MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL
(A constituent unit of MAHE, Manipal)

V SEMESTER B.TECH (IT) INTERNAL EXAMINATIONS NOV 2021

In-Semester (Online)

SUBJECT: EMBEDDED SYSTEMS [ICT 3158]

Date of Exam: **18/11/2021** Time of Exam: **4.00 PM - 5.20 PM** Max. Marks: **20**

Instructions to Candidates:

- ❖ Answer ALL the questions
- ❖ All the questions are pertaining to LPC1768 microcontroller.
- ❖ Upload the single PDF file of your answer booklet

1. Explain the following instructions with an example for each:

(a) MLS (b) TST (c) ORN

a. MLS:

Instruction Description :

Multiply with subtract

MLS{Cond} Rd, Rn, Rm, Ra; (Description of each entities)

Rd = Ra – Rn*Rm 0.5 Mark

Example:

Rn = 0X00 00 00 02

Rm = 0X00 00 00 03

Ra = 0X00 00 00 0A

Rd = 0X00 00 00 04

0.5 Mark

b. TST:

Instruction Description :

Test bits **3**

TST{cond} Rn, Operand2; (Description of each entities)

0.5

Mark

Example:

TST R0, #0x3F8 ; Perform bitwise AND of R0 value to 0x3F8; APSR is updated but result is discarded

0.5 Mark

c. ORN:

Instruction Description :

Logical OR NOT

op{S}{cond} {Rd,} Rn, Operand2; (Description of each entities)

0.5 Mark

Example:

ORN R7, R11, R14, ROR #4; R7 = R11 OR (NOT(R14 ROR #4))

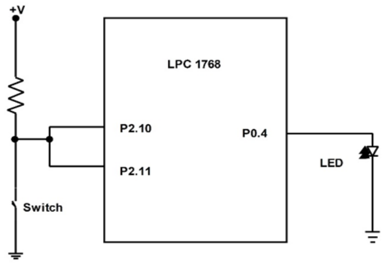
0.5 Mark

2. Write an embedded C program using timer interrupt to generate a square waveform of frequency 100 kHz and duty cycle 75% on P2.3 using TIMER-0 (PCLK = 3 MHz)

```
#include<stdio.h>
#include<LPC17xx.h>                                                    3
unsigned char flag=1;
void TIMER0_IRQHandler(void)
{
        if(flag)
        {
                flag=0;
                LPC_TIM0->TCR = 0x00000002;     // Timer0 Reset
                LPC_GPIO2->FIOCLR=0x00000008;
                LPC_TIM0->MR0 = 7;
                LPC_TIM0->TCR = 0x00000001;     // Timer0 Enable
        }
        else
                {
                flag=1;
                LPC_TIM0->TCR = 0x00000002;     // Timer0 Reset
                LPC_GPIO2->FIOSET=0x00000008;
                LPC_TIM0->MR0 = 22;
                LPC_TIM0->TCR = 0x00000001;     // Timer0 Enable
        }
                LPC_TIM0->IR = 1;
        }
void init_timer0(void)
{
        LPC_TIM0->TCR = 0x00000002;     // Timer0 Reset
        LPC_TIM0->CTCR =0x00;
        LPC_TIM0->MR0 = 22;
        LPC_TIM0->EMR = 0X30;
        LPC_TIM0->PR = 0;
        LPC_TIM0->MCR = 0x00000005;
        LPC_TIM0->TCR = 0x00000001;     // Timer0 Enable
        LPC_GPIO2->FIOSET=0x00000008;
        return;
        }


int main(void)
{
        LPC_GPIO2->FIODIR=0x00000008;
        init_timer0();
        NVIC_EnableIRQ(TIMER0_IRQn);
        while(1);}
```

Main-0.5, Timer init 1.5, ISS - 1

3. For the connections shown below, write an embedded C program using GPIO interrupt to turn ON the LED whenever the switch is pressed and turn OFF the LED whenever the switch is released.



```
#include<LPC17xx.h>
```

```
unsigned int x,y;
void EINT3_IRQHandler (void)
{
}
x = (LPC_GPIOINT->IO2IntStatR)>>10;
if (x== 0x01)
LPC_GPIO0->FIOCLR = 0x04;
y = (LPC_GPIOINT->IO2IntStatF)>>10;
if (y== 0x02)
LPC_GPIO0->FIOSET = 0x04;

LPC_GPIOINT->IO2IntClr = 0x03<<10;
}
void main(void)
{
LPC_PINCON -> PINSEL4 = (1<<20) | (1<<22) ;
LPC_GPIO0 ->FIODIR = 0x10;
LPC_GPIOINT->IO2IntEnR=0x01<<10; // P2.10 raising edge
LPC_GPIOINT->IO2IntEnF=0x01<<11; // P2.11 falling edge
NVIC_EnableIRQ(EINT3_IRQn);
while(1);
}
```

Functions – 1.5 each

4. Write an assembly language program to find the product of two single digit BCD numbers available in the code memory and store the BCD result in the data memory.

Loading the data into register from code memory – 1 Mark
Finding the product of two single digit BCD numbers – 1 Mark
Storing the BCD result in the data memory – 1 Mark

3

```
   EXPORT  __Vectors

__Vectors
        DCD  0x40001000     ; stack pointer value when stack is empty
   DCD  Reset_Handler  ; reset vector

   ALIGN

        AREA mycode, CODE, READONLY
        ENTRY
        EXPORT Reset_Handler

Reset_Handler

        LDR R0, =VALUE1   ;pointer to the first value1
        LDR R1,[R0]       ;load the first value into R1 Assuming data is single
digit BCD
        LDR R0,=VALUE2    ;pointer to the second value
        LDR R3, [R0]      ;load  second number into r3
        MUL R6, R1,R3     ;MUL two numbers and store the result in r6
       Logic for BCD Conversion
        LDR R2, =RESULT
        STR R6,[R2]   ; ; CY IS NOT STORED IN MEMORY
```

```
STOP
        B STOP

VALUE1 DCD 0X00000003  ; First BCD digit
VALUE2 DCD 0X00000002 ; Second BCD digit

        AREA data, DATA, READWRITE
RESULT DCD 0
        END
```

5. Assume that output of a square wave generator (Frequency range 0-9 Hz) is connected to P2.12 (EINT-2, Function-1) input. Write an embedded C program using external hardware interrupt to display the frequency of this square waveform on the seven-segment display connected to P0.7-P0.0. **4**

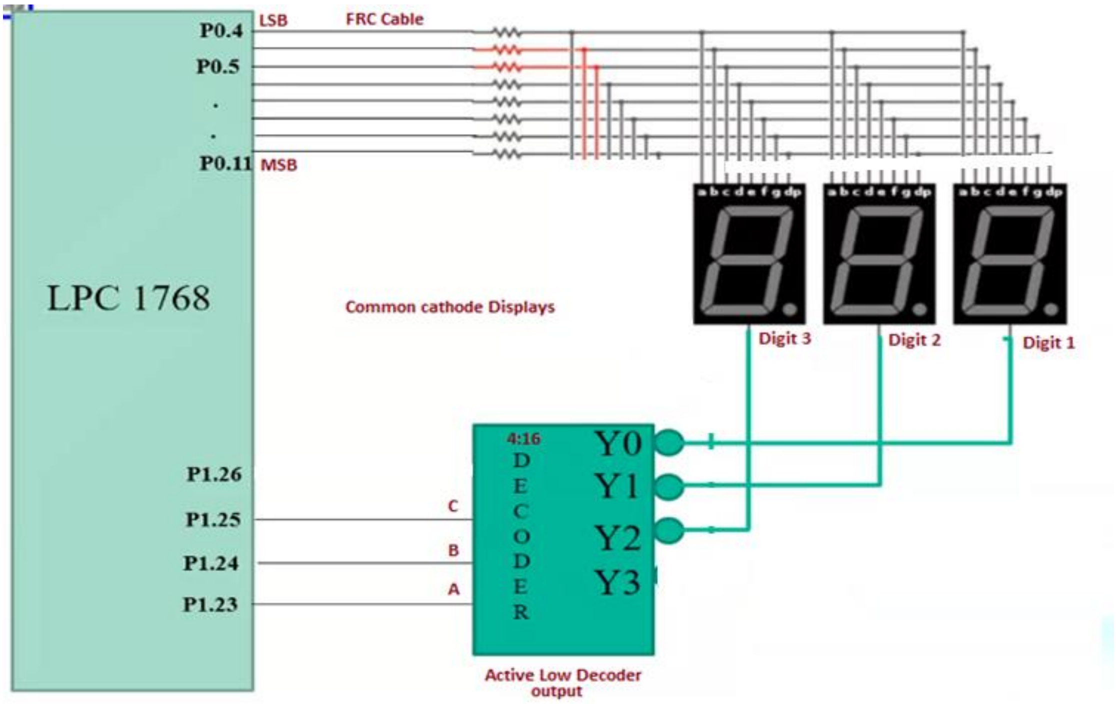
```
#include<LPC17xx.h>
unsigned int count =0;
unsigned char
seven_seg[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
void EINT2_IRQHandler(void)
{
count++;
}
void delay(void)
{
        LPC_TIM0->TCR = 0x00000002;     // Timer0 Reset
        LPC_TIM0->EMR = 0X20;//Set match bit upon match
        LPC_TIM0->PR = 3000; //for 1 ms
        LPC_TIM0->MR0 = 1000;     //for 1 second
        LPC_TIM0->MCR = 0x00000004;     // stop PC and TC on MR0
        LPC_TIM0->TCR = 0x00000001;     // Timer0 Enable
        while ( !(LPC_TIM0->EMR & 0x01)); // wait until match

}
int  main(void)
{
LPC_GPIO0->FIODIR = 0x00000FF;
LPC_PINCON -> PINSEL4 = (1<<24);
LPC_SC ->EXTMODE =0x04;
LPC_SC ->EXTPOLAR = 0x04;
NVIC_EnableIRQ(EINT2_IRQn);
while(1)
         {
          LPC_TIM1->TCR=2;//Reset Counter1
           Delay(); // wait for 1 second
          LPC_GPIO0->FIOPIN = seven_seg[count ] << 4; Counter1 on the seven
segment
count=0;
          }
```

EINT ISS -1, Other functions – 1.5 each

6. With a neat diagram, explain how a 3-digit multiplexed 7 segment display can be interfaced to microcontroller. Write an embedded C program to display 123 on this **4**

display.

Interfacing Diagram: 1 Mark
7 segment Initialization with Pin configuration: 1 Mark
Display Function to display 123 and delay: 2 Mark



```
#include<LPC17xx.h>
#include<stdio.h>

#define FIRST_SEG 0<<23
#define SECOND_SEG 1<<23
#define THIRD_SEG 2<<23

unsigned int dig count;
unsigned int digit_value = (0, 3, 2, 1)
unsigned int select_segment = (0, 0 << 23. 1<<23. 2<<23);
unsigned char seven_seg[3]={0x06, 0x5B, 0x4F};
 unsigned long int temp1, temp2 ,i=0;

void Display(void);
void delay(void);
int main(void) ;
Systemlnit();
SystemCoreClockUpdate();

LPC _PINCON->PINSELO = 0; P0.4 to P0.11 GPIO data lines
LPC PINCON•>PINSEL3 = 0;    P1.23 to P1.26 GPIO enable lines
LPC_GP100•>FIODIR = Ox00000FFO; P0.4 to P0.11 output
LPC_GP101->FIODIR = 0x07800000; HP1.23 to P1.26 output

while(1)
{
delay();
dig_count +=1;
if(dig_count == 0x04)
dig_count = Ox01:
Display()
} //end of while(1)
}//end of main

void Display(void) //To Display on 7•segmenb
{
LPC_GP101•>F1OPIN = select_segment[dig_count];
LPC_GP100•>FIOPIN = seven_seg_digit_value[dig_count]] << 4;
for(i=0;i<500;i++);
LPC_GP100->FIOCLR = Ox00000FFO;
}
void delay(void)
{
 for 1=0;i<500;i++);
}

void delay(void)
{
for i=0;i<500;i++);
if(count ==N)
{
flag = OxFF;
count = 0;
}
else count += 1;


if(flag == 0XFF)
{
 Flag = 0;
Digit_value[1] ==3;
Digit_value[2] ==2;
Digit_value[3] ==1;

}}
```

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

V SEMESTER B.TECH (IT) INTERNAL EXAMINATIONS NOV 2021

In-Semester (Online)

SUBJECT: EMBEDDED SYSTEMS [ICT 3158]

Date of Exam: **18/11/2021** Time of Exam: **4.00 PM - 5.20 PM** Max. Marks: **20**

Instructions to Candidates:

- Answer ALL the questions
- All the questions are pertaining to LPC1768 microcontroller.
- Upload the single PDF file of your answer booklet

1. Explain the following instructions with an example for each: **3**

(a) SMULL (b) TEQ (c) RRX

a. SMULL:

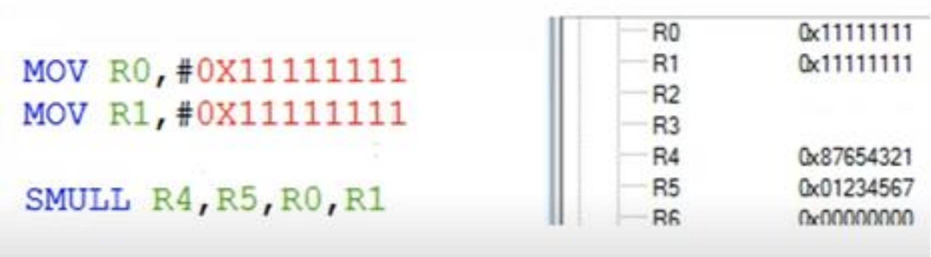
Instruction Description :

Signed Long Multiply

SMULL{cond}{S} RdLo, RdHi, Rm, Rs

The SMULL instruction interprets the values from Rs and Rs as two's complement signed integers. It multiplies these integers and places the least significant 32 bits of the result in RdLo, and the most significant 32 bits of the result in RdHi.

0.5 Mark



```
MOV R0,#0X11111111
MOV R1,#0X11111111

SMULL R4,R5,R0,R1
```

```
R0      0x11111111
R1      0x11111111
R2
R3
R4      0x87654321
R5      0x01234567
R6      0x00000000
```

b. TEQ:

Instruction Description :

Test Equivalence

TEQ{cond} Rn, Operand2 (Instruction Description)

0.5 Mark

Example:

TEQEQ R10, R9 ; Conditionally test if value in R10 is equal to; value in R9, APSR is updated but result is discarded.

0.5 Mark

c. RRX:

Instruction Description :

Rotate Right Extended by 1 bit

0.5 Mark

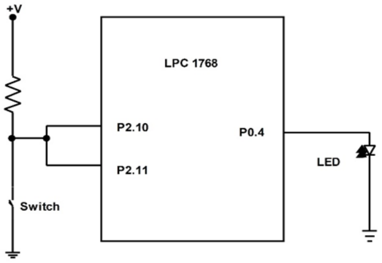
Example: 0.5 Mark

2. Assume that output of a square wave generator is connected to P1.29(CAP 1.1, Function-3). Write an embedded C program to generate a square waveform on the P1.25 (MAT 1.1, Function-3) whose frequency is one fourth of the frequency of the square wave input at P1.29. **3**

```
#include<stdio.h>
#include<LPC17xx.h>
void init_timer1(void)
{
        LPC_PINCON->PINSEL3 |=(3<<18 | 3<<26);//   MAT 1.1(P1.25) and CAP
1.1 (P1.29)
        LPC_TIM1->TCR=2;//Reset Counter1
        LPC_TIM1->CTCR = 0x5; // Counter at +ve edge of CAP1.1
        LPC_TIM1->MR1=0x01; //To count 2 clock pulses in half cycle
        LPC_TIM1->MCR=0x10;//Clear TC upon Match1
        LPC_TIM1->EMR=0xC0;//Toggle EM1 upon Match
        LPC_TIM1->TCR=1;//Start Counter1
}
int  main(void)
{
        init_timer1();
        while(1);
}
```

(MR Value -1, Program 2)

3. For the connections shown below, write an embedded C program using external hardware interrupt to turn ON the LED whenever the switch is pressed and turn OFF the LED whenever the switch is released.



```
#include<LPC17xx.h>

void EINT0_IRQHandler(void)
{
LPC_GPIO0 ->FIOSET = 0x10;
LPC_SC -> EXTINT = 0x01;
}
```

3

```
void EINT1_IRQHandler(void)
{
LPC_GPIO0 ->FIOCLR = 0x10;
LPC_SC -> EXTINT = 0x2;

}


void main(void)
{
LPC_PINCON -> PINSEL4 = (1<<20) | (1<<22) ;
LPC_GPIO0 ->FIODIR = 0x10;
LPC_SC ->EXTMODE =0x03;
LPC_SC ->EXTPOLAR = 0x02;
NVIC_EnableIRQ(EINT0_IRQn);
NVIC_EnableIRQ(EINT1_IRQn);

while(1);}
```

Main -2, Functions – 1each

4. Write an assembly language program to find the sum of all the digits of a 8-digit BCD number available in the code memory and store the BCD result in the data memory.

Loading the data into register from code memory – 0.5 Mark
find the sum of all the digits of a 8-digit BCD number – 1.5 Mark
Storing the BCD result in the data memory – 1 Mark

```
        AREA    RESET, DATA, READONLY
   EXPORT  __Vectors

__Vectors
        DCD  0x100000FF     ; stack pointer value when stack is empty
   DCD  Reset_Handler  ; reset vector

   ALIGN

        AREA mycode, CODE, READONLY
        EXPORT Reset_Handler
        ENTRY
Reset_Handler
        LDR R3, = NUM
        LDR R1, [R3]
MOV R5, R1
MOV R4, #8
MOV R7, #0 ; STORE SUM
LP:
AND R5, #0X0F
ADD R7,R5
CMP R7, #0X0A
BNE NT
ADD R7, 0X06
NX:
LSR R6, #4 (if left shift then see logic as per that)
MOV R5, R6
CMP R5, #0 OR IMPLEMENT COUNTER
BNE LP
LDR R8, #BCDSUM
STR R8, [R7]
STOP B STOP

NUM DCD 0X123456789

AREA data, DATA, READWRITE
BCDSUM DCD 0
```

3

END

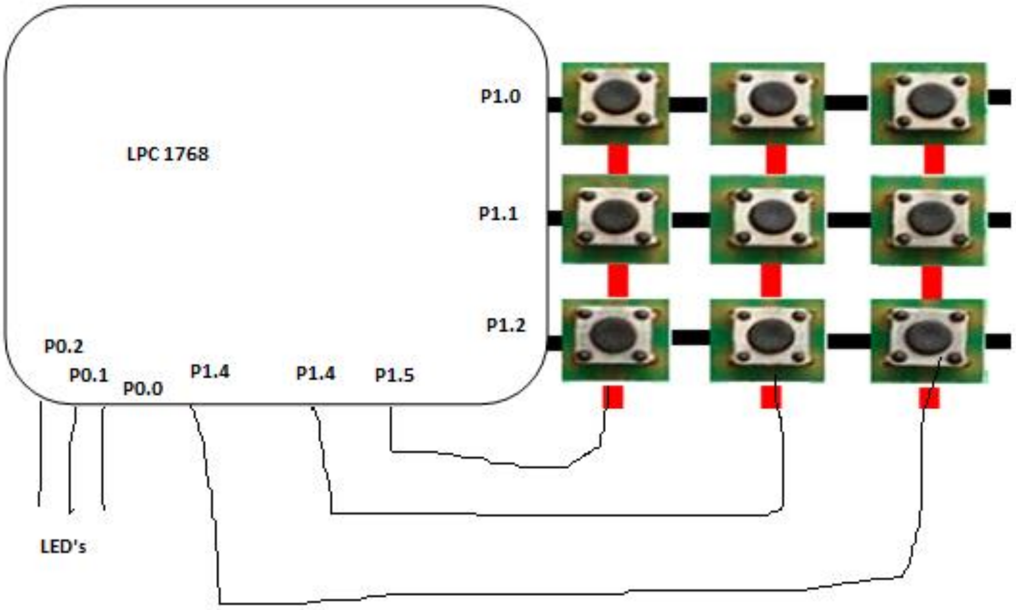
5. Assume that output of a square wave generator is connected to P2.12 input. Write an embedded C program using GPIO interrupt to generate a square waveform at P0.4 whose frequency is 0.125 times the frequency of the input square waveform at P2.12. **4**

```
#include<LPC17xx.h>
unsigned int x;
void EINT3_IRQHandler (void)
{
}
x ++
if (x==4) // for frequency 1/8
{
x=0;
LPC_GPIO0->FIOPIN = ~ (LPC_GPIO0->FIOPIN & 1<<4);
}
LPC_GPIOINT->IO2IntClr = 1<<12;
}
void main(void)
{
LPC_GPIO0 ->FIODIR = 1<<4;
LPC_GPIOINT->IO2IntEnR=1<<12; // P2.12 raising edge
NVIC_EnableIRQ(EINT3_IRQn);
while(1);
}
```

Functions – 2each

6. With a neat diagram, explain how a 3x3 matrix keyboard can be interfaced to microcontroller. Write an embedded C program to display the keycode of the key pressed on the LEDs connected to P0.2-P0.0 **4**

Interfacing Diagram: 1 Mark
LPC Pin configuration : 0.5 Mark
Polling the key pressed with identification : 1+1.5 Mark



```
  Main(void)
{
Initialization for P1.0 to P1.5 (Key Pad:GPIO);
Initialization for P0.0 to P0.2 (LED's GPIO);

Setting Direction Port1 : Input (Key Pad);
Setting Direction Port0 : output (LED);

Int flag, row;

While(1)
{
 For row = 0; row<3; row++)
{
Making each row high one after other;
Flag = 0 ;
Scan();
If(flag = 1)
Break;
}

If(flag = 1)
{
Keypress = 3*row + col;
LEDdisplay(Keypress);
}}}

Voidscan()
{
X = LPC_GPIO1→ FIOPIN;
X = x&07;
If(x!=0)
{
Flag=1;
Using switch case finding the column;
}

Void LEDdisplay(Keypress)
{
Based on keypressvalues

Using if statement or switch and LPC_GPIO→FIOSET enable the LED’s

}
```

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

V SEMESTER B.TECH (IT) INTERNAL EXAMINATIONS NOV 2021

In-Semester (Online)

SUBJECT: EMBEDDED SYSTEMS [ICT 3158]

Date of Exam: **18/11/2021** Time of Exam: **4.00 PM - 5.20 PM** Max. Marks: **20**

Instructions to Candidates:

- Answer ALL the questions
- All the questions are pertaining to LPC1768 microcontroller.
- Upload the single PDF file of your answer booklet

1. Explain the following instructions with an example for each:
(a) SMLAL (b) BGE (c) CMN

a. SMLAL:
Instruction Description :
Signed Long Multiply
SMLAL{cond}{S} RdLo, RdHi, Rm, Rs
Signed multiply and accumulate long
0.5 Mark

Example:

SMLAL R1,R0,R2,R3
(R0:R1) = (R0:R1) + R*R3
0.5 Mark **3**

b. BGE:
Instruction Description :
Greater than or Equal; Singed Integer comparison gave greater or equal
0.5 Mark
Example: 0.5 Mark

c. CMN:
Instruction Description :
Compare Negative
CMN{cond} Rn, Operand2; (Description of Entities)
0.5 Mark
Example: CMN R0, #12 0.5 Mark

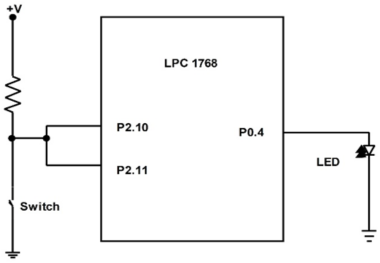
2. Write an embedded C program using timer interrupt to generate a square waveform of frequency 1 kHz and duty cycle 67% on P2.6 using TIMER-1 (PCLK = 6 MHz) **3**

```
#include<stdio.h>
#include<LPC17xx.h>
unsigned char flag=1;
void TIMER1_IRQHandler(void)
{

	if(flag)
	{
		flag=0;
		LPC_TIM1->TCR = 0x00000002;	// Timer1 Reset
		LPC_GPIO2->FIOCLR=1<<6;
		LPC_TIM1->MR0 = 1980;
		LPC_TIM1->TCR = 0x00000001;	// Timer1 Enable
```

```
        }
        else
                {
                flag=1;
                LPC_TIM1->TCR = 0x00000002;     // Timer1 Reset
                LPC_GPIO2->FIOSET=1<<6;
                LPC_TIM1->MR0 = 4020;
                LPC_TIM1->TCR = 0x00000001;     // Timer1 Enable
        }
                LPC_TIM1->IR = 1;
        }
void init_timer1(void)
{

        LPC_TIM1->TCR = 0x00000002;     // Timer1 Reset
        LPC_TIM1->CTCR =0x00;
        LPC_TIM1->MR0 = 4020;
        LPC_TIM1->EMR = 0X30;
        LPC_TIM1->PR = 0;
        LPC_TIM1->MCR = 0x00000005;
        LPC_TIM1->TCR = 0x00000001;     // Timer1 Enable
        LPC_GPIO2->FIOSET=1<<6;
        return;
        }


int main(void)
{
        LPC_GPIO2->FIODIR=1<<6;
        init_timer1();
        NVIC_EnableIRQ(TIMER1_IRQn);
        while(1);}
```

Main-0.5, Timer init 1.5, ISS - 1

3. For the connections shown below, write an embedded C program using GPIO interrupt to turn ON the LED after pressing and releasing the Switch FOUR times.



3

```
#include<LPC17xx.h>
unsigned int x, count1, count2, y;
void EINT3_IRQHandler (void)
{
}
x = (LPC_GPIOINT->IO2IntStatR)>>10;
y = (LPC_GPIOINT->IO2IntStatF)>>10;
if (x== 0x01)
count1++;
```

```
if (y== 0x02)
count2++;
if (count1==4 && count2 ==4)
LPC_GPIO0->SET = 0x04;

LPC_GPIOINT->IO2IntClr = 0x03<<10;
}
void main(void)
{
LPC_PINCON -> PINSEL4 = (1<<20) | (1<<22) ;
LPC_GPIO0 ->FIODIR = 0x10;
LPC_GPIOINT->IO2IntEnR=0x01<<10; // P2.10 raising edge
LPC_GPIOINT->IO2IntEnF=0x01<<11; // P2.11 falling edge
NVIC_EnableIRQ(EINT3_IRQn);
while(1);
}
```

Functions – 1.5 each

4. Write an assembly language program to find the sum of all the bits of a 8-digit BCD number available in the code memory and store the BCD result in the data memory.

Loading the data into register from code memory and find the sum of all the bits of a 8-digit BCD number – 2 Mark (0.5 + 1.5)

Storing the BCD result in the data memory – 1 Mark

3

```
AREA    RESET, DATA, READONLY
   EXPORT  __Vectors

__Vectors
         DCD  0x100000FF     ; stack pointer value when stack is empty
   DCD  Reset_Handler  ; reset vector

   ALIGN

         AREA mycode, CODE, READONLY
         EXPORT Reset_Handler
         ENTRY
Reset_Handler
         LDR R3, = NUM
         LDR R1, [R3]
MOV R5, R1
MOV R4, #0x20 ; count 32-bit
MOV R7, #0 ; STORE SUM
LP:
AND R5, #0X01
ADD R7,R5
```

```
CMP R7, #0X0A
BNE NT
ADD R7, 0X06
NT:
LSR R6, #1 (if left shift then see logic as per that)
MOV R5, R6
SUB R4, #0x01
CMP R4, #00
BNE LP
LDR R8, #BCDbitSUM
STR R8, [R7]
STOP B STOP

NUM DCD 0X123456789

AREA data, DATA, READWRITE
BCDbitSUM DCD 0

        END
```

5. Assume that output of a square wave generator with 50% duty cycle is connected to P2.12 (EINT2, Function-1). Write an embedded C program using external hardware interrupt to generate a square waveform on P0.4 with frequency one eighth of the frequency of the input square waveform at P2.12 and duty cycle 75%. **4**

```
#include<LPC17xx.h>
unsigned int count =0;
void EINT2_IRQHandler(void)
{
count++;
if (count==6)
LPC_GPIO0->FIOCLR = 1<<4;
if(count==8)
{
LPC_GPIO0->FIOSET = 1<<4;
count=0;
}
LPC_SC ->EXTINT = 0x04;
}
int  main(void)
{
LPC_GPIO0->FIODIR = 1<<4;
LPC_PINCON -> PINSEL4 = (1<<24);
LPC_SC ->EXTMODE =0x04;
LPC_SC ->EXTPOLAR = 0x04;
NVIC_EnableIRQ(EINT2_IRQn);
LPC_GPIO0->FIOSET = 1<<4;
while(1);
        }
```

EINT ISS -2, Main-2

6. With a neat diagram, explain how a 16x2 LCD can be interfaced to the microcontroller. Write an embedded C program to display the message “Best Wishes” **4**

on the LCD.

Interfacing Diagram:1 Mark

LPC Pin configuration with LCD configuration (Control and Data Register)
: 1.5 Mark

Port Write Function and Delay : 1+0.5 Mark

```
#include <lpc17xx.h>
#define RS  27 //P0.27
#define EN  28  //P0.28
#define DT  23  //P0.23 to P0.26 data lines

 unsigned long int temp1=0, temp2=0,i,j ;
 unsigned char flag1 =0, flag2 =0;
 unsigned char msg[] = {" Best Wishes "}; //As message is written in codes they are stored in
ASCII values

void lcd_write(void);
void port_write(void);
void delay_lcd(unsigned int);
unsigned long int init_command[] = {0x30,0x30,0x30,0x20,0x28,0x0c,0x06,0x01,0x80};
int main(void)
 {
        SystemInit();
        SystemCoreClockUpdate();
           LPC_GPIO0->FIODIR = 1<<RS|1<<EN|0XF<<DT;   //used to make all pins
output
            flag1 =0;     // flag1 = 0 all are command and flag1 = 1 all are data
         for (i=0; i<9;i++)
             {
          temp1 = init_command[i];
          lcd_write();
             }
            flag1 =1;
        i =0;
        while (msg[i] != '\0')
              {
               temp1 = msg[i]; // char by char
               lcd_write();
                     i+= 1;

if(i==16)  //check for 1 charactres in first line
                                                                          {

         flag1=0;  //if yes

         temp1=0xc0; //configure second line in command register

         lcd_write();

         flag1=1;
                                                                          }

              }
        while(1);
 }

  void lcd_write(void)
               {
                    flag2 =  (flag1 == 1) ? 0 :((temp1 == 0x30) || (temp1 == 0x20)) ? 1 : 0;
               temp2 = temp1 & 0xf0;//move data (26-8+1) times : 26 - HN place, 4 - Bits to
extract MSB and then LSB as nedd to send 4 bit at a time
                                                                                 temp2=temp2>>4;
            temp2 = temp2 << DT;//data lines from 23 to 26
            port_write();
                 if (!flag2)
                   {
               temp2 = temp1 & 0x0f; //26-4+1
               temp2 = temp2 << DT;
               port_write();
                  }
                }



 void port_write(void)
 {
            LPC_GPIO0->FIOPIN = 0;
            LPC_GPIO0->FIOPIN = temp2;
         if (flag1 == 0)
                 LPC_GPIO0->FIOCLR = 1<<RS;
         else
                        LPC_GPIO0->FIOSET = 1<<RS;

            LPC_GPIO0->FIOSET = 1<<EN;          //this and below 3 lines are used  give pulse
for enable and wait for some time interval
            delay_lcd(25);
            LPC_GPIO0->FIOCLR = 1<<EN;
                 delay_lcd(30000);                              // 3 ms  highest delay

  }

void delay_lcd(unsigned int r1)
 {
            unsigned int r;
            for(r=0;r<r1;r++);

    return;
 }
```