EEE336 Signal Processing and Digital Filtering

Lecture 4 Quantization

4_1 Introduction to Quantization

Zhao Wang

Zhao.wang@xjtlu.edu.cn

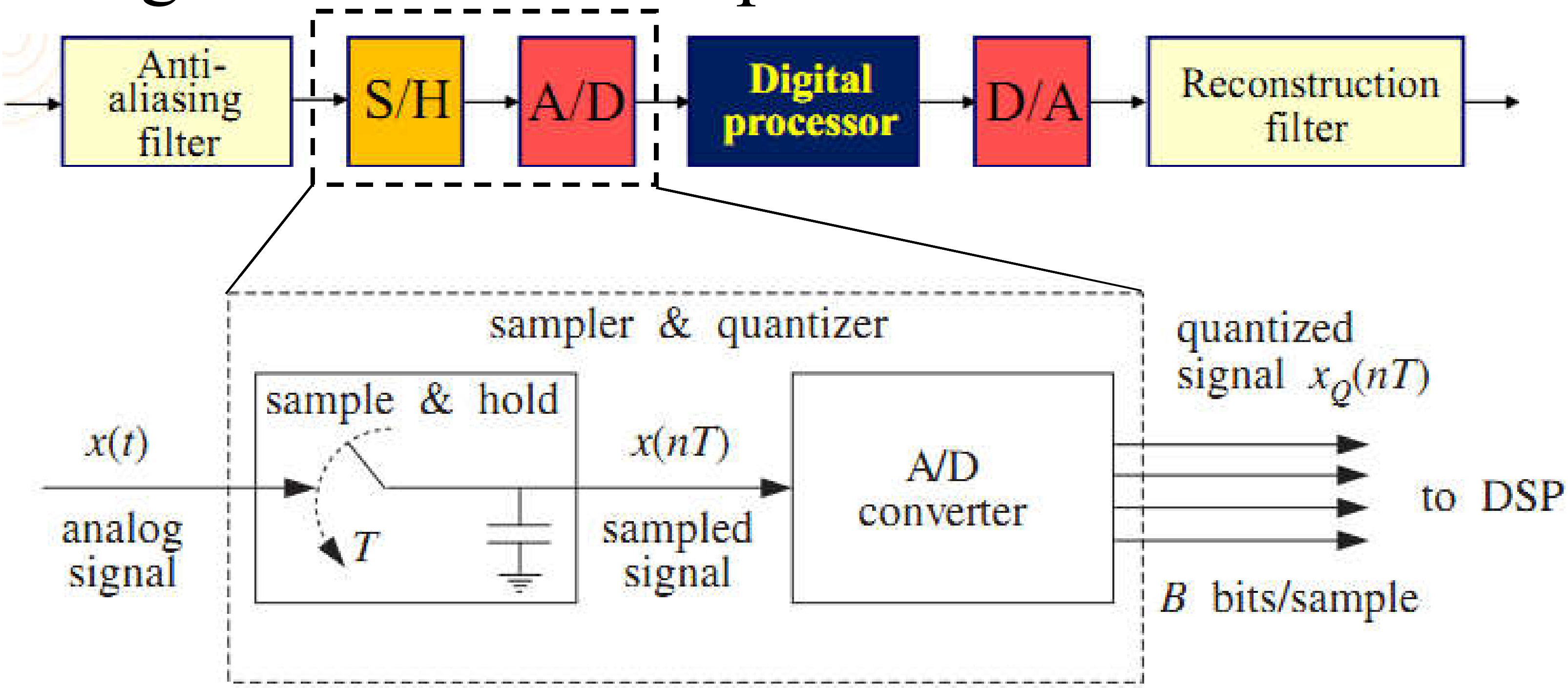
Room EE322

Xi'an Jiaotong-Liverpool University

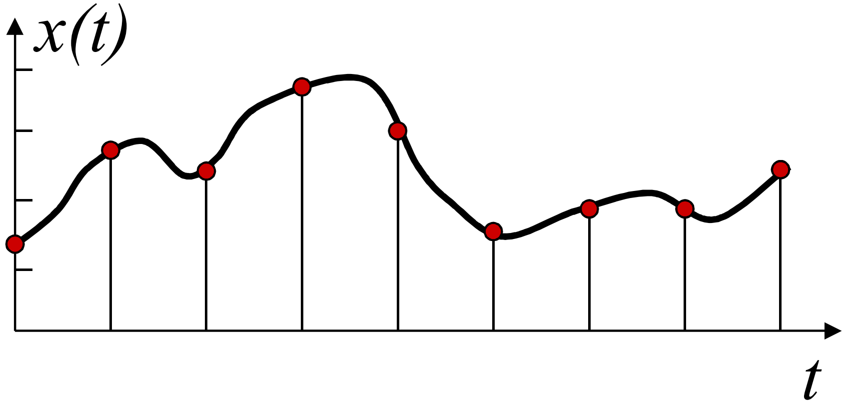
西交利物浦大學

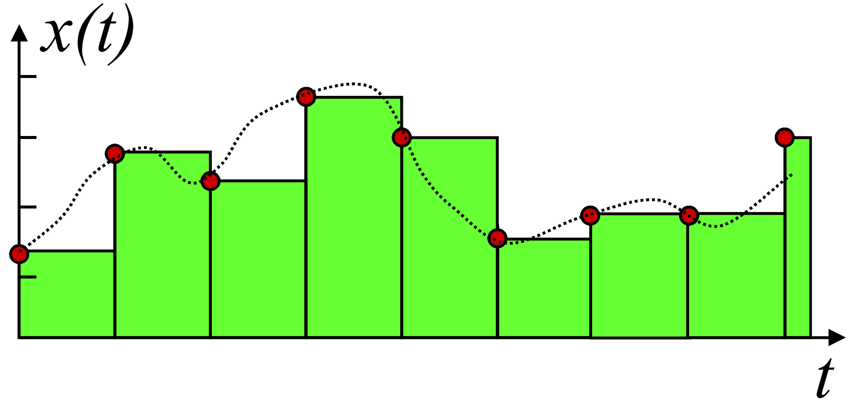
What is quantisation?

- Sampling: the process of converting a continuous-time signal into a discrete-time signal
- Quantization: converts a signal continuous in amplitude into a signal discrete in amplitude.

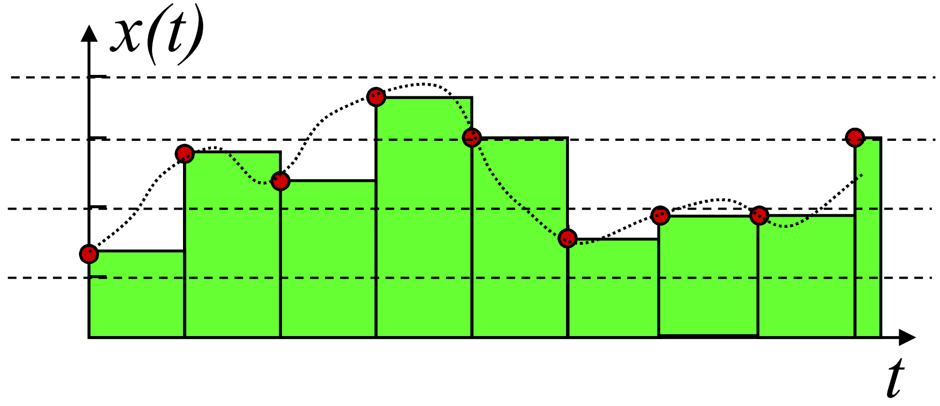


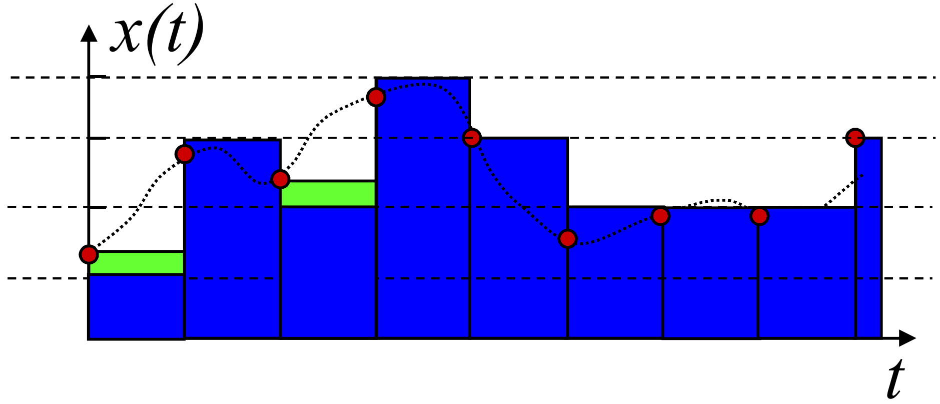
- The hold capacitor in the sampler holds each measured sample x(nT) for at most T seconds





- The A/D converter convert it to a quantized sample, $x_Q(nT)$, which is representable by a finite number of bits, say B bits.





Number representation

- Binary representation: the number is represented using the symbols 0 and 1, called ***bits***.
  - Eg: 10110101

1	**0**	**1**	**1**	**0**	**1**	**0**	**1**
a_{B-1}	a_{B-2}	…	…	…	…	a_{-1}	a_0

MSB

LSB

- The block of bits representing a number is called a ***word***, and the number of bits in the word is called the *wordlength* or *word size*.
  - *Wordlength*: integer power of 2, such as 8, 16, 32, etc.;
  - *Word size*: often expressed in units of 8 bits called a *byte*, such as a 4-byte word has the wordlength of 32 bits.

Number representation

- To avoid confusion between a decimal number containing only 1 and 0 and a binary number, we shall include a subscript 10 to the right of the LSB to indicate a decimal number, and a subscript 2 for binary number.
  - 1101_{10} represents a decimal number;
  - 1101_2 represents a binary number whose decimal equivalent is 13_{10}.

- There are two basic types of binary representations of numbers:
  - Fixed point: “int” in C/C++
  - Floating point: “float” in C/C++

Fixed and floating Point Numbers

- ***Fixed point representation***: the binary point is assumed to be fixed.
  - The range:

Non-negative integer	$0 \leq \eta \leq 2^{B}-1$
Negative-positive integer	$-2^{B-1} \leq \eta \leq 2^{B-1}-1$
Non-negative decimal	$0 \leq \eta \leq 1-2^{-B}$

  - ***Dynamic range* R** of the numbers that can be represented with B bits is given by $R = \eta_{max} - \eta_{min}$, where η_{max} and η_{min} are the maximum and the minimum values which can be represented.
  - The ***resolution*** of the representation is defined by:

$$Q = \frac{R}{2^B}$$

where Q is also known as the ***quantisation level***.

  - Fixed point is a simple representation, however causes worse finite word-length effects.

Fixed and floating Point Numbers

- ***Floating point representation***: in this case, a positive number η is represented using two parameters: the mantissa M and exponent, E in the form

$$\boxed{\eta = M \cdot 2^{E}}$$

  - E is either a positive or a negative binary integer;
  - M is a binary fraction restricted to lie in the range [0.5, 1)

- IEEE 32-bit floating-point format

$$\boxed{\eta = (-1)^{S} \cdot M \cdot 2^{E-127}}$$

  - Stored as:

0	**1**	**…**	**8**	**9**	**…**	**31**
S	E			M		

4_1 Wrap up

- The signals in the system:
  - Analog signal
  - Sampled and hold signal
  - Quantized signal
- Number representation
  - Decimal VS binary
- Fixed and floating point numbers
  - Fixed point
  - Floating point

EEE336 Signal Processing and Digital Filtering

Lecture 4 Quantization

4_2 Quantization and Errors

Zhao Wang

Zhao.wang@xjtlu.edu.cn

Room EE322

Xi'an Jiaotong-Liverpool University

西交利物浦大學

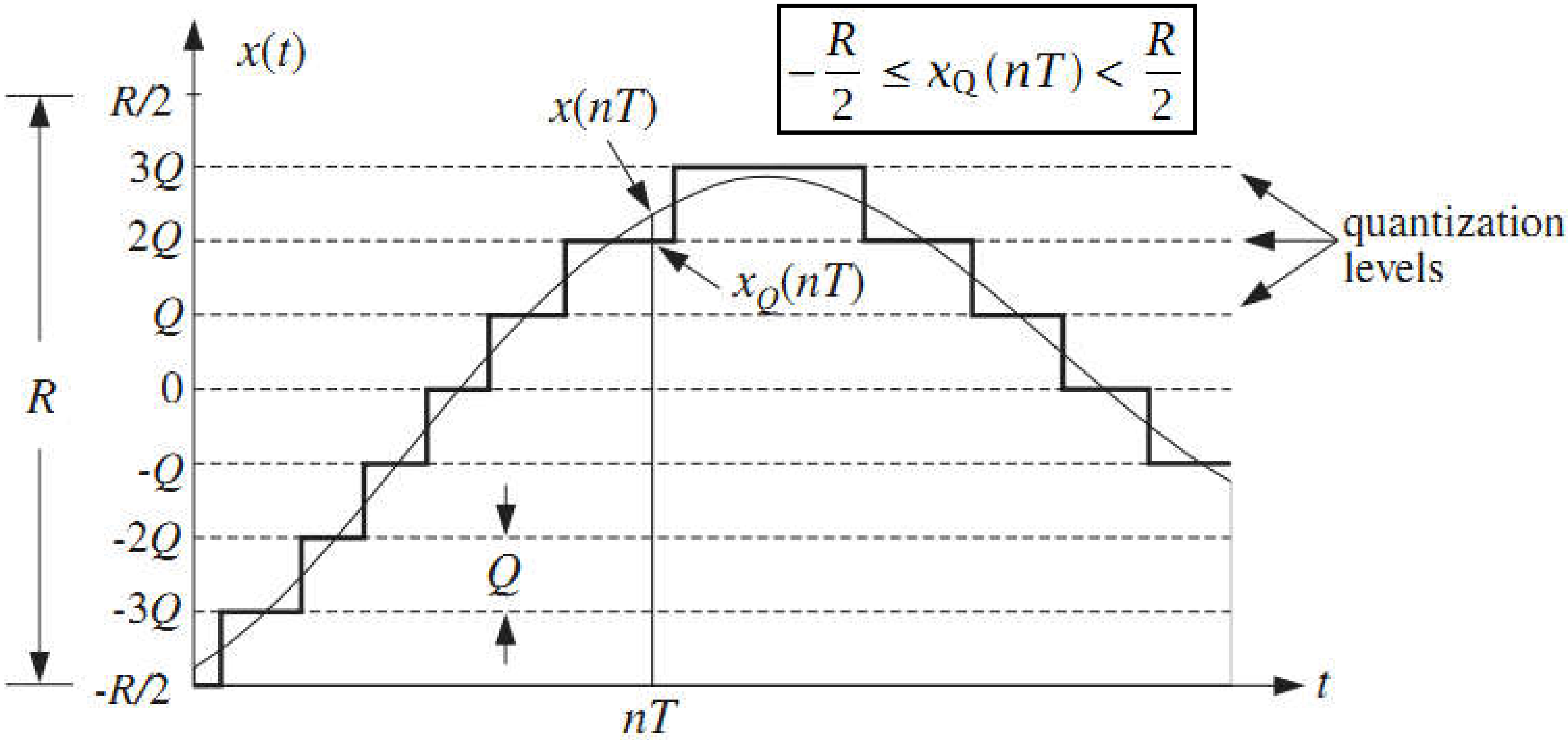
Quantisation Process

- R is the full-scale range which is divided equally (for a uniform quantizer) into 2^B *quantization levels*.
- The spacing between levels are called the *quantization width* / *quantization level* or *quantizer resolution* Q

$$\boxed{Q = \frac{R}{2^B}}$$

$$\Updownarrow$$

$$\boxed{\frac{R}{Q} = 2^B}$$



Quantisation Process

- Bipolar ADC: quantized values lie within the symmetric range:

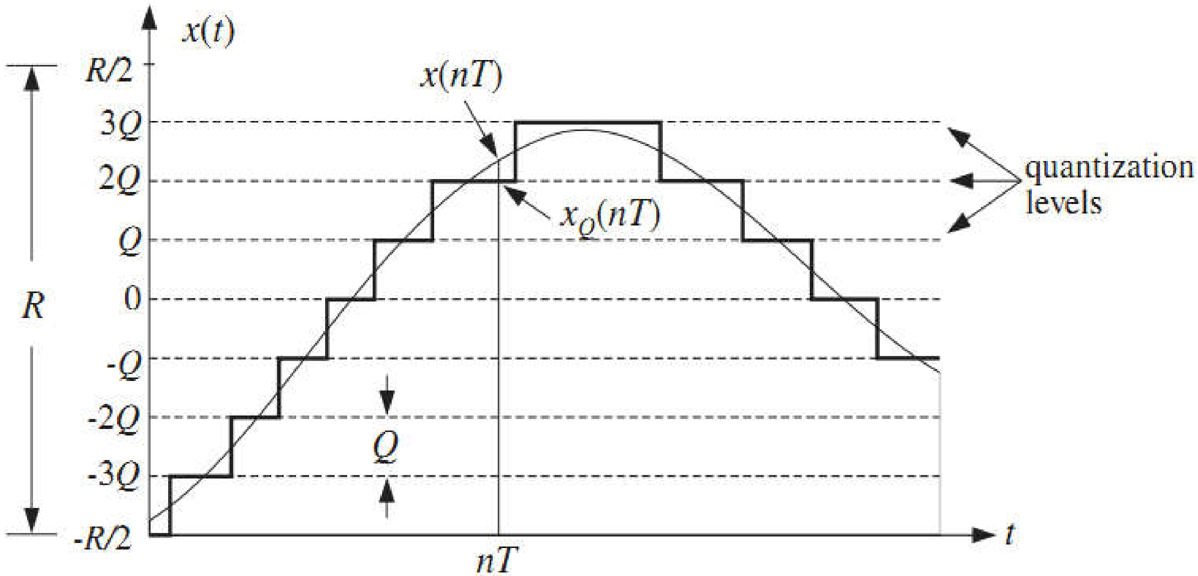
$$-\frac{R}{2} \leq x_Q(nT) < \frac{R}{2}$$

  – The upper end, R/2, of the full-scale range is not realized as one of the levels; rather, the maximum level is R/2 − Q.

- Unipolar ADC: quantized values lie within the asymmetric range:

$$0 \leq x_Q(nT) < R$$

- Quantization of x(t) was done by
  – Rounding
  – Truncation
  – Rounding is preferred in practice because it produces a less biased quantized representation of the analog signal.



Quantisation error

- The quantisation error is the error that results from using the quantised signal $x_Q(nT)$ instead of the true signal $x(nT)$

$$e(nT) = x_Q(nT) - x(nT) \quad \text{............(1)}$$

  - A more natural definition would be $e(nT) = x(nT) - x_Q(nT)$. The choice eq(1) is more convenient for making quantizer models.
  - If x lies between two levels, it will be rounded up or down depending on which is the closest level.
  - Therefore, the maximum error is $e_{max} = Q/2$ in magnitude. This is an overestimate for the typical error that occurs.
  - Error in quantizing a number x that lies in $[-R/2, R/2)$:

$$e = x_Q - x \quad \Longrightarrow \quad -\frac{Q}{2} \leq e \leq \frac{Q}{2}$$

Quantisation error

- To obtain a more representative value for the average error, we consider the mean and mean-square values of e:
  - Mean of error ***e***:

$$\overline{e} = \frac{1}{Q}\int_{-Q/2}^{Q/2} e\,de = 0$$

    - The result mean error e = 0 states that on the average half of the values are rounded up and half down. Thus, mean cannot be used as a representative error.
  - Mean-square of error ***e***:

$$\overline{e^2} = \frac{1}{Q}\int_{-Q/2}^{Q/2} e^2\,de = \frac{Q^2}{12}$$

  - A more typical value is the "Root-mean-square" of error ***e***:

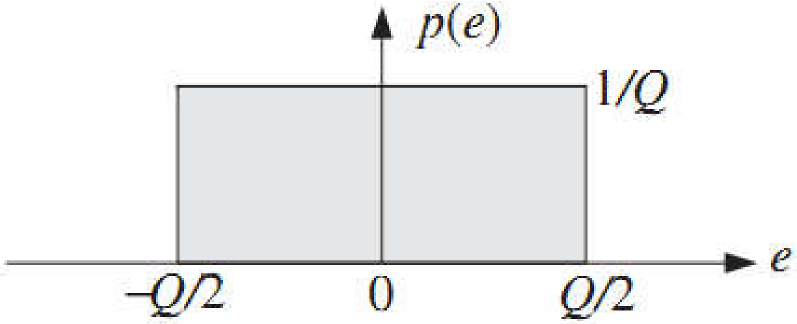
$$e_{\text{rms}} = \sqrt{\overline{e^2}} = \frac{Q}{\sqrt{12}}$$

Consider the mean and mean-square for "truncation"

Quantisation error

- The quantisation error ***e*** is a random variable which is distributed uniformly over the range [−Q/2, Q/2] with the probability density

$$p(e) = \begin{cases} \dfrac{1}{Q} & \text{if } -\dfrac{Q}{2} \leq e \leq \dfrac{Q}{2} \\ 0 & \text{otherwise} \end{cases}$$



- Signal-to-noise ratio (SNR)
  - R – range of signal
  - Q – range of noise

$$SNR = 20\log_{10}\left(\frac{R}{Q}\right) = 6B \text{ (dB)}$$

SNR is also called the *dynamic range* of the quantiser

4_2 Wrap up

- Key parameters in quantization:
  - Full scale range R
  - Quantization number of bits B
  - Quantization width / resolution Q
- Classification
  - Unipolar VS Bipolar
  - Rounding VS Truncation
- Quantization error
  - Error range, error mean and error root-mean-square

EEE336 Signal Processing and Digital Filtering

Lecture 4 Quantization

4_3 D/A and A/D Conversion

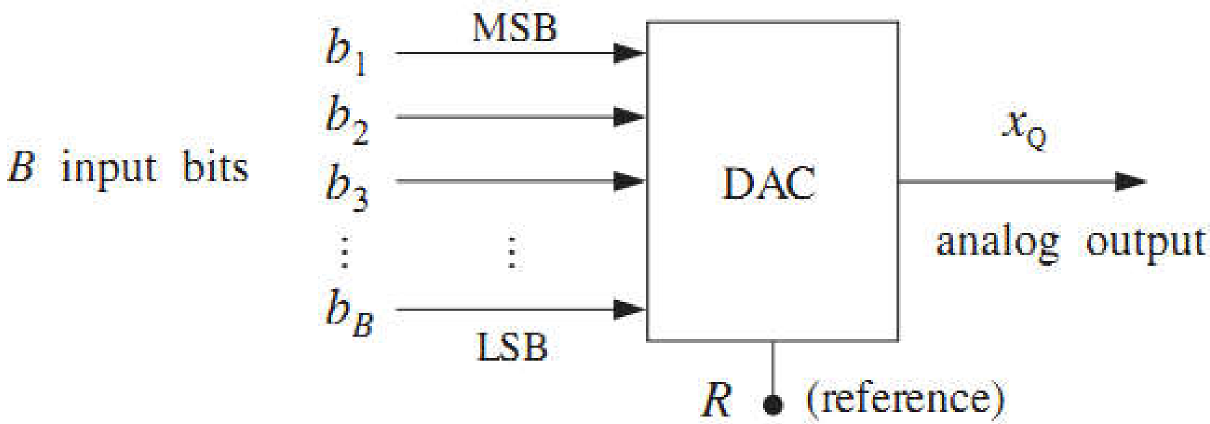
Zhao Wang

Zhao.wang@xjtlu.edu.cn

Room EE322

D/A Converters

- B-bit DAC with full-scale range R.
  - Input B bits of zeros and ones, $b = [b_1, b_2, \ . \ , b_B]$;
  - Output an analog value x_Q
    - x_Q lies on one of the 2^B quantization levels within the range R.
    - If the converter is unipolar, the output xQ falls in the range [0,R).
    - If it is bipolar, it falls in [−R/2,R/2).



B-bit D/A converter

An example

- Let's take the 4-bits, unipolar natural binary as an example:

2^{-1}	2^{-2}	2^{-3}	2^{-4}
b_1	b_2	b_3	b_4

MSB LSB

- The equation used to calculate the value expressed by this binary number is:

$$x_Q = R\left(b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + b_4 2^{-4}\right) \Longleftrightarrow x_Q = Qm$$

- Example: What does 1101 represent with the full scale range R = 8V?

D/A Converters

- Three types of converter and the coding conventions
  - Natural Binary: the unipolar natural binary

$$x_Q = R(b_1 2^{-1} + b_2 2^{-2} + \cdots + b_B 2^{-B}) \Longleftrightarrow x_Q = Qm \qquad (1)$$

where m is the integer whose binary representation is $(b_1 b_2 \cdots b_B)$

- LSB (Least Significant Bit): b_B
- MSB (Most Significant Bit): b_1

  - Offset Binary: the bipolar natural binary

$$x_Q = R(b_1 2^{-1} + b_2 2^{-2} + \cdots + b_B 2^{-B} - 0.5) \qquad (2)$$

  - 2's Combine: the two's complement

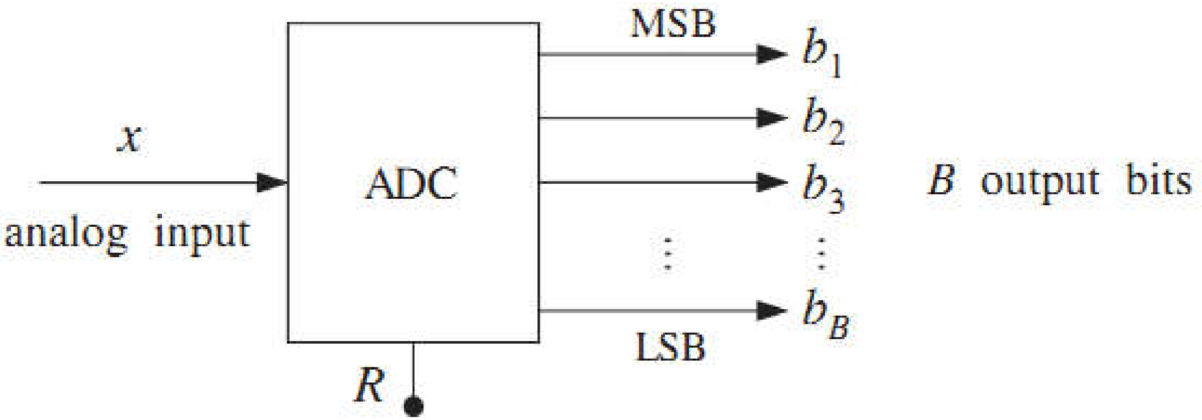
$$x_Q = R(\overline{b}_1 2^{-1} + b_2 2^{-2} + \cdots + b_B 2^{-B} - 0.5) \qquad (3)$$

Converter codes for B = 4 bits, R = 10 volts

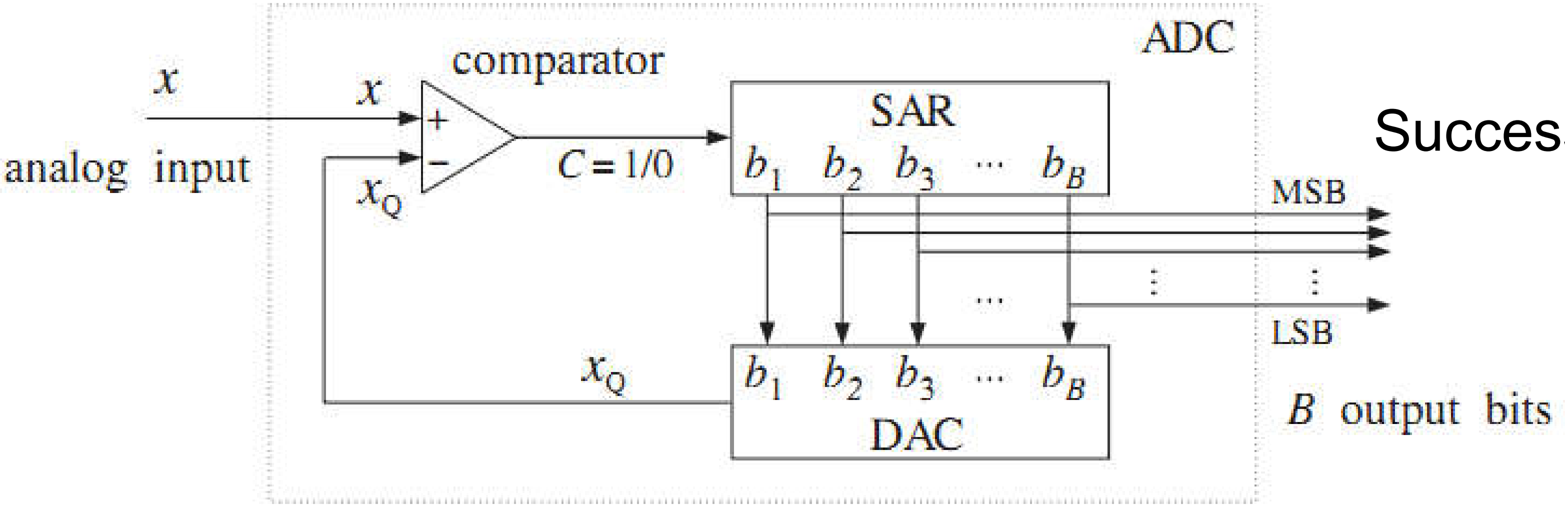
$b_1b_2b_3b_4$	natural binary		offset binary		2's C
	m	$x_Q = Qm$	m'	$x_Q = Qm'$	$b_1b_2b_3b_4$
—	16	10.000	8	5.000	—
1 1 1 1	15	9.375	7	4.375	0 1 1 1
1 1 1 0	14	8.750	6	3.750	0 1 1 0
1 1 0 1	13	8.125	5	3.125	0 1 0 1
1 1 0 0	12	7.500	4	2.500	0 1 0 0
1 0 1 1	11	6.875	3	1.875	0 0 1 1
1 0 1 0	10	6.250	2	1.250	0 0 1 0
1 0 0 1	9	5.625	1	0.625	0 0 0 1
1 0 0 0	8	5.000	0	0.000	0 0 0 0
0 1 1 1	7	4.375	−1	−0.625	1 1 1 1
0 1 1 0	6	3.750	−2	−1.250	1 1 1 0
0 1 0 1	5	3.125	−3	−1.875	1 1 0 1
0 1 0 0	4	2.500	−4	−2.500	1 1 0 0
0 0 1 1	3	1.875	−5	−3.125	1 0 1 1
0 0 1 0	2	1.250	−6	−3.750	1 0 1 0
0 0 0 1	1	0.625	−7	−4.375	1 0 0 1
0 0 0 0	0	0.000	−8	−5.000	1 0 0 0

A/D Converter

- A/D converters quantize an analog value x so that it is represented by B bits $[b_1, b_2, \ldots, b_B]$.



B-bit A/D converter



Successive approximation A/D converter

A/D Converter

- Algorithm of successive approximation A/D conversion
  - Initially all B bits are cleared to zero, b = [0, 0,..., 0], in SAR.
  - Then, starting with the MSB b_1, each bit is turned on in sequence and a test is performed to determine whether that bit should be left on or turned off.
  - The control logic puts the correct value of that bit in the right slot in the SAR register.
  - Then, leaving all the tested bits set at their correct values, the next bit is turned on in the SAR and the process repeated.
  - After B tests, the SAR will hold the correct bit vector $b = [b_1, b_2, \ldots, b_B]$, which can be sent to the output.

for each x *to be converted, do:*
    initialize $\mathbf{b} = [0, 0, \ldots, 0]$
    for $i = 1, 2, \ldots, B$ *do:*
        $b_i = 1$
        $x_Q = \mathrm{dac}(\mathbf{b}, B, R)$
        if $(x \geq x_Q)$
            $C = 1$
        else
            $C = 0$
        $b_i = C$

Implement truncation rather than rounding.

Examples

- Convert the analog values x = 3.5 and x =−1.5 volts to their offset binary representation, assuming B = 4 bits and R = 10 volts

x = 3.5

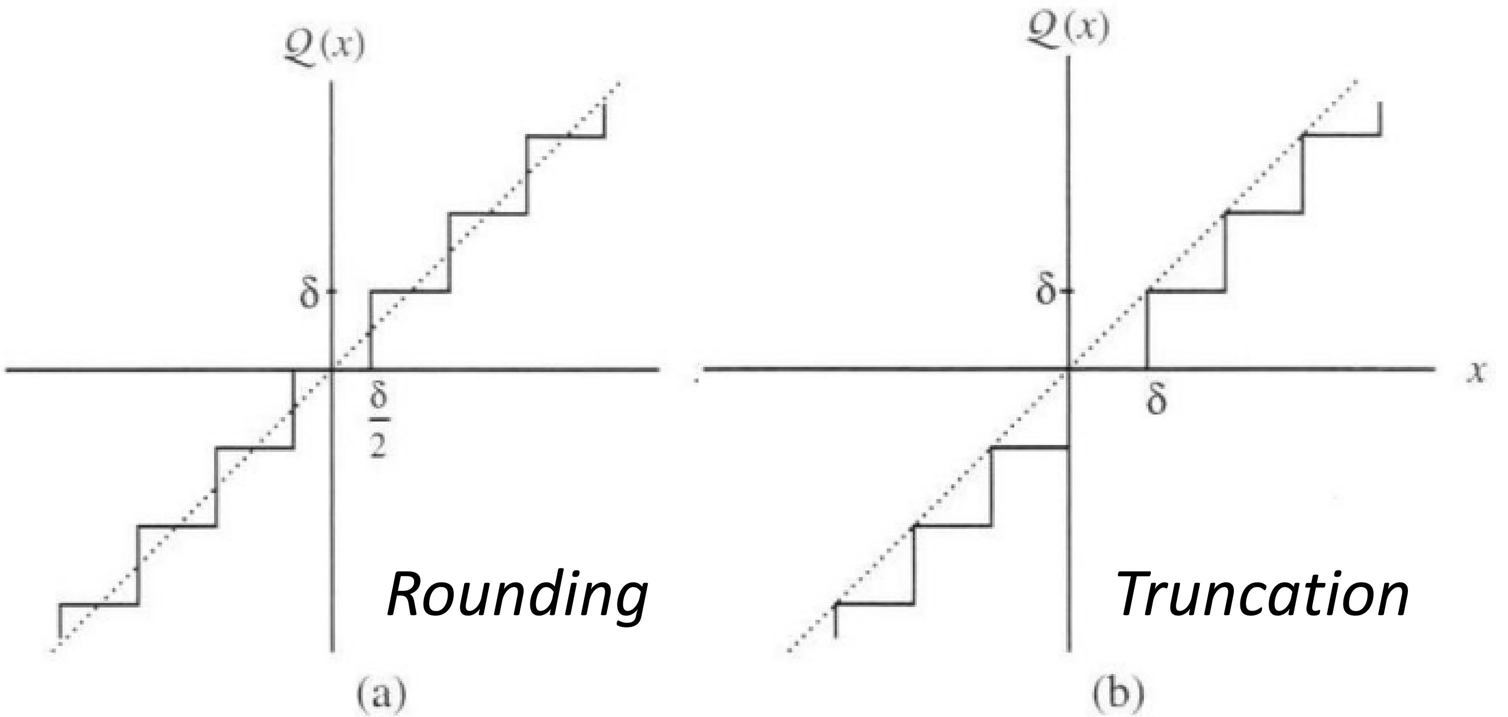
test	$b_1b_2b_3b_4$	x_Q	$C = u(x - x_Q)$
b_1	1 0 0 0	0.000	1
b_2	1 1 0 0	2.500	1
b_3	1 1 1 0	3.750	0
b_4	1 1 0 1	3.125	1
	1 1 0 1	3.125	

x =−1.5

test	$b_1b_2b_3b_4$	x_Q	$C = u(x - x_Q)$
b_1	1 0 0 0	0.000	0
b_2	0 1 0 0	−2.500	1
b_3	0 1 1 0	−1.250	0
b_4	0 1 0 1	−1.875	1
	0 1 0 1	−1.875	

Rounding and Truncation

- If the value of x[n] falls between two quantization levels, it will be either truncated or rounded.
- The input/output characteristic of the quantizer then depends on not only the truncation / rounding, but also on the type of number representation used:



A/D Converter

- In order to quantize by rounding to the nearest level, shift x by half the spacing between levels $\boxed{y = x + \frac{1}{2}Q}$
- Examples
  - To quantize the value x = 3.5 by rounding, we shift it to y = x + Q/2 = 3.5 + 0.625/2 = 3.8125.
  - For the case x =−1.5, we have y =−1.5 + 0.625/2 =−1.1875.

test	$b_1b_2b_3b_4$	x_Q	$C = u(y - x_Q)$
b_1	1 0 0 0	0.000	1
b_2	1 1 0 0	2.500	1
b_3	1 1 1 0	3.750	1
b_4	1 1 1 1	4.375	0
	1 1 1 0	3.750	

test	$b_1b_2b_3b_4$	x_Q	$C = u(y - x_Q)$
b_1	1 0 0 0	0.000	0
b_2	0 1 0 0	−2.500	1
b_3	0 1 1 0	−1.250	1
b_4	0 1 1 1	−0.625	0
	0 1 1 0	−1.250	

Chapter 4 Summary

- Conversion between analog and digital values
- D/A converter
  - What does a given binary word represent?
  - Three types of binary coding: natural, offset and 2's C.

- A/D converter
  - Successive approximation conversion
  - How to perform?
  - Rounding and truncation