**PART-A**

1)Write a program to multiply two 16-bit binary numbers.

PROGRAM:-

AREA MUL,CODE,READONLY

ENTRY

LDRH R1,N1

LDRH R2,N2

MUL R3,R1,R2

LDR R0,PRODUCT

STR R3,[R0]

B1 B B1

N1 DCW 5

N2 DCW 6

PRODUCT DCD 0X40000000

END2) Write a program to find the sum of first 10 integer numbers.

PROGRAM:-

AREA SUM,CODE,READONLY

ENTRY

MOV R0,#10

MOV R1,#1

MOV R2,#1

NEXT ADD R1,R1,R2

ADD R2,#1

SUBS R0,#1

BNE NEXT

LDR R3,RES

STR R1,[R3]

B1 B B1

RES DCD 0X40000000

END3)Write a program to find a factorial of a number.

0! = 1

1! = 1

2! = 2X1 = 2

4! = 4X3X2X1 = 24d = 0X18

PROGRAM:-

AREA FACTORIAL, CODE, READONLY

ENTRY

LDR R0, MEMORY

LDRB R1, [R0]

MOV R2, #1

CMP R1,#0

BEQ STORE

MOV R2, R1

UP ADD R1, R1, #-1

CMP R1, #0

BEQ STORE

MUL R3, R2, R1

MOV R2, R3

B UP

STORE LDR R0, RESULT

STR R2,[R0]

HERE B HERE

MEMORY DCD 0X40000000

RESULT DCD 0X40000010

END

4)Write a program to add an array of 16-bit has and store the 32-bit results in internal ram.

PROGRAM:-

AREA SUM, CODE, READONLY

ENTRY

LDR RO,MEMORY

MOV R1,#4

LDRH R2,[R0]

ADD R1,#-1

UP ADD R0,R0 #2

LDRH R3,[R0]

ADD R2,R3,R2

ADD R1,#-1

CMP R1,#0

BNE UP

LDR R0,RESULT

STR R2,[R0]

HERE B HERE

MEMORY DCD 0X40000000

RESULT DCD 0X40000010

END

5)Write a program to find the square of a number 1 to 10 using LOOKUP.

PROGRAM:-

AREA SQR,CODE,READONLY

ENTRY

LDR R0,=LOOKUP

LDRH R1,VALUE

LDR R2,=RESULT

MOV R1,R1,LSL#01

ADD R0,R0,R1

LDRH R3,[R2]

HERE B HERE

AREA SRC,DATA,READONLY

LOOKUP DCW 0,1,4,9,16,25,36,49,64,81,100

VALUE DCD 4

AREA DST,DATA,READWRITE

RESULT DCD 0X40000020

END 6)Write a program to find the largest/smallest number in an array of 32 numbers.

a)Largest

PROGRAM:-

AREA LARGEST,CODE,READONLY

ENTRY

MOV R5,#5

LDR R0,=A

LDR R2,[R0]

NEXT ADD R0,R0,#4

LDR R3,[R0]

CMP R2,R3

BHS Large

MOV R2,R3

Large SUBS R5,#1

BNE NEXT

LDR R1,=RES

STR R2,[R1]

HERE B HERE

AREA SRC,DATA,READONLY

A DCD 25,33,85,99,59,44

ALIGN

AREA DST,DATA,READWRITE

RES DCD 0x0

ALIGN

END b)Smallest

PROGRAM:-

AREA SMALLEST,CODE,READONLY

ENTRY

MOV R5,#5

LDR R0,=A

LDR R2,[R0]

NEXT ADD R0,R0,#4

LDR R3,[R0]

CMP R2,R3

BLS Large

MOV R2,R3

Small SUBS R5,#1

BNE NEXT

LDR R1,=RES

STR R2,[R1]

HERE B HERE

AREA SRC,DATA,READONLY

A DCD 25,33,85,99,59,44

ALIGN

AREA DST,DATA,READWRITE

RES DCD 0x0

ALIGN

END

7) Write a program to arrange the series of 32-bit numbers in ascending or descending order

a)Ascending order

PROGRAM:-

AREA ASCENDING,CODE, READONLY

ENTRY

MOV R8, #4

LDR R2, =cvalue

LDR R3, =dvalue

LOOP0 LDR R1, [R2], #4

STR R1, [R3], #4

SUBS R8, R8, #1

CMP R8, #0

BNE LOOP0

START1 MOV R5, #3

MOV R7, #0

LDR R1, =dvalue

LOOP LDR R2, [R1], #4

LDR R3, [R1]

CMP R2, R3

BGT LOOP2

STR R2, [R1], #-4

STR R3, [R1]

MOV R7, #1

ADD R1, #4

LOOP2 SUBS R5, R5, #1

CMP R5, #0

BNE LOOP

CMP R7, #0

BNE START1

STOP B STOP

cvalue

DCD 0X44444444

DCD 0X11111111

DCD 0X33333333

DCD 0X22222222

ALIGN

AREA DATA1, DATA, READWRITE

dvalue DCD 0X00000000

ALIGN

END

b)Desending order

PROGRAM:-

AREA ASCENDING,CODE, READONLY

ENTRY

MOV R8, #4

LDR R2, =cvalue

LDR R3, =dvalue

LOOP0 LDR R1, [R2], #4

STR R1, [R3], #4

SUBS R8, R8, #1

CMP R8, #0

BNE LOOP0

START1 MOV R5, #3

MOV R7, #0

LDR R1, =dvalue

LOOP LDR R2, [R1], #4

LDR R3, [R1]

CMP R2, R3

BLT LOOP2

STR R2, [R1], #-4

STR R3, [R1]

MOV R7, #1

ADD R1, #4

LOOP2 SUBS R5, R5, #1

CMP R5, #0

BNE LOOP

CMP R7, #0

BNE START1

STOP B STOP

cvalue

DCD 0X44444444

DCD 0X11111111

DCD 0X33333333

DCD 0X22222222

ALIGN

AREA DATA1, DATA, READWRITE

dvalue DCD 0X00000000

ALIGN

END

8) Write a program to count the number of ones and zeroes in two consecutive memory location.

PROGRAM:-

AREA ONEZERO,CODE,READONLY

ENTRY

MOV R2,#0

MOV R3,#0

MOV R7,#2

LDR R6,=VALUE

LOOP MOV R1,#32

LDR R0,[R6],#4

LOOP0 MOVS R0,R0,ROR #1

BHI ONES

ZEROS ADD R3,R3,#1

BNE LOOP1

ONES ADD R2,R3,#1

LOOP1 SUBS R1,R1,#1

BNE LOOP0

SUBS R7,R7,#1

CMP R7,#0

BNE LOOP

STOP B STOP

VALUE DCD 0X11111111,0XAA55AA55

END

**PART-B**

9)Write a program to display “HELLO WORLD” message using internal UART.

PROGRAM:-

#include<lpc214x.h>

char \*msg="hello world\n";

int main()

{

PINSEL0=0X05;

U0LCR=9083;

U0DLL=61;

U0DLM=0X00;

U0LCR=0X03;

while(\*msg!=0x00)

{

while(!(U0LSR &0X20));

U0THR=\*msg;

msg++;

}

}

10)Write a c program to execute interface and control the dc motor.

PROGRAM:-

#include<lpc214x.h>

void delay(unsigned int x)

{

unsigned int i,j;

for(i=0;i<x;i++)

for(j=0;j<1275;j++);

}

int main()

{

IODIR0|=0x00000900;

while(1)

{

IOSET0=0X00000100;

delay(5000);

IOCLR0=0X00000900;

delay(5000);

IOSET0=0X00000800;

delay(5000);

IOCLR0=0X00000900;

delay(5000);

}

}

11)Write a c program to interface a stepper motor and rotate it in clock wise and anti-clock wise direction.

PROGRAM:-

#include<lpc214x.h>

void clock\_wise (void);

void anti\_clock\_wise(void);

unsigned long int var1,var2;

unsigned int i=0,j=0,k=0;

int main()

{

PINSEL0=0X00FFFFFF;

IO0DIR|=0X0000F000;

while(1)

{

for(j=0;j<65;j++)

clock\_wise();

for(k=0;k<65;k++)

for(j=0;j<50;j++)

anti\_clock\_wise();

for(k=0;k<65;k++);

}

}

void clock\_wise(void)

{

var1=0x00000800;

for(i=0;i<=3;i++)

{

var1=var1<<1;

var2=~var1;

var2=var2&0x0000F000;

IO0PIN=~var2;

for(k=0;k<30000;k++);

}

}

void anti\_clock\_wise(void)

{

var1=0x00010000;

for(i=0;i<=3;i++)

{

var1=var1>>1;

var2=~var1;

var2=var2&0x0000F000;

IO0PIN=~var2;

for(k=0;k<30000;k++);

}

}

12)Write a c program to determine digital output for a given analog input using internal ADC of arm controller.

PROGRAM:-

#include<LPC214X.H>

#define RS 0X00400000

#define RW 0X20000000

#define EN 0X10000000

unsigned int RESULT;

float VOLTAGE;

char VOLT[18];

void DELAY(unsigned int X)

{

unsigned int I,J;

for(I=0;I<X;I++)

for(J=0;J<1275;J++);

}

void CMD(char C)

{

IOCLR0=0X00003FC0;

IOSET0=C<<6;

IOCLR0=RW;

IOCLR0=RS;

IOSET0=EN;

DELAY(100);

IOCLR0=EN;

}

void DATA(char C)

{

IOCLR0=0X00003FC0;

IOSET0=C<<6;

IOCLR0=RW;

IOSET0=RS;

IOSET0=EN;

DELAY(100);

IOCLR0=EN;

}

void LCD\_STR(char \*S)

{

while(\*S)

{

DATA(\*S);

S++;

DELAY(20);

}

}

void ADC\_INIT()

{

AD0CR=0X00210308;

PINSEL1=0X10000000;

}

void DISPLAY(unsigned int N)

{

if(N==0)

DATA(N+0X30);

if(N)

{

DISPLAY(N/10);

DATA((N%10)+0X30);

}

}

void INIT()

{

CMD(0X38);

CMD(0X0E);

CMD(0X80);

}

int main()

{

IODIR0|=0X30403FC0;

INIT();

ADC\_INIT();

while(1)

{

CMD(0X01);

while(AD0DR3 &(0X80000000)==0);

RESULT=(AD0DR3 & (0X3FF <<6));

RESULT=RESULT >> 6;

LCD\_STR("ADC:");

CMD(0X84);

DISPLAY(RESULT);

VOLTAGE = ((RESULT/1023.0)\*3.3);

SPRINTF(VOLT,"VOLTAGE:%.2F V",VOLTAGE);

CMD(0XC0);

LCD\_STR(VOLT);

DELAY(1000);

}

}

13)Write a c program to interface a DAC and generate triangular and square waveform.

a)Triangular

PROGRAM:-

#include <LPC21xx.h>

int main ()

{

unsigned long int temp=0x00000000;

unsigned int i=0;

IO0DIR=0x00FF0000;

while(1)

{

for(i=0;i!=0xFF;i++)

{

temp=i;

temp = temp << 16;

IO0PIN=temp;

}

for(i=0xFF; i!=0;i--)

{

temp=i;

temp = temp << 16;

IO0PIN=temp;

}

}

}

b)Square

PROGRAM:-

#include <lpc21xx.h>

void delay(void);

int main ()

{

PINSEL0 = 0x00000000 ;

PINSEL1 = 0x00000000 ;

IO0DIR = 0x00FF0000 ;

while(1)

{

IO0PIN = 0x00000000;

delay();

IO0PIN = 0x00FF0000;

delay();

}

}

void delay(void)

{

unsigned int i=0;

for(i=0;i<=95000;i++);

}

14)Write a c program for 4x4 keyboard and display the key code on an LCD.

PROGRAM:-

#include<lpc21xx.h>

#include<stdio.h>

void lcd\_init(void);

void clr\_disp(void);

void lcd\_com(void);

void lcd\_data(void);

void wr\_cn(void);

void wr\_dn(void);

void scan(void);

void get\_key(void);

void display(void);

void delay(unsigned int);

void init\_port(void);

unsigned long int scan\_code[16]= {0x00EE0000,0x00ED0000,0x00EB0000,0x00E70000,

0x00DE0000,0x00DD0000,0x00DB0000,0x00D70000,

0x00BE0000,0x00BD0000,0x00BB0000,0x00B70000,

0x007E0000,0x007D0000,0x007B0000,0x00770000};

unsigned char ASCII\_CODE[16]= {'0','1','2','3',

'4','5','6','7',

'8','9','A','B',

'C','D','E','F'};

unsigned char row,col;

unsigned char temp,flag,i,result,temp1;

unsigned int r,r1;

unsigned long int var,var1,var2,res1,temp2,temp3,temp4;

unsigned char \*ptr,disp[] = "4X4 KEYPAD";

unsigned char disp0[] = "KEYPAD TESTING";

unsigned char disp1[] = "KEY = ";

int main()

{

init\_port();

delay(3200);

lcd\_init();

delay(3200);

clr\_disp();

delay(500);

ptr = disp;

temp1 = 0x81;

lcd\_com();

delay(800);

while(\*ptr!='\0')

{

temp1 = \*ptr;

lcd\_data();

ptr ++;

}

while(1)

{

get\_key();

display();

}

} void get\_key(void)

{

unsigned int i;

flag = 0x00;

IO1PIN=0x000f0000;

while(1)

{

for(row=0X00;row<0X04;row++) {

if( row == 0X00)

{

temp3=0x00700000;

}

else if(row == 0X01)

{

temp3=0x00B00000;

}

else if(row == 0X02)

{

temp3=0x00D00000;

}

else if(row == 0X03)

{

temp3=0x00E00000;

}

var1 = temp3;

IO1PIN = var1;

IO1CLR =~var1;

scan();

delay(100);

if(flag == 0xff)

break;

}

if(flag == 0xff)

break;

}

for(i=0;i<16;i++)

{

if(scan\_code[i] == res1)

{

result = ASCII\_CODE[i];

break;

}

}

}// end of get\_key();

void scan(void)

{

unsigned long int t;

temp2 = IO1PIN;

temp2 = temp2 & 0x000F0000;

if(temp2 != 0x000F0000)

{

delay(1000);

temp2 = IO1PIN;

temp2 = temp2 & 0x000F0000; //changed condition is same

if(temp2 != 0x000F0000) // store the value in res1

{

flag = 0xff;

res1 = temp2;

t = (temp3 & 0x00F00000); //Verfying Row Write

res1 = res1 | t; //final scan value is stored in res1

}

else

{

flag = 0x00;

}

}

} // end of scan()

void display(void)

{

ptr = disp0;

temp1 = 0x80; // Display starting address of first line

lcd\_com();

while(\*ptr!='\0')

{

temp1 = \*ptr;

lcd\_data();

ptr ++;

}

ptr = disp1;

temp1 = 0xC0; // Display starting address of second line

lcd\_com();

while(\*ptr!='\0')

{

temp1 = \*ptr;

lcd\_data();

ptr ++;

}

temp1 = 0xC6; //display address for key value

lcd\_com();

temp1 = result;

lcd\_data();

}

void lcd\_init (void)

{

temp = 0x30;

wr\_cn();

delay(3200);

temp = 0x30;

wr\_cn();

delay(3200);

temp = 0x30;

wr\_cn();

delay(3200);

temp = 0x20;

wr\_cn();

delay(3200);

// load command for lcd function setting with lcd in 4 bit mode,

// 2 line and 5x7 matrix display

temp = 0x28;

lcd\_com();

delay(3200);

// load a command for display on, cursor on and blinking off

temp1 = 0x0C;

lcd\_com();

delay(800);

// command for cursor increment after data dump

temp1 = 0x06;

lcd\_com();

delay(800);

temp1 = 0x80;

lcd\_com();

delay(800);

}

void lcd\_data(void)

{

temp = temp1 & 0xf0;

wr\_dn();

temp= temp1 & 0x0f;

temp= temp << 4;

wr\_dn();

delay(100);

}

void wr\_dn(void) ////write data reg

{

IO0CLR = 0x000000FC; // clear the port lines.

IO0SET = temp; // Assign the value to the PORT lines

IO0SET = 0x00000004; // set bit RS = 1

IO0SET = 0x00000008; // E=1

delay(10);

IO0CLR = 0x00000008;

}

void lcd\_com(void)

{

temp = temp1 & 0xf0;

wr\_cn();

temp = temp1 & 0x0f;

temp = temp << 4;

wr\_cn();

delay(500);

}

void wr\_cn(void) //write command reg

{

IO0CLR = 0x000000FC; // clear the port lines.

IO0SET = temp; // Assign the value to the PORT lines

IO0CLR = 0x00000004; // clear bit RS = 0

IO0SET = 0x00000008; // E=1

delay(10);

IO0CLR = 0x00000008;

}

void clr\_disp(void)

{

// command to clear lcd display

temp1 = 0x01;

lcd\_com();

delay(500);

}

void delay(unsigned int r1)

{

for(r=0;r<r1;r++);

}

void init\_port()

{

IO0DIR = 0x000000FC; //configure o/p lines for lcd

IO1DIR = 0XFFF0FFFF;

}

15)Display the hex digits 0 to F on a seven segment LED interface, with an appropriate delay in between.

PROGRAM:-

#include<LPC21XX.h>

unsigned int delay;

unsigned int Switchcount=0;

unsigned int Disp[16]={0x003F0000, 0x00060000, 0x005B0000, 0x004F0000, 0x00660000,0x006D0000,

0x007D0000, 0x00070000, 0x007F0000, 0x006F0000, 0x00770000,0x007C0000,

0x00390000, 0x005E0000, 0x00790000, 0x00710000 };

#define SELDISP1 0x10000000

#define SELDISP2 0x20000000

#define SELDISP3 0x40000000

#define SELDISP4 0x80000000

#define ALLDISP 0xF0000000

#define DATAPORT 0x00FF0000

int main (void)

{

PINSEL0 = 0x00000000;

PINSEL1 = 0x00000000;

IO0DIR = 0xF0FF0000;

IO1DIR = 0x00000000;

while(1)

{

IO0SET |= ALLDISP;

IO0CLR = 0x00FF0000;

IO0SET = Disp[Switchcount];

if(!(IO1PIN & 0x00800000))

{

for(delay=0;delay<100000;delay++)

{

if((IO1PIN & 0x00800000))

{

Switchcount++;

if(Switchcount == 0x10)

{

Switchcount = 0;

IO0CLR = 0xF0FF0000;

}

}

}

}

}