**Experiments**

**Introduction**

The LPC2148 microcontrollers is based on a 32/16 bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combines the microcontroller with embedded high speed flash memory ranging from 32 kB to 512 kB. A 128-bit wide memory interface and unique accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30 % with minimal performance penalty.

Due to their tiny size and low power consumption, LPC2148 is ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale. A blend of serial communications interfaces ranging from a USB 2.0 Full Speed device, multiple UARTS, SPI, SSP to I2Cs and on-chip SRAM of 8 kB up to 40 kB, make these devices very well suited for communication gateways and protocol converters, soft modems, voice recognition and low end imaging, providing both large buffer size and high processing power. Various 32-bit timers, single or dual 10-bit ADC(s), 10-bit DAC, PWM channels and 45 fast GPIO lines with up to nine edge or level sensitive external interrupt pins make these microcontrollers particularly suitable for industrial control and medical systems.

**Experiment 11: Write C program to add two 16-bit operands using ARM controller**

**Aim:**

The aim of the experiment is to add two 16-bit operands using LPC2148 ARM controller

**Tools Required:**

1. VSK-2148 ARM7 LPC2148 Syllabus Board – 1
2. Power Card - 1
3. USB cable - 1

**Theory:**

The LPC2148 is a 32-bit ARM (Advanced RISC Microcontroller) controller which is a high speed device. It performs arithmetic and logic operations within few nano seconds. So, it is most preferred solution for embedded applications. In this experiment a ‘C’ program is developed to add two 16-bit operands. The program is developed in IAR Embedded Workbench. The program and the results are stored in the Flash memory and RAM respectively. To view the result, the data is moved to the PC through UART (serial port). The baud rate of the controller is adjusted by the program. The baud rate of the PC is adjusted by the settings in the WinXTalk/Hyper Terminal.

**Procedure**

1. Enter the following program in the IAR embedded workbench and create the executable file (HEX file)
2. Switch on the kit
3. Connect the USB cable and check the COM port in the device manager
4. Keep the SW21 in programming mode (PROG)
5. Download the code using Flash Magic tool
6. Open WinXtalk/Hyper Terminal with 9600 baud rate
7. Keep the SW21 in execution mode (EXE) and press reset
8. The result is displayed in the WinXtalk/Hyper Terminal

**Program**

//Header Files

#include<nxp/iolpc2148.h>

#include<stdio.h>

#define f 12000000 // PCLKFREQ ( controller frequency )

//Function to initialize Serial Port

void serial\_Init(void) // Function to initialise the serial port

{0X00000005;

VPBDIV=0X01;

baudrate = (f/(9600\*16)); // Baudrate setting (9600 bps)

U0LCR=0X80;

U0DLL = baudrate & 0X00FF;

U0DLM=( baudrate >>8) & 0X00FF;

U0LCR=0X03;

}

unsigned int baudrate ;

PINSEL0=

//Function to print the data in the serial port

int putchar(int ch)

{

if (ch == '\n')

{

while (!(U0LSR & 0x20));

U0THR = 0x0d;

}

while (!(U0LSR & 0x20));

return (U0THR = ch);

}

//Main Routine

void main()

{

unsigned int a,b,c;

serial\_Init(); // Initialise the serial port

a = 0xF34C;

b = 0x56EA;

c = a + b;

printf("\n\r Addition of two given 16-bit numbers is %x",c);

while(1);

}

**Result:**

Thus, the ‘C’ program is developed for LPC2148 ARM controller to add two 16-bit operands

**Experiment 12: Write C program to add two 32-bit operands using ARM controller**

**Aim:**

The aim of the experiment is to add two 32-bit operands using ARM controller

**Tools Required:**

1. VSK-2148 ARM7 LPC2148 Syllabus Board – 1
2. Power Card - 1
3. USB cable - 1

**Theory:**

The LPC2148 is a 32-bit ARM (Advanced RISC Microcontroller) controller which is a high speed device. It performs arithmetic and logic operations within few nano seconds. So, it is most preferred solution for embedded applications. In this experiment a ‘C’ program is developed to add two 32-bit operands. The program is developed in IAR Embedded Workbench. The program and the results are stored in the Flash memory and RAM respectively. To view the result, the data is moved to the PC through UART (serial port). The baud rate of the controller is adjusted by the program. The baud rate of the PC is adjusted by the settings in the WinXTalk/Hyper Terminal.

**Procedure**

1. Enter the following program in the IAR embedded workbench and create the executable file (HEX file)
2. Switch on the kit
3. Connect the USB cable and check the COM port in the device manager
4. Keep the SW21 in programming mode (PROG)
5. Download the code using Flash Magic tool
6. Open WinXtalk/Hyper Terminal with 9600 baud rate
7. Keep the SW21 in execution mode (EXE) and press reset
8. The result is displayed in the WinXtalk/Hyper Terminal

**Program**

//Header Files

#include<nxp/iolpc2148.h>

#include<stdio.h>

#define f 12000000 // PCLKFREQ ( controller frequency )

//Function to initialize Serial Port

void serial\_Init(void) // Function to initialise the serial port

{

unsigned int baudrate ;

PINSEL0=0X00000005;

VPBDIV=0X01;

baudrate = (f/(9600\*16)); // Baudrate setting (9600 bps)

U0LCR=0X80;

U0DLL = baudrate & 0X00FF;

U0DLM=( baudrate >>8) & 0X00FF;

U0LCR=0X03;

}

//Function to print the data in the serial port

int putchar(int ch)

{

if (ch == '\n')

{

while (!(U0LSR & 0x20));

U0THR = 0x0d;

}

while (!(U0LSR & 0x20));

return (U0THR = ch);

}

//Main Routine

void main()

{

unsigned int a,b,c;

serial\_Init(); // Initialise the serial port

a = 0x10Ab295C;

b = 0x5366EA9F;

c = a + b;

printf("\n\r Addition of two given 32-bit numbers is %x",c);

while(1);

}

**Result:**

Thus, the ‘C’ program is developed for LPC2148 ARM controller to add two 32-bit operands

**Experiment 13: Write C program to subtract two 16-bit operands using ARM controller**

**Aim:**

The aim of the experiment is to subtract two 32-bit operands using ARM controller

**Tools Required:**

1. VSK-2148 ARM7 LPC2148 Syllabus Board – 1
2. Power Card - 1
3. USB cable - 1

**Theory:**

The LPC2148 is a32-bit advanced microcontroller. It can perform all kind of arithmetic and logic operations. The operating speed of the controller is several times higher. It is much suitable for data processing functions where execution time is important. To view the result, the data is moved to the PC through UART (serial port). The baud rate of the controller is adjusted by the program. The baud rate of the PC is adjusted by the settings in the WinXTalk/Hyper Terminal. Keep same baud rates for the controller and PC.

**Procedure**

1. Enter the following program in the IAR embedded workbench and create the executable file (HEX file)
2. Switch on the kit
3. Connect the USB cable and check the COM port in the device manager
4. Keep the SW21 in programming mode (PROG)
5. Download the code using Flash Magic tool
6. Open WinXtalk/Hyper Terminal with 9600 baud rate
7. Keep the SW21 in execution mode (EXE) and press reset
8. The result is displayed in the WinXtalk/Hyper Terminal

**Program**

//Header Files

#include<nxp/iolpc2148.h>

#include<stdio.h>

#define f 12000000 // PCLKFREQ ( controller frequency )

//Function to initialize Serial Port

void serial\_Init(void) // Function to initialise the serial port

{

unsigned int baudrate ;

PINSEL0=0X00000005;

VPBDIV=0X01;

baudrate = (f/(9600\*16)); // Baudrate setting (9600 bps)

U0LCR=0X80;

U0DLL = baudrate & 0X00FF;

U0DLM=( baudrate >>8) & 0X00FF;

U0LCR=0X03;

}

//Function to print the data in the serial port

int putchar(int ch)

{

if (ch == '\n')

{

while (!(U0LSR & 0x20));

U0THR = 0x0d;

}

while (!(U0LSR & 0x20));

return (U0THR = ch);

}

//Main Routine

void main()

{

unsigned int a,b,c;

serial\_Init(); // Initialise the serial port

a = 0xFEDB;

b = 0x1234;

c = a - b;

printf("\n\r Subtraction of two given 16-bit numbers is %x",c);

while(1);

}

**Result:**

Thus the ‘C’ program is developed for LPC2148 microcontroller to subtract two 16-bit operands

**Experiment 14: Write C program to multiply two operands using ARM controller**

**Aim:**

The aim of the experiment is to multiply two operands using ARM controller

**Tools Required:**

1. VSK-2148 ARM7 LPC2148 Syllabus Board – 1
2. Power Card - 1
3. USB cable - 1

**Theory:**

The LPC2148 is a 32-bit ARM (Advanced RISC Microcontroller) controller which is a high speed device. This device can perform all kinds of arithmetic and logic operations. The ‘C’ programming language is used to simplify programming effort. The human understandable mnemonics and symbols of the ‘C’ language helps the user to develop application so quickly. To view the result, the data is moved to the PC through UART (serial port). The baud rate of the controller is adjusted by the program. The baud rate of the PC is adjusted by the settings in the WinXTalk/Hyper Terminal.

**Procedure**

1. Enter the following program in the IAR embedded workbench and create the executable file (HEX file)
2. Switch on the kit
3. Connect the USB cable and check the COM port in the device manager
4. Keep the SW21 in programming mode (PROG)
5. Download the code using Flash Magic tool
6. Open WinXtalk/Hyper Terminal with 9600 baud rate
7. Keep the SW21 in execution mode (EXE) and press reset
8. The result is displayed in the WinXtalk/Hyper Terminal

**Program:**

//Header Files

#include<nxp/iolpc2148.h>

#include<stdio.h>

#define f 12000000 // PCLKFREQ ( controller frequency )

//Function to initialize Serial Port

void serial\_Init(void) // Function to initialise the serial port

{

unsigned int baudrate ;

PINSEL0=0X00000005;

VPBDIV=0X01;

baudrate = (f/(9600\*16)); // Baudrate setting (9600 bps)

U0LCR=0X80;

U0DLL = baudrate & 0X00FF;

U0DLM=( baudrate >>8) & 0X00FF;

U0LCR=0X03;

}

//Function to print the data in the serial port

int putchar(int ch)

{

if (ch == '\n')

{

while (!(U0LSR & 0x20));

U0THR = 0x0d;

}

while (!(U0LSR & 0x20));

return (U0THR = ch);

}

//Main Routine

void main()

{

unsigned int a,b,c;

serial\_Init(); // Initialise the serial port

a = 0x7FD1;

b = 0x229B;

c = a \* b;

printf("\n\r Multiplication of two given 16-bit numbers is %x",c);

while(1);

}

**Result:**

Thus, the ‘C’ program is developed for LPC2148 microcontroller to multiply two 16-bit operands

**Experiment 15: Write C program to generate a triangular wave using ARM controller**

**Aim:**

The aim of the experiment is to generate a triangular wave using ARM controller

**Tools Required:**

1. VSK-2148 ARM7 LPC2148 Syllabus Board – 1
2. Power Card - 2
3. USB cable - 1
4. CRO/DSO - 1
5. CRO probe - 1

**Theory:**

The LPC2148 microcontroller is a 32-bit microcontroller which has many in-built peripherals like ADC, DAC, Timers/Counters, PWM module, SPI, I2C, USB and so on. The waveform generation is done by DAC in the microcontroller. The DAC (Digital to Analog Converter) converts digital data into analog (voltage). The program is stored in the Flash memory of the microcontroller. It has 10-bit DAC in which the output is 0V when 000 is given to the DAC register and the output is +3.3V when 3FF is given to the DAC register.

**Procedure**

1. Enter the following program in the IAR embedded workbench and create the executable file (HEX file)
2. Switch on the kit
3. Connect the USB cable and check the COM port in the device manager
4. Keep the SW21 in programming mode (PROG)
5. Download the code using Flash Magic tool
6. Keep the SW21 in execution mode (EXE) and press reset
7. Switch on the CRO/DSO and connect the probe at AOUT (P9 Connector) and ground
8. The generated triangular wave can be viewed in the CRO/DSO

**Program:**

//Header Files

#include<nxp/iolpc2148.h>

//Variable Declarations

unsigned int i;

//Main Routine

void main()

{

PINSEL1 = 0x00080000; /\* set P0.25 to DAC output \*/

while ( 1 )

{

for(i=0;i<1024;i++)

{

DACR = (i << 6);

}

for(i=1024;i>0;i--)

{

DACR = (i << 6);

}

}

}

**Result:**

Thus, the ‘C’ program is developed for LPC2148 ARM microcontroller to generate a triangular wave.

**Experiment 16: Write a C program to sample a sine wave at 1000 samples/second and store the samples in memory using ARM processor and ADC**

**Aim:**

The aim of the experiment is to sample a sine wave at 100s samples per second and store the samples in the memory using ARM processor.

**Tools Required:**

1. VSK-2148 ARM7 LPC2148 Syllabus Board – 1
2. Power Card - 3
3. USB cable - 1
4. CRO/DSO - 1
5. CRO probe - 2
6. Function Generator - 1

**Theory:**

The LPC2148 microcontroller is a 32-bit microcontroller which has many in-built peripherals like ADC, DAC, Timers/Counters, PWM module, SPI, I2C, USB and so on. The ADC of the LPC2148 microcontroller is 10-bit resolution and the conversion time for the ADC is 2.44uS. In this experiment totally 1000 samples are taken per second. The sine wave is given as the input. The ADC converts the sine wave samples for 1 second. The timer interrupt is used to calculate the accurate time.

**Procedure**

1. Enter the following program in the IAR embedded workbench and create the executable file (HEX file)
2. Switch on the kit
3. Connect the USB cable and check the COM port in the device manager
4. Keep the SW21 in programming mode (PROG)
5. Download the code using Flash Magic tool
6. Switch on the Function Generator and supply sine wave at ADC2 of connector P7
7. Open WinXTalk/Hyper Terminal at the baud rate of 9600
8. Keep the SW21 in execution mode (EXE) and press reset
9. The sine samples are displayed in the WinXtalk.

**Program:**

//Header Files

#include<nxp\iolpc2148.h>

#include<intrinsics.h>

#include<stdio.h>

//Declarations

#define f 12000000 // PCLKFREQ

unsigned int value[1000];

unsigned long int i=0;

//Function to initialize Serial Port

void serial\_Init(void)

{

unsigned int baudrate ;

PINSEL0|=0X00000005;

VPBDIV=0X01;

baudrate = (f/(9600\*16)); // Baudrate setting (9600 bps)

U0LCR=0X80;

U0DLL = baudrate & 0X00FF;

U0DLM=( baudrate >>8) & 0X00FF;

U0LCR=0X03;

}

//Function to print the data in the serial port

int putchar(int ch)

{

if (ch == '\n')

{

while (!(U0LSR & 0x20));

U0THR = 0x0d;

}

while (!(U0LSR & 0x20));

return (U0THR = ch);

}

//IRQ Handler

#pragma vector=0x18

\_\_irq \_\_arm void irq\_handler (void)

{

void (\*interrupt\_function)();

unsigned int vector;

vector = VICVectAddr;

interrupt\_function = (void(\*)())vector;

(\*interrupt\_function)();

VICVectAddr = 0;

}

//FIQ Handler

\_\_fiq \_\_arm void fiq\_handler (void)

{

while(1);

}

//Function to Initialize ADC1

void ADC1\_Init()

{

PINSEL0\_bit.P0\_12=0X03;

AD1CR\_bit.CLKDIV=2;

AD1CR\_bit.BURST=1;

AD1CR\_bit.PDN=1;

AD1CR\_bit.SEL=0X08;

AD1CR\_bit.START=0X001;

}

//Interrrupt Service Routine for Timer0

void timer0\_ISR()

{

T0IR = 1;

T0TC = 0xFFFFD11F;

while(AD1DR\_bit.DONE==0);

value[i]=(AD1DR3\_bit.RESULT & 0x3FF);

i = i+1;

}

//Function to Initialize Timer0 for 1 second

void Timer0\_Init()

{

T0IR =0xFF; // reset match and capture event interrupts

T0TC = 0 ; // Clear timer counter

T0PR = 0 ; // No Prescalar

T0MR0 = 0xFFFFFFFF;// Match Register, Interrupt occurs when T0TC=0xFFFFFFFF

T0MCR = 3; // Reset Timer Counter & Interrupt on match

T0TCR = 1 ; // Counting enable

VICIntSelect = 0; // Set all VIC interrupts to IRQ for now

VICIntEnClear = 0xFFFFFFFF; // Diasable all interrupts

VICProtection = 0; // VIC registers can be accessed in User or privileged mode

VICVectAddr = 0; // Clear interrupt

VICProtection = 0; // Accesss VIC in USR | PROTECT

VICIntSelect &= ~(1<<VIC\_TIMER0); // Timer 0 intrpt is an IRQ (VIC\_TIMER0 = 4)

VICVectAddr0 = (unsigned int)&timer0\_ISR; // Install ISR in VIC addr slot 0

VICVectCntl0 = 0x20 | VIC\_TIMER0; // IRQ type, TIMER 0 int enabled

VICIntEnable |= (1<<VIC\_TIMER0); // Turn on Timer0 Interrupt

T0TC = 0xFFFFD11F;

}

//Main Routine

void main()

{

serial\_Init();

printf("\nSampling Sine Wave");

Timer0\_Init();

ADC1\_Init();

i=0;

VICIntEnable |= (1<<VIC\_TIMER0); // Turn on Timer0 Interrupt

\_\_enable\_interrupt();

while(1)

{

if(i==1000)

{

T0IR = 1;

VICIntEnClear = 0xFFFFFFFF; // Diasable all interrupts

printf("\nSamples Per Second");

for(i=0;i<1000;i++)

printf("\n%d",value[i]);

T0TC=0;

T0TCR = 0;

\_\_disable\_interrupt();

i=0;

}

}

}

**Result:**

Thus, the ‘C’ program is developed for LPC2148 microcontroller to sample a sine wave at 1000 samples per second and the samples are stored in the memory.