DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

UNIT I
INTRODUCTION TO EMBEDDEDSYSTEM DESIGN

Syllabus:

Complex systems and microprocessors– Embedded system design process – Design example: Model train controller- Design methodologies- Design flows - Requirement Analysis – Specifications-System analysis and architecture design – Quality Assurance techniques - Designing with computing platforms – consumer electronics architecture – platform-level performance analysis.

Two mark questions:

1. **What is an embedded computer system? (Nov/Dec-2012)**

It is defined as any device that includes a programmable computer but is not itself intended to be 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.

2. **Why a programmable CPU is used rather than a hardwired unit?**

A programmable CPU is used rather than a hardwired unit for two reasons: first, it made the system easier to design and debug; and second, it allowed the possibility of upgrades and using the CPU for other purposes.

3. **List the characteristics of embedded computing applications. (Nov/Dec-2013)**

The characteristics of embedded computing applications are,

- Complex algorithms
- User interface
- Real time
- Multirate
- Manufacturing cost
- Power and Energy

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

4. **Why use microprocessors?**

There are two reasons for using microprocessors in embedded computing applications.

- Microprocessors are 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.

5. **Why not use PCs for all embedded computing? How many different hardware platforms do we need for embedded computing systems?**

PCs are widely used and provide a very flexible programming environment. Components of PCs are, in fact, used in many embedded computing systems. But several factors keep us from using the stock PC as the universal embedded computing platform.

6. **Define Cyber-physical systems. (Nov/Dec-2013)**

A cyber-physical system is one that combines physical devices, known as the plant, with computers that control the plant. The embedded computer is the cyber-part of the cyber-physical system. We can, in general, make certain trade-offs between the design of the control of the plant and the computational system that implements that control.

7. **List the challenges in embedded computing system design.**

The important problems that must be taken into account in embedded system design are,

- How much hardware do we need?
- How do we meet deadlines?
- How do we minimize power consumption?
- How do we design for upgradeability?
- Does it really work?

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

8. **What are the ways in which the nature of embedded computing machines makes their design more difficult? (Apr-2014)**

The nature of embedded computing machines makes their design more difficult in some ways. That are,

- Complex testing
- Limited observability and controllability
- Restricted development environments

9. **What are the different levels of abstractions in real-time behavior of an embedded computing system?**

In order to understand the real-time behavior of an embedded computing system, we have to analyze the system at several different levels of abstraction. That are,

- CPU
- Platform
- Program
- Task
- Multiprocessor

10. **What are the objectives of embedded system design process?**

There are two objectives for embedded system design process, that are,

- It will give us an introduction to the various steps in embedded system design before we delve into them in more detail.
- It will allow us to consider the design methodology itself.

11. **Why the design methodology is so important? (Nov/Dec-2012)**

A design methodology is important for three reasons,

- 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.
- It allows us to develop computer-aided design tools.
- A design methodology makes it much easier for members of a design team to communicate.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

12. What are the major level of abstraction in the design process?

The major level of abstraction in the Top-down design are,

- Requirements
- Specification
- Architecture
- Components
- System integration

The major level of abstraction in the Bottom-up design are,

- System integration
- Components
- Architecture
- Specification
- Requirements

13. What are all the major goals and the tasks we need to perform at every steps in the design process?

Goals:

- Manufacturing cost
- Performance
- Power consumption

Tasks at every steps:

- We must analyze the design at each step to determine how we can meet the specifications.
- We must then refine the design to add detail.
- We must verify the design to ensure that it still meets all system goals, such as cost, speed, and so on.

14. What are all the non-functional requirements in embedded system design process? (Nov/Dec-2013)

The typical non-functional requirements includes,

- Performance
- Cost
- Physical size and weight
- Power consumption

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

15. List out the entries in the requirements form as a check list.

The entries in the requirement form are,

- Purpose
- Inputs
- Outputs
- Functions
- Performance
- Manufacturing cost
- Power
- Physical size and weight

16. Define Unified Modeling Language (UML).

The UML is an Object-Oriented modeling language. Object-oriented design emphasizes two concepts of importance:

- It encourages the design to be described as a number of interaction objects, rather than a few large monolithic blocks of code.
- At least some of the objects will corresponds to real pieces of software or hardware in the system.

17. State the two complementary ways of Object-Oriented specification.

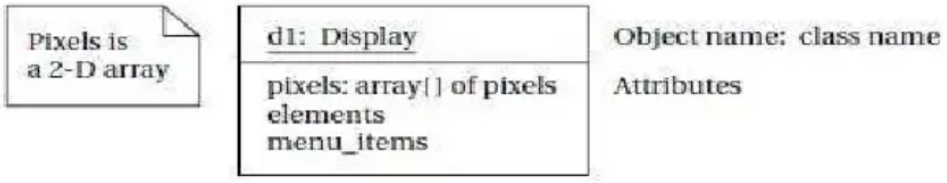
The two complementary ways Object-Oriented specification are,

- Object-Oriented specification allows a system to be described in a way that closely models real-world objects and their interactions.

- Object-Oriented specification provides a basic set of primitives that can be used to describe systems with particular attributes, irrespective of the relationships of those systems' components to real-world objects.

18. Figure an object describing a display in UML notation. (Nov/Dec-2014)

An object in UML notation



DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

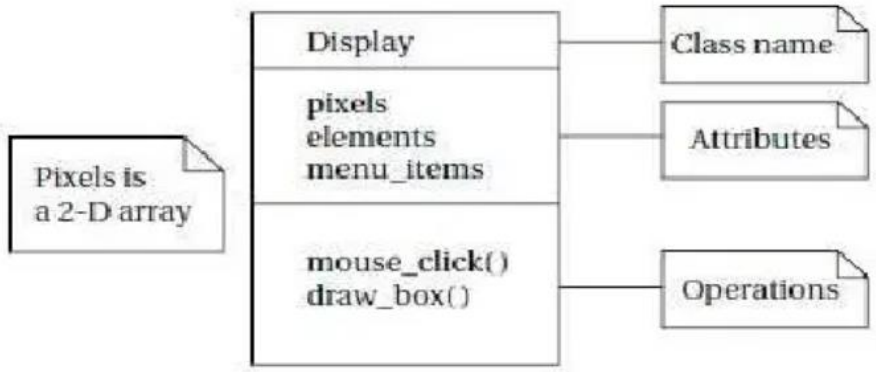
YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

19. Figure out A class in UML notation.

An object in UML notation.



20. What are the types of relationships that can exist between objects and classes?

The types of relationships that can exist between objects and classes

- **Association** occurs between objects that communicate with each other but have no ownership relationship between them.
- **Aggregation** describes a complex object made of smaller objects.
- **Composition** is a type of aggregation in which the owner does not allow access to the component objects.
- **Generalization** allows us to define one class in terms of another.

21. List the characteristics of Instruction set.

Instruction set can have a variety of characteristics, which are,

- Fixed versus variable length.
- Addressing modes.
- Numbers of operands.
- Types of operations supported.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

22. Compare RISC vs CISC. (Nov/Dec-2013)

CISC	RISC
Emphasis on hardware	Emphasis on software
Includes multi-clock	Single-clock
complex instructions	reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
high cycles per second, Small code sizes	Low cycles per second, large code sizes
Transistors used for storing complex instructions	Spends more transistors on memory registers

23. What are the basic features of assembly language?

Assembly languages usually share the same basic features:

- One instruction appears per line
- Labels, which gives names to memory locations, start in the first column
- Instructions must start in the second column or after to distinguish them from labels
- Comments run from some designated comment character to the end of the line.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

24. Define Packets & Data dependency.

A set of instructions is bundled together into a VLIW packet, which is a set of instructions that may be executed together. The execution of the next packet will not start until all the instruction in the current packet have finished executing. The compiler identifies packets by analyzing the program to determine sets of instructions that can always execute together.

A Data dependency is a relationship between the data operated on by instructions.

25. Compare VLIW vs Superscalar.

VLIW processors examine inter-instruction dependencies only within a packet of instructions. They rely on the compiler to determine the necessary dependencies and group instructions into a packet to avoid combinations of instructions that can't be properly executed in a packet. Superscalar processors, in contrast, use hardware to analyze the instruction stream and determine dependencies that need to be obeyed.

26. Define Load-Store Architecture in ARM Programming. (Nov/Dec-2013)

ARM is a load-store architecture-data operands must first be loaded into the CPU and then stored back to main memory to save the results.

27. What are the advanced ARM features?

The advanced ARM features are,

- DSP
- SIMD
- NEON
- TrustZone
- Jazelle
- Cortex

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

28. Define Busy-wait I/O and 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 and operation. If the CPU is performing multiple operations on a single device, such as writing several characters to an output devices, 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**.

29. What are the ways to generalized interrupts?

There are two ways in which interrupts can be generalized to handle multiple devices and to provide more flexible definitions for the associated hardware and software:

- Interrupt priorities allow the CPU to recognize some interrupts as more important than others
- Interrupt vectors allow the interrupting device to specify its handler

30. Define Masking. (Nov/Dec-2014)

The priority mechanism must ensure that a lower-priority interrupt does not occur when a higher-priority interrupt is being handled. The decision process is known as masking.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

UNIT II
ARM PROCESSOR AND PERIPHERALS

Syllabus:

ARM Architecture Versions – ARM Architecture – Instruction Set – Stacks and Subroutines – Features of the LPC 214X Family – Peripherals – The Timer Unit – Pulse Width Modulation Unit – UART – Block Diagram of ARM9 and ARM Cortex M3 MCU.

Two mark questions:

1. **What are all the major hardware components included in a typical computing platform?**
   - CPU: provides basic computational facility
   - RAM: used for program and data storage.
   - ROM: holds the boot program and some permanent data.
   - DMA controller: provides direct memory access capabilities.
   - Timers: used by the operating system for variety of purpose.
   - High speed bus: connect to the CPU bus through a bridge, allows fast devices to communicate efficiently with the rest of the system.
   - Low speed bus: provides an inexpensive way to connect simpler devices and may be necessary for backward compatibility as well.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

2. Define CPU bus. (Nov/Dec-2014)

The bus is the mechanism by which the CPU communicates with memory and devices. A bus is, at a minimum, 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 provide an interface to memory.

3. What is four cycle handshake? (Nov/Dec-2012)

The basic building block of most bus protocol 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** (enquiry) and **ACK** (acknowledge).

4. What are the major components on a typical bus?

The major components on a typical bus are,

- Clock provides synchronization to the bus components
- R/W is true when the bus is reading and false when the bus is writing
- Address is an a-bit bundle of signals that transmits the address for an access.
- Data is an n-bit bundle of signals that can carry data to or from the CPU
- Data ready signals when the values on the data bundle are valid.

5. What is timing diagram? (Nov/Dec-2014)

Sequence diagram don't give us enough detail to fully understand the hardware. To provide the required detail, the behavior of a bus is most often specified as a timing diagram. A timing diagram shows how the signals on a bus vary over time, but because values like the address and data can talk on many values, some standard notation is used to describe signals.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

6. Define Burst transfer.

The CPU sends one address but receives a sequence of data values. We add an extra line to the bus, called *burst'* here, which signals when a transaction is actually a burst. Releasing the *burst'* signal tells the device that enough data has been transmitted. To stop receiving data after the end of *data 4*, the CPU releases the *burst'* signal at the end of *data 3* because the device requires some time to recognize the end of the burst. Those values come from successive memory locations starting at the given address.

7. Write short note on DMA. (Nov/Dec-2012)

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.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

8. What are the three registers included in DMA controller?

The CPU controls the DMA operation through registers in the DMA controller. A typical DMA controller includes the following three registers:

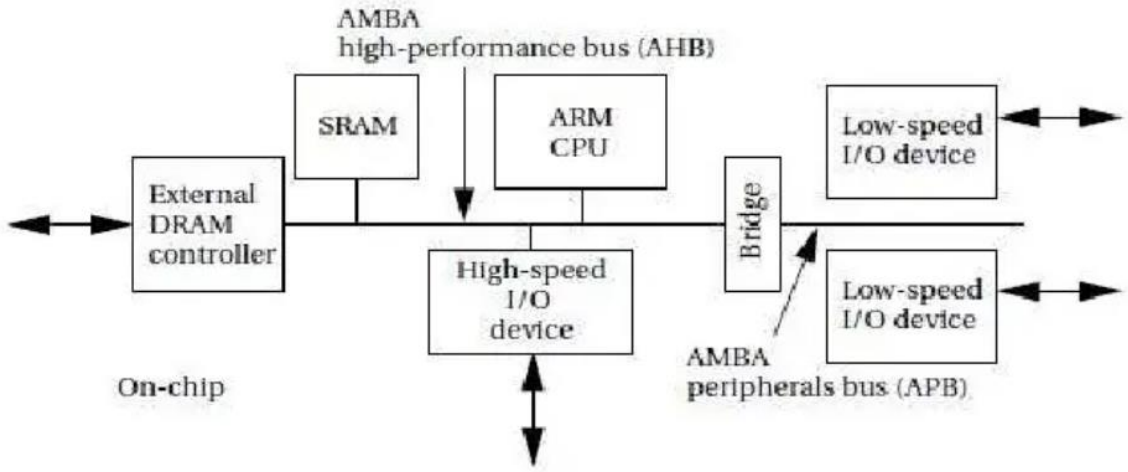
- A starting address register specifies where the transfer is to begin.
- A length register specifies the number of words to be transferred.
- A status register allows the DMA controller to be operated by the CPU.

9. What are the three reasons for a bridge allows the bus to connect to each other.

A small block of logic known as a ***bridge*** allows the buses to connect to each other. There are three reasons to do this:

- Higher-speed buses may provide wider data connections.
- A high-speed bus usually requires more expensive circuits and connectors. The cost of low-speed devices can be held down by using a lower-speed, lower-cost bus.
- The bridge may allow the buses to operate independently, thereby providing some parallelism in I/O operations.

10. Construct the ARM AMBA bus system. (Apr-2014)



DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

11.Explain about ARM bus Systems?

ARM has created a separate bus specification for single-chip systems. The AMBA bus [ARM99A] supports CPUs, memories, and peripherals integrated in a system-on-silicon. The AMBA specification includes two buses. The AMBA high-performance bus (AHB) is optimized for high-speed transfers and is directly connected to the CPU. It supports several high-performance features: pipelining, burst transfers, split transactions, and multiple bus masters.

A bridge can be used to connect the AHB to an AMBA peripherals bus (APB). This bus is designed to be simple and easy to implement; it also consumes relatively little power. The AHB assumes that all peripherals act as slaves, simplifying the logic required in both the peripherals and the bus controller. It also does not perform pipelined operations, which simplifies the bus logic.

12.List the types of DRAM.

Many types of DRAM are available. Each has its own characteristics, usually centering on how the memory is accessed.

- Synchronous DRAM (SDRAM)

- Extended data out DRAM (EDO DRAM)
- Fast page mode DRAM (FPM DRAM)
- Double data rate DRAM (DDR DRAM)

13. Define boot-block flash. (Nov/Dec-2014)

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***.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

14. Write note on intellectual property.

Intellectual property (IP) is something that we can own but not touch: software, net lists, and so on. Some examples of the wide range of IP that we use in embedded system design are,

- Run-time software libraries
- Software development environment
- Schematics, net lists, and other hardware design information.

15. Define cross-compiler.

A ***cross-compiler*** is a compiler that runs on one type of machine but generates code for another. After compilation, the executable code is downloaded to the embedded system by a serial link or perhaps burned in a PROM and plugged in.

16.What is test bench program? (Nov/Dec-2012)

A ***testbench program*** can be built to help debug the embedded code. The ***testbench*** generates inputs to simulate the actions of the input devices; it may also take the output values and compare them against expected values, providing valuable early debugging help. The embedded code may need to be slightly modified to work with the testbench, but careful coding (such as using the #ifdef directive in C) can ensure that the changes can be undone easily and without introducing bugs.

17.What are the important debugging tools available?

The two debugging tools are USB port & breakpoint.

18.Define In-circuit emulator (ICE).

The ***microprocessor in-circuit emulator (ICE)*** is a specialized hardware tool that can help debug software in a working embedded system. At the heart of an in-circuit emulator is a special version of the microprocessor that allows its internal registers to be read out when it is stopped. The in-circuit emulator surrounds this specialized microprocessor with additional logic that allows the user to specify breakpoints and examine and modify the CPU state.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

19.Write short note on Logic analyzer. (Nov/Dec-2012)

The ***logic analyzer*** [Ald73] is the other major piece of instrumentation in the embedded system designer's arsenal. Think of a logic analyzer as an array of inexpensive oscilloscopes—the analyzer can sample many different signals simultaneously (tens to hundreds) but can display only 0, 1, or changing values for each. All these logic analysis channels can be connected to the system to record the activity on many signals simultaneously.

20. What are the two modes that a typical logic analyzer can acquire data?

A typical logic analyzer can acquire data in either of two modes that are typically called ***state*** and ***timing modes.*** State and timing mode represent different ways of sampling the values. Timing mode uses an internal clock that is fast enough to take several samples per clock period in a typical system.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

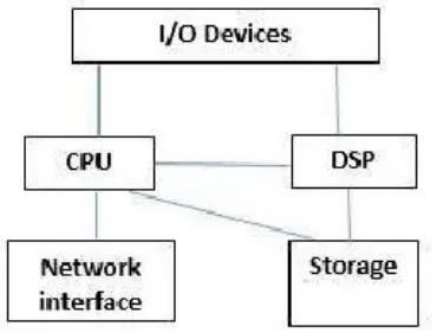
REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

21.Draw the hardware architecture of a generic consumer electronics device.



22.What is flash memory? (Nov/Dec-2014)

Many consumer electronic devices use flash memory for mass storage. Flash memory is a type of semiconductor memory that, unlike DRAM or SRAM, provides permanent storage. Values are stored in the flash memory cell as an electric charge using a specialized capacitor that can store the charge for years. The flash memory cell does not required an external power supply to maintain its value.

23. What are the steps to be performed to get data from memory to the CPU?

- Read from the memory
- Transfer over the bus to cache
- Transfer from the cache to the CPU

DMI COLLEGE OF ENGINEERING

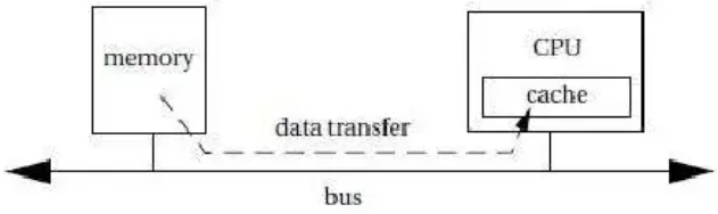
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

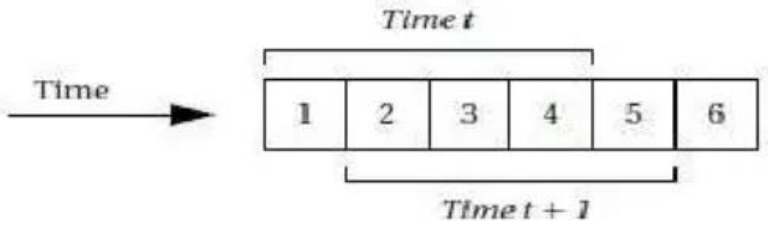
SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

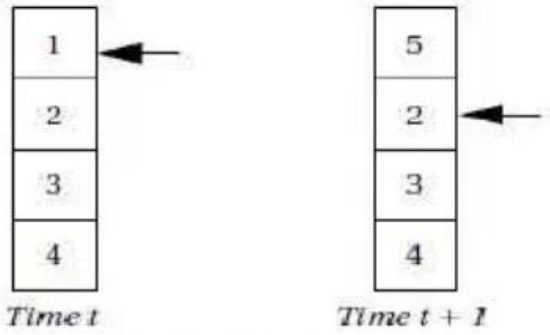


24. Define circular buffer.

The circular buffer is a data structure that lets us handle streaming data in an efficient way. Bellow figure illustrates how a circular buffer stores a subset of the data stream. At each point in time, the algorithm needs a subset of the data stream that forms a window into the stream. The window slides with time as we throw out old values no longer needed and add new values.



Data stream



Circular buffer

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

25. What is control/data flow graph (CDFG)? (Nov/Dec-2012)

Our fundamental model for programs is the ***control/data flow graph (CDFG)***. (We can also model hardware behavior with the CDFG.) As the name implies, the CDFG has constructs that model both data operations (arithmetic and other computations) and control operations (conditionals). Part of the power of the CDFG comes from its combination of control and data constructs. To understand the CDFG, we start with pure data descriptions and then extend the model to control.

26. 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.

27. Write note on Loader & Absolute address.

Loader: The program that brings the program into memory for execution is called a *loader*.

Absolute address: The simplest form of the assembler assumes that the starting address of the Assembly language program has been specified by the programmer. The addresses in such a program are known as ***absolute addresses***.

28. Define Linker. (Nov/Dec-2013)

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 links between files.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

29. Define re-entrant and re-locatable type of programming.

Re-entrant: A program is reentrant if can be interrupted by another call to the function without changing the results of either call. If the program changes the value of global variables, it may give a different answer when it is called recursively.

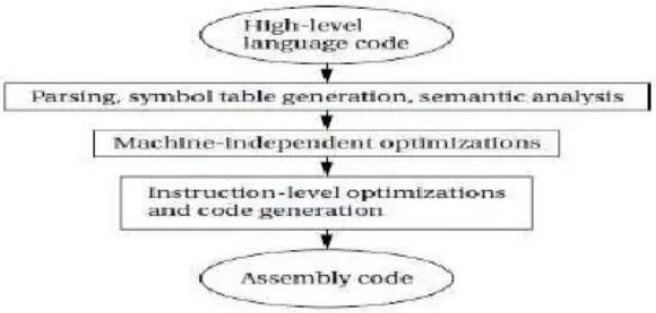
Re-locatable: A program is re-locatable if it can be executed when loaded into different parts of memory. Relocatability requires some sort of support from hardware that provides address calculation.

30. Draw the compilation process and state the formula. (Nov/Dec-2012)

Compilation Formula:

Compilation = translation + optimization

Compilation Process:



31. Define loop unrolling & Loop fusion.

A simple but useful transformation is known as ***loop unrolling***. Loop unrolling is important because it helps expose parallelism that can be used by later stages of the compiler.

Loop fusion combines two or more loops into a single loop. For this transformation to be legal, two conditions must be satisfied. First, the loops must iterate over the same values. Second, the loop bodies must not have dependencies that would be violated if they are executed together

32. Define Loop distribution & Loop tiling.

Loop distribution is the opposite of loop fusion that is, decomposing a single loop into multiple loops. ***Loop tiling*** breaks up a loop into a set of nested loops, with each inner loop performing the operations on a subset of the data.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

33.What is dead code? (Apr-2014)

Dead code is a code that can never be executed. Dead code can be generated by programmers, either inadvertently or purposefully. Dead code can also be generated by compilers. Dead code can be identified by reachability analysis-finding the other statements or instructions from which it can be reached. If a given piece of code cannot be reached, or it can be reached only by a piece of code that is unreachable from the main program, then it can be eliminated.

34.What is array padding?

Array padding adds dummy data elements to a loop in order to change the layout of the array in the cache.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

UNIT III
EMBEDDED PROGRAMMING

Syllabus:

Components for embedded programs- Models of programs- Assembly, linking and loading – compilation techniques- Program level performance analysis – Software performance optimization – Program level energy and power analysis and optimization – Analysis and optimization of program size- Program validation and testing.

Two mark questions:

1. Define Process & Threads. (Nov/Dec-2014)

A ***process*** is a single execution of a program. If we run the same program two different times, we have created two different processes. Each process has its own state that includes not only its registers but all of its memory. In some OSs, the memory management unit is used to keep each process in a separate address space. In others, particularly lightweight RTOSs, the processes run in the same address space. Processes that share the same address space are often called ***threads***.

2. Define Jitter. (Nov/Dec-2013)

Jitter of task is the allowable variation in the compilation of the task. Jitter can be important in a variety of applications: in the playback of multimedia data to avoid audio gaps or jerky images; I the control of machines to ensure that the control signal is applied at the right time.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

3. **What is CPU usage metrics?**

The simplest and most direct measure of CPU usage metrics is utilization

$$\text{Utilization} = \frac{\text{CPU time for useful work}}{\text{Total available CPU time}}$$

4. **What is preemptive operating system?**

A Preemptive real time operating system solves the fundamental problems of a cooperative multitasking system. It executes processes based upon timing requirements provided by the system designer. The most reliable way to meet timing requirements accurately to build a preemptive operating system and use priorities to control what process runs at any given time.

5. **Define kernel and time quantum. (Nov/Dec-2012)**

The kernel is the part of the operating system that determine what process is running. The kernel is activated periodically by the timer. The length of the timer period is known as the time quantum because it is the smallest increment in which we can control CPU activity.

6. **What is Context, Context Switching & record?**

The set of registers that defines a process is known as its context, and switching from one process's register set to another is known as context switching. The data structure that holds the state of the process known as the record.

7. **What is active object & active class?**

UML often refers to processes as active objects, that is, objects that have independent threads of control. The class that defines an active object is known as an active class.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

8. Write short note on round-robin scheduling. (Nov/Dec-2012)

A common scheduling algorithm is general purpose operating system is round-robin. All the processes are kept on a list and scheduled one after another. This is generally combined with preemption so that one process does not grab all the CPU time. Round-robin scheduling provides a form of fairness in that all processes get a chance to execute. However, it does not guarantee the completion time of any task; as the number of process increases, the number of response time of all the processes increases.

9. Define Rate-monotonic scheduling.

It was one of the first scheduling policies developed for real-time systems and is still very widely used. RMS is a static scheduling policy it assigns fixed priorities to processes. It turns out these fixed priorities are sufficient to effectively schedule the processes in many situations.

10.Define Rate-monotonic analysis. (Nov/Dec-2014)

The theory underlying RMS is known as rate-monotonic analysis (RMA). That are,

- All processes run periodically on a single CPU.
- Context switching time is ignored.
- There are no data dependencies between processes.
- The execution time for a process is constant.
- All deadlines are at the ends of their periods.
- The highest-priority ready process is always selected for execution.

11.Define critical instant for a process.

The critical instant for a process is defined as the instant during execution at which the task has the largest response time.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

12. State the equations for CPU utilization for a set of n tasks.

The total CPU utilization for a set of n tasks is

$$U = \sum_{i=1}^{n} \frac{T_i}{\tau_i}$$

13.Define Semaphore. (Nov/Dec-2014)

Semaphore is a protocol mechanism offered by most multitasking kernels. A semaphore 'S' is a synchronization tool which is an integer value that, apart from initialization, is accessed only through two standard atomic operations; wait and signal. Semaphores are used to

- Control access to shared resources
- Signal the occurrence of an event
- Allow two task to synchronize their activities

14.Define Mutex.

Mutex- Mutual exclusion.

It is the easiest way for tasks to communicate with each other is through shared data structure. This process is especially easy when all tasks exist in a single address space and can reference elements, such as global variables, pointers, linked lists and ring buffer. The most common methods of obtaining exclusive access to shared resources are,

- Disabling interrupts
- Performing test and set operations
- Disabling scheduling and
- Using semaphores

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

15.What is mean by priority inversion? (Nov/Dec-2013)

When tasks share a resource there is a possibility of getting into a problem, known as priority inversion problem. Priority inversion problem arises when a high priority task has to wait while a lower priority task execute. To overcome this problem, priority inheritance protocol is used.

16.Define Deadlock. (Nov/Dec-2012)

It is a situation in which a task waiting for semaphore. The release of a semaphore from a task and another different task waiting for another semaphore release to run. None of these is proceed further.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

17.What are the advantages of building ISR queues?

- Multiple function pointers are queued by the ISRs and device driving ISRs.
- If the multiple sources are provided then at the time of return from a service, the function called gets the function pointers from queue and then it executes the pointed functions.

18.Justify the term "Cyclic scheduling".

Cyclic scheduling means that the tasks from a list of ready tasks are scheduled in sequence. Thus a task that is executed first now becomes a last priority task.

19.What is dynamic program scheduling?

The software design may be such that the priorities can be rescheduled and fixed times redefined when a message or error message is received during the run.

20.List the ways in which an RTOS handles the ISR in a multitasking environment?(au-2004)

- Cyclic cooperative scheduling
- Cooperative scheduling with precedence constraints
- Cyclic cooperative scheduling with time slicing
- Preemptive scheduling
- Fixed time scheduling etc.

21.When do you use OS_ENTER_CRITICAL () and OS_EXIT_CRITICAL ()?(au-2004)

OS_ENTER_CRITICAL ()- Macro to disable interrupts
OS_EXIT_CRITICAL ()- Macro to enable interrupts

22.What are the three common model strategies that a scheduler may adapt?

- Control flow strategy
- Data flow strategy
- Control –Data flow strategy

23.What are the advantages of C++?

- A class binds all the member function together for creating objects
- A class can derive from another class also
- Methods can have same name in the inherited class(method overloading)
- Method can have same name, same number and type of arguments in the inherited class(method overloading)

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

24.Define high level language.

Programming language in which it is easier to write codes than in the assembly language and which also gives the important benefits of short development cycle for a complex and portability to system hardware modification.

25.Write the objective of a kernel. (Nov/Dec-2013)

The objectives of a kernel are

- Creation of deletion
- Processing resource request
- Allocation and de allocation
- Scheduling process communication services
- Mechanism device management

26.What is mean by thread?

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.

27.How to assign the priority to a task? (Nov/Dec-2012)

Rate monotonic scheduling has been established to assign task priority based on how often tasks executes. Simply put, tasks with the highest rate of execution are given the highest priority RMS assumption:

- All tasks are periodic
- Task do not synchronize with one another, share resources or exchange data
- The CPU must always execute the highest priority task that is ready to run.

28.Mention the two operation of mutex? (Nov/Dec-2014)

- Lock
- Unlock

29.Why does a processor system always need an "interrupt Controller"?

In many embedded systems there are more external sources for interrupts than interrupt pins on the processor. In this case, it is necessary to use an interrupt controller to provide a larger number of interrupt signals.

An interrupt controller performs several functions.

- It provides a large number of interrupt pins that can be allocated too many external devices. Typically this is at least eight and higher numbers can be supported by cascading two or more controllers together.
- It orders the interrupt pins in a priority level so that a high level interrupt will inhibit a lower level interrupt.
- This may provide registers for each interrupt pin which contain the vector number to be used during an acknowledge cycle.
- It allows the peripherals that do not have the ability to provide a vector to do so.
- This can provide interrupt masking.

30.When do you use cooperative scheduling and when preemptive?

Cooperative scheduling: A waiting task lets another task run till it finishes Preemptive scheduling: A scheduling algorithm in which a higher priority task is forced (preempted) to block by the scheduler to let a higher priority task run

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

UNIT IV
REAL TIME SYSTEMS

Syllabus:

Structure of a Real Time System — Estimating program run times – Task Assignment and Scheduling – Fault Tolerance Techniques – Reliability, Evaluation – Clock Synchronization.

Two mark questions:

1. What is the usage of mailboxes and pipes in RTOS?

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).

Message queue is same as pipe with the only difference that pipe is byte oriented while queue can be of any size.

2. What is meant by a pipe? How does a pipe differ from a queue?

In programming a "pipe" usually refers to a connection to a data source (a database). The difference between a queue and a pipe is that a queue usually caches some data from the data source while a pipe is just a connection to the data source.

3. Write the re-entrant function conditions. (Nov/Dec-2013)

- All the arguments pass the values and none of the argument is a pointer, whenever a calling function calls that function.
- When an operation is not atomic that function should not operate on any variable which is declared outside the function or which an interrupt
- Service routine uses or which is a global variable but passed by reference and not passed by value as an argument in to the function.
- Reentrant function doesn't call any other function that is not itself reentrant.

4. Why do you need a cross complier? (Nov/Dec-2012)

Two compliers are required. One compiler is for the host computer which does the development and design and also the testing and debugging. The second complier is cross a complier. The cross compiler runs on a host, but develops the machine codes for a targeted system.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

5. What is meant by RTOS?

OS for soft or hard real time tasks with scheduling with real time constraints (deadlines) using priority based scheduling, interrupt latency control, synchronization of tasks with IPCs, and predictable timing and synchronization behavior of the system.

Or

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.

Or

Define RTOS. (Nov/Dec-2014)

A real-time operating system (RTOS) is an operating system that has been developed for real-time applications. It is typically used for embedded applications, such as mobile telephones, industrial robots, or scientific research equipment.

6. What is the advantage of reentrant functions in embedded systems software?

The reentrant functions are used by several tasks at the same time, because its parameters required from stack data structure.

7. When is an RTOS needed in embedded software?(au-2004)

- An RTOS is necessary when scheduling of multiple processor, ISR.
- An RTOS is must to monitor the processor that is response time controlled and event controlled processors

8. Name the two important functions of RTOS.

- Inter Process Communication
- Multitasking

9. List out the task service functions.

- Unsigned byte OS Task create (void* task) (void *task pointer),(void*pmdata, OS_STK*task stack poniter , unsigned byte task priority)- to create a task

- Unsigned byte OSTaskSuspend(unisigned byte task priority) –to suspend(block) a task
- Unsigned byte OStaskResume (unsigned byte task priority) – to resume(unblocking task)
- Void OSTimeset 9unsigned int counts)- to set a system clock
- Unsigned int OSTimeGet (void)- To Get a system clock

10.What are the strategies used by RTOS on interrupt source calls?

- Direct call to ISR by an Interrupting Source
- Direct call to RTOS by an Interrupting Source and Temporary Suspension of a Scheduled Task
- Direct call to RTOS by an Interrupting Source and Scheduling of Tasks as well as ISRs by the RTOS

11.List the function of a Kernel. What can be the functions outside the kernel?

- Process management
- Process creation to deletion
- Processing resource requests
- Scheduling
- IPC
- Memory management
- I/O management
- Device management

12.When do use cooperative scheduling and when pre-emptive?

Co-operative scheduling: A waiting task lets another task run till it finishes.

Pre-emptive scheduling: A scheduling algorithm in which a higher priority task is forced (preempted) to block by the scheduler to let a higher priority task run.

13.What is meant by action plan? (Nov/Dec-2014)

A plan for action for the development process.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

14.What is a target system? How does this differ from the final embedded system?

A system for the targeted embedded system that is used during development phase and the final products of software and hardware are made from it.

15.Define emulator and ICE. (Nov/Dec-2013)

Emulator:

A circuit which emulates the target system

ICE:

An In Circuit Emulator for emulating the target system after connecting to processor at one end and to PC at another end

16.Compare between the assembler and dissembler.

An assembler is a program that translates the assembly mnemonics into the binary op-codes and instruction, i.e. into an executable file, called object file. It also creates a list file that can be printed.

A dissembler translates the object codes into the mnemonics form of assembly language. It helps in understanding previously made object codes.

17.What is meant by Human machine Interactions?

The Human-Machine Interface is quite literally where the human and the machine meet. It is the area of the human and the area of the machine that interact during a given task.

Interaction can include touch, sight, sound, heat transference or any other physical or cognitive function.

18.Define ROM emulator. (Nov/Dec-2012)

A circuit that helps debugs a ROM chip by simulating the ROM with RAM. The RAM circuit plugs into the ROM socket. Since RAM can be written over, whereas ROM cannot, programming changes can be made easily.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

19.What is the use of a simulator in a development phase?

It is usable during the development phase for application software for the system that is expected to employ a particular processor or processing device chip. The simulator is essentially software to simulate all functions of an embedded system circuit that includes any additional memory, peripherals devices and buses. Simulator uses the cross complier, linker, and locator like the actual target system.

20.What are a complier, linker, loader and interpreter? (Nov/Dec-2014)

Compiler - Translates High level language (C, C++ etc.) and generates Object code (machine readable but not directly executable)
Interpreter - Same as compiler but do that interactively (simply saying line by line)

Linker - Connects the compiler generated object code with library code to generated independent executable (like in C, you don't write how printf() works, linker add the code for printf() function in your program)

21.What are the various component of emulator? (Nov/Dec-2013)

- Interface circuit
- Socket
- External memory
- Emulator board display unit
- Twenty keys pad
- Registers
- Connectors

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

22. State the need for inter-process communication.

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.

23.When is an RTOS necessary and when is it not necessary in the embedded system?

An RTOS may not be necessary in a small scale embedded system. An RTOS is necessary when scheduling of multiple processor, ISR. An RTOS is must to monitor the processor that is response time controlled and event controlled processors.

24. What are the functions of queues? (Nov/Dec-2013)

- Creating a queue for an IPC
- Waiting for an IPC message at a queue
- Emptying the queue and eliminating all
- Sending a message pointer to the queue
- Sending a message pointer and inserting it at the queue front
- Querying to find the message and error information for the queue ECB.

25.Define software interrupt.

An interrupt by an instruction, by a software timer or by an error condition trap or illegal op-code.

26.Define transfer rate. (Nov/Dec-2012)

The bandwidth or maximum theoretical throughput of the front-side bus is determined by the product of the width of its data path, its clock frequency (cycles per second) and the number of data transfers it performs per clock cycle. For example, a 64-bit (8-byte) wide FSB operating at a frequency of 100 MHz that performs 4 transfers per cycle has a bandwidth of 3200 megabytes per second (MB/s)

27.List the best strategies for synchronization between the tasks and ISRs?

- Use appropriate precedence assignment strategy.
- The ISR coding should be like a reentrant function with no shared data problem. The ISR should be short and execute the codes that should not wait for actions by the RTOS and tasks.
- A task should not call another task as each task has to be under the RTOS control. Such an attempt should generate an error.
- A task can get the message and send the messages using the RTOS calls only.
- Semaphores, queues, and messages should not be globally shared variables, and each should be shared between a set of tasks only and encapsulated from the rest.

28.What is meant by watchdog timer? (Nov/Dec-2012)

Most embedded systems have no provision for resetting the processor. To facilitate the processor resetting, a watchdog timer is used.

This timer is set to a large value and is counted down. When the value reaches zero, the processor is reset. If everything is going fine and there is no need to reset the processor, the processor resets the watchdog timer to a large value again.

This procedure takes care of undiscovered bugs in the software. The watchdog Timer's use is to support time outs in a program while keeping the program structure simple.

29.What are the advantages of time slice scheduling by an RTOS?

Time slicing scheduling means that each task is allotted a time slice after which it blocks and wait for its turn on the next cycle. Cyclic scheduling with time slicing is simple and there is no insertion or deletion into the queue or list.

30.Write the condition for pre-emption events occur? (Nov/Dec-2014)

- Preemption event takes place when an interrupts occurs
- Preemption event takes when an RTCSWT(Real Time Clock driven Software Timer) interrupt occurs at the RTOS
- Preemption event takes place when any call to the RTOS occurs to enter the critical section.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

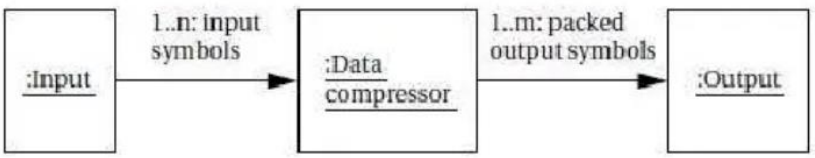
UNIT V
PROCESSES AND OPERATING SYSTEMS

Syllabus:

Introduction – Multiple tasks and multiple processes – Multirate systems- Preemptive real-time operating systems- Priority based scheduling- Inter-process communication mechanisms – Evaluating operating system performance- power optimization strategies for processes – Example Real time operating systems- POSIX-Windows CE. - Distributed embedded systems – MPSoCs and shared memory multiprocessors. – Design Example - Audio player, Engine control unit – Video accelerator.

Two mark questions:

1. **Illustrate the UML Collaboration diagram for the data compressor.**



2. **What are the additional behavior we provided for a fully functional system data compressor? (Nov/Dec-2012)**

For a fully functional system, we have to provide the following additional behavior.

- We have to be able to provide the compressor with a new symbol table.
- We should be able to flush the symbol buffer to cause the system to release all pending symbols that have been partially packed. We may want to do this when we change the symbol table or in the middle of an encoding session to keep a transmitter busy.

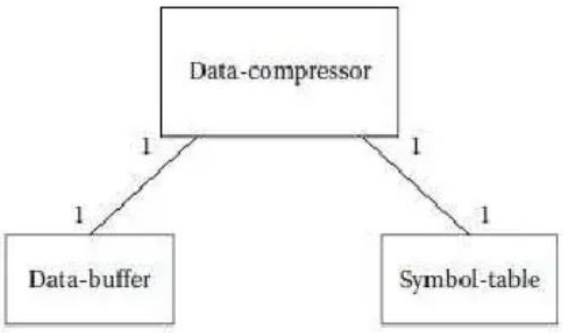
3. What are the three methods for table attributes in data compressor?

The class has three methods as follows:

- ***Encode*** performs the basic encoding function. It takes in a 1-byte input symbol and returns two values: a Boolean showing whether it is returning a full buffer and, if the Boolean is true, the full buffer itself.
- ***New-symbol-table*** installs a new symbol table into the object and throws away the current contents of the internal buffer.

- ***Flush*** returns the current state of the buffer, including the number of valid bits in the buffer.

4. Draw the diagram for relationships between classes in the data compressor. (Nov/Dec-2014)



5. State the requirements for alarm clock.

The time is shown as four digits in 12-h format; we use a light to distinguish between AM and PM. We use several buttons to set the clock time and alarm time. When we press the *hour* and *minute* buttons, we advance the hour and minute, respectively, by one. When setting the time, we must hold down the *set time* button while we hit the *hour* and *minute* buttons; the *set alarm* button works in a similar fashion. We turn the alarm on and off with the *alarm on* and *alarm off* buttons. When the alarm is activated, the *alarm ready* light is on. A separate speaker provides the audible alarm.

6. Draw the class diagram for alarm clock. (Nov/Dec-2013)

DMI COLLEGE OF ENGINEERING

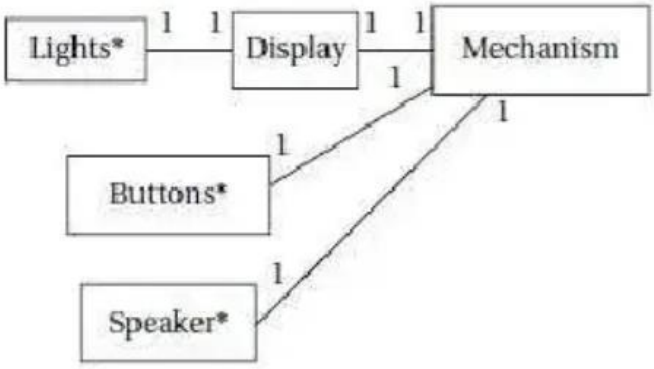
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

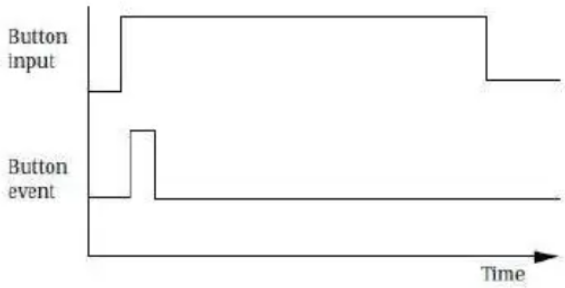
YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK



7. Illustrate preprocessing button inputs for alarm clock.



8. What are the three types of test has to be performed in Alarm clock?

Three types of tests can be performed. First, the clock's accuracy can be checked against a reference clock. Second, the commands can be exercised from the buttons. Finally, the buzzer's functionality should be verified.

9. State the theory of operations and basic functions of audio player.

Audio players are often called ***MP3 players*** after the popular audio data format. The earliest portable MP3 players were based on compact disc mechanisms. Modern MP3 players use either flash memory or disk drives to store music. An MP3 player performs three basic functions: audio storage, audio decompression, and user interface.

10.Define perceptual coding. (Nov/Dec-2012)

Audio compression is a lossy process that relies on perceptual coding. The coder eliminates certain features of the audio stream so that the result can be encoded in fewer bits. It tries to eliminate features that are not easily perceived by the human audio system. Masking is one perceptual phenomenon that is exploited by perceptual coding.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

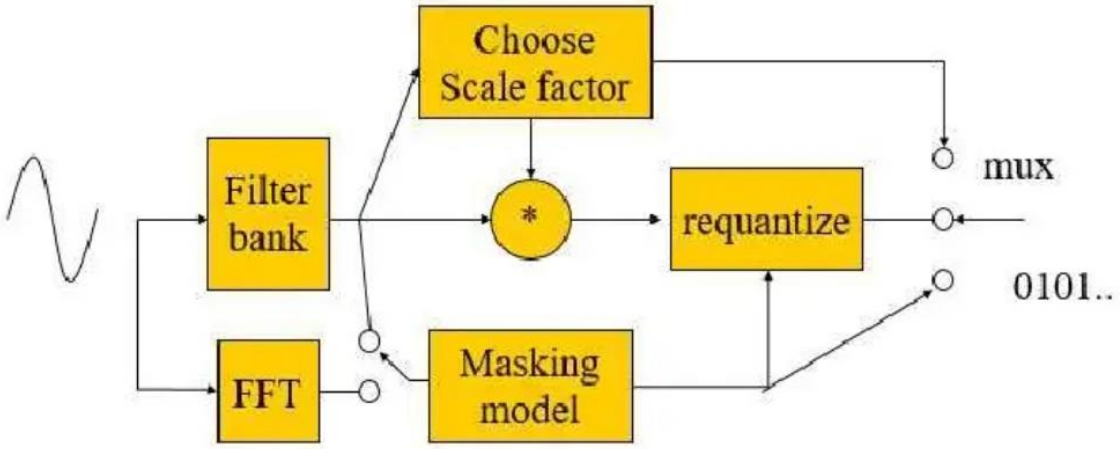
SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

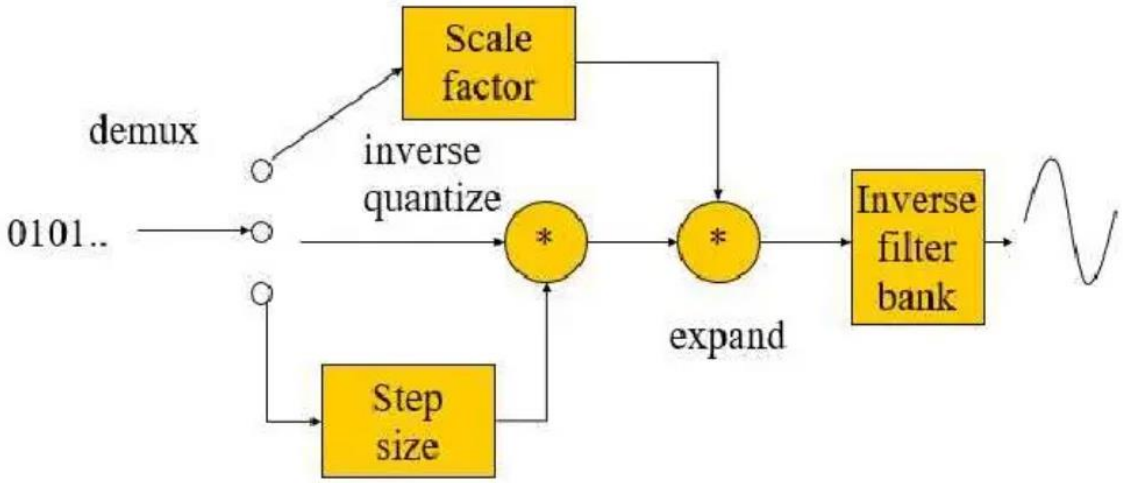
11.What are the three layers of audio compression standards?

- Layer 1: (MP1) uses a lossless compression of sub-bands and an optional, simple masking model.
- Layer 2: (MP2) uses a more advanced masking model.
- Layer 3: (MP3) Performs additional processing to provide lower bit rates.

12.Figure out the MPEG Layer 1 encoder diagram. (Nov/Dec-2013)



13.Figure out the MPEG Layer 1 encoder diagram.



DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

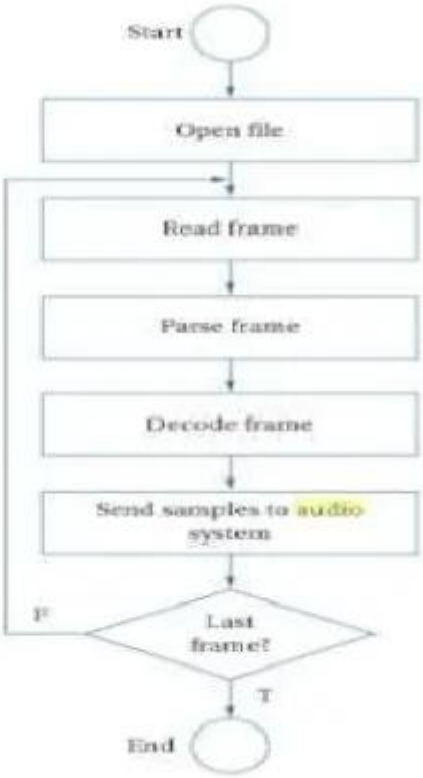
SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

14.Illustrate the MPEG Layer1 data frame format. (Nov/Dec-2012)

header	CRC	bit allocation	scale factors	subband samples	aux data

15.Draw the state diagram for audio playback.



DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

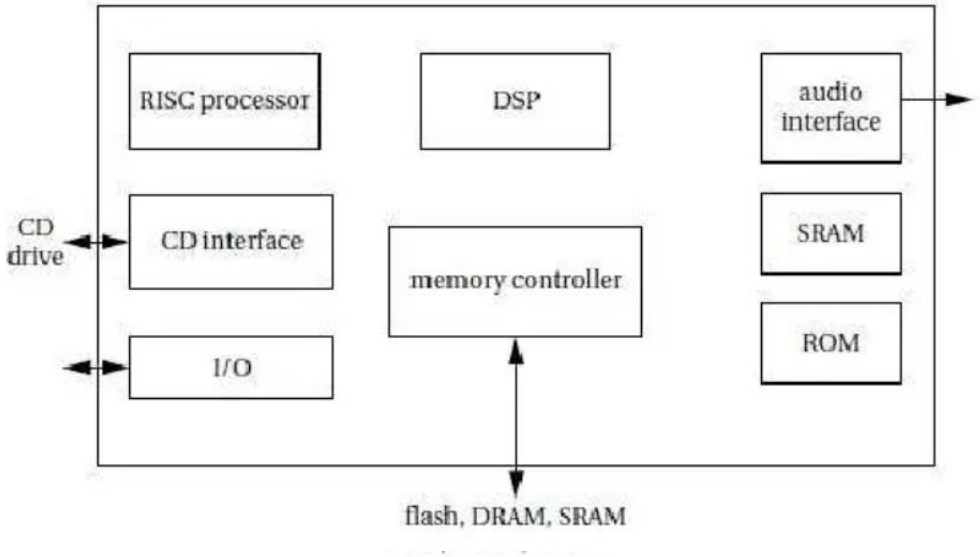
REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

16. Draw the architecture of a cirrus audio processor for CD/MP3 player.

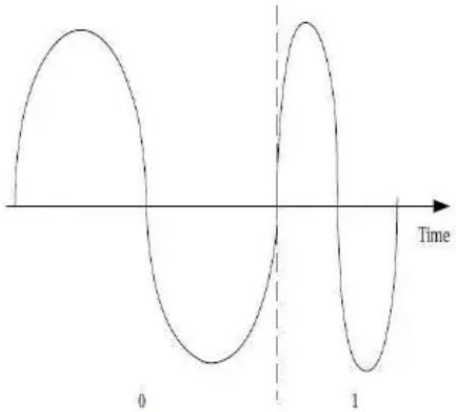


17. State the theory of operation and requirements for software modem.

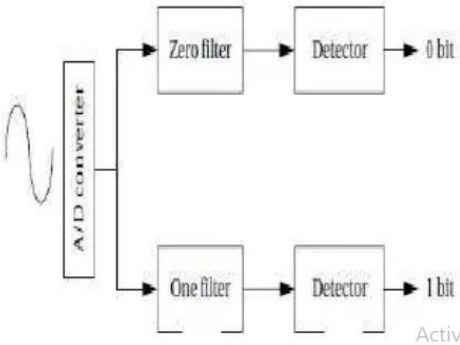
The modem will use ***frequency-shift keying (FSK)***, a technique used in 1200-baud modems. Keying alludes to Morse code—style keying. The FSK scheme transmits sinusoidal tones, with 0 and 1 assigned to different frequencies. Sinusoidal tones are much better suited to transmission over analog phone lines than are the traditional high and low voltages of digital circuits. The 01 bit patterns create the chirping sound characteristic of modems.

18. Draw the frequency shift keying and FSK detection scheme in the Software modem. (Nov/Dec-2014)

FSK:



FSK detection scheme:



DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

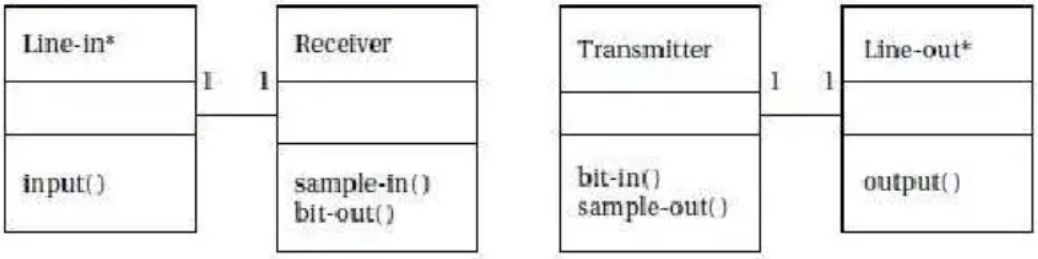
REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

19.Draw the class diagram for the modem.



20. What is loop-back testing? (Nov/Dec-2013)

Single-unit testing, called ***loop-back*** testing in the telecommunications industry, is simpler and a good first step. Loop-back can be performed in two ways. First, a shared variable can be used to directly pass data from the transmitter to the receiver. Second, an audio cable can be used to plug the analog output to the analog input. In this case it is also possible to inject analog noise to test the resiliency of the detection algorithm.

21.What are the steps performed by the modern digital camera?

Digital still cameras must perform many functions:

- It must determine the proper exposure for the photo.
- It must display a preview of the picture for framing.
- It must capture the image from the image sensor.

- It must transform the image into usable form.
- It must convert the image into a usable format, such as JPEG, and store the image in a file system.

22.What is Pixels, Luminance & Chrominance? (Nov/Dec-2012)

The image is divided into pixels; a pixels brightness is often referred to as its luminance; a color pixels brightness in a particular color is known as chrominance.

23.Define histogram.

The histogram is composed by sorting the pixels into bins by luminance; 265 bins is a common choice for the resolution of the histogram. The histogram gives us more information than does a single average.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

24.What are major approaches are used to determine focus in digital camera?

The three major approaches are used to determine focus that are,

- Active range finding
- Phase detection
- Contrast detection

25.What are the major types of image sensors used in modern camera?

The major types of image sensors used in the modern cameras are

- Charged coupled devices (CCDs)
- CMOS

26.Define Bayer pattern. (Nov/Dec-2013)

A **Bayer filter** mosaic is a color filter array (CFA) for arranging RGB color filters on a square grid of photo sensors. Its particular arrangement of color filters is used in most single-chip digital image sensors used in digital cameras, camcorders, and scanners to create a color image. The filter pattern is 50% green, 25% red and 25% blue, hence is also called **RGBG**, **GRGB**, or **RGGB**.

27.What are the five major steps used for JPEG images in the compression process?

- Color space conversion
- Color down sampling
- Block-based discrete cosine transform (DCT)
- Quantization
- Entropy coding

28.Draw the block diagram of ADPCM compression system.

DMI COLLEGE OF ENGINEERING

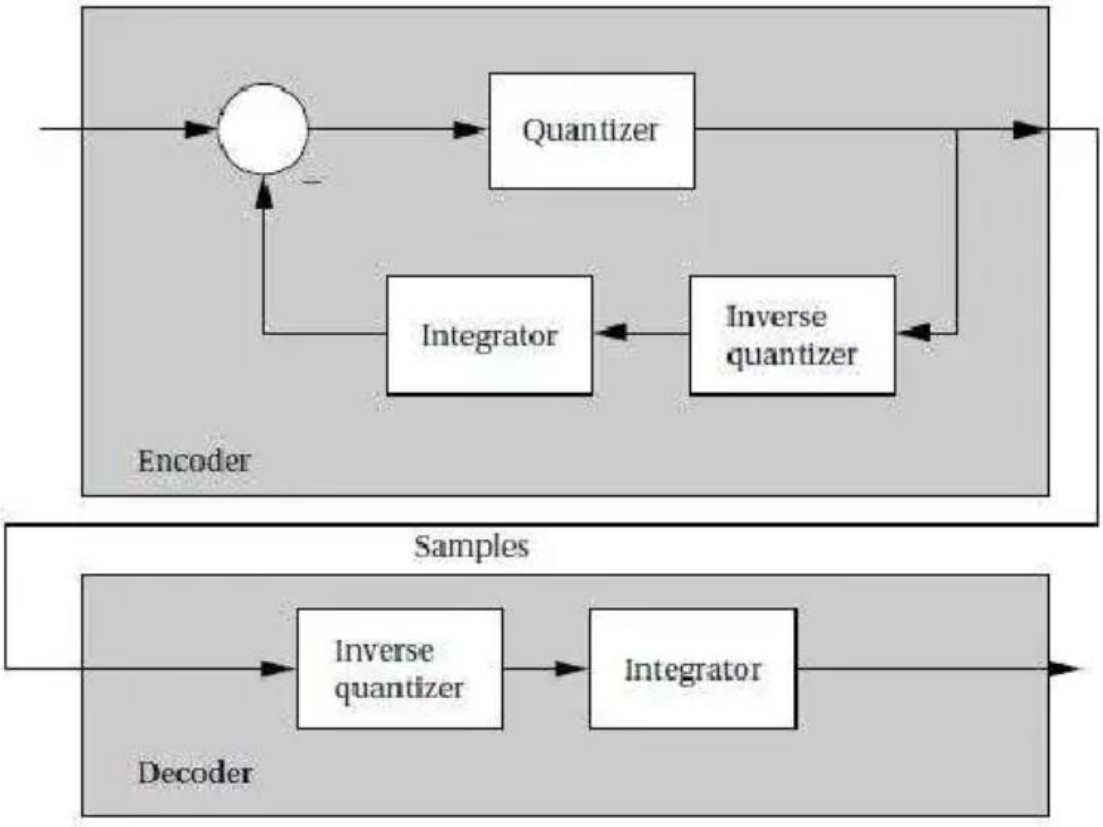
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK



29. Define Block-motion estimation.

JPEG-style of compression alone does not reduce video bandwidth enough for many applications.MPEG uses motion to encode one frame in terms of another. Rather than send each frame separately, as in motion JPEG, some frames are sent as modified forms of other frames using a technique known as block motion estimation.

DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

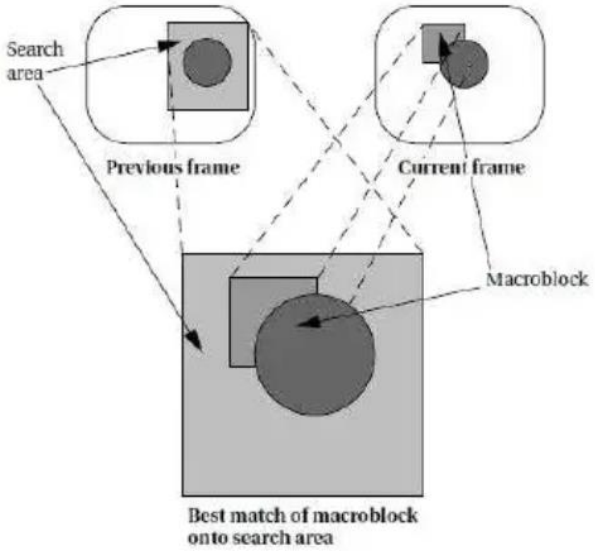
REGULATION 2017

YEAR/SEM:IV/VII

SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIMESYSTEMS

QUESTION BANK

30. Draw the block diagram for block-motion estimation. (Nov/Dec-2014)



DMI COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

REGULATION 2017

YEAR/SEM:IV/VII SUBJECT CODE &NAME:EC8791-EMBEDDED AND REAL TIME SYSTEMS

QUESTION BANK

UNIT -1 INTRODUCTION TO EMBEDDED SYSTEM DESIGN

1. **Show the characteristics of embedded system for computing applications**
   - Complex algorithms
   - User interface
   - Real time
   - Multirate
2. **Recall the details available in non-requirement form analysis**
   - Performance→ depends upon approximate time to perform a user-level function and also operation must be completed within deadline.
   - Cost→Manufacturing cost includes the cost of components and assembly.
   - Physical Size and Weight→The final system can vary depending upon the application.
   - Power Consumption→Power can be specified in the requirements stage in terms of battery life.
3. **Infer the main goal of a design process with an example**
   - The main goal of a design process is to create a product that does something useful.
   - Typical specifications for a product are functionality , manufacturing cost, performance and power consumption.
4. **Show the purpose of using capability maturity model**
   - It is used to measuring the quality of an organization's software development
   - It was created to apply to a broad range of industries, including limited to embedded software.
5. **What are all the factors involved to measure the system-level performance analysis**
   - System-Level Performance involves much more than the CPU.
   - To move data from memory to the CPU to process it. To get the data from memory to the CPU we must.

6. **List out the challenges involved embedded system design.**
    - To meet performance deadlines and manufacturing cost constraints, the choice of Hardware is important.
    - To speed up the hardware so that the program runs faster. But the system more expensive.
    - In battery-powered applications, power consumption is extremely important.
      In non-battery applications, excessive power consumption can increase heat dissipation

7. **Extend the design flow used in the embedded system design**
    - The main goal of a design process is to create a product that does something useful.
    - Typical specifications for a product are functionality , manufacturing cost, performance and power consumption.

8. **Write the requirement form for GPS moving map**

Name	**GPS moving map**
Purpose	Consumer-grade moving map for driving use
Inputs	Power button, two control buttons
Outputs	Back-lit LCD display 400 600
Performance	Updates screen within 0.25 seconds upon movement

9. **Recall the details available in requirement form analysis**
    - Name→Giving a name to the project
    - Purpose→Brief one- or two-line description of what the system is supposed to do.
    - Inputs& Outputs →Analog electronic signals? Digital data? Mechanical inputs?
    - Functions→ detailed description of what the system does

10. **Show the difference between association and aggregation**
    - **Association** →occurs between objects that communicate with each other but have no ownership relationship between them.
    - **Aggregation**→ describes a complex object made of smaller objects.

PART-B

1. Discuss the different factors involved in embedded system design process.

Design process has two objectives as follows.

1. It will give us an introduction to the various steps in embedded system design.
2. Design methodology

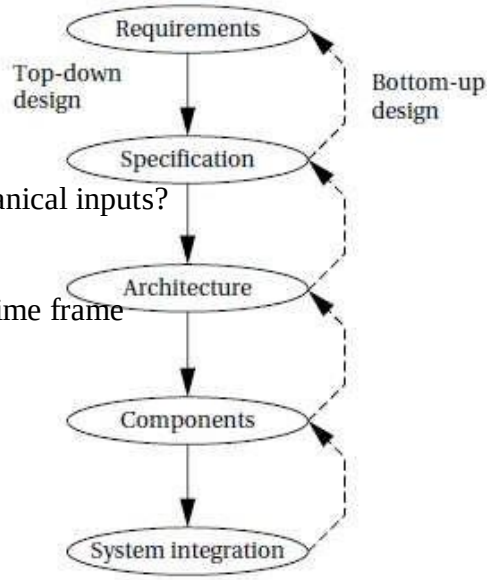
- Design to ensure that we have done everything we need to do, such as optimizing performance or performing functional tests.
- It allows us to develop computer-aided design tools.
- A design methodology makes it much easier for members of a design team to communicate.
- **Levels of abstraction in the design process.**

1) Requirements

- It can be classified in to functional or nonfunctional

1.1) Functional Requirements

- Gather an informal description from the customers.
- Refine the requirements into a specification that contains enough information to design the system architecture.
- Ex:Sample Requirements form
- Name→Giving a name to the project
  Purpose→Brief one- or two-line description of what the system is supposed to do.
- Inputs& Outputs →Analog electronic signals? Digital data? Mechanical inputs?
- Functions→ detailed description of what the system does
  Performance→ computations must be performed within a certain time frame
- Manufacturing cost→ cost of the hardware components.
  Power→ how much power the system can consume
  Physical size and weight→ indication of the physical size of the system



1.2) Non-Functional Requirements

- Performance→ depends upon approximate time to perform a user-level function and also operation must be completed within deadline.
- Cost→Manufacturing cost includes the cost of components and assembly.
- Nonrecurring engineering (NRE) costs include the personnel and other costs of designing the
- system
- Physical Size and Weight→The final system can vary depending upon the application.
- Power Consumption→Power can be specified in the requirements stage in terms of battery life.

2) SPECIFICATION

- The specification must be carefully written so that it accurately reflects the customer's requirements.
- It can be clearly followed during design.

3) Architecture Design

- The architecture is a plan for the overall structure of the system.
- It is in the form block diagram that shows a major operation and data flow.

4) Designing Hardware and Software Components

- The architectural description tells us what components we need include both hardware—FPGAs, boards & software modules

5) System Integration

- Only after the components are built, putting them together and seeing a working system.
- Bugs are found during system integration, and good planning can help us find the bugs quickly.

2. Analyze the requirements for designing a GPS moving map in embedded system design process

1. Requirements analysis of a GPS moving map

- The moving map is a handheld device that displays for the user a map of the terrain around the user's current position.
- The map display changes as the user and the map device change position.
- The moving map obtains its position from the GPS, a satellite-based navigation system.

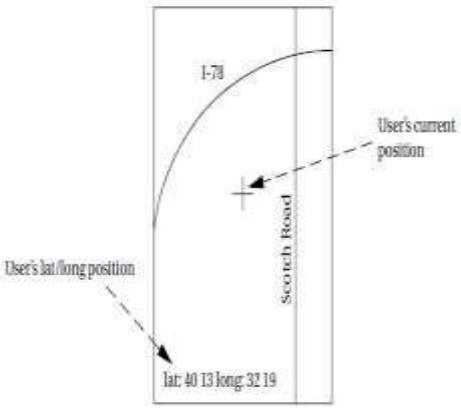
Name	GPS moving map
Purpose	Consumer-grade moving map for driving use
Inputs	Power button, two control buttons
Outputs	Back-lit LCD display 400 600
Functions	Uses 5-receiver GPS system; three user-selectable resolutions;always displays current latitude and Longitude
Performance	Updates screen within 0.25 seconds upon movement
Manufacturing cost	$30
Power	100mW
Physical size and weight	No more than 2"X 6, *" 12 ounces*

2) Functionality→This system is designed for highway driving and similar uses.

The system should show major roads and other landmarks available in standard topographic databases.

3)User interface→The screen should have at least 400X600 pixel resolution.
The device should be controlled by no more than 3 buttons.

→A menu system should pop up on the screen when buttons are pressed to allow the user to make selections to control the system.



4)Performance→ The map should scroll smoothly.

→Upon power-up, a display should take no more than 1sec to appear.

→The system should be able to verify its position and display the current map within 15 s.

5)Cost→ The selling cost of the unit should be no more than $100.

6)Physical size and weight→The device should fit comfortably in the palm of the hand.

7) Power consumption→ The device run for at least 8 hrs on 4 AA batteries.

8) specification

1. Data received from the GPS satellite constellation.
2. Map data.

3. User interface.
4. Operations that must be performed to satisfy customer requests.

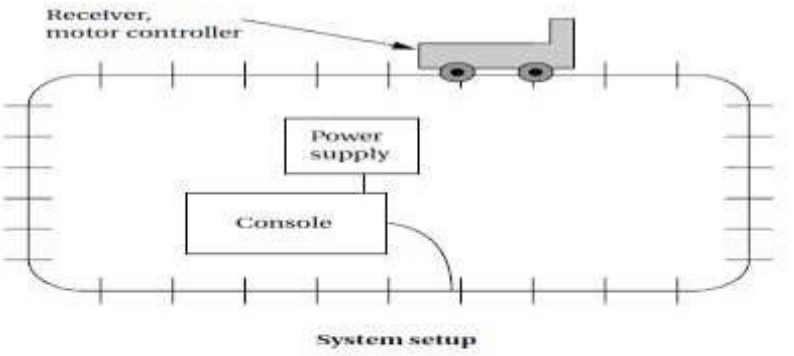
Background actions required to keep the system running, such as operating the GPS receiver

3. Discuss in detail the design steps Model train controller with the frame format of DCC.

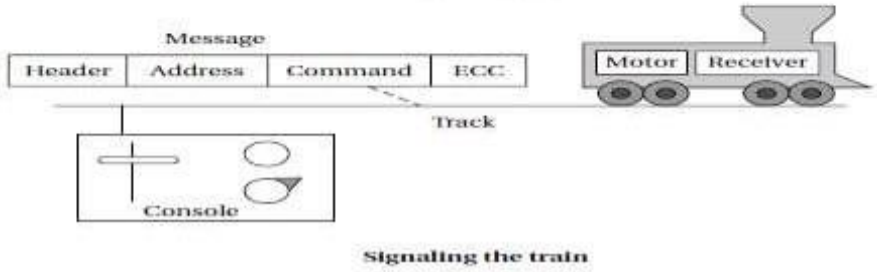
- In order to learn how to use UML to model systems→ specify a simple system (Ex: model train controller)
- The user sends messages to the train with a control box attached to the tracks.
- The control box may have controls such as a throttle, emergency stop button, and so on.
- The train Rx its electrical power from the two rails of the track.

CONSOLE

- Each packet includes an address so that the console can control several trains on the same track.
- The packet also includes an error correction code (ECC) to guard against transmission errors.
- This is a one-way communication system—the model train cannot send commands back to the user.



System setup



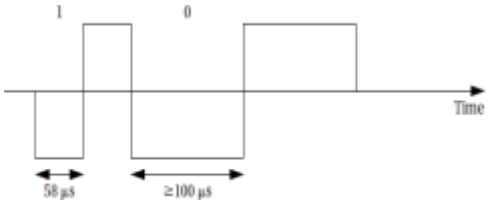
Signaling the train

REQUIREMENTS

- The console shall be able to control up to eight trains on a single track.
- The speed of each train controllable by a throttle to at least 63 different levels in each direction (forward and reverse).
- There shall be an inertia control→ to adjust the speed of train.
- There shall be an emergency stop button.
- An error detection scheme will be used to transmit messages.

Digital Command Control (DCC)

- Standard S-9.1→ how bits are encoded on the rails for transmission.
- Standard S-9.2→ defines the packets that carry information.
- The signal encoding system should not interfere with power transmission



- Data signal should not change the DC value of the rails.
- Bits are encoded in the time between transitions.
- Bit 0 is at least 100 s while bit 1 is nominally 58 s.

Packet Formation in DCC

- The basic packet format is given by $\mathrm{PSA(sD) + E}$
- P → preamble, which is a sequence of at least 10 1 bits.
- S → packet start bit. It is a 0 bit.
- A →address is 8 bits long. The addresses 00000000, 11111110, and 11111111 are reserved.
- s→ data byte start bit, which, like the packet start bit, is a 0.
- D →data byte includes 8 bits. A data byte may contain an address, instruction, data, or error correction information.
- E →packet end bit, which is a 1 bit.

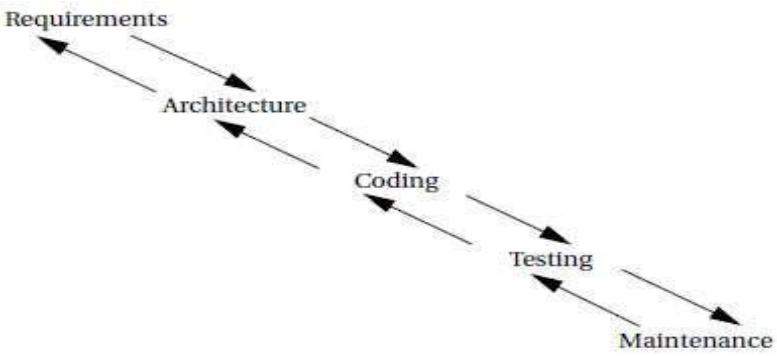
Baseline packet

- The minimum packet that must be accepted by all DCC implementations.

- It has three data bytes.
- Address data byte→ gives the intended receiver of the packet
- Instruction data byte→ provides a basic instruction
- Error correction data byte→ is used to detect and correct transmission errors.

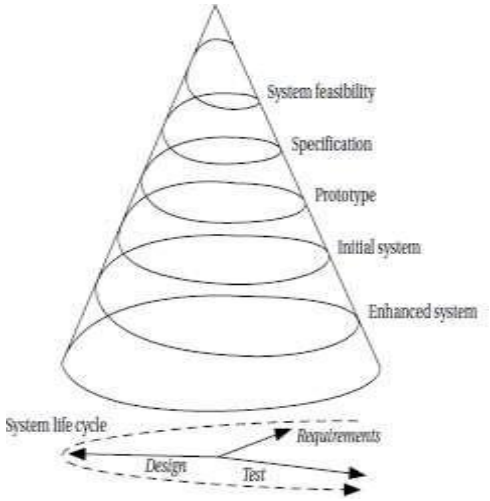
4. Explain about the design methodology of an embedded computing system in detail.

1)Waterfall model



- The waterfall development model consists of five major phases.
- Requirements analysis→ determines the basic characteristics of the system.
- Architecture design→It decomposes the functionality into major components
- Coding→It implements the pieces and integrates them.
- Testing→It detemines bugs.
- Maintenance→ It entails deployment in the field, bug fixes,and upgrades.
- The waterfall model makes work flow information from higher levels of abstraction to more detailed design steps.

2)Spiral model

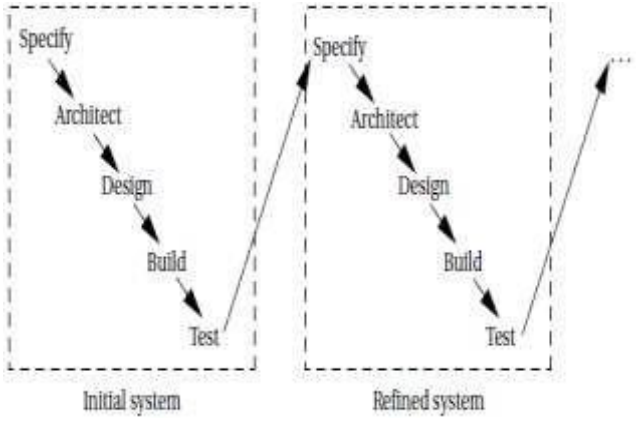


- The spiral model assumes that several versions of the system will be built.
- Each level of design, the designers go through requirements,construction,and testing phases.
- At later stages when more complete versions of the system are constructed.
- Each phase requires more work, widening the design spiral.

- The first cycles at the top of the spiral are very small and short.
- The final cycles at the spiral's bottom add detail learned from the earlier cycles of the spiral.
- The spiral model is more realistic than the waterfall model because multiple iterations needed to complete a design.
- But too many spirals may take long time required for design.

3) Successive refinement design model

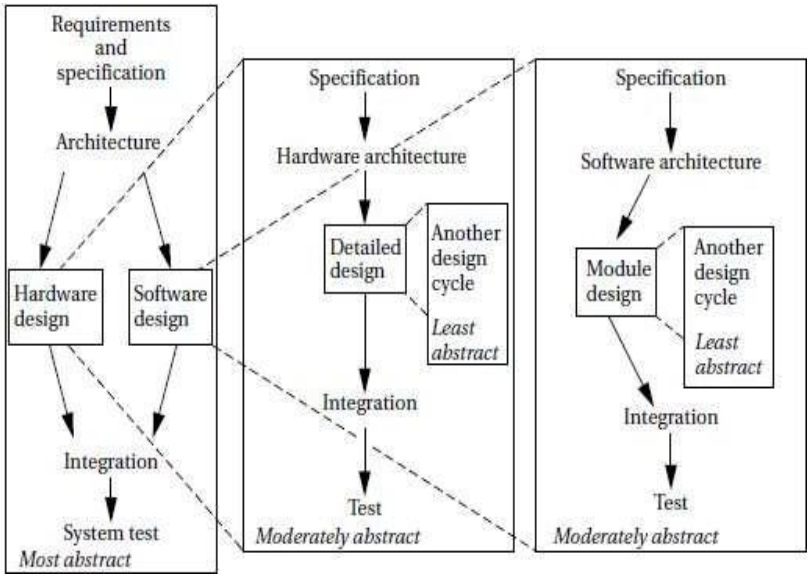
- In this approach, the system is built several times.
- A first system is used as a rough prototype.
- Embedded computing systems are involved the design of hardware/software project.
- Front-end activities→ are specification and architecture and also includes hardware and software aspects.



- Back-end activities → includes integration and testing.
- Middle activities→includes hardware and software development.

4) Hierarchical design flow

- Many complex embedded systems are built of smaller designs.
- The complete system may require the design of significant software components.
- It has many levels of abstraction to design flows for individual components.



- The implementation phase contains a complete flow from specification through testing.
- Each flow will probably be handled by separate people or teams.
- The teams must rely on each other's results.
- The component teams take their requirements from team handling the next higher level of abstraction.
- The higher-level team relies on the quality of design and testing performed by the component team.

5. Observe in detail Quality Assurance Process using the following

i) Quality Assurance Techniques

- The quality assurance (QA) process is vital for the delivery of a satisfactory system.
- International Standards Organization (ISO) has created a set of quality standards known as ISO 9000.
- It was created to apply to a broad range of industries, including limited to embedded hardware and software.

ISO 9000 quality management parameters

- Process is crucial→Knowing what steps are to be followed to create a high-quality product.
- Documentation is important→helps internal quality monitoring groups to ensure that the required processes and helps outside groups understand the processes and how they are being implemented.
- Communication is important→ people should understand not only their specific tasks but also how their jobs can affect overall system quality.

- **Capability Maturity Model (CMM)**
  - It is used to measuring the quality of an organization's software development.
  - Initial→A poorly organized process, with very few well-defined processes. Success of a project depends on the efforts of individuals, not the organization itself.
  - Repeatable→ provides basic tracking mechanisms to understand cost, scheduling .
  - Defined→The management and engineering processes are documented and standardized.
  - Managed→detailed measurements of the development process and product quality.

- Optimizing→feedback from detailed measurements is used to continually improve the organization's processes.

ii) Verifying the Specifications

- **Verifying the specification→**Discovering bugs early is crucial because it prevents bugs from being released to customers, minimizes design costs, and reduces design time.
- Validation of specifications→creating the requirements, including correctness, completeness, consistency, and so on
- **Design reviews**
- The review leader coordinates the pre-meeting activities, the design review itself, and the post-meeting follow-up.
- The reviewer records the minutes of the meeting so that designers and others know which problems need to be fixed.
- The review audience studies the component.

UNIT II ARM PROCESSOR AND PERIPHERALS

1. **List out the features of ARM processor**
    - It is a 32 bit processor also supports 8 and 16 bits data types.
    - It can be configured either Little-endian mode (lowest-order byte stored in the low-order bits of the word) or Big-endian mode (lowest-order byte stored in the highest bits of the word).
    - It uses a Intelligent Energy Meter(IEM) technology to optimally balancing the workload and energy consumption.
    - It use Advanced High Performance Bus interface(AMBA) for on-chip interconnect purpose.

2. **Extend the operation of ARM processor in supervisor mode**

- Provide hardware checks to ensure that the programs do not interfere with each other.
- It is provided by the CPU.
- Software debugging is important but can leave some problems in a running system,hence Hardware checks ensure an additional level of safety.
- Supervisor mode→SWI CODE_1

3. **Compare the operation of logical shift left and logical shift right**
    - Logical Shift Left (LSL)→ Logical Shift Left of a 32-bit number causes it to shift left and the vacant bits on the right are filled with zeros.
    - Logical Shift Right (LSR)→ The vacant bit positions on the left are filled with zeros, and the last bit shifted out is retained in the carry flag

4. **Outline the importance of procedure in ARM processor**
    - A procedure starts with a 'CALL' instruction. This causes the action of pushing the current value of PC onto the stack.
    - The procedure ends with a 'RETURN' instruction. This causes the PC value to be popped back
5. **Summarize the functions of pin connect block in GPIO**
    - The purpose of PCB is to configure the pins to the desired functions.
    - Each pin of the chip has a maximum of four functions.
    - To select one specific function for a pin, a multiplexer with two select pins, is necessary.
    - The select pins function is provided by the bits of the PINSEL registers.
6. **How does stack and subroutines have been performed in ARM processor?**
    - A stack is an area in memory, the accessing of which is done in a special way. Most stacks are Last-In First-Out (LIFO) type stack.
    - Two operations are defined for a stack, that is, the PUSH, in which data is written into the stack, and POP in which data is read out and loaded into registers

7. **Extend the functions of timer control register in PWM unit**
    - This is an 8-bit register in which only the lowest two bits need be used.
    - Bit 0–E→ When this Enable bit is '1', the counter is enabled and starts.
    - Bit 1–R→When Reset bit '1', the counter is reset on the next positiveedge of PCLK.

8. **Recall the about A-Pofile in ARM processor**
    - This profile which has the ARMv7-A architecture is meant for high end applications.(mobile phones and video systems)
9. **Recall the about A-Pofile in ARM processor**
    - This profile which has the ARMv7-R architecture has been designed for high-end applications which require real-time capabilities. (automatic braking systems and other safety critical applications)
10. **Mention the abort mode operation in ARM processor**
    - Abort mode→A privileged mode that is entered whenever a pre-fetch abort or data abort exception occurs

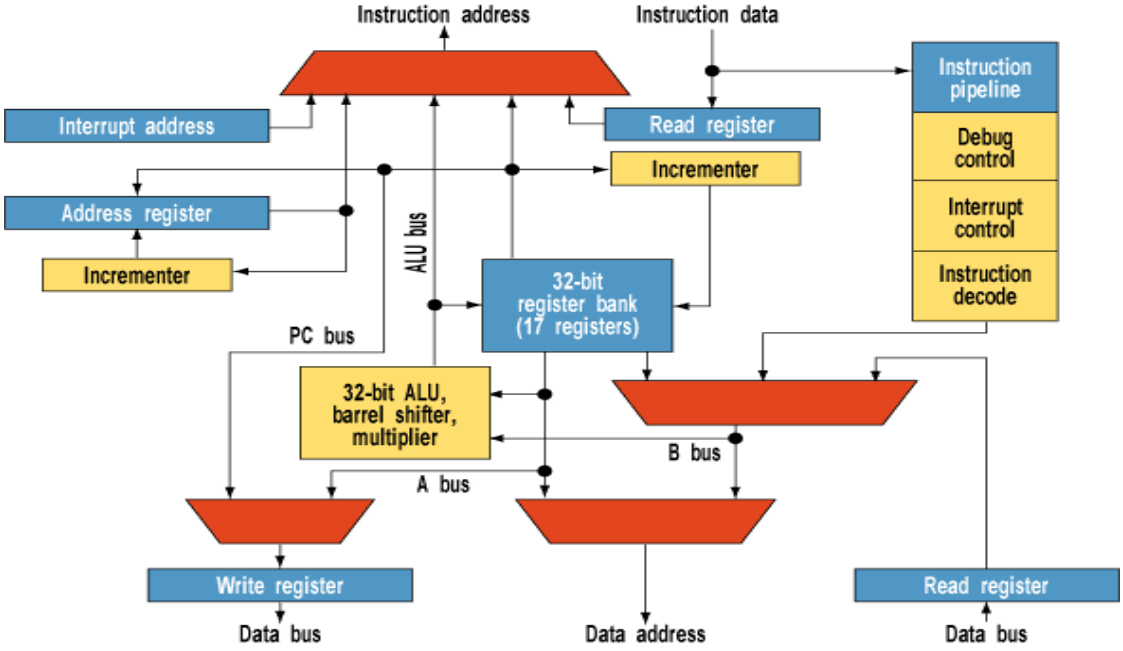
PART-B

1. **Construct the ARM processor architecture and describe the features of the same.**
    - ARM→Advanced Reduced Instruction Set Machine,describes a family of computer processor.
    - It was developed in the year 1980

    Features
    - It is a 32 bit processor also supports 8 and 16 bits data types.
    - It can be configured either Little-endian mode (lowest-order byte stored in the low-order bits of the word) or Big-endian mode (lowest-order byte stored in the highest bits of the word).
    - It can reduces the code length and fast program implementation.
    - Debug and trace quickly by using real time software.
    - Codes are compatible with higher versions
    - It uses a Intelligent Energy Meter(IEM) technology to optimally balancing the workload and energy consumption.
    - It use Advanced High Performance Bus interface(AMBA) for on-chip interconnect purpose.

ARM-Architecture

- Arrow→ represents flow of data.
- Line→ represents the buses
- Boxes→ Represent either storage area or operation unit



- Data enters the processor core through data bus. It is either data item or instruction to execute.
- If it is a instruction then the instruction decoder translates instructions before they are executed.
- If it is data then the data item are placed in the register file (32bit size).
- It have register (Rn,Rm-source register and Rd-destination register).
- Source operands are read from the register file using internal buses Aand B.
- ALU takes the register values Rn and Rm from A and B buses and write the result Rd directly to the register file using result bus.
- Load and store instructions use the ALU to generate an address and it stores in address register.

CPU modes

- User mode→ The only non-privileged mode.
- FIQ (fast Interrupt request)mode→A privileged mode that is entered whenever the processor accepts a fast interrupt request.

- IRQ mode→A privileged mode that is entered whenever the processor accepts an interrupt.
- Supervisor (svc) mode→ A privileged mode entered whenever the CPU is reset or when an SVC instruction is executed.
- Abort mode→A privileged mode that is entered whenever a pre-fetch abort or data abort exception occurs.
- Undefined mode→ A privileged mode that is entered whenever an undefined instruction exception occurs.
- System mode→It can only be entered by executing an instruction that explicitly writes to the mode bits of the Current Program Status Register (CPSR) from another privileged mode (not from user mode).

Data Operations

- In ARM processor→ Arithmetic and logical operations can't be performed directly on memory locations.
- ARM is a load-store architecture—data operands must first be loaded into the CPU and then stored back to main memory to save the results.
- Current program status register (CPSR)→ set automatically during every arithmetic, logical, or shifting operation.
- Based on the result of arithmetic/logical operation,CPSR four bits are affected as follows
- 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.

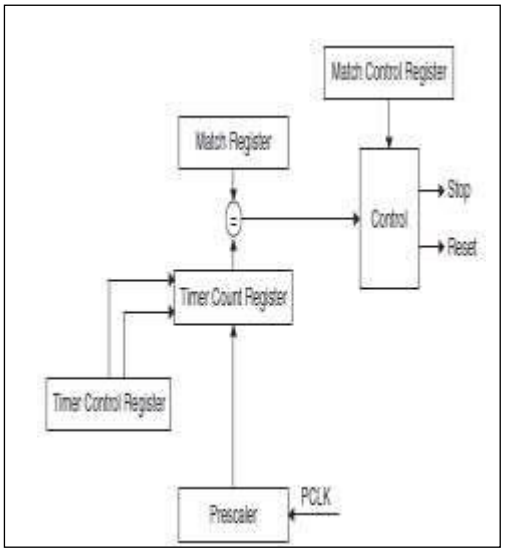
2. Examine the operation of timer unit with the relevant along with its modes of operation

- A timer and a counter are functionally equivalent, except that a timer uses the PCLK for its timing, while a counter uses an external source.

Timer Operation

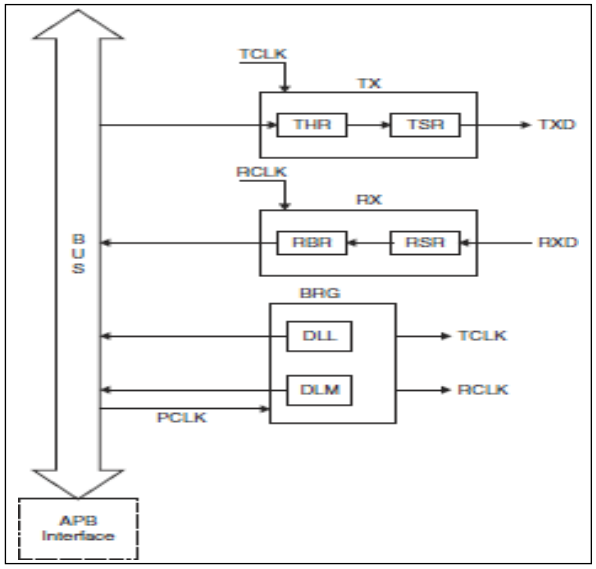
- Load a number in a match register.
- Start the timer by enabling the 'E' bit in T0TCR.

- The timer count register (T0TC) starts incrementing for every tick of the peripheral clock PCLK (no pre-scaling is done).
- When the content of the T0TC equals the value in the match register, timing is said to have occurred.
- One of many possibilities can be made to occur when this happens.
- The possibilities are to reset the timer count register, stop the timer, or generate an interrupt. This 'setting' is done in the T0MCR register.



Timer Count Register–T0TC

- This is a 32-bit register, which gives it a range of counting from 0 to 0xFFFF FFFF and then wraps back to the value 0x0000 0000.
- This register is incremented on every tick of the clock (i.e. PCLK), if the prescale counter is made 0



Timer Control Register–T0TCR

- This is an 8-bit register in which only the lowest two bits need be used.
- Bit 0–E→ When this Enable bit is '1', the counter is enabled and starts.
- Bit 1–R→When Reset bit '1', the counter is reset on the next positiveedge of PCLK.

Match Registers (MR0 to MR3)

- There are four 32-bit match registers available: MR0 to MR3.
- For the operation of one timer, one of the match registers may be sufficient and is used by loading a number into it.
- During timer operation, the timer count register starts incrementing, and at sometime, its count 'matches' with the number in the match register.

Match Control Register–T0MCR

- This is a 16-bit register used to specify the event to occur when the match occurs.
- The lowest three bits are for controlling the operations related to the Match register 0.
- The next three are for MR1, MR2 and MR3, in that order.

- **Pre-scaler**

  To generate a lower frequency output

  prescale counter increments for every PCLK, and when it counts up to the value in the prescale counter (T0PR), it allows the timer counter (T0TC) to increment its value by 1.

3. Design UART Transmitter and receiver system for ARM processor with its necessary register

- This chip has two UARTs, namely, UART0 and UART1.

The Transmitter

- When a data byte arrives in the Transmitter Holding Register (THR),(from CPU) it is 'framed' (by adding start and stop bits) and transferred to the Transmitter Shift Register (TSR)and sent out through the TxD pin one bit at a time, by clocking the
- TSR at the baud decided by the transmitter clock TCLK.

The Receiver

- The data received serially through the RxD line ,is moved bit by bit into the Receiver Shift Register (RSR), and then transferred
- to the Receiver Buff er Register (RBR). after de-framing. From the RBR it is copied to the CPU registers through the bus.

The BAUD Rate Generator (BRG)

- It takes PCLK as input, and generates the baud rates for the transmitter and receiver.

Registers of UART0

- UART0 for transferring a character string from the LPC 2148 board to a PC, using the 'hyper-terminal', at a baud of 9600.

Pinselect Register (PINSEL0)

- The pin selection for the TxD and RxD pins of UART0 are referred.
- PINSEL0 register selects pins P0.0 as TxD and P0.1 as RxD, by writing PINSEL0 = 0x5.

UART0 Transmit Holding Register (U0THR)

- It is an 8-bit register and part of the transmit buffer.
- New characters are to be loaded into this register for being transmitted.
- The data to be transmitted is written into this 'write only' register.

UART0 Divisor Latch Registers (U0DLL and U0DLM)

- The UART0 divisor Latch is part of the UART0 Fractional Baud Rate Generator and holds the value used to divide the clock supplied by the fractional prescaler in order to produce the baud rate clock, which must be 16x the desired baud rate.
- The U0DLL and U0DLM registers together form a 16-bit divisor where U0DLL contains the lower 8 bits of the divisor and U0DLM contains the higher 8 bits of the divisor.

UART0 FIFO Control Registers (U0FCR)

- Bit 0: E: This bit must be set for enabling the Tx and Rx FIFOs
- Bit 1: Rx FIFO Reset: This must be set, to clear all bytes in UART0 Rx FIFO and
- reset the pointer logic. This bit is self-clearing.
- Bit 2: Tx FIFO Reset: This must be set to clear all bytes in UART0 Tx FIFO and
- reset the pointer logic. This bit is self-clearing.
- Bits 7 and 6: Rx trigger level: These two bits determine how many receiver FIFO characters must be written before an interrupt is activated.

UART0 Line Control Register (U0LCR)

- It is an 8-bit register which determines the format of the data character that is to be transmitted or received.
- Bits 1: 0 These two bits have been chosen to be '11' to indicate 8-bit character length
- Bit 2: This is made '0' to select one stop bit
- Bit 7: This is the Divisor Latch Access Bit (DLAB) and is set, to enable the use of the divisor latch

UART0 Line Status Register (U0LSR)

- It is a read-only register that provides status information regarding the UART0 TX and RX blocks.

4. **Evaluate the performance Pulse width modulator along with its control registers unit**.

- Pulse width modulation which it is possible to control the period and duty cycle of a square wave.

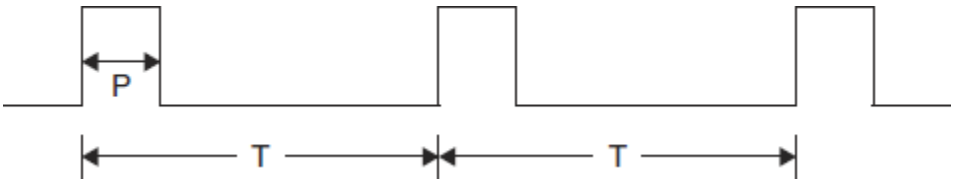
Single Edge Controlled PWM

i) All single edge controlled PWM outputs go high at the beginning of a PWM cycle.

ii) Each PWM output will go low when its match value (in MR1 to MR6) is reached. If no match occurs the PWM output remains continuously high.

iii) When a match occurs, actions can be triggered automatically. The possible actions are to generate an interrupt, reset the PWM timer counter, or stop the timer.

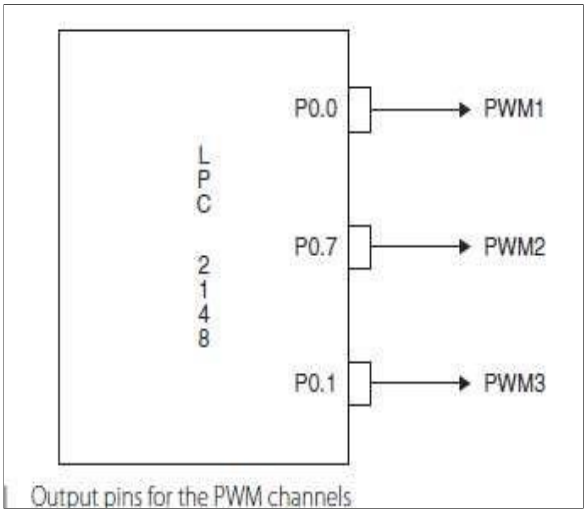
- The duty cycle is the ratio of ON period (P) to the total period T.
- Corresponding to the six match registers, there are six PWM output pins, and they
- are called the PWM channels



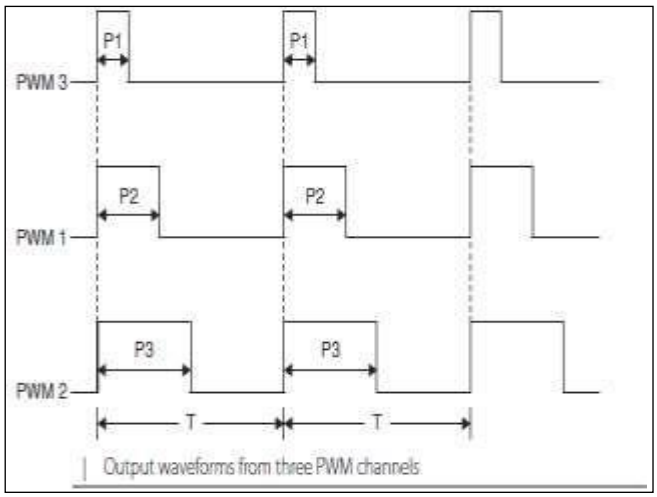
Control Registers of the PWM Unit

- PWMTCR(PWM Timer Control Register)
- It is is an 8-bit register. Only the lower 4 bits of this register need to be used.
- Bit 0–CE→ COUNTER ENABLE When '1', the PWM timer counter and Prescale counter are enabled.
- Bit 1–CR→ COUNTER RESET 'When '1', the above mentioned PWM timer count register and Prescale counter are reset on the next positive going edge of PCLK.
- Bit 2→R-Reserved
- Bit 3→ PE-PWM ENABLE. When '1', the PWM mode is enabled. Otherwise the
- PWM unit acts as just a timer.

- P



Output pins for the PWM channels



Output waveforms from three PWM channels

Control Register)

- It is a 16-bit register and is used to enable and select the type of each PWM channel.
- This register enables or disables the six PWM outputs, and also chooses between double and single edge control.
- Bits 0, 1, 7 and 8 and 15 are unused

PWMLER (PWM Latch Enable Register)

- The PWM latch enable register is an 8-bit register used to control the update of the PWM match registers when they are used for PWM generation.

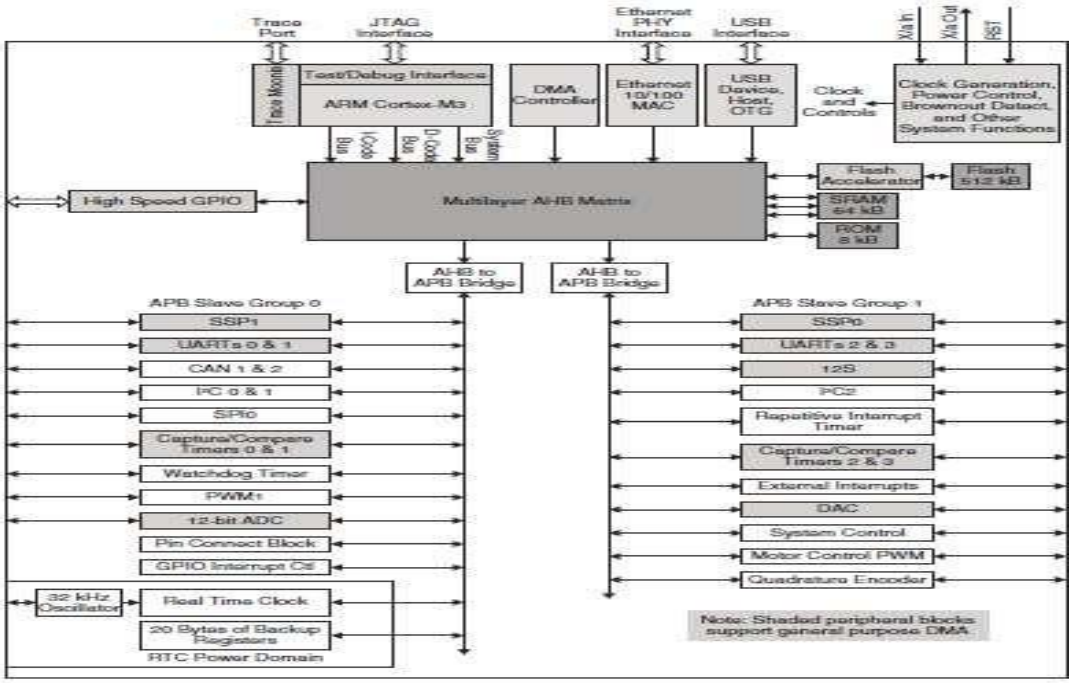
5. Design ARM Cortex M3 MCU based microcontroller for embedded applications.

- The LPC 17xx is an ARM Cortex-M3 based microcontroller for embedded applications requiring a high level of integration and low-power dissipation.
- It is a next generation core that offers debug features and a higher level of support block integration**.**

Feautres

- High speed versions (LPC 1769 and LPC 1759) operate at up to a 120 MHz
- CPU frequency.
- It incorporates a 3-stage pipeline.
- It uses a Harvard architecture with separate local instruction and data buses as well as a third bus for peripherals.

- It also includes an internal prefetch unit that supports speculative branches.
- The peripheral complement of the LPC 17xx includes up to 512 kB of flash memory.
- It uses up to 64 kB of data memory.



- Consists of Ethernet MAC, a USB interface that can be configured as either Host, Device.
- Support 8 channel general purpose DMA controller.
- Consists of 4 UARTs, 2 CAN channels, 2 SSP controllers, SPI interface, 3 I2C interfaces, 2-input plus 2- output I2S interface.
- It has 8 channel 12-bit ADC, 10-bit DAC, motor control PWM, Quadrature Encoder interface.
- It consists of 4 general purpose timers, 6-output general purpose PWM.
- It uses ultra-low power RTC with separate battery supply.
- support up to 70 general purpose I/O pins.

UNIT III EMBEDDED PROGRAMMING

1. **Compare the operation between assembler and compiler**
    - Compilers → used to create the instruction-level program in to assembly language code.
    - Assembler's→ used to translate symbolic assembly language statements into bit-level representations of instructions known as object code and also translating labels into addresses

2. **Recall the techniques involved in program-level energy and power analysis and optimization**
    - To replace the algorithms with others that consume less power.
    - By optimizing memory accesses ,able to significantly reduce power.
    - To turn off the subsystems of CPU, chips in the system, in order to save power

3. **Summarize the purpose of instruction level simulator in program's performance measurement.**
    - An instruction set simulator (ISS) is a simulation model, usually coded in a high-level programming language, which mimics the behavior of a mainframe or microprocessor by reading instructions and maintaining internal variables which represent the processor's registers.
4. **Show the steps to improve energy consumption in power level optimization method.**
    - To replace the algorithms with others that consume less power.
    - By optimizing memory accesses, able to significantly reduce power.
    - To turn off the subsystems of CPU, chips in the system, in order to save power
5. **Extend the disadvantages of cache memory**
    - Cache memory comes at an increased marginal cost than main memory and thus can increase the cost of the overall system.
    - Cached data is stored only so long as power is provided to the cache
6. **Distinguish between data flow graph with control 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 have only one entry and exit point—is known as a basic block.
    - CDFG uses a data flow graph as an element, adding constructs to describe control

7. **Recall about the logic analyser**
    - It is used to measure the start and stop times of a code segment.
    - The length of code that can be measured is limited by the size of the logic analyzer's buffer.

8. **Outline the significance of CDFG**
    - A CDFG uses a data flow graph as an element,adding constructs to describe control
    - Decision nodes→used to describe the control in a sequential program
    - Data flow nodes→ encapsulates a complete data flow graph to represent a data.

9. **Distinguish between block box testing and clear box testing**
    - **White/Clear-box Testing→** generate tests ,based on the program structure.
    - **Black-box Testing→** generate tests ,without looking at the internal structure of the program

10. **Mention the importance of state machine**
    - The reaction of most systems can be characterized in terms of the input received and the current state of the system.
    - The finite-state machine style of describing the reactive system's behavior..
    - Finite-state machines are usually first encountered in the context of hardware design.

PART-B

1. **Discuss in detail about the fundamental model used for program development.**

- Programs are collection of instructions to execute a specified task.
- Models for programs are more general than source code.
- source code can't be used directly because of different type s such as assembly language,C code.
- Single model to describe all of them.
- control/data flow graph (CDFG)→it is the fundamental model for programs

```
w = a + b;
x1 = a - c;
y = x1 + d;
x2 = a + c;
z = y + e;
```

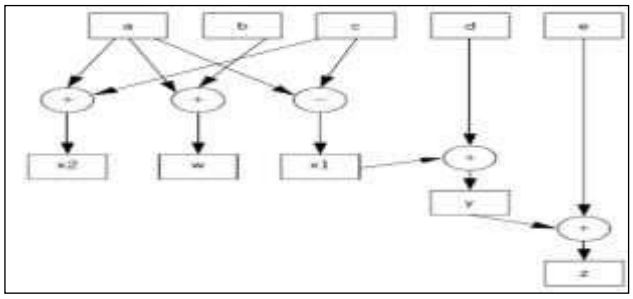
1)DATA FLOW GRAPH

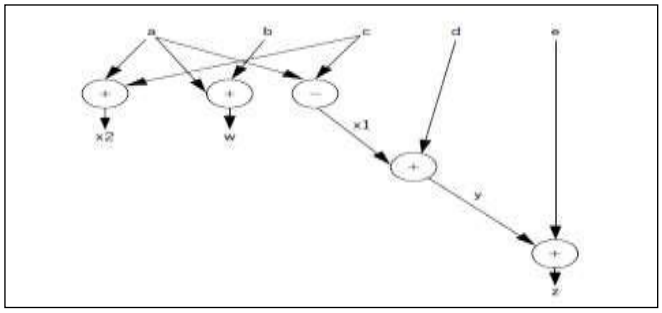
- A data flow graph is a model of a program with no conditionals.

- In a high-level programming language, a code segment with no conditionals have only one entry and exit point—is known as a basic block.

An extended data flow graph for our sample basic block

- The basic block in single-assignment form
- Round nodes→denote operators
- Square nodes→denote values.
- The value nodes may be either inputs(a,b) or variables(w,x1).

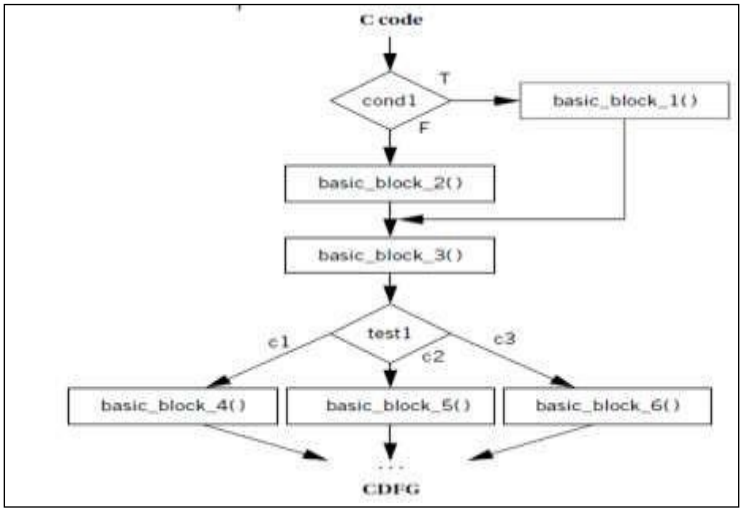




2) Control/Data Flow Graphs(CDFG)

CDFG having following two types of nodes.

- Decision nodes→used to describe the control in a sequential program
- Data flow nodes→ encapsulates a complete data flow graph to represent a data.



C code and its CDFG

```
if (cond1)
basic_block_1( );
else
basic_block_2();
basic_block_3( );
switch (test1) {
```

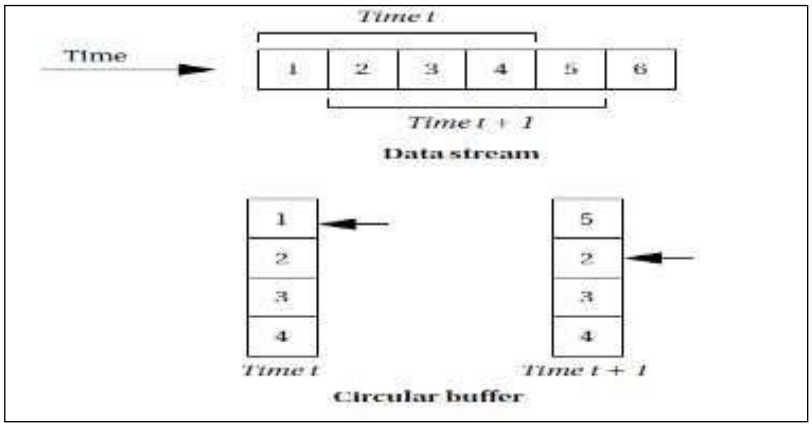
```
case c1: basic_block_4( ); break;
case c2: basic_block_5( ); break;
case c3: basic_block_6( ): break;
```

2. Examine the components involved in embedded program design with an suitable example

- Embedded components are given by State machine, Circular buffer, and the Queue.

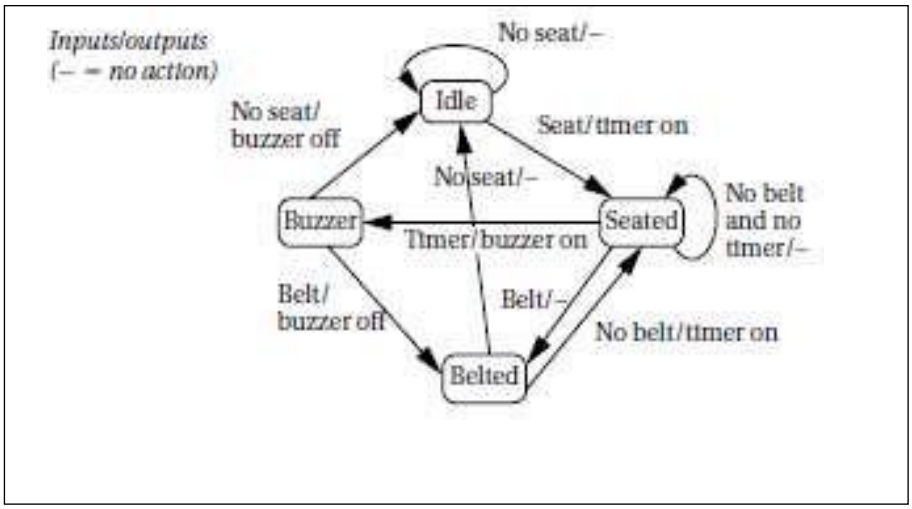
STATE MACHINE

- The reaction of most systems can be characterized in terms of the input received and the current state of the system.
- The finite-state machine style of describing the reactive system's behavior..
- Finite-state machines are usually first encountered in the context of hardware design.
- The circular buffer is a data structure that handle streaming data in an efficient way.
- Size of the window does not change.
- Fixed-size buffer to hold the current data.
- To avoid constantly copying data within the buffer, move the head of the buffer in time.
- The buffer points to the location at which the next sample will be placed.



- Every time add a sample, automatically overwrite the oldest sample, which is the one that needs to be thrown out.
- When the pointer gets to the end of the buffer, it wraps around to the top.
- Queues are also used in signal processing and event processing.

- Queues are used whenever data may arrive and depart at somewhat unpredictable times or when variable amounts of data may arrive.
- A queue is often referred to as an Elastic buffer.

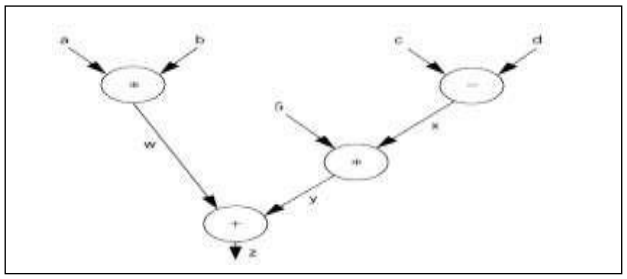


3. Elucidate the process involved in basic compilation technique with an example.

- Compilation=Translation+optimization
- Compilation begins with high-level language code (C) and produces assembly code.
- The high-level language program is parsed to break it into statements and expressions.
- symbol table is generated, which includes all the named objects in the program.
- Instruction-level optimizations used to generating code. (real instructions or on a pseudo-instruction)
- This level of optimization used to create simple code .

Statement Translation

- Translating the high-level language program with little or no optimization.
- Eg→Compiling an arithmetic expression
- a*b + 5*(c – d)
- In the above example the variable is written in terms of program variables.
- In ARM, first load the variables into registers.



- This requires choosing which registers receive not only the named variables but also intermediate results such as (c d).

Procedures

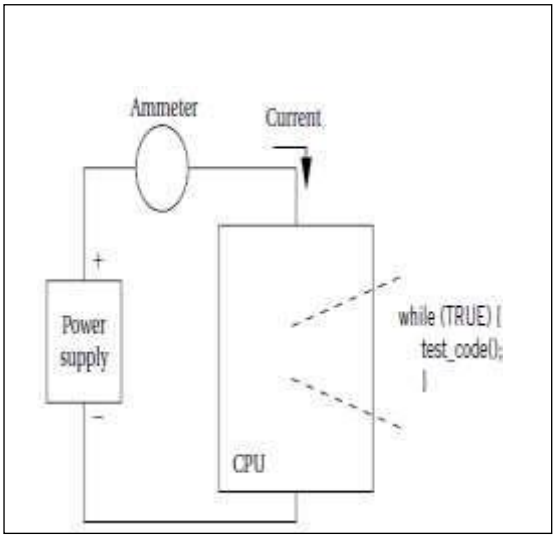
- Another major code generation problem is the creation of procedures.
- Procedure stacks are typically built to grow down from high addresses.
- stack pointer (sp)→ defines the end of the current frame.
- frame pointer (fp)→ defines the end of the last frame.
- The procedure can refer to an element in the frame by addressing relative to sp.
- When a new procedure is called, the sp and fp are modified to push another frame onto the stack.
- r1-r3 → are used to pass parameters into the procedure.
- r0→ used to hold the return value.
- r4- r7→ hold register variables.
- r11→ is the frame pointer
- r13→ is the stack pointer.
- r10 → to check for stack overflows.

4. Discuss in detail the program level embedded system performance analysis

- Power consumption is a important design metric for battery-powered systems.
- It is increasingly important in systems that run off the power grid.
- Fast chips run hot, and controlling power consumption is an important element of increasing reliability and reducing system cost.

- **Power consumption reduction techniques.**
  - To replace the algorithms with others that consume less power.
  - By optimizing memory accesses ,able to significantly reduce power.
  - To turn off the subsystems of CPU, chips in the system, in order to save power.

Measuring energy consumption for a piece of code



- Program's energy consumption→ how much energy the program consumes.
- To measure power consumption for an instruction or a small code fragment.
- It is used to executes the code under test over and over in a loop.
- By measuring the current flowing into the CPU,we are measuring the power consumption of the complete loop, including both the body and other code.
- By separately measuring the power consumption of a loop with no body.
- we can calculate the power consumption
- of the loop body code as the difference b/w the full loop and the bare loop energy cost of an instruction.
- List of the factors contribution for energy consumption of the program.
- Energy consumption varies somewhat from instruction to instruction.
- The sequence of instructions has some influence.
- The opcode and the locations of the operands also matter.
- Steps to Improve Energy Consumption
- Try to use registers efficiently(r4)
- Analyze cache behavior to find major cache conflicts.
- Make use of page mode accesses in the memory system whenever possible.
- Moderate loop unrolling eliminates some loop control overhead. when the loop is unrolled too much, power increases.
- Software pipelining reducing the average energy per instruction.
- Eliminating recursive procedure calls where possible saves power by getting rid of function call overhead.
- Tail recursion can often be eliminated, some compilers do this automatically.

5. Design and develop suitable testing process involved in developing in embedded system.

- Testing→requires the control/data flow graph of a program's source code.
- To test the program→ exercise both its control and data operations.
- To execute and evaluate the tests→ control the variables in the program and observe the results .

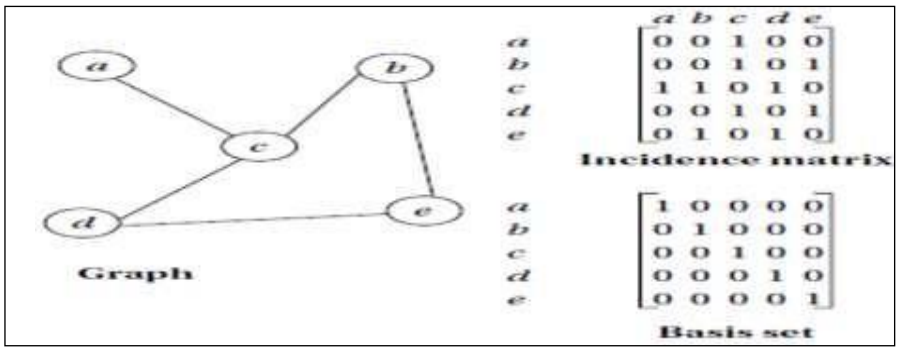
The following three things to be followed during a test

1. Provide the program with inputs for the test.
2. Execute the program to perform the test.
3. Examine the outputs to determine whether the test was successful.

- Execution Path→To test the program by forcing the program to execute along chosen paths. (giving it inputs that it to take the appropriate branches)

Graph Theory

- It help us get a quantitative handle on the different paths required.
- Undirected graph-→form any path through the graph from combinations of basis paths.
- Incidence matrix contains each row and column represents a node.
- 1 is entered for each node pair connected by an edge.



Cyclomatic Complexity

- It is a software metric tool.
- Used to measure the control complexity of a program.
  M = e – n + 2p.
- e→ number of edges in the flow graph
- n → number of nodes in the flow graph
- p → number of components in the graph

1)Branch testing

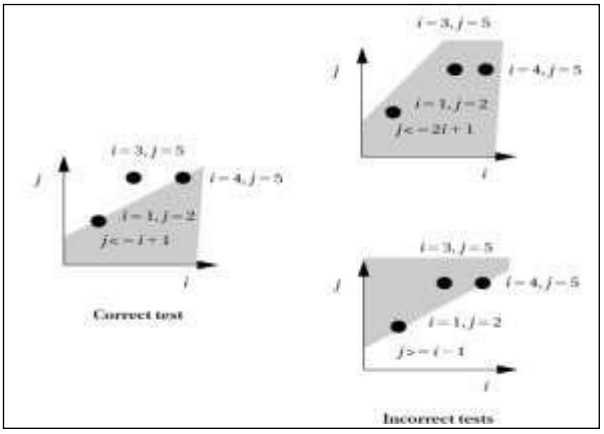
- This strategy requires the true and false branches of a conditional.

- Every simple condition in the conditional's expression to be tested at least once.
  if ((x == good_pointer) && (x->field1 == 3))
  { printf("got the value\n"); }

The bad code we actually wrote

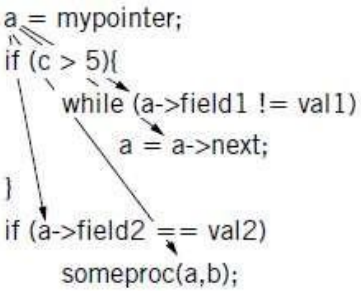
```
if ((x = good_pointer) && (x->field1 == 3))
{ printf("got the value\n"); }
```

2)Domain testing



- It concentrates on linear-inequalities.
- The program should use for the test is j <= i + 1
- We test the inequality with three test points
- Two on the boundary of the valid region
- Third outside the region but between the i values of the other two points.

Data flow testing



- It use of def-use analysis (definition-use analysis).
- It selects paths that have some relationship to the program's function.
- Compilers→ which use def-use analysis for Optimization.
- A variable's value is defined when an assignment is made to the variable.
- It is used when it appears on the right side of an assignment.

2) Block Box Testing

- Black-box tests are generated without knowledge of the code being tested.
- It have a low probability of finding all the bugs in a program.
- We can't test every possible input combination, but some rules help us select reasonable sets of inputs.

Random Tests

- Random values are generated with a given inputs.
- The expected values are computed first, and then the test inputs are applied.

Regression Tests

- When tests are created during earlier or previous versions of the system.
- Those tests should be saved→ apply to the later versions of the system.
- It simply exercise current version of the code and possibly exercise different bugs.
- In digital signal processing systems→ Signal processing algorithms are implemented to save hardware costs.
- Data sets can be generated for the numerical accuracy of the system.
- These tests can often be generated from the original formulas without reference to the source code.

UNIT IV REAL TIME SYSTEMS

1. **List the challenges involved in designing of a real time system**
   - **Predictability-** Able to predict the future consequences of current actions
   - **Testability-**Easy to test if the system can meet all the deadlines n
   - **Maintainability-**Modular structure to ease system modification
   - **Fault tolerance-**Hardware and software failures should not cause the system

2. **Outline the functions of real time operating systems**
   - It is a time-bound system which has well-defined, fixed time constraints.
   - It is required to complete the work on a time constraint
3. **Define fault and mention its types**
   - Fault→It is a defect or flow that occurs in some hardware or software component.

   Types of Faults
   - Hardware Faults
   - Software Faults

4. **Infer about working of utilization balancing algorithm**
    - This algorithm is suitable when the number of processors in a multiprocessor is fixed.
    - The utilization balancing algorithm can be used when the tasks at the individual processors are scheduled using EDF.
5. **Interpret about the domino effect in timing redundancy**
    - Each process is independent, not working in operation with any other process. If this is not the case and the processes interact dings become more complex.
6. **Show the characteristics of real time systems**
    - **Large and complex**-Vary from a few hundred lines of assembler or C
    - **Concurrent control of separate system components**- Devices operate in parallel in the real-world
    - **Facilities to interact with special purpose hardware** - Need to be able to program devices in a reliable and abstract way
    - **Mixture of Hardware/Software**-Some modules implemented in hardware, even whole systems
7. **Define task and classify its types.**
    - Each task has resource requirements
    - **Types**
    - Periodic
    - Sporadic
    - Aperiodic
8. **Recall about the offline scheduling**

    Precomputed (offline scheduling)
    - Involves scheduling in advance of the operation, with specifications of when the periodic tasks will be run and slots for the sporadic/aperiodic tasks in the event that they are involved.
9. **Outline the impact of online scheduling**

    Dynamically (online Scheduling)
    - The tasks are scheduled as they arrive in the system.

- The algorithms used in online scheduling must be fast and it takes to meet their deadlines is clearly useless

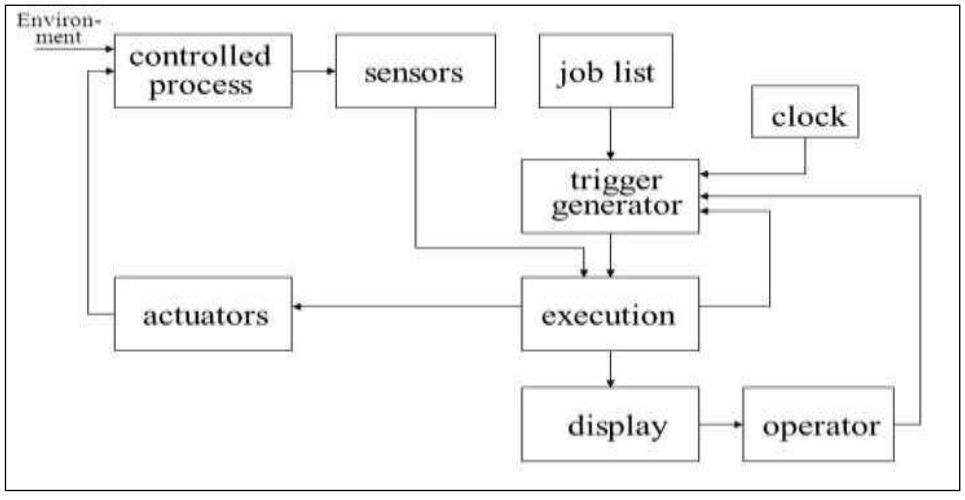
10. Comare the operation of Release Time and dead line

- It is the time at which all the data that are required to begin executing the task are available.
- **Deadline**
- It is the time by which the task must complete its execution.
- It may be hard or soft, depending on the nature of the corresponding task.

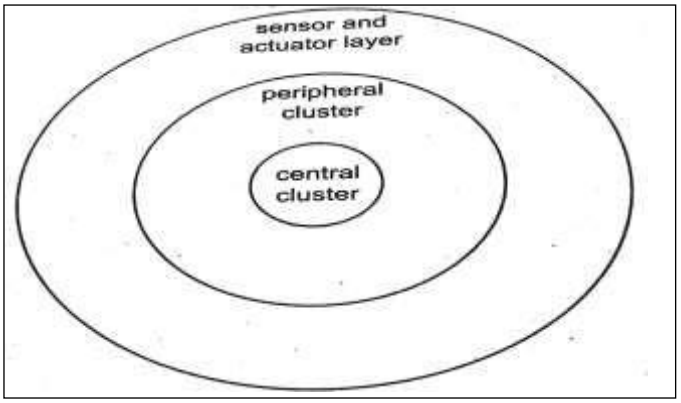
PART-B

1. Construct the structure of a real time systems along with its characteristics

- Trigger generator used to trigger the execution of individual jobs. It is not really a separate hardware unit, typically it is part of the executive software.
- The schedule for these jobs can be obtained offline and loaded as a lookup table to be used by the scheduler.
- Jobs can also be initiated depending on the state of the controlled process or on the operating environment.
- The output of the computer is fed to the actuators and the displays.
- Fault tolerant techniques ensure the erroneous outputs from the computer.
- The actuators typically have a mechanical or a hydraulic component, and so their time constants are quite high.



- The sensors and actuators run at relatively low data rates.
- The computer itself must be fast enough to execute the control algorithms, and these can require throughputs in excess of 50 million instructions per second(MIPS).
- **System separates into three areas**
- An outer low rate area consisting of the sensors, actuators, displays and input panels.
- A middle or peripheral area consisting of the processing that is necessary to format the data from and to this layer properly.
- The central cluster of processors where the control algorithms are executed.



2. Discuss in details about the various techniques involved in estimating program run times.

- Real-time systems should meet deadlines; it is important to be able to accurately estimate program run times.
- Estimating the execution time of any given program is a very difficult task and it depends on the following factors

a) Source code

- Source code that is carefully tuned and optimized takes less time to execute.

b) Compiler

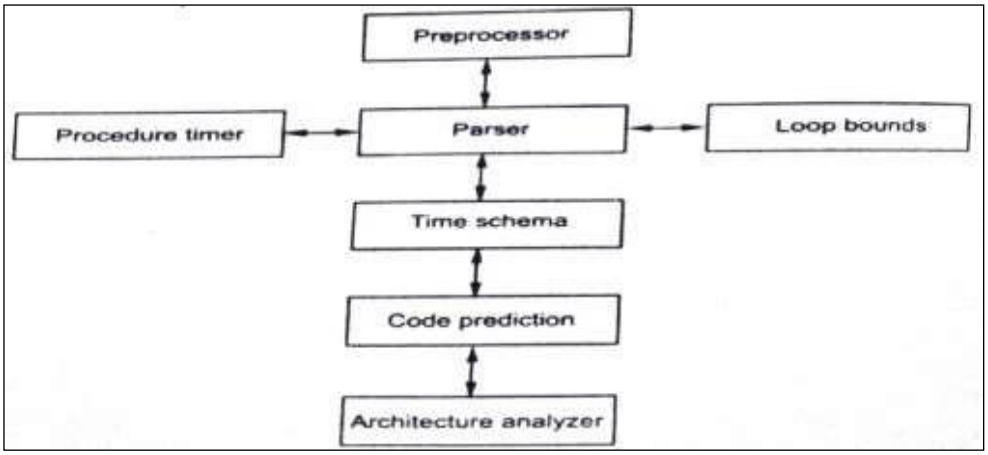
- The compiler maps the source-level code into a machine-level program.
- The actual mapping will depend on the actual implementation of the particular compiler that is being used.

c) Operating system

- The operating system determines such issues as task scheduling and memory management, and also it determines the interrupt handling overhead

Machine architecture

- Executing a program may require much interaction between the processors and the memory and I/O devices.
- The interaction can take place over an interconnection network (e.g., a bus) that can be shared by other processors.
- The number of registers per processor affects how many variables can be held in the CPU.
- The greater the number of registers and the cleverer the compiler is in managing these registers.
- This results in reducing the memory-access time, and hence the instruction-execution time.
- The size and organization of the cache (if any) will also affect the memory-access time, as will the clock rate.
- To keep the contents of these memories, we need to periodically refresh them
- This is done by periodically reading the contents of each memory location and writing them



Preprocessor

- The pre-processor produces compiled assembly language code and marks off blocks of code to be analyzed.

Parser

- The parser analyzes the input source program.

Procedure timer

- It maintains a table of procedures and their execution time

Loop bounds

- It obtains number of iterations for the various loops in the system.

Time schema

- It computes the execution times of each block using the execution time estimates computed by the code prediction module.

Code prediction

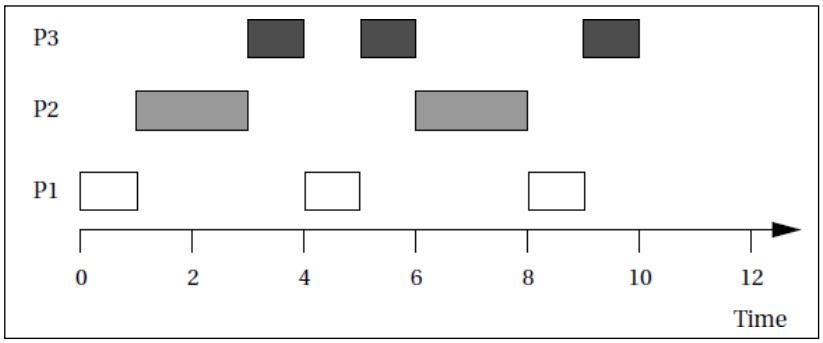
- The code prediction module does this by using the code generated by the pre-processor and using the architecture analyzer to include the influence of the architecture

3. With an suitable example explain about the RMA algorithm and calculate the various task execution time

RMS is known as rate-monotonic analysis (RMA), as summarized below.

- All processes run periodically on a single CPU.
- Context switching time is ignored.
- There are no data dependencies between processes.
- The execution time for a process is constant.
- All deadlines are at the ends of their periods.
- The highest-priority ready process is always selected for execution.
- Priorities are assigned by rank order of period, with the process with the shortest period being assigned the highest priority.
- According to RMA → Assign highest priority for least execution period.

Process	Execution time	Period
P1	1	4
P2	2	6
P3	3	12



- Hence P1 the highest priority, P2 the middle priority,and P3 the lowest priority.
- First execute P1 then P2 and finally P3.(T1>T2>T3)
- After assigning priorities, construct a time line equal in length to hyper period, which is 12 in this case.
- Every 4 time intervals P1 executes 1 units.(Execution time intervals for P1 0-4,4-8,8-12)
- Every 6 time intervals P2 executes 2 units. .(Execution time intervals for P2 0-6,6-12)
- Every 12 intervals P3 executes 3 units. .(Execution time intervals for P3 0-12)
- Time interval from 10-12 no scheduling available because no process will be available for execution. All process are executed already.
- P1 is the highest-priority process, it can start to execute immediately.
- After one time unit, P1 finishes and goes out of the ready state until the start of its next period.
- At time 1, P2 starts executing as the highest-priority ready process.
- At time 3, P2 finishes and P3 starts executing.
- P1's next iteration starts at time 4, at which point it interrupts P3.
- P3 gets one more time unit of execution between the second iterations of P1 and P2, but P3 does not get to finish until after the third iteration of P1.

4. Develop Earliest Deadline First algorithm and calculate the various task execution time.

- Earliest deadline first (EDF)→ is 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 also very simple.
- It assigns priorities in order of deadline.
- Assign highest priority to a process who has Earliest deadline.
- Assign lowest priority to a process who has farthest deadline.
- After assigning scheduling procedure, the highest-priority process is chosen for execution.

- **<u>Consider the following Example</u>**
  - Hyper-period is 60
  - There is one time slot left at t= 30, giving a CPU utilization of 59/60.
  - EDF can achieve 100% utilization

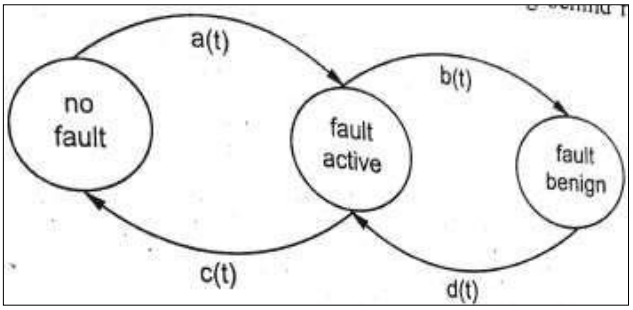
Process	Execution time	Period
P1	1	3
P2	1	4
P3	2	5

5. Elaborate about the fault tolerance techniques in real time systems along with its types.

- It is the ability of a system to maintain its functionality,even in the presence of faults.
- Fault→It is a defect or flow that occurs in some hardware or software component.
- Error→It is a manifestation of a fault.
- Failure→It is a departure of a system from the service required.

- **Types of Faults**
- **Hardware fault**
  - It is some physical defect that can cause a component to malfunction.
  - A broken wire or the output of a logic gate that is perpetually stuck some logic value (0 or 1) are hardware faults.
- **Software Faults**
  - A software fault is a "bug" that can cause the program to fail for a given set of inputs.
  - Failures Causes
  - (i) Errors in the specifications or design
  - (ii) Defects in the components and
  - (iii) Environmental effects
- **Fault Types**
  - Faults are classified according to their temporal behavior and output behavior.
  - Temporal Behaviour Classification
  - 1.Permanent- does not die away with time but remains until it is repaired or the affected unit is replaced
  - 2. Intermittent- faults cycles between the fault active and fault benign states.

- 3.TransientPermanent- It is hard to catch. Since quite often by the time the system has recognized that such a failure has occurred it has disappeared, leaving behind no permanent defect that can .be located



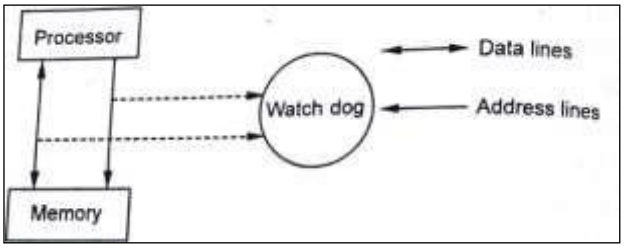
1)Fault detection

(i) Online detection

- Goes on in parallel with normal system operation.
- Fetching an opcode from a location containing data.
- Writing into a portion of memory to which the process has no write access.
- Fetching an illegal opcode.
- Inactive for more than a prescribed period.
- A monitor (watch dog processor) is associated with each processor, looking for signs that the processor is faulty.

(ii) Offline detection

- When a processor is running such a test, it obviously cannot be executing the applications software.Diagnostic tests can be scheduled



UNIT V PROCESSES AND OPERATING SYSTEMS

1. **MPEG layer-1 data frame format**

Header	CRC	Bit allocation	Scale factors	Subband samples	Aux data

2. **List out the Functions of video accelerator**

- It is a hardware circuits on a display adapter that speed up fill motion video.
- Primary video accelerator functions are color space conversion, which converts YUV to RGB**.**

3. **What is meant by preemptive RTOS?**

- A pre emptive OS →solves the fundamental problem in multitasking system.
- It executes processes based upon timing requirements provided by the system designer

4. **Define context switching mechanism**

- To understand the basics of a context switch, let's assume that the set of tasks is in steady state.Everything has been initialized, the OS is running, and we are ready for a timer interrupt**.**

5. **What do you understand the concept of priority inversion?**

- It is a process of low-priority process blocks execution of a higher priority process by keeping hold of its resource.

6. **Summarize the importance of tasks and processes in embedded system**

- Task is nothing but different parts of functionality in a single system.
- **Eg-Mobile Phones**
- A process is a single execution of a program.
- If we run the same program two different times, we have created two different processes

7. **Outline the main functions performed by engine control unit**
    - This unit controls the operation of a fuel-injected engine based on several measurements taken from the running engine.
8. **Define task and process**
    - Task is nothing but different parts of functionality in a single system.Eg-Mobile Phones
    - A process is a single execution of a program. If we run the same program two different times, we have created two different processes
9. **Infer the features of priority based scheduling algorithm**
    - It is one of the first scheduling policies developed for real-time systems.
    - It assigns priorities are sufficient to efficiently schedule the processes in many situations.
    - RMS and EDF are a example of scheduling policy.

10. **What does the concept of multitasking? What does it signify?**
    - Multi-tasking→ The ability of an operating system to hold multiple processes in memory and switch the processor for executing one process
    - Multiprocessor system can execute multiple processes simultaneously with the help of multiple CPU.

PART-B

1. **With an suitable example explain about the dynamic priority algorithm and calculate the various task execution time**

- Earliest deadline first (EDF)→ is 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 also very simple.
- It assigns priorities in order of deadline.
- Assign highest priority to a process who has Earliest deadline.
- Assign lowest priority to a process who has farthest deadline.
- After assigning scheduling procedure, the highest-priority process is chosen for execution.
- **Consider the following Example**

Process	Execution time	Period
P1	1	3
P2	1	4
P3	2	5

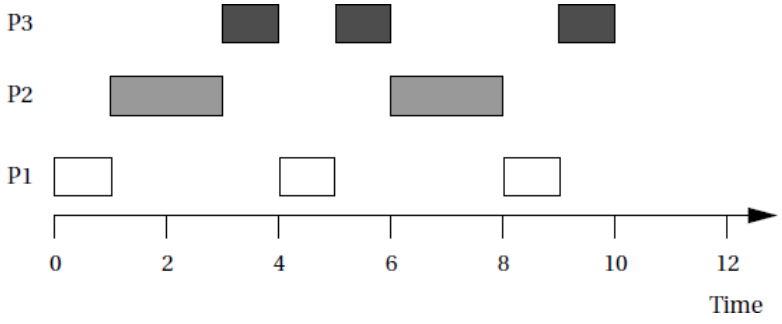
- Hyper-period is 60
- There is one time slot left at *t= 30, giving a CPU utilization of 59/60.*
- EDF can achieve 100% utilization

RMS is known as rate-monotonic analysis (RMA), as summarized below.

- All processes run periodically on a single CPU.
- Context switching time is ignored.
- There are no data dependencies between processes.
- The execution time for a process is constant.
- All deadlines are at the ends of their periods.
- The highest-priority ready process is always selected for execution.

Process	Execution time	Period
P1	1	4
P2	2	6
P3	3	12

- Priorities are assigned by rank order of period, with the process with the shortest period being assigned the highest priority.
- According to RMA →Assign highest priority for least execution period.



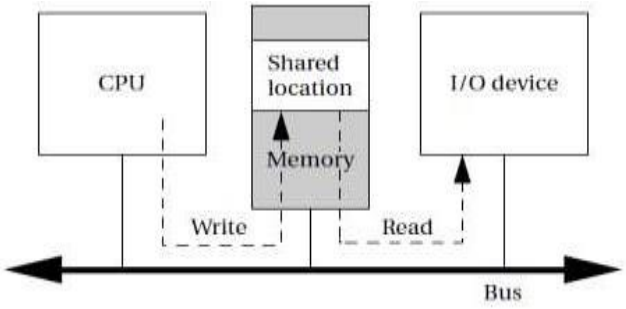
- Hence P1 the highest priority, P2 the middle priority,and P3 the lowest priority.
- First execute P1 then P2 and finally P3.(T1>T2>T3)
- After assigning priorities, construct a time line equal in length to hyper period, which is 12 in this case.
- Every 4 time intervals P1 executes 1 units.(Execution time intervals for P1 0-4,4-8,8-12)
- Every 6 time intervals P2 executes 2 units. .(Execution time intervals for P2 0-6,6-12)
- Every 12 intervals P3 executes 3 units. .(Execution time intervals for P3 0-12)

- Time interval from 10-12 no scheduling available because no process will be available for execution. All process are executed already.
- P1 is the highest-priority process, it can start to execute immediately.
- After one time unit, P1 finishes and goes out of the ready state until the start of its next period.
- At time 1, P2 starts executing as the highest-priority ready process.
- At time 3, P2 finishes and P3 starts executing.
- P1's next iteration starts at time 4, at which point it interrupts P3.
- P3 gets one more time unit of execution between the second iterations of P1 and P2, but P3 does not get to finish until after the third iteration of P1.

2. Compare the principle, merits and limitations of inter-process communication mechanism.

Interprocess communication

- It is provided by the operating system as part of the process abstraction.
- Blocking Communication→ The process goes into the waiting state until it receives a response
- Non-blocking Communication→It allows a process to continue execution after sending the communication.



<u>Types of inter-process communication</u>

1. **Shared Memory Communication**
2. **Message Passing**
3. **Signals**

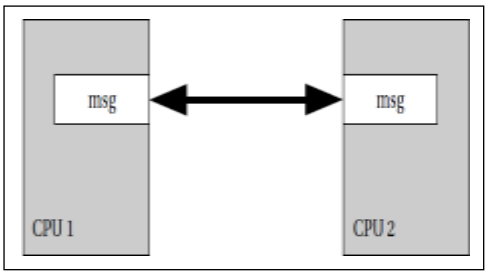
<u>1) Shared Memory Communication</u>

- The communication between inter-process is used by bus-based system.
- CPU and an I/O device, communicate through a shared memory location.
- The software on the CPU has been designed to know the address of the shared location.
- The shared location has also been loaded into the proper register of the I/O device.
- If CPU wants to send data to the device, it writes to the shared location.
- The I/O device then reads the data from that location.
- The read and write operations are standard and can be encapsulated in a procedural interface.

- CPU and the I/O device want to communicate through a shared memory block.
- There must be a flag that tells the CPU when the data from the I/O device is ready.
- The flag value of 0 when the data are not ready and 1 when the data are ready.
- If the flag is used only by the CPU, then the flag can be implemented using a standard memory write operation.
- If the same flag is used for bidirectional signaling between the CPU and the I/O device, care must be taken.

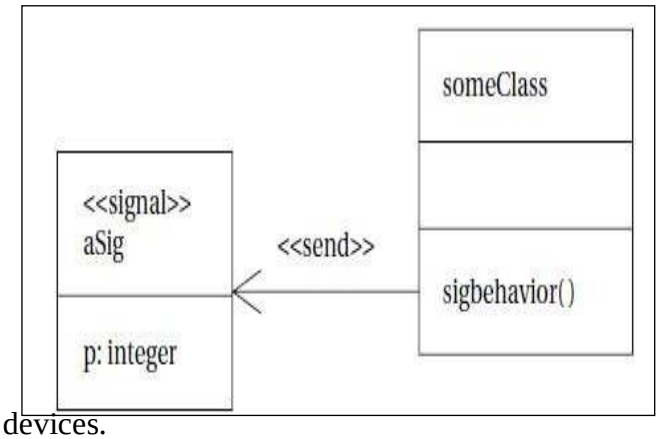
Consider the following scenario to call flag

1. CPU reads the flag location and sees that it is 0.
2. I/O device reads the flag location and sees that it is 0.
3. CPU sets the flag location to 1 and writes data to the shared location.
4. I/O device erroneously sets the flag to 1 and overwrites the data left by the CPU.



2) Message Passing

- Here each communicating entity has its own message send/receive unit.
- The message is not stored on the communications link, but rather at the senders/ receivers at the end points.
- Ex:Home control system
- It has one microcontroller per household device—lamp, thermostat, faucet, appliance.
- The devices must communicate relatively infrequently.
- Their physical separation is large enough that we would not naturally think of them as sharing a central pool of memory.
- Passing communication packets among the devices is a natural way to describe coordination between these devices.



3) Signals

- Generally signal communication used in Unix .

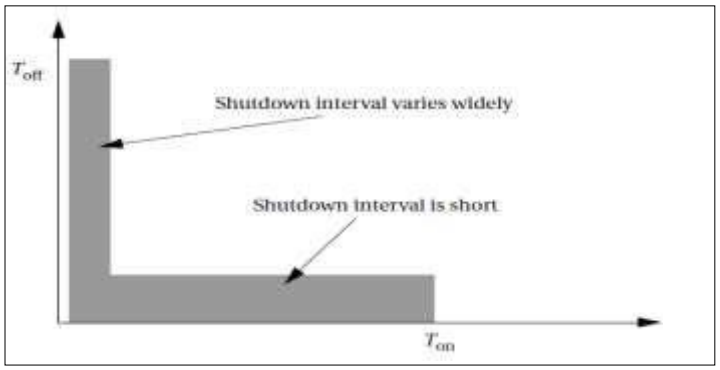
- A signal is analogous to an interrupt, but it is entirely a software creation.
- A signal is generated by a process and transmitted to another process by the OS.
- A UML signal is actually a generalization of the Unix signal.
- Unix signal carries no parameters other than a condition code.
- UML signal is an object, carry parameters as object attributes.
- The sigbehavior( ) →behavior of the class is responsible for throwing the signal, as indicated by<<send>>.
- The signal object is indicated by the <<signal>>

3. Adapt the suitable power optimization strategies mechanisms in embedded system

- A power management policy is a strategy for determining when to perform certain power management operations.
- The system can be designed based on the static and dynamic power management mechanisms.

Power saving strategies

- Avoiding a power-down mode can cost unnecessary power.
- Powering down too soon can cause severe performance penalties.
- Re-entering run mode typically costs a considerable amount of time.
- A straightforward method is to power up the system when a request is received



Predictive shutdown

- The goal is to predict when the next request will be made and to start the system just before that time, saving the requestor the start-up time.
- Make guesses about activity patterns based on a probabilistic model of expected behavior.

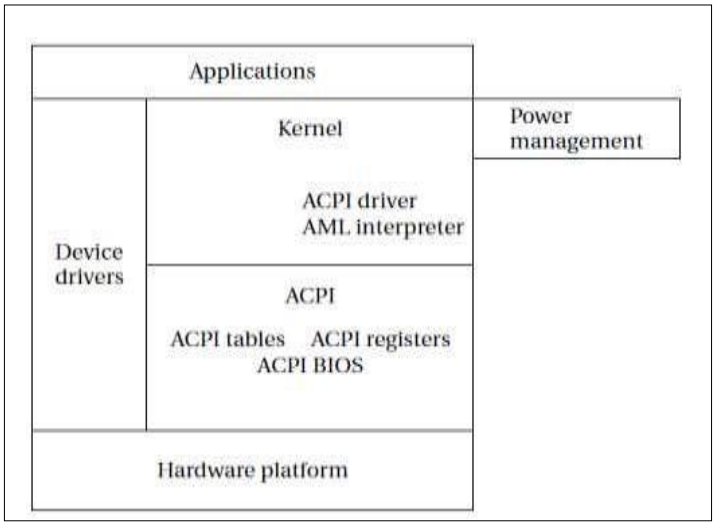
An L-shaped usage distribution

- A very simple technique is to use fixed times.
- If the system does not receive inputs during an interval of length Ton, it shuts down.
- Powered-down system waits for a period Toff before returning to the power-on mode.

- In this distribution, the idle period after a long active period is usually very short, and the length of the idle period after a short active period is uniformly distributed.
- Based on this distribution, shutdown when the active period length was below a threshold, putting the system in the vertical portion of the L distribution.

Advanced Configuration and Power Interface (ACPI)

- It is an open industry standard for power management services.
- It is designed to be compatible with a wide variety of OSs.
- A decision module →determines power management actions.



ACPI supports the following five basic global power states.

1. G3, the mechanical off state, in which the system consumes no power.
2. G2, the soft off state, which requires a full OS reboot to restore the machine to working condition. This state has four sub-states:

- S1, a low wake-up latency state with no loss of system context
- S2, a low wake-up latency state with a loss of CPU and system cache state
- S3, a low wake-up latency state in which all system state except for main
- memory is lost.

S4, the lowest-power sleeping state, in which all devices are turned off.

3. G1, the sleeping state, in which the system appears to be off.
4. G0, the working state, in which the system is fully usable.
5. The legacy state, in which the system does not comply with ACPI.

4. Explain in detail about the functional and non-functional requirement, operations and testing phases involved in audio player.

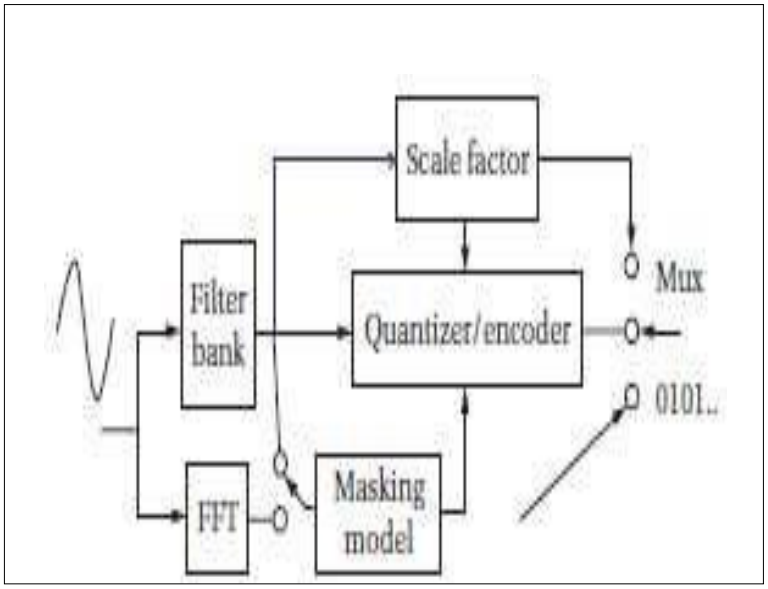
Operation and requirements

- MP3 players use either flash memory or disk drives to store music.
- It performs the following functions such as audio storage, audio decompression, and user interface.
- Audio compression→ It is a lossy process. The coder eliminates certain features of the audio stream so that the result can be encoded in fewer bits.
- Audio decompression→ The incoming bit stream has been encoded using a Huffman style code, which must be decoded.
- Masking→ One tone can be masked by another if the tones are sufficiently close in frequency.

Audio compression standards

- Layer 1 (MP1) →uses a lossless compression of sub bands and simple masking model.
- Layer 2 (MP2) →uses a more advanced masking model.
- Layer 3 (MP3)→ performs additional processing to provide lower bit rates.

MPEG Layer 1 encoder



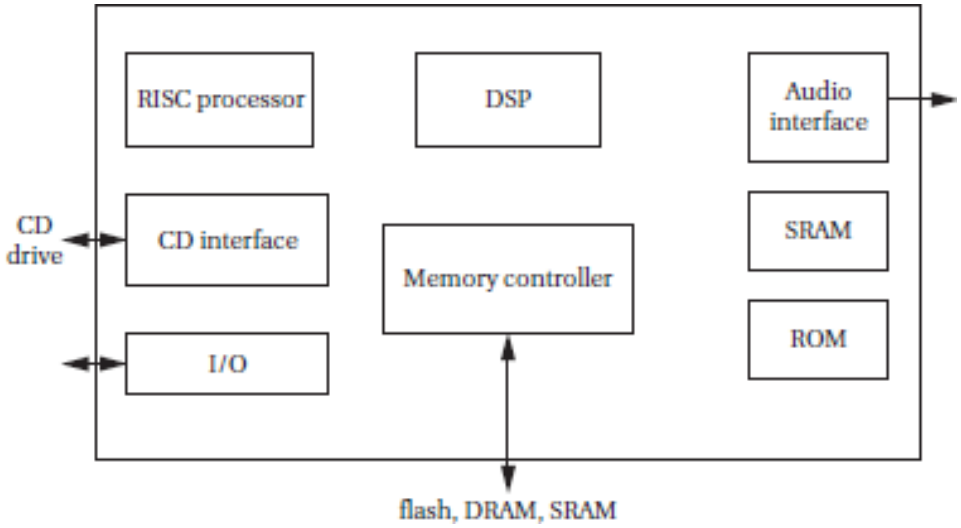
- Filter bank→ splits the signal into a set of 32 sub-bands that are equally spaced in the frequency domain and together cover the entire frequency range of the audio.
- Encoder→It reduce the bit rate for the audio signals.
- Quantizer →scales each sub-band(fits within 6 bits), then quantizes based upon the current scale factor for that sub-band.
- Masking model → It is driven by a separate Fast Fourier transform (FFT), the filter bank could be used for masking, a separate FFT provides better results.
- The masking model chooses the scale factors for the sub-bands, which can change along with the audio stream.
- Multiplexer→ output of the encoder passes along all the required data.

MPEG Layer 1 decoder

- After disassembling the data frame, the data are un-scaled and inverse quantized to produce sample streams for the sub-band.

- An inverse filter bank then reassembles the sub-bands into the uncompressed signal.
  User interface→ MP3 player is simple both the physical size and power consumption of the device. Many players provide only a simple display and a few buttons.
  File system→ player generally must be compatible with PCs. CD/MP3 players used compact discs that had been created on PCs.

System architecture



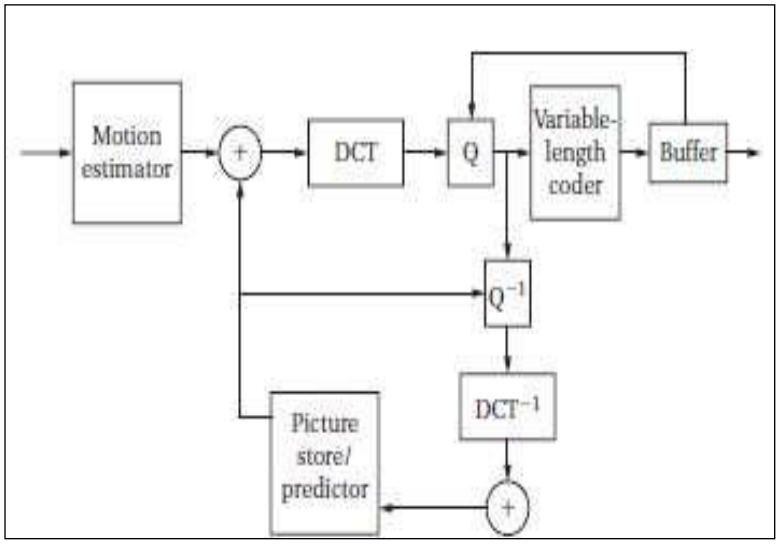
- The audio controller includes two processors.
- The 32-bit RISC processor is used to perform system control and audio decoding.
- The 16-bit DSP is used to perform audio effects such as equalization.
- The memory controller can be interfaced to several different types of memory.
- Flash memory can be used for data or code storage.
- DRAM can be used to handle temporary disruptions of the CD data stream.
- The audio interface unit puts out audio in formats that can be used by A/D converters.
- General- purpose I/O pins can be used to decode buttons, run displays.

Component design and testing

- The audio output system should be tested separately from the compression system.
- Testing of audio decompression requires sample audio files.
- The standard file system can either implement in a DOS FAT or a new file system.
- While a non-standard file system may be easier to implement on the device, it also requires software to create the file system.
- The file system and user interface can be tested independently .

5. Interpret in details about the embedded concepts in the design of video accelerator

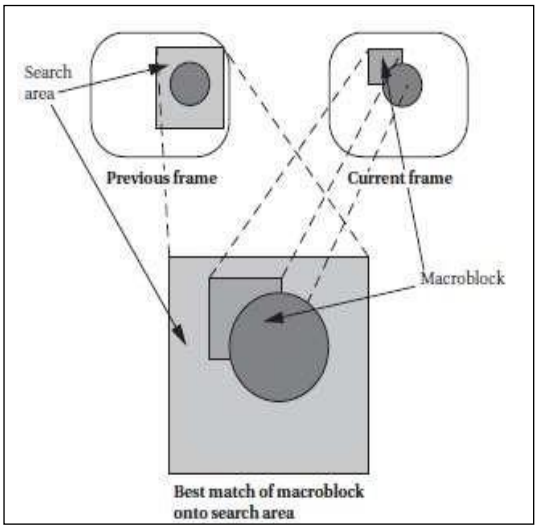
- It is a hardware circuits on a display adapter that speed up fill motion video.
- Primary video accelerator functions are color space conversion, which converts YUV to RGB.
- Hardware scaling is used to enlarge the image to full screen and double buffering which moves the frames into the frame buffer faster.
- **Video compression**



- MPEG-2 forms the basis for U.S. HDTV broadcasting.
- This compression uses several component algorithms together in a feedback loop.
- Discrete cosine transform (DCT) used in JPEG and MPEG-2.
- DCT used a block of pixels which is quantized for lossy compression.
- Variable-length coder→assign number of bits required to represent the block.

Block motion Estimation

- MPEG uses motion to encode one frame in terms of another.
- some frames are sent as modified forms of other frames using a technique known as block motion estimation.
- During encoding, the frame is divided into macro blocks.
- Encoder uses the encoding information to recreate the lossily-encoded picture, compares it to the original frame, and generates an error signal that can be used by the receiver to fix smaller errors.
- Decoder must keep some recently decoded frames in memory so that it can retrieve the pixel values of macro-blocks.



- This internal memory saves a great deal of transmission and storage bandwidth

Concept of Block motion estimation

- To find the best match between regions in the two frames.
- We divide the current frame into 16 x 16 macro blocks.
- For every macro block in the frame, to find the region in the previous frame that most closely matches the macro block.
- measure similarity using the following sum-of-differences measure
- M(i,j) is the intensity of the macro block at pixel i,j,
- S(i,j) is the intensity of the search region
- n is the size of the macro block in one dimension
- <ox, oy>is the offset between the macro block and search region
- Intensity is measured as an 8-bit luminance that represents a monochrome pixel—color information is not used in motion estimation.
- We choose the macro block position relative to the search area that gives us the smallest value for this metric. The offset at this chosen position describes a vector from the search area center to the macro block's center that is called the motion vector.

Requirements

Name	Block motion estimator
Purpose	Perform block motion estimation within a PC system
Inputs	Macroblocks and search areas
Outputs	Motion vectors
Functions	Compute motion vectors using full search algorithms
Performance	As fast as we can get
Manufacturing cost	Hundreds of dollars
Power	Powered by PC power supply
Physical size and weight	Packaged as PCI card for PC

Specification

- The specification for the system is relatively straightforward because the algorithm is simple.
- The following classes used to describe basic data types in the system
- motion vector, macro block, search area.

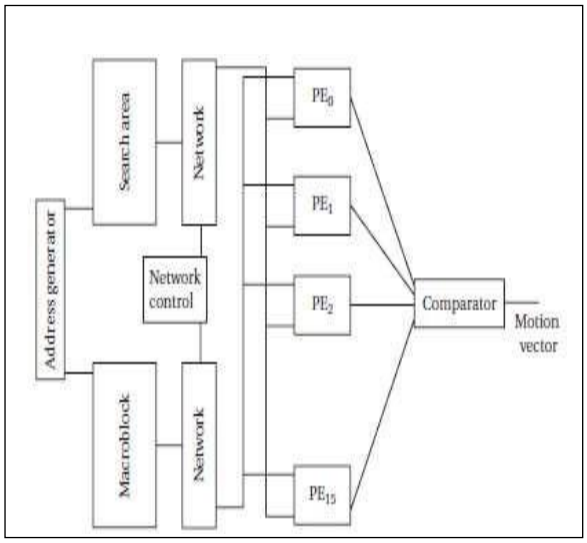
Motion-vector
x, y

Macroblock
pixels[ ]

Search-area
pixels[ ]

Architecture

- The macro block has 16 x16 = 256.
- The search area has (8 + 8 + 1 + 8 + 8)2 = 1,089 pixels.
- The FPGA probably will not have enough memory to hold 1,089 (8-bit)values.
- The machine has two memories, one for the macro block and another for the search memories.
- It has 16 processing elements that perform the difference calculation on a pair of pixels.
- The comparator sums them up and selects the best value to find the motion vector.



System testing

- Testing video algorithms requires a large amount of data.
- we are designing only a motion estimation accelerator and not a complete video compressor, it is probably easiest to use images, not video, for test data.
- use standard video tools to extract a few frames from a digitized video and store them in JPEG format.
- Open source for JPEG encoders and decoders is available.
- These programs can be modified to read JPEG images and put out pixels in the format required by your accelerator.