The 29th Annual ARRL and TAPR Digital Communications Conference

DSP Short Course

Session 2: DSP Tools and Filters

Rick Muething, KN6KB/AAA9WK

Some Good Advice!

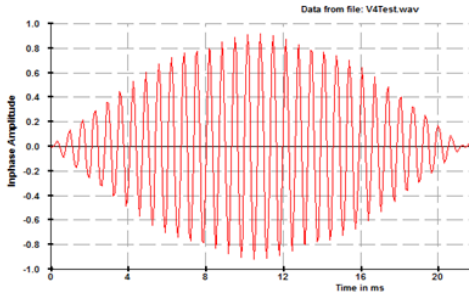
- Abraham Lincoln (1809-65) wisely said:
  - "Give me six hours to chop down a tree and I will spend the first four sharpening the axe.“
  - He knew the value of good tools!

  - To work with DSP we need tools too. These tools will allow us to save, see, interpret and understand the signals we are processing
  - This session focuses on a few of the important tools we will need to be able to do meaningful DSP.

Session 2 Overview

- Representing the waveform in the computer
- Wave utilities you will need (VB Samples on CD)
- Viewing a wave file in time Scope DSP utility
- Viewing a wave file in frequency Scope DSP
- Interpreting Frequency displays
- The “sound card” on the PC – Playback & Capture
- Setting up the sound card in WINDOWS (VB Samples on CD)
- Real time processing … not your 1980’s BASIC!
- Digital Filters FIR, IIR, special case. The IOWegian software tools Scope FIR and Scope IIR (Demo versions on CD)
- Example filter designs using Scope FIR and Scope IIR
- Specialized Frequency Selective Filters
- Session 2 Summary

Representing the Waveform on a Computer



- **Remember we are talking about sampled data systems so our "waveform" is a collection of numeric samples taken at equal intervals (the sample interval).**
- **It can be expressed in different ways in the computer:**
  - **An array of bytes, integers or floating point numbers**

  (This is most efficient when we need to do DSP)
  - **A standard interface (file or stream) compatible with sound cards and other programs.**

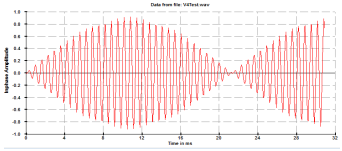
  (This is more convenient when we need to communicate the waveform to another application) For these examples we will use the universal .wav file or .wav stream format

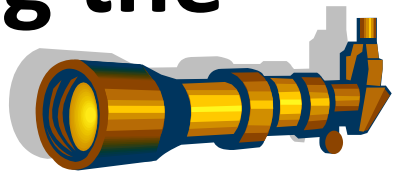
http://en.wikipedia.org/wiki/WAV

Useful Wave Utilities

- **To debug and work with other software packages you will need some Wave Utilities (VB samples on the CD)**
- **These can be used to capture sampled data or send a .wav file or stream to the sound card or other programs (e.g. Scope DSP)**
- **You may want several versions depending on the source of the waveform (Byte, floating etc)**
- **The CD has simple utilities to take a waveform array and create a standard format .wav file**
- **e.g. *WriteFloatingRIFF(strFilename As String, intSampleRate As Integer, intFmtChunkLength As Integer, aryFloatingData() As Double)***

Viewing a .wav file in Time using the Scope DSP utility



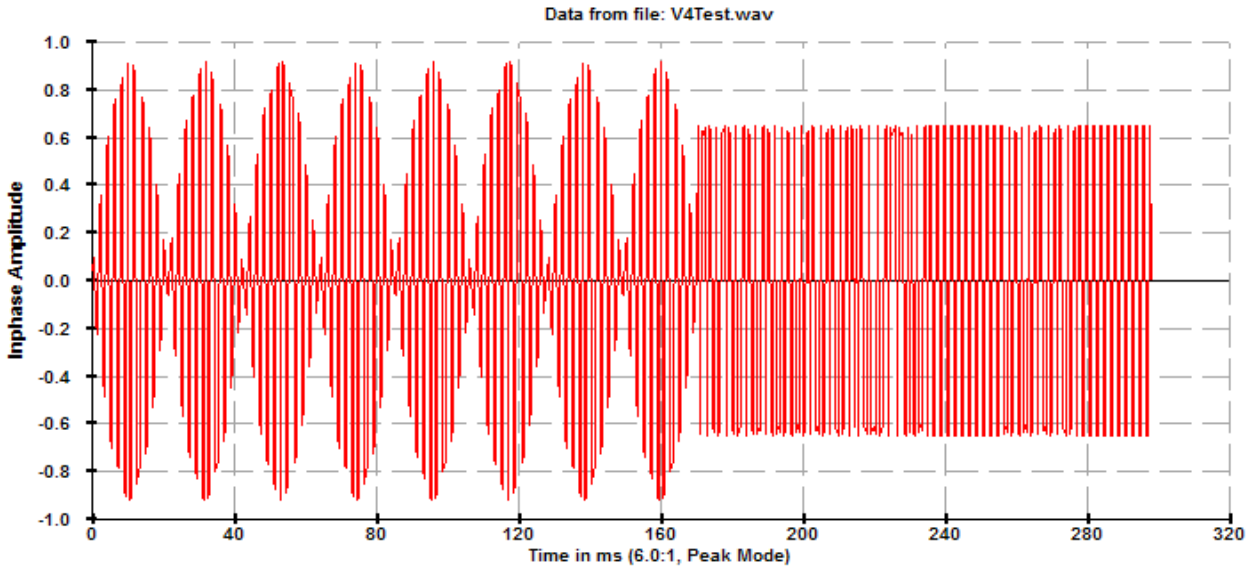


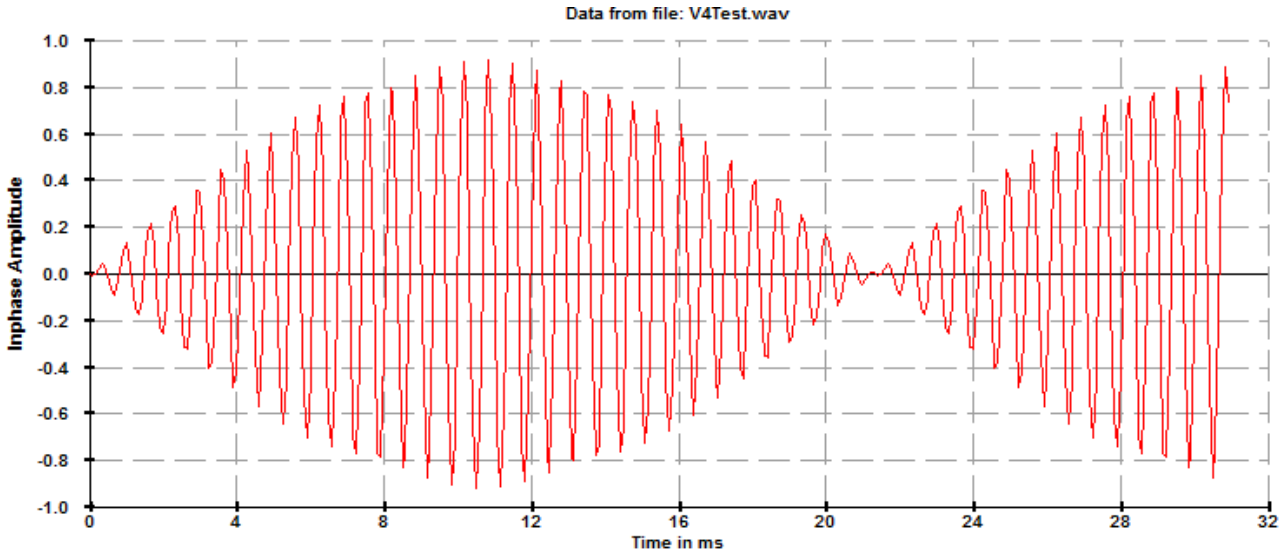
- This was the utility I used to display the wave file examples in Session 1
- You tell Scope DSP:
  - where the file is (directory path and filename)
  - 8 or 16 bits format (we'll usually use 16 bits)
  - and that it is a binary wave
  - The sample rate only has a few options and this can be adjusted after loading

You can then zoom in on the time data by time or sample index to view the waveform

Scope DSP example

The V4 Test.wav



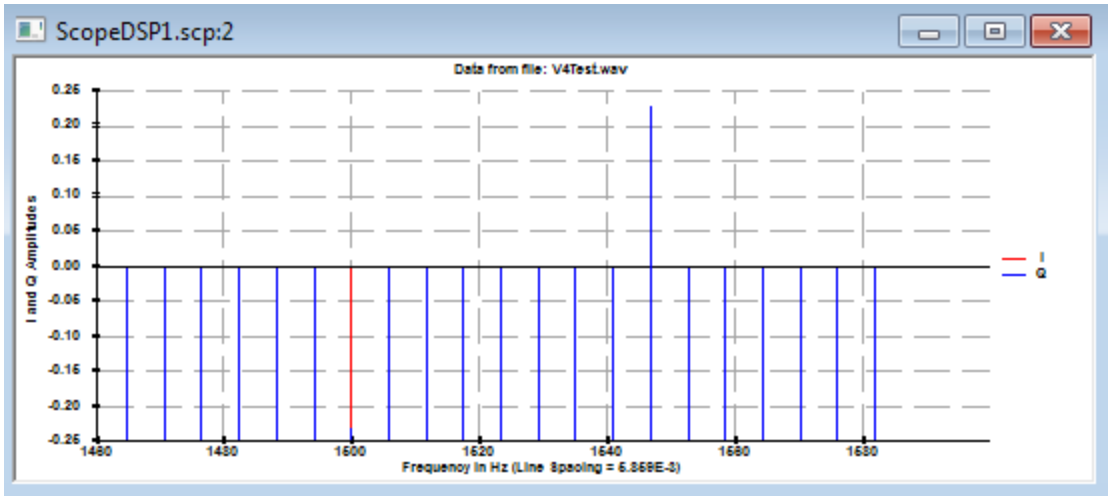


Zoom expansion
Of the first part of
V4 Test.wav

Scope DSP also allows Viewing the DFT of the displayed wave…but It takes practice and experience to interpret what you see!

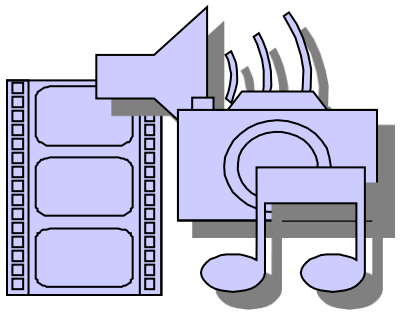
Here is the full spectrum Displaying the Complex magnitude

Display of both I and Q Components zoomed



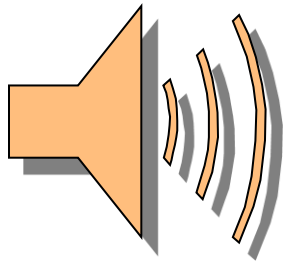
The “Sound Card” on the PC

- First it isn’t just a sound card…it is a **capture device** (the thing that records sound)
- AND a **playback device** … what plays the wave file or stream.
- In most sound “cards” these are two separate devices…they can usually be used at the same time (full duplex)
- Sound “cards” can be internal or external (usually USB)
- Multiple sound cards can be installed and used
- It is difficult for most real-time DSP application to “share” the sound card with another application
  - Who has control?
  - Real time compromises (priority?)



http://en.wikipedia.org/wiki/Sound_card

Steps to use the PC sound card



- Select the Playback and Capture devices
  - Enumerate the installed devices for user
- To Capture waveforms:
  - Configure the Capture device (sample rate, buffer size, notify size, sample format)
  - Start the “Notify thread” a separate thread that Notifies (interrupts) when capture data is available. Your DSP program has to “capture” and process these samples.
- To play a sound:
  - Configure the sound card playback buffers
  - Set the desired volume
  - Pass the .wav file or the wave stream to the playback device
- There are examples of these routines on the CD. ***Note these are Development Environment, language and OS dependent. What is on the CD is VB.NET, using MS Direct X, Direct Sound***

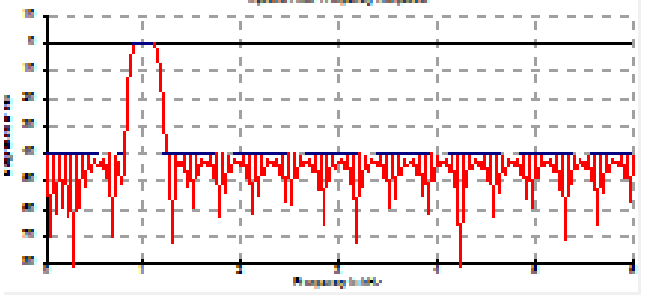
Real-Time processing

- Most of what we do in DSP is what is called *real-time* processing. We process the data as it is received or sent so we must keep up!
  - If I am processing speech and it takes me longer to process than the speech the I have no choice but to delay my result.
  - Some short delay may be acceptable (latency) and we use buffers to handle temporary storage so we don't loose data
- Windows and Linux were not designed as true real-time Operating Systems. …Sometimes CPU and system resources are "borrowed" by the OS.
  - There are sometimes tricks needed to handle this.
  - Timers, Multi threading, thread priorities, synchronization techniques, recovery mechanisms
  - Sometimes dedicated resources (e.g. Sound card) are required

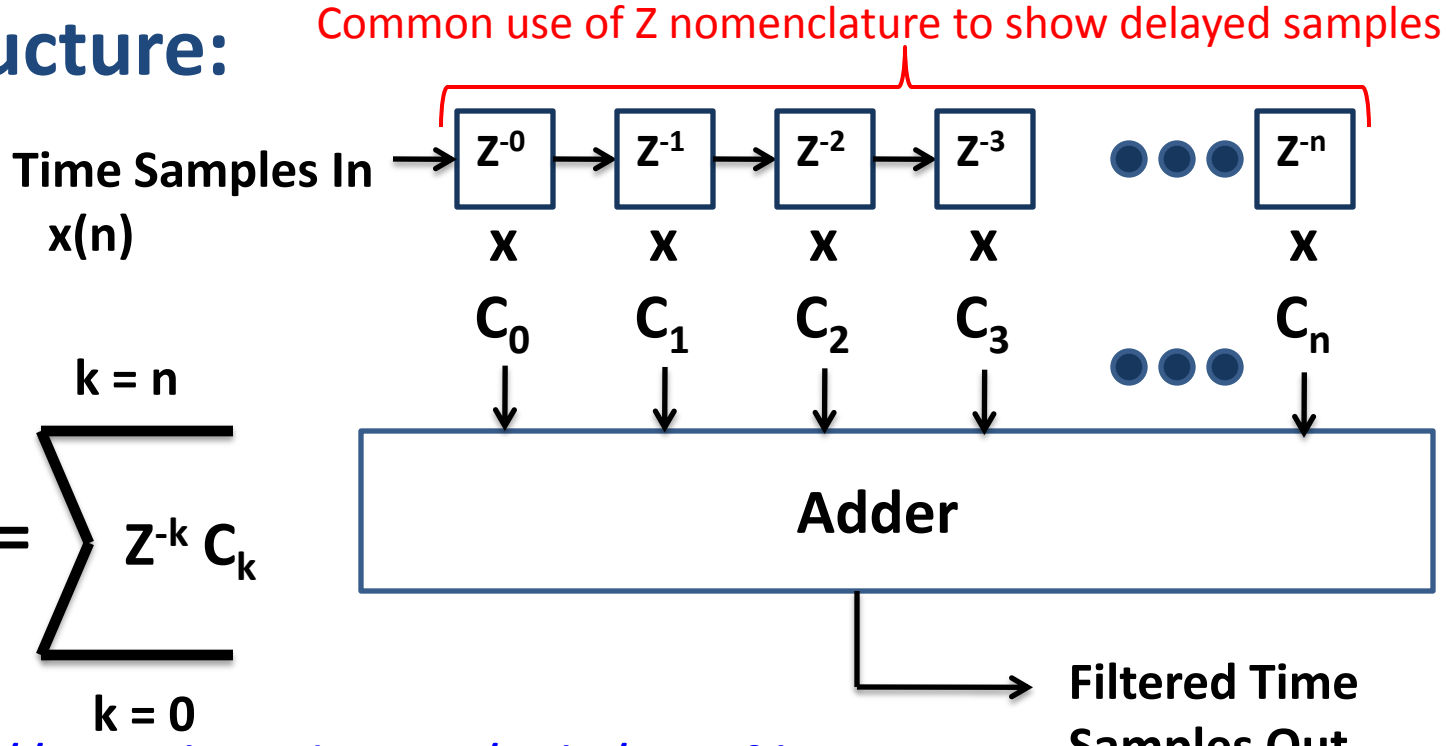
Digital Filters

- One of the most useful things we can do with DSP is Filtering signals digitally.
  - Low pass, band pass, high pass and band stop (notch) filters are possible
  - We can often make filters far better than those possible with analog components BUT …
  - The NFL Theorem kicks in…the better the filter the more computations and the longer delay (latency).
- Lets take a look now at filter types and some examples of filter design tools.

The Classic FIR Filter



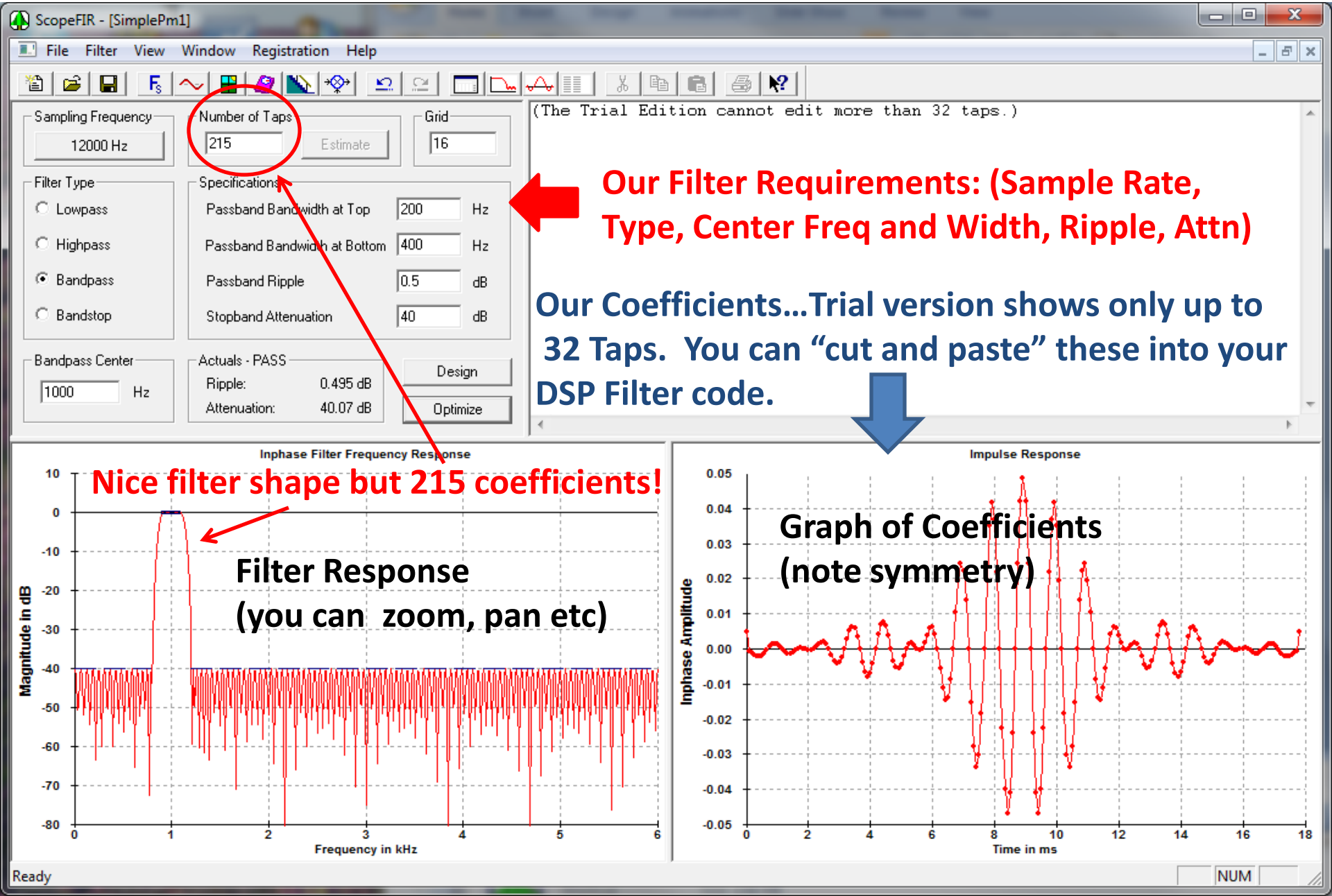
- FIR means Finite Impulse Response
  - Always stable, Fixed delay, just Multiply and Add
  - The filter response is a function of the C_n
- Structure:



$$Y(n) = \sum_{k=0}^{k=n} Z^{-k} C_k$$

http://en.wikipedia.org/wiki/FIR_filter

FIR Band pass Example using SCOPE FIR

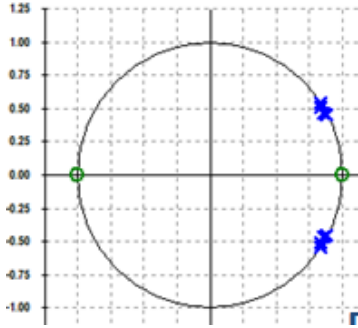


Example VB.NET Filter Code

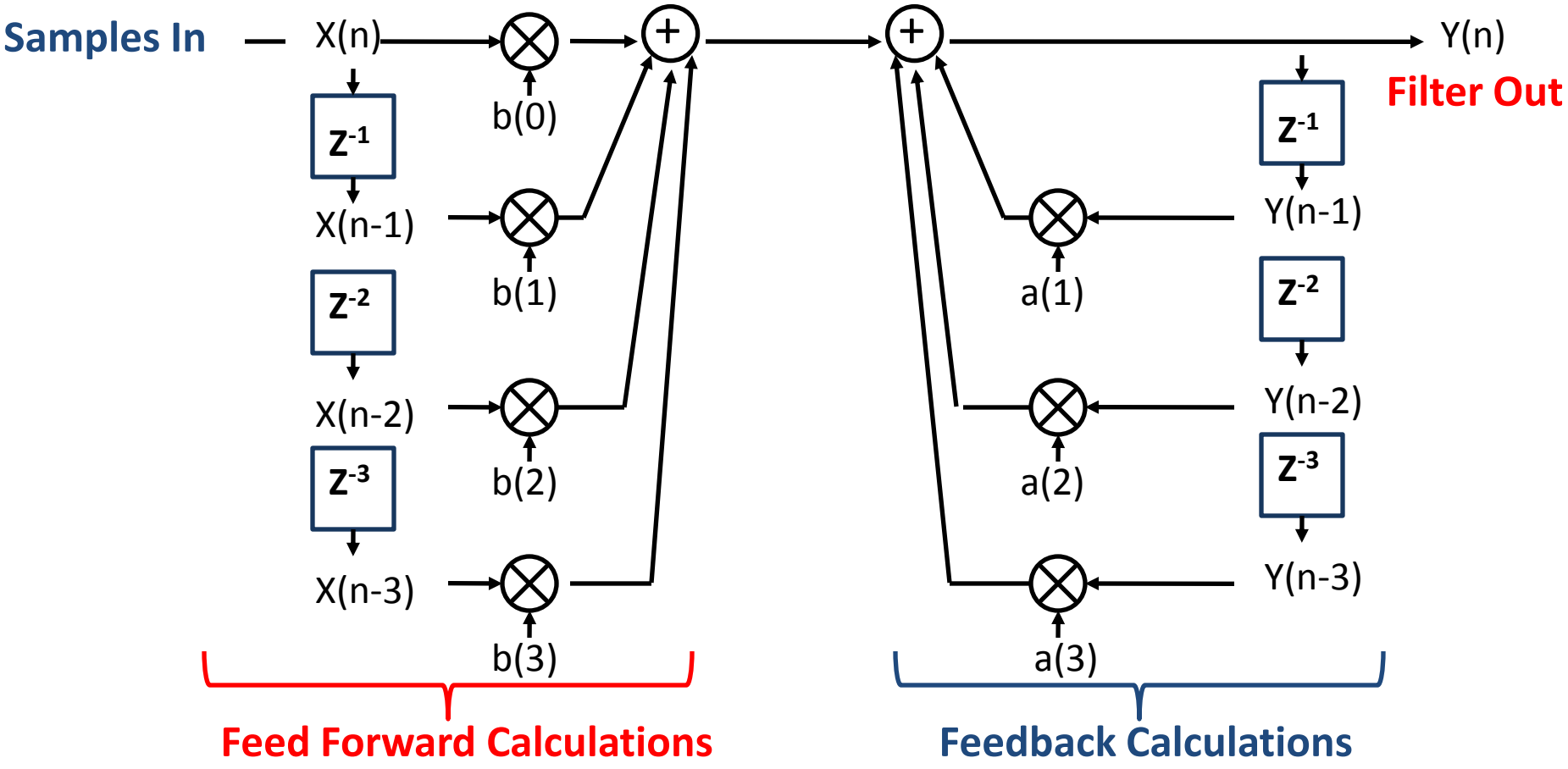
```
Private Sub FIRbp1125(ByRef dblIn() As Double, ByRef dblOut() As Double, ByVal intMaxSamples As Integer)
   ' Implements the FIR bandpassfilter on dblIn at 1125 Hz
   ' assumes the FIR filter order is ODD and there is symmetry about the last coefficient.
   ' uses the symmetry of coeff to minimize multiplies
   Dim intOrder As Integer = 2 * dblBP1125x257FIRcoeff.Length - 1
   Dim dblFil(intOrder - 1) As Double ' a circular buffer for the filter
   Dim intFilPtr As Integer = 0 ' the pointer to the buffer
   Dim Imax As Integer = Math.Min(intMaxSamples, dblIn.Length) ' the number of filter output samples
   ReDim dblOut(Imax - 1)
   For i As Integer = 0 To dblOut.Length - 1
      dblFil(intFilPtr) = dblIn(i)

      For j As Integer = 0 To dblBP1125x257FIRcoeff.Length - 1
         If j <> dblOut.Length - 1 Then
            dblOut(i) += dblBP1125x257FIRcoeff(j) * (dblFil((intFilPtr + intOrder - j) Mod intOrder) + _
               dblFil((intFilPtr + 1 + j) Mod intOrder))
         Else
            dblOut(i) += dblBP1125x257FIRcoeff(j) * dblFil((intFilPtr + 1 + j) Mod intOrder)
         End If
      Next j
      intFilPtr = (intFilPtr + 1) Mod intOrder
   Next i
End Sub 'FIRbp1250
```

The Code that actually implements the filter is fairly simple. This code takes advantage of the fact that band pass FIR filters have symmetric Coefficients. This reduces the number of multiplies by 50% (CD Contains example VB Code)

The IIR Filter



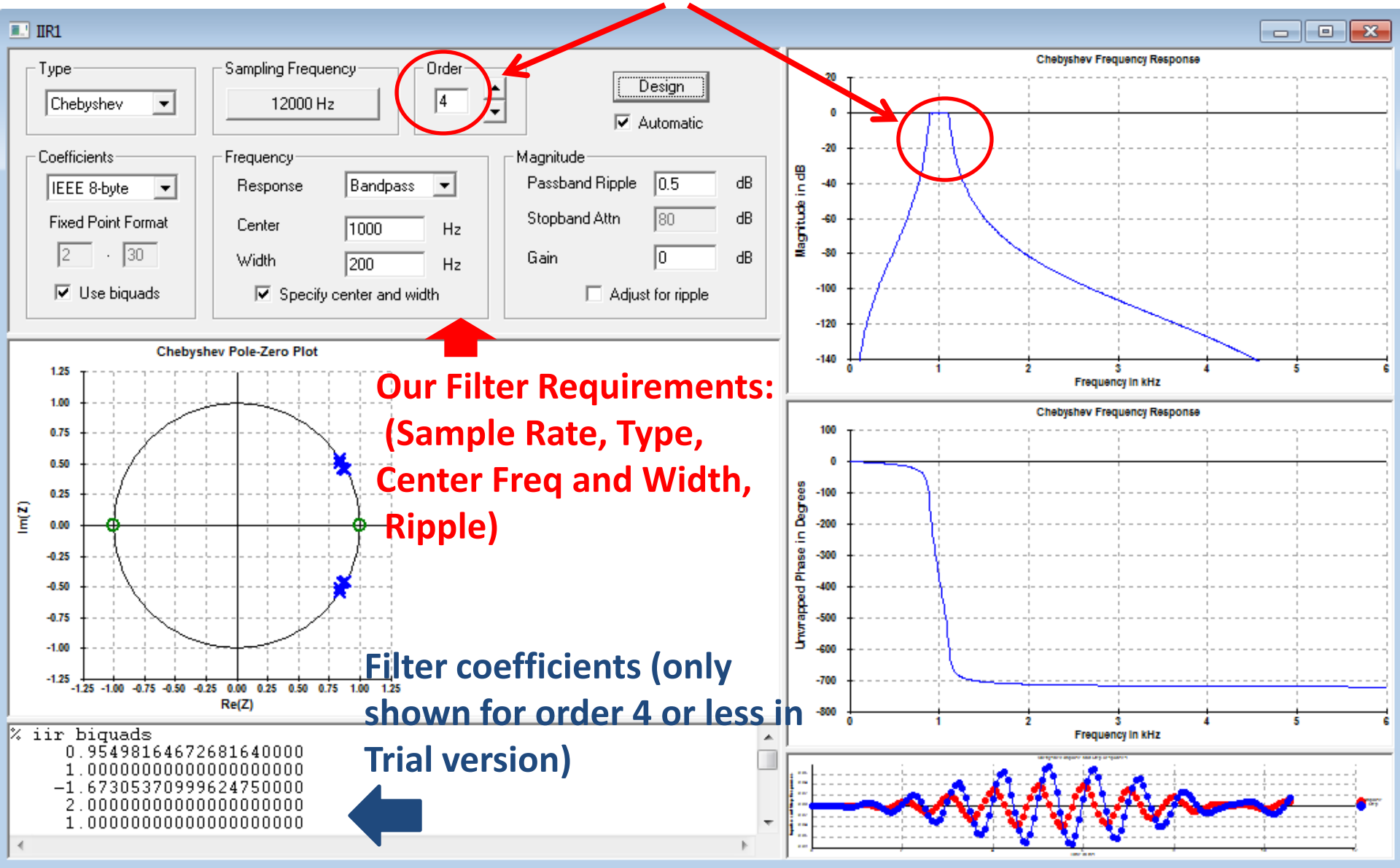
- **IIR means Infinite Impulse Response**
  - **Uses feedback to reduce the number of adds & multiplies**
  - **Conditionally stable! (coefficients must be selected carefully!)**



http://en.wikipedia.org/wiki/Iir_filter

IIR Band pass Example using SCOPE IIR

Note pretty good filter with only 4 coefficient pairs!



Our Filter Requirements: (Sample Rate, Type, Center Freq and Width, Ripple)

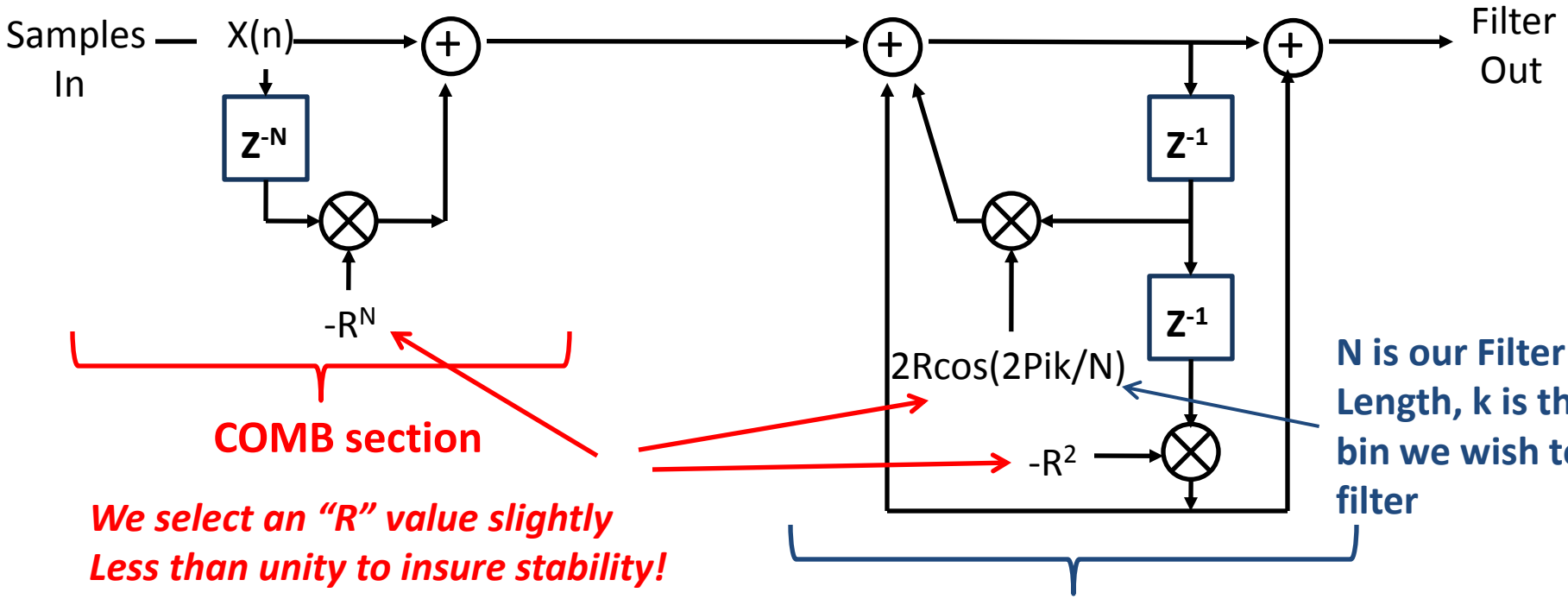
Filter coefficients (only shown for order 4 or less in Trial version)

Frequency Sampling Filters

Can reduce calculations like IIR filters but still have some of the advantages of FIR filters:

Linear phase, Constant group delay (important in some applications!)

Type IV Frequency Selective Filter



We select an "R" value slightly Less than unity to insure stability! (typical R = .9999)

Excellent Treatment in Understanding Digital Signal Processing, 2nd edition, by Lyons

Session 2 Summary

- We have reviewed ways of representing the waveform in the computer and identified needed utilities.
- We have examined a commercial program for viewing wave data and converting it to frequency. (Scope DSP Utility)
- We have talked about the “sound card” on the PC and in general how it is setup… the CD has the details for VB.NET
- We have touched on the challenge of Real-time processing
- We examined three common implementations for Digital Filters:
  - FIR (Finite Impulse Response)
  - IIR (Infinite Impulse Response)
  - Frequency Selective Filters
- We did a simple example of FIR and IIR design using the Scope FIR and Scope IIR programs (trial versions with tutorials on the CD)
- Once again we see and accept the No Free Lunch Theorem