UTHM
Universiti Tun Hussein Onn Malaysia

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER I
SESSION 2022/2023

COURSE NAME	:	EMBEDDED SYSTEMS DESIGN
COURSE CODE	:	BEJ42203 / BEC41703
PROGRAMME CODE	:	BEJ
EXAMINATION DATE	:	FEBRUARY 2023
DURATION	:	3 HOURS
INSTRUCTIONS	:	1. ANSWER **ALL** QUESTIONS. 2. THIS FINAL EXAMINATION IS CONDUCTED VIA **CLOSED BOOK**. 3. STUDENTS ARE **PROHIBITED** TO CONSULT THEIR OWN MATERIAL OR ANY EXTERNAL RESOURCES DURING THE EXAMINATION CONDUCTED VIA CLOSED BOOK

THIS QUESTION PAPER CONSISTS OF **SIX (6)** PAGES

Q1 (a) List **FOUR (4)** critical design metrics to design embedded systems for consumer electronics applications.

(4 marks)

(b) Compare the hardware architecture between embedded systems and general purpose systems in terms of types of components and capacity or size.

(4 marks)

(c) Select the most suitable platform either FPGA and microcontroller for telecommunication networking devices with real time processing and higher energy efficiency. Justify your answer.

(5 marks)

(d) Suggest design methods to implement parallel processing in software and hardware using Zynq-7000 SoC FPGA for high performance applications.

(4 marks)

Q2 **Figure Q2** provides the specifications for an embedded system. Answer the following questions:

- an autonomous mopping robot for restaurant kitchen and dining area.
- requires high energy efficiency to operate in long duration per battery charge.
- compact size to function in various spaces.
- short developing time as possible.

Figure Q2

(i) Define hardware software co-design methodology.

(4 marks)

(ii) List **FOUR (4)** major design tasks in hardware software co-design flow.

(4 marks)

(iii) If hardware software co-design is used to develop the system, discuss whether the requirements in **Figure Q2** can be met or not.

(5 marks)

(iv) Select the most suitable hardware software co-design approach either software-centric, hardware driven and iterative partitioning to design the system with high energy efficiency using Zynq-7000 SoC FPGA. Give a reason to support your answer.

(5 marks)

Q3 **Figure Q3** provides the specifications for an embedded system that requires real time operating system (RTOS). Answer the following questions:

- Task 1 – will stop all tasks when errors are detected during operation or an emergency button is pressed.
- Task 2 – a camera will continuously take photos of items running through a conveyor in 1 ms interval.
- Task 3 – will execute and complete within 1 ms based on the output photos from task 2 or total system failure will occur irrespective of other functions.
- Task 4 – will execute and complete within 5 ms based on the output of task 3.

Figure Q3

(i) Explain briefly the real time operating system (RTOS).

(3 marks)

(ii) Hard real time is more suitable than soft real time for the system. Give a reason to support the statement.

(2 marks)

(iii) List **THREE (3)** possible states for tasks 1, 2, 3 and 4.

(3 marks)

(iv) Arrange the appropriate priority for tasks 1, 2, 3 and 4 starting with the highest priority to the lowest priority.

(4 marks)

(v) Elaborate the concept of preemptive scheduling in RTOS based on the priority setting assigned in **Q3(iv)**.

(4 marks)

Q4 **Listing Q4** shows an excerpt of a real time application using FreeRTOS. Answer the following questions:

```
1   #include "semphr.h"
2
3   SemaphoreHandle_t xSemaphore;
4   TaskHandle_t xHandleTask1, xHandleTask2, xHandleTask3;
5
6   int main() {
7           xSemaphore = xSemaphoreCreateBinary();
8           xTaskCreate(task1, "task1", 100, NULL, 1, &xHandleTask1);
9           xTaskCreate(task2, "task2", 100, NULL, 1, &xHandleTask2);
10          xTaskCreate(task3, "task3", 100, NULL, 1, &xHandleTask3);
11          xSemaphoreGive(xSemaphore);
12  }
13
14  void task1(void *pvParameters) {
15          while(1) {
16          xSemaphoreTake(xSemaphore, portMAX_DELAY);
17          task1_function();
18          xSemaphoreGive(xSemaphore);
19          vTaskDelay(10);
20          }
21  }
22
23  void task2(void *pvParameters) {
24          while(1) {
25          xSemaphoreTake(xSemaphore, portMAX_DELAY);
26          task2_function();
27          xSemaphoreGive(xSemaphore);
28          vTaskDelay(10);
29          }
30  }
31
32  void task3(void *pvParameters) {
33          while(1) {
34          xSemaphoreTake(xSemaphore, portMAX_DELAY);
35          task3_function();
36          xSemaphoreGive(xSemaphore);
37          vTaskDelay(10);
38          }
39  }
```

Listing Q4

(i) Write the code to modify line 7 in the program by using mutex semaphore for the task's synchronization.

(2 marks)

(ii) Write the code to change the priority level of task3, task2 and task1 to be the highest, second highest and lowest priority respectively.

(6 marks)

(iii) Write the code to suspend task3 and task2 for 10ms. Assume configTICK_RATE_HZ is defined 1000.

(6 marks)

(iv) Write the code to resume suspended task2 and task3.

(4 marks)

(v) Write the code to delete the semaphore created in **Q4(i)**.

(2 marks)

Q5 (a) Describe the reasons for having many types of interface protocol (UART, I2C, SPI, CAN bus) for embedded system applications.

(2 marks)

(b) Differentiate between I2C, SPI and CAN bus in terms of error detection and correction.

(4 marks)

(c) Automotive applications primarily use CAN bus for the electronic communication system. Discuss the characteristics of CAN bus for such application.

(4 marks)

Q6 (a) List **THREE (3)** phases of testing.

(3 marks)

(b) Elaborate **TWO (2)** negative implications of less testing during embedded system development.

(4 marks)

(c) Discuss the impact of software testing to the development time and cost of an embedded system.

(4 marks)

(d) Compare between functional testing and structural or coverage testing using plate number recognition system as an example.

(4 marks)

Q7 **Listing Q7** shows an excerpt of C program for an embedded system application. Answer the following questions:

```
1   #include <stdio.h>
2
3   int fibonacci(int i) {
4       if(i == 0) {
5       return 0;
6       }
7
8       if(i == 1) {
9       return 1;
10      }
11
12      return fibonacci(i-1) + fibonacci(i-2);
13  }
14
15  int main() {
16      int i;
17
18      for (i = 0; i < 10; i++) {
19      printf("%d\t\n", fibonacci(i));
20      }
21
22      return 0;
23  }
```

Listing Q7

(i) Suggest **ONE (1)** example to perform white box testing. Please cite specific line of code from **Listing Q7** to support the answer.

(2 marks)

(ii) Suggest **ONE (1)** example to perform exploratory testing. Please cite specific line of code from **Listing Q7** to support the answer.

(2 marks)

– END OF QUESTIONS –