ECE4703 Real-Time DSP Orientation Lab

D. Richard Brown III
Associate Professor
Worcester Polytechnic Institute
Electrical and Computer Engineering Department
drb@ece.wpi.edu

25-Oct-2006
C6713 DSK Overview

- 225 MHz TMS320C6713 **floating point** DSP
- AIC23 stereo codec (ADC and DAC)
  - Ideal for audio applications
  - 8-96 kHz sample rates
  - Line in/out (use these most of the time)
  - Built-in headphone amplifier
  - Built-in microphone preamplifier
- General purpose I/O
  - 4 LEDs
  - 4 DIP switches
- Memory
  - 16 MB dynamic RAM
  - 512 kB nonvolatile FLASH memory
- USB interface to PC
C6713 DSK Physical Layout

- Microphone input
- Line input (stereo)
- Line output (stereo)
- Headphone output (stereo)
- Codec
- CPLD
- 10MB SDRAM
- Flash memory
- DSP 225MHz
- DC power input
- USB port
- DIP switches
- LEDs
- Reset switch
Is my DSK working?

DSK Power On Self Test

- Power up DSK and watch LEDs
- Power On Self Test (POST) program stored in FLASH memory automatically executes
- POST takes 10-15 seconds to complete
- All DSK subsystems are automatically tested
- During POST, a 1kHz sinusoid is output from the AIC23 codec for 1 second
  - Listen with headphones or watch on oscilloscope
- If POST is successful, all four LEDs blink 3 times and then remain on
Is my DSK working?

DSK Diagnostic Utility

- Use DSK diagnostic utility to test DSK functionality

Press start

ok!
Code Composer Studio IDE

- Integrated Development Environment
  - Editor
  - Compiler (C and ASM, with optional optimization)
  - Linker
  - Load binaries to “target” (the DSK)
  - Debugger
  - Performance analyzer
  - Graphics
  - etc...

- How to get started:
  - Connect power supply to DSK
  - Wait for POST to complete
  - Connect USB cable from PC to DSK
  - Launch Code Composer Studio C6713 DSK
  - CCS will load and wait for your input
Code Composer Studio IDE
Useful TI documentation (installed on the hard drive):
SPRU509F.PDF  CCS v3.1 IDE Getting Started Guide
C6713DSK.HLP  C6713 DSK specific help material

Note that we are using CCS v3.1. When consulting documentation, be sure to check the version of CCS for which the documentation was written.
Connecting to the C6713 DSK

The target is now connected.
Opening an Existing Project

Project->Open

Select a .PJT file and press “Open”. Chassaing example projects should be in c:\myprojects\ Other example projects for the C6713 can be found in c:\myprojects\examples\dsk6713 You also have several example projects on your Kehtarnavaz textbook CD.
Compiling/Building a Project

Project->Build (F7)
Loading and Running a Project on the C6713 DSK

File-> Load Program (ctrl+L)

Select the .out file in the project\Debug directory. Program is sent to DSK.

Debug->Run (F5 or the Run button)

Program is sent to DSK.
Halting a Running Program on the C6713 DSK

Debug->Halt (shift+F5 or the Halt button).
Fixing Problems with the Chassaing/Kehtarnavaz Examples

- During compilation, the compiler can’t find some header (.h) files?
  - Fix: Add an item to the CCS search path

- During compilation, the linker can’t find some libraries?
  - Fix: Remove hard links to libraries and add libraries manually to the project

- During compilation, you get warnings about “far calls” to data?
  - Fix: Set the memory model to “data=far”
Add C:\CCStudio_v3.1\C6000\dsk6713\include to the search path

Project -> Build Options
Removing hard links to libraries

Problem is caused by a bad path for the include libraries in the linker options (Project -> Build Options -> Linker tab)

A fix for this is do remove rts6700.lib, DSK6713bsl.lib, and csl6713.lib from the linker options and add these files manually (Project -> Add files to Project...)

Build Complete.
Change the memory model to "data=far"

Project -> Build Options
Creating a New Project (1 of 5)

1. Create new project

   Project->New

   ![Project Creation Window](image)
Creating a New Project (2 of 5)

2. Write your C code:
   File->New->Source File

3. Save it in your project directory (make sure it has a .c extension):
   File->Save

4. Add your C code to the project:
   Project->Add Files to Project
Creating a new project (3 of 5)

5. Add required support files to project
   Project->Add Files to Project
   a) A linker command file (.cmd)
      [you can find good examples of these in your Kehtarnavaz textbook CD]
   b) c6000\cgtools\lib\rts6700.lib
      [run-time support library functions - required]

6. Add optional support files to project, e.g.
   Project->Add Files to Project
   a) vectors.asm (from your Kehtarnavaz CD)
      [used to set up interrupt vectors]
   b) c6000\dsk6713\lib\dsk6713bsl.lib
      [DSK board support library functions – useful for interfacing to the codec, DIP switches, and LEDs]
   c) c6000\csl\lib\csl6713.lib
      [chip support library functions]

Note that the optional support files are not needed for all projects.
Creating a New Project (4 of 5)

7. Set up the build options for C6713:

   **Project -> Build Options**
   (compiler tab)

   - Make sure target version is C671x
   - Also make sure opt level is “none” (this will help with debugging)
Creating a New Project (5 of 5)

8. Scan all file dependencies to automatically bring all header files and includes into the project:
   Project -> Scan all file dependencies

9. Build the project:
   Project -> Build

10. If successful, load the .out file to the DSK:
    File -> Load Program
    Select the Debug directory. Select the .out file.

11. Run it:
    Debug -> Run or F5 or the run button.
Optional: Suppress linker warnings

Project->Build Options (linker tab)

Uncheck “warn about output sections” (or put in values for stack and heap)
Important Reference Material

- Kehtarnavaz Chapter 4
- Kehtarnavaz CD
- Chassaing examples in c:\myprojects directory
- CCS Help system
- **SPRU509F.PDF** CCS v3.1 IDE Getting Started Guide
- **C6713DSK.HLP** C6713 DSK specific help material
- Spectrum Digital TMS320C6713 DSK reference

Latest TI documentation available at
http://www.ti.com/sc/docs/psheets/man_dsp.htm