EXP. No: 9	SIGNAL GENERATION AND BASIC OPERATION ON SIGNALS USING DSP PROCESSOR
Date:	

AIM:

To generate basic signals and perform basic operations using DSP processor.

SOFTWARE USED:

Code Composer Studio V5

HARDWARE USED:

LCDKC6748 (Low Cost Development Kit)

PROCEDURE:

- Go to View->Target Configurations
- On the Right Hand Side of your project window, you can notice a target configuration space.
- Click on the new target configuration file as shown in the screen shot.
- Give a suitable name to yourtarget device (Here its named as My_target.ccxml)
- Click on Finish.
- The Connection should be “Texas Instruments XDS100V2 USB Emulator”. The Board or Device should be LCDKC6748.(you may also use XDS100v3,so change accordingly)
- Click on Save and then Test Connection. If your connection is OK, then a message as “JTAG-DR integrity test succeeded” will appear.
- Click on the “Advanced” tab and then Click on C674x_0, verify the GEL file path.
- Close “Target Configuration” window and Right Click on My_target and select SET AS DEFAULT and then LAUNCH SELECTED CONFIGURATION.
- Debug window will be launched.
- After the above step is done, click on Load Program button, if you know the path for the executable file (.out), specify the location and load the .out file of a particular project.
- Once the executable file is loaded, you can start running the program by clicking on the Resume button.
- Click on Terminate button to stop the debug session and go back to CCS EDIT window.

PROGRAM:

AUDIO INTERRUPT:

```
// L138_loop_intr.c

#include "L138_LCDK_aic3106_init.h" //header file for

interrupt void interrupt4(void) // interrupt service routine
{

  uint32_t sample; //16bit input variable
  sample = input_sample(); // read L + R samples from ADC
  output_sample(sample); // write L + R samples to DAC
  return;

}

int main(void)
{

  L138_initialise_intr(FS_48000_HZ,ADC_GAIN_0DB,DAC_ATTEN_0DB,LCDK_LINE_INPUT);
  while(1);

}
```

PROGRAM:

AUDIO LOOP BACK

```
#include "L138_LCDK_aic3106_init.h"

int main(void)

{

 uint32_t sample;

L138_initialise_poll(FS_48000_HZ,ADC_GAIN_0DB,DAC_ATTEN_0DB,LCDK_LINE_INPUT);

while(1)

 {

 sample = input_sample();

 output_sample(sample);

 }}
```

PROGRAM:

FIR LOW PASS FILTER:

```
float h[N] = {-0.000035,-0.000234,-0.000454,0.000000,0.001933,0.004838,0.005671,
-0.000000,-0.013596,0.028462,0.029370,0.000000,0.064504,0.148863,0.221349
,0.0249983,0.221349,0.148863,0.064504,0.000000,-0.029370,-0.028462,-0.013596,
-0.000000,0.005671,0.004838,0.001933,0.000000,-0.000454,-0.000234,
-0.000035};
float x[N]; // filter delay line
interrupt void interrupt4(void)
{
  short i;
  float yn = 0.0;
  x[0] = (float)(input_left_sample()); // input from ADC
  for (i=0 ; i<N ; i++)                 // compute filter output
  yn += h[i]*x[i];
  for (i=(N-1) ; i>0 ; i--)             // shift delay line
    x[i] = x[i-1];
  output_left_sample((uint16_t)(yn)); // output to DAC
 return;
}
int main(void)
{
  L138_initialise_intr(FS_8000_HZ,ADC_GAIN_0DB,DAC_ATTEN_0DB,LCDK_LINE_INPUT);
while(1);
```

PROGRAM:

GENERATION OF SINE WAVE:

```
#include "L138_LCDK_aic3106_init.h"

//Fs=8khz, Fc=3khz
#define N 31
#include "L138_LCDK_aic3106_init.h"
#define LOOPLENGTH 48
int16_t sine_table[LOOPLENGTH] = {0, 1305, 2588, 3827,
 5000, 6088, 7071, 7934, 8660, 9239, 9659, 9914, 10000,
 9914, 9659, 9239, 8660, 7934, 7071, 6088, 5000, 3827,
2588, 1305, 0, -1305, -2588, -3827, -5000, -6088, -7071,
-7934, -8660, -9239, -9659, -9914, -10000, -9914, -9659,
-9239, -8660, -7934, -7071, -6088, -5000, -3827, -2588,
-1305};
int sine_ptr = 0; // pointer into lookup table
interrupt void interrupt4(void) // interrupt service routine
{
 uint16_t left_sample;
 left_sample = sine_table[sine_ptr];
 sine_ptr = (sine_ptr+1)%LOOPLENGTH;
 output_left_sample(left_sample);

 return;
}
int main(void)
{
 L138_initialise_intr(FS_48000_HZ,ADC_GAIN_0DB,DAC_ATTEN_0DB,LCDK_LINE_INPUT);
 while(1);}
```

RUBRICS:

Practical component	**Indicator**	**Excellent (80-100%)**	**Good (79-50%)**	**Satisfactory (<50%)**	**Marks**
Conduct of Experiment (20)	Analyze the problem and develop programming constructs (15)				
	Completeness of the experiment (5)				
Observation and result (20)	Interpretation of the findings (15)				
	Simulation and graph (5)				
Record (10)	Adherence to record submission deadline (5)				
	Presentation and completion of record (5)				
Viva Voce (10)	Ability to recall the theoretical concepts				
				TOTAL (60)	

RESULT:

Thus, basic signals and waveforms are generated and arithmetic operations using them are successfully done using Simulink.