE STUDY

PART A

1. Which kind of modulation is used in software modem?
2. Draw the block diagram of FSK detection Scheme in MODEM?
3. What is the specification language give an example?
4. Justify the advantages of software implementations.
5. Evaluate the UML diagram for Data Compressor.
6. Define flush in data compressor.
7. Summarize the New symbol Table.
8. Outline the state diagram of encoding behavior.
9. Outline the state diagram of insert behavior.
10. Identify the advantages of Software modem?
11. Analyze four cycle handshake signals.
12. Analyze the class diagram for alarm clock.
13. Draw the brayer pattern of color image.
14. Propose the architecture of digital still camera?
15. Illustrate the role of ADPCM scheme in Telephone answering machine with a neat diagram.
16. Elaborate the functions of digital camera.
17. Find out the need of block motion Estimation.
18. Identify release time.
19. Justify which compression technique is used for telephone answering machine.
20. Describe about White balance.

PART-B

1. Evaluate in detail the principle operation of software modem. (16)
2. List the features of PDA and data compressor in detail. (16)
3. Demonstrate in detail about Design Example of Alarm Clock? (16)
4. Briefly Explain the working of audio player in detail. (16)
5. Describe the following design stages used in the working of telephone answering machine in detail.
    - i. Theory of operations and requirements. (4)
    - ii. Specification (4)
    - iii. System Architecture (4)
    - iv. Component designing and testing (2)
    - v. System integration and testing (2)
6. Develop the working of Engine control unit in detail.
    - i. Theory of operations and requirements. (4)
    - ii. Specification (4)

iii. System Architecture (4)
iv. Component designing and testing (2)
v. System integration and testing (2)

7. Explain the working of video accelerator in detail.

    i. Theory of operations and requirements. (4)
    ii. Specification (4)
    iii. System Architecture (4)
    iv. Component designing and testing (2)
    v. System integration and testing (2)

8. Summarize the sequence diagram of taking picture with digital still camera.

    i. Theory of operations and requirements. (4)
    ii. Specification (4)
    iii. System Architecture (4)
    iv. Component designing and testing (2)
    v. System integration and testing (2)