DSP Design

Numbering Systems
Basic Building Blocks
Scaling and Round-off Noise

Viktor Öwall

viktor.owall@eit.lth.se

Number Representation

Floating vs. Fixed point

In floating point a value is represented by
- mantissa determining the resolution/precision
- exponent determining the dynamic range

In fixed point we only have a single value

Floating point gives higher dynamic range but the cost is high in
- energy
- area
- calculation time

For energy efficient implementations fixed point is preferred

Binary numbers, unsigned integers

MSB =
Most Significant Bit

LSB =
Least Significant Bit

N bits → 2^N ord

2^2	2^1	2^0	
0	0	0	(0)
0	0	1	(1)
0	1	0	(2)
0	1	1	(3)
1	0	0	(4)
1	0	1	(5)
1	1	0	(6)
1	1	1	(7)

Dynamic range and Resolution

Nr. of bits	Nr. of levels	Resolution V_{fs}=0.5V	Dynamic Range V_{LSB}=0.03125
4	16	0.03125V	0.5V
8	256	2mV	8V
12	4096	0.12mV	128V
16	65 536	7.6μV	2042V

How do we use the bits?
Depends on the application!

Unsigned Number Representation

Fixed radix (base) systems

The digits $a \in \{0, 1, 2, \ldots\ r-1\}$ in a radix r system:

$$\sum_{i=k-1}^{-l} r^i \times a_i =$$

$$= r^{k-1}a_{k-1} + r^{k-2}a_{k-2} \cdots r^1 a_1 + r^0 a_0 + r^{-1}a_{-1} \cdots r^{-l}a_{-l}$$

described in a fixed point positional number system:

$$a_i\, a_{i-1} \cdots a_1\, a_0 \,.\, \underbrace{a_{-1} \cdots a_{-l}}_{\text{Fractional part}}$$

Example: Unsigned Number

$$\sum_{i=k-1}^{-l} 10^i a_i = \quad \{a \in \{0, 1, 2, \ldots\ 9\} \text{ in radix } 10\}$$

$$= 10^{k-1}a_{k-1} + 10^{k-2}a_{k-2} \cdots 10^1 a_1 + 10^0 a_0 + 10^{-1}a_{-1} \cdots 10^{-l}a_{-l}$$

$$\sum_{i=k-1}^{-l} 2^i a_i = \quad \{a \in \{0, 1\} \text{ in radix } 2\}$$

$$= 2^{k-1}a_{k-1} + 2^{k-2}a_{i-2} \cdots 2^1 a_1 + 2^0 a_0 + 2^{-1}a_{-1} \cdots 2^{-l}a_{-l}$$

Example: Unsigned Number

$$\sum_{i=k-1}^{-l} 2^i a_i = \quad \{a \in \{0, 1\} \text{ in radix } 2\}$$

$$= 2^{k-1}a_{k-1} + 2^{k-2}a_{i-2} \cdots 2a_1 + a_0 + 2^{-1}a_{-1} \cdots 2^{-l}a_{-l}$$

$$1010.0110 \Rightarrow$$

$$1 \bullet 2^3 + 0 \bullet 2^2 + 1 \bullet 2^1 + 0 \bullet 2^0 + 0 \bullet 2^{-1} + 1 \bullet 2^{-2} + 1 \bullet 2^{-3} + 0 \bullet 2^{-4} =$$

$$8 + 2 + \frac{1}{4} + \frac{1}{8}$$

Signed Digit Number Representation

The digits $a \in \{-\alpha, \ldots 0, \; \ldots \; r-1-\alpha\}$ **in a radix** r **system:**

$$\sum_{i=k}^{-l} r^i \times a_k$$

Example Radix 10: $a \in \{-4, -3, \ldots 0, \; \ldots \; 4, 5\}$

$$(3 \quad -1 \quad 5)_{10} = 10^2 \times 3 - 10^1 \times 1 + 10^0 \times 5 = 300 - 10 + 5 = 295$$

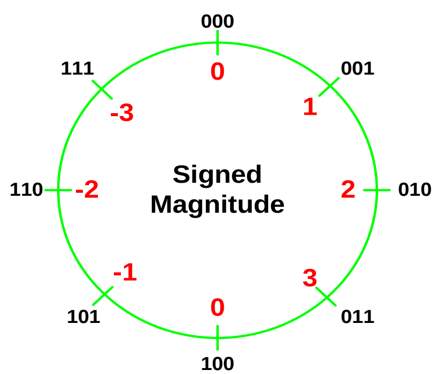
$$(3 \quad .-1 \quad 5)_{10} = 10^0 \times 3 - 10^{-1} \times 1 + 10^{-2} \times 5 = 3 - 0.1 + 0.05 = 2.95$$

Signed Number Representation

Sign Magnitude
One's Complement
Two's Complement

Signed Magnitude

Unsigned numbers with a sign-bit



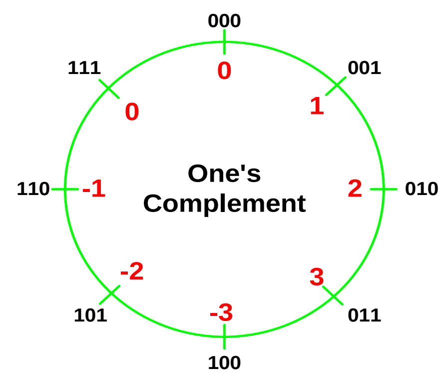
- Two Zeros

+ Low Power?

+ Easy to convert to Negative

One's Complement

Signed numbers by inverting (Complement)

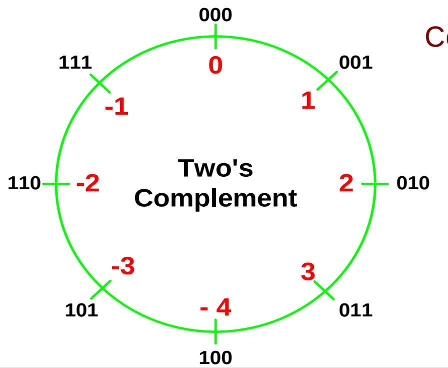


- Two Zeros

+ Easy to convert to Negative

Two's Complement

Most widely used fixed point numbering system



Complement + LSB

+ One Zero

+ Easy Addition

- Not so easy to convert to Neg.

Two's Complement

The digits $a \in \{0,1\}$ in a radix 2 system:

$$-2^{k-1} \times a_{k-1} \quad + \sum_{i=k-2}^{-l} 2^i \times a_i =$$

$$= -2^{k-1}a_{k-1} + 2^{k-2}a_{k-2} \cdots 2^1 a_1 + 2^0 a_0 + 2^{-1}a_{-1} \cdots 2^{-l}a_{-l}$$

described in a fixed point positional number system:

$$a_{k-1}\, a_{k-2} \cdots a_1\, a_0 \,.\, \underbrace{a_{-1} \cdots a_{-l}}_{\text{Fractional part}}$$

Sign Bit

Example: 2's complement

$$1010.0110 \Rightarrow$$

$$-1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} =$$

$$-8 + 2 + \frac{1}{4} + \frac{1}{8}$$

Sign Extension in Two's Complement

$$-2^{k-1}a_{k-1} + 2^{k-2}a_{k-2} \cdots 2a_1 + a_0 =$$

$$-2^{k}a_{k-1} + 2^{k-1}a_{k-1} + 2^{k-2}a_{k-2} \cdots 2a_1 + a_0 =$$

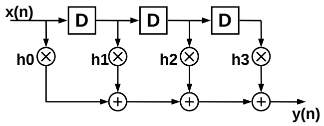
$$-2^{k+1}a_{k-1} + 2^{k}a_{k-1} + 2^{k-1}a_{k-1} + 2^{k-2}a_{k-2} \cdots 2a_1 + a_0$$

Example:

$$10010 = 110010 = 1110010 = 11110010 = \cdots$$

$$00010 = 000010 = 0000010 = 00000010 = \cdots$$

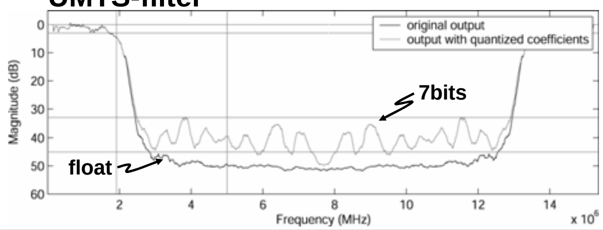
The Wordlength, i.e. nr of bits



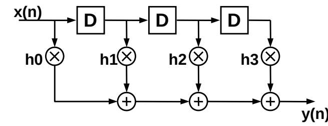
Every extra bit costs
- energy/power
- delay
- area

⇨ the wordlength has to be reduced

UMTS-filter



The Wordlength, i.e. nr of bits

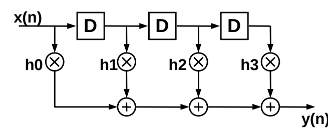


The output of
- adder output needs an extra bit to be sure of no overflow, e.g.
  decimal: 2+2 = 4 ⇨ binary: 10+10=100
- multiplier MxN bits ⇨ M+N bits for full precision

⇨ Precision has to be limited

Basic Building Blocks

Basic Building Blocks



In the FIR filter
- adders
- multipliers
- registers

in other algrithms also: shift, minus, division,...

- left shift is multiply by 2
- right shift is a dived by 2

but is low complexity!

Comparing Basic Building Blocks

High

Complexity

Low

Divider

Multiplier

Adder/Subtarct

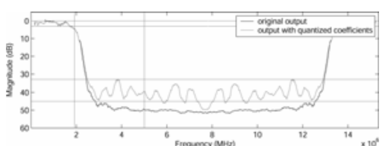
Shifter

			a_3	a_2	a_1	a_0
			b_3	b_2	b_1	b_0
			$a_3 b_0$	$a_2 b_0$	$a_1 b_0$	$a_0 b_0$
		$a_3 b_1$	$a_2 b_1$	$a_1 b_1$	$a_0 b_1$	
	$a_3 b_2$	$a_2 b_2$	$a_1 b_2$	$a_0 b_2$		
$a_3 b_3$	$a_2 b_3$	$a_1 b_3$	$a_0 b_3$			
p_6	p_5	p_4	p_3	p_2	p_1	p_0

a_3	a_2	a_1	a_0
b_3	b_2	b_1	b_0
s_3	s_2	s_1	s_0

Scaling and Round-off Noise

Quantization



Two Types

- Coefficient Quantization
  - Non-Ideal Transfer Function
    - Compare to analog component variations
- Signal Quantization
  - Round-off Noise
  - Limit Cycles

Quantization

Round-off Noise

- Affect the output as a random disturbance

Limit Cycle Oscillations

- Undesired periodic components
- Due to non-linear behavior in the feedback (rounding or overflow)

Quantization Analysis

Using “real” rounding, truncation, and overflow

- Give exact result
- Tricky - need integer representation

Using noise models

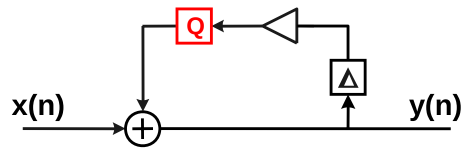
- Floating point representation can still be used
- Suitable for Matlab, C/C++ ...

DSP Design

Rounding Truncation

Rounding/Truncation is “always” there!

Especially necessary in recursive systems



Without quantization - infinite wordlength

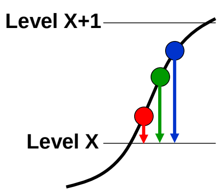
Multiplication ⇒ n+m output bits

Addition ⇒ n+1 output bits

Viktor Öwall, Dept. of Electrical and Information Technology, Lund University, Sweden - www.eit.lth.se

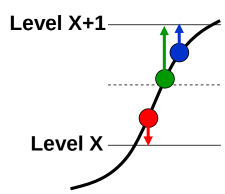
DSP Design

Truncation and Rounding



Truncation
All values approximated in the same direction

Max error = 1LSB



Avrundning
Values approximated up or down

Max error = 1/2 LSB

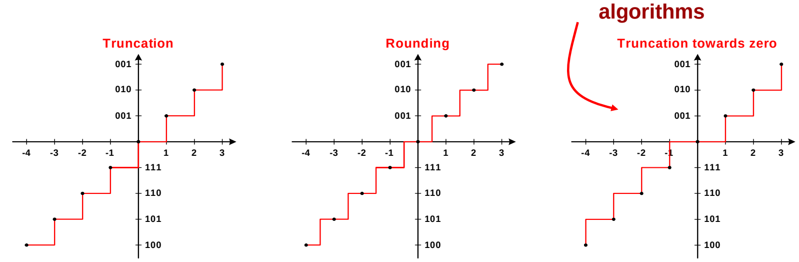
Viktor Öwall, Dept. of Electrical and Information Technology, Lund University, Sweden - www.eit.lth.se

DSP Design

Rounding Truncation

No energy added to the system

Often used in recursive algorithms



DC error

”Rounded to even”

Add LSB before truncation if negative

Viktor Öwall, Dept. of Electrical and Information Technology, Lund University, Sweden - www.eit.lth.se

Scaling

Adjust signal range to fit the hardware

Unchanged transfer function (Scaled coefficients might move the pole-zeros)

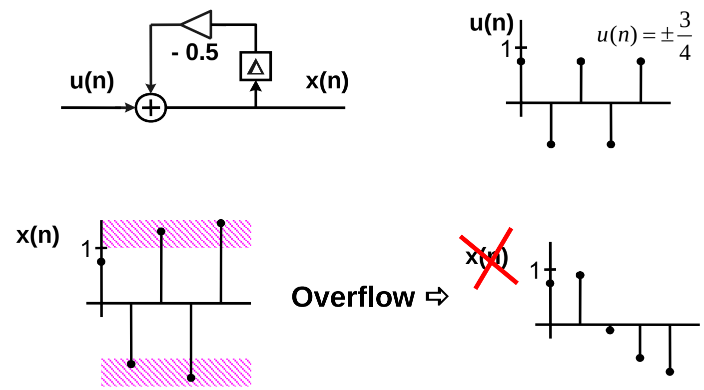
Trade-off

- Scale up to reduce roundoff noise
- Scale down to avoid overflow

But you loose precision!

DSP Design

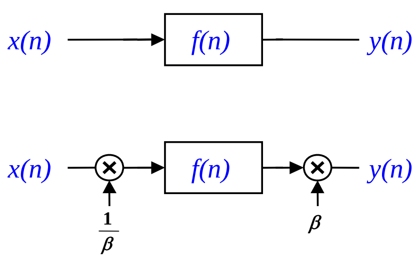
Example Where Scaling is Needed



Viktor Öwall, Dept. of Electrical and Information Technology, Lund University, Sweden - www.eit.lth.se

DSP Design

Scaling



Safe scaling if

$$\beta = \sum_{i=0}^{\infty} |f(i)|$$

Where $f(i)$ is the unit sample response

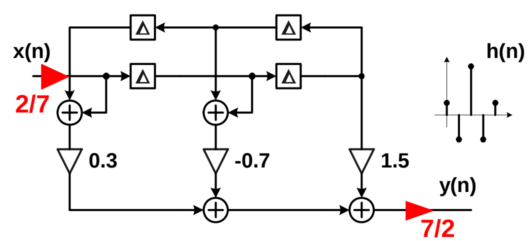
Viktor Öwall, Dept. of Electrical and Information Technology, Lund University, Sweden - www.eit.lth.se

DSP Design

Example: Safe Scaling

$\frac{2}{7}x(n)$ *and* $\frac{7}{2}y(n)$ *give safe scaling*

$$\beta = \sum_{i=0}^{\infty} |f(i)| = 0.3 \times 2 + |-0.7 \times 2| + 1.5 = 3.5$$



Increased roundoff noise

Internal scaling might improve

(Linear phase FIR. Note the strength reduction)

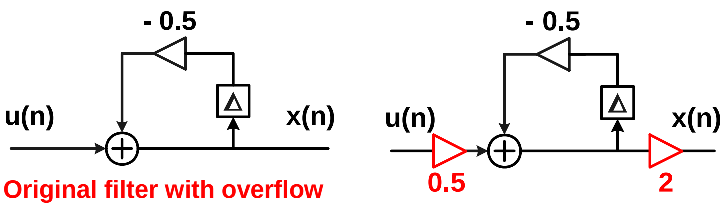
Viktor Öwall, Dept. of Electrical and Information Technology, Lund University, Sweden - www.eit.lth.se

Example: Safe Scaling

$$\beta = \sum_{i=0}^{\infty} |f(i)| = \sum_{i=0}^{\infty} \left|(-0.5)^i\right| =$$

Geometric series

$$\left|(-0.5)^0\right| + \left|(-0.5)^1\right| + \left|(-0.5)^2\right| + \cdots = \frac{1}{1-0.5} = 2$$



DSP Design

Scaling

- Safe scaling is pessimistic
  - Alternative is scaling with

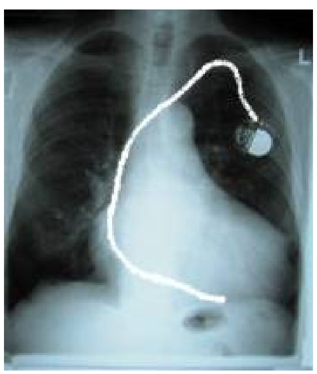
$$\beta = \sqrt{\sum_{i=0}^{\infty} (|f(i)|^2)}$$

- In practice: Scaling with $\beta = 2^{\pm n}$
  - Easy to do - a shift
- Increased internal wordlength an alternative

Viktor Öwall, Dept. of Electrical and Information Technology, Lund University, Sweden - www.eit.lth.se

DSP Design

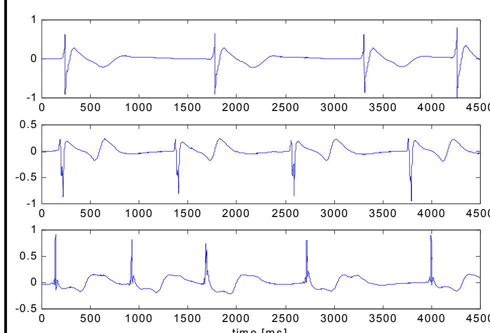
Pacemaker example

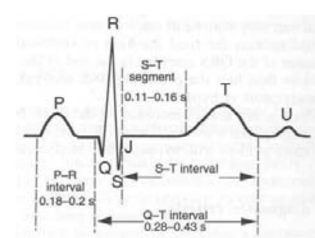


Viktor Öwall, Dept. of Electrical and Information Technology, Lund University, Sweden - www.eit.lth.se

DSP Design

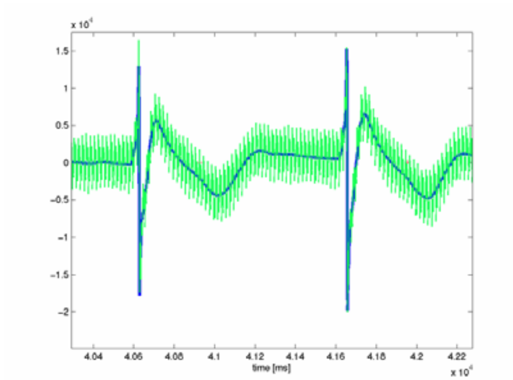
The Electrocardiogram (EGM)





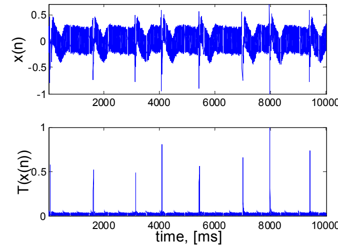
Viktor Öwall, Dept. of Electrical and Information Technology, Lund University, Sweden - www.eit.lth.se

The Interfered signal



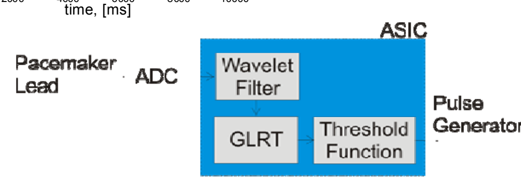
EGM with added interference

Filtering Performance

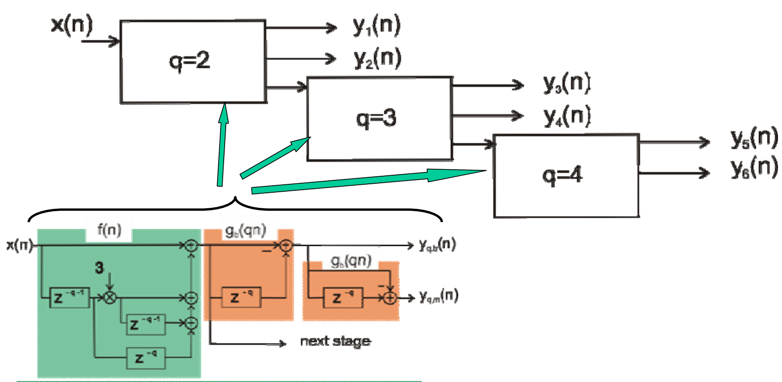


EGM + Interference from AC hand drill 20dB SNR

Output of the GLRT and threshold



Wavelet Filterbank



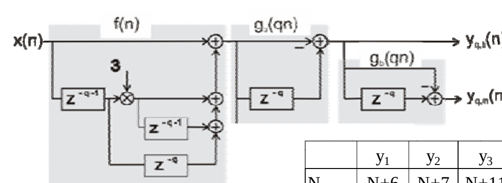
$$F(z) = 1 + 3z^{-1} + 3z^{-2} + z^{-3}$$

$$G_b(z) = -1 + z^{-1}$$

Bit-optimization

- **Signals have been monitored to determine the upper bound of the wordlength**

Comparison of worst-case wordlength and implemented wordlength at the wavelet output:

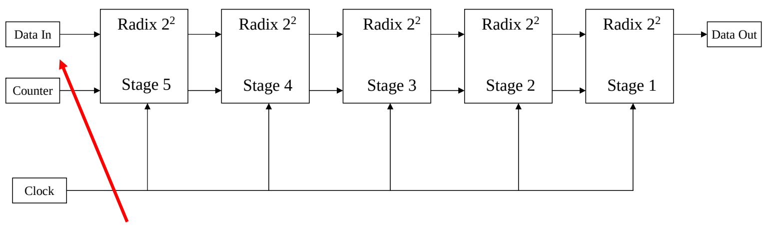


	y_1	y_2	y_3	y_4	y_5	y_6
N_{wc}	N+6	N+7	N+11	N+12	N+14	N+15
N_{Imp}	N+1	N+1	N+2	N+1	N+2	N+1

Example: Internal Scaling

1024-point FFT

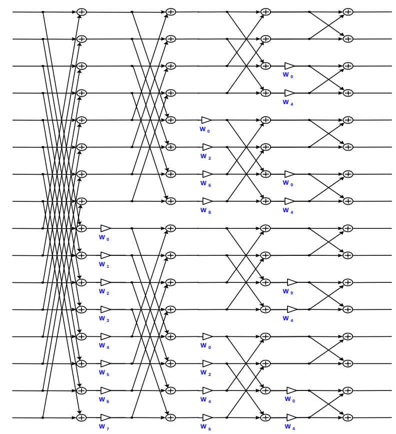
- VHDL bit-level simulation
- Compared with Matlab floating-point simulation
- Optimized internal scaling



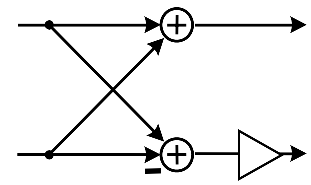
White noise input

Source: Fredrik Kristensen

A 16-point Radix-2 FFT



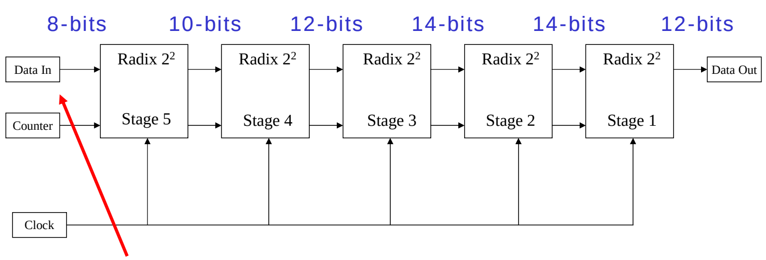
Basic Butterfly unit



Example: Internal Scaling

1024-point FFT

- VHDL bit-level simulation
- Compared with Matlab floating-point simulation
- Optimized internal scaling



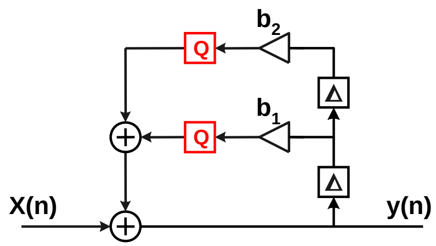
White noise input

Source: Fredrik Kristensen

Limit Cycles

Limit Cycles

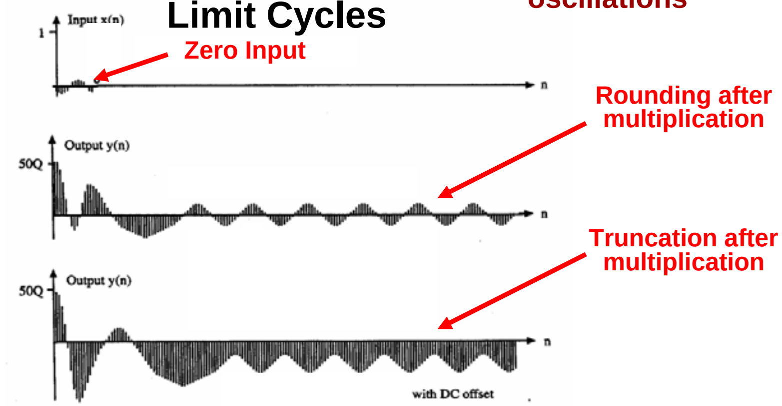
Example: zero input oscillations in 2nd order IIR



$$b_1 = \frac{489}{256} = 1.91015625; \; b_2 = -\frac{15}{16} = -0.9375$$

Limit Cycles

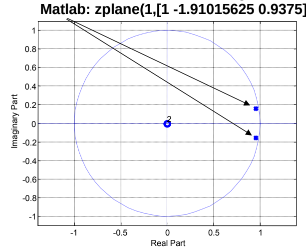
Example: zero input oscillations



Source: Lars Wanhammar, "DSP Integrated circuits"

Limit Cycles

Poles close to the unity circle
Matlab: zplane(1,[1 -1.91015625 0.9375])



Zero input oscillations

- Often not accepted in audio

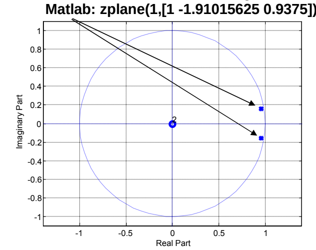
Very difficult problem

- In general, no solutions for structures
  > 2^{nd} order
- Can be limited by increased internal wordlength
- Can in some 2^{nd} order structures be eliminated by pole positioning
- 2^{nd} order Wave Digital Filters are free from parasitic oscillations

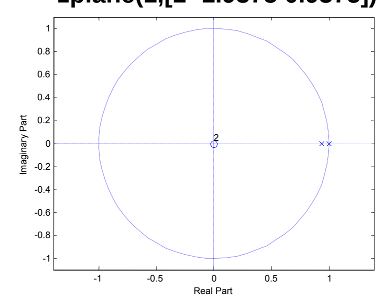
Limit Cycles

Changing the precision move the poles!

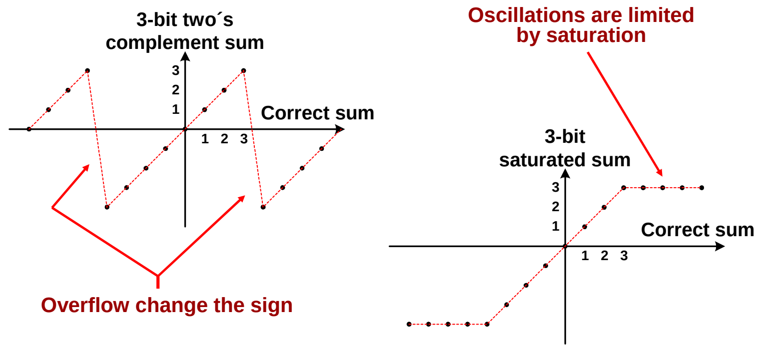
Poles close to the unity circle
Matlab: zplane(1,[1 -1.91015625 0.9375])



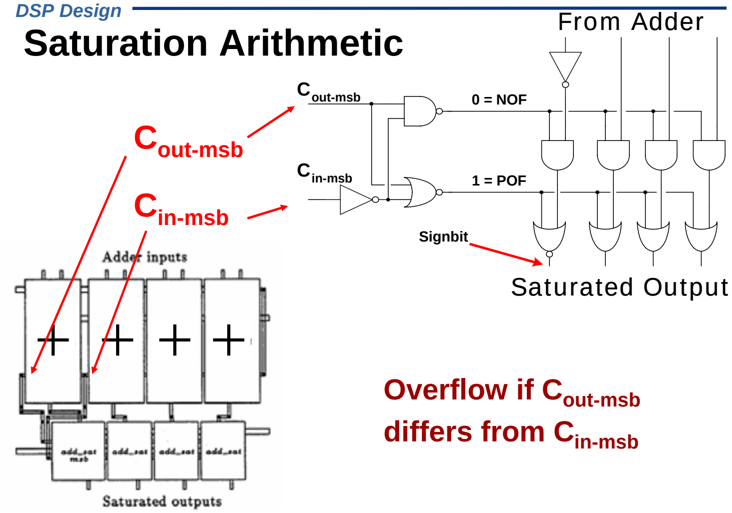
zplane(1,[1 -1.9375 0.9375])



Overflow Oscillations

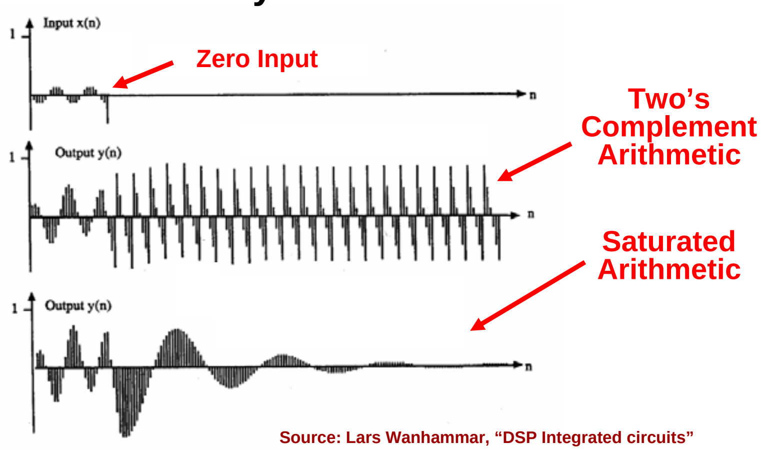


Saturation Arithmetic



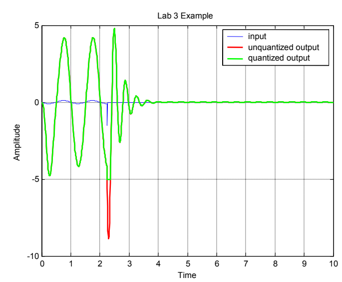
Overflow if $C_{out\text{-}msb}$ differs from $C_{in\text{-}msb}$

Limit Cycles due to overflow



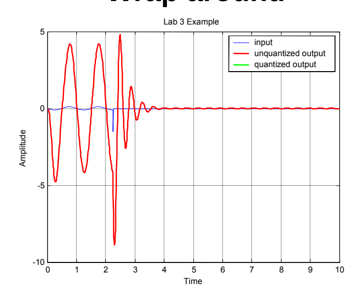
Source: Lars Wanhammar, "DSP Integrated circuits"

Limit cycles due to overflow

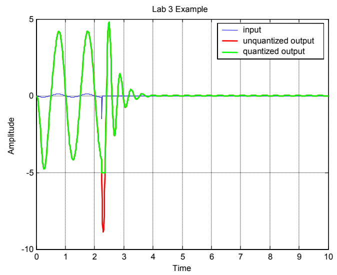


Saturation

Wrap around

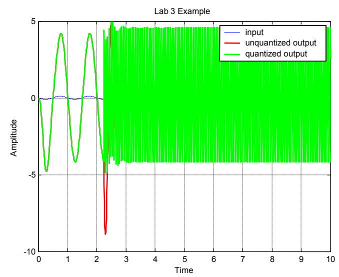


Limit cycles due to overflow



Saturation

Wrap around



Simple Noise Analysis

Scaling and White Noise Input

$$\beta = \sum_{i=0}^{\infty} |f(i)|, \; Safe - scaling$$

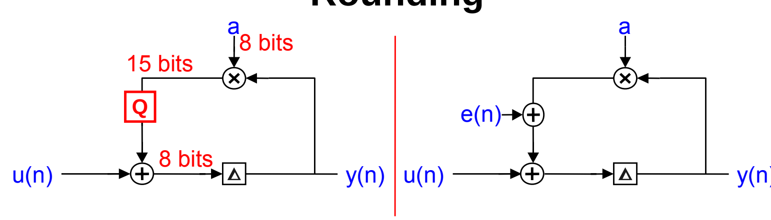
$$\beta = \delta \sqrt{\sum_{i=0}^{\infty} f^2(i)}, \; possible\ overflow$$

$f(i) =$ unit sample response, $\sum_{i=0}^{\infty} f^2(i) =$ Variance white noise input

- "Safe scaling" but not guaranteed
- δ sets the probability for an overflow
- Typically one overflow every 10^6 sample is accepted in audio [Wanhammar]

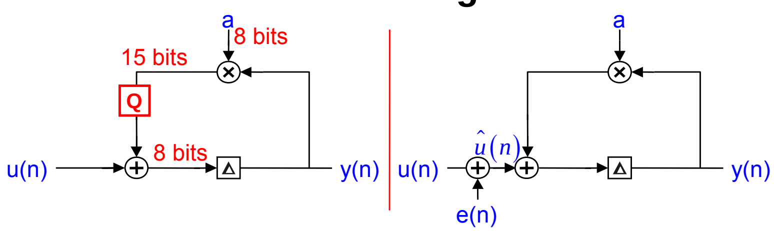
Rounding

Model



Rounding

Model



$$e(n) = \hat{u}(n) - u(n)$$

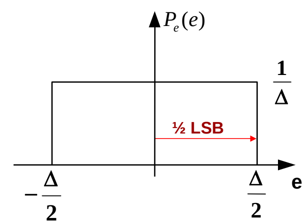
Modeled with added noise as an input error

Roundoff Noise

If the quantization error probability is uniformly distributed in the interval

$$-\frac{\Delta}{2} \le e(n) \le \frac{\Delta}{2} \quad where \quad \Delta = 2^{-(W-1)}$$

W **is the number of bits after the rounding**

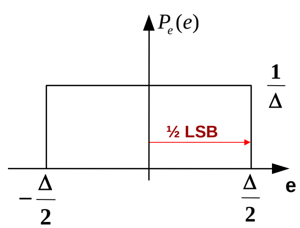


Roundoff Noise

$$Mean \ \ value \quad \bar{e} = E[e(n)] = 0$$

$$Variance = E[(e-\bar{e})^2] = \int_{-\Delta/2}^{\Delta/2} (e-\bar{e})^2 P_e(e)de = [\bar{e} = 0]$$

$$\left[\frac{1}{\Delta}\frac{e^3}{3}\right]_{-\Delta/2}^{\Delta/2} = \frac{1}{\Delta}\left[\frac{\Delta^3/8}{3} - \frac{-\Delta^3/8}{3}\right] = \frac{\Delta^2}{12} = \frac{2^{-2W}}{3}$$



Example: Roundoff Noise

In the case of rounding *(mean=0)* **the variance and the average power are the same, i.e. if a value is rounded the quantization noise becomes:**

$$\sigma_e^2 = \frac{2^{-2W}}{3}$$

If we scale down one bit:

$$\frac{2^{-2(W-1)}}{3} = \frac{2^2 \times 2^{-2W}}{3} = 4\sigma_e^2$$

Signal to Noise Ratio (SNR)

One extra bit reduces quantization error by a factor 4

$$SNR = 10\log\frac{4\sigma_e^2}{\sigma_e^2} = 6.02\ dB$$

Good to remember: 6 dB increase in SNR per bit

Signal to Noise Ratio (SNR)

Signal power (variance)

$$SNR = 10\log\frac{\sigma_x^2}{\sigma_e^2} = 10\log\frac{3}{2^{-2W}}\sigma_x^2$$

Roundoff error power (variance)

Signal to Noise Ratio (SNR)

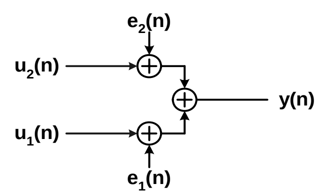
Example: Full scale sinus wave rounded to 8 bits

$$SNR = 10\log\frac{3}{2^{-2\times 8}}\left(\frac{A}{\sqrt{2}}\right)^2 = 50\ \text{ dB};\ -1 \le A \le 1$$

Roundoff Noise: Addition

$$E[(e_1 + e_2)^2] = E[e_1^2 + 2e_1e_2 + e_2^2] =$$

$$= E[e_1^2] + \underbrace{E[2e_1e_2]}_{\substack{\text{zero if} \\ u_1 \text{ and } u_2 \\ \text{independent}}} + E[e_2^2] =$$



$$= E[e_1^2] + E[e_2^2]$$

Example: Roundoff Noise

First order IIR-filter, the variance is:

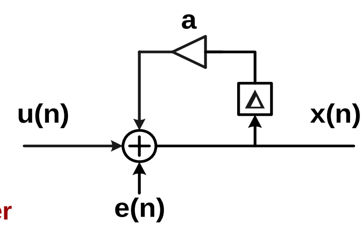
$$\sigma_e^2 \sum_{i=0}^{\infty} f^2(i) = \sigma_e^2 (1 + (a^1)^2 + (a^2)^2 + (a^3)^2 \cdots) = \sigma_e^2 \frac{1}{1-a^2}$$

$$a = 0.100 \quad \Rightarrow \quad 1.01\,\sigma_e^2$$

$$a = 0.500 \quad \Rightarrow \quad 1.33\,\sigma_e^2$$

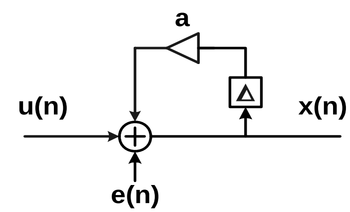
$$a = 0.998 \quad \Rightarrow \quad 500\,\sigma_e^2$$

Narrow band filter



Example: SNR

Example: Full scale sinus, rounded to 8 bits in IIR



$$\sigma_e^2 \sum_{i=0}^{\infty} f^2(i) = \sigma_e^2 \frac{1}{1-a^2}$$

$$\text{No feedback} \quad \Rightarrow \quad SNR = 50\,\text{dB}$$

$$a = 0.998 \quad \Rightarrow \quad 500\,\sigma_e^2 \quad \Rightarrow \quad SNR = 23\,\text{dB}$$