EE/TE 4385

Lab 2: Sinwave Generation and LabView DSP Integration Lab Report Due: 9/06/06, 2PM

Lab 2a. Sinwave Generation

The objective of this project is to generate 1000Hz sinewave by using look-up table. Assume that the sampling frequency, fs, is 8000Hz.

- 1. Download source file from the lab web side
- 2. Open a new project named Lab2a in CCStudio.
- 3. Your source code is sine.c. You need to modify the source code to perform the sinusoid generation.
- 4. Examine provided source code and note the line
 - a. float sine_array[8] = {....}
 - b. This is where you need to insert sine samples for one period.
- 5. Generate and plot one period of sinewave consisting of 8 data samples in MATLAB. Record the matlab figure window for your report and sample values for your lab report.
- 6. Add source files to the project (if they are not already added):
 - a. sine.c
 - b. Lab02.cmd (Linker command file)
 - c. vector.asm (Interrupt vector table)
- 7. Add necessary library files to the project:
 - a. C:\ti\c6000\cgtools\lib\rts6700.lib"
 - b. C:\ti\c6000\bios\lib\cs16713.lib"
 - c. C:\ti\c6000\dsk6713\lib\dsk6713bsl.lib"
- 8. Build and run your project to verify the output on the oscilloscope. You need the connect the right audio output of TI DSK to the oscilloscope. Optionally you can connect to the amplified speaker to listen since the frequency is audible. Record the oscilloscope screen for your lab report.
- 9. Use the same method to generate 750 Hz sinwave. What do you need to change? Include your results in the lab report

Lab 2b. LabVIEW DSP Integration Lab:

This lab will show how to combine LabVIEW and TI DSP implementation. The lab is based on Combined LabVIEW DSP Implementation example available in LabVIEW examples directory.

- 1. Start LabVIEW.
- 2. Copy gain and shared folder from Examples → DSPTest→dsk6713→ file under LabVIEW 8.0 directory into your myprojects directory.
- 3. Open Gain \rightarrow Gain_dsk6713.vi in its new location.
- 4. The Gain example uses LabVIEW signal generation to create a waveform and applies a waveform gain factor in TI DSP to the generated waveform. Then, the generated waveform is read into LabVIEW for viewing from TI DSP.
- 5. Display the Block Diagram to inspect the blocks
- 6. Using RTDX technology, the example sends the waveform to the **cinput** channel of the target code and sends the waveform gain to the **cgain** channel of the target code. The

target code combines the original waveform and gain and then returns a scaled waveform to LabVIEW through the **coutput** channel.

- 7. Complete the following steps to demonstrate communication between a LabVIEW VI and target code.
- 8. Run the Gain VI example.
- 9. Select the waveform type from the Signal Type control.
- 10. Use the **Gain** control on the front panel to add gain to the original sine wave. The front panel waveform graph displays the original waveform and the scaled waveform.
- 11. Use the **Frequency** control on the front panel to change the frequency of the signal LabVIEW generates. Record the graph window for few examples of frequency and gain values for your lab report.
- 12. Click the **STOP** button to stop the example VI.
- 13. Next, instead of gain adjust, add the delay to the waveform by modifying the gain slide control on the LabVIEW VI and gain.c file in the Code Composer.
- 14. Change the range of gain slide controller to 0 to 6. Since this generates floating point numbers, you need to convert it to nearest integer.
- 15. Modified Block Diagram should look like the following. Notice the only difference as the change from Gain to Delay block and rounding to nearest integer (Round to Nearest VI).



- 16. You also need to modify gain.c file for DSP side since DSP will buffer and delay the samples generated by LabVIEW.
- 18. Modify gain function in gain.c

```
output[i]=longinput[i+kBUFFER_SIZE-gain];
```

{
longinput[i]=longinput[i+kBUFFER_SIZE];
longinput[i+kBUFFER_SIZE]=input[i];
}

- 20. Note that you need to make few other changes to successfully compile and run your program, i.e., variable name cname in gain.c should be changed to cdelay to match the LabVIEW side.
- 21. Compile and run your program.

- 22. Use the **Delay** control on the front panel to change the delay of the signal LabVIEW generates. Record the graph window for few examples of frequency and delay values for your lab report.
- 23. Interpret the actual delay amount that corresponds to delay index on the LabVIEW delay slider in your lab report.