CBCS SCHEME

USN | | | | | | | | | | 18EC62

Sixth Semester B.E. Degree Examination, Feb./Mar. 2022
Embedded Systems

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. With a neat diagram, explain the architectural features of ARM Cortex-M3 processor. **(07 Marks)**

b. Explain RESET sequence in ARM Cortex-M3 processor. **(07 Marks)**

c. Explain the three subdivided Program Status Registers (PSRs). **(06 Marks)**

OR

2 a. Explain the features of Nested Vectored Interrupt Controller (NVIC) included in ARM Cortex-M3 processor. **(07 Marks)**

b. With a neat transition diagram, explain the operation modes and privilege levels available in Cortex-M3 processor. **(07 Marks)**

c. Briefly explain the Thumb-2 Technology and Instruction Set Architecture. **(06 Marks)**

Module-2

3 a. Find the value of register after each of the following construction are executed:

(i) LDR R1, [R0, #4]!, consider a system set as big endian with following memory layout and R0 = 0X20008000.

Address	Data
0X20008007	0X79
0X20008006	0XCD
0X20008005	0XA3
0X20008004	0XFD
0X20008003	0X0D
0X20008002	0XEB
0X20008001	0X2C
0X20008000	0X1A

(ii) SBFX R4, R3, #4, #8; Assume R3 = 0X1234CDEF

(iii) REV.W R2, R5; Assume R5 = FEFF3278

(iv) BIC R2, R0, R1; Assume R0 = 0XD5755755, R1 = 0XAABAAAA9 **(08 Marks)**

b. Translate the following C code into an assembly code using I T (IF-THEN) instruction block:

```
if ( R0 = = R1) then
  { R3 = R4 + R5
    R3 = R3/2}
else
  { R3 = R6 + R7
    R3 = R3/2}
```

(06 Marks)

c. With a neat diagram of Cortex Microcontroller Software Interface Standard (CMSIS) Structure, explain the CMSIS organization. **(06 Marks)**

OR

4 a. Find the value of register after each of the following instruction are executed.

(i) SSAT.W R1, #16, R0; Assume R0 = 0XFFFF7FFF

(ii) ORN R2, R0, R1; Assume R0 = 0XD5755755, R1 = 0XAABAAAA9

(iii) ASR R1, R0, #2; Assume R0 = 0X00001016

(iv) SXTB R1, R0; Assume R0 = 0X11228091 **(08 Marks)**

b. Translate the following C code into an Assembly code:

```
int i = 10;
int sum = 0;
while (i > 0) {
sum + = i;
i – – ;
   }
```

(06 Marks)

c. Explain the method to access System Tick (SYSTICK) Timer peripheral registers as pointers to elements in a Data Structure in C language. (06 Marks)

Module-3

5 a. Explain the Zig Bee communication interface. (07 Marks)

b. With a neat diagram, explain the operation of Static RAM (SRAM) cell. (07 Marks)

c. Classify embedded systems based on complexity and performance. (06 Marks)

OR

6 a. With a neat diagram, explain the I2C communication interface. (07 Marks)

b. Differentiate between Embedded System and General Purpose computing system. (07 Marks)

c. With a neat diagram, explain the features of Opto coupler. (06 Marks)

Module-4

7 a. Define operational quality attribute and explain any three operational quality attributes to be considered in embedded system design. (07 Marks)

b. With a neat functional block diagram, explain the working of a washing machine. (07 Marks)

c. Explain 'super loop' based and Embedded Operating System (OS) based embedded firmware design approaches. (06 Marks)

OR

8 a. Explain the different communication buses used in automotive application. (07 Marks)

b. Design an automatic tea/coffee vending machine based on FSM model for the following requirement:
The tea/coffee vending is initiated by user inserting a 5 rupee coin. After inserting the coin, the user can select 'Coffee' or 'Tea' or press 'Cancel' to cancel the order and take back the coin. (07 Marks)

c. Explain the 'High Level Language' based 'Embedded firmware' development technique. (06 Marks)

Module-5

9 a. Three processes with process IDs P1, P2, P3 with estimated completion time 10, 5, 7 milliseconds respectively enters the ready queue together. A new process P4 with estimated completion time 2 ms enters the 'Ready' queue after 2 ms. Calculate the waiting time and turnaround time for all the processes. Also calculate average waiting time and average turnaround time. The algorithm used is Shortest Job First (SJF) based pre-emptive scheduling. Assume all the process contains only CPU operation and no I/O operation are involved. (10 Marks)

b. Explain the Out-of-Circuit and In System Programming (ISP) techniques for embedding the firmware in to hardware. (10 Marks)

OR

10 a. Explain the different functional and non-functional requirements that needs to be evaluated in the selection of Real-Time Operating Systems (RTOS). (10 Marks)

b. With a neat diagram, explain the Boundary Scan based hardware debugging. (10 Marks)

* * * * *