EC8791 – EMBEDDED AND REAL TIME SYSTEMS

COMPLETE 2-MARKS QUESTION BANK WITH ANSWERS

Anna University | B.E. ECE | 7th Semester | Regulations 2017 All 5 Units | 50 Questions | Curated from PYQ 2020–2024

★ = Frequently asked in exams | Answer in 3–5 lines for full 2 marks

UNIT I: INTRODUCTION TO EMBEDDED SYSTEM DESIGN

Complex Systems | Design Process | Model Train | QA | Consumer Electronics

Q1. What is an embedded system? Give two examples. ND2020

An embedded system is a computer system designed to perform a dedicated function within a larger mechanical or electrical system. It has fixed hardware and software optimized for specific tasks.

★ **Key: Dedicated function + real-time constraints**

Q2. Mention the challenges in embedded computing system design. ND2021

Key challenges: (1) Meeting real-time deadlines, (2) Limited memory and processing power, (3) Power/battery constraints, (4) Hardware-software co-design complexity, (5) Testing on target hardware is difficult, (6) Safety and reliability requirements.

★ **Key: At least 4 challenges for full marks**

Q3. Distinguish between microcontroller and microprocessor. ND2024

Microprocessor: CPU only, requires external RAM/ROM/peripherals, high power, general purpose (e.g., Intel Core). Microcontroller: CPU + RAM + ROM + I/O all on one chip, low power, used in embedded systems (e.g., ARM Cortex-M).

★ **Key: All-in-one vs. separate components**

Q4. Compare top-down and bottom-up design methodologies. ND2021

Top-Down: Start from system requirements, decompose into sub-modules. Bottom-Up: Start from basic components, combine to form system. Top-down ensures system-level correctness; bottom-up promotes component reuse.

★ **Key: Direction of design decomposition**

Q5. What is the need of microprocessor in embedded system? ND2020

Microprocessor provides computational intelligence to the embedded system — it executes the control algorithm, processes sensor data, makes decisions, and controls actuators. Without it, only hard-wired logic would be possible.

★ **Key: Intelligence + flexibility + control**

Q6. What is UML? Give its significance in embedded system design. ND2023

UML (Unified Modeling Language) is a graphical notation standard for visualizing, specifying, and documenting software systems. In embedded systems, it is used to model state machines, class diagrams, sequence diagrams, and use case diagrams before implementation.

★ **Key: Modeling tool for design visualization**

Q7. What is meant by requirement analysis? Why is it important? ND2022

Requirement analysis is the process of identifying what the system must do (functional requirements) and how it must behave (non-functional: speed, power, size). It is important to avoid design errors early, as fixing bugs later is 100x more expensive.

★ **Key: What the system must do**

Q8. Enumerate the services to be provided by consumer electronics. ND2023

Consumer electronics must provide: (1) User-friendly interface (display, buttons), (2) Multiple media formats (audio/video), (3) Connectivity (WiFi, Bluetooth), (4) Long battery life, (5) Low cost, (6) Safety and durability.

★ **Key: User-centric services**

Q9. Why is platform compatibility critical in consumer electronics? ND2024

Platform compatibility ensures software and peripherals work correctly across different hardware versions and operating systems. Incompatibility leads to product failures, costly redesigns, and poor user experience.

★ **Key: Interoperability across hardware**

Q10. What is Quality Assurance in embedded systems? Name two techniques. ND2021

QA is the process of ensuring an embedded system meets its requirements for correctness, reliability, and performance. Techniques: (1) Black-box testing — test based on I/O behavior, (2) White-box testing — test based on internal code structure.

★ **Key: Ensuring system meets specifications**

UNIT II: ARM PROCESSOR AND PERIPHERALS

ARM Architecture | Instruction Set | PWM | Timer | UART | Stacks | Subroutines

Q1. List the three different profiles of ARM Cortex Processor. ND2020

(1) Cortex-A (Application): For complex OS like Linux/Android, used in smartphones. (2) Cortex-R (Real-Time): For hard real-time applications, used in automotive systems. (3) Cortex-M (Microcontroller): Low power MCU applications, IoT devices.

★ **Key: A = Application, R = Real-time, M = Microcontroller**

Q2. Distinguish between single-edge and double-edge PWM. ND2020

Single-Edge PWM: Output rises at start of period, falls when match register value is reached. Simple control, one match register per channel. Double-Edge PWM: Output rises and falls within a period, controlled by two match values — allows finer duty cycle control. Used for motor control.

★ **Key: One edge vs. two edges per period**

Q3. How do you return from an ARM procedure? ND2021

Use the instruction: MOV PC, LR (copy Link Register back to Program Counter). During a subroutine call using BL, the return address is automatically saved in LR (R14). The instruction LDMFD SP!, {PC} also restores PC from the stack.

★ **Key: MOV PC, LR or LDMFD SP!, {PC}**

Q4. What is the difference between PCLK and CCLK in LPC214X? ND2023

CCLK (CPU Clock): The main processor clock frequency that drives the ARM7 CPU core. PCLK (Peripheral Clock): Derived from CCLK using the VPBDIV register. PCLK = CCLK / 1, 2, or 4. Peripherals (UART, PWM, Timer) use PCLK for their clock source.

★ **Key: CCLK = CPU, PCLK = Peripheral (CCLK/n)**

Q5. Depict the three address format for instructions in ARM processors. ND2022

ARM instructions use 3 addresses: Destination register (Rd), First source register (Rn), Second operand (Rm or immediate). Format: ADD Rd, Rn, Rm (e.g., ADD R0, R1, R2 means R0 = R1 + R2).

★ **Key: Rd = Rn op Rm (3-address RISC format)**

Q6. What is the concept and need for mode switching in ARM? ND2023

ARM has 7 processor modes (User, FIQ, IRQ, Supervisor, Abort, Undefined, System). Mode switching occurs when exceptions/interrupts happen or via MSR instruction. It is needed to protect system resources — privileged modes access all resources while User mode is restricted.

★ **Key: 7 modes + CPSR mode bits**

Q7. How are subroutines used in ARM Assembly Language? ND2024

Subroutines are called using BL (Branch with Link) instruction which saves return address in LR register. Return using MOV PC, LR. For nested calls, push LR onto stack: STMFD SP!, {LR}, and return with LDMFD SP!, {PC}.

★ **Key: BL = call, MOV PC,LR = return**

Q8. How do you generate a 1 kHz PWM signal with 50% duty cycle using LPC214X? ND2024

At PCLK=60MHz: PWMMR0 = PCLK/freq = 60,000,000/1000 = 60,000 (sets period). For 50% duty: PWMMR3 = PWMMR0/2 = 30,000. Set PWMPCR to enable PWM3 output. Set PWMLER = 0x09 to latch values.

★ **Key: PWMMR0=period, PWMMR3=duty cycle**

Q9. Give one difference between ARM Stack and Heap. ND2022

Stack: Fixed-size memory region, automatically managed (PUSH/POP), grows downward, used for local variables and return addresses. Heap: Dynamic memory region, manually managed (malloc/free), grows upward, used for dynamically allocated data.

★ **Key: Stack = automatic, Heap = dynamic**

Q10. Explain the operation of instruction: ADD r3, r2, r1, LSL#3 ND2022

LSL#3 shifts r1 left by 3 bits (equivalent to r1 × 8). Then r2 is added to the shifted value. Result: r3 = r2 + (r1 × 8). This is a barrel shifter operation combined with addition — unique to ARM's flexible second operand.

★ Key: r3 = r2 + r1*8 (barrel shifter)

UNIT III: EMBEDDED PROGRAMMING

Compilation | Linking | Loading | Performance | Optimization | Testing | DFG | CDFG

Q1. Differentiate compiler and cross-compiler. ND2020

Compiler: Compiles code for the SAME machine it runs on (e.g., gcc on x86 produces x86 code). Cross-Compiler: Compiles code on HOST machine for a DIFFERENT TARGET machine (e.g., arm-linux-gcc runs on x86 PC but produces ARM binary code for embedded target).

★ **Key: Same platform vs. different platform**

Q2. Mention the different components for embedded programs. ND2020

(1) Interrupt Service Routines (ISR), (2) Main program loop, (3) Device drivers, (4) Data structures (queues, buffers), (5) Configuration/initialization code, (6) Real-time tasks/threads.

★ **Key: 6 components**

Q3. What does a linker mean? What does it do? ND2021

A linker is a program that combines multiple object files (.o) and library files into a single executable. It performs: (1) Symbol resolution — matches function calls to definitions, (2) Relocation — assigns final memory addresses to all code and data sections.

★ **Key: Combines .o files → executable**

Q4. Bring out the difference between program counter and program location counter. ND2021

Program Counter (PC): Hardware register in CPU that holds the address of the NEXT instruction to execute. Program Location Counter (PLC): Assembler concept that tracks the current memory address being assigned during assembly — used to compute relative addresses.

★ **Key: PC = CPU register, PLC = assembler concept**

Q5. Draw the schematic of DFG and CDFG with an example. ND2023

DFG (Data Flow Graph): Nodes are operations (+,-,*), edges are data dependencies. Example: z=x+y → nodes [x],[y],[+],[z], edges x→+, y→+, +→z. CDFG (Control/Data Flow Graph): DFG + control nodes (branches, loops). Adds diamond-shaped decision nodes for if/else, loop back edges for while loops.

★ **Key: DFG = data only, CDFG = data + control**

Q6. Mention the features of program optimization in embedded systems. ND2023

Features: (1) Code size reduction (less flash usage), (2) Speed improvement (fewer CPU cycles), (3) Power reduction (fewer operations), (4) Loop optimization (unrolling, tiling), (5) Dead code elimination, (6) Register allocation optimization.

★ **Key: Size + Speed + Power = 3 key goals**

Q7. Bring out the difference between black-box and white-box testing. ND2024

Black-box: Tests based on I/O behavior only; tester does NOT see internal code; uses equivalence partitioning, boundary value analysis. White-box: Tests based on internal code structure; tester KNOWS the code; uses path coverage, cyclomatic complexity, def-use pairs.

★ **Key: External behavior vs. internal structure**

Q8. List the major software optimization techniques used in embedded systems. ND2024

(1) Loop unrolling, (2) Code motion (move invariant code out of loop), (3) Strength reduction (× → shift), (4) Inlining, (5) Dead code elimination, (6) Common subexpression elimination, (7) Register allocation.

★ **Key: 7 main techniques**

Q9. State the use of breakpoint and watchpoint support in ARM debugging. ND2022

Breakpoint: Pauses program execution when a specific INSTRUCTION ADDRESS is reached — allows inspection of register/memory state at that point. Watchpoint (Data breakpoint): Pauses execution when a specific MEMORY ADDRESS is read/written — useful for tracking variable corruption.

★ **Key: Break on instruction vs. on data access**

Q10. What is cyclomatic complexity? Give the formula. ND2021

Cyclomatic complexity V(G) measures the number of independent execution paths in a program — equals the minimum number of test cases needed for full branch coverage. Formula: V(G) = E – N + 2P, where E=edges, N=nodes, P=connected components. Also: V(G) = number of decision nodes + 1.

★ Key: V(G) = E – N + 2P

UNIT IV: REAL TIME SYSTEMS

RTS Structure | Scheduling | Fault Tolerance | WCET | Reliability | Clock Sync

Q1. Define Performance measures for real-time systems. ND2020

Key performance measures: (1) WCET (Worst Case Execution Time), (2) CPU Utilization U = Σ(Ci/Ti), (3) Response time = time from task release to completion, (4) Jitter = variation in response time, (5) Schedulability = can all deadlines be met?

★ **Key: WCET + Utilization + Response time**

Q2. Outline the definition for a schedule as a function. ND2020

A schedule is a function σ(t) that maps each time instant t to the task to be executed at that time. σ(t) = τi means task τi is running at time t. A valid schedule ensures every task completes before its deadline within its allocated time slots.

★ **Key: σ(t) → task assigned at time t**

Q3. Mention the limitation of Rate Monotonic (RM) algorithm. ND2021

Limitations: (1) Utilization bound decreases as tasks increase (→ ln2 = 0.693 for many tasks), (2) Cannot handle aperiodic tasks directly, (3) May reject schedulable task sets (pessimistic), (4) Does not work when deadline < period (D < T), (5) Priority inversion can occur.

★ **Key: Cannot guarantee full CPU utilization**

Q4. What is the important metric used for evaluating embedded system performance? ND2021

WCET (Worst Case Execution Time) is the most important metric. It is the maximum time any execution path of a task can take. It determines whether hard real-time deadlines can be guaranteed. Other metrics: CPU utilization, response time, jitter, throughput.

★ **Key: WCET is the most critical metric**

Q5. What is dynamic priority algorithm? State its advantages. ND2023

Dynamic priority algorithm assigns priorities to tasks at runtime based on current state (e.g., EDF assigns highest priority to task with earliest absolute deadline). Advantages: (1) Better CPU utilization (up to 100%), (2) Optimal for uniprocessor preemptive scheduling, (3) Handles aperiodic tasks more flexibly.

★ **Key: EDF is the main dynamic priority algorithm**

Q6. What are sporadic and aperiodic tasks? Give examples. ND2023

Aperiodic task: Arrives at unknown unpredictable times with no minimum inter-arrival constraint. Example: User keypresses, random sensor trigger. Sporadic task: Like aperiodic but with a MINIMUM inter-arrival time guarantee. Example: Emergency brake activation (minimum 50ms apart).

★ **Key: Aperiodic = unpredictable, Sporadic = min gap**

Q7. What is meant by priority inversion? ND2024, ND2020

Priority inversion occurs when a high-priority task is blocked waiting for a resource held by a low-priority task, and a medium-priority task preempts the low-priority task — effectively allowing the medium-priority task to run before the high-priority task.

★ **Key: High priority blocked by medium via low**

Q8. Compare hard real-time and soft real-time systems. ND2024

Hard RT: Missing deadline causes catastrophic failure (pacemaker, ABS brakes). Deadline MUST be guaranteed. Soft RT: Missing deadline causes performance degradation, not system failure (video player, stock quotes). Timing is desired but not mandatory.

★ **Key: Catastrophic vs. degraded on deadline miss**

Q9. What is priority inheritance? ND2020

Priority inheritance is a protocol to solve priority inversion: when a low-priority task holds a resource needed by a high-priority task, the low-priority task's priority is temporarily RAISED to match the high-priority task. This prevents medium-priority tasks from preempting the resource holder.

★ **Key: Raise L to H's priority while holding resource**

Q10. State how control of aborting illegal memory accesses helps in fault tolerance. ND2022

If a task accesses an invalid memory address (null pointer, out-of-bounds), the MMU detects the violation and generates an abort exception. The OS can then terminate only the faulty task, log the error, and restart it — preventing one buggy task from corrupting the entire system.

★ **Key: MMU abort → isolate fault → restart task**

UNIT V: PROCESSES AND OPERATING SYSTEMS

RTOS | IPC | Priority Scheduling | MPSoC | POSIX | Windows CE | Distributed Systems

Q1. List the advantages and limitations of priority-based process scheduling. ND2020

Advantages: (1) Simple to implement, (2) High-priority tasks always get CPU first, (3) Good for real-time systems. Limitations: (1) Starvation of low-priority tasks, (2) Priority inversion risk, (3) Requires careful priority assignment, (4) Dynamic workloads may need priority adjustment.

★ **Key: Advantages + Starvation limitation**

Q2. What is significance of Shared Memory Multiprocessors? ND2023

Shared memory multiprocessors allow multiple processors to access a common memory space, enabling fast inter-processor communication without message passing overhead. They support parallel task execution, reduce latency, and simplify programming. Key issue: cache coherence must be maintained.

★ **Key: Fast communication via common memory**

Q3. Enumerate the major functions of POSIX RTOS. ND2023

(1) Task/thread management (create, delete, schedule), (2) Priority-based preemptive scheduling, (3) IPC mechanisms (semaphores, message queues, shared memory), (4) Real-time clock and timer support, (5) Interrupt handling with bounded latency, (6) Memory management.

★ **Key: 6 major functions**

Q4. Give one challenge in developing code for MPSoCs. ND2022

Task partitioning challenge: Deciding which tasks run on which processor core to maximize parallelism while minimizing inter-core communication overhead and ensuring load balancing is very complex. Also: Cache coherence across cores is difficult to manage.

★ **Key: Task partitioning + cache coherence**

Q5. How does the ARM SAP instruction provide atomic execution? ND2022

SWP (Single Word Swap) instruction performs an atomic read-modify-write operation: it reads a memory location into a register AND writes a new value to that location in ONE uninterruptible bus cycle. This ensures mutual exclusion in multi-processor systems without a separate lock/unlock sequence.

★ **Key: Atomic = read + write in one bus cycle**

Q6. State the role of multi-rate systems. ND2024

Multi-rate systems execute different tasks at different sampling rates. For example: fast sensor reading (1ms), control computation (10ms), display update (100ms). This optimizes CPU usage — critical tasks run fast while non-critical tasks run slowly, saving CPU time.

★ **Key: Different tasks at different frequencies**

Q7. List the various power optimization strategies for processes. ND2024

(1) Dynamic Power Management (DPM) — power off idle units, (2) DVFS (Dynamic Voltage Frequency Scaling), (3) Task consolidation — batch tasks to maximize sleep, (4) Slack stealing — use spare time wisely, (5) Clock gating, (6) Memory power gating.

★ **Key: 6 strategies**

Q8. What is meant by requirement analysis if doing memory scaling for a video accelerator? ND2022

Memory scaling requirement analysis identifies: (1) Frame resolution (1080p, 4K), (2) Frame rate (30/60fps), (3) Color depth (24-bit RGB), (4) Buffer requirements (double/triple buffering), (5) Memory bandwidth needed = resolution × fps × bytes_per_pixel.

★ **Key: Memory bandwidth = resolution × fps × bpp**

Q9. List two special functional units of an embedded processor for audio player design. ND2022

(1) DSP unit (Digital Signal Processor): For audio decoding (MP3/AAC decompression), filtering, equalization — required for real-time audio processing. (2) Audio DAC controller: Converts decoded PCM digital samples to analog audio output signal.

★ **Key: DSP + Audio DAC**

Q10. Compare the major functionalities of POSIX RTOS and Windows CE. **ND2020, ND2021**

POSIX RTOS: Open standard, microsecond timer precision, strict real-time guarantees, used in safety-critical systems, supports processes + threads. Windows CE: Microsoft proprietary, millisecond precision, softer real-time, richer GUI support, used in consumer electronics like PDAs and set-top boxes.

★ **Key: Open + hard RT vs. Proprietary + rich GUI**

END OF EC8791 – 2 MARKS COMPLETE QUESTION BANK

Total: 50 Questions across 5 Units | Curated from PYQs: ND2020, ND2021, ND2022, ND2023, ND2024