Digital Filters

- Advantages of digital filters
  - Dynamic range
  - No coefficient errors, aging
  - Programmable
  - Always work on first silicon if …
- FIR filters
  - Linear phase
  - Synthesis
- FIR / IIR comparison
- Implementation issues
  - Coefficient rounding
  - Intermediate result dynamic range
  - Limit cycles

Analog versus Digital DR

- It’s much less expensive to add dynamic range to digital circuits than analog circuits
- To double the dynamic range of a digital datapath, we need to add only a bit to an already-wide datapath:

  ![Dynamic Range Diagram]
For comparison, consider summing the outputs of 4 identical analog circuits with identical inputs:

Analog noise is typically uncorrelated in each of the blocks A1-A4:
Analog versus Digital DR

- Doubling analog DR is very expensive:
  - 4X the power
  - 4X the area

- Doubling digital DR is relatively cheap,
  - And cost/function decreases by 29%/year (3dB/year)!

- Practical circuits tolerate very little loss of DR due to finite datapath precision in their DSP sections
  - Analog dynamic range is too precious to lose
  - Digital DR loss of 5% (~ 0.4dB) of total noise power is typical

- Why use analog filters at all?

ADC Dynamic Range

- The figure shows the DR of the best standalone ADCs in 2000

- Dynamic range decreases as converter bandwidth increases

- From 1975-1995, ADC performance at any sampling frequency improved by 2dB/year
ADC Dynamic Range

- ADCs embedded in IC “Systems on a Chip” (SoCs) have less DR than the best standalone ADCs.
- The embedded ADC performance level is shown in red.
- Analog-digital crosstalk and design risk issues limit embedded ADC DR to about 100dB.
- 1 GHz, 30dB DR levels are much more forgiving and the performance gap narrows.

ADC Dynamic Range

- Minimization of analog signal processing is a key goal of mixed-signal IC architecture.
- However, analog signal processing is almost unavoidable “above the red line”.
Practical Constraints

- Only few ADC design teams in the world can produce “green line” dynamic range

- If your SoC architecture requires one of those teams to succeed, think again!

- Mixed-signal SoC architectures fail when their architects choose to ignore long-established, empirically-proven performance scaling laws

FIR Filters

- Only finite zeros

- Linear phase if coefficients are symmetric

- Implement with delays, multipliers, adders

- Lack of good analog delays prevents widespread use of analog FIR filters

- Good synthesis tools (e.g. Remez-Exchange algorithm)
FIR Filter Phase Response

• Consider the Nth-order FIR filter with transfer function:

\[ H(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + ... + a_{N-2} z^{-2-N} + a_{N-1} z^{1-N} + a_N z^{-N} \]

• Suppose the filter coefficients are symmetric about the middle term, i.e.:

\[ H(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + ... + a_2 z^{-2-N} + a_1 z^{1-N} + a_0 z^{N} \]
FIR Filter Phase Response

- The term in brackets \([\cdot]\) is a sum of cosine terms with no phase shift:
  \[
  H(e^{j\omega T}) = e^{-j\omega NT/2} \left[ 2a_0\cos(\omega NT/2) + \text{more real cos terms} \right]
  \]

- The constant group delay of the symmetric coefficient FIR filter is obvious:
  \[
  \theta(\omega) = -\omega NT/2 \quad \tau_{GR} = NT/2
  \]

Coefficient Symmetry

- Three classes of zero groupings produce symmetric coefficients and linear phase

- The first is real axis zeroes at \(r\) and \(1/r\):
  \[
  H(z) = z^{-2} - (r+1/r)z^{-1} + 1
  \]
Coefficient Symmetry

• Conjugate pairs of unit circle zeroes provide linear phase:

\[ H(z) = z^{-2} - 2z^{-1} \cos \theta + 1 \]

Coefficient Symmetry

• Finally, groups of four zeroes at \( re^{\pm j\theta} \) and \((1/r)e^{\pm j\theta}\) provide linear phase

• The filter coefficients for these 4 zeroes are:

\[
\begin{align*}
1 \\
-2(r+1/r)\cos\theta \\
4+r^2+1/r^2 \\
-2(r+1/r)\cos\theta \\
1
\end{align*}
\]
FIR Filter Phase Response

- Another interesting case involves antisymmetric filter coefficients:
  \[ H(z) = a_0 + a_1z^{-1} + a_2z^{-2} + \ldots - a_2z^{2-N} - a_1z^{1-N} - a_0z^{-N} \]
  
- It’s easy to show that
  \[
  H(e^{j\omega T}) = e^{j\frac{\omega NT}{2}}e^{j\pi/2}[2a_0\sin(\frac{\omega NT}{2}) + \text{more sin terms}]
  \]

FIR Filter Phase Response

- For the antisymmetric coefficient case
  \[
  \theta(\omega) = \frac{\pi}{2} - \frac{\omega NT}{2} \quad \tau_{GR} = \frac{NT}{2}
  \]
  
- It’s still linear phase, but with the frequency independent 90° phase shift characteristic of differentiators
Linear Phase FIR Example

```matlab
fs = 1e6;
Fp = 0.10*fs; Fs = 0.13*fs;
Rp = 0.1; Rs = 60;
x = (10^(Rp/20)1)/(10^(Rp/20)+1);
y = 10^(-Rs/20);
[N,fo,ao,W]=remezord(...
    [Fp Fs],[1 0],[x y],fs);
b = remez(N, fo, ao, W);
Hr = tf(b, 1, 1/fs);
Hr = Hr / 10^(rpass/40);
```

z-Plane

```
```

```
```
Phase Response

- Linear?

FIR / IIR Comparison

- 91\textsuperscript{st} order linear phase FIR
- or
- 7\textsuperscript{th} order elliptic IIR
FIR Coefficient Precision

- Finite precision FIR filters add transfer functions of two filters
  - The infinite precision FIR filter
  - A rounding error FIR filter
- The infinite precision FIR dominates the passband response
- The rounding error FIR filter sets stopband attenuation when the infinite precision FIR response is much smaller
FIR Coefficient Precision

- Random rounding errors transform to white "stopband noise"
  - Stopband attenuation increases by about 6dB for each bit of coefficient precision
- If you don’t like the highest bump in the stopband response, generate a new pattern of rounding error
  - Use slightly different dc gain
  - Or slightly different Parks-McClellan (remez) bands
- Trial and error can improve filter stopbands by several dB at a given coefficient precision

Filter Dynamic Range

- Digital filters need more numeric dynamic range than the signals they process
  - They must not overload
  - They must not surprise you with quantization noise
- Digital multiplier/accumulators are multiplexed
  - Difference equations are added up term-by-term, giving us "intermediate transfer functions" to worry about
  - Intermediate overload is as bad as overload
- Let’s look at an IIR bandstop filter example…
2nd-Order Bandstop Filter

- Bandstop filters have transfer functions:
  \[ H(z) = \frac{z^{-1} - z^{-1}(2\cos \Theta) + 1}{r(z^{-1} - z^{-1}(2r\cos \Theta) + 1)} \]
  \[ \Theta = \frac{2\pi f_p}{f_s} \]
  \[ r \approx 1 - \frac{\pi f_p}{Qf_s} \]

- Their gains are close to unity at both dc and \( f_s/2 \)

2nd-Order Bandstop Filter

- Bandstop design specifications:
  - \( f_s = 1 \) MHz
  - \( f_p = 20 \) kHz
  - \( Q_p = 100 \)
Direct Form Realization

\[
\frac{Y(z)}{X(z)} = \frac{z^{-2} - z^{-1}(2\cos\Theta) + 1}{r^2z^{-2} - z^{-1}(2r\cos\Theta) + 1}
\]

\[
Y(z)[r^2z^{-2} - z^{-1}(2r\cos\Theta) + 1] = X(z)[z^{-2} - z^{-1}(2\cos\Theta) + 1]
\]

\[
y_{k-2}r^2z^{-2} - y_{k-1}(2r\cos\Theta) + y_k = x_{k-2} - x_{k-1}(2\cos\Theta) + x_k
\]

Note: Direct form realizations are not ideal for higher order IIR filters. Lattice filters (and variants), which simulate LC ladders, are less susceptible to finite coefficient precision and dynamic range.

2\textsuperscript{nd}-Order Bandstop Filter

- We can build a direct form bandstop this way (method #1):

\[
y_k = x_k - x_{k-1}(2\cos\Theta) + x_{k-2} + y_{k-1}(2r\cos\Theta) - y_{k-2}r^2z^{-2}
\]

Intermediate result 1

Intermediate result 2

Intermediate result 3

- Or this way (method #2):

\[
y_k = y_{k-1}(2r\cos\Theta) - y_{k-2}r^2z^{-2} + x_k - x_{k-1}(2\cos\Theta) + x_{k-2}
\]

- The order of addition matters if we overload in the middle!
2\textsuperscript{nd}-Order Bandstop Filter

- Proceeding from left to right, the difference equation generates 3 intermediate transfer functions plus the complete bandstop transfer function.

- All 4 of these transfer function magnitude responses for method 1 are shown on the next slide.
Method #1 Bandstop Responses

- The complete bandstop filter never exceeds unity gain for sinusoidal inputs

- Intermediate gains exceed 12dB
  - That’s 2-bits above the input MSB

- Let’s examine the area of the notch in more detail …
Method #1 Bandstop Responses

- Frequency (kHz)
- Gain (dB)
- 19.5 20.5
- no “high Q surprises”

Method #2 Bandstop Responses

- Next, we'll examine all the intermediate transfer functions for the method #2 difference equation
- The following figures show significantly different intermediate frequency responses and somewhat lower maximum intermediate gains
Method #2 Bandstop Responses

Method #2 Bandstop Responses
Avoiding Overload

- Sinusoidal steady-state responses for intermediate results are easy to compute and provide useful insight, but sinusoidal inputs are “never” worst cases for overload

- Absolute values of filter impulse response coefficients can give worst-case conditions for overload and intermediate overload

Digital Filter Models

- The order of arithmetic operations in digital signal processing matters

- **Digital filter models must be “cycle true”**

- Unanticipated filter overloads are inexcusable design errors
  - Real-world chip developments must never be late-to-market because of such easy-to-avoid errors!
Biquad Quantization Noise

- Suppose we build our bandstop difference equation with B-bit registers and a BxB = 2B hardware multiplier:

\[ y_k = Gx_k - Gx_{k-1}(2 \cos \Theta) + Gx_{k-2} + y_{k-1}(2r \cos \Theta) - y_{k-2}r^2z^{-2} \]

- Build up the difference equation leaving partial results in a 2B-bit accumulator
  - Accumulate y(k)’s with a minimum number (i.e. 1) of rounding operations
  - If each of the 5 products above is rounded to B-bits, you’ll have 5X more quantization noise power

- Output noise from rounding operations can be large for high Q digital biquads

Digital Filter Models

- Datapath rounding operations can degrade digital filter dynamic ranges by surprisingly large amounts

- **Digital filter models must be “bit true”**

- Bit true and cycle true models require that filter models (and modelers) provide exact test vectors for integrated digital filters
Limit Cycles

- A disadvantage of digital IIR filters relative to digital FIR filters is that their responses get strange as they settle in response to transients.

- As settling error approaches rounding error, offsets and oscillations can occur:
  - Non-zero offsets lead to “dead zones”
  - Oscillations are called “limit cycles”

- A combination of rounding (or truncation) and feedback is required for limit cycles.

\[
H(z) = \frac{0.125 (z^{-2} - 1)}{z^{-2} + 0.75}
\]

- We'll look for limit cycles in the bandpass filter:
- Note that this filter passes frequencies near \( f_s/4 \).
Bandpass Transient Response

- Let's examine the bandpass filter's response to the initial condition $y(1)=y(2)=10$

- The bandpass filter output **should** decay to 0
  - The floating point filter output does
  - The fixed point filter output doesn’t
  - Let’s take a look…
Limit Cycles

- This bandpass filter limit cycle oscillation occurs right at $f_s/4$
  - Right in the middle of the filter passband
  - Could this be a low-level input to the filter at $f_s/4$?

- IIR filter designers must evaluate and be wary of limit cycle oscillations
Digital Filter Models

• Bit true and cycle true digital filter models allow simulation and evaluation of:
  – Overload and intermediate overload
  – Quantization noise
  – Limit cycles and dead zones
  – Finite precision coefficient effects

• Spending time and money on silicon without such models is crazy!