1 Overview

The purpose of this lab is to familiarize you with Simulink, Real Time Workshop, Link for CCS and how they interact with Code Composer Studio (CCS). This lab involves building relatively simple systems using Simulink rather than CCS.

2 Software overview

Simulink uses a block based approach to algorithm design and implementation. Real-Time Workshop converts these Simulink models into ANSI C/C++ code that can be compiled using CCS. The Embedded Target for TI C6000 DSP provides the APIs (Advanced programming interface) required by Real-Time Workshop to generate code specifically for the C6000 platform. Link for CCS is used to invoke the code building process from within CCS to build an executable. This code can then be downloaded on the target DSP from where it runs. This approach allows for rapid prototyping. Also, a model built once in Simulink can be made to run on various system as long as one has the appropriate Embedded target toolbox.

3 Initial configuration (for all labs)

This section describes how configure Simulink and Real-Time Workshop for use with the C6713 DSK. These configuration setting are required and are similar for all the experiments. While this configuration generally has to be done only once, it must be ensured that these setting do exist. This is the first step in any experiment using Simulink.

3.1 Check if CCS is properly installed

To verify that CCS is properly installed on the system, enter

ccsboardinfo
Figure 1: Software overview

at the Matlab command line. Matlab should return information similar to the following listing:

<table>
<thead>
<tr>
<th>Board</th>
<th>Board</th>
<th>Proc</th>
<th>Processor</th>
<th>Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num</td>
<td>Name</td>
<td>Num</td>
<td>Name</td>
<td>Type</td>
</tr>
<tr>
<td>0</td>
<td>C6713 DSK</td>
<td>0</td>
<td>CPU_1</td>
<td>TMS320C6x1x</td>
</tr>
</tbody>
</table>

To ensure Embedded Target for TI C6000 DSP is installed, enter
c6000lib

Matlab should display the C6000 block library containing the libraries: C6000 DSP Core Support, C62x DSP Library, C64x DSP Library, C6416 DSK Board Support, C6701 EVM Board Support, C6711 DSK Board Support, C6713 DSK Board Support, RTDX Instrumentation, TMDX326040 Daughtercard Support.

3.2 Configuration Parameters for C6000 Hardware

1. Launch Matlab

2. At the Matlab command line, type
simulink

to launch Simulink

3. Create a new model in Simulink.

4. To open the Configuration Parameters, select Simulation→Configuration Parameters

5. In the Select tree, chose the Real-Time Workshop category.

6. For Target Selection, choose the file ti_c6000.tlc. Real-Time Workshop will automatically change the Make command and Template makefile selections.

7. Choose the Optimization category in the Select tree. For Simulation and Code generation, un-select Block reduction optimization and Implement logic signals....

8. Choose the TI C6000 target sel.... Set Code generation target type to C6713DSK.

9. Choose the TI C6000 compiler. Set Symbolic debugging.

10. In the Select tree, choose the Debug category. Select Verbose build here.

11. In the Select tree, choose the Solver category. Ensure that Solver is set to Fixed type / discrete.

4 Building an audio effects processor in Simulink

In this experiment, an audio effects processor is created to echo, reverberation and flange.

4.1 Building a Simple I/O system with the C6713 DSK

To build the model for a Single Audio Effect, follow these steps:

1. Perform the steps in section 2.4.1 and 2.5.

2. The DSK has four input and output jacks: Headphone, Line Out, Line In, and Mic In. The goal is to implement a system that accepts an audio signal via the Mic In jack and passes that signal through, unchanged, to the Headphone jack.

3. Select the Simulink Library Browser window and select Embedded Target for TI C6000. Within this group, select the C6713 DSK Board Support library. There will be five (5) blocks displayed: ADC, DAC, LED, Reset, and Switch.
4. Place the ADC (Analog-to-Digital Converter) and DAC (Digital-to-Analog Converter) blocks on your model by clicking and dragging.

5. Once they are on your model, connect the two, as shown in figure 2.

   ![Simple I/O diagram](image)

   **Figure 2: Simple I/O**

6. Double clicking a block displays the block parameters. While some parameters are userdefinable, others are not. The ADC source field can be set to either Iic In or Line In. If the input is from the microphone then the field should be set to Mic In; otherwise, it should be set to Line In. In the ADC block, Stereo should be unchecked and the output data type set to integer. NOTE: The sample rate can also be set here. For most applications, this is either set to 8kHz (speech applications) or 44.1kHz (music applications). The Samples per Frame parameter tells the system how many samples to process at one time. The greater the samples per frame, the longer it takes to process the signal (due to memory needs and computational load).

7. Double click the DAC block. Ensure that the Word Length field is the same (16-bit) for both the ADC and DAC blocks.

8. Save the model:

   Select File > Save As > ...location

9. Ensure that the hardware is connected properly by:

   (a) Connecting the microphone to the Mic In jack.
   (b) Connecting the headphones to the Headphone jack.

10. The model is now complete and ready to run. Verify that the DSK board is connected properly. Use the **Incremental Build** command on the Simulink model toolbar (the icon with the 3 arrows) to begin compiling the model and transferring the code to the DSK.

11. If this is the first time the board has been used since turning on the computer, a DSK Startup window will open temporarily to configure the board.

12. The system will first compile code in Matlab, then Code Composer Studio (CCS) will be opened, where assembly code will be compiled and loaded onto the DSP. Once in CCS, a progress window will open as the code is loaded onto the board.
13. The program, once loaded will begin to run. Test the model by speaking into the microphone and listening to your speech through the headphones. After verifying that the code works, stop the program by using the **Halt** command in CCS. Close CCS and return to the model in Simulink.

### 4.2 Designing a single echo effect in real time

In the above section, a basic I/O system for audio was built, compiled and run. This section investigates data manipulation and representation on the DSK by designing a single echo filter.

In this section, an echo filter is designed. An echo is a delayed, attenuated version of the original signal. The difference equation describing a system with a single echo is defined as $y[n] = x[n] + ax[n - D]$, where $a$ is the gain and $D$ is the delay. Figure 3 shows the Simulink block diagram that implements this filter.

1. Build and run the model in figure 3 (using the Incremental Build button on the model toolbar).
2. Add the C6713DSK target preferences block from the TI C6000 Target Preferences block library. This block is not connected to any other block in the model.
3. Test the model using the microphone (Mic In) and a pair of headphones (Headphone).
4. Change the delay and gain by double clicking on the desired block and explore the effect of each on the output signal.

![Figure 3: Real time single echo filter](image)

---

5
4.3 Real time audio reverberation model

In this section a model that simulates audio reverberation is created and built. Reverberation is the result of an infinite number of echoes added together, and can be represented mathematically as $y[n] = x[n] + ax[n-D] - by[n-D]$, where $a$ and $b$ are the gains and $D$ is the delay. For this experiment, a microphone is connected to the Mic In connector on the C6713 DSK and speakers (and an oscilloscope) connected to the Line Out connector on the C6713 DSK. To test the model, speak into the microphone and listen to the output from the speakers.

1. Open Simulink.
2. Create a new model by selecting File -> New -> Model from the Simulink menu bar.
3. Use Simulink blocks to create the model shown in figure 4.
4. The Integer Delay block is in Discrete library of the Simulink blockset. The Gain block is in the Commonly Used library of the simulink blockset. The Line In and Line Out block for the C6713 DSK are in the C6713 Board Support library of the Embedded Target for TIC6000 blockset.
5. From the TI C6000 Target Preferences block library, add the C6713DSK target preferences block. This block is not connected to any other block in the model.
6. Click the C6713 DSK ADC block to select it. Select Block Parameters from the Simulink Edit menu. Set the following parameters for the block:
   - Select the +20 dB mic gain boost check box
   - For Output data type, select Double from the list
   - Set Scaling to Normalize
   - Set Source gain to 0.0
   - Enter 64 for Samples per frame
7. For ADC source, select Mic In.
8. Click OK to close the Block Parameters: ADC dialog.
9. Now set the options for the C6711 DSK DAC block.
   - Set Scaling to Normalize
   - For DAC attenuation, enter 0.0
   - Set Overflow mode to Saturate.
10. Click OK to close the dialog.
11. Open the Configuration Parameters dialog, select Solver on the Select tree, and set the appropriate options on the Solver pane for your model and for the target.

   - Under Solver options, select the fixed-step and discrete settings from the lists.
   - Set the Fixed step size to auto and select Single Tasking for the Tasking mode.

12. Build and run the model (using the Incremental Build button on the model toolbar).

13. Speak into the microphone connected to the board. The model should generate a reverberation effect out of the speakers, delaying and echoing the words you speak into the mike.

Audio Reveberation: C6713 DSK Implementation

Figure 4: Real time reverberation filter
4.4 Real time audio flange model

A flange effect is a (time domain) audio effect, which is achieved by combining to identical signal with each other, with one signal delayed by a gradually changing amount. The difference equation for the flange effect is given by, \( y[n] = x[n] + ax[n - d(n)] \), where the time varying delay is given by, \( d(n) = D + \frac{D}{2}(1 - \sin(\frac{2\pi f_d n}{f_s})) \), where \( D \) is a fixed delay, \( f_d \) is the flanging frequency, and \( f_s \) is the sampling rate. Figure 5 shows the Simulink block diagram that implements this system.

1. Build and run the model in figure 5 (using the Incremental Build button on the model toolbar).

2. Add the C6713DSK target preferences block from the TI C6000 Target Preferences block library. This block is not connected to any other block in the model.

3. Test the system using recorded music or speech played through the Line In connector of the DSK and a pair of headphones connected to the Headphone connector.

![Figure 5: Real time flange filter](image-url)