Universiti Tun Hussein Onn Malaysia

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER I
SESSION 2019/2020

COURSE NAME	:	MICROPROCESSOR AND MICROCONTROLLER
COURSE CODE	:	BEC30403
PROGRAMME CODE	:	BEJ
EXAMINATION DATE	:	DECEMBER 2019 / JANUARY 2020
DURATION	:	3 HOURS
INSTRUCTION	:	ANSWER **ALL** QUESTIONS

THIS QUESTION PAPER CONSISTS OF **SEVEN (7)** PAGES

Q1 (a) Explain your understanding about microprocessor.

(3 marks)

(b) Give **TWO (2)** types of ARM profile and its function.

(4 marks)

(c) Ammar writes 3 lines of coding and when he assembles it, its produces its machine language. Assume that the program memory start at address 0x20000000. Determine the value of PC, when the microprocessor executes

(i) CMP R0,#5
(ii) ADD R1,#10
(iii) MOV R2,#8

Assembly code
CMP R0,#5
ADD R1,#10
MOV R2,#8

Machine languange
2805
F101010A
F04F0208

(3 marks)

(d) "If data from memory are to be processed, they have to be loaded from the memory to a register in the register bank, processed inside the processor, and then written back to the memory if needed". Write lines of coding to show this concept.

(5 marks)

Q2 (a) In writing Assembly Language coding in Keil uVersion, define **THREE (3)** first important lines.

(3 marks)

(b) Illustrate how the data will be allocated at the memory starting at address 0x20000000 based on the coding below.

```
MyData  DCB  0x10, 0x11, 0x12, 0x13
```

(4 marks)

(c) Based on the C code below, write the equivalent assembly language coding.

```
if (A == B)
    C = 5;
else
    C = -5;
```

(5 marks)

(d) Analyses the coding below by giving the purpose on each of instruction.

```
gcd  CMP  R0,R1
     BEQ  end
     BLT  less
     SUB  R0,R0,R1
     B  gcd
less SUB R1,R1,R0
     B gcd
```

(10 marks)

(e) Translate the following conditions into ARM instruction:
"Add registers R3 and R6 only if N is clear. Store the result in register R7"

(8 marks)

Q3 (a) (i) Define the concept of memory map I/O

(2 marks)

(ii) Explain the relation between FIO0DIR0 and FIO0PIN0 with the aid of diagram

(3 marks)

(b) **Figure Q3(b)** shows the coding in C language using case style. Change it using different style of programming.

```
1:#include "mbed.h"
2:BusIn Switch12(p19,p20); //p19 - Switch1, P20 - Switch2
3:BusOut LED_Out(p28,p27,p26,p25,p24,p23,p22,p21);
4:
5:int main() {
6:    while(1) {
7:       switch (Switch12){
8:           case (0x0): LED_Out = 0x00;break;
9:           case (0x1): LED_Out = 0xF0;break;
10:          case (0x2): LED_Out = 0x0F;break;
11:          case (0x3): LED_Out = 0xFF;break;
12:          default: LED_Out = 0x00;break;
13:          }
14:}
```

Figure Q3(b)

(8 marks)

(c) (i) Write a program in assembly language to blink the LED connected at bit 0 at PORT2 for 10 times. The flowchart is shown at **Figure Q3 (c)(i).**

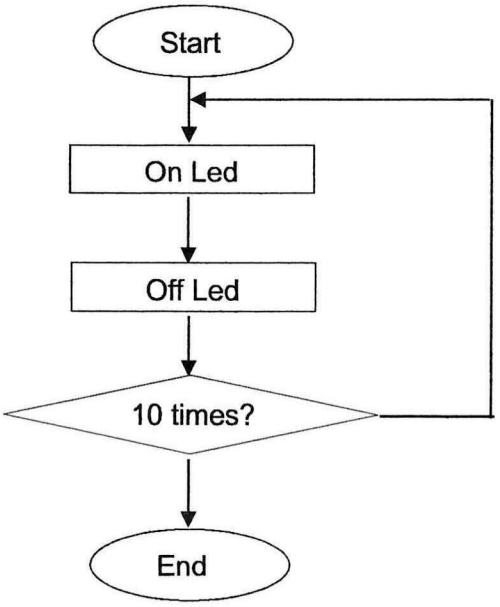

Figure Q3(c)(i)

(9 marks)

(ii) Write C language based on the same cases as **Q3(c)(i)**

(8 marks)

Q4 (a) Differentiate between SPI and I^2C in terms of circuit connection.

(2 marks)

(b) Explain the disadvantage of synchronous over asynchronous

(4 marks)

(c) Write an instruction in C language for SPI communication in the following cases:

(i) To test if data transfer has occurred

(2.5 marks)

(ii) To read the received data

(2.5 marks)

(d) The program in **Listing Q4(d)** demonstrates a simple interrupt application on the LPC1768 using the mbed API. Modify the program such that there are two ISRs, from the same push-button input. One toggles LED1 on a rising interrupt edge, and the other toggles LED2 on a falling edge.

```
#include "mbed.h"
InterruptIn button(p5);
DigitalOut led(LED1);
DigitalOut flash(LED4);
void ISR1() {
     led = !led;
}
int main() {
     button.rise(&ISR1);
     while(1) {
          flash = !flash;
          wait(0.25);
     }
}
```

Listing Q4(d)

(4 marks)

(e) The distance range for the sensor is from 2 cm to 400 cm. A pulse duration of 58 us is equal to 1 cm. Without using the Timer( ) function from the mbed API, write a program to read the distance from the HC-SR04 ultrasonic sensor to the variable "distance".

(10 marks)

– END OF QUESTIONS –

FINAL EXAMINATION

SEMESTER / SESSION : SEM I / 2019/2020
COURSE NAME : MICROPROCESSOR AND MICROCONTROLLER

PROGRAMME CODE: BEJ
COURSE CODE : BEC 30403

Frequently used instruction set summary

Instruction		Description
MOV	<Rd>,<Rm>	Moving data from a register to another register
MOV	<Rd>,<#value>	Moving immediate value into a register
MOVW	<Rd>,<#value>	Move wide (write a 16-bit immediate value to register)
MOVT	<Rd>,<#value>	Move top (write an immediate value to the top half-word of destination reg). The lower 16-bit is unchanged
MVN	<Rd>,<#value>	Moving immediate complement value into a register
STR	<Rd>, [<Rm>]	Writing data from a register into a memory location
STRB	<Rd>, [<Rm>]	Writing data byte from a register into a memory location
STRH	<Rd>, [<Rm>]	Writing half-word data from a register into a memory location
LDR	<Rd>, [<Rm>]	Reading data from memory location to a register
LDRB	<Rd>, [<Rm>]	Reading data byte from memory location to a register
LDRH	<Rd>, [<Rm>]	Reading half-word data from memory location to a register
ADDS	<Rd>, <Rm>, <Op2>	Add two 32-bit numbers and update flags
SUBS	<Rd>, <Rm>, <Op2>	Subtract a 32-bit number from a 32-bit number in destination register and update flags
MUL	<Rd>, <Rn>, <Rm>	Multiply, 32-bit result
SMULL	<RdLo>, <RdHi>, <Rn>, <Rm>	Signed multiply, 64-bit result
UMULL	<RdLo>, <RdHi>, <Rn>, <Rm>	Unsigned multiply, 64-bit result
SDIV	<Rd>, <Rn>, <Rm>	Performs a signed integer division of the value in Rn by the value in Rm
UDIV	<Rd>, <Rn>, <Rm>	Performs an unsigned integer division of the value in Rn by the value in Rm
AND	<Rd>, <Rm>, Op2	AND 32-bit
ORR	<Rd>, <Rm>, Op2	OR 32-bit
EOR	<Rd>, <Rm>, Op2	XOR 32-bit
BIC	<Rd>, <Rm>, Op2	AND 32-bit Rm with complement value of Op2
ORN	<Rd>, <Rm>, Op2	OR 32-bit Rm with complement value of Op2
ASR	<Rd>, <Rm>, Op2	Arithmetic shift right Rm by Op2 times
LSL	<Rd>, <Rm>, Op2	Logical shift left Rm by Op2 times
LSR	<Rd>, <Rm>, Op2	Logical shift right Rm by Op2 times
ROR	<Rd>, <Rm>, Op2	Rotate right Rm by Op2 times
RRX	<Rd>, <Rm>	Rotate right with extend
B		Unconditional branch
NOP		No operation
PUSH	{<Rm>}	Push the content of Rm into the top of the stack
POP	{<Rd>}	Pop the value from the top of the stack to Rd
BL		Call subroutine
BX		Return from subroutine

FINAL EXAMINATION

SEMESTER / SESSION : SEM I / 2019/2020
COURSE NAME : MICROPROCESSOR AND MICROCONTROLLER
PROGRAMME CODE: BEJ
COURSE CODE : BEC 30403

Condition for Branches or Other Conditions Operations

Symbol	Condition	Flag
EQ	Equal	Z set
NE	Not equal	Z clear
CS/HS	Carry set/unsigned higher or same	C set
CC/LO	Carry clear/unsigned lower	C clear
MI	Minus/negative	N set
PL	Plus/positive or zero	N clear
VS	Overflow	V set
VC	No overflow	V clear
HI	Unsigned higher	C set and Z clear
LS	Unsigned lower or same	C clear or Z set
GE	Signed greater than or equal	N set and V set, or N clear and V clear (N == V)
LT	Signed less than	N set and V clear, or N clear and V set (N != V)
GT	Signed greater than	Z clear, and either N set and V set, or N clear and V clear (Z == 0, N == V)
LE	Signed less than or equal	Z set, or N set and V clear, or N clear and V set (Z == 1 or N != V)
AL	Always (unconditional)	—

GPIO port Direction control byte and half-word accessible register description

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxDIR0	Fast GPIO Port x Direction control register 0. Bit 0 in FIOxDIR0 register corresponds to pin Px.0 … bit 7 to pin Px.7.	8 (byte) R/W	0x00	FIO0DIR0 - 0x2009 C000 FIO1DIR0 - 0x2009 C020 FIO2DIR0 - 0x2009 C040 FIO3DIR0 - 0x2009 C060 FIO4DIR0 - 0x2009 C080

GPIO Port pin value byte and half-word accessible register description

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxPIN0	Fast GPIO Port x Pin value register 0. Bit 0 in FIOxPIN0 register corresponds to pin Px.0 … bit 7 to pin Px.7.	8 (byte) R/W	0x00	FIO0PIN0 - 0x2009 C014 FIO1PIN0 - 0x2009 C034 FIO2PIN0 - 0x2009 C054 FIO3PIN0 - 0x2009 C074 FIO4PIN0 - 0x2009 C094