QUESTION BANK

Name of the Department : **Electronics and Communication Engineering**

Subject Code & Name : **EC8791 & Embedded and Real Time Systems**

Year & Semester : **IV & VII**

UNIT I INTRODUCTION TO EMBEDDEDSYSTEM DESIGN

PART-A

1. What is an embedded computer system?

It is any device that includes a programmable computer but is not itself intended to be a general-purpose computer. Thus, a PC is not itself an embedded computing system, although PCs are often used to build embedded computing systems. But a fax machine or a clock built from a microprocessor is an embedded computing system.

2. Quote some processors used in complex embedded system.

The first microprocessor, the Intel 4004, was designed for an embedded application, namely, a calculator.

An 8-bit ***microcontroller*** is designed for low-cost applications and includes on-board memory and I/O devices;

A 16-bit microcontroller is often used for more sophisticated applications that may require either longer word lengths or off-chip I/O and memory; and

A 32-bit ***RISC*** microprocessor offers very high performance for computation-intensive applications. Digital television makes extensive use of embedded processors.

A high-end automobile may have 100 microprocessors, but even inexpensive cars today use 40 microprocessors.

3. List the characteristics of embedded computing applications.

Embedded computing systems have to provide sophisticated functionality:

- Complex algorithms
- User interface

Embedded computing operations must often be performed to meet deadlines:

- Real time
- Multirate

Costs of various sorts are also very important:

- Manufacturing cost
- Power and energy

4. Summarize the challenges in embedded computing system design?

Some 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 upgradability?
- Does it really work?
- Complex testing
- Limited observability and controllability
- Restricted development environments

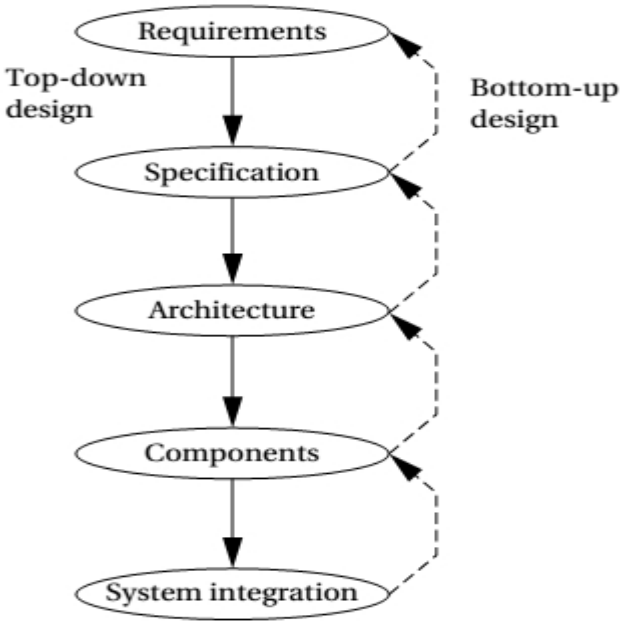
5. How does the performance in general purpose computing differ from the performance of embedded computing systems?

Most general-purpose programmers use no tools that are designed to help them improve the performance of their programs. Embedded system designers, in contrast, have a very clear performance goal in mind—their program must meet its ***deadline***. At the heart of embedded computing is ***real-time computing***, which is the science and art of programming to deadlines. We need tools to help us analyze the real-time performance of embedded systems**.**

6. Give the components that describe performance in embedded computing?

- CPU
- Platform: The platform includes the bus and I/O devices.
- Program
- Task
- Multiprocessor

7. Summarize the major steps in the embedded system design process?



8. Summarize the major goals of embedded system design?

- manufacturing cost
- performance (both overall speed and deadlines); and
- power consumption.

9. What are the entries of a requirement form?

Name, Purpose, Inputs and outputs, Functions, Performance, Manufacturing cost, Power, Physical size and weight.

10. Define UML.

The ***Unified Modeling Language(UML)*** is an ***object-oriented*** modeling language. It is a visual language that can be used to capture all the design tasks.

11. Describe the relationship that exists 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.

12. What are the three types of events defined by UML?

A ***signal*** is an asynchronous occurrence. A ***call event*** follows the model of a procedure call in a programming language. A ***time-out event*** causes the machine to leave a state after a certain amount of time.

13. What are the basic set of requirements for model train controller?

- The console shall be able to control up to eight trains on a single track.
- The speed of each train shall be controllable by a throttle to at least 63 different levels in each direction (forward and reverse).
- There shall be an inertia control that shall allow the user to adjust the responsiveness of the train to commanded changes in speed. Higher inertia means that the train responds more slowly to a change in the throttle, simulating the inertia of a large train. The inertia control will provide atleast eight different levels.
- There shall be an emergency stop button.
- An error detection scheme will be used to transmit messages.

14. What are the various issues in real time computing?

- Real-time Response.
- Recovering from Failures.
- Working with Distributed Architectures.
- Asynchronous Communication.
- Race Conditions and Timing.

15. What are the goals of system design process?

The obvious goal of a design process is to create a product that does something useful. Typical specifications for a product will include functionality, manufacturing cost, performance, power consumption. A design process has several important goals be yond function, performance, and power. Three of these goals are: Time-to-market, Design cost, Quality.

16. What is a design flow?

A ***design flow*** is a sequence of steps to be followed during a design. Some of the steps can be performed by tools, such as compilers or CAD systems; other steps can be performed by hand.

17. Give different types of design flows in a software development process.

- Waterfall model

- Spiral model
- Successive refinement design methodology
- Hierarchical design flow
- Concurrent engineering

18. What are the major phases of waterfall model?

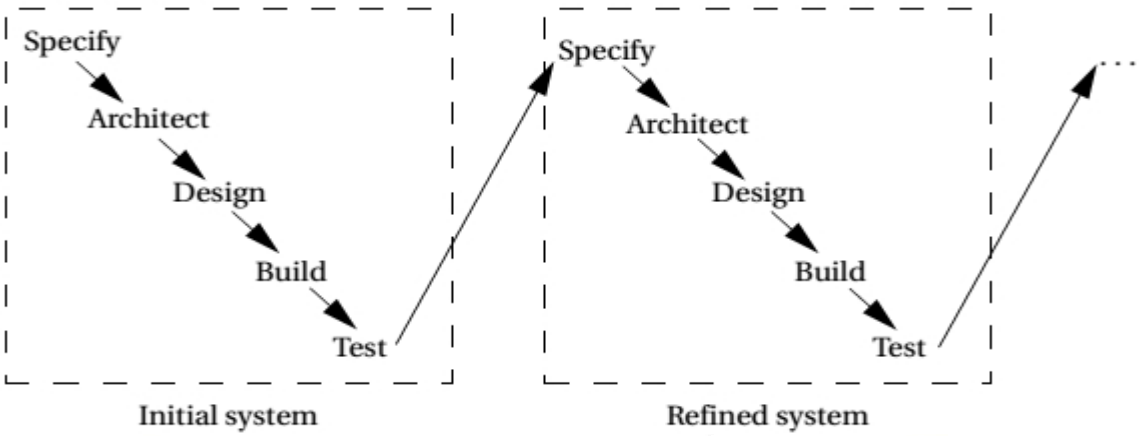
- requirements analysis
- architecture design
- coding
- testing
- maintenance

19. Differentiate waterfall model with spiral model.

While the waterfall model assumes that the system is built once in its entirety, the spiral model assumes that several versions of the system will be built. As design progresses, more complex systems will be constructed. At 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, while 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 are often necessary to add enough detail to complete a design. However, a spiral methodology with too many spirals may take too long when design time is a major requirement.

20. Draw the design flow of successive refinement development model.



21. What do you know about concurrent engineering?

Concurrent engineering attempts to take a broader approach and optimize the total flow. Reduced design time is an important goal for concurrent engineering. It tries to eliminate "over-the-wall" design steps, in which one designer performs an isolated task and then throws the result over the wall to the next designer, with little interaction between the two. In particular, reaping the most benefits from concurrent engineering usually requires eliminating the wall between design and manufacturing.

22. What are requirements and specifications in the design process?

Requirements are informal descriptions of what the customer wants.

Specifications are more detailed, precise, and consistent descriptions of the system that can be used to create the architecture.

Both requirements and specifications are, however, directed to the outward behavior of the system, not its internal structure.

23. List some techniques used for specification during system design.

Control-Oriented Specification Languages: i) state machine specification language – ***SDL language,***

ii)The ***Statechart*** - a technique for state-based specification

24. Briefly describe about requirement types in the design process.

We have two types of requirements: ***functional*** and ***nonfunctional***.

A functional requirement states what the system must do, such as compute an FFT.

A nonfunctional requirement can be any number of other attributes, including physical size, cost, power consumption, design time, reliability, and so on.

25. What is SDL specification?

SDL language was developed by the communications industry for specifying communication protocols, telephone systems, and so forth. SDL specifications include states, actions, and both conditional and unconditional transitions between states. SDL is an event-oriented state machine model since transitions between states are caused by internal and external events.

26. What is the purpose of using statechart?

The ***Statechart*** is one well-known technique for state-based specification to eliminate clutter and clarify the important structure of a state-based specification. The Statechart notation uses an event-driven model. Statecharts allow states to be grouped together to show common functionality. There are two basic groupings: OR state and AND state.

27. What is CRC card?

The ***CRC card*** methodology is a well-known and useful way to help analyze a system's structure. It is particularly well suited to object-oriented design since it encourages the encapsulation of data and functions. The acronym *CRC* stands for the following three major items that the methodology tries to identify:

■ *Classes* define the logical groupings of data and functionality.
■ *Responsibilities* describe what the classes do.
■ *Collaborators* are the other classes with which a given class works.

28. What are the steps to be followed while using CRC card to analyze a system?

1. Develop an initial list of classes
2. Write an initial list of responsibilities and collaborators
3. Create some usage scenarios
4. Walk through the scenarios
5. Refine the classes, responsibilities, and collaborators
6. Add class relationships

29. What are the standard techniques available for quality assurance?

1.The International Standards Organization (ISO) has created a set of quality standards known as ISO 9000. ISO 9000 was created to apply to a broad range of industries, including but not limited to embedded hardware and software. ISO 9000 concentrates on processes used to create the product or service. The processes used to satisfy ISO 9000 affect the entire organization as well as the individual steps taken during design and manufacturing.

2.***Capability Maturity Model (CMM)*** developed by Carnegie Mellon University's Software Engineering Institute. The CMM provides a model for judging an organization. The CMM provides a benchmark by which organizations can judge themselves and use that information for improvement.

30. List out the observations about quality management based on ISO 9000 standard.

- Process is crucial
- Documentation is important
- Communication is important

31. What are the CMM maturity levels?

CMM defines the following five levels of maturity:

1. Initial
2. Repeatable
3. Defined
4. Managed
5. Optimizing

32. What is a design review?

The ***design review*** is a critical component of any QA process. The design review is a simple, low-cost way to catch bugs early in the design process. A design review is simply a meeting in which team members discuss a design, reviewing how a component of the system works.

33. What are the factors to be considered while choosing a platform for embedded system?

Hardware:

- CPU
- Bus
- Memory
- Input and output devices

Software:

- Run time components: the operating system, code libraries
- Support components: code development environment, debugging tools

34. What is the meaning of Intellectual Property (IP)?

Intellectual Property is something that we can own but not touch: software, netlists, and so on.

Examples:

- Run-time software libraries
- Software development environments
- Schematics, netlists, and other hardware design information.

35. What is a 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.

36. Name some embedded system debugging tools.

- The USB serial port
- breakpoint
- LEDs
- microprocessor in-circuit emulator (ICE)
- Logic analyzer

37. What is 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.

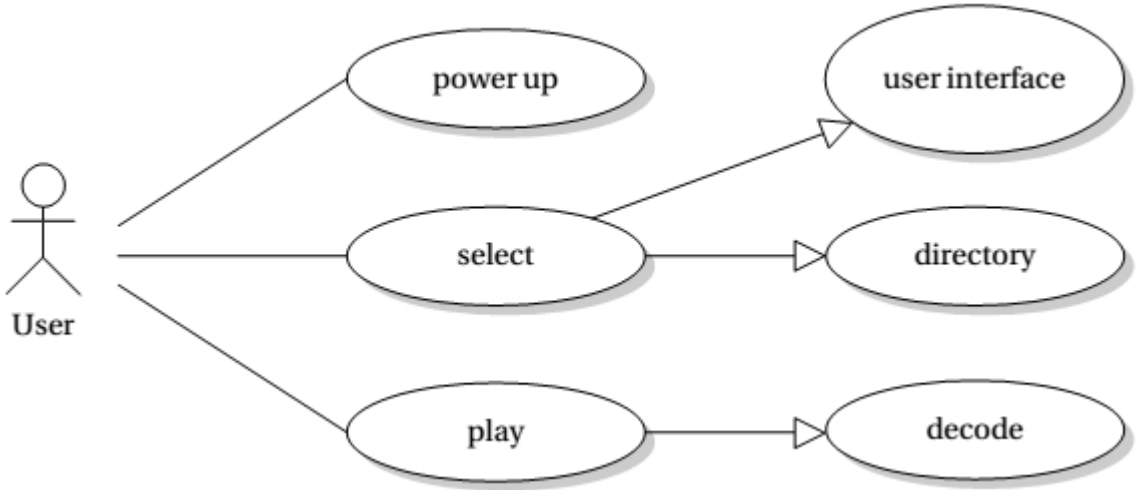
38. State the purpose of logic analyzer.

The ***logic analyzer*** is the other major piece of instrumentation in the embedded system designer's arsenal. The logic analyzer records the values on the signals into an internal memory and then displays the results on a display once the memory is full or the run is aborted. Logic analyzers typically provide a number of formats for viewing data. One format is a timing diagram format. Many logic analyzers allow not only customized displays, such as giving names to signals, but also more advanced display options. For example, an inverse assembler can be used to turn vector values into microprocessor instructions.

39. What are the challenges in debugging environment?

Logical errors in software can be hard to track down, but errors in real - time code can create problems that are even harder to diagnose. Real-time programs are required to finish their work within a certain amount of time; if they run too long, they can create very unexpected behavior. The exact results of missing real-time deadlines depend on the detailed characteristics of the I/O devices and the nature of the timing violation. This makes debugging real-time problems especially difficult.

40. Draw the use case for playing multimedia.



41. What is flash memory?

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 electric charge using a specialized capacitor that can store the charge for years. The flash memory cell does not require an external power supply to maintain its value. Furthermore, the memory can be written electrically and, unlike previous generations of electrically-erasable semiconductor memory, can be written using standard power supply voltages and so does not need to be disconnected during programming.

42. What are the basic measures of platform level performance analysis?

The most basic measure of performance is bandwidth—the rate at which we can move data.

- Bus bandwidth
- Component bandwidth

PART-B

1. Explain about the embedded system design process with suitable diagrams.
2. Describe the structural and behavioral descriptions of methods used for designing an embedded system.
3. Explain in detail the design steps of Model Train controller with suitable diagrams.
4. Explain about various software development models (or) design flows during software development process.
5. (i) Explain in detail about some advanced techniques for specification.
   (ii) Categorize the steps involved in system analysis using CRC card.
6. Explain the quality assurance techniques in detail.
7. Write short notes on consumer electronics system architecture.
8. Explain debugging techniques and challenges in embedded system computing platform design.
9. Explain in detail about platform-level performance analysis.

UNIT II ARM PROCESSOR AND PERIPHERALS

PART-A

1. List out the important features that make ARM ideal for embedded applications.

ARM stands for 'Advanced RISC Machine'. ARM is the most popular of the 32-bit processors in the market.

The outstanding features of this RISC processor are as follows:

i) The number of transistors needed is much less than that of a CISC processor of comparable computational power.
ii) The die size is less because of the reduced hardware involved.
iii) Due to these aspects power dissipation is very low.
iv) Data bus width
v) Computational capability
vi) Pipelining
vii) Multiple register instructions
vii) DSP enhancements

2. List out the features of ARM architecture versions.

Table 10.3 | Features of the Architecture Variants of ARM

Architecture Versions	Features
v4	ARM instructions only
v4T	THUMB instructions also added
v5	More advanced ARM and THUMB instructions
v5E	Advanced ARM instructions and enhanced DSP instructions
v6	Advanced ARM and THUMB. SIMD and memory support instructions added
v7	THUMB-2 technology, in which both 16-bit and 32-bit instructions are supported, and there is no need to switching between ARM and THUMB instruction sets

3. Illustrate the three different profiles of ARM cortex Processor.

The following three profiles of CORTEX series have been defined:

i) **The A profile:** This profile which has the ARMv7-A architecture is meant for high end applications. Typical applications requiring such a profile are mobile phones and video systems.

ii) **The R profile:** This profile which has the ARMv7-R architecture has been designed for high-end applications which require real-time capabilities. Typical applications are automatic braking systems and other safety critical applications.

iii) **The M profile:** This profile which has the ARMv7-M architecture has been designed for deeply embedded microcontroller type systems. This is to be used in industrial control applications.

4. Illustrate the three stage pipeline in ARM7.

ARM7 has a 3-stage pipeline. The basic task that any processor does is 'fetch, decode and execute'. In the simplest form of pipelining (3 stage), all the three stages are active all the time. While the first stage is fetching an instruction, the next stage, that is,. the decode stage, is busy with the decoding of the previously fetched instruction, and the execute stage is executing the instruction which had been previously decoded. Thus at any time, there are three instructions simultaneously present in the pipeline, at different levels of processing.

5. List out the operating modes of ARM.

ARM has seven operating modes.

i) **User**: Unprivileged mode under which most tasks run.

ii) **FIQ (Fast Interrupt Request):** Entered on a high priority (fast) interrupt request.

iii) **IRQ (Interrupt Request):** Entered on a low priority interrupt request.

iv) **Supervisor**: Entered on reset and when a software interrupt instruction (SWI) is executed.

v) **Abort**: Used to handle memory access violations.

vi) **Undef**: Used to handle undefined instructions.

vii) **System**: Privileged mode using the same registers as user mode.

6. Compare Von Neumann architecture with Harvard architecture.

Von Neumann architecture:

A computer whose memory holds both data and instructions is known as a Von Neumann machine.

The CPU has several internal registers that store values used internally. One of those registers is the program counter (PC),which holds the address in memory of an instruction. The program counter does not directly determine what the machine does next, but only indirectly by pointing to an instruction in memory. By changing only the instructions, we can change what the CPU does.

Harvard architecture:

A Harvard machine has separate memories for data and program. The program counter points to program memory, not data memory. As a result, it is harder to write self-modifying programs (programs that write data values, then use those values as instructions) on Harvard machines. Having two memories with separate ports provides higher memory bandwidth; not making data and memory compete for the same port also makes it easier to move the data at the proper times.

7. Compare CISC with RISC.

CISC: ***complex instruction set computers (CISC).*** These machines provided a variety of instructions that may perform very complex tasks, such as string searching; they also generally used a number of different instruction formats of varying lengths.

RISC: ***reduced instruction set computers (RISC).*** These computers tended to provide somewhat fewer and simpler instructions. The instructions were also chosen so that they could be efficiently executed in ***pipelined*** processors.

8. What are the instruction set characteristics useful for embedded programming?

Instructions can have a variety of characteristics, including:

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

9. How does the little-endian mode differ from big-endian mode?

The ARM processor can be configured at power-up to address the bytes in a word in either ***little-endian*** mode (with the lowest-order byte residing in the low-order bits of the word) or ***big-endian*** mode (the lowest-order byte stored in the highest bits of the word).

10. What is load-store architecture?

In the ARM processor, arithmetic and logical operations cannot be performed directly on memory locations. While some processors allow such operations to directly reference main memory, 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.

11. Name the registers set of ARM processor.

ARM has 37 registers each of which is 32 bits long. They are listed as follows:

i) 1 dedicated program counter (PC)

ii) 1 dedicated current program status register (CPSR)

iii) 5 dedicated saved program status registers (SPSR)

iv) 30 general purpose registers

12. List out the user mode registers of ARM processor.

Table 10.4 | Registers in the User Mode

Register Numbers	Designations
R0–R12	General purpose registers
R13	Stack pointer (SP)
R14	Link register (LR)
R15	Program counter (PC)

13. What is the purpose of link register of ARM processor.

R14 is the link register (LR), a special register. It is used whether there is a procedure call or an interrupt, that is, branching to a location.

When branching becomes necessary, the value of PC is saved in the link register. When returning to the original sequence, the PC value can be retrieved from the link register. Having such a register, that is, the LR, to store return addresses helps to reduce the delay associated with procedure calls and interrupts.

14. Draw the CPSR format of ARM processor.

Figure 10.6 | Current Program Status Register (CPSR) bit configuration

Table 10.5 | CPSR Bits

Bit Nos.	Notation	Interpretation
0 to 4	Mode	Specifies the current mode of operation
5	T	Specifies whether in ARM(T =) or Thumb(T = 0) state
6	F	Disables (F = 1)FIQ
7	I	Disables (I = 1)IRQ
8 to 23, 25 to 26	Undefined	
24	J	In Jazelle state (J = 1)
27	Q	Sticky overflow flag
28 to 31	V, C, Z, N	Conditional flags

15. State the purpose of using SPSR.

There are five 'Saved Program Status Registers', that is, one for each of the 'exception' modes of operation. When an exception, that is, an interrupt occurs, the corresponding SPSR saves the current CPSR value into it (so as to be able to retrieve it on returning to the previous mode).

16. Classify the instruction set of ARM processor.

The ARM instruction set can be broadly classified as follows:

i) Data processing instructions
ii) Load store instructions—single register, multiple register
iii) Branch instructions
iv) Status register access instructions

17. What is the function of barrel shifter? List some of the barrel shifter instructions of ARM.

The barrel shifter can do shifting and rotation. Two types of shifts are possible: logical and arithmetic.

LSL	Logical shift left (zero fill)
LSR	Logical shift right (zero fill)
ASL	Arithmetic shift left
ASR	Arithmetic shift right
ROR	Rotate right
RRX	Rotate right extended with C

Shift/rotate

18. Write down the significance of TST instruction.

TST is an instruction similar to compare, but it does ANDing and then sets conditional flags. If the result of ANDing is a zero, the Z flag is set. It can be used to verify if at least one of the bits of a data word is set or not.

19. Outline the significance of TEQ instruction.

TEQ does exclusive ORing which tests for equality. If both the operands are equal, the Z flag is set. It verifies if the value in a register is equal to a specified number.

20. List the branch instructions of ARM.

The B (branch) instruction is the basic mechanism in ARM for changing the flow of control.

Table 10.13 | List of Branch Instructions

Mnemonic	Instruction
B	Branch
BL	Branch and link
BX	Branch and Exchange
BLX	Branch Exchange with link

21. List the load and store instructions of ARM.

In ARM, data is brought into registers using a load instruction. After computation, the result can be 'stored' in memory. The only instructions which access RAM are 'load' and 'store'.

Table 10.16 | List of Load and Store Instructions

LDR	Load Word	STR	Store Word
LDRH	Load Half Word	STRH	Store Half Word
LDRSH	Load Signed Half Word		
LDRB	Load Byte	STRB	Store Byte
LDRSB	Load Signed Byte		

22. List the arithmetic and logical instructions of ARM.

Addition and subtraction are three operand instructions. Multiplication is a complex operation. Division is another complex instruction.

Table 10.8 | List of Arithmetic Instructions

Instruction	Operation	Calculation
ADD R3, R4, R5	Add	R3 = R4 + R5
ADC R3, R4, R5	Add with carry	R3 = R4 + R5 + C
SUB R3, R4, R5	Subtract	R3 = R4 – R5
SBC R3, R4, R5	Subtract with carry	R3 = R4 – R5 – C
RSB R3, R4, R5	Reverse subtract	R3 = R5 – R4
RSC R3, R4, R5	Reverse subtract with carry	R3 = R5 – R4 – C

Table 10.9 | List of Logical Instructions

Instruction	Operation	Logical Result
AND R3, R4, R5	Logical AND of 32 bit values	R3 = R4 AND R5
ORR R3, R4, R5	Logical OR of 32 bit values	R3 = R4 OR R5
EOR R3, R4, R5	Logical XOR of 32 bit values	R3 = R4 XOR R5
BIC R3, R4, R5	Logical bit clear	R3 = R4 (AND NOT) R5

23. List the compare instructions of ARM.

Table 10.10 | List of 'Compare' Instructions

CMP R3, R4	Compare	R3 – R4, but only flags affected
CMN R3, R4	Compare negated	R3 + R4, but only flags affected
TST R3, R4	Test	R3 AND R4 but only flags affected
TEQ R3, R4	Test Equivalence	R3 OR R4 but only flags affected

Table 10.11 | Flag Settings After a Compare Instruction

If	C	Z
R3 > R4	1	0
R3 < R4	0	0
R3 = R4	1	1

24. What is stack?

A stack is an area in memory. Most stacks are Last-In First-Out (LIFO) type stacks. This means that the last data that was stored is the first one that can be taken out. It is sequential access that is done, and not random access. 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.

25. Define Program Counter.

Program Counter is the special function of the General-Purpose Register R15. It is a 32bit register. It holds the address of the next instruction to be executed. The increment value of the PC is determined based on the instruction executed ie. for ARM instructions the value is 4, thumb instructions the value is 2 and for Jazelle instructions the value is 1.

26. What are the types of stacks?

Types of Stacks: Ascending/Descending and Empty/Full

An ascending stack grows upwards. It starts from a low memory address and, as items are pushed onto it, progresses to higher memory addresses.

A descending stack grows downwards. It starts from a high memory address, and as items are pushed onto it, it progresses to lower memory addresses.

In an empty stack, the stack pointer points to the next free (empty) location on the stack, i.e., to the place where the next item to be pushed, will be stored.

In a full stack, the stack pointer points to the topmost item in the stack, that is the location of the last item pushed onto the stack. In practice, stacks are almost always full and

descending.

Most stacks are 'Full descending' types.

27. What is stack pointer?

The stack has a pointer to its top which is called the Stack pointer (SP). For ARM, this is register R13. This means that the address of the top of the stack is to be available in SP.

28. Draw the sequence of actions needed for a nested procedure.

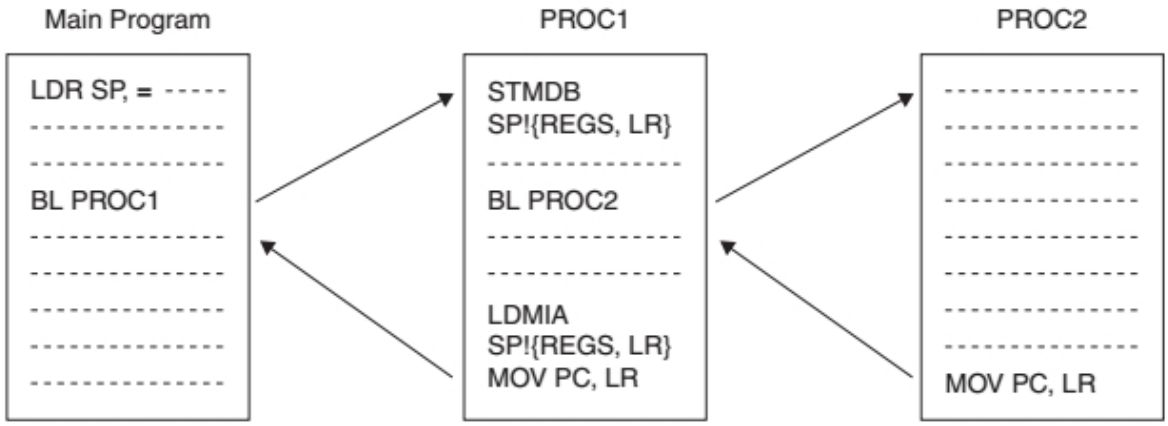


Figure 10.20 | Sequence of actions needed for a nested procedure

29. What is the purpose of current program status register(CPSR)and Z-bit?

CPSR: It is set automatically during every arithmetic, logical or shifting operations. The top 4 bits of CPSR hold useful information about the result.

Z-Bit: The zero(Z)bit is set when every bit of the result is zero.

30. What is LPC214x family?

LPC is a family of 32bit ARM7 Microcontrollers founded by NXP Semiconductors (formerly known as Philips Semiconductors). All the LPC family of semiconductors have common features namely the on chip S-RAM, on chip Flash Memory, ISP, A/D, D/A, USB, External/Internal Timers, PWM Module, Capture/Compare Module etc.

31. Mention the features of LPC214x.

The main features provided by this family are listed below:

i) The core ARM 7TDMI-S in a tiny LQFP64 package

ii) 8 KB to 40 KB of on-chip static RAM

iii) 32 KB to 512 KB of on-chip flash memory

iv) 128-bit wide interface/accelerator enables high-speed 60 MHz operation

v) USB 2.0 Full-speed compliant device controller with 2 KB of endpoint RAM.

Additional features are:

i) 10-bit ADCs provide a total of 6/14 analog inputs

ii) Single 10-bit DAC provides variable analog output (LPC2142/44/46/48 only)

iii) Two 32-bit timers/external event counters (with four capture and four compare channels each), PWM unit (six outputs) and watchdog

iv) Low power real-time clock (RTC) with independent power and 32 kHz clock input

v) Multiple serial interfaces including two UARTs (16C550), two fast I2C-bus(400 Kbit/s), SPI and SSP with buffering and variable data length capabilities

vi) Vectored interrupt controller (VIC) with configurable priorities and vector addresses

vii) Up to 45 of 5 V tolerant fast general purpose I/O pins in a tiny LQFP64 package
viii) Up to 21 external interrupt pins available
ix) 60 MHz maximum CPU clock available from programmable on-chip PLL
x) On-chip integrated oscillator operates with an external crystal from 1 MHz to 25 MHz
xi) Power saving modes include idle and power-down
xii) Processor wake-up from power-down mode via external interrupt or BOD
xiii) Single power supply chip with POR and BOD circuits

32. Distinguish between the power-down and idle modes of LPC214x.

One of the main features of ARM processors are their low-power dissipation.

Low power modes are also available: They are the idle mode and power-down mode.

In the idle mode, instruction execution is suspended until either a reset or interrupt occurs. But peripheral functions can continue operation and may generate interrupts to cause the processor to resume execution. The idle mode eliminates power used by the processor, memory systems, related controllers and internal buses.

In the power-down mode, the oscillator is shut down and the chip receives no internal clocks. This mode can be terminated and normal operation resumed by either a reset or certain specific interrupts that are able to function without clocks. Since all dynamic operation of the chip is suspended, this mode reduces chip power consumption to almost zero.

33. Distinguish between the various internal buses of LPC214x in terms of speed and function.

Advanced Microcontroller Bus Architecture' or AMBA is a standard defined by ARM in 1996, for on-chip buses in its SoC designs. Three buses with different protocols and speeds have been defined.

The fastest bus is the system or the local bus, which connects the processor core with memory, as memory accesses have to be very fast. In the LPC 21xx series, a GPIO (General Purpose I/O) block is also connected to the local bus. This permits peripherals connected to this fast GPIO block, to use the high speed of the local bus.

Along with the core, an AHB bridge is seen which defines the high speed AMBA's 'Advanced High performance Bus' facilitating the 'Vectored Interrupt Controller'.

The third bus is the VPB bus which stands for VLSI Peripheral Bus; there is a bridge which communicates between the low speed VPB bus and the higher speed AHB bus. The VPB is the one that connects to all the peripherals of the LPC 214x SoC.

34. What is the necessity for having the MAM module? How does it function?

Instructions are executed by fetching them from memory. In a typical MCU, program code, that is, the instructions are stored in flash memory. Flash memory is rather slow, which means that program execution gets slowed down. The solution is to have the 'memory accelerator module'. In simple terms, the memory accelerator module (MAM) attempts to have the next ARM instruction that will be needed, in its latches in time to prevent CPU fetch stalls.

35. What is PCLK and CCLK?

The processor clock is designated as CCLK, while the peripheral clock is called PCLK. On reset, PCLK is ¼ of CCLK.

36. What is the purpose of VPB bus and divider?

VPB bus divider is a register whose settings can be used to divide the output frequency of the PLL so as to get a reduced clock frequency (1/2 to ¼) for the VPB peripherals which need to operate at a frequency lower than the processor.

37. What is a Watch dog timer?

Watch Dog timer is used to reset the processor whenever it enters into a locked state in case of hardware fault or program error to prevent failure of the system. The timer is regularly cleared if no errors are encountered, once the overflow of the timer occurs the system is forcefully reset.

38. What are the notable features of LPC214x memory map?

Each of the LPC214x peripherals has addresses and the peripherals use the memory mapped I/O scheme of addressing– this means that both memory and I/O share the same address space. The total address space is from 0x0 to 0xFFFFFFF, i.e., a 4GB space.

i) The 512KB of flash (non-volatile) memory has an address starting from 0x0.

ii) The 40 KB of static RAM has the addresses starting at 0x4000 0000.

iii) There is a RAM allotted for 'USB with DMA' applications.

iv) The peripherals attached to the AHB have addresses from 0xF000 0000.

v) The peripherals attached to the lower speed VPB bus have the addresses from 0xE000 0000.

39. Mention the interrupts in ARM.

ARM7 supports two types of interrupts: fast interrupt requests (FIQs) and interrupt requests (IRQs). An FIQ takes priority over an IRQ.

40. What do you know about traps?

A ***trap***, also known as a ***software interrupt***, is an instruction that explicitly generates an exception condition. The most common use of a trap is to enter supervisor mode. The entry into supervisor mode must be controlled to maintain security.

41. What is the purpose of supervisor mode?

Normal programs run in ***user mode***. The supervisor mode has privileges that user modes do not. Control of the memory management unit (MMU) is typically reserved for supervisor mode to avoid the obvious problems that could occur when program bugs cause inadvertent changes in the memory management registers. Not all CPUs have supervisor modes. The ARM, however, does have such a mode. The ARM instruction that puts the CPU in supervisor mode is called SWI. SWI causes the CPU to go into supervisor mode and sets the PC to 0x08. The argument to SWI is a 24-bit immediate value that is passed on to the supervisor mode code; it allows the program to request various services from the supervisor mode.

42. Summarize any five peripherals in the LPC 2148 MCU.

a) GPIO (General Purpose I/O)

b) The Timer Unit

c) Vectored Interrupt Controller (VIC)

d) The Pulse Width Modulation Unit

e) UART

43. What is the purpose of pin connect block?

The purpose of this is to configure the pins to the desired functions. This acts like a multiplexer. Each pin of the chip has a maximum of four functions. The select pins function is provided by the bits of the PINSEL registers.

Table 11.2 | Generalization of the Function of the Pinselect Register Bits

PINSEL0 and PINSEL1	Function
00	Primary function, typically GPIO
01	First alternate function
10	Second alternate function
11	Reserved

44. List the GPIO pin registers of LPC214x.

i) **IODIR (IO Direction register):** The bit setting of this register decides whether a pin is to be an input(0) or output(1).

ii) **IOSET (IO Set register):** This register is used to set the output pins of the chip. To make a pin to be '1', the corresponding bit in the register is to be '1'. Writing zeros have no effect.

iii) **IOCLR (IO Clear register):** To make an output pin to have a '0' value, i.e., to clear it. The corresponding bit in this register has to be a '1'. Writing zeros have no effect.

iv) **IOPIN (IO Pin register):** From this register, the value of the corresponding pin can be read, irrespective of whether the pin is an input or output pin.

45. State the operation of the timer unit in the LPC 2148 MCU.

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. The first step in timer operation is to load a number into what is called a 'match register'. Then a timer count register is started. This register keeps incrementing for each PCLK cycle or a lower rate pre-scaled cycle. When the content of this timer count register becomes equal to the value in the match register, i.e., a match occurs, the delay that occurs from the starting time can be used for our 'timing'.

46. How many timers does the LPC2148 have?

The LPC2148 or the ARM7 has a total of two 32bit timer/counters. The timer operates with the PCLK (Peripheral Clock) or external clock. The timer/counters are Timer0/Counter0 and Timer1/Counter1.

47. How does the pre-scalar in a timer unit function in the LPC 2148 MCU?

To get a lower frequency output, there is the facility of using the prescale counter. There are two registers associated with 'prescaling'—the prescale counter and the prescale register.

The 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. This causes the T0TC to increment on every PCLK when PR = 0, every 2 PCLKs when PR = 1, every three PCLKs when T0PR = 2, every four PCLKs when T0PR = 3 and so on.

48. What are the features of Vectored Interrupt Controller(VIC) of LPC 2148 MCU?

VIC manages all the interrupts of the ARM core (IRQs and FIQs) as well as interrupt requests from the peripherals.

Features of VIC

- 32 interrupt request inputs
- 16 vectored IRQ interrupts
- 16 priority levels dynamically assigned to interrupt requests
- Software interrupt generation

49. Define PWM

PWM stands for Pulse Width Modulation where the Width of the Pulse is modulated in accordance with the message or the information signal which is given as input. This technique is primarily used to control the speed of motors in industrial applications.

50. Define Capture mode.

The capture feature offered in LPC 214x series is used to capture/trap the timer counter value on the events like rising edge, falling edge of the input. The functions like frequency, pulse width, duty cycle of the input can be measured for the inputs.

51. Distinguish between single and double edged PWM of LPC 2148 MCU.

Single Edge Controlled PWM: Here only the falling edge of a pulse train is controlled. All single edge controlled PWM outputs go high at the beginning of a PWM cycle. Each PWM output will go low when its match value (in MR1 to MR6) is reached. 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. Actions are controlled by the settings in the PWMMCR register.

In double edge controlled PWM, both the rising and falling edges of the PWM waveform are controlled by the match registers.

52. How many PWM channels are available in ARM7 for output generation?

The ARM7 has a total of 6 PWM channels which can generate 6 different PWM waveforms simultaneously. It has 7 Match registers to control the frequency and Pulse width of the different PWM waveforms. The PWM waveform can be single edge controlled or double edge controlled.

53. Define UART in ARM7.

The ARM7 has two inbuilt UART (Universal Asynchronous Receive Transmit) modules. Each UART module has 16 bits receive and transmit FIFO registers. The data transfer happens based on the baud rate of operation defined.

54. What are the important units of UART of LPC 2148 MCU?

The three important units of LPC 2148 MCU are the transmitter, receiver and the baud rate generator blocks.

55. What are the registers of UART of LPC 2148 MCU?

- Pinselect Register (PINSEL0)
- UART0 Transmit Holding Register (U0THR)
- UART0 Divisor Latch Registers (U0DLL and U0DLM)
- UART0 FIFO Control Registers (U0FCR)
- UART0 Line Control Register (U0LCR)

56. What are the control registers of PWM unit of LPC 2148 MCU?

- PWMTCR: 8-bit PWM Timer Control Register
- PWMPCR: 16-bit PWM Control Register.

- PWMLER: 8-bit PWM Latch Enable Register.

57. What are important special function registers(SFRs) of timer unit of LPC 2148 MCU?

- 32-bit Timer Count Register–T0TC
- 8-bit Timer Control Register–T0TCR
- 32 bit Match Registers - (MR0 to MR3)
- 16-bit Match Control Register–T0MCR

58. Illustrate the GPIO ports of LPC214x.

There are two general purpose 32-bit ports, P0 and P1.

Port 0: Port 0 is a 32-bit I/O port with individual direction controls for each bit. 28 pins of the Port 0 can be used as general purpose bi-directional digital I/Os, while P0.31 provides digital output functions only. Pins P0.24, P0.26 and P0.27 are 'reserved' and not available for use.

Port 1: Port 1 is a 32-bit bi-directional I/O port with individual direction controls for each bit. Pins 0 through 15 are not available. Out of the remaining pins from 16 to 31, the pins 16 to 25 are 'reserved'.

59. What are the features of ARM9 MCU?

The ARM9 core is a more advanced member of the ARM family (compared to ARM7). It has a 5 stage pipeline and operates at a frequency range of approximately double that of ARM7. Many ARM9 cores have DSP instructions and thus are 'Enhanced' ARM9E processors. Because the core is so powerful, it is used for more complex operations and an MCU which is based on ARM9, typically has more peripherals than an ARM7 based MCU.

ARM9 board developed by NXP contains an MCU of the LPC 29xx series. 'The LPC 29xx combine an 125 MHz ARM968E-S CPU core, Full Speed USB 2.0 host and device (LPC 2927/29 only), CAN and LIN, 56 KB SRAM, up to 768 KB flash memory, external memory interface, three 10-bit ADCs, and multiple serial and parallel interfaces in a single chip'.

60. What are the features of ARM Cortex-M3 MCU?

The ARM Cortex-M3 is a next generation core that offers system enhancements such as modernized debug features and a higher level of support block integration. The LPC 17xx is an ARM Cortex-M3 based microcontroller for embedded applications requiring a high level of integration and low-power dissipation. High speed versions (LPC 1769 and LPC 1759) operate at up to a 120 MHz CPU frequency. Other versions operate at up to an 100 MHz CPU frequency.

The ARM Cortex-M3 CPU incorporates a 3-stage pipeline and uses a Harvard architecture with separate local instruction and data buses as well as a third bus for peripherals. It also includes an internal pre-fetch unit that supports speculative branches.

The peripheral of the LPC 17xx includes up to 512 kB of flash memory, up to 64 kB of data memory, Ethernet MAC, a USB interface that can be configured as either Host, Device, or OTG, 8 channel general purpose DMA controller, 4 UARTs, 2 CAN channels, 2 SSP controllers, SPI interface, 3 I2C interfaces, 2-input plus 2-output I2S interface, 8 channel 12-bit ADC, 10-bit DAC, motor control PWM, Quadrature Encoder interface, 4

general purpose timers, 6-output general purpose PWM, ultra-low power RTC with separate battery supply, and up to 70 general purpose I/O pins.

PART-B

1. Draw and explain ARM architecture in detail.
2. Classify the ARM instruction set. Explain the function of ARM processor instructions with examples.
3. Discuss about the types of stacks and subroutines supported by ARM processor.
4. Explain in detail the Pulse Width Modulation unit of ARM processor and the relevant registers.
5. Explain in detail the Timer unit of ARM processor and the relevant registers.
6. Explain in detail the UART unit of ARM processor and the relevant registers.
7. (i) With neat block diagram explain the architecture of the ARM9 CPU.
   (ii) With neat block diagram explain the architecture of the ARM Cortex M3 CPU.
8. Draw and explain the block diagram and memory map of LPC214x with its features.

UNIT III EMBEDDED PROGRAMMING

PART-A

1. Mention the different components for embedded programs.

Three structures or components that are commonly used in embedded software: the state machine, the circular buffer, and the queue. State machines are well suited to reactive systems such as user interfaces; circular buffers and queues are useful in digital signal processing.

2. Evaluate the importance of circular buffer.

Circular buffer is a buffer that always has a fixed number of data elements. The circular buffer is a data structure that lets us handle streaming data in an efficient way. At each point in time, the algorithm needs a subset of the data stream that forms a window into the stream. When the pointer gets to the end of the buffer, it wraps around to the top.

3. What is a state machine?

State Machine is a component of embedded program design. In this method the embedded system is represented as the exhaustive collection of finite states and the events which trigger the system to move from one state to another along with the respective outputs produced.

4. Evaluate the importance of Queues.

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. One way to build a queue is with a linked list. This approach allows the queue to grow to an arbitrary size. A queue may have varying numbers of elements in it.

5. What is a 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.

6. Outline the significance of CDFG?

Control/Data Flow Graph (CDFG) has constructs that model both data operations (arithmetic and other computations) and control operations (conditionals). A CDFG uses a data flow graph as an element, adding constructs to describe control.

7. State the basic principle of compilation technique.

We can summarize the compilation process with a formula:

Compilation = translation+ optimization

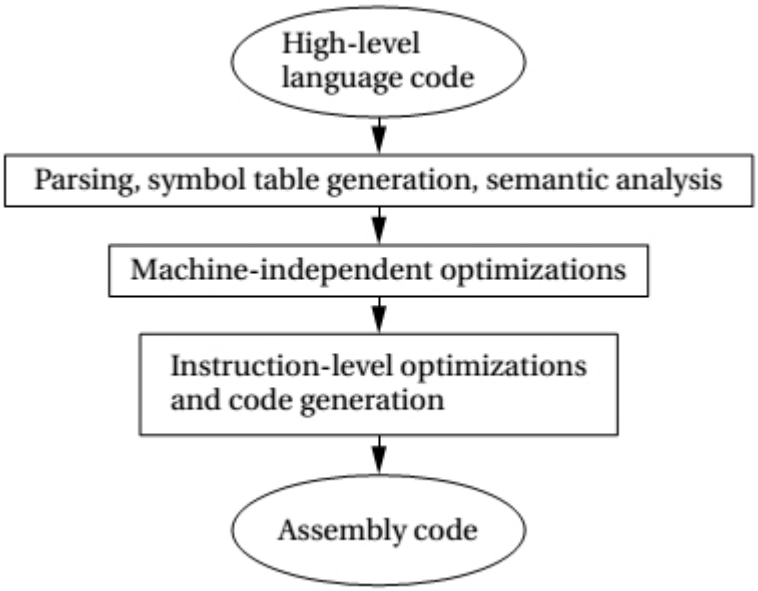
The high-level language program is translated into the lower-level form of instructions; optimizations try to generate better instruction sequences.

8. List the basic compilation methods.

- Statement translation
- Procedures

- Data structures

9. Outline the compilation process.



10. Illustrate the significance of frame for a call to procedure.

The information for a call to a procedure is known as a frame; the frames are stored on a stack. Procedure stacks are typically built to grow down from high addresses. A stack pointer (sp) defines the end of the current frame, while a frame pointer (fp) defines the end of the last frame. When a new procedure is called, the sp and fp are modified to push another frame onto the stack. The frame also holds the locally declared variables.

11. What are the different phases of compiler optimizations?

- Loop transformations
- Dead code elimination
- Register allocation
- Scheduling
- Instruction selection

12. Discuss about different forms of loop transformations for compiler optimization.

Loops are important program structures. Many techniques have been designed to optimize loops.

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.

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.

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

13. What are the last steps in the compilation process?

Assembly and linking are the last steps in the compilation process—they turn a list of instructions into an image of the program's bits in memory. Loading actually puts the program in memory so that it can be executed.

14. What is the function of assembler?

The assembler's job is to translate symbolic assembly language statements into bit – level representations of instructions known as object code. The assembler takes care of instruction formats and does part of the job of translating labels into addresses.

15. What are the two passes of assembly source code during label processing?

1. The first pass scans the code to determine the address of each label.
2. The second pass assembles the instructions using the label values computed in the first pass.

16. What is the purpose of linker?

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.

17. What is loader?

The program that brings the program into memory for execution is called a loader. Loading is the process of getting the starting address of the compiled small portions of the program to run consecutively. Also the main job of the loader is to resolve external references based on available entry points.

18. How to evaluate the program execution time?

The key to evaluating execution time is breaking the performance problem into parts. Program execution time can be seen as

execution time = program path + instruction timing.

The path is the sequence of instructions executed by the program. The instruction timing is determined based on the sequence of instructions traced by the program path, which takes into account data dependencies, pipeline behavior, and caching.

19. State the purpose of cycle-accurate simulator and instruction-level simulator.

For purposes of performance analysis, the most important type of CPU simulator is the cycle-accurate simulator, which performs a sufficiently detailed simulation of the processor's internals so that it can determine the exact number of clock cycles required for execution.

A simulator that functionally simulates instructions but does not provide timing information is known as instruction-level simulator.

20. Illustrate the need of symbol table and PLC in Assemblers.

During assembly process, the name of each symbol and its address is stored in a symbol table that is built during the first pass. The symbol table is built by scanning from the first instruction to the last. During scanning, the current location in memory is kept in a program location counter (PLC). The PLC always makes exactly one pass through the program.

21. What are the several ways to measure program performance?

We can measure program performance in several ways:

- Simulators go beyond functional simulation to measure the execution time of the program.
- A timer connected to the microprocessor bus can be used to measure performance of executing sections of code.
- A logic analyzer can be connected to the microprocessor bus to measure the start and stop times of a code segment.

22. What are the different types of performance measures on programs?

Three different types of performance measures on programs are:

- **average-case execution time:** This is the typical execution time we would expect for typical data when defining typical inputs.
- **worst-case execution time:** The longest time that the program can spend on any input sequence is clearly important for systems that must meet deadlines.
- **best-case execution time:** This measure can be important in multirate real-time systems.

23. What are the effects of caching on instruction execution time?

The access time for main memory can be 10–100 times larger than the cache access time, caching can have huge effects on instruction execution time by changing both the instruction and data access times. Caching performance inherently depends on the program's execution path because the cache's contents depend on the history of accesses.

24. What is the disadvantage of nested loops in embedded programs?

The disadvantages of nested loops are it repeats the operation with n number of times with more delay. It will reduce the system performance.

25. Define compiler and cross-compiler.

Compiler is software. It translates high level language into machine code or machine language. A cross compiler is a kind of a compiler that runs on one type of machine but generates code for another machine. After compilation, the executable code is downloaded to the embedded system by a serial link or perhaps burned in a PROM and plugged in.

26. What are the techniques for optimizing software performance?

Loop optimization and cache optimization.

27. What are the three important techniques in optimizing loops?

There are three important techniques in optimizing loops: code motion, induction variable elimination, and strength reduction.

Code motion lets us move unnecessary code out of a loop.

An induction variable is a variable whose value is derived from the loop iteration variable's value. The compiler often introduces induction variables to help it implement the loop. A nested loop is a good example of the use of induction variables.

Strength reduction helps us reduce the cost of a loop iteration. Strength reduction can often be performed with simple substitution rules.

28. List some software performance optimization strategies.

- Try to use registers efficiently.
- Make use of page mode accesses in the memory system whenever possible.
- Analyze cache behavior to find major cache conflicts. Restructure the code to eliminate as many of these as you can as follows:

❖ For instruction conflicts try to rewrite the segment to make it as small as possible so that it better fits into the cache. For conflicts across larger spans of code, try moving the instructions or padding with NOPs.

❖ For scalar data conflicts, move the data values to different locations to reduce conflicts.

❖ For array data conflicts, consider either moving the arrays or changing your array access patterns to reduce conflicts.

***29.* List the factors contributing to the 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.
- Accesses to registers are the most energy efficient
- Caches are an important factor in energy consumption.

30. What are the energy optimization strategies to improve power consumption?

- Try to use registers efficiently.
- 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.
- Software pipelining reduces pipeline stalls, thereby reducing the average energy per instruction.
- Eliminating recursive procedure calls where possible saves power by getting rid of function call overhead.

31. How power can be optimized at the program level?

Power optimization is done at the program level by many ways. Some of the ways can be given as avoiding power down mode, predictive shutdown. By avoiding power down mode much power gets saved. In predictive shutdown the power conservation is achieved by turning down the blocks when not used.

32. List some techniques used to optimize program size.

The memory footprint of a program is determined by the size of its data and instructions. Both must be considered to minimize program size.

- Data can sometimes be packed
- A very low-level technique for minimizing data is to reuse values. Data buffers can often be reused at several different points in the program.
- Encapsulating functions in subroutines can reduce program size
- Architectures that have variable-size instruction lengths are particularly good candidates for careful coding to minimize program size.
- When reducing the number of instructions in a program, one important technique is the proper use of subroutines. In some cases, proper instruction selection may reduce code size.

33. What are the major types of testing strategies?

The two major types of testing strategies :

■ ***Black-box*** methods generate tests without looking at the internal structure of the program.

■ ***Clear-box*** (also known as ***white-box***) methods generate tests based on the program structure.

34. List the types of Co-verification techniques.

Co-Verification phase is about white box testing. Techniques include statement coverage, condition coverage, and decision coverage. - Demonstration: It is about black box testing. Techniques include error guessing, boundary-value analysis, and equivalence partitioning.

35. What is clear box testing?

Clear-box (also known as **white-box**) methods generate tests based on the program structure. The control/data flow graph extracted from a program's source code is an important tool in developing clear-box tests for the program. To adequately test the program, we must exercise both its control and data operations.

We must accomplish the following three things in a test:

• Provide the program with inputs that exercise the test we are interested in.

• Execute the program to perform the test.

• Examine the outputs to determine whether the test was successful.

36. Define cyclomatic complexity in white-box testing.

A simple measure, cyclomatic complexity allows us to measure the control complexity of a program. Cyclomatic complexity is an upper bound on the size of the basis set. If e is the number of edges in the flow graph, n the number of nodes, and p the number of components in the graph, then the cyclomatic complexity is given by

$$M = e - n + 2p$$

37. What are the different measures used to execute white-box testing?

- Execution paths
- Branch testing
- Domain testing
- data flow testing
- Testing loops

38. What is black-box testing?

Black-box methods generate tests without looking at the internal structure of the program. Black-box tests are generated without knowledge of the code being tested. When used alone, black-box tests have a low probability of finding all the bugs in a program. But when used in conjunction with clear-box tests they help provide a well-rounded test set.

39. What is random testing?

Random tests form one category of black-box test. Random values are generated with a given distribution. The expected values are computed independently of the system, and then the test inputs are applied. A large number of tests must be applied for the results to be statistically significant, but the tests are easy to generate.

40. What is regression testing?

Regression tests form an extremely important category of tests. When tests are created during earlier stages in the system design or for previous versions of the system, those tests should be saved to apply to the later versions of the system. Clearly, unless the system specification changed, the new system should be able to pass old tests. In some cases old bugs can creep back into systems, such as when an old version of a software module is inadvertently installed. In other cases regression tests simply exercise the code in different ways than would be done for the current version of the code and therefore possibly exercise different bugs.

41. What is numerical accuracy in black-box testing?

Aggressive data sets can be generated to stress the numerical accuracy of the system. These tests can often be generated from the original formulas without reference to the source code.

42. How to evaluate functional tests?

One interesting method for analyzing the coverage of your tests is error injection. First, take your existing code and add bugs to it, keeping track of where the bugs were added. Then run your existing tests on the modified program. By counting the number of added bugs your tests found, you can get an idea of how effective the tests are in uncovering the bugs. If the bugs are too easy or too difficult to find or simply require different types of tests, then bug injection's results will not be relevant.

PART-B

1. Analyze the components of embedded program and discuss in detail about each component.
2. Discuss in detail about assembly, linking and loading with suitable examples.
3. (i) List the different models of Program.
   (ii) Briefly explain with neat diagrams on various models of program with examples.
4. Describe in detail about various compilation techniques.
5. Outline the Program level energy and power analysis and optimization.
6. Illustrate with necessary diagrams about the program level performance analysis.
7. Evaluate the different techniques used in software performance optimization.
8. Describe the following techniques used for program validation and testing.
   i) Black box Testing
   ii) White box Testing
9. Discuss in detail about the optimization of program size of an embedded system.

UNIT IV REAL TIME SYSTEMS

PART-A

1. What are real-time systems?

Real-time systems are systems that are bounded to give the response within a predefined period which is called as deadline. The deadline for the system is fixed considering the responsivity and the constraints of the system.

2. What are the characteristics of real-time systems?

1. Time constraints
2. Event - driven, reactive
3. Concurrency / multiprogramming
4. Reliability / fault tolerance requirements
5. Predictable behavior
6. High cost of failure

3. What is Hard Real-Time?

Hard real-time systems - A real time system in which the result is produced within predefined time period (or) within deadline. If any deadline is missed, then the system is incorrect.
Ex: Nuclear power plant control, flight control, anti-missile system.

4. What is Soft Real-Time?

Soft real-time systems - A real time system in which the result may be produced within average defined time period (or) deadline is soft. A soft

deadline may occasionally be missed.

Ex: Telephone switches, multimedia applications, web browsing, seat reservation in a railway reservation application.

5. Define Multi Rate Systems.

Multi Rate systems are systems that have more than one degree of freedom. Its state is defined by multiple state variables. In multi rate systems multiple processes run at different rates having different instantiation times and different deadlines.

6. What is a Deadline?

A deadline for a process is given as the time instant at which the process needs to be completed with its execution. The deadline is fixed based on the execution time of the process.

7. Define Release Time.

Release time of a process is given as the time instant at which the process becomes ready for execution. The process can only be taken for execution when the release time is met. Based on data dependencies further delay in execution of the process can happen

8. What is meant by response time?

The response time of a process is given as the length of time from the release time of the process to the instant when it gets completed.

9. Draw the structure of a real time system.

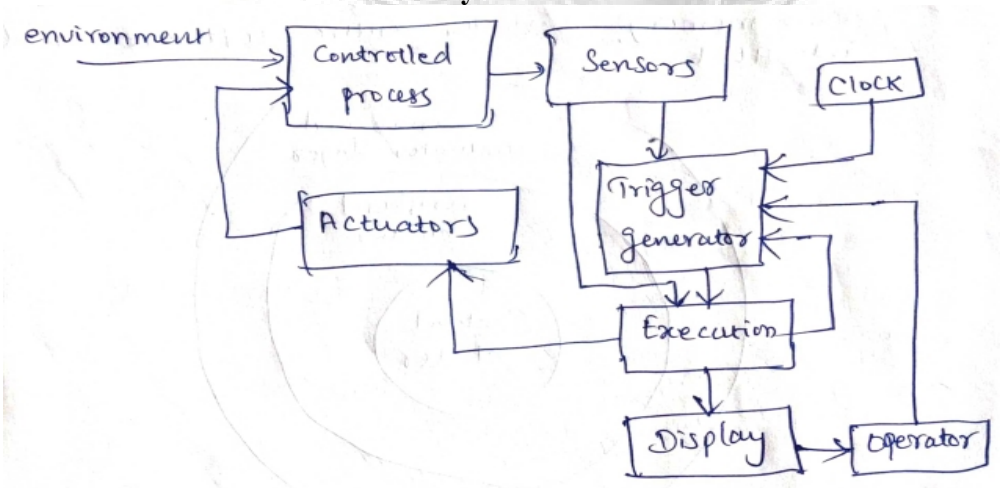


fig: A schematic block diagram of a real-time system

11. Define absolute Deadline.

Absolute deadline is the moment in time in which the job must be completed. So, absolute deadline is equal to its release time plus its relative deadline.

12. What are sporadic or aperiodic jobs?

The jobs which are triggered on the occurrence of external event are called as sporadic or aperiodic jobs. The release times of these jobs are not known until the triggering event occurs.

13. Define Periodic Task Model.

A periodic task model can be defined as the workload model whose occurrence is deterministic in nature. The scheduling algorithms for these models have better performance and well understood behaviour.

14. Define Relative Deadline.

The maximum allowable response time of the process can be given as the relative deadline of the process. Relative deadline of a process is considered when multiple process executions are happening simultaneously.

15. Give some applications of real time systems.

1. Industrial applications - Process control systems, industrial automation systems, SCADA applications, test and measurement equipment, and robotic equipment.
2. Telecommunication applications - Cellular system, etc.
3. Aerospace - avionics, flight simulation, satellite tracking systems etc.
4. Internet and multimedia applications:- video conferencing, Internet routers and switches etc.
5. Consumer electronics - Set-top boxes, washing machines, microwave ovens, home security systems, air conditioning, cell phones etc.
6. Defence applications:- missile guidance systems, anti-missile systems, satellite based surveillance systems.
7. Miscellaneous applications - Railway reservation system.

16. What are the factors for estimating program run times?

Estimating the execution time of any given program depends on the following factors:

1. Source code - Source code that is carefully tuned and optimized takes less time to execute.
2. Compiler – The compiler maps the source-level code into a machine-level program. This mapping is not unique; Also the execution time will depend on the nature of the mapping.
3. Machine architecture – Many aspects of the machine architecture have an effect on the execution time. The time spent waiting to gain access to the network affects execution time. The size and organization of the cache will also affect the memory-access time. Also, many machines use dynamic RAM in which the refresh task has priority over the CPU and thus affects the memory-access time.
4. Operating system – The operating system determines task scheduling and memory management, both of which have a major impact on the execution time. Along with the machine architecture, it determines the interrupt handling overhead.

17. What is called as a scheduler?

Jobs are scheduled and allocated resources according to a chosen set of scheduling algorithms and resource access-control protocols. The module which implements these algorithms is called the *scheduler*.

18. What are the task assignment/scheduling characteristics?

A task assignment / scheduling characteristics are:

(i) Feasible schedule - a valid schedule

(ii) Tasks are allocated to various processors and scheduled on them.
- Online (dynamically) scheduling or offline (precomputed) scheduling.

(iii) Static-priority or dynamic priority algorithms – Ex: Rate-Monotonic (static) algorithm and Earliest Deadline First (EDF) algorithm (dynamic).

(iv) Pre-emptive or non-preemptive scheduling.

19. Draw the schematic of a timing estimation system.

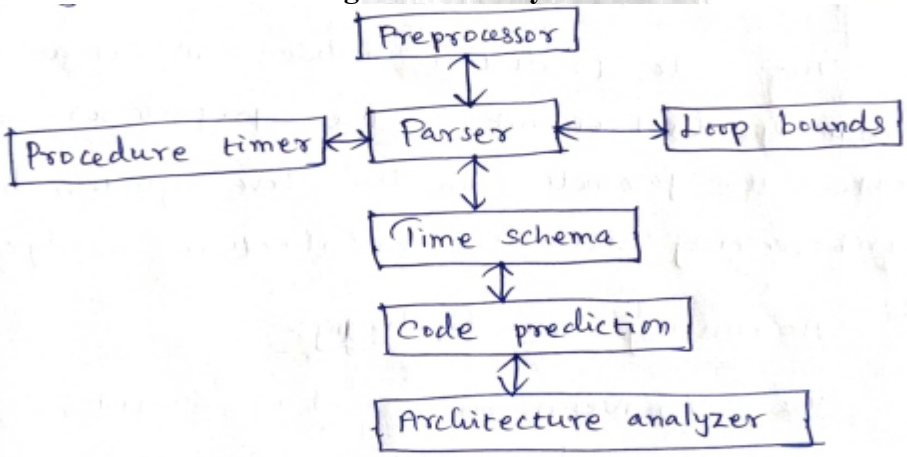


Fig: Schematic of a timing estimation system

19. List the uni-processor scheduling algorithms.

- Traditional rate-monotonic (RM)
- Rate-monotonic deferred server (DS)
- Earliest Deadline First (EDF)
- Precedence and exclusion conditions
- multiple task versions
- IRIS (Increased Reward with Increased Service) tasks.

20. List the multi-processor scheduling algorithms.

- Utilization balancing algorithm
- Next-fit algorithm
- Bin-packing algorithm
- Myopic offline scheduling algorithm
- Focussed addressing and bidding algorithm
- Buddy strategy
- Assignment with precedence constraints.

21. State the differences between rate monotonic and Deadline monotonic scheduling.

Rate Monotonic Scheduling	Deadline Monotonic Scheduling
1) It is a Static or fixed priority based scheduling policy. 2) The priority to the process is assigned based on the periods of the process. 3) Process with short period gets the highest priority.	1) It is a Static or fixed priority based scheduling policy. 2) The priority to the process is assigned based on the deadlines of the process. 3) Process with short deadline gets the highest priority.

22. Define schedulable utilization of a scheduling algorithm.

The schedulable utilization of a scheduling algorithm is defined as follows: A scheduling algorithm can feasibly schedule any set of periodic tasks on a processor if the total utilization of the tasks is equal to or less than the schedulable utilization of the algorithm.

23. Write the necessary and sufficient condition for a set of periodic tasks in RMS.

In RMS, the necessary condition for a set of periodic real-time tasks is:

$$\sum_{i=1}^{n} \frac{e_i}{p_i} = \sum_{i=1}^{n} u_i \leq 1$$

Where e_i is the worst case execution time and p_i is the period of the task T_i, n is the number of tasks to be scheduled and u_i is the CPU utilization due to the task T_i.

The sufficient condition for a set of n real-time periodic tasks under RMA is,

$$\sum_{i=1}^{n} u_i \leq n(2^{1/n} - 1),$$

where u_i is the utilization due to task T_i.

24. What are the shortcomings of EDF?

Shortcomings of EDF :-

1) Transient overload problem
2) Resource sharing problem
3) Efficient implementation problem

25. What are the advantages of EDF over RMS?

Advantages of EDF over Rate monotonic :-

1) No need to define priorities offline
2) Less context switching than RMS.
3) 100 % processor utilization factor.

26. What are IRIS tasks?

→ Increased Reward with Increased Service (IRIS)

- In this, tasks are not necessary to run to completion.
- These are iterative algorithms.
- The reward function associated with an IRIS task increases with the amount of service given to it.

Reward function $R(x)$:

$$\begin{cases} 0 & \text{if } x < m \\ r(x) & \text{if } m \leq x \leq o+m \\ r(o+m) & \text{if } x > o+m \end{cases}$$

where $r(x)$ is monotonically nondecreasing in x, and m, o are execution time of the mandatory and optional parts respectively.

27. What is utilization balancing algorithm?

This algorithm maintains the tasks in a queue in increasing order of their utilizations. It removes tasks one by one from queue and allocates them to the least utilized processor each time.

28. What is next-fit algorithm for RMS?

In this algorithm, a task set is partitioned so that each partition is scheduled on a uniprocessor using RMA scheduling.

→ This algorithm attempts to use as few processors as possible.

→ Each task has constant period and deadline constraints.

→ Each task is independent and no precedence constraints.

29. What is myopic offline scheduling algorithm?

→ Myopic offline scheduling (MOS) is an assignment / scheduling algorithm for non-preemptive tasks.

→ It is an offline algorithm – given in advance arrival times, execution times and deadlines.

30. 30.

30. What is bin-packing assignment algorithm for EDF?

This algorithm attempts to allocate tasks to a multiprocessor (identical processors) using EDF such that, ~~tasks assigned to a processor~~ the sum of utilizations of the tasks assigned to a processor does not exceed 1, and minimize the number of processors needed.

Several bin packing algorithms exist.

- First fit random algorithm and first fit decreasing algorithm.

31. What is Focussed Addressing and Bidding(FAB) algorithm?

→ Online dynamic real time task allocation algorithm.

→ It is used for task sets consisting of both critical and non-critical real-time tasks.

→ Local scheduler handles critical tasks arriving at a given node; global scheduler schedules noncritical tasks.

32. What is buddy strategy?

→ Dynamic real time task allocation strategy.

→ The buddy algorithm tries to overcome the high communication overhead of the FAB. It is very similar to FAB algorithm, but differs in the manner in which the target processors are found.

→ In this algorithm, a processor may be in any of the two states: underloaded and overloaded. The status of a node is underloaded if its utilization is less than some threshold value.

(1) A processor P_i is said to be underloaded if $U_i < Th$.

(2) A processor is said to be overloaded if its utilization is greater than the threshold value (i.e $U_i \geq Th$).

33. What are fault-tolerant systems?

The failure rates of real-time computers must be extraordinarily small. Indeed, they must be smaller than the failure rates of the components from which they are built. Such computers must therefore be *fault-tolerant*, that is, be able to continue operating despite the failure of a limited subset of their hardware or software.

34. What is meant by hardware and software fault?

A *hardware fault* 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 at some logic value (0 or 1) are hardware faults.

A *software fault* is a "bug" that can cause the program to fail for a given set of inputs.

35. Discuss about fault latency and error latency.

The *fault latency* is the duration between the onset of a fault and its manifestation as an error. This duration can be considerable. Since the faults themselves are invisible to the outside world, only showing themselves when they cause errors, such latency can, as we shall see, impact the reliability of the overall system.

The *error latency* is the duration between when an error is produced and when it is either recognized as an error or causes the failure of the system. Like fault latencies, error latencies can have a considerable impact on the overall system reliability.

36. What is forward and backward error recovery?

Error recovery is the process by which the system attempts to recover from the effects of an error. There are two forms of error recovery; forward and backward. In *forward error recovery*, the error is masked without any computations having to be redone. In *backward error recovery*, the system is rolled back to a moment in time before the error is believed to have occurred (this roll back involves the restoration of system state to the state at that moment), and the computation is carried out again. Backward error recovery uses time redundancy, since it consumes additional time to mask the effects of failure.

37. What causes failures?

There are three causes of failure: errors in the specification or design, defects in the components, and environmental effects.

38. Categorize the fault types.

Faults are classified according to their temporal behavior and output behavior. A fault is said to be *active* when it is physically capable of generating errors and to be *benign* when it is not.

1 Temporal Behavior Classification

There are three fault types: permanent, intermittent, and transient. A *permanent* fault does not die away with time, but remains until it is repaired or the affected unit is replaced. An *intermittent* fault cycles between the fault-active and fault-benign states. A *transient* fault dies away after some time.

2 Output Behavior Classification

A fault is also characterized by the nature of the errors that it generates. There are two categories of output behavior, nonmalicious and malicious.

39. Discuss about malicious or byzantine failures.

Failures corresponding to an inconsistent output are much harder to neutralize than the nonmalicious type. Indeed, we may think of them—somewhat whimsically—as the output of some malicious intelligence that has captured the device and is putting forth errors in such a way as to cause maximum disruption to the functioning of the system. Such failures are known as *malicious* or *Byzantine* failures. In short, a malicious fault is one that is assumed to behave arbitrarily.

40. What are the different ways to determine that a processor is malfunctioning?

There are two ways to determine that a processor is malfunctioning; online and offline.

Online detection goes on in parallel with normal system operation. One way of doing this is to check for any behavior that is inconsistent with correct operation. The following actions are indicative of a faulty processor.

Offline detection consists of running diagnostic tests. When a processor is running such a test, it obviously cannot be executing the applications software. Diagnostic tests can be scheduled just like ordinary tasks. The greater the failure rate, the greater must be the frequency with which these tests are run.

41. What are the types of redundancy?

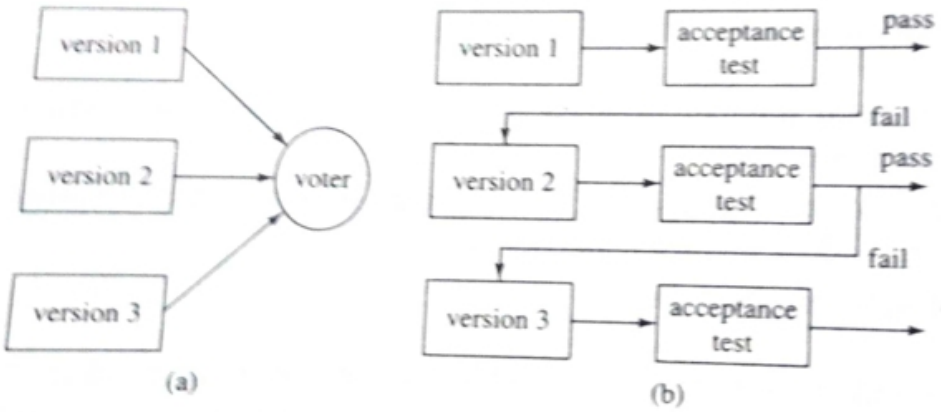
Hardware redundancy: The system is provided with far more hardware than it would need if all the components were perfectly reliable, typically, between two and three times as much.

Software redundancy: The system is provided with different software versions of tasks, preferably written independently by different teams of programmers, so that when one version of a task fails under certain inputs, another version can be used.

Time redundancy: The task schedule has some slack in it, so that some tasks can be rerun if necessary and still meet critical deadines.

Information redundancy: The data are coded in such a way that a certain number of bit errors can be detected and/or corrected.

42. What are the different software fault tolerant structures?



Software fault-tolerant structures: (a) *N*-version programming; (b) recovery-block approach.

43. What are the different ways to structure hardware redundancy?

1. Static pairing, 2. N-Modular Redundancy(NMR)

44. Describe the use of information redundancy.

USE OF INFORMATION REDUNDANCY. The most frequent use of coding is to detect and/or correct errors in information transmission and memory. Error-correcting codes used in memory are used during memory *scrubbing*; periodically, each memory location is accessed and checked for integrity. If an error is detected, the system corrects it using the error-correcting property of the code, and writes it back. This prevents the accumulation of errors due to transient faults.

45. What is the use of data diversity approach?

Data diversity is an approach that can be used in association with any of the redundancy techniques considered above. The idea behind it is as follows. Sometimes, hardware or software may fail for certain inputs, but not for other inputs that are very close to them. So, instead of applying the same input data to the redundant processors, we apply slightly different input data to them. Thus we have in some cases another line of defense against failure. This approach will only work if the sensitivity of the output is either very small with respect to small changes in the input or if perturbing the output can be corrected analytically.

46. How can we model the reliability of a system?

In order to model the reliability of a system, we must express the reliability of each of its components, and take into account the impact of the failure of each component on the functioning of the overall system.

47. State the purpose of software error models.

Software-error models predict the rate at which software faults will produce errors and are meant to determine when to stop debugging, by providing some guidance on the reliability of the software at each stage of debugging.

48. Give some example software error models.

1. Jelinski-Moranda model
2. Goel-Okumoto model
3. Littlewood model

49. What is clock synchronization?

Two clocks are said to be synchronized if the times they tell are sufficiently close. More precisely, clocks c_i and c_j are synchronized at c-time T if for some given $\delta > 0$,

$$|c_i(T) - c_j(T)| < \delta$$

$|c_i(T) - c_j(T)|$ is the *skew* between clocks c_i and c_j at c-time T.

An alternative definition of synchronization is to define the clock skew at real time t as $|C_i(t) - C_j(t)|$, and to say that clocks c_i and c_j are synchronized at real time t if

$$|C_i(t) - C_j(t)| < \delta$$

$|C_i(t) - C_j(t)|$ is the clock skew at real time t. The two conditions are nearly equivalent.

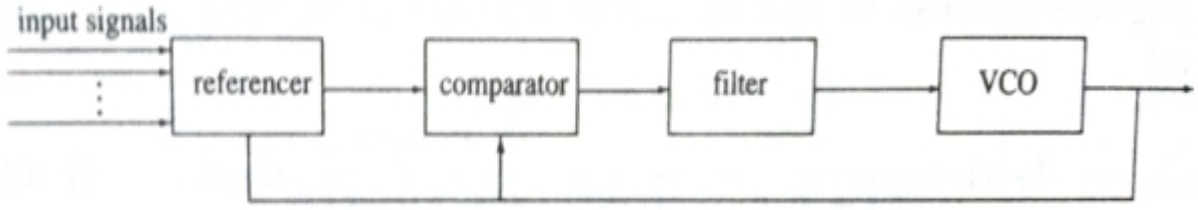
50. List the ways in which synchrony can be lost.

Synchronization is carried out by clocks exchanging timing messages and adjusting themselves appropriately. There are two ways in which synchrony can be lost as a result of some clocks becoming faulty, when multiple nonoverlapping cliques are formed and when the clocks are driven too fast or too slow.

51. How the fault-tolerant synchronization in hardware takes place?

To synchronize in hardware, we can use phase-locked loops. The objective is to align, as closely as possible, the output of the oscillator with an oscillatory signal input. The comparator puts out a signal that is proportional to the difference between the phase of the input and that of the oscillator. This is passed through a filter, and the resultant signal is used to modify the frequency of a voltage-controlled oscillator (VCO).

52. Draw the structure of a phase-locked loop used in synchronization?



53. What are the advantages and disadvantages of hardware synchronization?

The chief advantage of hardware synchronization is the very small clock skews that can be attained. Typically, these skews are on the order of nanoseconds. This is due to the frequency with which the resynchronizations are carried out.

Another advantage is that hardware synchronization places no burden on the rest of the system. Unlike software synchronization, it does not consume precious processor resources. Since the clocks are interconnected by dedicated links, the traffic in time signals does not take up any bandwidth on the interprocessor network. It is transparent to both the applications software and the operating system.

The chief disadvantage is the cost of the hardware. For applications where very tight synchronization is not needed, this approach may be too expensive, and software synchronization may be the preferred approach.

54. What do you know about synchronization in software?

When the extremely tight synchronization that is provided by phase-locking is not needed, synchronization can be carried out in software. In software-based synchronization, we have an underlying hardware clock, and a software-based correction. The clock time is the sum of the hardware time and the correction.

PART-B

1. Summarize the following:
   (i) Loss of synchrony.
   (ii) Principle of clocks in basic level.
2. Explain the structure of a real time system.
3. Explain in detail about the factors in estimating program run times.
4. Discuss the preemptive earliest deadline first algorithm.
5. Evaluate utilization bound for the RM algorithm and explain it in detail.
6. Explain in detail the various multiprocessor scheduling algorithms.
7. Explain on how redundancy plays an important role in fault tolerance of systems.
8. Construct the reliability models for hardware redundancy.
9. Explain in detail about the fault tolerant synchronization in hardware.

UNIT V PROCESSES AND OPERATING SYSTEMS

PART-A

1. Define a process.

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.

2. What is RTOS?

Real-time operating systems (RTOSs) are OSs that provide facilities for satisfying real-time requirements. A RTOS allocates resources using algorithms that take real time into account.

3. What are threads?

Processes that share the same address space are often called ***threads***.

4. What are multi-rate systems?

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

5. Define Multitasking.

The capability of the processor to execute multiple tasks simultaneously. Multitasking, in an operating system, is allowing a user to perform more than one computer task at a time.

6. Define multi-processing systems.

System in which there is more than one CPU is called as multiprocessing system. The simultaneous processing of a number of different processes by different processors at the same time is termed as multiprocessing.

7. What are the time requirements on process?

- Initiation time- The initiation time (also known as release time) is the time at which the process goes from the waiting to the ready state.
- Deadline- A deadline specifies when a computation must be finished.

8. How is RTOS uniquely different than a general purpose OS?

A RTOS allocates resources using algorithms that take real time into account. General-purpose OSs, in contrast, generally allocate resources using other criteria like fairness. Trying to allocate the CPU equally to all processes without regard to time can easily cause processes to miss their deadlines.

9. What is deadline for a process?

A deadline specifies when a computation must be finished. The deadline for an aperiodic process is generally measured from the release time, since that is the only reasonable time reference. The deadline for a periodic process may in general occur at some time other than the end of the period.

10. What happens when a process misses a deadline?

The practical effects of a timing violation depend on the application—the results can be catastrophic in an automotive control system, whereas a missed deadline in a multimedia

system may cause an audio or video glitch. Even if the modules are functionally correct, their timing improper behavior can introduce major execution errors.

11. What is the basic measure of CPU usage?

The simplest and most direct measure is utilization:

$$U = \frac{\text{CPU time for useful work}}{\text{total available CPU time}}.$$

Utilization is the ratio of the CPU time that is being used for useful computations to the total available CPU time. This ratio ranges between 0 and 1, with 1 meaning that all of the available CPU time is being used for system purposes.

12. State the processor states and scheduling.

The first job of the OS is to determine that process runs next. The work of choosing the order of running processes is known as scheduling.

The OS considers a process to be in one of three basic scheduling states: waiting, ready, or executing. There is at most one process executing on the CPU at any time.

13. What is the function of preemptive RTOS?

A preemptive RTOS executes processes based upon timing constraints provided by the system designer. The most reliable way to meet timing constraints accurately is to build a preemptive OS and to use priorities to control what process runs at any given time. Preemption is an alternative to the C function call as a way to control execution. Priority based scheduling is a way for the programmer to control the order in which processes run.

14. Define kernel.

The kernel is the part of OS that determines processing time, activated by timer. The length of the timer is called time quantum.

15. What is context switching?

The set of registers that define a process are known as its context and switching from one process's register set to another is known as context switching.

16. What is PCB or record?

The data structure that holds the state of the process is known as the process control block(PCB) or record.

17. What is round-robin scheduling?

A common scheduling algorithm in general purpose OS is round-robin. All the processes are kept on a list and scheduled one after the other. This scheduling provides a form of fairness in that all processes get a chance to execute. It does not guarantee the compilation time of any task; as the number of processes increases, the response time of all the processes increases.

18. List the advantages and limitations of Priority based process scheduling.

Advantages of Priority based scheduling can be given as

i) Since process priorities are taken into consideration the execution slot is given to the process with the highest priority.

ii) Efficiency of the scheduling is greatly increased as dead slots or idle slots are removed.

Limitations of Priority Based Scheduling

It is done as two types as static and dynamic. The static scheduling requires less hardware resources but the CPU utilization is less but dynamic scheduling uses more hardware resources but the CPU utilization is better than the other.

19. What is Rate-Monotonic Scheduling(RMS)?

Rate-monotonic scheduling (RMS) was one of the first scheduling policies developed for real-time systems and is still very widely used. We say that RMS is a static scheduling policy because it assigns fixed priorities to processes.

The theory underlying RMS is known as rate-monotonic analysis (RMA). This theory, as summarized below, uses a relatively simple model of the system.

- 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. This fixed-priority scheduling policy is the optimum assignment of static priorities to processes, in that it provides the highest CPU utilization while ensuring that all processes meet their deadlines.

20. Define earliest deadline first (EDF) scheduling?

Earliest deadline first (EDF) is another well-known scheduling policy. It 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. It assigns priorities in order of deadline. The highest-priority process is the one whose deadline is nearest in time, and the lowest priority process is the one whose deadline is farthest away.

21. Differentiate RMS with EDF.

EDF can extract higher utilization out of the CPU, but it may be difficult to diagnose the possibility of an imminent overload. Because the scheduler does take some overhead to make scheduling decisions.

RMS achieves lower CPU utilization but is easier to ensure that all deadlines will be satisfied. The implementation of EDF is more complex than the RMS code.

***22.* What is race condition?**

If combinations of events from the two tasks operate on the device in the wrong order, there is a critical timing race or race condition that causes erroneous operation.

23. What is critical section?

A set of instructions that must be atomic for the system to work properly is often called a critical section. To prevent race condition, we need to control the order in which some operations occur. We do so by enclosing sensitive sections of code in a critical section that executes without interruption.

24. Define semaphore.

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

25. What is priority inversion?

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

26. What is priority inheritance?

The most common method for dealing with priority inversion is priority inheritance: promote the priority of any process when it requests a resource from the operating system. Once the process is finished with the resource, its priority is demoted to its normal value.

27. What is blocking and non-blocking communication?

In general, a process can send a communication in one of two ways: blocking or nonblocking. After sending a blocking communication, the process goes into the waiting state until it receives a response. Nonblocking communication allows the process to continue execution after sending the communication. Both types of communication are useful.

28. Define Inter process communication. What are the major styles of inter process communication?

Interprocess communication mechanisms are provided by the operating system as part of the process abstraction. Processes often need to communicate with each other. An output from one task passed to another task through the scheduler and use of signals, exception, semaphore, queues, mailbox, pipes, sockets, and RPC.

There are two major styles of inter process communication: shared memory and message passing.

29. Define queue.

A queue is a common form of message passing. The queue uses a FIFO discipline and holds records that represent messages.

30. Define signal and mailbox?

Another form of inter process communication commonly used in Unix is the ***signal***. A signal is simple because it does not pass data beyond the existence of the signal itself. 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 operating system.

The mailbox is the simple mechanism for asynchronous communication. Some architectures define mailbox registers. These mailboxes have a fixed number of bits and can be used for small messages.

31. Define interrupt latency.

Interrupt latency for an RTOS is the duration of time from the assertion of a device interrupt to the completion of the device's requested operation.

32. What are the factors that affect interrupt latency?

- the processor interrupt latency
- The execution time of the interrupt handler
- Delays due to RTOS scheduling

33. What are the power optimization strategies for processes?

The RTOS and system architecture can use static and dynamic power management mechanisms to help manage the system's power consumption. A power management policy is a strategy for determining when to perform certain power management operations. A more sophisticated technique is 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. Several predictive techniques are possible: L-shaped distribution. The Advanced Configuration and Power Interface (ACPI) is an open industry standard for power management services.

34. Give any two examples for RTOS.

POSIX and Windows CE

35. What is POSIX?

POSIX is a version of the Unix operating system created by a standards organization. POSIX-compliant operating systems are source-code compatible-an application can be compiled and run without modification on a new POSIX platform assuming that the application uses only POSIX-standard functions. It is a Linux based OS. It has dual kernel based OS Core. It uses concepts like the Semaphores, Message Queue etc.

36. What are the features of Windows CE?

- Windows CE supports devices such as smartphones, electronic instruments, etc.
- Windows CE is designed to run on multiple hardware platforms and instruction set architectures.
- Windows CE provides support for virtual memory with a flat 32-bit virtual address space.
- Windows CE supports priority inheritance and two kernel-level units of execution.

37. What is multiprocessor?

A multiprocessor is, any computing system with two or more processors coupled together. Multiprocessors used for scientific or business applications tend to have regular architectures: several identical processors that can access a uniform memory space.

38. What are distributed embedded systems?

In distributed system, the processing elements are physically separated. The networks for distributed systems give higher latencies than are possible on a single chip. In distributed systems, the network is fairly lightweight. Message passing is widely used in distributed embedded systems. The most notable example of a safety-critical real-time distributed embedded system is found in the automobile. These systems can be modeled as message-passing systems. Example distributed systems are: CAN bus, I^2C, Ethernet and Internet Protocol.

39. List the OSI layers from lowest to highest level of abstraction.

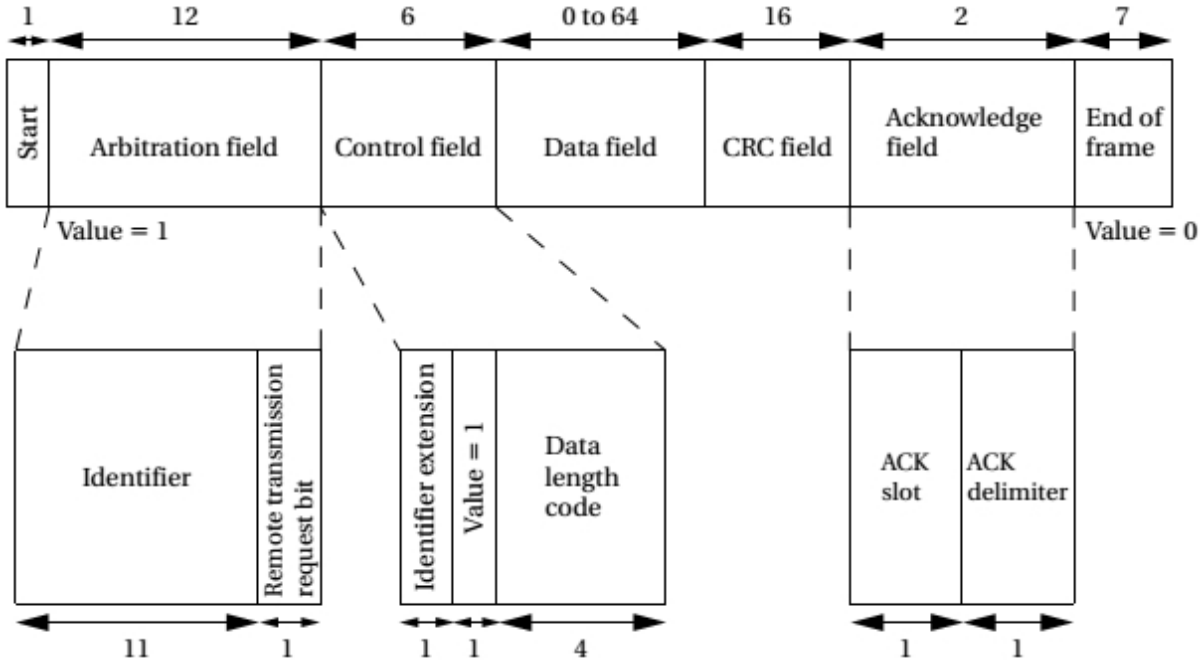
The OSI model includes seven levels of abstraction:

Application	End-use interface
Presentation	Data format
Session	Application dialog control
Transport	Connections
Network	End-to-end service
Data link	Reliable data transport
Physical	Mechanical, electrical

40. State the features of CAN bus.

i. The CAN bus uses bit-serial transmission.
ii. CAN runs at rates of 1Mbits/s with 11-bit addressing over a twisted pair connection of 40m.
iii. The bus protocol supports multiple masters on the bus.
iv. In CAN terminology, a logical 1 on the bus is called recessive and a logical 0 is dominant.
v. Data are sent on the network in packets known as data frames.
vi. CAN is a synchronous bus—all transmitters must send at the same time for bus arbitration to work.
vii. Control of the CAN bus is arbitrated using a technique known as Carrier Sense Multiple Access with Arbitration on Message Priority (CSMA/AMP).

41. Draw the data frame format of CAN?



42. What are the features of I^2C bus?

i. The I^2C bus(Inter Integrated Circuit) is a well-known bus commonly used to link microcontrollers into systems.
ii. I^2C is designed to be low cost, easy to implement, and of moderate speed (up to 100 KB/s for the standard bus and up to 400 KB/s for the extended bus).
iii. it uses only two lines: the serial data line (SDL) for data and the serial clock line (SCL),which indicates when valid data are on the data line.
iv. The I^2C bus is designed as a multi master bus—any one of several different devices may act as the master at various times.

43. What is the bus transaction procedure of I^2C bus?

A bus transaction is initiated by a start signal and completed with an end signal as follows:

- A start is signaled by leaving the SCL high and sending a 1 to 0 transition on SDL.
- A stop is signaled by setting the SCL high and sending a 0 to 1 transition on SDL.

44. Draw the format of I^2C bus transaction.

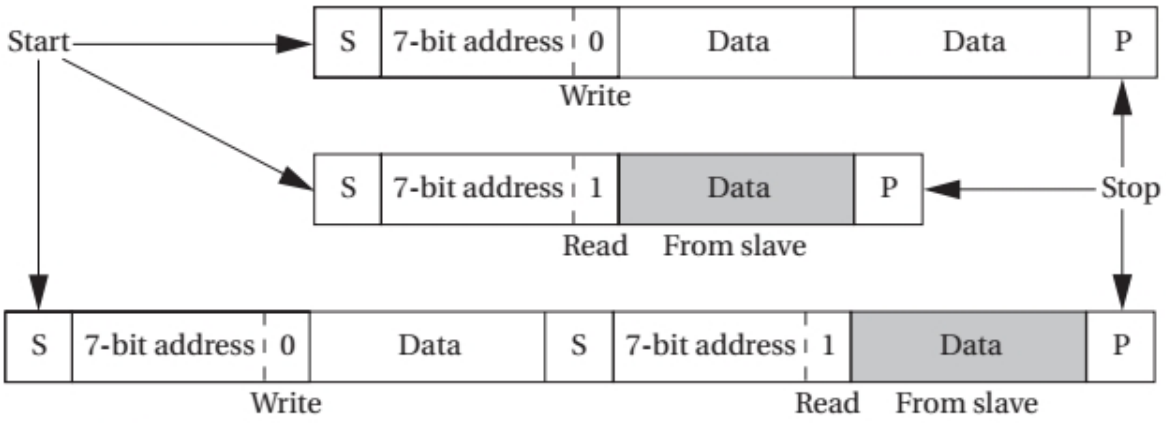


FIGURE 8.13

Typical bus transactions on the I²C bus.

45. What is Ethernet?

Ethernet is very widely used as a local area network for general-purpose computing. The network is a bus with a single signal path.

46. What are the features of Ethernet?

i. Used as a network for embedded computing
ii. The Ethernet standard allows for several different implementations such as twisted pair and coaxial cable.
iii. Nodes on the Ethernet are not synchronized—they can send their bits at any time.
iv. The Ethernet arbitration scheme is known as Carrier Sense Multiple Access with Collision Detection (CSMA/CD).
v. Ethernet was not designed to support real-time operations; the exponential backoff scheme cannot guarantee delivery time of any data.

47. Draw the packet format of Ethernet.

Preamble	Start frame	Destination address	Source address	Length	Data	Padding	CRC

48. What is Internet Protocol (IP)?

The Internet Protocol (IP) is the fundamental protocol on the Internet. It provides connectionless, packet-based communication. It is an internetworking standard. IP works at the network layer.

49. What is a router?

A node that transmits data among different types of networks is known as a router.

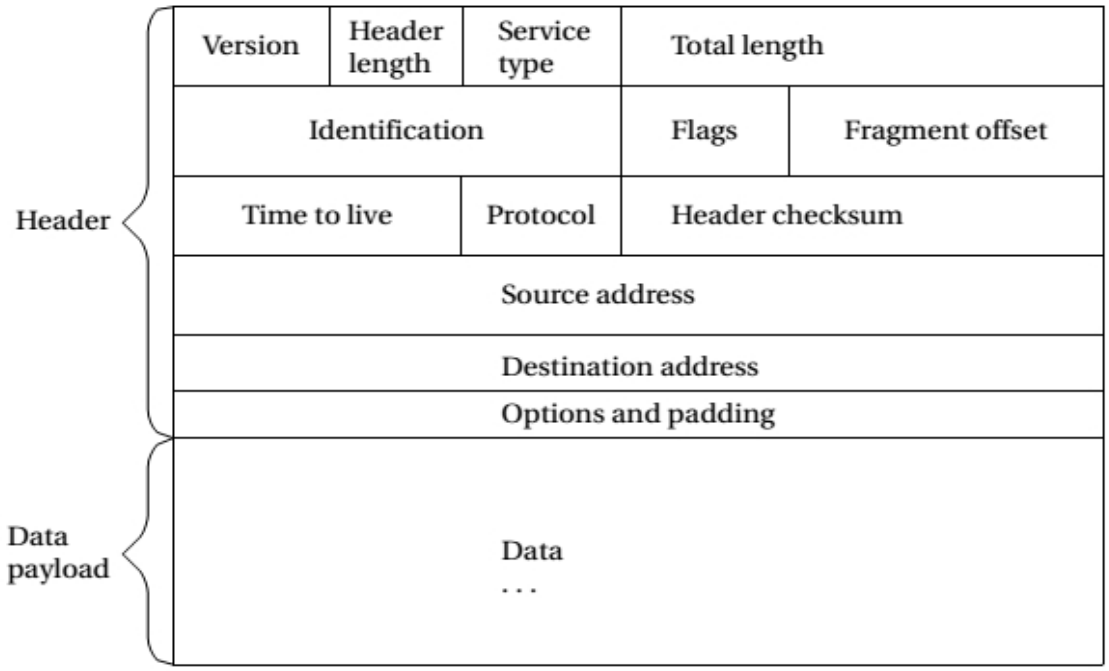
50. What are the features of IP?

i. An Internet address is a number (32 bits in early versions of IP, 128 bits in IPv6).
ii. The IP address is typically written in the form xxx.xx.xx.xx.
iii. IP works at the network layer tells us that it does not guarantee that a packet is delivered to its destination. Packets that do arrive may come out of order. This is referred to as best-effort routing.

51. What is the function of DNS?

The names by which users and applications typically refer to Internet nodes, such as foo.baz.com, are translated into IP addresses via calls to a Domain Name Server, one of the higher-level services built on top of IP.

52. Draw the IP packet structure.



53. What is TCP?

The Transmission Control Protocol (TCP) provides a connection oriented service that ensures that data arrive in the appropriate order, and it uses an acknowledgment protocol to ensure that packets arrive. Because many higher level services are built on top of TCP, the basic protocol is often referred to as TCP/IP.

54. What are the services provided by TCP using IP as the foundation?

Using IP as the foundation, TCP is used to provide File Transport Protocol (FTP) for batch file transfers, Hypertext Transport Protocol (HTTP) for World Wide Web service, Simple Mail Transfer Protocol (SMTP) for email, and Telnet for virtual terminals. A separate transport protocol, User Datagram Protocol (UDP), is used as the basis for the network management services provided by the Simple Network Management Protocol (SNMP).

55. What are the major types of multiprocessor architectures?

1. Shared memory: systems have a pool of processors that can read and write a collection of memories.

2. Message passing: systems have a pool of processors that can send messages to each other. Each processor has its own local memory.

56. What is MPSoC?

A multiprocessor system on chip (MPSoC) is a system-on-chip with multiple processing elements. The networks used for MPSoCs will be fast and provide lower-latency communication between the processing elements. Most MPSoCs are shared memory systems.

57. How does the distributed system differ from MPSoCs?

In a distributed system, the processing elements are physically separated. The networks for distributed systems give higher latencies than are possible on a single chip. In distributed systems, the network is fairly lightweight. Message passing is widely used in distributed embedded systems. The most notable example of a safety-critical real-time distributed embedded system is found in the automobile. These systems can be modeled as message-passing systems. Example distributed systems are: CAN bus, I^2C, Ethernet and Internet Protocol.

A multiprocessor system-on-chip (MPSoC) is a system-on-chip with multiple processing elements. In general, the networks used for MPSoCs will be fast and provide lower-latency communication between the processing elements. Most MPSoCs are shared memory systems. Shared memory processors are well-suited to applications that require a large amount of data to be processed. Signal processing systems stream data and can be well-suited to shared memory processing.

58. State the need for accelerators?

One important category of PE for embedded multiprocessor is the accelerator. An accelerator is attached to CPU buses to quickly execute certain key functions. Accelerators can provide large performance increases for applications with computational kernels that spend a great deal of time in a small section of code. Accelerators can also provide critical speedups for low-latency I/O functions.

59. What is hardware/software co-design?

Partitioning the system into hardware and software and the simultaneous design of hardware and software to meet system objectives is known as hardware/software co-design. The design of accelerated systems is one example of hardware/software co-design.

60. What are the accelerator performance analysis?

Speedup: The speedup factor depends in part on whether the system is *single threaded* or *multithreaded*, that is, whether the CPU sits idle while the accelerator runs in the single-threaded case or the CPU can do useful work in parallel with the accelerator in the multithreaded case. Another equivalent description is *blocking* vs. *nonblocking*. Does the CPU's scheduler block other operations and wait for the accelerator call to complete, or does the CPU allow some other process to run in parallel with the accelerator?

Execution time: A simple accelerator will read all its input data, perform the required computation, and then write all its results. In this case, the total execution time may be written as

$$t_{accel} = t_{in} + t_x + t_{out}$$

where t_x is the execution time of the accelerator assuming all data are available, and t_{in} and t_{out} are the times required for reading and writing the required variables, respectively.

The total speedup S for a kernel can be written as :

$$S = n(t_{CPU} - t_{accel})$$
$$= n[t_{CPU} - (t_{in} + t_x + t_{out})]$$

where t_{CPU} is the execution time of the equivalent function in software on the CPU and n is the number of times the function will be executed.

61. How will you deal with the scheduling and allocation design problems when designing a distributed embedded system?

i. We must schedule operations in time, including communication on the network and computations on the processing elements.

ii. We must allocate computations to the processing elements to determine what communications are required.

62. What are the basic functions of MP3 audio player?

Audio players are often called MP3 players after the popular audio data format. An MP3 player performs three basic functions: audio storage, audio decompression, and user interface.

63. Define the term MP3 audio compression.

The term MP3 comes from MPEG-1, layer 3; the MP3 spec is part of the MPEG-1 standard. That standard defined three layers of audio compression:

- 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.

64. State the requirements of audio player.

Name	Audio player
Purpose	Play audio from files
Inputs	Flash memory socket, on/off, play/stop, menu up/down
Outputs	Speaker
Functions	Display list of files in flash memory, select file to play, play file
Performance	Sufficient to play audio files at required rate
Manufacturing cost	Approximately $25
Power	1 AAA battery
Physical size and weight	Approximately 1 in x 2 in, less than 2 oz

65. What are the features of Cirrus CS7410 audio controller?

The Cirrus CS7410 is an audio controller designed for CD/MP3 players. 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 as a buffer 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, etc. Cirrus provides a reference design for a CD/MP3 player.

66. What are the requirements for the design of Engine Control Unit?

Name	ECU
Purpose	Engine controller for fuel-injected engine
Inputs	Throttle, RPM, intake air volume, intake manifold pressure
Outputs	Injector pulse width, spark advance angle
Functions	Compute injector pulse width and spark advance angle as a function of throttle, RPM, intake air volume, intake manifold pressure
Performance	Injector pulse updated at 2-ms period, spark advance angle updated at 1-ms period
Manufacturing cost	Approximately $50
Power	Powered by engine generator
Physical size and weight	Approx 4 in × 4 in, less than 1 pound.

67. State the function of Engine Control Unit (ECU)?

ECU controls the operation of a fuel-injected engine based on several measurements taken from the running engine.

68. What are the two major processes in the system architecture of ECU?

The two major processes, pulse-width and advance-angle, compute the control parameters for the spark plugs and injectors. The control parameters rely on changes in some of the input signals.

69. What is the purpose of video accelerator?

Video accelerator is an example of an accelerated embedded system. Digital video is still a computationally intensive task, so it is well suited to acceleration. Block motion estimation engines within a PC system are used in real-time search engines. Video Accelerator makes your videos stream faster and play smoother and Powerful.

70. State the operation of MPEG-2 video compression.

MPEG-2 forms the basis for U.S. HDTV broadcasting. This compression uses several component algorithms together in a feedback loop. The discrete cosine transform(DCT) used in JPEG also plays a key role in MPEG-2. As in still image compression, the DCT of a block of pixels is quantized for lossy compression and then subjected to lossless variable-length coding to further reduce the number of bits required to represent the block.

71. State the concept of block motion estimation?

The goal is to perform a two - dimensional correlation to find the best match between regions in the two frames. We divide the current frame into macroblocks (typically, 16 x 16). For every macroblock in the frame, we want to find the region in the previous frame that most closely matches the macroblock. Searching over the entire previous frame would be too expensive, so we usually limit the search to a given area, centered around the macroblock and larger than the macroblock. We try the macroblock at various offsets in the search area.

72. State the requirements of video accelerator?

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

73. Mention the classes describing basic data types in the video accelerator?

Some classes that describe basic data types in the system are: the motion vector, the macroblock, and the search area.

74. **What is motion vector?**

Vector from search area center to macroblock center is called motion vector.

75. List out the major components of audio player.

The major components of audio player are Interface (CD Drive), Digital Signal Processor, I/O,
Memory Controller, DAC etc.

76. What is macro block?

Macro block is a sub unit of image or sequence of frames. It consists of a fixed size of 16x16 samples. This block is further sub divided into prediction blocks which the motion estimation gives as output based on the estimated motion vector.

PART-B

1. Develop the Multirate system using condition of multi task and processes? Explain with suitable example and its necessary conditions.
2. Explain how multiple processes are handled by Preemptive real-time operating system.
3. Explain about Ethernet as a network for distributed embedded system.
4. With a suitable example(Internet) explain the operation of Internet enabled system.
5. Discuss about the various priority based scheduling algorithms.
6. Describe in detail about the Inter Process Communication mechanisms.
7. How to evaluate operating system performance? Explain?
8. Discuss in detail the power optimization strategies for processes.
9. Describe the real time operating system called POSIX in detail.
10. Explain the real time operating system called Windows CE in detail.
11. Discuss in detail about CAN used for distributed embedded computing.
12. Illustrate I^2C bus used as distributed embedded system with meaningful diagrams?
13. Explain the concepts of Multiprocessor System-On-Chip (MPSoC) and Shared memory multiprocessor are used in embedded applications. (OR) Explain about accelerated system design and accelerator performance analysis with suitable diagrams.
14. Formulate the working of Audio player in detail.
    i). Theory of operations and requirements,
    ii). Specification,
    iii). System Architecture,
    iv). Component designing and testing,
    v). System integration and debugging.
15. Formulate the working of Engine Control Unit(ECU) in detail.
    i). Theory of operations and requirements,
    ii). System Architecture,
    iii). Component designing and testing,
    iv). System integration and testing.
16. Write in detail about the embedded concepts in the design of video accelerator.