Fixed-point Numerics

V. Rajbabu

rajbabu@ee.iitb.ac.in

EE443: DSP Lab

Dept. of Electrical Engineering
IIT Bombay

24 Feb 2010

Outline

1. Fixed-point Representation
2. Quantization
3. Quantization Effects

Reference

Most of the materials used are from

- Fixed-point Signal Processing Systems

  (Manuscript in preparation) by

  Profs. Wayne T. Padgett, David V. Anderson, and Tyson S. Hall
- Prof.Preeti Rao's lecture

Outline

1. Fixed-point Representation
2. Quantization
3. Quantization Effects

Number Systems

Basic number systems

- Signed magnitude
- One's complement
- Two's complement

Two's complement

- Most commonly used in binary system
- Unique representation for zero
- Simple mathematical operations
- Subtraction performed using addition

Fractional Binary Numbers

- Designer keeps track of radix (decimal) point

Normalized Binary

Normalized M-bit binary numbers are written as:

$$x = b_0.b_1 b_2 \cdots b_{M-1}$$

where b_0 is the sign bit

Decimal representation

$$x_{(10)} = -b_0 + \sum_{i=1}^{M-1} b_i 2^{-i}$$

Number Circle

An easy way to visualize two's complement representation

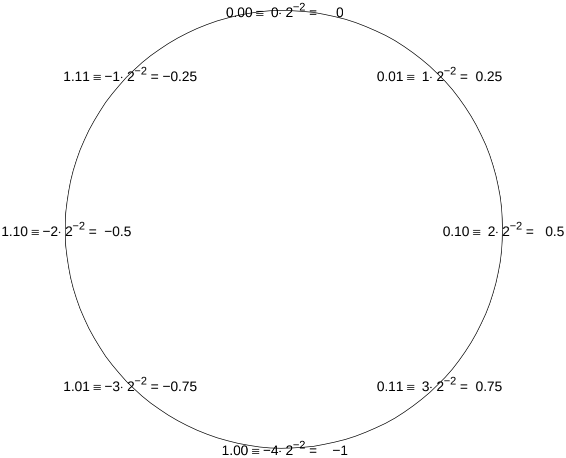


Figure: 3-bit $Q2$ numbers

Q-Format

Q-format is a formal mechanism to keep track of radix (fixed) point

Q-Format

$Q\#\#$ refers to a binary number with $\#\#$ bits to the right of the radix point

- Total word length depends on the system
- In DSPs, $Q15$ is a common format
- A 16-bit number in $Q15$ has 1 sign bit and 15 fractional bits

$$s.b_0 b_1 \cdots b_{14}$$

Q-Format - Example

Convert the following numbers to their signed integer value in $Q15$

Q-Format

$$\begin{array}{r} 0.5 \\ -0.5 \\ -1.0 \\ 1.0 \end{array}$$

Q-Format - Example

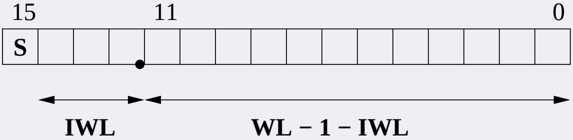
Convert the following numbers to their signed integer value in $Q15$

Q-Format

$$
\begin{aligned}
0.5 &= \quad 16384 \\
-0.5 &= -16384 \\
-1.0 &= -32768 \\
1.0 &= \text{out of range} \\
&\approx 32767 = 1 - 2^{-15}
\end{aligned}
$$

Q-Format Conversion

$Q12$ Number



$-2^{\mathrm{IWL}} \leq \text{Range} < 2^{IWL} - 2^{-12}$

Let x be a fractional number that needs to be represented as a B-bit (WL) signed integer, Qf format as x_q

- For positive x, $x_q = \texttt{round}(x \cdot 2^f)$
- For negative x, $x_q = -\texttt{round}(|x| \cdot 2^f)$

Q-Format: Addition

To obtain: $C = A + B$, with Q_c, Q_a, Q_b

- Require Q_a and Q_b to be equal
- Let M_a and M_b be size of registers for A and B
- Intermediate values
  - Intermediate result size $= \max(M_a, M_b) + 1$
  - Intermediate $Q_I = Q_a = Q_b$
- Final values
  - Top M_c bits are used, lowest fractional bits are discarded

$$\begin{aligned} Q_c &= Q_a - (M_c - \max(M_a, M_b) - 1) \\ &= Q_b - (M_c - \max(M_a, M_b) - 1) \end{aligned}$$

Adding N numbers of length M

Final word length : ???

Q-Format: Addition

To obtain: $C = A + B$, with Q_c, Q_a, Q_b

- Require Q_a and Q_b to be equal
- Let M_a and M_b be size of registers for A and B
- Intermediate values
  - Intermediate result size $= \max(M_a, M_b) + 1$
  - Intermediate $Q_I = Q_a = Q_b$
- Final values
  - Top M_c bits are used, lowest fractional bits are discarded

$$\begin{aligned} Q_c &= Q_a - (M_c - \max(M_a, M_b) - 1) \\ &= Q_b - (M_c - \max(M_a, M_b) - 1) \end{aligned}$$

Adding N numbers of length M

Final word length : $M + \lceil \log_2 N \rceil$

Q-Format: Multiplication

To obtain: $C = A \times B$, with Q_c, Q_a, Q_b

- Let M_a and M_b be size of registers for A and B
- M_a and M_b or Q_a and Q_b need not be equal
- Intermediate values
  - Intermediate result size $= M_a + M_b$
  - Intermediate $Q_I = Q_a + Q_b$
- Final values
  - Top M_c bits are used and lowest fractional bits are discarded
  - $Q_c = (Q_a + Q_b) - (M_a + M_b - M_c)$

Outline

1. Fixed-point Representation
2. Quantization
3. Quantization Effects

Definition

Quantization - represents numerical values with finite number of bits

Quantization in Signal processing

- data - round-off errors
- coefficients/parameters - changes system transfer function

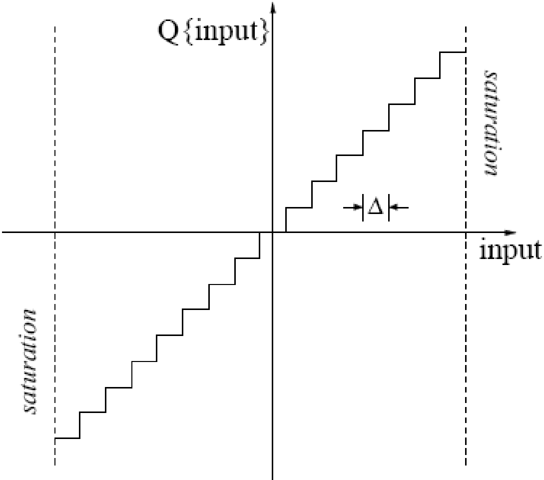
Quantization Errors

Possible errors in a quantized (fixed-point) system

- Input quantization
- Coefficient quantization
- Product quantization (round-off error, underflow)
- Overflow

Quantization Model

Data quantization using B-bit analog-to-digital converter (ADC) implies 2^B quantization levels



Quantization Model

Model quantization (non-linear) as additive noise (linear)

$$\mathbf{Q}\{x[n]\} = x[n] + e[n]$$

with the following assumptions

- $e[n]$ is uniformly distributed on $[-\triangle/2, \triangle/2]$?
- $e[n]$ is white noise
- $e[n]$ is uncorrelated with $x[n]$
- $x[n]$ is a stationary process

Quantization Model - Validity

Assumed model is valid, if

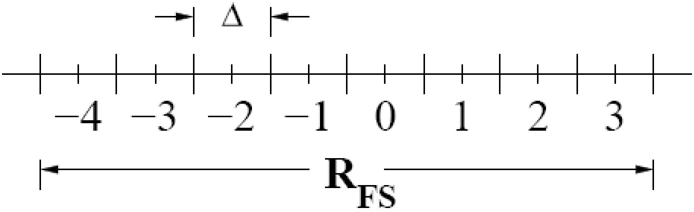
- Quantization steps are sufficiently small compared to the signal amplitude
- Sufficiently varying signal $x[n]$
- No saturation or overflow errors

Input Quantization

Full-scale range (R_{FS}) in 2ś complement representation

$$-(2^B+1)\frac{\triangle}{2} < x_{\text{input}}[nT] \leq (2^B-1)\frac{\triangle}{2} \quad (1)$$

where $\triangle = \frac{R_{FS}}{2^B}$



Quantization Error - (1/2)

- Amplitude of quantization error $e[n]$ is bounded by

$$\frac{\triangle}{2} < e[n] \leq \frac{\triangle}{2}$$

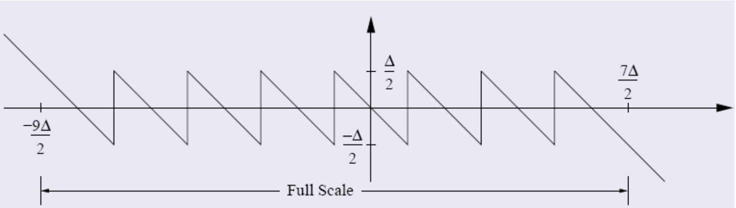


Figure: $e[n]$ as a function of $x[n]$ for a 3-bit ADC

Quantization Error - (2/2)

For $Q15$ representation: 15 bits for fraction, data range ± 1, the step size

$$\triangle = \frac{2}{2^{16}} = 2^{-15}$$

Range of quantization error

$$\pm\frac{\triangle}{2} = \pm 2^{-16}$$

Quantization Noise Power

- Quantization error (noise) is distributed uniformly in $[-\frac{\triangle}{2}, \frac{\triangle}{2}]$

  Is this true always ?

Noise power

$$\sigma_e^2 = \int_{-\frac{\triangle}{2}}^{\frac{\triangle}{2}} x^2 \frac{1}{\triangle} dx$$

$$= \frac{\triangle^2}{12}$$

Substituting for $\triangle$:

$$\sigma_e^2 = \frac{{R_{FS}}^2}{12 \cdot 2^{2B}}$$

Quantization Noise Power

- Quantization error (noise) is distributed uniformly in $[-\frac{\triangle}{2}, \frac{\triangle}{2}]$

  Is this true always ?

Noise power

$$\sigma_e^2 = \int_{-\frac{\triangle}{2}}^{\frac{\triangle}{2}} x^2 \frac{1}{\triangle} dx$$
$$= \frac{\triangle^2}{12}$$

Substituting for $\triangle$:

$$\sigma_e^2 = \frac{{R_{FS}}^2}{12 \cdot 2^{2B}}$$

Quantization Noise Power

- Quantization error (noise) is distributed uniformly in $[-\frac{\triangle}{2}, \frac{\triangle}{2}]$

  Is this true always ?

Noise power

$$\sigma_e^2 = \int_{-\frac{\triangle}{2}}^{\frac{\triangle}{2}} x^2 \frac{1}{\triangle} dx$$

$$= \frac{\triangle^2}{12}$$

Substituting for $\triangle$:

$$\sigma_e^2 = \frac{{R_{FS}}^2}{12 \cdot 2^{2B}}$$

Signal-to-Noise ratio (SNR) for Sinusoid

Signal-to-quantization-noise ratio

For a sinusoidal input $x[n] = A\cos(\omega n)$,

$$\text{Signal power} = \frac{A^2}{2}$$

$$\begin{aligned}\text{SNR for Sinusoid} &= \frac{A^2/2}{\triangle^2/12} \\ &= 6\frac{A^2}{\triangle^2}\end{aligned}$$

In $Q15$ format, $SNR = \frac{6}{2^{-30}} = 6.44 \times 10^9 = 98.09$ dB (CD Specification)

Signal-to-Noise ratio (SNR) per Bit for Sinusoid

For M bits, $\triangle = \frac{2}{2^M}$

$$\begin{aligned}\text{SNR for sinusoid} &= 6\frac{A^2}{4/2^{2M}} \\ &= \frac{3}{2}A^2 2^{2M}\end{aligned}$$

$$\begin{aligned}\text{SNR (dB)} &= 10\log_{10}(\frac{3}{2}A^2) + 2M\log_{10} 2 \\ &= 20\log_{10}(A) + 6.021M + 7.7815 \text{ dB}\end{aligned}$$

Signal-to-Noise ratio (SNR) - General Signals

For general signals, SNR depends on signal power $P_x = \sigma_x^2$ and quantization noise power $P_e = \sigma_e^2$

$$\begin{aligned}\text{SNR (dB)} &= 10\log_{10}\left(\frac{\sigma_x^2}{\frac{R_{FS}^2}{12\cdot 2^{2B}}}\right)\\ &= 10.792 + 6.021B + 20\log_{10}\left(\frac{\sigma_x}{R_{FS}}\right)\end{aligned}$$

Saturation and Overflow

NOTE

We have ignored saturation and overflow errors while analyzing quantization noise

Signal exceeds maximum or minimum quantization limits

- Saturation: input exceeds maximum representable value, quantization error is large
- Overflow: upper bits of sample are lost, signal is noise like

Designer has to perform appropriate scaling to eliminate or compensate for these errors

Matlab Tools

Fixed-point Toolbox

```
>> help fi
```

Link for CCS

```
>> help ccsdsp
```

Filter design

```
>>fdatool
```

Outline

1. Fixed-point Representation
2. Quantization
3. Quantization Effects

Quantization Effects

Finite word length effects

- Overflow errors

  Can be avoided by appropriate scaling

- Round-off errors

  Difficult to avoid - requires appropriate fixed-point arithmetic

Round-off Noise

Product of fixed-point numbers

- Product output requires more bits than inputs
- Truncation or rounding of result can lead to errors
    - Extended precision registers help in reducing this error

Sum of fixed-point numbers

- Output sum requires one-bit more than inputs
- Truncation or rounding of result can lead to errors
    - Not as severe in product

Scaling

Scaling

- Prevents overflow
- Provides a trade-off between SNR and overflow

Scaling in filter design/implementation

- Normalize inputs, coefficients to ± 1
- Based on magnitude of frequency response

Approaches to Scaling

Absolute scaling

- Scale assuming worst-case inputs/data
- Guarantees no overflow
- Leads to less accurate results (more quantization error)

Dynamic scaling

- Monitor range of variables and scale if required
- Increases computation

Floating-point to Fixed-point

- Implement and verify floating-point algorithm
  - Estimate minimum/maximum (range) of variables
- Convert floating-point variables to fixed-point
- Decide on scaling, based on architecture (word length)
  - Range of variables can help in fixing integer word length (IWL)
- Replace floating-point arithmetic with fixed-point arithmetic
  - Consider available accumulator and register word lengths

Fixed-point Arithmetic

Table: Fixed-point Arithmetic

Floating-point	Fixed-point		IWL of result
	$I_X > I_Y$	$I_X < I_Y$	
$X := Y$	$X := (Y \gg (I_X - I_Y))$	$X := (Y \ll (I_Y - I_X))$	I_X
$X + Y$	$X + (Y \gg (I_X - I_Y))$	$(X \gg (I_Y - I_X)) + Y$	$\max(I_X, I_Y) + 1$
$X * Y$	$X * Y$	$X * Y$	$I_X + I_Y$

[a] I_X, I_Y - Integer word length (IWL) of X and Y
[b] Overflow needs to be avoided for valid results