UNIT 1
INTRODUCTION TO EMDBEDDED COMPUTING

PART A

1.Define Embedded system

An embedded system employs a combination of hardware & software (a "computational engine") to perform a specific function; is part of a larger system that may not be a "computer"; works in a reactive and time-constrained environment

2. What are the main components of an embedded system?

The three main components of an embedded system are Hardware Main application software RTOS

3. What are the various classifications of embedded system?

Small scale embedded systems ,Medium scale embedded systems ,Sophisticated embedded systems

4. Mention the major challenges in embedded system design.

- Hardware needed
- Meeting the deadlines
- Minimizing the power consumption
- Design for upgradeability

5. What is the need of memory management methods?

- To provide storage for the software to run
- To store program variables and intermediate results
- To store information

6. Why does a program reside in ROM in embedded systems?

Final stage software is also called as ROM image .The final implement able software for a product embeds in the ROM as an image at a frame. Bytes at each address must be defined for creating the image..

7. What is the need for watchdog timer?

A watchdog timer resets the microprocessor and starts the software over from the beginning if the software does not restart it periodically. It is used to rescue the system if a fault develops and the program gets stuck.

8. What are embedded Real time systems? Give an example

In real time system the task in hand must be completed within a bounded time. Eg.ACVM,

9. What types of hardware parts are typically found in ES? Write in brief Power source

- Clock oscillator circuit
- Timers
- Memory units
- DAC & ADC
- LCD & LED displays
- Keyboards and keypads

10. What is the difference between Hard RealTime and Soft Real Time systems?

Hard Real Time system is a Real Time system where missing a deadline could cause drastic results that could lead to loss of life and / or property is called a hard real-time system. Examples are aircraft, biomedical instruments (like pacemakers), nuclear reactors etc. SoftReal Time system is a

realtime system where a few missed deadlines may not cause any significant [illegible] use. Examples are TV, multimedia etc.

11. What are the steps involved in the build process? The steps involved are

- Preprocessing
- Compiling
- Linking
- Locating

12. What are the typical characteristics of an embedded system?

Typical characteristics:

- Perform a single or tightly knit set of functions;
- Increasingly high-performance & real-time constrained;
- Power, cost and reliability are often important attributes
- That influence design;
- Application specific processor design can be a significant component of some embedded systems.

Other characteristics:

- Application specific
- Digital signal processing in ECS
- Reactive
- Real-time Distributed

13. What are the applications of an embedded system?

- Consumer electronics, e.g., cameras, camcorders, etc.,
- Consumer products, e.g., washers, microwave ovens, etc.,
- Automobiles (anti-lock braking, engine control, etc.,)
- Industrial process controllers & avionics/defense applications
- Computer/Communication products, e.g., printers, FAX machines, etc.,
- Emerging multimedia applications & consumer electronics

14. What are the complicating factors in embedded design?

- Complicating factors in the design of embedded systems
- Many of the subtasks in design are intertwined.
- Allocation depends on the partitioning, and scheduling presumes a certain allocation.
- Predicting the time for implementing the modules in hardware or software is not very
- easy, particularly for tasks that have not been performed before.

15. What are the functional requirements of embedded system?

Data Collection

- Sensor requirements
- Signal conditioning
- Alarm monitoring

Direct Digital Control

- Actuators
- Man-Machine Interaction
- informs the operator of the current state of the controlled object
- assists the operator in controlling the system.

16. What are the real-time requirements of an embedded system?

Hard-real time systems: where there is a high penalty for missing a deadline e.g., control systems for aircraft/space probes/nuclear reactors; refresh rates for video, or DRAM. Soft realtime systems: where there is a steadily increasing penalty if a deadline is missed. e.g., laser printer: rated by pages-per-minute, but can take differing times to print a page (depending on the \"complexity\" of the page) without harming the machine or the customer.

17. Define device driver.

A device driver is software for controlling, reading, sending a byte of stream of bytes from/to the device.

18. Give some examples for small scale embedded systems.

68HC05, PIC 16F8x, 8051, etc.,

19. Give some examples for medium scale embedded systems

8051, 80251, 80x86, 80196,68HC11xx

20. Give some examples for sophisticated embedded systems

ARM7, Power PC, Intel 80960, etc.,

PART B

For all the 16 mark questions include the following points

Introduction :What is Embedded systems, classifications of embedded system the major challenges in embedded system design

conclusion :What are the applications of an embedded system

1. Explain in detail about the build process for embedded systems. (16)

- Introduction about built process
- Steps to convert the source code representation of embedded software
- The embedded software development process flow chart
- Explanations for compiling linking and locating

2. Describe the structural units in embedded processor. (16)

- List out the various processing units
- Block diagram for structural units.
- Define each blocks such as Buses, MDR,MAR,BIU,PC,SP,caches, AOU
- Features
- Functions of different units
- Characteristics

3. How to select the processor based upon its architecture and applications. (16)

- List the consideration for selection of processor
- Performance consideration
- Power consideration
- Peripheral set
- Selection of memory devices (list the various characteristics)

4. Explain the concept of DMA. (16)

- Definition about DMA
- Operation
- Draw the figure of Bus with a DMA controller
- Functions of DMA

- DMA registers
- UML sequence diagram
- Justify CPU at the time of DMA transfer

5. **Discuss the methods in memory management. (16)**
   - Define memory allocation
   - Diagram of segmented memory allocation
   - Conditions for memory management
   - Fixed and dynamic block allocation techniques
   - Dynamic page allocations
   - Dynamic address relocation
6. **Illustarte the concept of watch dog timer. (16)**
   - Define watch dog timer
   - Functions
   - Examples
   - Applications
7. **Describe the working principle of incirucit emulator. (16)**
   - Define emulator
   - Functions of ICE
   - Drawback
   - Advantages
   - Diagram to connect target board
   - Emulation device
   - Emulation memory
   - Emulation control logic
   - Devi ce adaper
8. **Discuss in detail about target hardware debugging. (16)**
   - Introduction about hardware debugging
   - Magnifying glass
   - Multimeter
   - Digital CRO
9. Explain Model Train controller with neat diagram
   - Introduction about embedded system
   - Challenges in embedded system, computing
   - Requirements
   - Digital command control
   - Conceptual specification
   - Detailed specification
10. Explain ARM processor
    - Introduction about embedded system
    - Computer architecture Taxonomy
    - Von Neumann Architecture
    - Harvard architecture
    - Reduced instruction set computers
    - Assembly language
    - Instruction sets
11. Explain the major operations of CPU in Embedded system
    - Programming input and output

- Input and output devices
- Input and output prelimineries
- Busy wait i/o
- Interrupts

12. Explain supervisory modes
    - Excetution
    - Prioritization
    - Vectoring
    - Traps
13. Explain Complexs system and Microprocessor
    - Embedded computers
    - Charcteristics of Embedded Computing Applications3
    - Reason to select microprocessors for system design
    - Challenges in embedded system, computing
    - Performance in embedded computing