MASTERs 2013

The premier technical training conference for embedded control engineers

17085 DSP

Implementing DSP on the PIC32

John Haroian & Brandon Schell

Class Objectives

When you walk out of this class you will….

- **Create a DSP application for the PIC32**
- **Design and simulate DSP elements with free, multi-platform tools**
- **Measure performance and understand maximum capacity on the PIC32**
- **Implement a DSP based system on the PIC32**

Agenda

- A practical DSP system example
- DSP Constructs on the PIC32
- Filters using Octave
- Lab 1 – Creating Silence and then filter it
- DSP Ideals vs. Implementation
- Lab 2 – Create a LR-4 Crossover
- Real-World Correction
- Lab 3 – Implement a Correction Filter
- Summary

A practical DSP system example

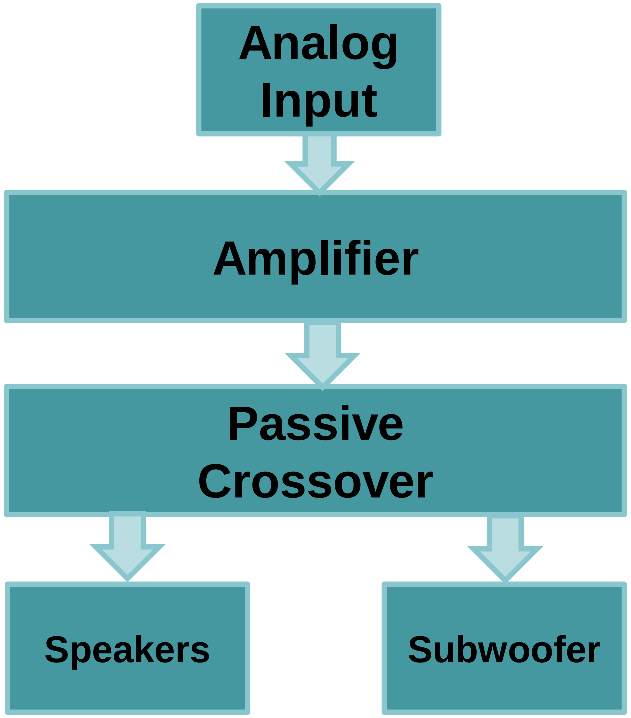
Bi-amped 2.1 Speaker System

- **Speaker System with built-in subwoofer**
- **Bi-amped**
  - Two amplifiers
  - Class D
- **Active Digital Crossover**

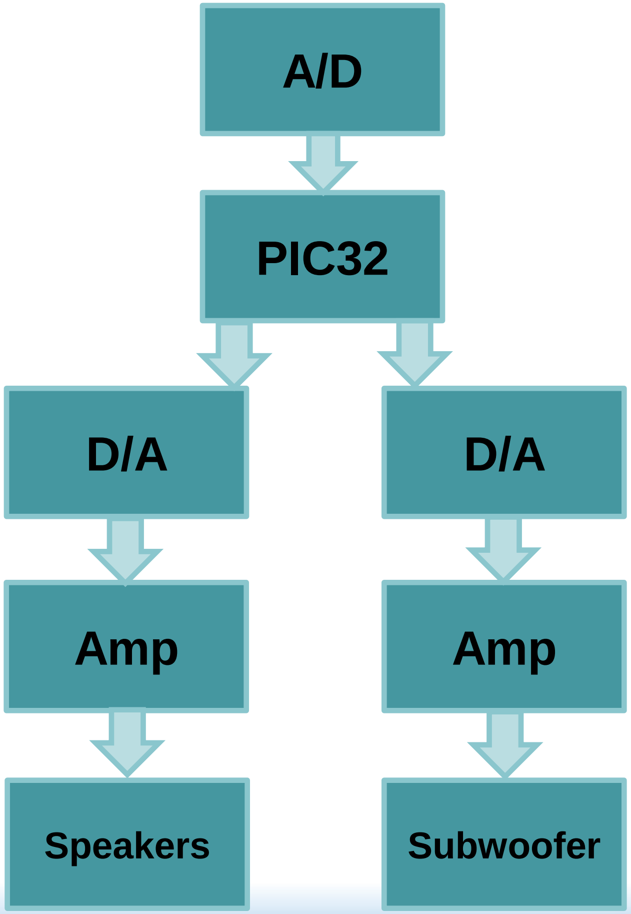
Microchip MASTERs 2013

Block Diagrams

Analog System



Digital System



Comparison

Analog System

- **Pros**
  - Simple and inexpensive
  - Fewer active components
- **Cons**
  - Higher cost speakers
  - Higher power amplifier required

Digital System

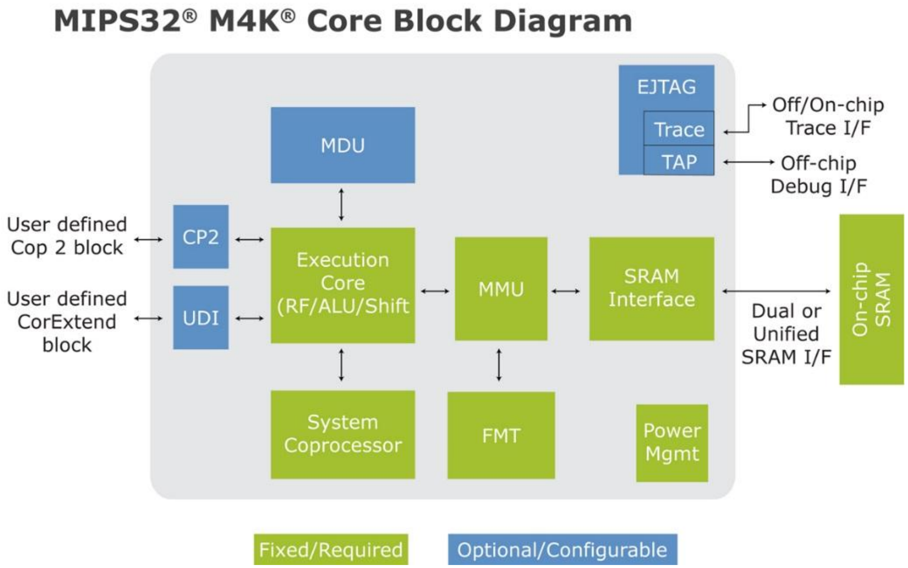
- **Pros**
  - Higher order filtering
  - Equalization
  - Higher SPL with less power
- **Cons**
  - More complex system
  - Software in the signal path

Class Hardware

- **PIC32MX1 & MX2**
  - SPI w/ I^{2}S support
  - 40MHz (50MHz and higher available)
- **AK4645A**
  - Stereo codec
    - **A/D 16 bits**
    - **D/A 16-24 bits**
- **One codec on board**

PIC32 DSP Capabilities

- **32 bit data path and registers**
- **32x16 multiplier with 64 bit result (HI and LO)**
- **Specific DSP instructions**
  - MADD & MSUB
    - **Multiplies and then adds the result to HI/LO**
  - MUL
    - **Multiplies and sends 32 bit result to register**
- **Maximum issue rates**
  - One 32x16 multiply per clock
  - One 32x32 multiply every other clock



DSP Constructs on the PIC32

DSP Constructs

- Filter
- Equalizer
- Delay
- Attenuation

Filter

Selectively attenuates and passes frequencies

Cut off frequency (f_0)

-3dB down point

Order

6dB per octave per order

2^{nd} order = 12dB/octave

Q

Quality factor

0.707 = Butterworth

Octave

Interval between frequencies of ½ or double

100 - 200Hz

1000 - 2000Hz

DSP Constructs

- Filter
- **Equalizer**
- Delay
- Attenuation

Equalizer

A band pass filter with change in gain

Center frequency (f_0)

Bandwidth (BW)

Defined in octaves (1, ½, 1/3, 1/24, etc.)

dB gain

Can be positive or negative

DSP Constructs

- Filter
- Equalizer
- Delay
- Attenuation

Delay

Time shift

FIFO structure

Delay measured in sample periods

Used to align signals in time

DSP Constructs

- **Filter**
- **Equalizer**
- **Delay**
- **Attenuation**

Attenuation

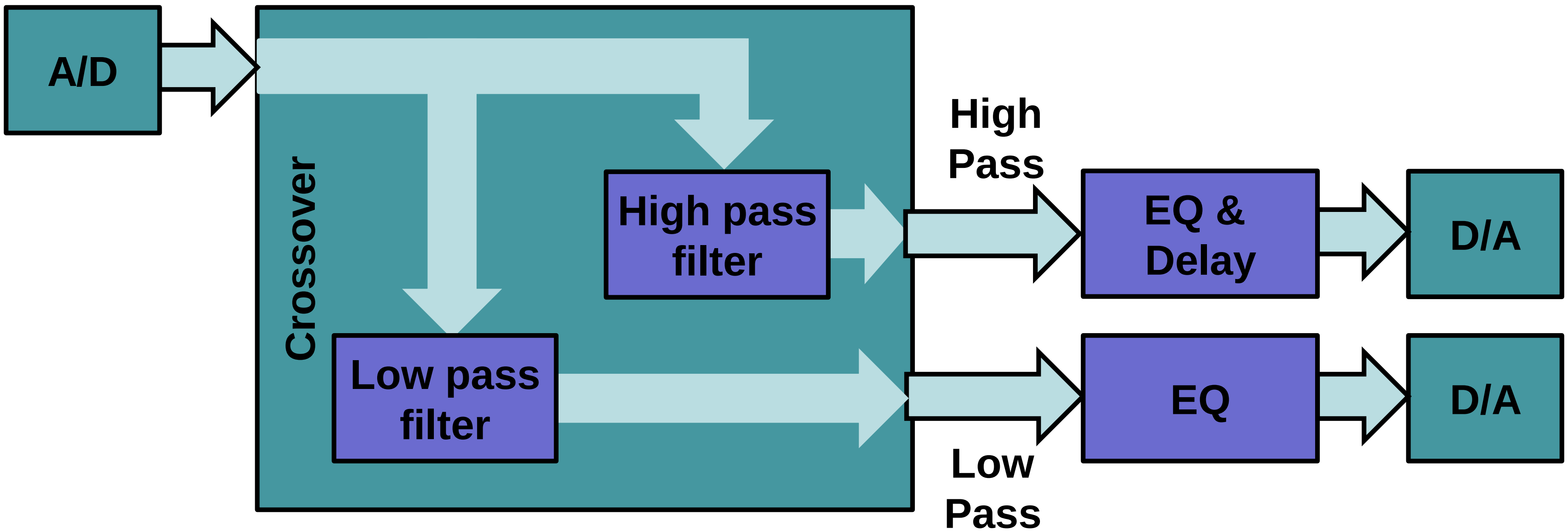
Used to match amplitudes of signals

Always implemented as a reduction

Fractional multiply

Reduces resolution

PIC32 Signal Path



Number Formats

- **AK4645A format**
  - Two's Complement

Examples:

0000 0000 0000 0000 = 0
0000 0000 0000 0001 = 1
1111 1111 1111 1111 = -1
0111 1111 1111 1111 = 32767
1000 0000 0000 0001 = -32767

- **DSP Library format**
  - Q1.15
  - Fractional format
  - Represented by int16
  - No conversion needed

Examples:

0000 0000 0000 0000 = 0
0000 0000 0000 0001 = 0.000030517578125
1111 1111 1111 1111 = -0.000030517578125
0111 1111 1111 1111 = 0.999969482421875
1000 0000 0000 0001 = -0.999969482421875

Math and Number Formats

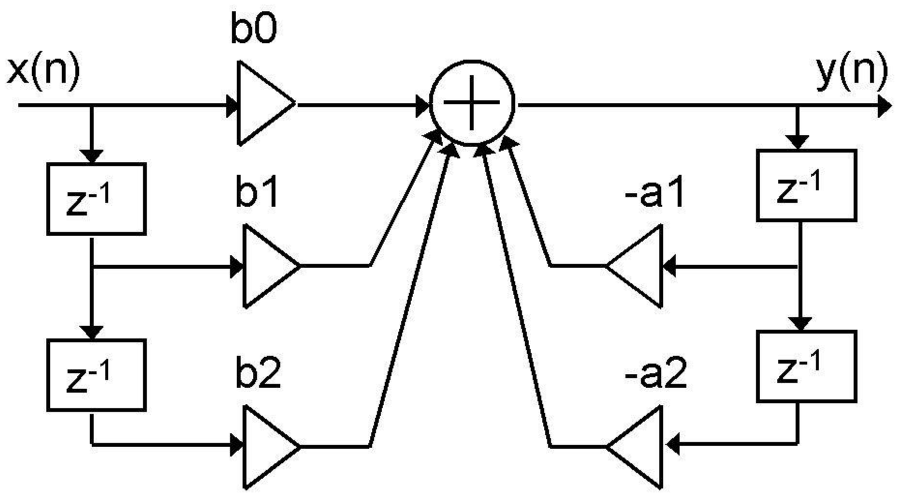
- **Add**
  - Add two numbers (16 bit + 16 bit)
  - Result is 16 bits
  - Q1.15
- **Multiply**
  - Multiply two numbers (16 bit * 16 bit)
  - Result is 32 bits
  - Q2.30 shifted to Q1.31
- **Precision**
  - Q1.15 – Single precision
  - Q1.31 – Double precision

Filter Choice

- **Butterworth filter**
  - Maximally flat pass band
  - 6 dB/octave roll-off per pole (or order)
- **IIR chosen for digital implementation**
  - Can approximate a Butterworth response
  - Lower order than equivalent Finite Impulse Response (FIR)
  - Recursive with potential for instability

Filter Implementation

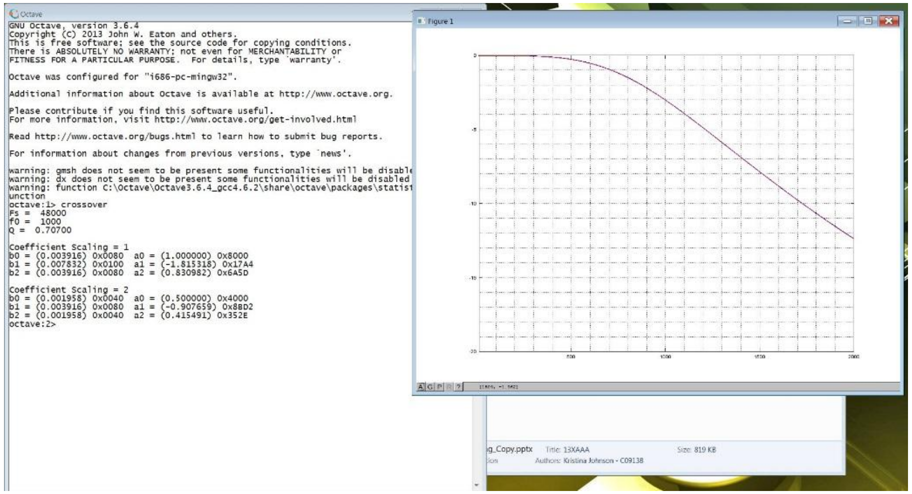
- 2^{nd} order Bi-quad
- Direct Form I
- Preferred for audio due to single sum
- 5 multiplies and 4 delay elements



Filters using Octave

GNU Octave

- High-level interpreted language for complex math
- Open source alternative to MATLAB®
- Available on Windows, Mac, and Linux
- www.gnu.org/software/octave



Using Octave

- **Text based input**
- **.m files are collections of commands (basic scripting language)**
- **All .m files are in Octave folder inside class directory**
- **Used to design and simulate DSP functions**

```
# Filter Parameters
Fs = 48000  #Hz
f0 = 1000 #Hz
dBGain = 0; #dB - only for peaking filters
Q = .707    #Filter shape for Butterworth = .707 (Q=0.54 and Q=1.31 for 4th order Butterworth)

#Intermediate Calculations
A = sqrt(10^(dBGain/20));
w0 = 2*pi*f0/Fs;

alpha = sin(w0)/(2*Q);

#LPF

# Coefficients
blpf = [ ((1-cos(w0))/2)/(1+alpha), (1-cos(w0))/(1+alpha), ((1-cos(w0))/2)/(1+alpha)];
alpf = [ (1+alpha)/(1+alpha), (-2*cos(w0))/(1+alpha), (1-alpha)/(1+alpha) ];

# Coefficients - Scaled down by 2
blpfs = blpf/2;
alpfs = alpf/2;

# Quantized Q15 Coefficients
blpfq = fix(blpf * 2^15)/2^15;
alpfq = fix(alpf * 2^15)/2^15;

# Quantized Q15 Coefficients - Hex
blpfhex = ndec2hex(fix(blpf * 2^15),16);
alpfhex = ndec2hex(fix(alpf * 2^15),16);

printf("\nCoefficient Scaling = 1\n");
for n=1:3
    printf("b%u = (%f) 0x%s\ta%u = (%f) 0x%s\n",n-1,blpf(n),ndec2hex(fix(blpf(n) * 2^15),16),n-1,
end
printf("\nCoefficient Scaling = 2\n");
for n=1:3
    printf("b%u = (%f) 0x%s\ta%u = (%f) 0x%s\n",n-1,blpfs(n),ndec2hex(fix(blpfs(n) * 2^15),16),n-
```

MATrix LABoratory | length : 1489 | lines : 56 | Ln : 8 | Col : 5 | Sel : 0 | 0 | Dos\Windows | ANSI | INS

MICROCHIP MASTERs 2013

Tools for Your Project

DSP17085lib.c

- **Q15iirQ15_DF1**
  - IIR Filter – Direct Form I
  - Single Precision
- **Q31iirQ31_DF1**
  - IIR Filter – Direct Form I
  - Double Precision
- **Q15toQ31**
  - Convert Q1.15 to Q1.31
- **Q31toQ15t**
  - Truncate Q1.31 to Q1.15
- **Q31toQ15r**
  - Round Q1.31 to Q1.15

M Files

- **lpf.m**
  - Low pass filter coefficients
  - Single & Double precision
- **hpf.m**
  - High pass filter coefficients
  - Single & Double precision
- **eqf.m**
  - EQ filter coefficients
  - Single & Double precision

Lab 1: Creating Silence and then filter it

Lab 1 Objectives

- Set up Audio_Start and get silence
- Design a low pass filter using Octave
- Implement filter on PIC32
- Estimate maximum, simultaneous filters on PIC32

Lab 1 Summary

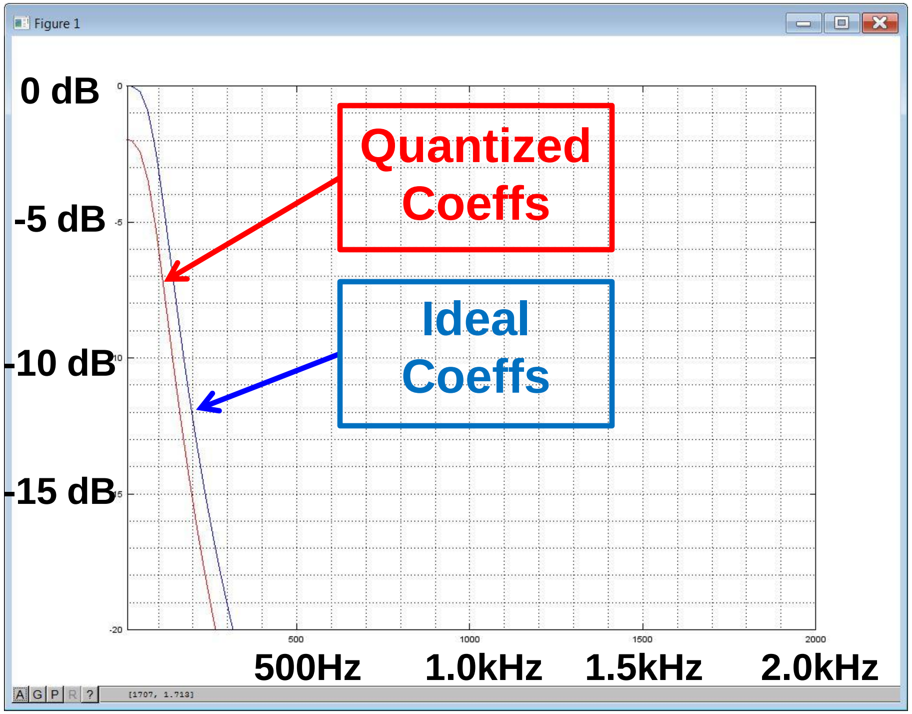
- Set up Audio_Start and heard silence
- Designed a low pass filter using Octave
- Implemented filter on PIC32
- Estimated maximum, simultaneous filters on PIC32

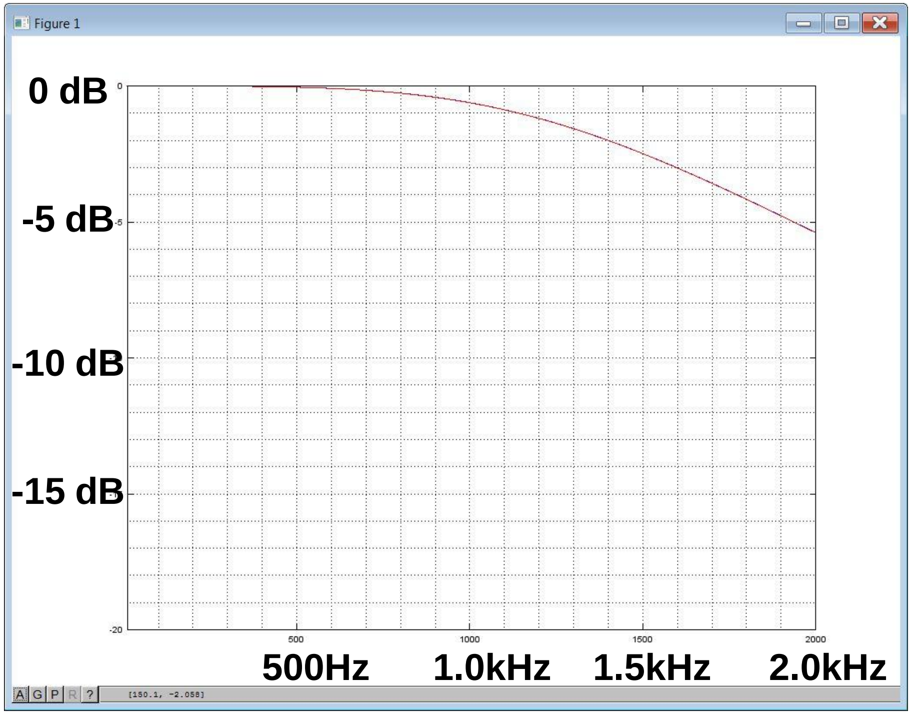
DSP Ideals vs. Implementation

Why 1.6kHz?

- **The filter in Lab 1 implemented a single precision IIR filter**
  - 16 bit coefficients and data
  - 32 bit internal results
- **Quantization imposes limits on pole placement**
- **At 48kHz sample rate, single precision falls apart below 300Hz**

100Hz vs 1.6kHz





Filters below 300Hz?

- **Two methods**
  - Decimation/Interpolation
  - Double precision – 32 bit coefficients / 64 bit results
- **Decimation**
  - Filters work well when away from the extremes
    - **100/48000 = 0.002**
    - **1600/48000 = 0.03**
  - Decimation changes the sample rate
  - Decimate by 16 changes the sample rate to 3kHz
    - **100/3000 = 0.03**

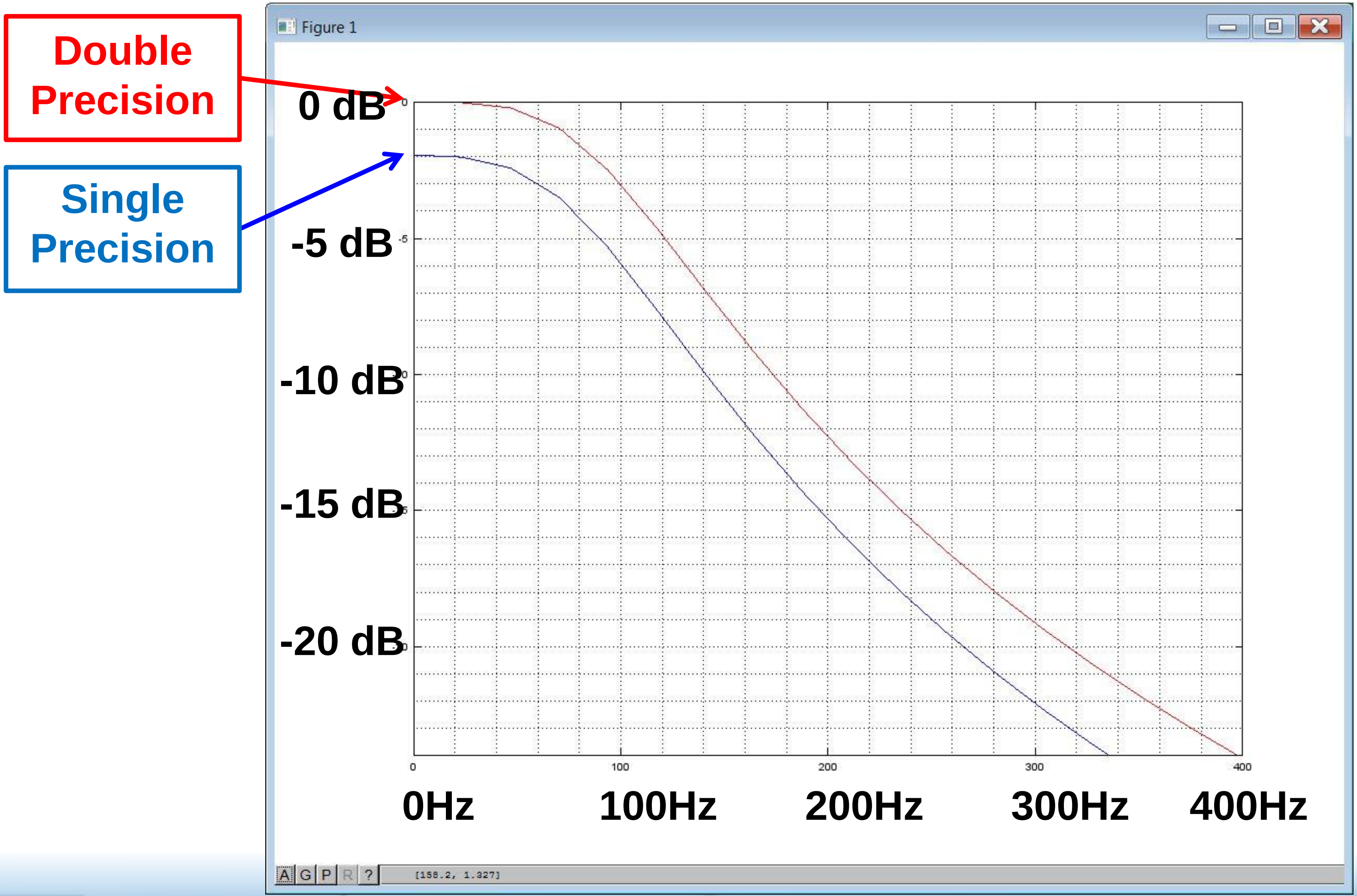
Filters below 300Hz?

- **Double precision**
  - 32 bit coefficients
  - Slower, but much better filter shape at low frequencies (<20Hz)

$$\mathbf{2^{-15} = 0.000030517578125}$$

$$\mathbf{2^{-31} = 0.000000000465661}$$

Double vs Single Precision



Crossovers

What is a Crossover?

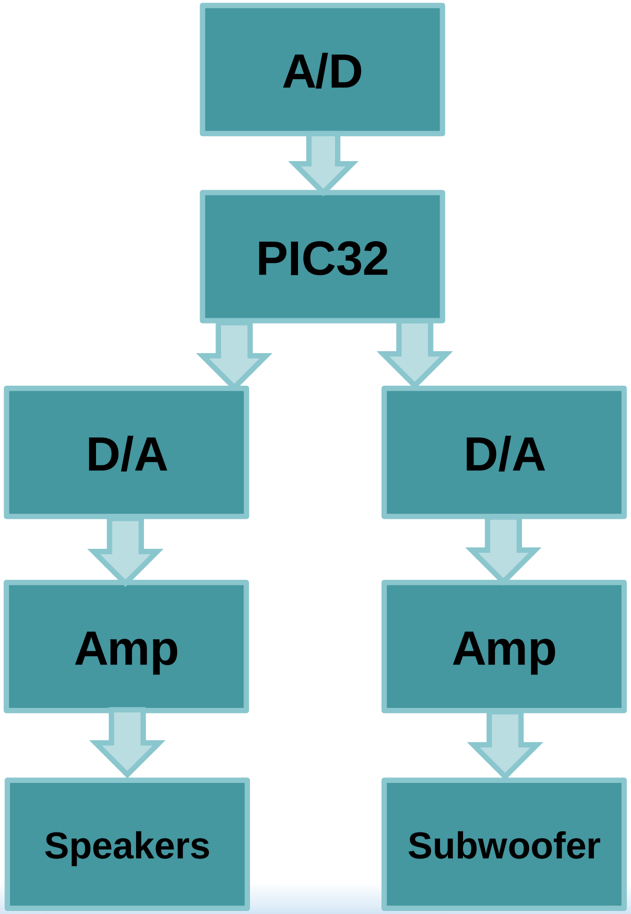
- **A crossover divides the audio spectrum into two or more section**
- **Two-way, three-way, or more**
- **Implementation**
  - Passive or active
  - Analog or digital
- **We are focused on active, digital crossovers**

Remember our example

- **2.1 speaker system**
  - Left & Right speakers
  - Subwoofer
- **Common in computer speaker systems**
- **Left & Right speakers**
  - 250Hz to 20kHz
- **Subwoofer**
  - 20Hz to 250Hz

- **Active means the crossover is applied before the amplifiers**
- **Digital means the crossover is implement mathematically inside the PIC32**

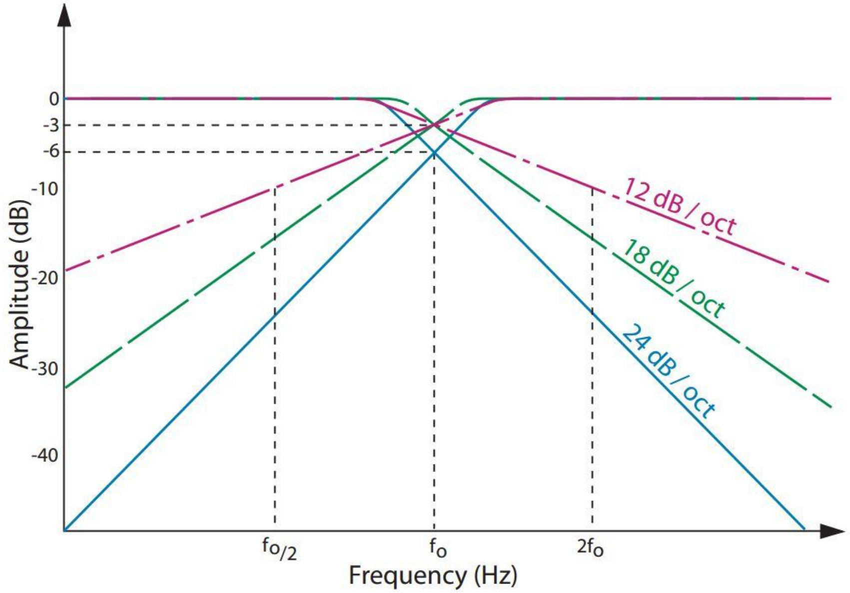
Active Digital Xover



So, how do I make one?

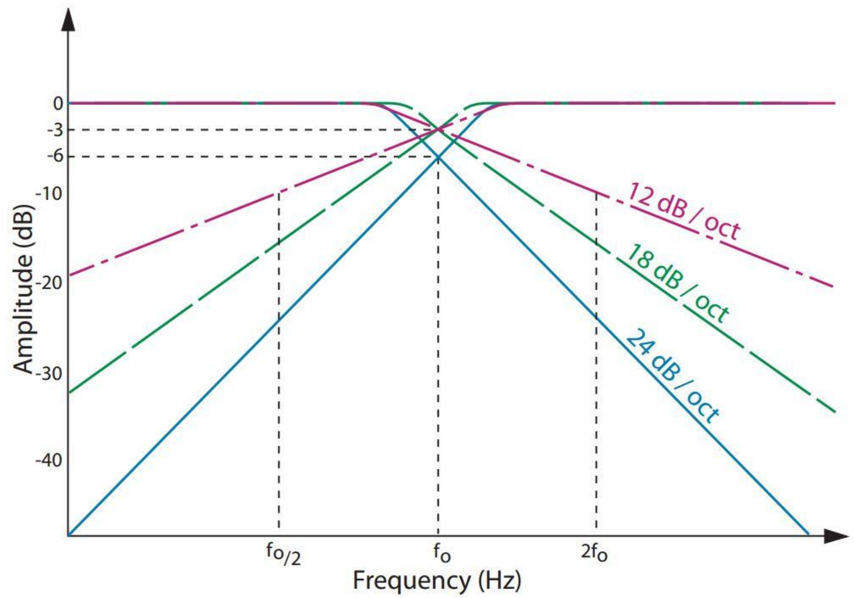
- **Two-way crossover**
  - High pass filter
  - Low pass filter
- **What order?**
- **What frequency?**
- **What type?**

- **Continuing our example….**



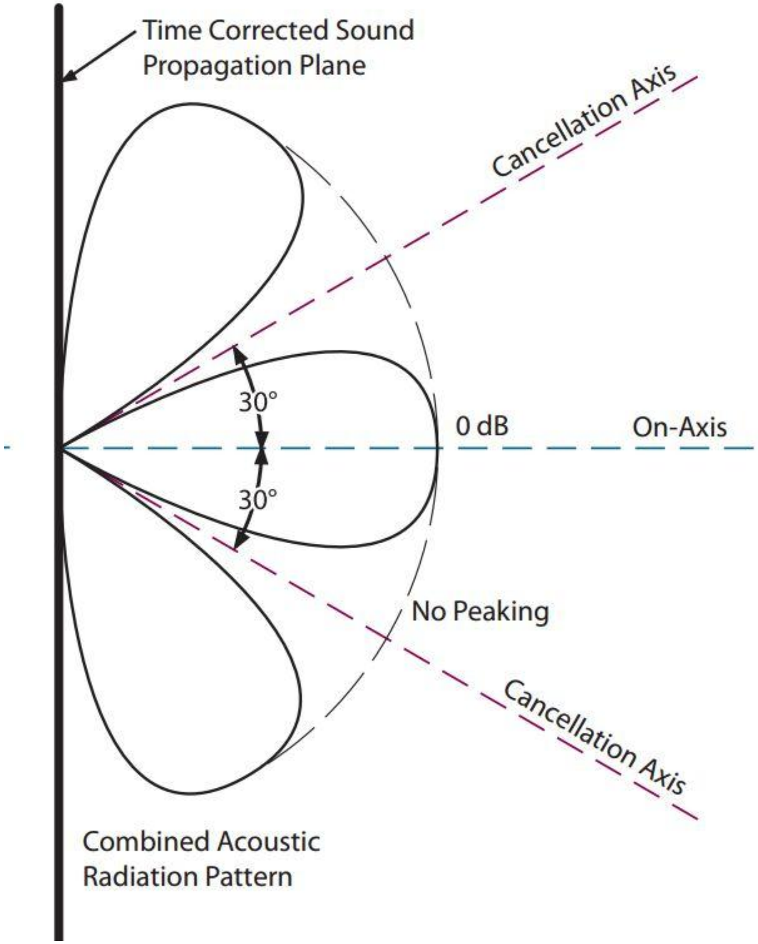
So, how do I make one?

- **Two-way crossover**
  - High pass filter
  - Low pass filter
- **What order?**
  - 4^{th} order
- **What frequency?**
  - 250Hz
- **What type?**
  - Linkwitz-Riley



Linkwitz-Riley Crossover Alignment

- **2 HP engineers in 1976 publish a paper**
  - Siegfried Linkwitz
  - Russ Riley
- **Benefits**
  - In-phase outputs
  - 24dB/octave slope
  - Perfect combined radiation pattern at the crossover point
- **How to Align**
  - Two 2nd order Butterworth filter in series



See http://www.rane.com/note160.htm for more information. (RaneNote 160 by Dennis Bohn)

Lab 2: Create a LR-4 Crossover

Lab 2 Objectives

- **Implement a double precision crossover**
- **Understand how to create a 4^{th} order Linkwitz-Riley filter using biquads**
- **Measure performance impact of double precision on PIC32**

Lab 2 Summary

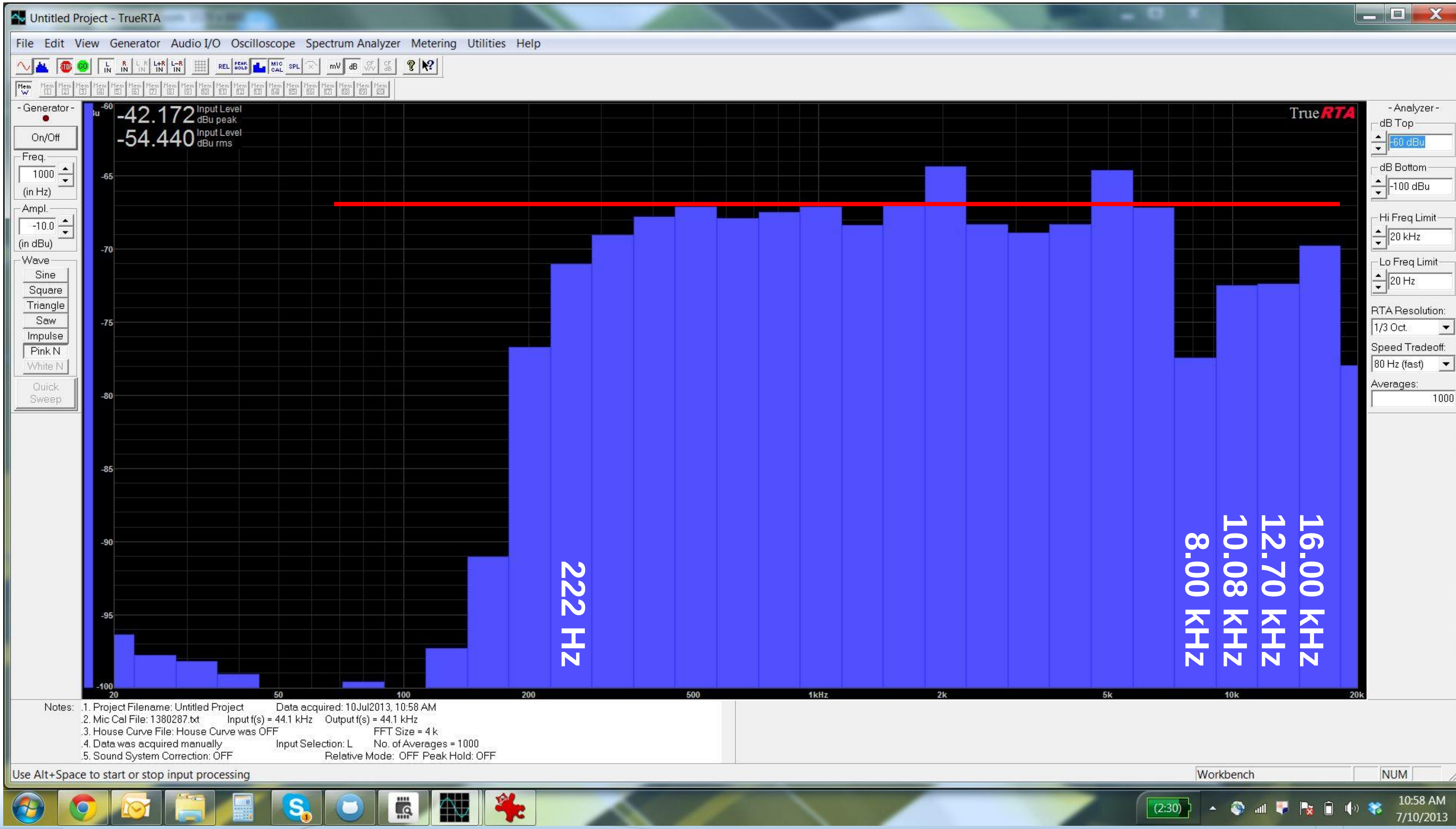
- Implemented a 250Hz, LR-4 crossover
- Used two biquads in series to create 4th order Linkwitz-Riley filter
- Measured the performance impact of double precision filters on the PIC32

Real-World Correction

And then reality hits….

- All of the graphs have had razor sharp lines and perfect slopes
- All speakers are electro-mechanical systems
- Therefore, imperfect

Actual driver measurement



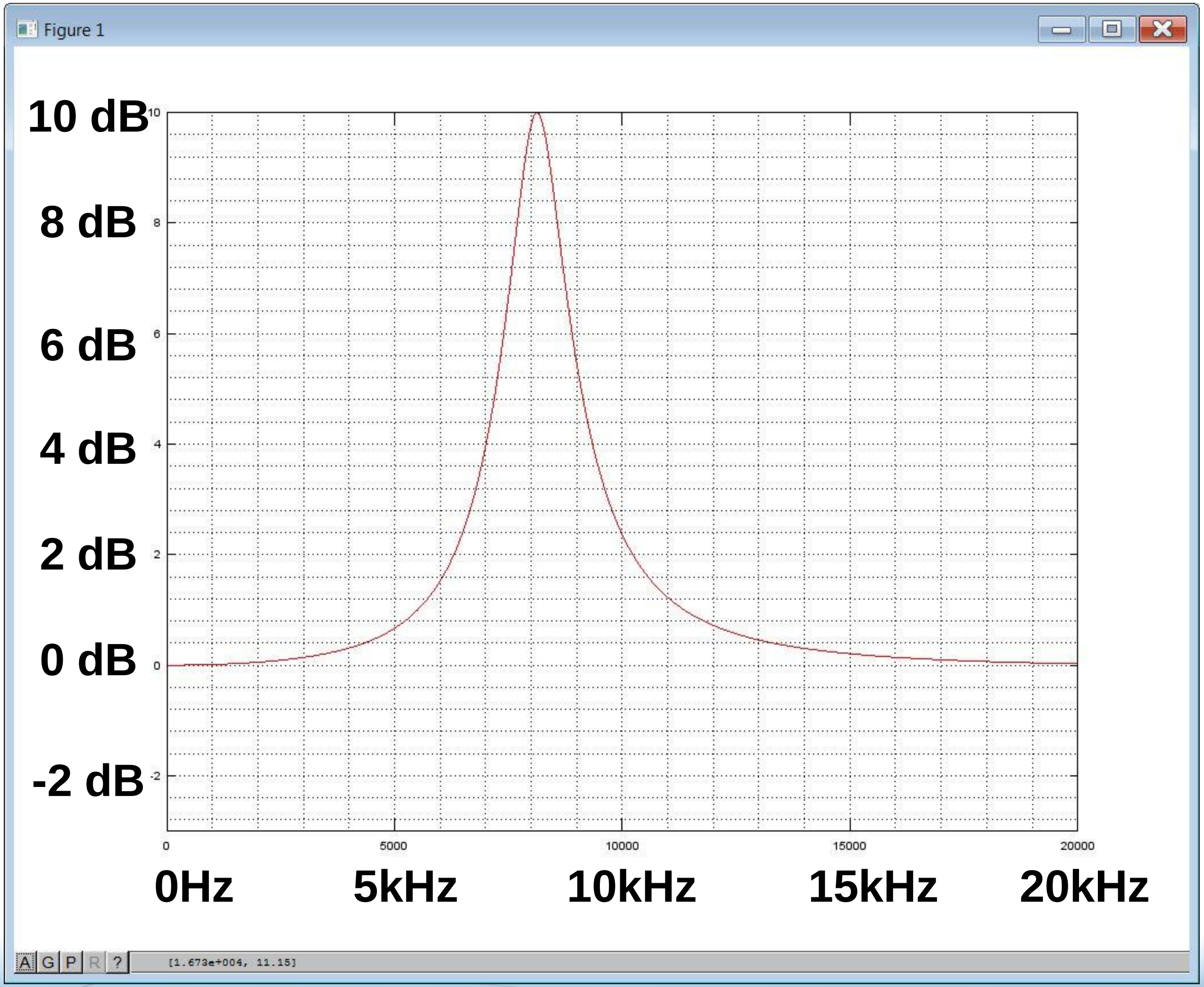
Not bad, except…

- Goal for speaker design is +/-3dB response
- Response curve droops and peaks above 8kHz
- Ideal response is 20Hz to 20kHz

EQ filter to the rescue

- Use EQ filters to correct the response
- Again, use IIR biquads to create bandpass filters of the same width with gain
- 1/3 octave wide and +10dB at 8kHz

1/3 octave EQ @ 8kHz



Lab 3: Implement a Correction Filter

Lab 3 Objectives

- Implement a correction filter for the speaker response curve
- Understand the placement of this filter in the signal chain
- Understand the impact on the system of this filter

Lab 3 Summary

- Implemented a correction filter for the speaker response curve
- Understand the placement of this filter in the signal chain
- Understand the impact on the system of this filter

32Bit Middleware and Software Ecosystem

- **Software Platform Key Features and Benefits:**
  - Modular peripheral drivers and Middleware layers
  - Dynamic Multi-client driver support
  - Designed to be RTOS friendly
  - Designed for Interoperability and 32Bit MCU scalability
  - Integrated, Verified & Supported 3rd Party Partner solutions
- **Interested in learning more? Attend "17033 MPLAB® Harmony" for the Next Generation Middleware/Ecosystem or stop by the "Ask-the-Experts" booth for more info**

Demo

- **We've spent the past 3.5 hours working and testing the pieces of a 2.1 speaker system**
- **It is only appropriate to try this out and see if it actually works**

Summary

Summary

- **Today we covered:**
  - Creating a DSP application for the PIC32
  - Designing and simulating DSP elements with free, multi-platform tools
  - Measuring performance and understand maximum capacity on the PIC32
  - Implementing a DSP based system on the PIC32

Additional Resources

- **Internet**
  - Introduction to Signal Processing
    - **http://www.ece.rutgers.edu/~orfanidi/intro2sp/**
  - RaneNote 160 by Dennis Bohn
    - **http://www.rane.com/note160.htm**
  - Cookbook formulae for audio EQ biquad filter coefficients
    - **http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt**

Parts Used in Demo

- **Microchip**
  - DM320014 - PIC32 USB Digital Audio Accessory Board (Qty. 2)
- **Parts Express**
  - 320-330 – 2x15W Class D Amp (Qty. 2)
  - 120-056 – 12Vdc, 5A Power supply
  - 300-7064 – 0.56cu ft MDF cabinet
  - 285-111 – Dayton Audio 2” speaker (Qty. 2)
  - 290-352 – Goldwood 6.5” DVC Subwoofer

Dev Tools For This Class

- **DM320014 - PIC32 USB Digital Audio Accessory Board**

- **DV244005 - MPLAB® REAL ICE™ In-Circuit Emulator**

- **AC164110 - RJ-11 to ICSP Adaptor**

LEGAL NOTICE

SOFTWARE:

You may use Microchip software exclusively with Microchip products. Further, use of Microchip software is subject to the copyright notices, disclaimers, and any license terms accompanying such software, whether set forth at the install of each program or posted in a header or text file.

Notwithstanding the above, certain components of software offered by Microchip and 3rd parties may be covered by “open source” software licenses – which include licenses that require that the distributor make the software available in source code format. To the extent required by such open source software licenses, the terms of such license will govern.

NOTICE & DISCLAIMER:

These materials and accompanying information (including, for example, any software, and references to 3rd party companies and 3rd party websites) are for informational purposes only and provided “AS IS.” Microchip assumes no responsibility for statements made by 3rd party companies, or materials or information that such 3rd parties may provide.

MICROCHIP DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY DIRECT OR INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND RELATED TO THESE MATERIALS OR ACCOMPANYING INFORMATION PROVIDED TO YOU BY MICROCHIP OR OTHER THIRD PARTIES, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBLITY OF SUCH DAMAGES OR THE DAMAGES ARE FORESEEABLE.

TRADEMARKS:

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC32 logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.