CONFIDENTIAL

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2022/2023

COURSE NAME	:	MICROPROCESSOR AND MICROCONTROLLER
COURSE CODE	:	BEJ 30203
PROGRAMME CODE	:	BEJ
EXAMINATION DATE	:	JULY / AUGUST 2023
DURATION	:	3 HOURS

INSTRUCTION

1. ANSWER **ALL QUESTIONS** IN **PART A** AND **ONE (1)** QUESTION IN **PART B**.
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 **THIRTEEN (13)** PAGES

CONFIDENTIAL

PART A

Q1 (a) Explain the differences between microprocessor and microcontroller.

(6 marks)

(b) Discuss why the Reduced Instruction Set Computer (RISC) implements "Load-Store" architecture.

(5 marks)

(c) For a 16-bit microprocessor, determine the value of N, Z, V, C flags after the microprocessor executes the following arithmetic operations:

(i) 127 + 2

(1 mark)

(ii) 32766 + 2

(1 mark)

(iii) -2000 + 2000

(1 mark)

(iv) -100 + 200

(1 mark)

(v) -100 – 200

(1 mark)

(d) Suppose that the Program Counter (register R15) contains the hex value 0x00004008. From what address would a Cortex-M3 fetch an instruction (current instruction). Assume that all instructions are 32-bit wide.

(2 marks)

(e) What is the size of the ARM Cortex-M3's address bus?

(2 marks)

Q2 (a) Given a Cortex-M3 instruction as follow, **STR R1, [R0, #4]** where the initial values in R0 and R1 are 0x10000000 and 0x12345678 respectively,

(i) Name the addressing mode of the instruction.

(1 mark)

(ii) Determine the memory address and its content after the Cortex-M3 executes the instruction.

(4 marks)

(iii) What is the content of R0 after the Cortex-M3 executes the instruction?

(1 mark)

(b) **Figure Q2** shows an assembly program of a Cortex-M3 and its disassembly listing. From the program, answer the following questions:

(i) Explain what the program does.

(3 marks)

(ii) Draw a flowchart to represent the operation of the program.

(4 marks)

(iii) Give the machine code of the instruction **ADDS R5, R4**.

(1 mark)

(iv) Determine the size of the program in bytes.

(2 marks)

(v) Give the start and end address of the program.

(2 marks)

(vi) Determine the value in R5 after the Cortex-M3 executes the program if the initial value in R4 is 5.

(2 marks)

Q3 (a) **Figure Q3(a)** shows a simple assembly program to access the LPC1768 microcontroller input/output (I/O) ports.

(i) From the program, give the address of the port direction control register for *Port*2, *Byte*0.

(2 marks)

(ii) Explain the instructions in Line 5 and 6, respectively. Give the purpose of both instructions in this program.

(4 marks)

(iii) The *Delay* subroutine is used to provide a delay to the program when it is called. Explain how to increase the delay time in the program.

(2 marks)

(b) By referring to **Figure Q3(b)**, write a C program using a Common Microcontroller Software Interface Standard (CMSIS) to perform the operation as follows: when sw1 is pressed, the LEDs will act as a binary counter. While when sw2 is pressed, all the LEDs will turn off. You need to provide the program flowchart, and use these codes for input/output initialization:

LPC_GPIO1->FIODIR0 |= 0xFC; // Configure bit P1.0 sw1, P1.1 as sw2
LPC_GPIO2->FIODIR0 |= 0xFF; // Configure bit P2.7 - P2.0 as outputs

(12 marks)

Q4 (a) For an Analog to Digital Converter (ADC) with 10-bit resolution, convert the following analog voltage to its equivalent digital value if the reference voltage is 3.3V.

(i) 500 mV

(1 mark)

(ii) 2.5 V

(1 mark)

(iii) 3.25 V

(1 mark)

(iv) 1.4 V

(1 mark)

(b) **Figure Q4(b)** shows a program to read the analog input from an LM35 sensor through the AD0.0 pin of a LPC1768 microcontroller and turn on the LEDs connected to P2.0, P2.1 and P2.2 according to the reading temperature. Based on the given program, answer the following questions.

(i) Explain the code in Line 15, which is

```
result = ((LPC_ADC -> ADDR0 >> 4) & 0xFFF);
```

(3 marks)

(ii) The A/D Control Register (ADCR) and Pin Function Select Register 1 (PINSEL1) are given in **Figure Q4(b)(i)** and **(ii)**, respectively. Modify the program in **Figure Q4(b)** such that the microcontroller now reads the analog input from AD0.2. *Note*: show the codes that are modified only.

(4 marks)

(iii) Modify the program such that the LED at P2.2 will turn on if the temperature is more than 40°C. *Note*: show the codes that are modified only.

(2 marks)

(c) **Figure Q4(c)(i)** shows the initialization of pulse width modulation (PWM) for the LPC1768 microcontroller to create a PWM signal with the following specifications:

PWM frequency = 1 KHz
PWM duty cycle = 50%
PWM channel = PWM1.1

By referring to the servo motor characteristics in **Figure 4(c)(ii)**, write a program to move the servo motor to 0°, 90° and 180° for 1s, 2s and 3s respectively.

Use this function to create 1s delay: **wait_ms(1000);**

(7 marks)

PART B

Q5 (a) Describe how a timer circuit can be implemented in hardware as part of a microprocessor's architecture.

(4 marks)

(b) If you are required to design an application that needs precise timings, discuss how a 1ms timing resolution can be achieved by using the LPC1768 hardware timer. Given the microcontroller is clocked at 100 MHz, the peripheral clock (PCLK) is set at CCLK/4.

(5 marks)

(c) **Figure Q5(c)** shows a program that detects the interrupt input at pin P0.0 of the LPC1768 microcontroller. The interrupt input is detected at the falling edge of the input. Based on the given program, answer the following questions:

(i) Modify the code such that the interrupt is now detected at the rising edge of the input. *Note*: show the codes that are modified only.

(2 marks)

(ii) Explain what will happen if the code in Line 20 is removed?

(4 marks)

(d) Based on the program in **Q5(c)**, write a new program to turn on an LED connected to P2.0 whenever the input at P0.0 is changed from 0 to 1 by using interrupt. The LED will remain on if P2.0 remains 1 and will turn off when P2.0 changes to 0.

(5 marks)

Q6 (a) If the LPC1768 is used as a master in an application that uses serial communication to transfer the data between devices. What are the limitations for the number of devices that can be connected to the LPC1768 for I2C, SPI or UART bus?

(3 marks)

(b) The LPC1768 is configured as SPI master and is to be connected to three other LPC1768, each configured as slave. Sketch a circuit showing how this interconnection could be made.

(6 marks)

(c) The serial data is needed to be transmitted at 115200 baud using the UART communication protocol. The baud rate equation for the LPC1768 microcontroller is given by

$$Baud\ rate = \frac{PCLK}{16 \times (256 \times DLM + DLL) \times \left(1 + \frac{DivAddVal}{MulVal}\right)}$$

Given that the microcontroller is clocked at 100 MHz and PCLK = CCLK/4 is selected, determine the value of:

(i) MulVal

(1 mark)

(ii) DivAddVal

(1 mark)

(iii) DLM

(1 mark)

(iv) DLL

(1 mark)

Note that the The value of *MulVal* and *DivAddVal* should comply to the following conditions:

- $1 \leq MulVal \leq 15$
- $0 \leq DivAddVal \leq 14$
- $DivAddVal < MulVal$

(d) You need to set up a serial network, which will have one master and four slaves. By using the SPI protocol, every second, data has to be distributed such that one byte is sent to Slave 1, four to Slave 2, three to Slave 3 and four to Slave 4. If the complete data transfer must take not more than 200 ms, estimate the minimum clock frequency that is allowable for the SPI. Assume there are no other timing overheads.

(5 marks)

-END OF QUESTIONS-

FINAL EXAMMINATION

SEMESTER / SESSION	: SEM II 2022/2023	PROGRAMME CODE	: BEJ
COURSE NAME	: MICROPROCESSOR AND MICROCONTROLLER	COURSE CODE	: BEJ 30203

Assembly program

```
__main   MOV    R3, #200
         MOV    R5, #0
loop     ADDS   R5, R4
         SUBS   R3, #1
         BNE    loop
         NOP
         END
```

Disassembly listing

```
0x00000110   F04F03C8   MOV    r3, #0xC8
0x00000114   F04F0500   MOV    r5, #0x00
0x00000118   192D       ADDS   r5, r5, r4
0x0000011A   3B01       SUBS   r3, r3, #0x01
0x0000011C   D1FC       BNE    0x00000118
0x0000011E   BF00       NOP
```

Figure Q2

```
FIO2DIR0   EQU    0x2009C040
FIO2PIN0   EQU    0x2009C054

__main     LDR    R0, =FIO2DIR0
           LDR    R1, =FIO2PIN0
           MOV    R2, #0xFF
           STRB   R2, [R0]
Loop       MOV    R2, #0x20
           STRB   R2, [R1]
           BL     Delay
           MOV    R2, #0x00
           STRB   R2, [R1]
           BL     Delay
           B      Loop

Delay      MOVW   R3, #0x0000
           MOVT   R3, #0x0010
Ag         SUBS   R3, #1
           BNE    Ag
           BX     LR
```

Figure Q3(a)

FINAL EXAMMINATION

SEMESTER / SESSION : SEM II 2022/2023 PROGRAMME CODE : BEJ
COURSE NAME : MICROPROCESSOR AND MICROCONTROLLER COURSE CODE : BEJ 30203

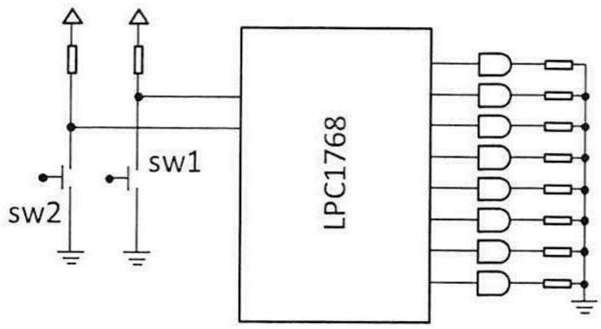

Figure Q3(b)

```
#include <lpc17xx.h>

void delay(void);
int main(void)
{
    unsigned int result = 0;
    float volts;
    LPC_SC -> PCONP |= (1<<12);            //Enable the ADC Peripheral
    LPC_ADC -> ADCR |= (1<<21);            //Power-on the ADC
    LPC_ADC -> ADCR |= (1<<8);             //Devide frequency by 1
    LPC_PINCON -> PINSEL1 |= (1<<14) ;     //select AD0.0 for P0.23
    LPC_GPIO2 -> FIODIR |= (7<<0);         //Output LEDs at P2.0, P2.1, P2.2

    while(1)
    {
        LPC_ADC -> ADCR |= (1<<24); //Start the conversion
        while((LPC_ADC -> ADGDR & (1u<<31)) == 0)
        {}                      //Wait for conversion to finish
        result = ((LPC_ADC -> ADGDR >> 4) & 0xFFF);
        volts = (result*3.3)/4096.0;
        if (volts < 0.2)
                LPC_GPIO2 -> FIOPIN0 = 0x04;
        else if (volts > 0.35)
                LPC_GPIO2 -> FIOPIN0 = 0x02;
        else
                LPC_GPIO2 -> FIOPIN0 = 0x01; //turn on "OK" LED
        delay();
    }
}
```

Figure Q4(b)

FINAL EXAMMINATION

SEMESTER / SESSION : SEM II 2022/2023
COURSE NAME : MICROPROCESSOR AND MICROCONTROLLER
PROGRAMME CODE : BEJ
COURSE CODE : BEJ 30203

A/D Control Register (ADCR)

Bit	Symbol	Value	Description	Reset value
7:0	SEL		Selects which of the AD0.7:0 pins is (are) to be sampled and converted. For AD0, bit 0 selects Pin AD0.0, and bit 7 selects pin AD0.7. In software-controlled mode, only one of these bits should be 1. In hardware scan mode, any value containing 1 to 8 ones is allowed. All zeroes is equivalent to 0x01.	0x01
15:8	CLKDIV		The APB clock (PCLK_ADC0) is divided by (this value plus one) to produce the clock for the A/D converter, which should be less than or equal to 13 MHz. Typically, software should program the smallest value in this field that yields a clock of 13 MHz or slightly less, but in certain cases (such as a high-impedance analog source) a slower clock may be desirable.	0
16	BURST	1	The AD converter does repeated conversions at up to 200 kHz, scanning (if necessary) through the pins selected by bits set to ones in the SEL field. The first conversion after the start corresponds to the least-significant 1 in the SEL field, then higher numbered 1-bits (pins) if applicable. Repeated conversions can be terminated by clearing this bit, but the conversion that's in progress when this bit is cleared will be completed. **Remark:** START bits must be 000 when BURST = 1 or conversions will not start.	0
		0	Conversions are software controlled and require 65 clocks.	
20:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
21	PDN	1	The A/D converter is operational.	0
		0	The A/D converter is in power-down mode.	
23:22	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
26:24	START		When the BURST bit is 0, these bits control whether and when an A/D conversion is started:	0
		000	No start (this value should be used when clearing PDN to 0).	
		001	Start conversion now.	
		010	Start conversion when the edge selected by bit 27 occurs on the P2.10 / EINT0 / NMI pin.	
		011	Start conversion when the edge selected by bit 27 occurs on the P1.27 / CLKOUT / USB_OVRCRn / CAP0.1 pin.	
		100	Start conversion when the edge selected by bit 27 occurs on MAT0.1. Note that this does not require that the MAT0.1 function appear on a device pin.	
		101	Start conversion when the edge selected by bit 27 occurs on MAT0.3. Note that it is not possible to cause the MAT0.3 function to appear on a device pin.	
		110	Start conversion when the edge selected by bit 27 occurs on MAT1.0. Note that this does not require that the MAT1.0 function appear on a device pin.	
		111	Start conversion when the edge selected by bit 27 occurs on MAT1.1. Note that this does not require that the MAT1.1 function appear on a device pin.	
27	EDGE		This bit is significant only when the START field contains 010-111. In these cases:	0
		1	Start conversion on a falling edge on the selected CAP/MAT signal.	
		0	Start conversion on a rising edge on the selected CAP/MAT signal.	
31:28	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Figure Q4(b)(i)

FINAL EXAMMINATION

SEMESTER / SESSION : SEM II 2022/2023 PROGRAMME CODE : BEJ
COURSE NAME : MICROPROCESSOR AND MICROCONTROLLER COURSE CODE : BEJ 30203

PIN FUNCTION SELECT REGISTER 1 (PINSEL1)

PINSEL1	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P0.16	GPIO Port 0.16	RXD1	SSEL0	SSEL	00
3:2	P0.17	GPIO Port 0.17	CTS1	MISO0	MISO	00
5:4	P0.18	GPIO Port 0.18	DCD1	MOSI0	MOSI	00
7:6	P0.19[1]	GPIO Port 0.19	DSR1	Reserved	SDA1	00
9:8	P0.20[1]	GPIO Port 0.20	DTR1	Reserved	SCL1	00
11:10	P0.21[1]	GPIO Port 0.21	RI1	Reserved	RD1	00
13:12	P0.22	GPIO Port 0.22	RTS1	Reserved	TD1	00
15:14	P0.23[1]	GPIO Port 0.23	AD0.0	I2SRX_CLK	CAP3.0	00
17:16	P0.24[1]	GPIO Port 0.24	AD0.1	I2SRX_WS	CAP3.1	00
19:18	P0.25	GPIO Port 0.25	AD0.2	I2SRX_SDA	TXD3	00
21:20	P0.26	GPIO Port 0.26	AD0.3	AOUT	RXD3	00
23:22	P0.27[1][2]	GPIO Port 0.27	SDA0	USB_SDA	Reserved	00
25:24	P0.28[1][2]	GPIO Port 0.28	SCL0	USB_SCL	Reserved	00
27:26	P0.29	GPIO Port 0.29	USB_D+	Reserved	Reserved	00
29:28	P0.30	GPIO Port 0.30	USB_D−	Reserved	Reserved	00
31:30	-	Reserved	Reserved	Reserved	Reserved	00

Figure Q4(b)(ii)

FINAL EXAMMINATION

SEMESTER / SESSION : SEM II 2022/2023
COURSE NAME : MICROPROCESSOR AND MICROCONTROLLER
PROGRAMME CODE : BEJ
COURSE CODE : BEJ 30203

```
LPC_PINCON->PINSEL4 |= (1<<0);          //Set pin function to PWM1.1, P2.0
LPC_PWM1->PCR = (1<<9);                 //PWM1 output enabled
LPC_PWM1->PR = 24;                      //Prescaler 1us resolution
LPC_PWM1->MR0 = 1000;                   //set PWM cycle to 1 ms
LPC_PWM1->MR1 = 500;                    //Set 50% Duty Cycle (0.5 ms)
LPC_PWM1->LER = (1<<1) | (1<<0);        //latch values in MR1 and MR0
LPC_PWM1->MCR |= (1<<1);                //Reset on PWM MR0 & TC if it matches MR0
LPC_PWM1->TCR = (1<<0) | (1<<3);        //enable counters and PWM Mode
```

Figure Q4(c)(i)

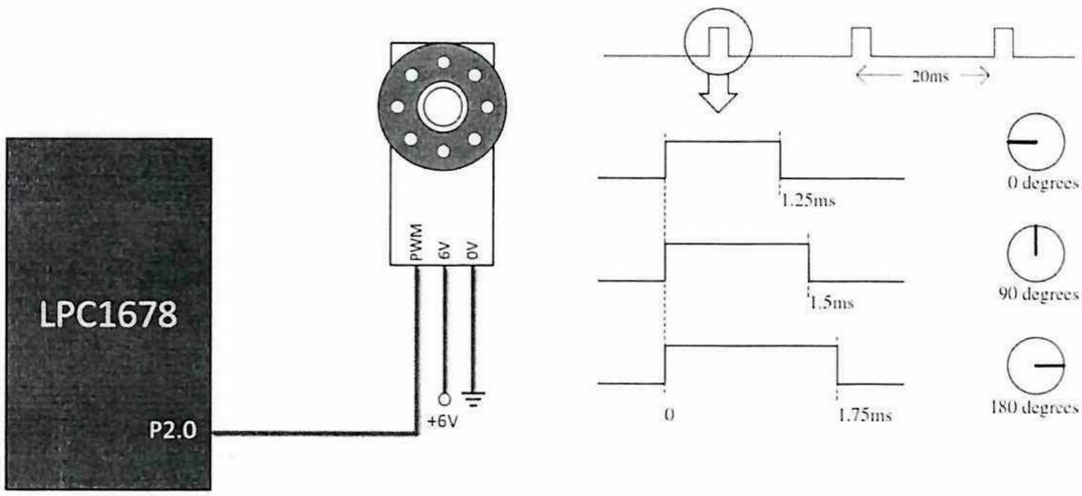

Figure Q4(c)(ii)

FINAL EXAMMINATION

SEMESTER / SESSION : SEM II 2022/2023 PROGRAMME CODE : BEJ
COURSE NAME : MICROPROCESSOR AND MICROCONTROLLER COURSE CODE : BEJ 30203

PIN FUNCTION SELECT REGISTER 4 (PINSEL4)

PINSEL4	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P2.0	GPIO Port 2.0	PWM1.1	TXD1	Reserved	00
3:2	P2.1	GPIO Port 2.1	PWM1.2	RXD1	Reserved	00
5:4	P2.2	GPIO Port 2.2	PWM1.3	CTS1	Reserved [2]	00
7:6	P2.3	GPIO Port 2.3	PWM1.4	DCD1	Reserved [2]	00
9:8	P2.4	GPIO Port 2.4	PWM1.5	DSR1	Reserved [2]	00
11:10	P2.5	GPIO Port 2.5	PWM1.6	DTR1	Reserved [2]	00
13:12	P2.6	GPIO Port 2.6	PCAP1.0	RI1	Reserved [2]	00
15:14	P2.7	GPIO Port 2.7	RD2	RTS1	Reserved	00
17:16	P2.8	GPIO Port 2.8	TD2	TXD2	ENET_MDC	00
19:18	P2.9	GPIO Port 2.9	USB_CONNECT	RXD2	ENET_MDIO	00
21:20	P2.10	GPIO Port 2.10	$\overline{\text{EINT0}}$	NMI	Reserved	00
23:22	P2.11[1]	GPIO Port 2.11	$\overline{\text{EINT1}}$	Reserved	I2STX_CLK	00
25:24	P2.12[1]	GPIO Port 2.12	$\overline{\text{EINT2}}$	Reserved	I2STX_WS	00
27:26	P2.13[1]	GPIO Port 2.13	$\overline{\text{EINT3}}$	Reserved	I2STX_SDA	00
31:28	-	Reserved	Reserved	Reserved	Reserved	0

Figure Q4(c)(ii)

FINAL EXAMMINATION

SEMESTER / SESSION	: SEM II 2022/2023		PROGRAMME CODE	: BEJ
COURSE NAME	: MICROPROCESSOR AND MICROCONTROLLER		COURSE CODE	: BEJ 30203

```
#include <lpc17xx.h>
void EINT3_IRQHandler(void)
void delay(void);
int main(void)
{
   NVIC_EnableIRQ(EINT3_IRQn);
   LPC_GPIOINT->IO0IntEnF |= (1<<0);
   LPC_GPIO2->FIODIR |=(3<<0);

   while(1)
   {
      LPC_GPIO2->FIOSET |=(1<<1);
      delay();
      LPC_GPIO2->FIOCLR |=(1<<1);
      delay();
   }
}
void EINT3_IRQHandler(void)
{  LPC_GPIO2->FIOPIN ^=(1<<0);
   LPC_GPIOINT->IO0IntClr |= (1<<0);
}
```

Figure Q5(c)