Integration of Matlab Tools for DSP Code Generation

Bradley University
ECE Department
December 6th, 2005

Team Members:
Kwadwo Boateng and Charles Badu

Advisor:
Professor Thomas Stewart

Project Outline

- Project Summary
- A Detailed Description
- Project Schedule
- Bibliography
- Equipment List

Project Summary

This project focuses on developing a software user interface using Matlab tools that will allow a user to easily execute C, C++ and assembly language code on a Texas Instrument DSP board (TMS320C6713). A signal processing description to be executed on this DSP board will first be implemented using either a Matlab M-file or Simulink block diagrams. The block diagram in Simulink is first converted to C code using the Real-Time Workshop feature in Simulink. The Code Composer Studio software package converts the C code into an executable file, which is downloaded onto the board and displayed on the oscilloscope.

A Detailed Description

- **Functional Description**
  System Block Diagram
  System Flow Chart
- **Standards**
- **Datasheets**

Functional Description

System Block Diagram

The software interface to be developed for the code integration will operate according to the block diagram shown below in Figure 1.

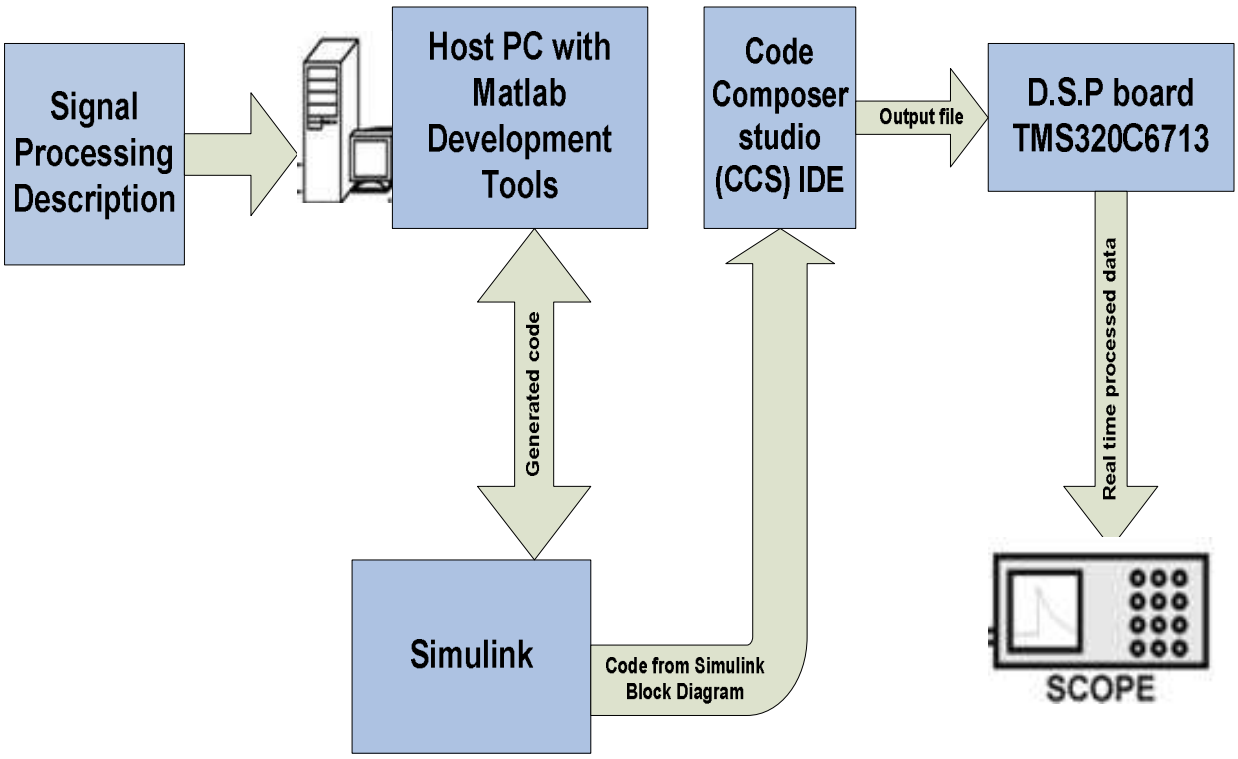

Figure 1 is a High-level system block diagram

Signal Processing Description

This is a given problem to be implemented on the D.S.P board. It can involve implementing a control feedback system on the D.S.P board or analyzing a complex data acquisition scheme on a D.S.P board. Any problem whose output is in the form of signals or data that can adequately be processed in real time on the TMS3206713 is our Signal Processing Description.

Host PC with MATLAB Development Tools

The signal processing description is implemented by first writing code using Matlab development tools. If the signal processing description is in C++, C or assembly language code, it has to be converted to a MEX file, which is interfaced with Matlab. The Matlab code generated is saved into an input file or a workspace to be used in Simulink.

Simulink

Simulink, an extension of Matlab will be used also to solve our Signal Processing Description by first obtaining from Matlab an input file and displaying our graphical results numerically. A number of design blocks which may be unavailable in Simulink need to be developed in Matlab by writing C code. Only a constructed block diagram representation of the Signal Processing Description in Simulink is sent to the Code Composer Studio (CCS).

Code Composer Studio (CCS)

The CCS integrated development software converts the Simulink block diagram into C and assembly language, which downloaded as an output file onto the TMS3206713. The running process can be accessed only from the CCS debugging tools or across a link for CCS or Real-time data Exchange. Otherwise the running process is not accessed.

D.S.P Board (TMS3206713)

The TMS3206713 replaces the work of Simulink. The TMS3206713 outputs real time processed data or signals to the oscilloscope. The output on the scope represents our signal processing result.

System Flow Chart

A system flow chart explaining the interconnection between Matlab Simulink and the DSP board is shown in Figure 2.

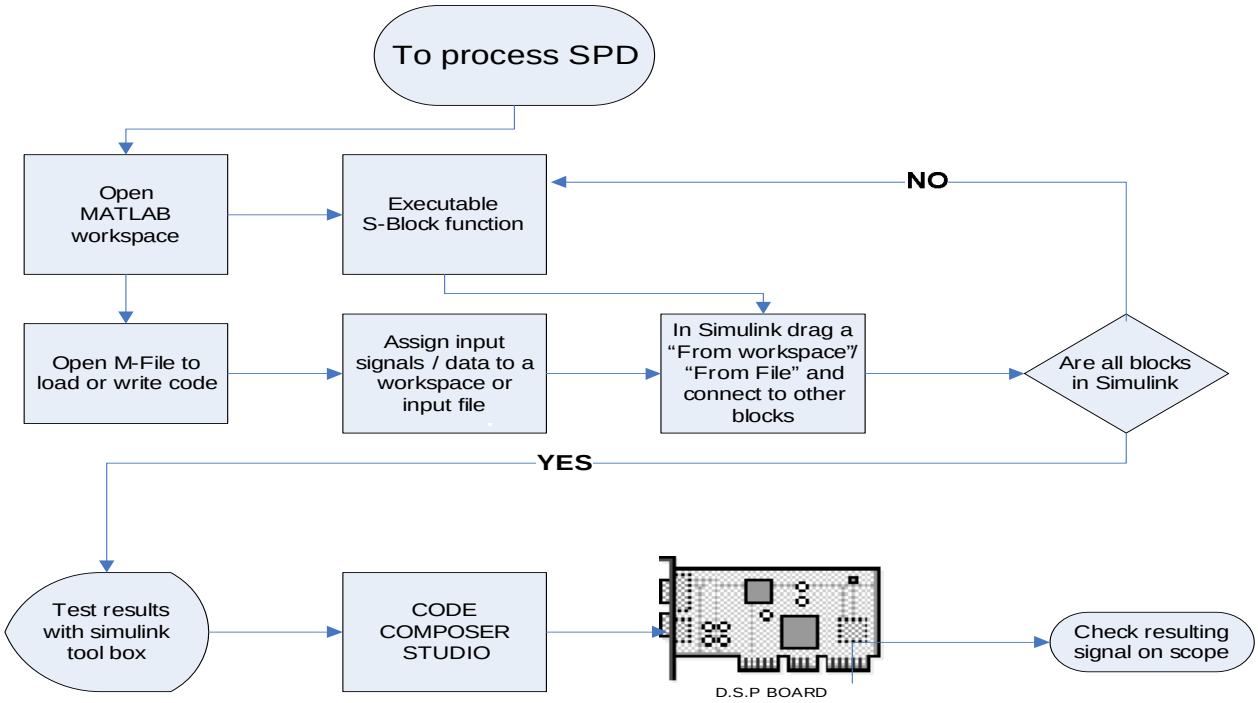

Figure 2 is a system flow chart

Datasheets:

MathWork's Complete Simulink Tutorial

Preliminary Computer Simulations

An M-file was used to generate a sine wave and the magnitude of the sine wave and time were saved in a Matlab work space as simin. A sample plot of the sine wave is shown below in figure 3. A discrete time low-pass filter with a cutoff frequency of 714Hz was designed using a Simulink model for the input and output workspaces corresponding to simin and simout respectively. The Simulink model is shown in figure 4. The output workspaces were tested in real time on Tektronix oscilloscope in figure 5.

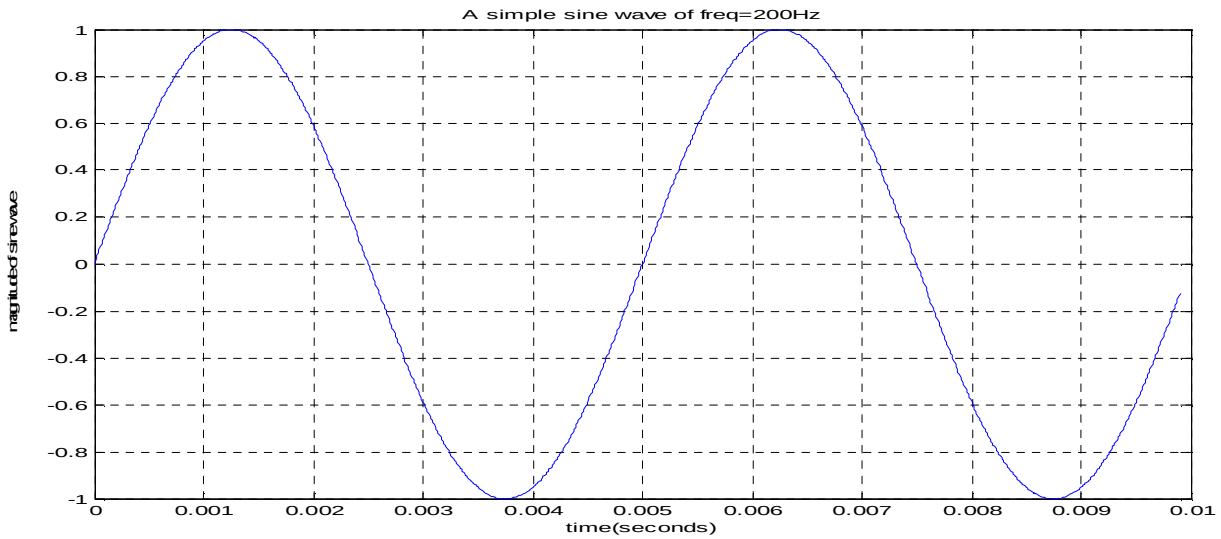

Figure 3 is the input sine wave

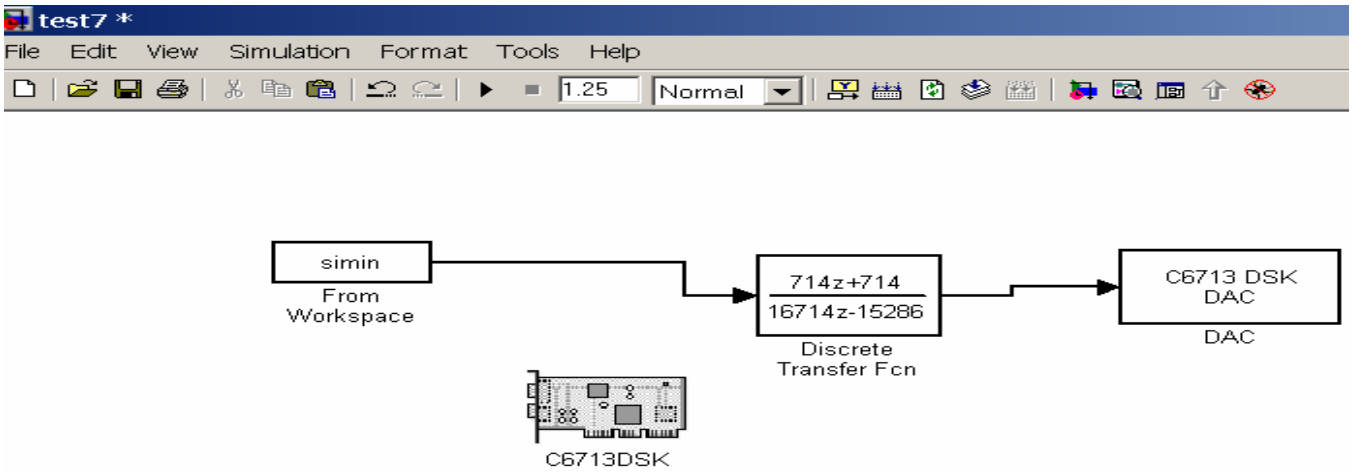

Figure 4 is the Simulink Model for input/output workspaces

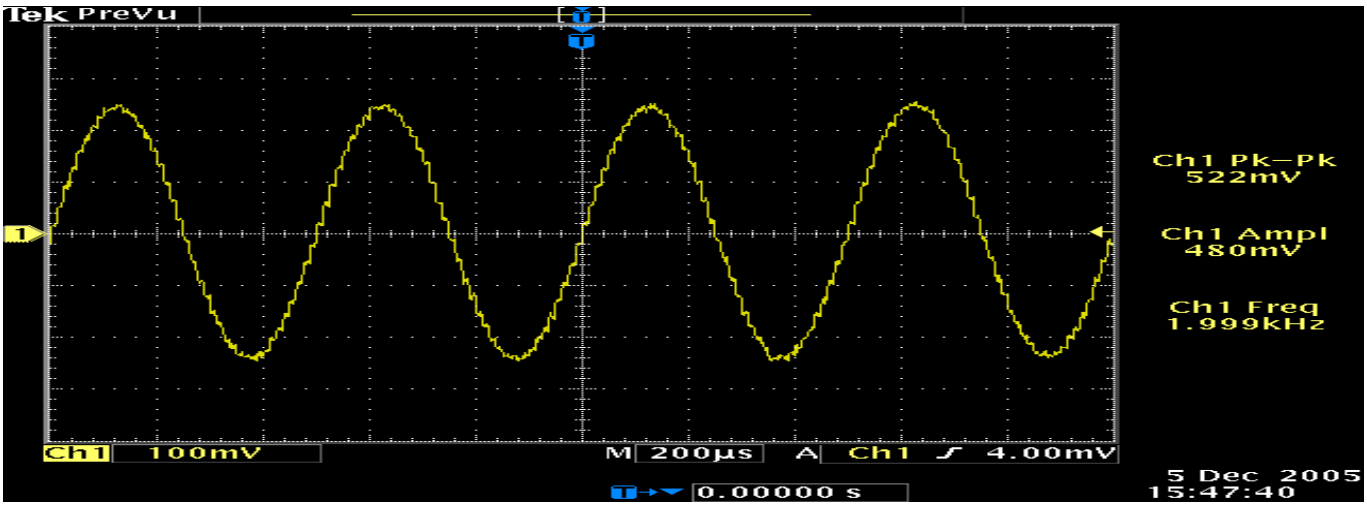

Figure 5 is the Simout plot from oscilloscope

The DSP board was first tested by sending varying frequencies of different wave input (sine, square, and ramp) as an input from a functional generator to the LINE IN connector on the board. Using the Simulink model in figure 6 below, we observed the output from the DSP through the LINE OUT connector and oscilloscope in figure 7.

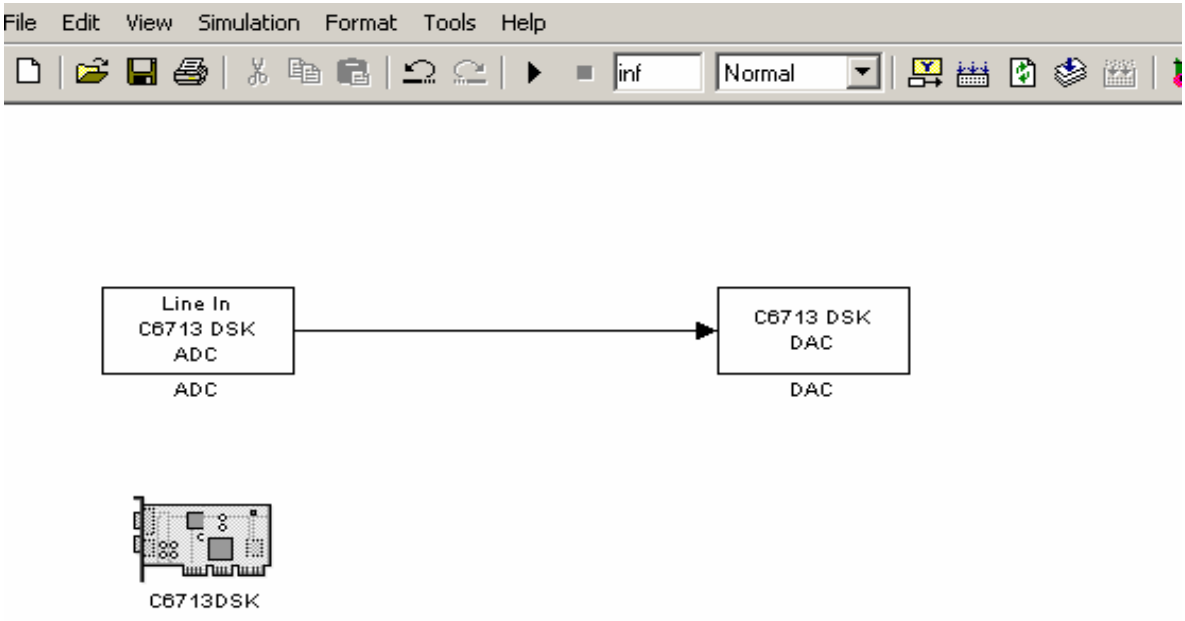

Figure 6 is the Simulink Model for testing the DSP board

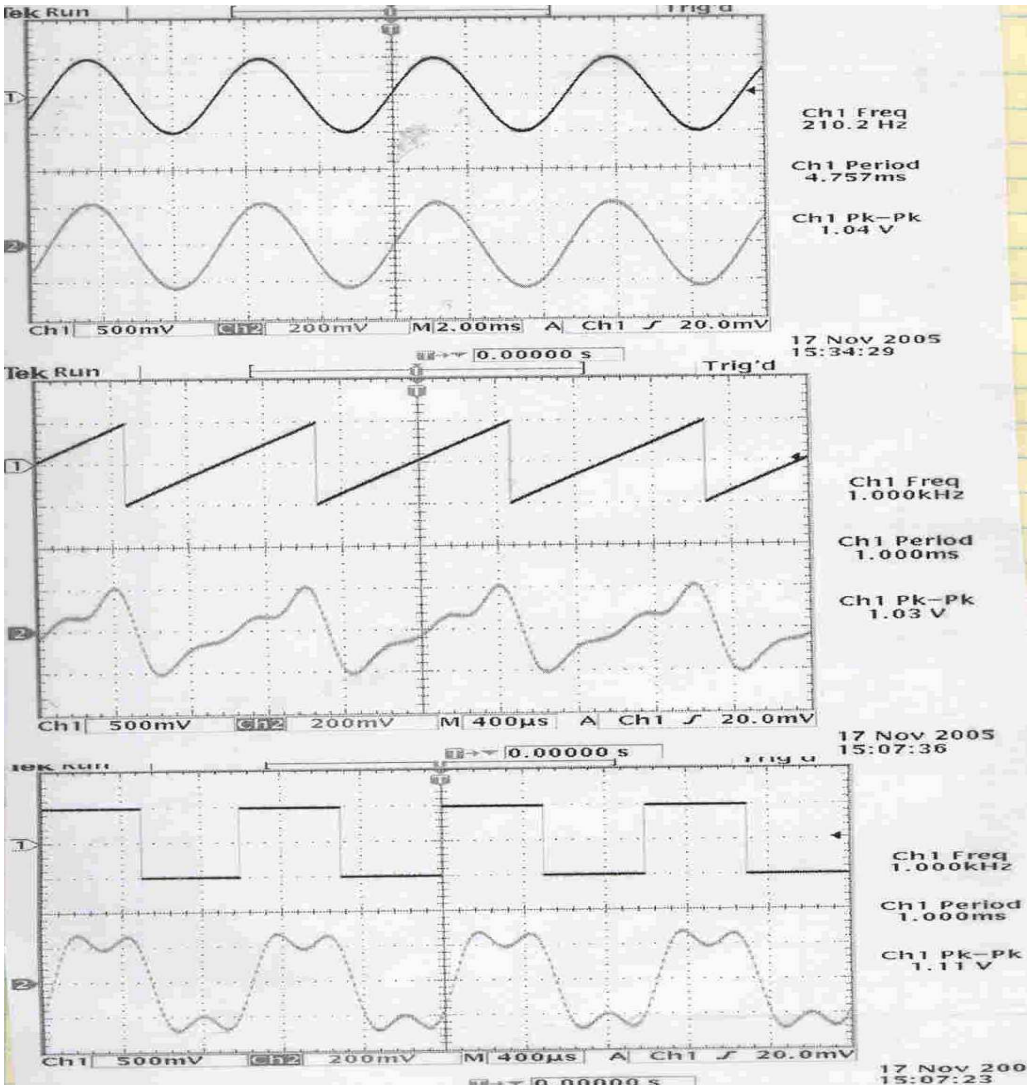

Figure 7 is the output obtained from the DSP board for input test signals

Project Schedule (Next Semester)

WEEK	CHARLES	KWADWO
12/06/2005	ORAL PRESENTATION	ORAL PRESENTATION
01/21/2006	FIR filter implementation	FIR filter implementation
01/27/2006	FIR filter implementation	FIR filter implementation
02/4/2006	IIR filter implementation	IIR filter implementation
02/11/2006	IIR filter implementation	IIR filter implementation
02/18/2006	AM/FM/QAM	AM/FM/QAM
02/25/2006	AM/FM/QAM	AM/FM/QAM
03/02/2006	AM/FM/QAM	AM/FM/QAM
03/02/2006	AM/FM/QAM	AM/FM/QAM
03/09/2006	Writing C/C++ code and using MEX-files	Writing C/C++ code and using MEX-files
03/16/2006	Writing C/C++ code and using MEX-files	Writing C/C++ code and using MEX-files
03/23/2006	Writing C/C++ code and using MEX-files	Writing C/C++ code and using MEX-files
03/30/2006	Writing C/C++ code and using MEX-files	Writing C/C++ code and using MEX-files
04/6/2006	Writing C/C++ code and using MEX-files	Writing C/C++ code and using MEX-files

Bibliography

1. Digital Spectrum INC TMS320C3713 DSK Technical Reference
2. J. G. Proakis, and D. G. Manolakis *Digital Signal Processing,* 3rd Edition.
3. L.W Couch, II , *Digital and Analog Communication Systems*, 6th Edition, Prentice Hall, Upper Saddle River, NJ

Equipment and Parts

1. Texas Instruments DSP Starter Kit(DSK) board TMS320C6713
2. Code Composer Studio software package.