**PROGRAM**

### [A]Square waveform generation using internal DAC of LPC2148

#include <lpc214x.h> #include <stdint.h>

void delay\_ms(uint16\_t j)

{

uint16\_t x,i; for(i=0;i<j;i++)

{

for(x=0; x<6000; x++); /\* loop to generate 1 milisecond delay with Cclk = 60MHz \*/

}

}

int main (void)

{

uint16\_t value; uint8\_t i;

i = 0;

PINSEL1 = 0x00080000; /\* P0.25 as DAC output \*/ while(1)

{

value = 1023;

DACR = ( (1<<16) | (value<<6) );

delay\_ms(100); value = 0;

DACR = ( (1<<16) | (value<<6) );

delay\_ms(100);

}

}

**EX.NO : DATE :**

**WAVEFORM GENERATION USING 10 BIT DAC**

**AIM:**

To write the embedded C program to generate a triangular and square wave form using

internal 10 bit DAC using LPC2148 ARM Micro controller.

**APPARATUS REQUIRED :**

1. LPC 2148 ARM Microcontroller Development board.

2 Keil µVision version 5

1. Flash Magic.

### Input:

**Output:**

DAC pin (P0.25)

**ALGORITHM:**

* + First, configure P0.25/AOUT pin as DAC output using PINSEL Register.
  + Then set settling time using BIAS bit in DACR Register.
  + Now write 10-bit value (which we want to convert into analog form) in VALUE field of DACR Register.

### [B]Triangular wave generation using internal DAC of LPC2148

#include <lpc214x.h> #include <stdint.h>

void delay\_ms(uint16\_t j)

{

uint16\_t x,i; for(i=0;i<j;i++)

{

for(x=0; x<6000; x++); /\* loop to generate 1 milisecond delay with Cclk = 60MHz \*/

}

}

int main (void)

{

uint16\_t value; uint8\_t i;

i = 0;

PINSEL1 = 0x00080000; /\* P0.25 as DAC output \*/ while(1)

{

value = 0;

while ( value != 1023 )

{

DACR = ( (1<<16) | (value<<6) );

value++;

}

while ( value != 0 )

{

DACR = ( (1<<16) | (value<<6) );

value--;

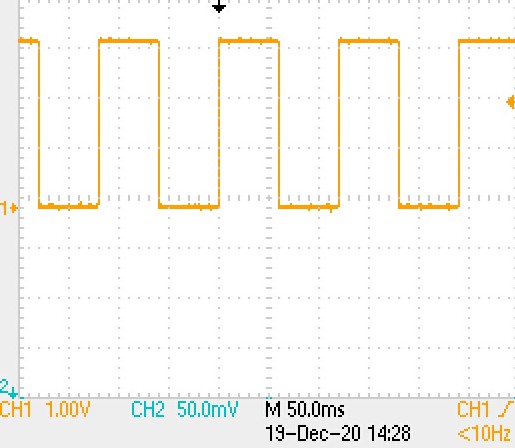
}

}

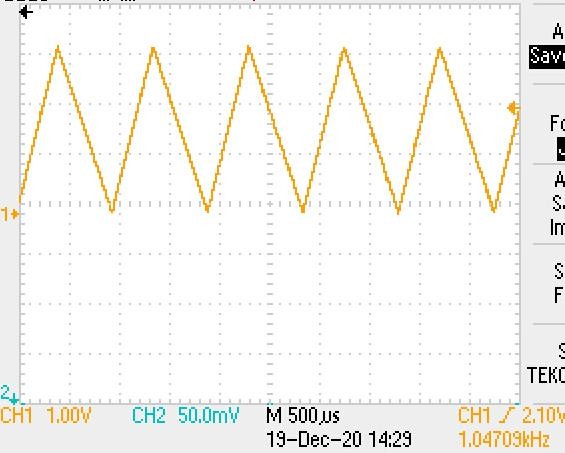
}

**OUTPUT WAVEFORMS**

**[A] SQUARE WAVE FORM**

****

**[B]TRIANGULAR WAVEFORM**

****

**RESULT :**

Thus the embedded C program to generate the triangular and Square wave form with internal 10 bit DAC using LPC2148 ARM Micro controller was executed and output was verified with oscilloscope.

**Arithmetic Operations Using LPC2148 Kit**

**AIM** The LPC2148 microcontroller, based on the ARM7 TDMI-S architecture, is well-suited for performing arithmetic operations. This project involves implementing basic arithmetic operations (addition, subtraction, multiplication, and division) using the LPC2148 development kit.

**Hardware Requirements**

1. LPC2148 Development Board
2. 16x2 LCD Display (optional, for output display)
3. Keypad (4x4, for input) or pre-defined values in the code
4. USB-UART Cable (for programming and debugging)
5. Power Supply

**Steps to Implement Arithmetic Operations**

**1. System Initialization**

* Initialize the LPC2148 microcontroller.
* Configure GPIO pins for interfacing with peripherals such as the keypad or LCD.

**2. Input Data**

* **Keypad Input:** Use a 4x4 keypad to accept user inputs for numbers and the arithmetic operation.
* **Predefined Input:** If a keypad is unavailable, hard-code the input values for testing purposes.

**3. Perform Arithmetic Operations**

The microcontroller performs basic arithmetic operations such as addition, subtraction, multiplication, and division using standard C programming constructs.

**4. Display Output**

* **LCD Display (Optional):** Use a 16x2 LCD to show the input and output.
* **Serial Terminal:** Send the results via UART to a connected PC for debugging or display.

**Software Implementation**

**Algorithm:**

1. Initialize the peripherals (LCD, UART, Keypad, or Timer).
2. Accept two input numbers from the user and the desired operation.
3. Use a switch-case or if-else logic to determine the operation to perform.
4. Calculate the result and display it on the LCD or send it via UART.

**Code Snippet:**

Below is an example code for performing arithmetic operations:

c

Copy code

#include <lpc214x.h>

// Function prototypes

void init\_UART0(void);

void send\_UART0(char \*str);

int perform\_operation(char op, int a, int b);

int main(void) {

char operation;

int num1 = 10, num2 = 5; // Example inputs

int result;

init\_UART0(); // Initialize UART for output

// Example: operation = '+' (Addition)

operation = '+';

result = perform\_operation(operation, num1, num2);

// Send results via UART

char result\_msg[50];

sprintf(result\_msg, "Result of %d %c %d = %d\r\n", num1, operation, num2, result);

send\_UART0(result\_msg);

while (1);

}

// Function to perform arithmetic operations

int perform\_operation(char op, int a, int b) {

switch (op) {

case '+': return a + b;

case '-': return a - b;

case '\*': return a \* b;

case '/': return (b != 0) ? (a / b) : 0; // Avoid division by zero

default: return 0;

}

}

// UART0 initialization

void init\_UART0(void) {

PINSEL0 = 0x00000005; // Enable UART0 on P0.0 (TXD) and P0.1 (RXD)

U0LCR = 0x83; // 8-bit data, no parity, 1 stop bit

U0DLM = 0; // Set baud rate to 9600

U0DLL = 97;

U0LCR = 0x03; // Disable divisor latch access

}

// Send string over UART0

void send\_UART0(char \*str) {

while (\*str) {

while (!(U0LSR & 0x20)); // Wait for the transmitter to be ready

U0THR = \*str++;

}

}

1. Display a number in seven segment LED in LPC 2148 kit
   1. Decimal Numbers
   2. Hexa-Decimal Numbers
   3. Alphabets

#include <LPC214x.h>

unsigned char dig[]={0x88,0xeb,0x4c,0x49,0x2b,0x19,0x18,0xcb,0x8,0x9,0xa,0x38,0x9c,0x68};

void delay(unsigned int count)

{

int j=0,i=0;

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

{

for(i=0;i<120;i++);

}

}

int main(void)

{

unsigned char count=0;

unsigned int i=0;

IO0DIR|=(1<<11);//Set Digit control lines as Outputs

IO0SET=(1<<11);

IO0DIR|=0x007F8000;

while(1)

{

count++;

if(count==16)count=0;

for(i=0;i<800;i++)//change to inc/dec speed of count

{

IO0CLR=0x007F8000;

IO0SET=(dig[count]<<15);

delay(200);

}

}

}