UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2021/2022

COURSE NAME	:	EMBEDDED SYSTEMS DESIGN
COURSE CODE	:	BEJ42203
PROGRAMME CODE	:	BEJ
EXAMINATION DATE	:	JULY 2022
DURATION	:	3 HOURS
INSTRUCTION	:	1. ANSWER ALL QUESTIONS. 2. THIS FINAL EXAMINATION IS ONLINE ASSESSMENT AND CONDUCTED VIA **OPEN BOOK**.

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

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

(4 marks)

(b) Compare the hardware architecture between embedded systems and general purpose systems in terms of similarities and differences.

(4 marks)

(c) Between SRAM, FLASH and anti-fuse FPGA technology, select the most suitable technology for wearable devices in terms of high logic density, configurability and non-volatile requirements. Justify your answer.

(4 marks)

(d) You are developing an embedded system application using Zynq-7000 SoC FPGA. The application requires high performance operation through parallel processing. Suggest design methods related to programmable logic (PL) section in the Zynq-7000 SoC FPGA to incorporate parallel processing for the application.

(4 marks)

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

- an autonomous drainage system inspection robot with a camera to detect and report blockages.
- main components includes image processing in HD to detect blockages, distance tracker and storing blockage images.
- requires high energy efficiency to operate at long duration per charge.
- compact size to inspect different drainage sizes.
- short development time as possible.

Figure Q2

(i) Define hardware software co-design methodology.

(4 marks)

(ii) Argue how hardware software co-design can be used to develop the system.

(4 marks)

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

(4 marks)

(iv) Decide the hardware software partitioning of the components to achieve higher energy efficiency and compact system size as per specifications.

(4 marks)

(v) Between software-centric, hardware driven and iterative partitioning approaches, select the most suitable hardware software co-design to design the system using Zynq-7000 SoC FPGA with short development time. Give **TWO (2)** reasons 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 – a camera will continuously take photos of items running through a conveyor.
- Task 2 – will execute and complete within 1ms after task 1 or total system failure will occur irrespective of other functions.
- Task 3 – will execute and complete within 5 ms after task 2.
- Task 4 – will only execute to stop all tasks when specific conditions related to the items are met.

Figure Q3

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

(2 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) Assign suitable priority setting to Task 1, 2, 3 and 4 with the highest priority first.

(4 marks)

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

(4 marks)

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

```
#include "semphr.h"

SemaphoreHandle_t xSemaphore;
TaskHandle_t xHandleTask1, xHandleTask2, xHandleTask3;

int main() {
        xSemaphore = xSemaphoreCreateBinary();
        xTaskCreate(task1, "task1", 100, NULL, 0, &xHandleTask1);
        xTaskCreate(task2, "task2", 100, NULL, 0, &xHandleTask2);
        xTaskCreate(task3, "task3", 100, NULL, 0, &xHandleTask3);
        xSemaphoreGive(xSemaphore);
}

void task1(void *pvParameters) {
        while(1) {
        xSemaphoreTake(xSemaphore, portMAX_DELAY);
        task1_function();
        xSemaphoreGive(xSemaphore);
        vTaskDelay(10);
        }
}

void task2(void *pvParameters) {
        while(1) {
        xSemaphoreTake(xSemaphore, portMAX_DELAY);
        task2_function();
        xSemaphoreGive(xSemaphore);
        vTaskDelay(10);
        }
}

void task3(void *pvParameters) {
        while(1) {
        xSemaphoreTake(xSemaphore, portMAX_DELAY);
        task3_function();
        xSemaphoreGive(xSemaphore);
        vTaskDelay(10);
        }
}
```

Listing Q4

(i) Identify the priority for task1, task2 and task3.

(3 marks)

(ii) Write the code to modify line 7 in the program to use mutex semaphore for the task's synchronization.

(2 marks)

(iii) Write the code to change the priority level of task3 and task2 to be the highest and second highest priority respectively.

(4 marks)

(iv) Write the code to suspend task1 for 10ms and continue execution after that. Assume configTICK_RATE_HZ is defined 1000.

(6 marks)

Q5 Based on the following specifications, select a suitable communication interface between UART, I2C, SPI and CAN protocol in an FPGA-based embedded system design. Justify your answers.

(i) Secure Digital (SD) memory card interface to FPGA board for reading and writing large data.

(3 marks)

(ii) Clock timer module interface to FPGA for real time clock input.

(3 marks)

(iii) Communication between personal computer and FPGA board to send short messages.

(3 marks)

(vi) Communication between a few devices using high speed, accurate and reliable data transfer.

(3 marks)

Q6 (a) Elaborate **TWO (2)** negative implications of not doing enough testing during embedded system development.

(4 marks)

(b) An embedded software engineer is testing software program for smart fridge application. 5% structures remain to be tested and 90% testing are completed. Compare testing coverage and testing completeness based on the given condition.

(4 marks)

(c) Relate software testing to the reliability and development time of an embedded system.

(4 marks)

(d) Compare between black box testing and white box testing using plate number recognition system as an example.

(4 marks)

Q7 **Listing Q7** shows the code snippet to send frames using CAN bus in Zynq-7000 SoC FPGA. Answer the following questions:

```
static void SendFrame(XCanPs *InstancePtr) {
        u8 *FramePtr;
        int Index;
        int Status;

        TxFrame[0] = (u32)XCanPs_CreateIdValue((u32)MESSAGE_ID, 0, 0, 0, 0);
        TxFrame[1] = (u32)XCanPs_CreateDlcValue((u32)FRAME_DATA_LEN);

        FramePtr = (u8 *)(&TxFrame[2]);

        for (Index = 0; Index < FRAME_DATA_LEN; Index++) {
                *FramePtr++ = (u8)Index;
        }

        while (XCanPs_IsTxFifoFull(InstancePtr) == TRUE);

        Status = XCanPs_Send(InstancePtr, TxFrame);

        if (Status != XST_SUCCESS) {
                LoopbackError = TRUE;
                SendDone = TRUE;
                RecvDone = TRUE;
        }
}
```

Listing Q7

(i) Identify the phase of testing for the given code.

(1 mark)

(ii) Suggest **TWO (2)** methods to conduct white box testing.

(6 marks)

– END OF QUESTIONS –