Digital Signal Processing

405109

Chapter 4

FIR filtering and Convolution

Nguyen Thanh Tuan, M.Eng.
Department of Telecommunications (113B3)
Ho Chi Minh City University of Technology
Email: nttbk97@yahoo.com

Content

- Block processing methods
  - Convolution: direct form, convolution table
  - Convolution: LTI form, LTI table
  - Matrix form
  - Flip-and-slide form
  - Overlap-add block convolution method
- Sample processing methods
  - FIR filtering in direct form

Introduction

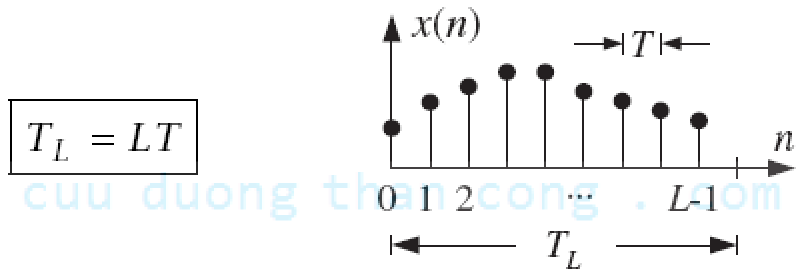
- Block processing methods: data are collected and processed in blocks.
  - FIR filtering of finite-duration signals by convolution
  - Fast convolution of long signals which are broken up in short segments
  - DFT/FFT spectrum computations
  - Speech analysis and synthesis
  - Image processing

- Sample processing methods: the data are processed one at a time-with each input sample being subject to a DSP algorithm which transforms it into an output sample.
  - Real-time applications
  - Digital audio effects processing
  - Digital control systems
  - Adaptive signal processing

1. Block Processing method

- The collected signal samples x(n), n=0, 1,…, L-1, can be thought as a block:

$$x=[x_0, x_1, \ldots, x_{L-1}]$$



The duration of the data record in second: $T_L=LT$

- Consider a casual FIR filter of order M with impulse response:

$$\mathbf{h}=[h_0, h_1, \ldots, h_M]$$

The length (the number of filter coefficients): $L_h=M+1$

1.1. Direct form

- The convolution in the direct form:

$$y(n)=\sum_m h(m)x(n-m)$$

- For DSP implementation, we must determine
  - The range of values of the output index n
  - The precise range of summation in m

- Find index n:

index of h(m) → $0 \le m \le M$
index of x(n-m) → $0 \le n-m \le L-1$

→ $0 \le m \le n \le m+L-1 \le M+L-1$

$$\boxed{0 \le n \le M+L-1}$$

- L_x=L input samples which is processed by the filter with order M yield the output signal y(n) of length $\boxed{L_y = L+M = L_x + M}$

Direct form

❖ Find index m: index of $h(m)$ → $0 \le m \le M$

index of $x(n-m)$ → $0 \le n-m \le L-1$ → $n+L-1 \le m \le n$

$$\boxed{\max(0,\ n-L+1) \le m \le \min(M,\ n)}$$

❖ The direct form of convolution is given as follows:

$$\boxed{y(n) = \sum_{m=\max(0,n-L+1)}^{\min(M,n)} h(m)x(n-m) = \mathbf{h} * \mathbf{x}} \quad \text{with} \quad \boxed{0 \le n \le M+L-1}$$

❖ Thus, **y** is longer than the input **x** by M samples. This property follows from the fact that a filter of order M has memory M and keeps each input sample inside it for M time units.

Example 1

❖ Consider the case of an order-3 filter and a length of 5-input signal. Find the output ?

$$\mathbf{h}=[h_0, h_1, h_2, h_3]$$

$$\mathbf{x}=[x_0, x_1, x_2, x_3, x_4]$$

$$\mathbf{y}=\mathbf{h}*\mathbf{x}=[y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7]$$

1.2. Convolution table

- It can be observed that $y(n) = \sum_{\substack{i,j \\ i+j=n}} h(i)x(j)$

- Convolution table

- The convolution table is convenient for quick calculation by hand because it displays all required operations compactly.

$\longrightarrow j$

$i \downarrow$

	x_0	x_1	x_2	x_3	x_4
h_0	h_0x_0	h_0x_1	h_0x_2	h_0x_3	h_0x_4
h_1	h_1x_0	h_1x_1	h_1x_2	h_1x_3	h_1x_4
h_2	h_2x_0	h_2x_1	h_2x_2	h_2x_3	h_2x_4
h_3	h_3x_0	h_3x_1	h_3x_2	h_3x_3	h_3x_4

Example 2

- Calculate the convolution of the following filter and input signals?

$$\mathbf{h}=[1, 2, -1, 1], \qquad \mathbf{x}=[1, 1, 2, 1, 2, 2, 1, 1]$$

- **Solution**:

h\x	1	1	2	1	2	2	1	1
1	1	1	2	1	2	2	1	1
2	2	2	4	2	4	4	2	2
-1	-1	-1	-2	-1	-2	-2	-1	-1
1	1	1	2	1	2	2	1	1

sum of the values along anti-diagonal line yields the output **y:**

$$\mathbf{y}=[1, 3, 3, 5, 3, 7, 4, 3, 3, 0, 1]$$

Note that there are $L_y=L+M=8+3=11$ output samples.

1.3. LTI Form

❖ LTI form of convolution: $y(n)=\sum_{m} x(m)h(n-m)$

❖ Consider the filter **h**=[h_0, h_1, h_2, h_3] and the input signal **x**=[x_0, x_1, x_2, x_3, x_4]. Then, the output is given by

$$y(n)=x_0h(n)+x_1h(n-1)+x_2h(n-2)+x_3h(n-3)+x_4h(n-4)$$

❖ We can represent the input and output signals as blocks:

$$\begin{aligned}
\mathbf{x} &= x_0[1,0,0,0,0] & \mathbf{y} &= x_0[h_0,h_1,h_2,h_3,0,0,0,0] \\
&+ x_1[0,1,0,0,0] & &+ x_1[0,h_0,h_1,h_2,h_3,0,0,0] \\
&+ x_2[0,0,1,0,0] \quad \xrightarrow{H} & &+ x_2[0,0,h_0,h_1,h_2,h_3,0,0] \\
&+ x_3[0,0,0,1,0] & &+ x_3[0,0,0,h_0,h_1,h_2,h_3,0] \\
&+ x_4[0,0,0,0,1] & &+ x_4[0,0,0,0,h_0,h_1,h_2,h_3]
\end{aligned}$$

1.3. LTI Form

❖ LTI form of convolution:

	h_0	h_1	h_2	h_3	0	0	0	0
x_0	x_0h_0	x_0h_1	x_0h_2	x_0h_3	0	0	0	0
x_1	0	x_1h_0	x_1h_1	x_1h_2	x_1h_3	0	0	0
x_2	0	0	x_2h_0	x_2h_1	x_2h_2	x_2h_3	0	0
x_3	0	0	0	x_3h_0	x_3h_1	x_3h_2	x_3h_3	0
x_4	0	0	0	0	x_4h_0	x_4h_1	x_4h_2	x_4h_3
	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7

cuu duong than cong . com

❖ LTI form of convolution provides a more intuitive way to under stand the linearity and time-invariance properties of the filter.

Example 3

❖ Using the LTI form to calculate the convolution of the following filter and input signals?

h=[1, 2, -1, 1], **x**=[1, 1, 2, 1, 2, 2, 1, 1]

❖ **Solution**:

n	0	1	2	3	4	5	6	7	8	9	10	
x\h	1	2	-1	1								partial output
1	1	2	-1	1								$x_0 h_n$
1		1	2	-1	1							$x_1 h_{n-1}$
2			2	4	-2	2						$x_2 h_{n-2}$
1				1	2	-1	1					$x_3 h_{n-3}$
2					2	4	-2	2				$x_4 h_{n-4}$
2						2	4	-2	2			$x_5 h_{n-5}$
1							1	2	-1	1		$x_6 h_{n-6}$
1								1	2	-1	1	$x_7 h_{n-7}$
y_n	1	3	3	5	3	7	4	3	3	0	1	$\sum_m x_m h_{n-m}$

1.4. Matrix Form

❖ Based on the convolution equations $y(n) = \sum_{m=\max(0,n-L+1)}^{\min(n,M)} h(m)x(n-m)$

we can write $\mathbf{y} = \mathbf{H}\mathbf{x}$

- ❑ $\mathbf{x}$ is the column vector of the L_x input samples.
- ❑ $\mathbf{y}$ is the column vector of the $L_y = L_x + M$ put samples.
- ❑ $\mathbf{H}$ is a rectangular matrix with dimensions $(L_x+M) \mathrm{x} L_x$.

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 & 0 \\ h_3 & h_2 & h_1 & h_0 & 0 \\ 0 & h_3 & h_2 & h_1 & h_0 \\ 0 & 0 & h_3 & h_2 & h_1 \\ 0 & 0 & 0 & h_3 & h_2 \\ 0 & 0 & 0 & 0 & h_3 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = H\mathbf{x}$$

1.4. Matrix Form

- It can be observed that **H** has the same entry along each diagonal. Such a matrix is known as Toeplitz matrix.

- Matrix representations of convolution are very useful in some applications:
  - Image processing
  - Advanced DSP methods such as parametric spectrum estimation and adaptive filtering

Example 4

❖ Using the matrix form to calculate the convolution of the following filter and input signals?

$$\mathbf{h}=[1, 2, -1, 1], \qquad \mathbf{x}=[1, 1, 2, 1, 2, 2, 1, 1]$$

❖ **Solution**: since Lx=8, M=3 → $L_y=L_x+M=11$, the filter matrix is 11x8 dimensional

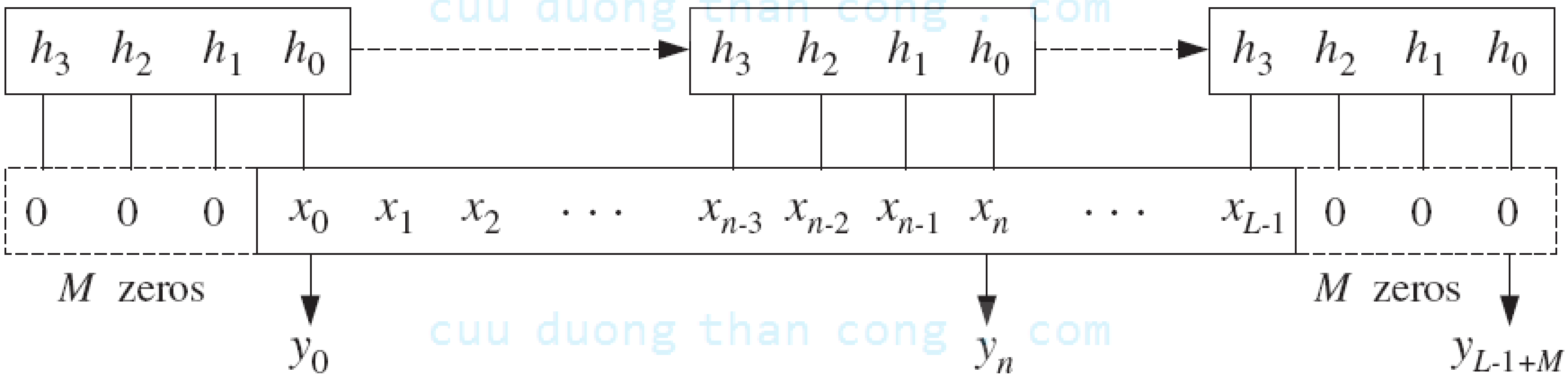
$$H\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 3 \\ 5 \\ 3 \\ 7 \\ 4 \\ 3 \\ 3 \\ 0 \\ 1 \end{bmatrix}$$

1.5. Flip-and-slide form

❖ The output at time n is given by

$$y_n = h_0 x_n + h_1 x_{n-1} + \ldots + h_M x_{n-M}$$

❖ Flip-and-slide form of convolution



❖ The flip-and-slide form shows clearly the input-on and input-off transient and steady-state behavior of a filter.

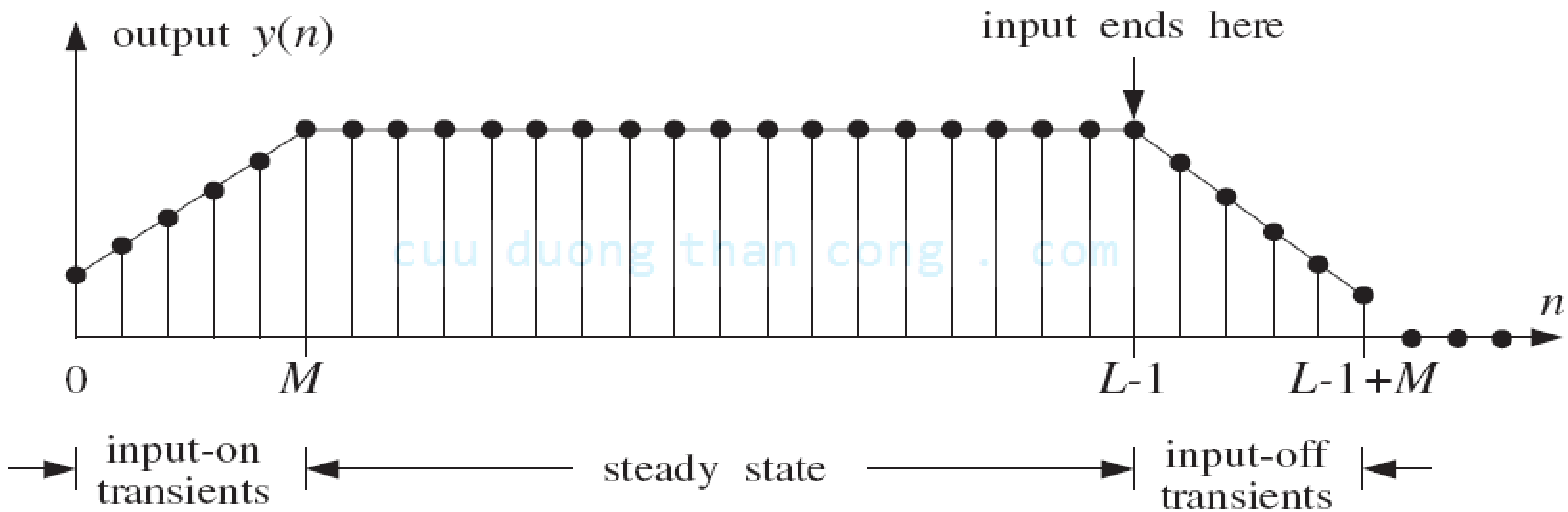
1.6. Transient and steady-state behavior

❖ From LTI convolution: $y(n) = \sum_{m=0}^{M} h(m)x(n-m) = h_0 x_n + h_1 x_{n-1} + \ldots + h_M x_{n-M}$

❖ The output is divided into 3 subranges:

$0 \le n < M$	(input-on transients)
$M \le n \le L-1$	(steady state)
$L-1 < n \le L-1+M$	(input-off transients)

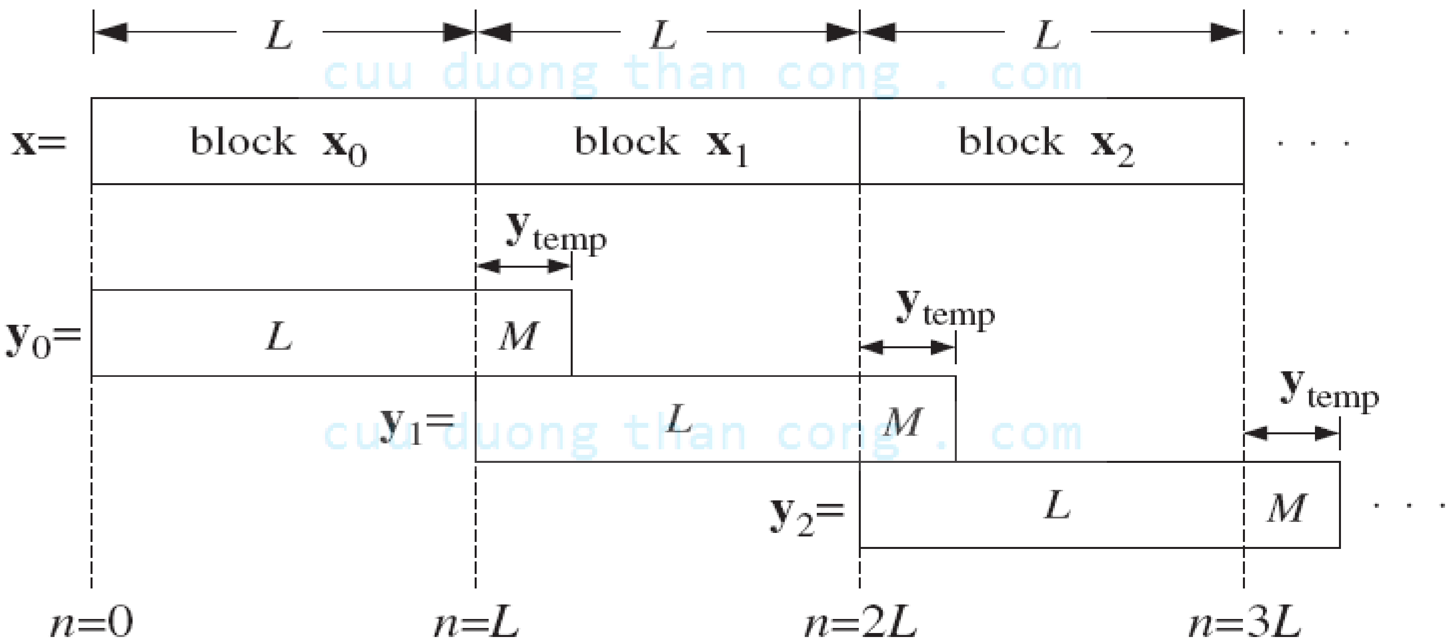
❖ Transient and steady-state filter outputs:



1.7. Overlap-add block convolution method

❖ As the input signal is infinite or extremely large, a practical approach is to divide the long input into contiguous non-overlapping blocks of manageable length, say L samples.

❖ Overlap-add block convolution method:



Example 5

- Using the overlap-add method of block convolution with each bock length L=3, calculate the convolution of the following filter and input signals? **h**=[1, 2, -1, 1], **x**=[1, 1, 2, 1, 2, 2, 1, 1]
- **Solution**: The input is divided into block of length L=3

$$\mathbf{x} = [\underbrace{1,1,2}_{\mathbf{x}_0},\underbrace{1,2,2}_{\mathbf{x}_1},\underbrace{1,1,0}_{\mathbf{x}_2}]$$

The output of each block is found by the convolution table:

	block 0			block 1			block 2		
h\x	1	1	2	1	2	2	1	1	0
1	1	1	2	1	2	2	1	1	0
2	2	2	4	2	4	4	2	2	0
-1	-1	-1	-2	-1	-2	-2	-1	-1	0
1	1	1	2	1	2	2	1	1	0

Example 5

❖ The output of each block is given by

$$\mathbf{y}_0 = \mathbf{h} * \mathbf{x}_0 = [1, 3, 3, 4, -1, 2]$$

$$\mathbf{y}_1 = \mathbf{h} * \mathbf{x}_1 = [1, 4, 5, 3, 0, 2]$$

$$\mathbf{y}_2 = \mathbf{h} * \mathbf{x}_2 = [1, 3, 1, 0, 1, 0]$$

❖ Following from time invariant, aligning the output blocks according to theirs absolute timings and adding them up gives the final results:

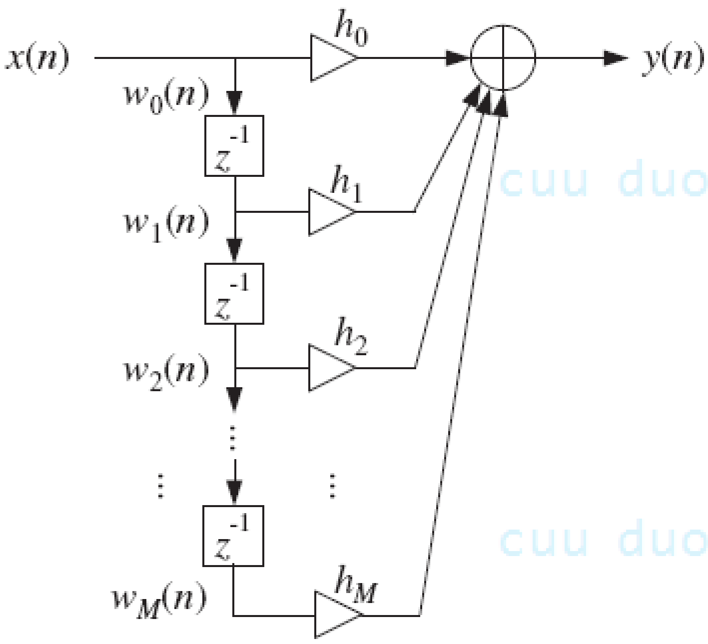
n	0	1	2	3	4	5	6	7	8	9	10
$\mathbf{y}_0$	1	3	3	4	-1	2					
$\mathbf{y}_1$				1	4	5	3	0	2		
$\mathbf{y}_2$							1	3	1	0	1
$\mathbf{y}$	1	3	3	5	3	7	4	3	3	0	1

2. Sample processing methods

- The direct form convolution for an FIR filter of order M is given by

$$y(n) = h_0x(n) + h_1x(n-1) + \cdots + h_Mx(n-M)$$

- Introduce the internal states $w_i(n) = x(n-i)$



$$y(n) = h_0w_0(n) + h_1w_1(n) + \cdots + h_Mw_M(n)$$

Sample processing algorithm

for each input sample x do:

$w_0 = x$

$y = h_0w_0 + h_1w_1 + \cdots + h_Mw_M$

for $i = M, M-1, \ldots, 1$ *do:*

$w_i = w_{i-1}$

Fig: Direct form realization of Mth order filter

- Sample processing methods are convenient for real-time applications

Example 6

❖ Consider the filter and input given by

$$\mathbf{h} = [1, 2, -1, 1], \qquad \mathbf{x} = [1, 1, 2, 1, 2, 2, 1, 1]$$

Using the sample processing algorithm to compute the output and show the input-off transients.

Example 6

Solution: The I/O equation of this filter is

$$y(n) = x(n) + 2x(n-1) - x(n-2) + x(n-3)$$

Introducing the internal states $w_i(n) = x(n-i)$, $i = 1, 2, 3$, and setting $w_0(n) = x(n)$, we obtain the following system describing the output equation and the state updating:

$$w_0(n) = x(n)$$

$$y(n) = w_0(n) + 2w_1(n) - w_2(n) + w_3(n)$$

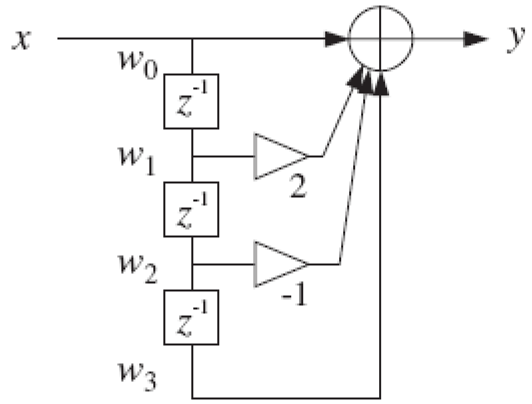
$$w_3(n+1) = w_2(n)$$

$$w_2(n+1) = w_1(n)$$

$$w_1(n+1) = w_0(n)$$

The corresponding block diagram realization and sample processing algorithm are shown below:

Example



for each input sample x do:

$$w_0 = x$$
$$y = w_0 + 2w_1 - w_2 + w_3$$
$$w_3 = w_2$$
$$w_2 = w_1$$
$$w_1 = w_0$$

The sample processing algorithm generates the following output samples:

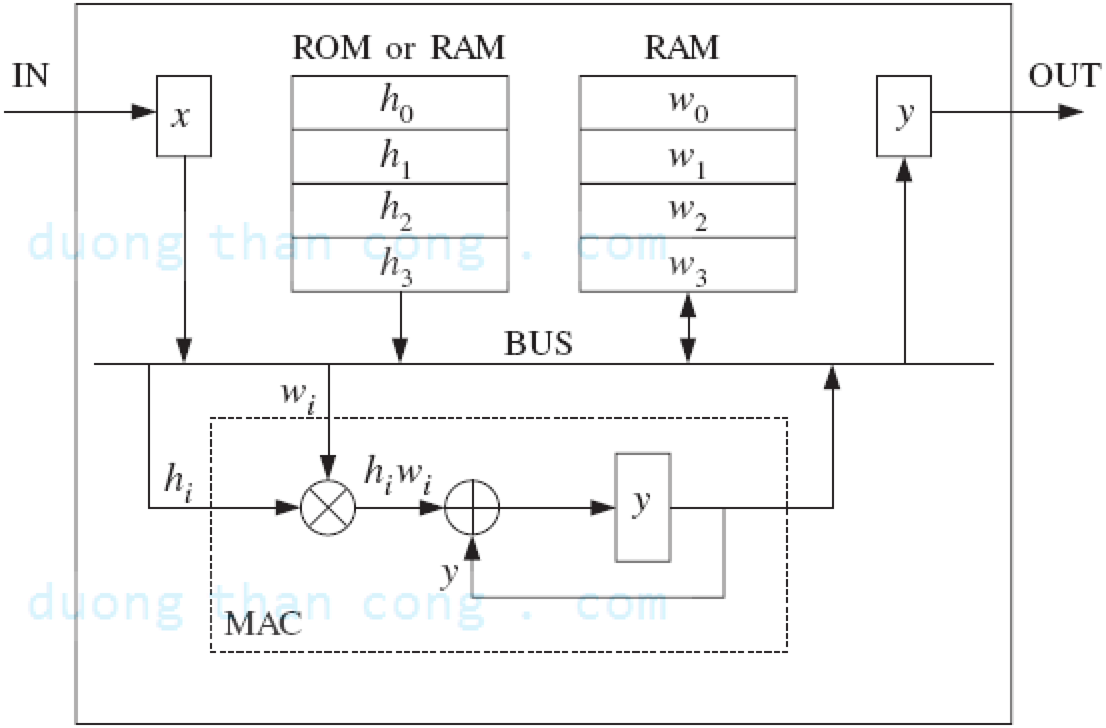
n	x	w_0	w_1	w_2	w_3	$y = w_0 + 2w_1 - w_2 + w_3$
0	1	1	0	0	0	1
1	1	1	1	0	0	3
2	2	2	1	1	0	3
3	1	1	2	1	1	5
4	2	2	1	2	1	3
5	2	2	2	1	2	7
6	1	1	2	2	1	4
7	1	1	1	2	2	3
8	0	0	1	1	2	3
9	0	0	0	1	1	0
10	0	0	0	0	1	1

Hardware realizations

❖ The FIR filtering algorithm can be realized in hardware using DSP chips, for example the Texas Instrument TMS320C25

❖ MAC: Multiplier Accumulator

$$y := y + h_i w_i$$



Hardware realizations

❖ The signal processing methods can efficiently rewritten as

$$
\begin{aligned}
&\textit{for each input sample } x \textit{ do:}\\
&\quad w_0 := x\\
&\quad y := h_M w_M\\
&\quad \textit{for } i = M-1, \dots, 1, 0 \textit{ do:}\\
&\quad\quad w_{i+1} := w_i\\
&\quad\quad y := y + h_i w_i
\end{aligned}
$$

❖ In modern DSP chips, the two operations

$$w_{i+1} := w_i$$
$$y := y + h_i w_i$$

can carried out with a single instruction.

❖ The total processing time for each input sample of Mth order filter:

$$\boxed{T_{\text{proc}} = (M+1)\,T_{\text{instr}}}$$

where T_{instr} is one instruction cycle in about 30-80 nanoseconds.

❖ For real-time application, it requires that

$$T \ge T_{\text{proc}} \quad \Rightarrow \quad f_s \le \frac{1}{T_{\text{proc}}}$$

Example 7

❖ What is the longest FIR filter that can be implemented with a 50 nsec per instruction DSP chip for digital audio applications with sampling frequency f_s=44.1 kHz ?

Solution:

$$T = (M+1)T_{\text{instr}} \quad \Rightarrow \quad M+1 = \frac{T}{T_{\text{instr}}} = \frac{1}{f_s T_{\text{instr}}} = \frac{f_{\text{instr}}}{f_s}$$

where the *instruction rate* is $f_{\text{instr}} = 1/T_{\text{instr}} = 20$ million instructions per second (MIPS). For digital audio at $f_s = 44.1$ kHz, we find

$$M+1 = \frac{f_{\text{instr}}}{f_s} = \frac{20 \cdot 10^6}{44.1 \cdot 10^3} = 453 \text{ taps}$$

This filter length is quite sufficient to implement several digital audio algorithms. □

Homework 1

The impulse response $h(n)$ of a filter is nonzero over the index range $3 \le n \le 6$. The input signal $x(n)$ to this filter is nonzero over the index range $10 \le n \le 20$. Consider the direct and LTI forms of convolution:

$$y(n) = \sum_m h(m)x(n-m) = \sum_m x(m)h(n-m)$$

a. Determine the overall index range n for the output $y(n)$. For each n, determine the corresponding summation range over m, for both the direct and LTI forms.

b. Assume $h(n) = 1$ and $x(n) = 1$ over their respective index ranges. Calculate and sketch the output $y(n)$. Identify (with an explanation) the input on/off transient and steady state parts of $y(n)$.

Homework 2

An LTI filter has infinite impulse response $h(n) = a^n u(n)$, where $|a| < 1$. Using the convolution summation formula $y(n) = \sum_m h(m)x(n-m)$, derive closed-form expressions for the output signal $y(n)$ when the input is:

a. A unit step, $x(n) = u(n)$

b. An alternating step, $x(n) = (-1)^n u(n)$.

In each case, determine the steady state and transient response of the filter.

Homework 3

Let $\mathbf{x} = [1, 1, 2, 2, 2, 2, 1, 1]$ be an input to the filter described by the I/O equation:

$$y(n) = x(n) - x(n-2) + 2x(n-3)$$

a. Determine the impulse response $h(n)$ of this filter.

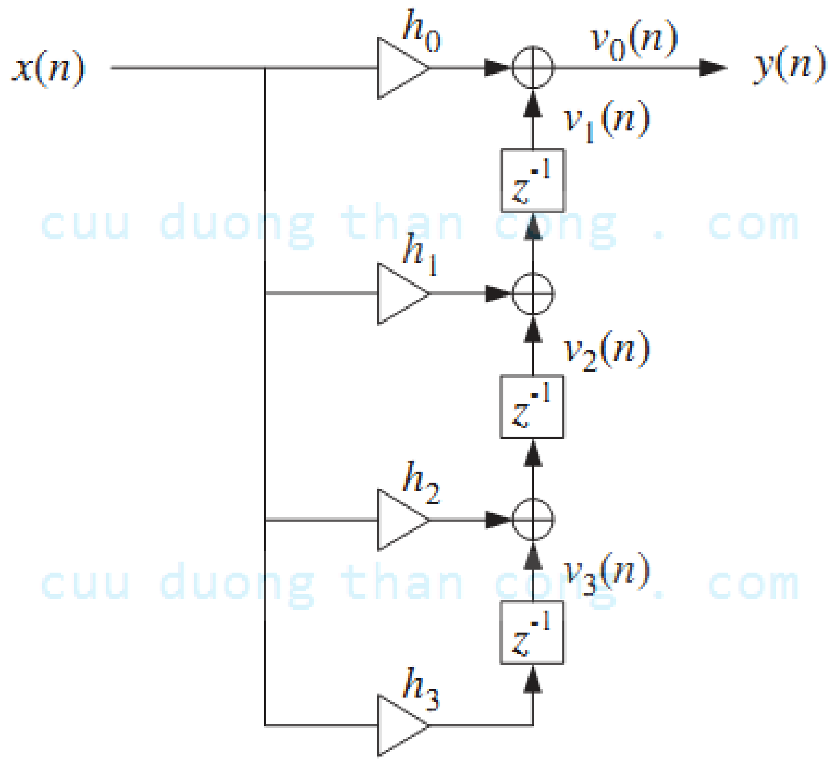
b. Compute the corresponding output signal $y(n)$ using the *LTI form* of convolution. Show your computations in table form.

c. Compute the same output using the overlap-add method of block convolution by partitioning the input signal into length-4 blocks.

d. Draw a block diagram realization of this filter. Then, introduce appropriate internal states and write the corresponding sample processing algorithm.

Homework 4

❖ Compute the output y(n) of the filter h(n) = {1, -1, 1, -1} and input x(n) = {1, 2, 3, 4, @, -3, 2, -1}



Homework 5

❖ Compute the convolution, $y = h * x$, of the filter and input, $h(n) = \{1, -1, -1, 1\}$, $x(n) = \{1, 2, 3, 4, @, -3, 2, -1\}$ using the following methods:

1. The convolution table.
2. The LTI form of convolution, arranging the computations in a table form.
3. The overlap-add method of block convolution with length-3 input blocks.
4. The overlap-add method of block convolution with length-4 input blocks.
5. The overlap-add method of block convolution with length-5 input blocks.