|  |  |
| --- | --- |
| 1 | You are developing firmware for a flood monitoring system using an ARM7 processor. The system continuously monitors water levels, and when a flood is detected, it initiates appropriate responses. Assume the ARM7 processor has a classic 3-stage pipeline (Fetch, Decode, and Execute) and a branch delay slot.  ; ARM7 Assembly Code for Flood Detection  Loop:  LDR R0, [WaterLevelSensorAddress] ; Load water level data  CMP R0, #ThresholdWaterLevel ; Compare with threshold  BLE NormalWaterLevel ; Branch if less than or equal to threshold    ; Process flood condition  ; Here you can include code to initiate specific responses for flood detection  ...    B Loop ; Repeat the loop  NormalWaterLevel:  ; Additional processing for normal water level  B Loop ; Repeat the loop |
|  |  |
| 2 | For the ARM7 assembly code, describe the sequence of events in terms of the CPSR changes during execution. Explain how the CPSR reflects the status of the processor at different points in the code.  ; ARM7 Assembly Code for Arithmetic Operations  MOV R0, #5 ; Load the first number into register R0  MOV R1, #3 ; Load the second number into register R1  ; Addition  ADD R2, R0, R1 ; R2 = R0 + R1  ; Subtraction  SUB R3, R0, R1 ; R3 = R0 - R1  ; Multiplication  MUL R4, R0, R1 ; R4 = R0 \* R1  ; Division  MOV R5, #0 ; Clear R5 to avoid unpredictable behavior  CMP R1, #0 ; Check if the divisor (R1) is zero  BEQ DivisionByZero ; Branch to handle division by zero  SDIV R5, R0, R1 ; R5 = R0 / R1  DivisionByZero:  ; Handle division by zero, if necessary  ...  ; Continue with the rest of the program  ... |
| 3 | Imagine actively participating in the development of a real-time control system incorporating an ARM7 microcontroller. Investigate the distinctive characteristics embedded in the ARM7 architecture that render it particularly suitable for real-time applications. Present an in-depth analysis of your approach to harnessing hardware features such as **bus technology, memory, and peripherals** to guarantee accuracy in timing and control across the entire system. |
|  |  |
| 4 | While developing firmware for a data acquisition system based on the ARM7 architecture with a primary emphasis on real-time processing, describe your methodology for creating software that effectively captures, processes, and transmits real-time data. Discuss the importance of integrating **initialization code**, **operating systems, and applications** in your overall software strategy. |
|  |  |
| 5 | Envision the creation of a cutting-edge smart home system integrating diverse interconnected devices for automation and control. In this context, define what an embedded system entails and offer examples of **four distinct functions** that embedded systems fulfil within the framework of this advanced smart home technology. |
|  |  |
| 6 | Imagine playing a pivotal role in the advancement of a sophisticated gas leakage detection system aimed at improving early warning capabilities. In this scenario, clarify the ***concept of an actuator*** and highlight its importance in the design of embedded systems, specifically within the context of gas leakage detection system. |
|  |  |
| 7 | Imagine yourself in the role of the lead developer for a crucial real-time system implemented in a medical device that demands accurate timing and responsiveness. In this context, clarify the differences in ***memory management between a general-purpose kernel and a real-time kernel.*** Discuss the implications of these distinctions on the reliability and performance of the medical device, taking into account its stringent real-time requirements. |
|  |  |
| 8 | Imagine you are a software engineer assigned to develop a new process within a complex software system. Elaborate on the various activities involved in ***creating this process, considering the entire process life cycle.*** Address the key stages, methodologies, and best practices that should be followed to ensure a successful and efficient development process. Additionally, discuss how considerations like requirements analysis, design, implementation, testing, deployment, and maintenance play a role in shaping the overall life cycle of the process within the software system. |
|  |  |
| 9 | Examine and evaluate the ***diverse categories of memory*** crucial to the construction of embedded systems, particularly in the context of smart watch. Investigate the unique characteristics, benefits, and trade-offs associated with various memory types, including RAM, ROM, Flash, and supplementary storage mediums like SD cards. Analyze how the selection of each memory type contributes to the overall performance, user experience, and storage capabilities of smartwatch. Assess the implications of memory choices on factors such as application responsiveness, multitasking capabilities, and the seamless integration of features in contemporary smartphone designs. Provide specific examples and scenarios to illustrate the intricate role played by each type of memory in optimizing the functionality and efficiency of embedded systems within the smartphone domain. |