UTHM
Universiti Tun Hussein Onn Malaysia

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2023/2024

COURSE NAME	:	EMBEDDED SYSTEMS DESIGN
COURSE CODE	:	BEJ42203
PROGRAMME CODE	:	BEJ
EXAMINATION DATE	:	JULY 2024
DURATION	:	3 HOURS
INSTRUCTIONS	:	1. ANSWER ALL QUESTIONS. 2. THIS FINAL EXAMINATION IS CONDUCTED VIA □ Open book ☒ 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 **SEVENTEEN (17)** PAGES

Q1 Answer the following questions about embedded systems characteristics and design platforms.

(a) Embedded systems and general purpose systems are similar in some ways. General purpose systems can be used to run all applications but it is not cost efficient and impractical. Many applications require specific hardware and software components to run the desired functions. Compare the hardware architecture between embedded systems and general purpose systems.

(4 marks)

(b) Reliability is a crucial design factor in many embedded system applications such as vehicle control systems and industrial manufacturing systems. Without reliability, many embedded system applications will not be useful to users and may be dangerous. Define reliability for embedded system devices.

(3 marks)

(c) Computer vision systems are being used in the manufacturing industry to increase operation efficiency. One of the applications of computer vision is product's quality inspection where defects can be analysed in real time using multiple high definition cameras. Such application requires real time processing and high performance system to be able to function accurately and reliably. Select the most suitable platform between Field Programmable Gate Arrays (FPGA) and microcontroller for such applications. Justify your answer.

(5 marks)

Q2 **Figure Q2.1** provides the specifications for an embedded system. Answer the following questions:

- Autonomous border control drone that can detect and locate illegal border crossing people and smuggler through jungle.
- Hardware and software architecture must be fully optimised especially on the thermal camera technology to maximise flight duration during day and night.
- Short developing time such that revised detection model with higher efficiency can be designed and deployed quickly.

Figure Q2.1 Specifications for an autonomous surveillance drone

(a) Define hardware software co-design methodology.

(4 marks)

(b) One of the specifications in **Figure Q2.1** is short development time. Hardware software co-design can be implemented either through software-centric, hardware driven or iterative partitioning. Select the most suitable co-design approach to reduce application development time. Justify your answers.

(5 marks)

(c) Hardware software co-design does not offer advantages for all types of embedded systems. Only certain applications with specific characteristics will benefits from this approach. Identify the types of embedded systems that are not advantages to apply hardware software co-design. Justify your answers.

(4 marks)

Q3 **Figure Q3.1** provides the specifications for an embedded system that requires real time operating system (RTOS). Answer the following questions:

- Task A – will stop all tasks in execution when errors are detected during operation or an emergency button is pressed.
- Task B – a camera will continuously take photos of items running through a conveyor in 1 ms interval.
- Task C – will execute and complete within 5 ms based on the output photos from task B or total system failure will occur irrespective of other functions.
- Task D – will execute and complete within 15 ms based on the output of task C or total system failure will occur irrespective of other functions.

Figure Q3.1 Specifications for an embedded system application using RTOS

(a) Differentiate between Real Time Operating System (RTOS) and general purpose operating systems in terms of preemptive scheduling and latency bounded. Your answers should relate to the specifications given in **Figure Q3.1**.

(6 marks)

(b) Assign the appropriate priority level for tasks A, B, C and D starting with the highest priority to the lowest priority. Justify the priority setting that you choose.

(6 marks)

(c) Elaborate the concept of preemptive scheduling in RTOS based on the priority setting assigned in **Q3(b)**. Use appropriate diagram to support your answer.

(4 marks)

(d) Although RTOS can be used to implement real time processing in embedded systems, several issues must be considered to decide between RTOS or bare metal applications (non-RTOS). If these issues are not carefully analysed in the design development, it will have negative impacts to the overall performance. Explain **TWO (2)** challenges faced when using RTOS in embedded systems design.

(4 marks)

(e) Real time systems are classified into hard, firm and soft real time. It is crucial to identify suitable classification to design an embedded system application where necessary implementation mechanisms can be adopted. From the specification in **Figure Q3.1**, determine the suitable classification. Justify your answer.

(3 marks)

Q4 **Listing Q4.1** shows an excerpt of a real time application using FreeRTOS.

```
#include "semphr.h"

SemaphoreHandle_t xSemaphore;
TaskHandle_t xHandleTask1, xHandleTask2, xHandleTask3;

int main() {
    xSemaphore = xSemaphoreCreateBinary();
    xTaskCreate(task1, "task_store_image_camera1", 100, NULL, 3, &xHandleTask1);
    xTaskCreate(task2, "task_store_image_camera2", 100, NULL, 2, &xHandleTask2);
    xTaskCreate(task3, "task_store_image_camera3", 100, NULL, 1, &xHandleTask3);
    xSemaphoreGive(xSemaphore);
}

void task_store_image_camera1(void *pvParameters) {
    while(1) {
        xSemaphoreTake(xSemaphore, portMAX_DELAY);
        store_image_camera1();
        xSemaphoreGive(xSemaphore);
        vTaskDelay(10);
    }
}

void task_store_image_camera2(void *pvParameters) {
    while(1) {
        xSemaphoreTake(xSemaphore, portMAX_DELAY);
        store_image_camera2();
        xSemaphoreGive(xSemaphore);
        vTaskDelay(10);
    }
}

void task_store_image_camera3(void *pvParameters) {
    while(1) {
        xSemaphoreTake(xSemaphore, portMAX_DELAY);
        store_image_camera3();
        xSemaphoreGive(xSemaphore);
        vTaskDelay(10);
    }
}
```

Listing Q4.1 Excerpt of a real time application program using FreeRTOS

(a) Determine the highest and lowest priority levels for tasks in the code. Justify your answers.

(3 marks)

(b) Explain the operation of semaphore synchronization using specific examples from the code. Cite relevant line numbers from the code in your explanation.

(4 marks)

(c) The code in **Listing Q4.1** uses binary semaphore for synchronization. If the binary semaphore is replaced with mutex semaphore, predict the effect to the code execution.

(4 marks)

(d) Accurate timing is important especially for hard real time applications to have deterministic execution. One way to achieve deterministic execution is by assigning different priority levels for the tasks which can be implemented through several methods. Suggest **TWO (2)** possible methods to assign priority levels for all tasks in the code. Justify your answer.

(4 marks)

Q5 Answer the following questions about interfaces in embedded systems design.

(a) Different serial interface protocols exist that can be used to design various applications depending on the requirements. As an embedded system engineer working on an embedded system project, you need to decide the suitable interface protocol for your system. Thus, you need to clearly understand each interface protocol. Differentiate between I2C, SPI and UART protocols in terms of their unique characteristics.

(6 marks)

(b) In automotive applications, CAN bus is used for critical electronic communication systems such as braking and safety systems due to high reliability compared to other interface protocols. CAN bus has better communication reliability is because of differential signaling. Discuss differential signaling mechanism using appropriate diagram.

(4 marks)

Q6 Answer the following questions about system integration in embedded systems design.

(a) System integration is crucial for embedded systems design. It significantly impacts the overall system performance and therefore must be carefully designed. Clock is one of the important elements in the system integration that has significant impact to the overall power consumption and performance of the system. Elaborate the impact of clock design to the performance and power consumption.

(4 marks)

(b) Several power management methods to design power-sensitive embedded systems can be implemented using using Zynq-7000 SoC FPGA. The methods can applied either to programmable logic (PL) and processing system (PS) sections or both for maximum impact. Explain **TWO (2)** power management methods applicable to PL section.

(4 marks)

(c) Reset mechanisms can also contribute to the overall power consumption optimization in embedded system applications. In Zynq-7000 SoC FPGA, several reset mechanisms can be implemented in PS and PL sections. Describe **THREE (3)** general guidelines to use reset in PL section to optimize power consumption.

(3 marks)

Q7 **Listing Q7.1** shows an excerpt of code for a specific application.

```
#include <stdio.h>
#include <math.h>
#include <complex.h>
#include <xtime_l.h>

double PI;
typedef double complex cplx;

void _fft(cplx buf[], cplx out[], int n, int step)
{
    if (step < n) {
        _fft(out, buf, n, step * 2);
        _fft(out + step, buf + step, n, step * 2);
        for (int i = 0; i < n; i += 2 * step) {
            cplx t = cexp(-I * PI * i / n) * out[i + step];
            buf[i / 2]     = out[i] + t;
            buf[(i + n)/2] = out[i] - t;
        }
    }
}

void fft(cplx buf[], int n)
{
    cplx out[n];
    for (int i = 0; i < n; i++) out[i] = buf[i];
    _fft(buf, out, n, 1);
}

void show(const char * s, cplx buf[])
{
    printf("%s", s);
    for (int i = 0; i < 8; i++){
        if (!cimag(buf[i]))
            printf("%g  ", creal(buf[i]));
        else
            printf("(%g, %g)  ", creal(buf[i]), cimag(buf[i]));
    }
}

```

Listing Q7.1 Excerpt of code for a specific application

```
int main()
{
    XTime start, end;
    PI = atan2(1, 1) * 4;
    cplx inOut[] = {1, 1, 1, 1, -1, -1, -1, -1};
    show("Data: ", inOut);
    XTime_GetTime(&start);
    fft(inOut, 8);
    XTime_GetTime(&end);
    show("\n FFT: ", inOut);
    printf("\n Execution Time: %f",(end-start)*1000000.0/COUNTS_PER_SECOND);
    return 0;
}
```

Listing Q7.1 Excerpt of code for a specific application (continued)

(a) Several methods can be used to test embedded software applications such as unit, subsystem and system testing. Differentiate between unit testing and system testing by citing relevant codes in **Listing Q7.1**.

(4 marks)

(b) Embedded software testing is implemented using functional and structural testing to detect bugs and eventually improve reliability. Differentiate between functional and structural testing. Your answers should relate to the code excerpt in **Listing Q7.1**.

(4 marks)

(c) Embedded software testing can improve system's reliability, but other factors must be considered during the development such as time-to-market and cost. Test-driven development (TDD) method is a software development method that can increase reliability through extensive testing process. Discuss the impact of TDD to the time-to-market and cost in embedded systems development.

(4 marks)

(d) You have been instructed to develop an embedded system application for transportation management system in logistics industry. The application uses vision systems to automatically monitor trucks entering and leaving warehouses. Using the code in **Listing Q7.1** that will be a part of the system application, describe the process to implement TDD.

(4 marks)

– END OF QUESTIONS –