Digital Signal Processing and Applications with the **TMS320C6713 DSK** Day 1

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

October 15-16, 2007



Workshop Goals

- Become familiar with
 - DSP basics
 - TMS320C6713 floating point DSP architecture
 - TMS320C6713 DSP starter kit (DSK)
 - Code composer studio integrated development environment (IDE)
 - Matlab design and analysis tools
- Learn how to program the C6713
 - Writing and compiling code
 - Fixing errors
 - Downloading code to the target and executing
 - Debugging
- Write and run useful programs on the C6713 DSK
- Learn about DSP applications
- Learn where to find help





Take Home Items



• "Digital Signal Processing and Applications with the C6713 and C6416 DSK" by Rulph Chassaing, 2005

• Texas Instruments TMS320C6713 DSK including

- DSK board with TMS320C6713 DSP chip
- USB cable
- Power supply
- CD with Code composer studio IDE (v3.1) and electronic documentation
- DSK technical reference manual
- DSK quick start installation guide
- Matlab/Simulink trial CD and other promotional material



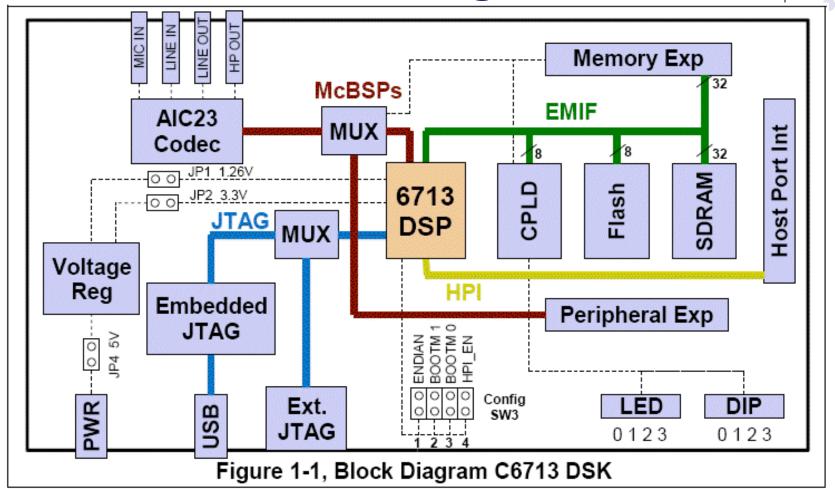
C6713 DSK Overview

- 225 MHz TMS320C6713 *floating point* DSP
- AIC23 stereo codec (ADC and DAC)
 - Ideal for audio applications
 - 8-96 kHz sample rates
- Memory
 - 16 MB dynamic RAM
 - 512 kB nonvolatile FLASH memory
- General purpose I/O
 - 4 LEDs
 - 4 DIP switches
- USB interface to PC



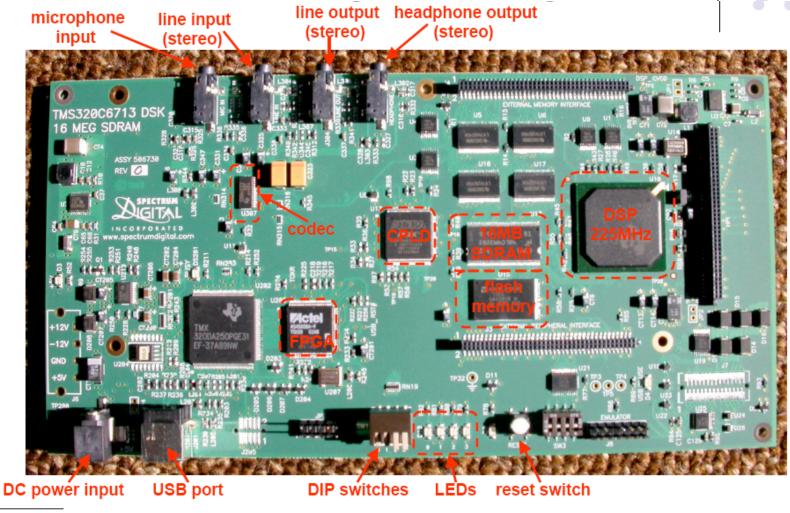


C6713 DSK Functional Block Diagram





C6713 DSK Physical Layout





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

- Install CCS 3.1
 - Directions in "Quick Start Installation Guide"
 - Diagnostic utility automatically installed

| p 671 3DSK Diagnostics | | | 50 671 3DSK Diagnostics | |
|---|---|-----------------|--|---|
| General Advanced Overall Diagnostic Test Diagnostic Status: © USB Diagnostics Diagnostic Status: © DSP Diagnostics DSK: © External Memory Component Value © IED Diagnostics Diagnostics © LED Diagnostics Diagnostics | About Start Stop Beset Emu Reset DSK Save As | _press start | General Advanced Overall Diagnostic Test Diagnostic Status: © USB Diagnostics Diagnostic Status: © DSP Diagnostics DSK: © DSP Diagnostics DSK: © Flash Diagnostics Component Value © Codec Diagnostics 0 © LED Diagnostics CPLD Version 2 © Diagnostics Diagnostics 0 © Diagnostics DSK: 0 © Flash Diagnostics Component Value © Diagnostics 0 0 © Diagnostics 0 0 | About <u>Start</u> <u>Reset Er</u> <u>Reset D</u> <u>Save ≜</u> |
| Diagnostic Results | | | Diagnostic Results | <u>⊣</u> eip |





6713 DS

Code Composer Studio IDE

- Connect power supply to DSK
- Wait for POST to complete
- Connect USB cable from PC to DSK
 - If this is the first time connecting the DSK, you may be asked to install a driver. The driver is on the Code Composer Studio CD and will automatically be found by Windows if the CD is in the drive.
- Launch Code Composer Studio C6713 DSK —
- CCS will load and wait for your input

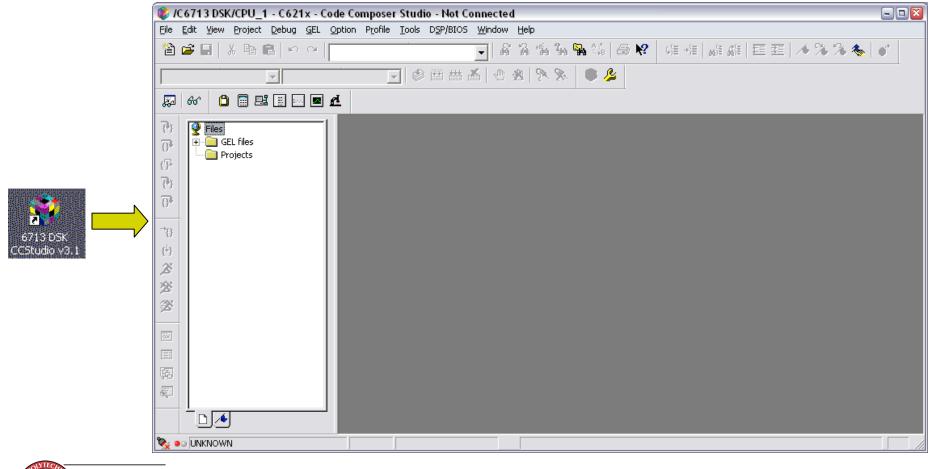








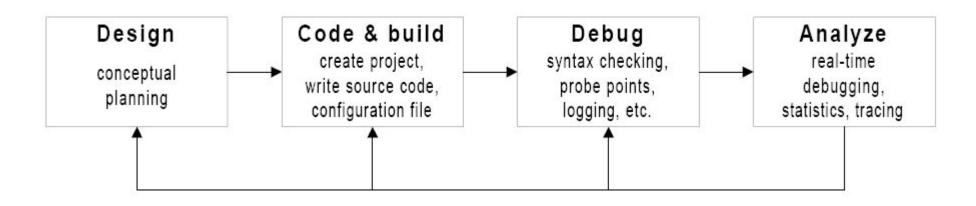
Code Composer Studio IDE





CCS Integrated Development Environment





<u>Useful TI documentation (available online or on your hard drive)</u>: **SPRU509F.PDF** CCS v3.1 IDE Getting Started Guide **C6713DSK.HLP** C6713 DSK specific help material

Note that your DSK includes CCS v3.1. Updates and patches are available after registering CCS.





Connecting to the C6713 DSK

| 😻 /C6713 DSK/CPU_ | 1 - C621x - Code Composer | Studio - Not Connec | ted | 😻 /C6713 DSK/CPU_1 - C671x - Code Composer Studio |
|------------------------|------------------------------|---------------------|-----------|---|
| File Edit View Project | Debug GEL Option Profile | Tools DSP/BIOS Wind | ow Help | File Edit View Project Debug GEL Option Profile Tools DSP/BIOS Window Help |
| 웥 🚅 🔚 🐰 🖻 | Breakpoints | | A A 4 | |
| | Probe Points | | | 2 2 日 米 時 6 9 9 1 |
| | . Step Into | F11 | D 🕸 🗫 🛙 | |
| 🔊 60' 📋 🖩 🗉 | s Step Over | F10 | | |
| | Step Out | Shift+F11 | | 🛛 💭 60° 📋 🖽 🖽 🖂 🗖 🕰 |
| Te P Files | Run | F5 | | |
| GEL files | Halt | Shift+F5 | | 🕑 🔮 Files 🔤 Disassembly 🖃 🖬 🖾 |
| Projects | Animate | Alt+F5 | | Image: Constraint of the second sec |
| (P | Run Free | Ctrl+F5 | | |
| | Run to Cursor | Ctrl+F10 | | 000008C8 20000390 [B0] → 000008CC 00008000 |
| a | Set PC to Cursor | Ctrl+Shift+F10 | | |
| U. | Multiple Operation | | | Image: Contract of the |
| -38 | Assembly/Source Stepping | • | | 30 |
| (†) | Reset CPU | Ctrl+R | | 000000000000000000000000000000000000000 |
| | Restart | Ctrl+Shift+F5 | | ↔ 000008E0 00006000 000008F4 0210805A ✓ |
| 1 A S | Go Main | Ctrl+M | | |
| * | Reset Emulator | Ctrl+Shift+R | | ※ |
| 28 | Connect | Alt+C | | |
| | Restore Debug State | | | |
| 55K | Enable Thread Level Debuggir | ng | | GEL StartUp Complete. |
| | Real-time Mode | | | |
| 扇 | Enable Rude Real-time Mode | | | |
| | Flush Pipeline on Halt | | | |
| | | | | The target is now connected |
| | | | | |
| | | | Toggle | ALTED |
| | , , , , | | 1.235/0 | |





Opening an Existing Project

| 🈻 /C6713 DSK | /CPU_1 - C671x - Code Composer | r Studio |
|----------------|--|--------------|
| File Edit View | Project Debug GEL Option Profile | Tools D |
| 12 🚔 🗐 🛛 | New | |
| | Open | |
| | Use External Makefile | 🛛 🌮 É |
| | Export to Makefile | |
| 🔊 🚳 🗋 | Add Files to Project | |
| 2. | Save | |
| Piles | Close | ibly |
| 🖓 🛨 🛄 GE | Source Control | 0800 |
| (P | Compile File Ctrl F7 | -D8C4 |
| (5) | Compile File Ctrl F7 Build F7 | 18C8 18CC |
| | Rebuild All | 18D0 |
| 0° | Stop Build | D8D4 |
| | Build Clean | D8D8 |
| 0 | | -18DC |
| (+) | Configurations | DSEO |
| 答 | Build Options File Specific Options | 18F4 |
| 295 | Function Level Options | |
| | | |
| 🌋 É 🖸 🗸 | Project Dependencies | |
| | Show Project Dependencies | |
| GEL Startl | Show File Dependencies | |
| | Scan All File Dependencies | |
| | Recent Project Files | |

Project->Open

| Project Open | | | ? 🔀 |
|--------------------------|-----------------------------------|---|--|
| Look in: 🗀 myproject | \$ | - 🗕 🗗 🗖 | |
| 🚞 Adaptc 🚞 AdaptIDFIR | 🚞 Adaptpredict 🛅 Adaptpredict_2IN | bios_sine_ctrl bios_sine_intr | |
| AdaptIDFIRW | 🚞 Aliasing | 🛅 bpsk | b 🖾 |
| Adaptinoise | BeatDetector | i Code_casm | |
| 🛅 Adaptnoise_2IN | 🛅 bios_4led | 🛅 DFT | ⊂ ⊃ C |
| < | | | > |
| File <u>n</u> ame: | | | <u>O</u> pen |
| Files of type: Project F | iles (*.pit) | • (| Cancel |
| | | | <u>H</u> elp |
| | Look jn: myproject | Look jn: myprojects Adaptc AdaptDFIR AdaptIDFIR AdaptIDFIRW AdaptIDFIRW AdaptIDIIR AdaptIDIR Adaptnoise IBeatDetector Adaptnoise_2IN File name: | Look in: myprojects Image: Control of type: Adaptpredict Image: Control of type: Image: Control of type: |

Select a .PJT file and press "Open". Chassaing example projects should be in

c:\CCStudio_v3.1\myprojects\

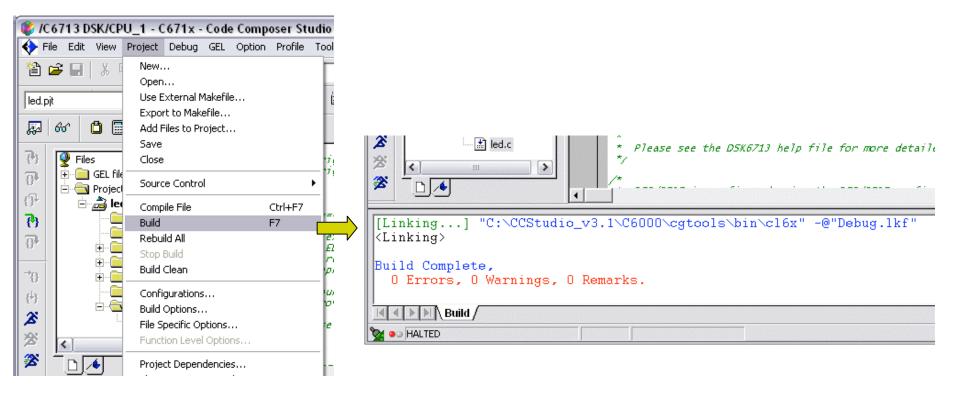
Other example projects for the C6713 can be found in c:\CCStudio_v3.1\examples\dsk6713





Compiling/Building a Project

Project->Build (F7)

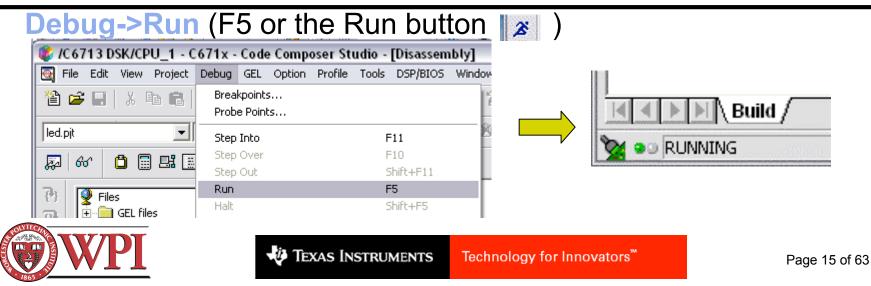




Loading and Running a Project on the C6713 DSK

| F | | e-> Loa | ad Progr | ram (o | trl+L) | Load Progra | am | | | ? 🛛 |
|---|-----|---------------|------------------|-------------|----------|--------------------|-------|----------|-------|--------------|
| k | | | PU_1 - C671x - C | · · · · · · | <u> </u> | Look jn: 📔 | Debug | <u> </u> | • 🗢 🗈 | * 🖩 |
| F | 0 | | Project Debug (| | | ed.out | | | | |
| I | 쎹 | New | | • 🗖 | | | | | | |
| L | _ | Open | Ctrl+O | · L | _ | | | | | |
| Ш | lea | Close | | | T. | | | | | |
| H | - | Save | Ctrl+S | | 1 | | | | | |
| Ш | 2 | Save As | | | | | | | | |
| H | _ | Save All | | | ★ | File <u>n</u> ame: | 1 | | | <u>O</u> pen |
| | Ð | Load Program. | Ctrl+L | ľ | * / . | Files of type: | *.out | | • | Cancel |
| Ш | Ð | | | L 201. A 1 | * * / | | | | | <u>H</u> elp |
| | 0 | Reload Progra | n Ctrl+Sł | hirt+L | */ | | | | | |

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



Halting a Running Program on the C6713 DSK



Debug->Halt (shift+F5 or the Halt button).

| 😻 /C6713 DSK/CPU_1 - C671x - Code Composer Studio - [Disassembly] | | | | | | |
|---|-----------------------------|--------------------------|------------|--|--|--|
| 🧕 File Edit View Project | Debug GEL Optio | n Profile Tools DSP/BIOS | Window H | | | |
| 🏠 🚅 🖬 🕺 🖻 💼 | Breakpoints Probe Points | | A ' | | | |
| led.pjt | Step Into F11 | | | | | |
| 💹 60' 🗂 🖩 🖽 🔚 | Step Over | F10 | | | | |
| | Step Out | Shift+F11 | | | | |
| 🕑 👰 Files | Run | F5 | | | | |
| | Halt | Shift+F5 | | | | |
| GEL files | Animate | Alt+F5 | | | | |



Chassaing textbook examples: Fixing the search path

Add C:\CCStudio_v3.1\C6000\dsk6713\include to the search path

Project -> Build Options -> [Compiler tab] -> [Preprocessor category]

| Build Options for Sine8_LED.pjt (Debug) 🔹 👔 👔 | | | | |
|--|---|--|------|--|
| General Compile | er Linker Link Order | | | |
| -i ⁿ 'C:\CCStudio_ | | e8_LED\Debug'' lude''-d''CHIP_6713''-mv6710 | ~ | |
| Category: Basic Advanced Feedback Files Assembly Parser Preprocessor Diagnostics | Preprocessor Include Search Path (-i): Pre-Define Symbol (-d): Undefine Symbol (-u): Preprocessing: Continue with Comp | Judio_v3.1\C6000\dsk6713\in CHIP_6713 None | | |
| | 01 | Cancel | Help | |





Chassaing textbook examples: Fixing the mem model

Change the memory model to "data=far"

Project -> Build Options -> [Compiler tab] -> [Advanced category]

| Build Options for Sine8_LED.pjt (Debug) 🔹 👔 👔 | | | | | | | |
|--|---|---|-------|--|--|--|--|
| General Compiler Linker Link Order | | | | | | | |
| -g -s -fr''C:\CCStudio_v3.1\MyProjects\sine8_LED\Debug'' -i''C:\CCStudio_v3.1\C6000\dsk6713\include'' -d''CHIP_6713'' -mv6710 mem_model:data=far | | | | | | | |
| Category: | Advanced | | | | | | |
| Basic Advanced | RTS Modifications: | Defns No RTS Funcs 🔹 | | | | | |
| Feedback | Auto Inline Threshold (-oi) | | | | | | |
| Files Assembly | Endianness: | Little Endian 🖃 | | | | | |
| Parser Preprocessor | Memory Models: | Far (mem_model:data=far) | - | | | | |
| Diagnostics | RTS Calls: | Far Aggregate Far (mem_model:data=far) | | | | | |
| | Aliasing: | Near (mem_model:data=near) | | | | | |
| | Interrupt Threshold (-mi): | | | | | | |
| | 🔲 Speculate Threshold (| -mh): | | | | | |
| Turn Off Software Pipelining (-mu) | | | | | | | |
| | 🔲 Old 6400 Alignment C | ompatibility (-mb) | | | | | |
| | Turn Off Reorder of Associative Floating Pt Ops (-mc) | | | | | | |
| | 🖵 Use Function Subsections (-mo) | | | | | | |
| Historic C Pointer to Const Alias Disambiguation (-ox) | | | | | | | |
| | | Cancel He | eln l | | | | |
| | | | -φ. | | | | |





Things to Try

- Open Sin8_LED project and fix the search path and the memory model (see previous pages). Then build, load, and run it.
 - Press DIP switch 0. You should see LED 0 light up and a 1kHz sinusoid should appear on the left channel of the codec. This is a good test to see if the DSK is working.
- Make an error in the source code Sin8_LED.c and build the project to see what happens.
- Change the amplitude of the sinusoid (gain variable), rebuild, reload, and see what happens.
- Modify the code to generate a 500Hz sinusoid.
- Open, build, and load other projects in "myprojects"





Creating a New Project (1 of 5)

- 1. Create new project
 - **Project->New**

| Project Creation | on 🛛 🔊 |
|-----------------------|-------------------------------------|
| Project <u>N</u> ame: | helloworld |
| Location: | C:\CCStudio_v3.1\MyProjects\hellowo |
| <u>P</u> roject Type: | Executable (.out) |
| <u>T</u> arget | TMS320C67XX |
| | < <u>B</u> ack Finish Cancel Help |





Creating a New Project (2 of 5)

- Write your C code:
 File->New->Source File
- 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) myprojects\support\c6713dsk.cmd [linker command file – this or another cmd file is required]
 - 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) myprojects\support\vectors_poll.asm or vectors_intr.asm [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\bios\lib\csl6713.lib [chip support library functions]





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(imization) Level is "none" - this will help with debugging

| Build Optio | ns for helloworld.pjt (Debug) | ? 🔀 |
|--|---|-----|
| General Co | ompiler Linker Link Order | |
| | i_dir)\Debug'' -d''_DEBUG'' -mv6710 | |
| Category: Basic Advanced Feedback Files Assembly Parser Preprocess Diagnostics | Target Version: C671x (-mv6710) ▼ Generate Debug Info: Full Symbolic Debug (-g) ▼ Opt Speed vs Size: Speed Most Critical (no -ms) ▼ Opt Level: None ▼ | • |
| | OK Cancel Hel | p |



Creating a New Project (5 of 5)

- 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
- 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 in the Basic category)

| Build Options | s for helloworld.pjt (Debug) | ? 🔀 | | | |
|--|---|-----|--|--|--|
| General Comp | piler Linker Link Order | | | | |
| -q -c -m".\Debug\helloworld.map" -o".\Debug\helloworld.out" -x | | | | | |
| Category: Basic Advanced | Advanced Disable Conditional Linking (-j) Disable Debug Symbol Merge (-b) Strip Symbolic Information (-s) Make Global Symbols Static (-h) Warn About Output Sections (-w) Resolve Symbols to First Library (-priority) Disable Size-based Allocation (default_order) XML Link Info File (xml_link_info=): Define Global Symbol (-g): Create Unresolved Ext Symbol (-u): | | | | |
| | OK Cancel He | elp | | | |



Tip: Problems finding files during linking

| [Loop_store.c] "C:\CCStudio_v3.1\C6000\cgtools\bin\c | :16x" -g -q -fr"C:∕ |
|--|---|
| [Linking] "C:\CCStudio_v3.1\C6000\cgtools\bin\cl6 <linking> >> C:\DOCUME~1\drb\LOCALS~1\Temp\TI5643, line 21: can't find input file 'DSK6713bsl.lik</linking> | Build Options for Loop_store.pjt (Debug) |
| >> Compilation failure | -q -c -m".\Debug\Loop_store.map" -o".\Debug\Loop_store.out" -x -I"rts6700.lib" -I"DSK6713bsl.lib" -I"csl6713.lib" |
| Build Complete. | |
| Problem is caused by a bad path for the include libraries in the linker options (Project -> Build Options -> Linker tab) | Category: Basic Basic Suppress Banner (-q) Advanced Exhaustively Read Libraries (-x) Output Module: Image: Comparison of the state of the |
| A fix for this is to remove rts6700.lib, DSK6713bsl.lib, and csl6713.lib from the linker options and add these files manually (Project -> Add files to Project) | Heap Size (-heap): Stack Size (-stack): Fill Value (-f): Code Entry Point (-e): Library Search Path (-i): Include Libraries (-I): rts6700.lib;DSK6713bsl.lib;csl6713.lib |
| TEXAS INSTRUMENTS | OK Cancel Help |

A Simple Program to Try: "helloworld"

// helloworld.c
// D. Richard Brown III
// 9-Oct-2006

#include <stdio.h>

void main() {

}

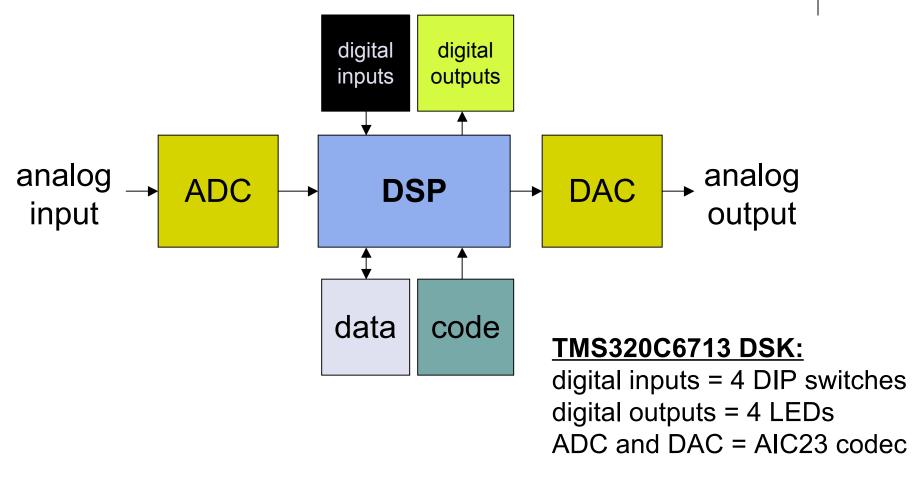
printf("Hello world.\n");







More Interesting Programs: Interfacing with the Real World





Interfacing with the DIP Switches and LEDs

LED and DIP switch interface functions are provided in **dsk6713bsI.lib**.

```
Initialize DIP/LEDs with
DSK6713_DIP_init() and/or DSK6713_LED_init()
Read state of DIP switches with
DSK6713_DIP_get(n)
Change state of LEDs with
DSK6713_LED_on(n) or
DSK6713_LED_off(n) or
DSK6713_LED_toggle(n)
where n=0, 1, 2, or 3.
```

Documentation is available in C:\CCStudio_v3.1\docs\hlp\c6713dsk.hlp



Interfacing with the AIC23 codec: C6x Interrupt Basics

- Interrupt sources must be mapped to interrupt events
 - 16 "interrupt sources" (timers, serial ports, ...)
 - 12 "interrupt events" (INT4 to INT15)
- Interrupt events have associated "interrupt vectors". An "interrupt vector" is a special pointer to the start of the "interrupt service routine" (ISR).
- Interrupt vectors must be set up in your code (usually in the file "vectors.asm").
- You are also responsible for writing the ISR.





Setting up an interface with the AIC23 Codec (step 1 of 3)



We can write the ISR first:

```
49 interrupt void serialPortRcvISR()
50 {
51 Uint32 temp;
52
53 temp = MCBSP_read(DSK6713_AIC23_DATAHANDLE); // read L+R channels
54 MCBSP_write(DSK6713_AIC23_DATAHANDLE,temp); // write L+R channels
55 }
```

Remarks:

- MCBSP_read() requests samples from the codec's ADC
- MCBSP_write() sends samples to the codec's DAC
- This ISR simply reads in samples and then sends them back out.



Codec data format and how to separating the left/right channels

// we can use the union construct in C to have
// the same memory referenced by two different variables
union {Uint32 combo; short channel[2];} temp;

temp.channel[0] (short) temp.channel[1] (short)

temp.combo (Uint32)

// the McBSP functions require that we
// read/write data to/from the Uint32 variable
temp.combo = MCBSP_read(DSK6713_AIC23_DATAHANDLE);
MCBSP_write(DSK6713_AIC23_DATAHANDLE, temp.combo);

```
// but if we want to access the left/right channels individually
// we can do this through the short variables
Leftchannel = temp.channel[1];
Rightchannel = temp.channel[0];
```



Setting up an interface with the AIC23 Codec (step 2 of 3)

- Now we can set up the interrupt vector to point to the ISR.
- In this example, our ISR is called "serialPortRcvISR".
- We will link the codec interrupt event to INT15.
- Here is the appropriate code in the **vectors.asm** file:

| 150 | INT15: | | | |
|-----|--------|-----|--------------------|----|
| 151 | MVKL | .S2 | _serialPortRcvISR, | BØ |
| 152 | MVKH | .S2 | _serialPortRcvISR, | BØ |
| 153 | В | .S2 | BØ | |
| 154 | NOP | | | |
| 155 | NOP | | | |
| 156 | NOP | | | |
| 157 | NOP | | | |
| 158 | NOP | | | |





Setting up an interface with the AIC23 Codec (step 3 of 3)



| Initialization steps: | 21 int | terrupt void serialPortRcvISR(| void); | // ISR functio | n prototype |
|---|--|--|---|---|-------------|
| Initialize the DSK Open the codec with the default configuration. Configure multi- channel buffered serial port (McBSP) Configure codec parameters, e.g. set the sampling rate | | <pre>id main() DSK6713_init(); // Initia hCodec = DSK6713_AIC23_openCo // Configure buffered serial // This allows transfer of bo MCBSP_FSETS(SPCR1, RINTM, FRI MCBSP_FSETS(SPCR1, XINTM, FRI MCBSP_FSETS(RCR1, RWDLEN1, 3 MCBSP_FSETS(XCR1, XWDLEN1, 3 DSK6713_AIC23_setFreq(hCodec</pre> | <pre>odec(0, &config); ports for 32 bit operation oth right and left channels 4); 4); 2BIT); 2BIT);</pre> | // Open the co n s in one read/wr | dec |
| Configure and enable interrupts Do normal processing (we just enter a loop here) | 36 37 38 39 40 41 42 43 44 45 46 47 | <pre>// Interrupt setup IRQ_globalDisable(); IRQ_nmiEnable(); IRQ_map(IRQ_EVT_RINT1,15); IRQ_enable(IRQ_EVT_RINT1); IRQ_globalEnable(); while(1) { }</pre> | | errupt physical interru terrupts | |



Setting the Sampling Rate

Here we open the codec with the default configuration:

26 hCodec = DSK6713_AIC23_openCodec(0, &config); // Open the codec

The structure "config" is declared in dsk6713_aic23.h

Rather than editing the header file, we can change the sampling frequency after the initial configuration:

DSK6713_AIC23_setFreq(hCodec, DSK6713_AIC23_FREQ_48KHZ); // set the sampling rate

Frequency definitions are in dsk6713_aic.h

| ∕* Frequ | uency Definitions */ | |
|----------|--------------------------|---|
| #define | DSK6713_AIC23_FREQ_8KHZ | 1 |
| #define | DSK6713_AIC23_FREQ_16KHZ | 2 |
| | DSK6713_AIC23_FREQ_24KHZ | 3 |
| #define | DSK6713_AIC23_FREQ_32KHZ | 4 |
| #define | DSK6713_AIC23_FREQ_44KHZ | 5 |
| #define | DSK6713_AIC23_FREQ_48KHZ | 6 |
| #define | DSK6713_AIC23_FREQ_96KHZ | 7 |





Other Codec Configuration

- Input volume (individually controllable for left and right channels)
- Headphone output volume (individually controllable for left and right channels)
- Digital word size (16, 20, 24, or 32 bit)
- Other settings, e.g. byte order, etc. For more details, see:
 - dsk6713_aic23.h
 - Codec datasheet (TLV320AIC23B)
 - C:\CCStudio_v3.1\docs\hlp\c6713dsk.hlp



Some Things to Try

- Make a new project that:
 - Polls DIP switch 0. If pressed, light up all four LEDs.
 - Sets the sampling rate of the AIC23 codec to 44.1kHz.
 - Uses an ISR to sample the left and right channels.
 - Multiplies the left and right channels by a variable gain.
 - Outputs the modified samples to the left and right channels.
- Bonus: Swap the channels, i.e. Left_in -> Right_out, Right_in -> Left_out, when DIP switch 0 is pressed.
- Bonus: Try changing the input/output volumes (hint: look at default configuration in dsk6713_aic23.h)





Lunch Break

Workshop resumes at 1:30pm...



Page 38 of 63

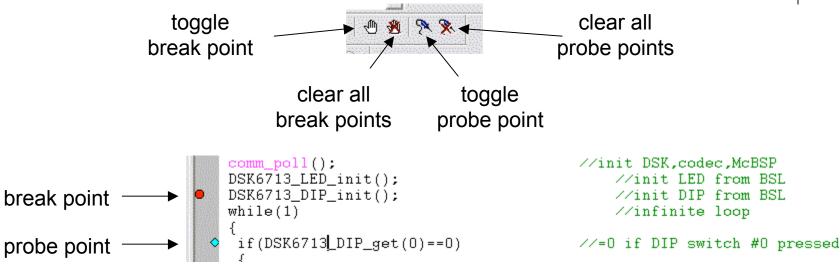
Debugging and Other Useful Features of the CCS IDE

- Breakpoints
- Probe points
- Watch variables
- Plotting arrays of data
- Animation
- General Extension Language (GEL)





Breakpoints and Probe Points

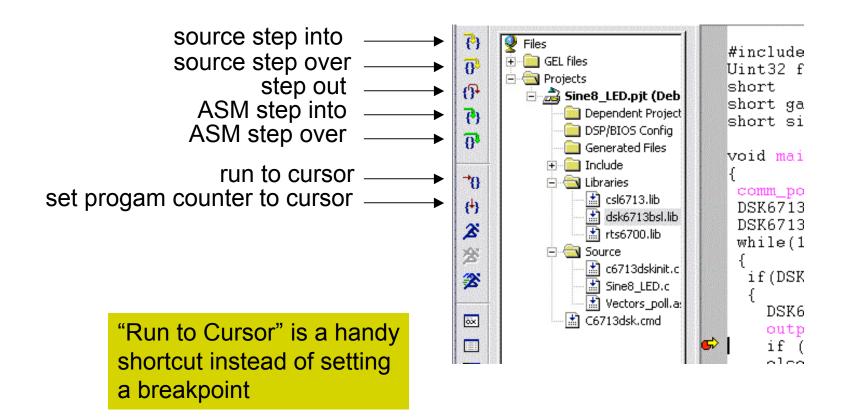


- Breakpoints: stop code execution at this point to allow state examination and step-by-step execution.
- **Probe points**: force window updates and/or read/write samples from/to a file at a specific point in your code.





Breakpoints





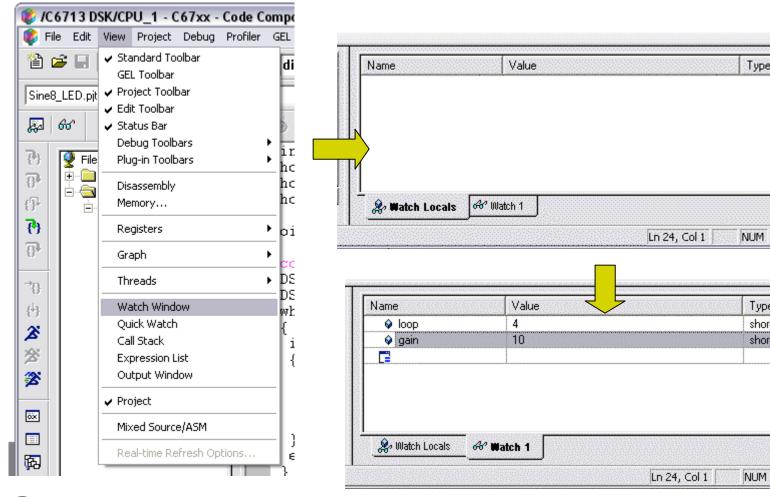
Probe Points



- Differ from breakpoints: Halt the DSP momentarily, perform an action, and then automatically resume execution.
 - Note that this may cause problems with real-time operations.
- Facilitate repeatable testing via automatic file input and/or output (on PC).
- For more details, see CCS Getting Started Guide (SPRU509F.PDF) or CCS help.



Watch Variables







1.1

Radix

Туре

Туре

short

short

Radix

dec

dec

Watch Variables



- In the Watch Locals tab, the debugger automatically displays the Name, Value, and Type of the variables that are *local* to the currently executing function.
- In the Watch tab, the debugger displays the Name, Value, and Type of the local and global variables and expressions that you specify.
- Can add/delete tabs.





Plotting Arrays of Data

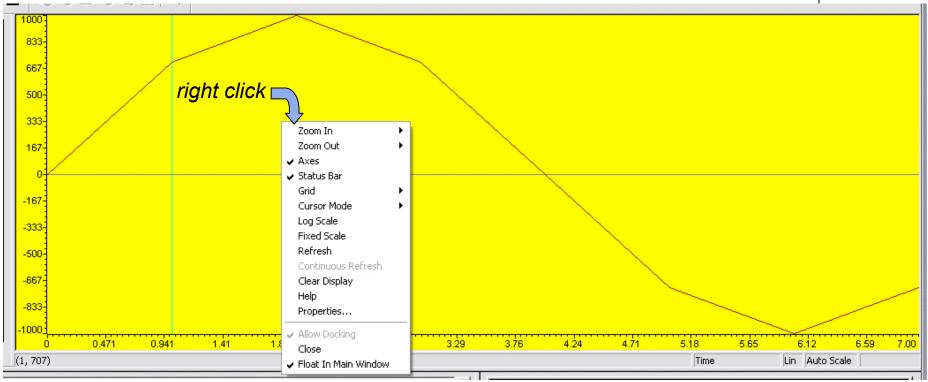
| | 1 | | | | | | | | | | | | |
|---------------|-----|--|---|-----|-----------------------|----------------|----------|------------------------------|------------------------------------|-------------|--------------|---------|----|
| | | | SK. | /CP | U_1 - C | | | | | ser Stu | dio 'C | 6713 | C |
| 🧳 F | ile | Edit | Vie | BM | Project | Debug | Profiler | 9 | EL | Option | Tools | PBC | I |
| 1 | | ✓ Standard Toolbar GEL Toolbar | | | | dip | | | | | | | |
| Sine8_LED.pit | | | ✓ Project Toolbar ✓ Edit Toolbar | | | · 🖉 🗮 着 | | | | | | | |
| F | 60 | 5° | ~ | | itus Bar bug Toolb | arc | | | | _ | b Q | | |
| { ! } | | File | | | g-in Toolb | | | Þ | | t32 f rt | `s = loop | | |
| 8ª | | + <mark> </mark> | Disassembly Memory | | | | | | hort gain = 10; hort sine_table | | | | |
| ው የት የት | | <u> </u> | | | gisters | | | • | L . | d mai | | | Č |
| () () | | | _ | | - | | | | | | | | 1 |
| 0 | | | | Gra | aph | | | • | | ime/Frec | | • | |
| →{} | } | | | Thr | reads | | ١ | Constellation Eye Diagram | | | | i: i | |
| {+} | | | | Wa | itch Wind | ow | | | I | mage | | | [|
| æ | | | | - | ick Watch | I | | | { | | | | |
| | | | | | l Stack | | | | | f(DSK | 6713 | _DI | P. |
| 滏 | | | | • | pression L | | | | { | | | | |
| 2 | | | | Ou | tput Wind | low | | | L . | | 5713_ | | _ |
| õx | | | ~ | Pro | ject | | | | | if (| ut_s loop | < | 7 |
| | | | | Mix | ed Sourc | e/ASM | | | } | else | : loo | p = | |
| ه | | | _ | Re | al-time Re | fresh Op | tions | | | lse I | SK67 | 13_ | L |

| 🔁 Graph Property Diale | og 🛛 🖉 |
|---------------------------|--------------------------------|
| Display Type | Single Time |
| Graph Title | Graphical Display |
| Start Address | sine_table |
| Acquisition Buffer Size | 8 |
| Index Increment | 1 |
| Display Data Size | 8 |
| DSP Data Type | 16-bit signed integer |
| Q-value | 0 |
| Sampling Rate (Hz) | 8000 |
| Plot Data From | Left to Right |
| Left-shifted Data Display | Yes |
| Autoscale | On |
| DC Value | 0 |
| Axes Display | On |
| Time Display Unit | s |
| Status Bar Display | On |
| Magnitude Display Scale | Linear |
| Data Plot Style | Line |
| Grid Style | Zero Line |
| Cursor Mode | Data Cursor |
| | <u>O</u> K <u>Cancel H</u> elp |





Plotting Arrays of Data





Animation





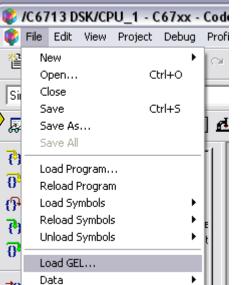
- Runs the program until a breakpoint is encountered.
 - At the breakpoint, execution stops and all windows not connected to any Probe Points are updated.
 - Program execution then automatically resumes
 - Useful for updating graphical displays
 - Note: Animation may cause problems with real-time operation
- Can pause execution at each breakpoint:
 Option->Customize: Debug Properties tab Animate Speed (0-9s) (zero = no pause)

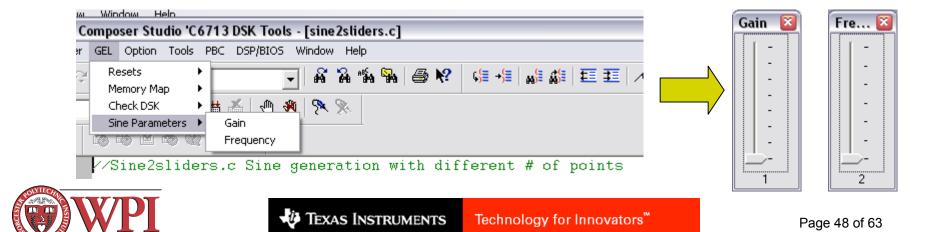




General Extension Language

- Create functions to extend the functionality of Code Composer Studio
- GEL files are not loaded with a project
- Often used to change variables "on-the-fly"
- Examples from Chassaing textbook: sin2sliders.pjt and sin2sliders.gel





General Extension Language

- Useful GEL files can be pretty simple
- From sin2sliders.gel:

```
/*Sine2sliders.gel Two sliders to vary gain and frequency*/
menuitem "Sine Parameters"
slider Gain(1,8,1,1,gain_parameter) /*incr by 1,up to 8*/
{
    gain = gain_parameter; /*vary gain*/
}
slider Frequency(2,8,2,2,frequency_parameter) /*incr by 2,up to 8*/
{
    frequency = frequency_parameter; /*vary frequency*/
}
```

 Syntax details can be found in CCS help: Help->Contents->Making a Code Composer Studio Project -> Building and Running your Project -> Automating Tasks with General Extension Language (GEL)



Some Things to Try



- Try out the debugging tools on the code you wrote in the morning session
 - breakpoints
 - probe points
 - watch variables
 - animation
- Modify your stereo in/out project to have the output gain changeable via a GEL slider
- Try out the CCS plotting tools
 - Modify your code to have a buffer (i.e., store samples in an array) and plot the contents.
- Try to have CCS animate a plot window

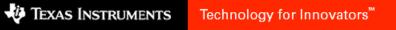


Finite Impulse Response (FIR) Filters

- Frequently used in real-time DSP systems
 - Simple to implement
 - Guaranteed to be stable
 - Can have nice properties, e.g. linear phase
- Input/output relationship

$$y[n] = \sum_{m=0}^{M-1} h[m]x[n-m]$$

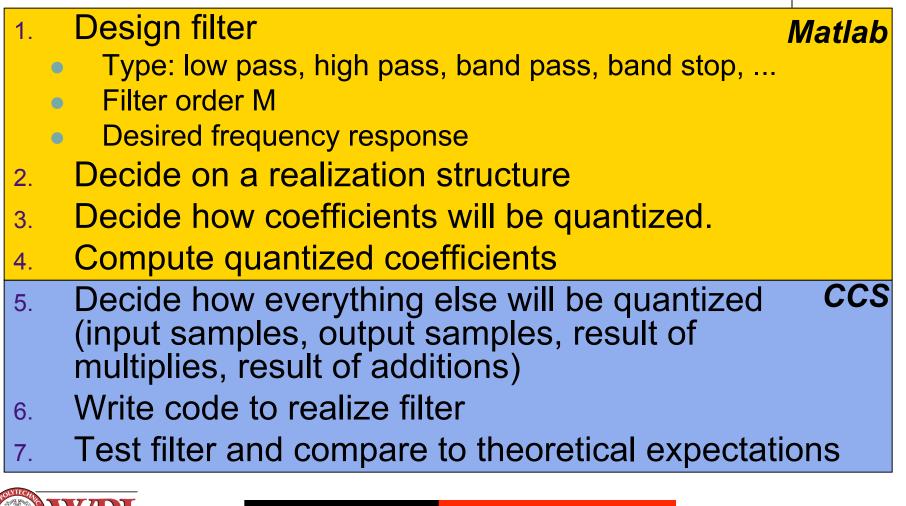
x=input, **y**=output, **h**=filter coefficients, **M**=# of filter coefficients





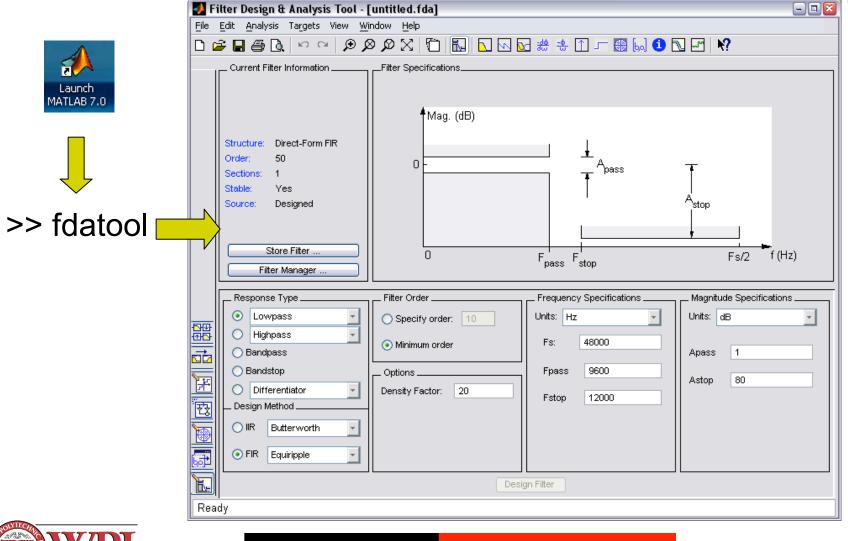


Creating FIR Filters





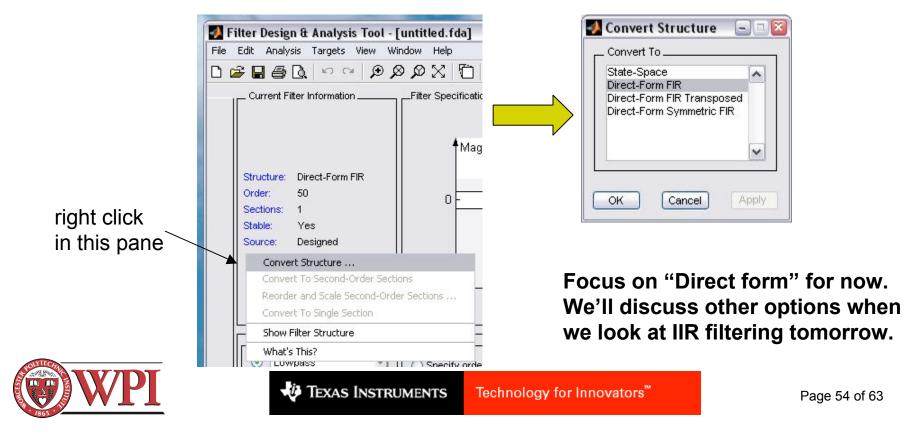
Designing FIR Filters





Filter Realization Structures

- Lots of different structures available
 - Direct form I, direct form II, transposed forms, cascade, parallel, lattice, ...
 - All have same input/output relationship
 - Choice of structure affects computational complexity and how quantization errors are manifested through the filter



| File | Edit Analysis Targets View W | [untitled.fda] indow Help | | 90 |
|--------------------------------------|---|---|---|--|
| 1000 | | The second se | 🐱 🚓 🚓 🗅 🖵 🛞 😡 🚺 | N |
| | Current Filter Information | | | |
| 1 1 日 日 日 日 日 日 | Structure: Direct-Form FIR Order: 50 Sections: 1 Stable: Yes Source: Designed Store Filter Filter Manager Response Type O Lowpass Highpass O Highpass | Filter Order | Fpass Fstop Frequency Specifications Units: Hz Fs: 48000 | A _{stop} Fs/2 f (Hz) Magnitude Specifications |
| | O Bandpass | Options | Fpass 9600 | |
| F | O Differentiator | Density Factor: 20 | Fstop 12000 | Astop 80 |
| TT TT | Design Method | | | |
| 70 23 | | | | |



Make Coefficient File For CCS

| Generate C Header | | | | | | | | | | | | |
|----------------------------|------------------------------------|--|------------|--|------------------|------------------|-----------------------|---------|--|--|--|--|
| | Filter Design & File Edit Analysis | t Analysis Tool - [untitled Targets View Window Help | da * | | _ Variable names | in C header file | | | | | | |
| | 🗅 🖻 🖥 🖨 🕻 | Generate C header | | | Numerator: | В | Numerator length: | BL | | | | |
| | Current Filte | Code Composer Studio (r) ID | SY | | | | | | | | | |
| | | XILINX Coefficient (.COE) File | <u>- 1</u> | ľ | | | | | | | | |
| | | Generate HDL | | | Number of secti | ons: NS | | | | | | |
| | | | - | | | | | | | | | |
| Data type to use in export | | | | | | | | | | | | |
| | | | | Export suggested: Double-precision | | | | | | | | |
| | | | | | | | floating point | | | | | |
| | | | | | C |) Export as: | Signed 32-bit integer | · · · · | | | | |
| | | | | | | | Fractional length: 31 | | | | | |
| | | | | | | OK Cance | I Help Apply | ן | | | | |
| | | | | | | | | | | | | |

Here you can change the coefficient data type to match your desired quantization.



Example DP-FP Coefficient File

```
/*
 * Filter Coefficients (C Source) generated by the Filter Design and Analysis Tool
 *
 * Generated by MATLAB(R) 7.0 and the
 * Generated on: 19-Aug-2005 13:04:09
 *
 */
/*
 * Discrete-Time FIR Filter (real)
 * ------
 * Filter Structure : Direct-Form FIR
 * Filter Order
                     . 8
 * Stable
                     : Yes
                     : Yes (Type 1)
 * Linear Phase
 */
/* General type conversion for MATLAB generated C-code */
#include "tmwtypes.h"
/*
 * Expected path to tmwtypes.h
 * C:\MATLAB7\extern\include\tmwtypes.h
 */
const int BL = 9;
                                        Can edit these to agree with your code.
const real64 T B[9] = {
    0.02588139692752, 0.08678803067191,
                                           0.1518399865268, 0.2017873498839,
                        0.2017873498839,
     0.2205226777929,
                                           0.1518399865268, 0.08678803067191,
    0.02588139692752
};
                             U Texas Instruments
                                                 Technology for Innovators<sup>™</sup>
```



Quantization Considerations

- Key choice: floating point vs. fixed point
- Advantages of floating point math:
 - Less quantization error
 - Don't have to worry about scaling factors
 - Less likelihood of overflow/underflow
 - Much easier to code
- Disadvantages of floating point math:
 - Requires floating point DSP (higher cost, higher power)
 - Executes slower than fixed point
- C code allows you to "cast" variables into any datatype





Write Code to Realize FIR Filter

• Direct form I implies direct realization of the convolution equation

$$y[n] = \sum_{m=0}^{M-1} h[m]x[n-m]$$

- Some considerations:
 - Allocate buffer of length M for input samples.
 - Move input buffer pointer as new data comes in or move data?



FIR filter example Code

interrupt void serialPortRcvISR()

```
union {Uint32 combo; short channel[2];} temp;
int i = 0;
float result = 0.0;
```

```
temp.combo = MCBSP_read(DSK6713_AIC23_DATAHANDLE);
```

```
samples[0] = (float)temp.channel[0]; // store right channel
```

}

Ł

Note that all math here is floating point. Filter coefficients are also assumed to be floating point.

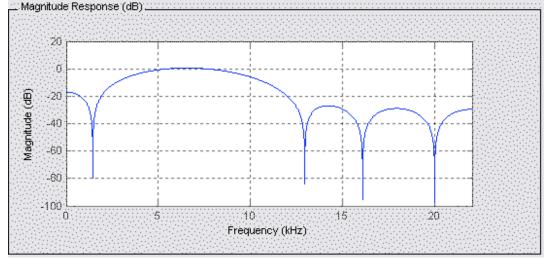




Some Things to Try

- Try creating an FIR filter with the following specs:
 - Bandpass
 - 8th order Direct Form I
 - Least-squares design
 - 44100Hz sampling rate
 - Fstop1 = 3000Hz
 - Fpass1 = 4000Hz
 - Fpass2 = 8000Hz
 - Fstop2 = 12000Hz
 - Equal weighting in all bands
 - All floating point math (single or double precision)
- Use an oscilloscope and a function generator to compare the magnitude response of your filter to the theoretical prediction.





Workshop Day 1 Summary

What you learned today:

- Basics of the TMS320C6713 DSK and Code Composer Studio
- How to test the DSK
- How to open, build, load, and run existing projects
- How to create, build, load, and run new projects
- How to interface with DSK I/O (LEDs, DIP switches, and the AIC23 codec)
- How to debug code in CCS including
 - Setting and clearing breakpoints and probe points
 - Setting up watch variables
 - Plotting arrays of data
 - Animation
- How to use, modify, and create GEL files in CCS.
- How to use Matlab's filter design/analysis tool "fdatool"
- How to implement an FIR filter on the C6713



Workshop Day 1 Reference Material

- Chassaing textbook Chapters 1-2, and 4
- CCS Help system
- **SPRU509F.PDF** CCS v3.1 IDE Getting Started Guide
- C6713DSK.HLP C6713 DSK specific help material
- AIC23 Codec datasheet
- DSK Quick Start Guide (included in your DSK box)
- Spectrum Digital TMS320C6713 DSK reference (included in your DSK box)
- TMS320C6000 Programmer's Guide (SPRU198G.PDF)
- Matlab fdatool help (>> doc fdatool)

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



