# Signal Processing First

## Lecture 10
## Filtering Intro

# READING ASSIGNMENTS

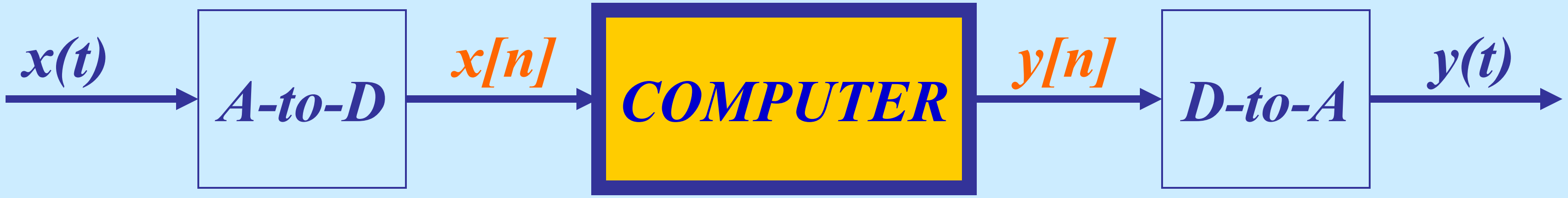
- This Lecture:
  - Chapter 5, Sects. 5-1, 5-2 and 5-3 (partial)

- Other Reading:
  - Recitation: Ch. 5, Sects 5-4, 5-6, 5-7 and 5-8
    - CONVOLUTION
  - Next Lecture: Ch 5, Sects. 5-3, 5-5 and 5-6

# LECTURE OBJECTIVES

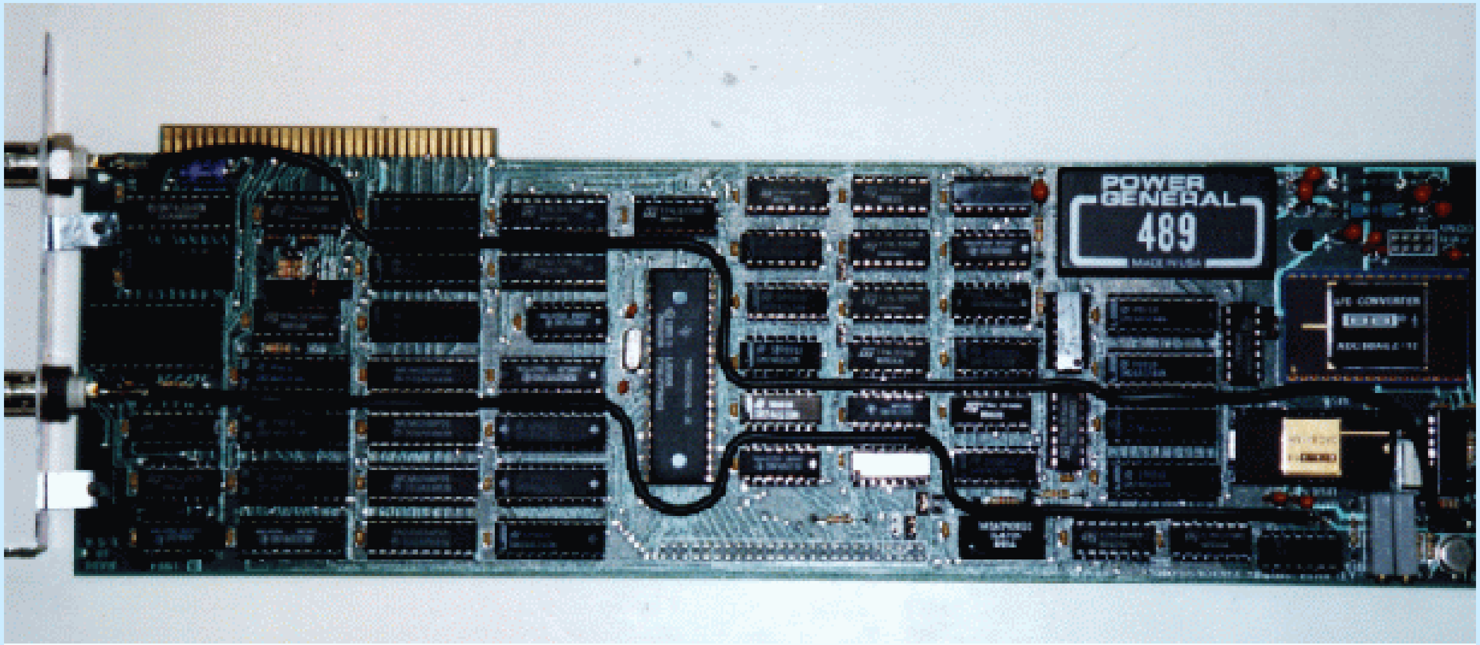
- INTRODUCE FILTERING IDEA
  - Weighted Average
  - Running Average
- FINITE IMPULSE RESPONSE FILTERS
  - **FIR** Filters
  - Show how to **compute** the output y[n] from the input signal, x[n]

# DIGITAL FILTERING



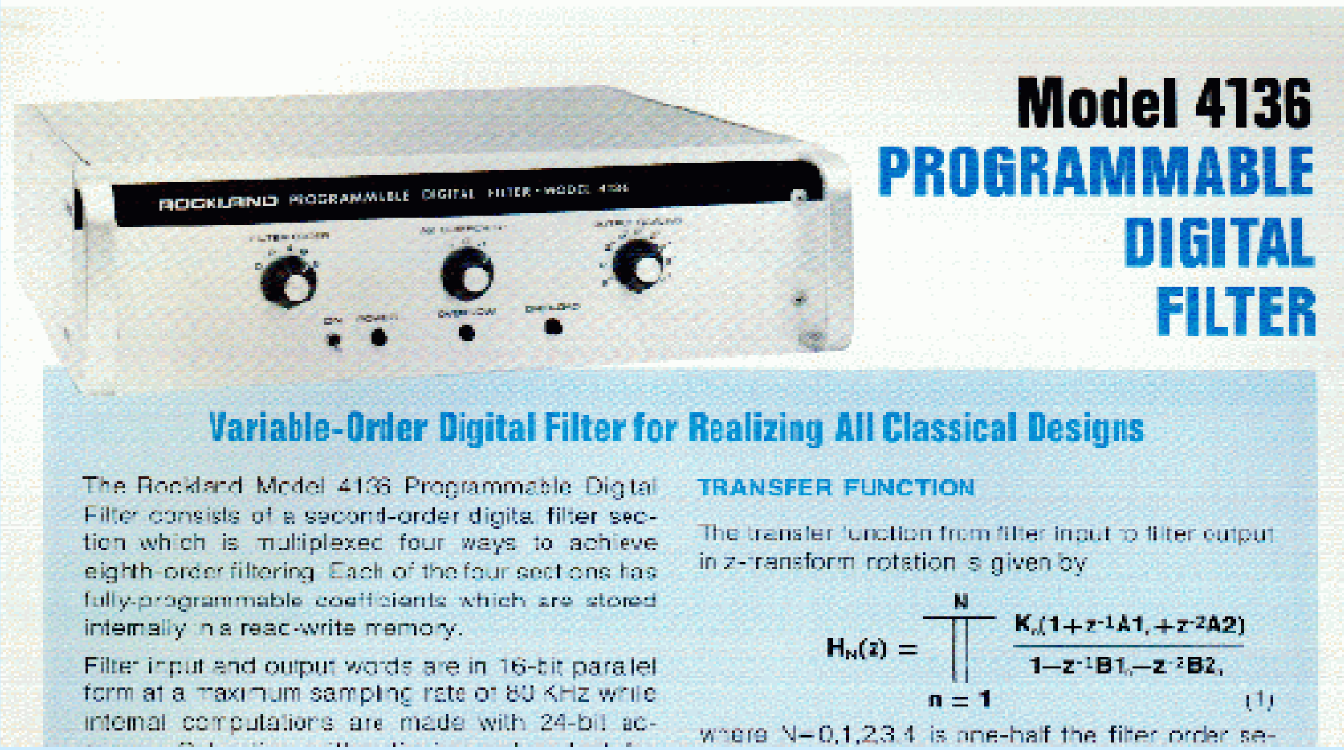

- CONCENTRATE on the COMPUTER
  - PROCESSING ALGORITHMS
  - SOFTWARE (MATLAB)
  - HARDWARE: DSP chips, VLSI
- DSP: DIGITAL SIGNAL PROCESSING

# The TMS32010, 1983




First PC plug-in board from Atlanta Signal Processors Inc.

# Rockland Digital Filter, 1971



Model 4136
PROGRAMMABLE DIGITAL FILTER

ROCKLAND PROGRAMMABLE DIGITAL FILTER • MODEL 4136

## Variable-Order Digital Filter for Realizing All Classical Designs

The Rockland Model 4136 Programmable Digital Filter consists of a second-order digital filter section which is multiplexed four ways to achieve eighth-order filtering. Each of the four sections has fully-programmable coefficients which are stored internally in a read-write memory.

Filter input and output words are in 16-bit parallel form at a maximum sampling rate of 80 KHz while internal computations are made with 24-bit ac-

**TRANSFER FUNCTION**

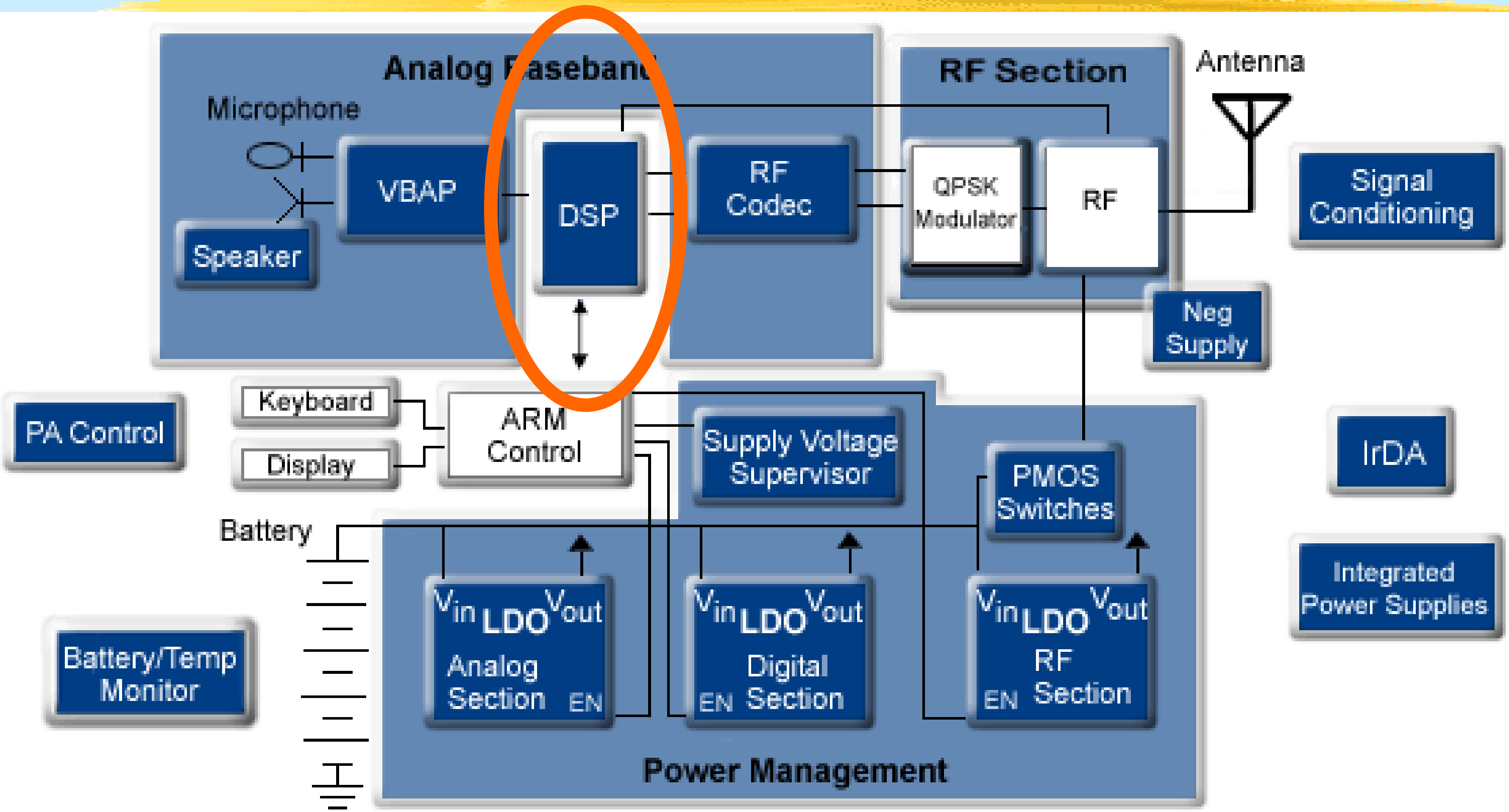
The transfer function from filter input to filter output in z-transform notation is given by

$$H_N(z) = \prod_{n=1}^{N} \frac{K_n(1+z^{-1}A1_n+z^{-2}A2_n)}{1-z^{-1}B1_n-z^{-2}B2_n} \qquad (1)$$

where N=0,1,2,3,4 is one-half the filter order se-

For the price of a small house, you could have one of these.

# Digital Cell Phone (ca. 2000)




Now it plays video

# DISCRETE-TIME SYSTEM

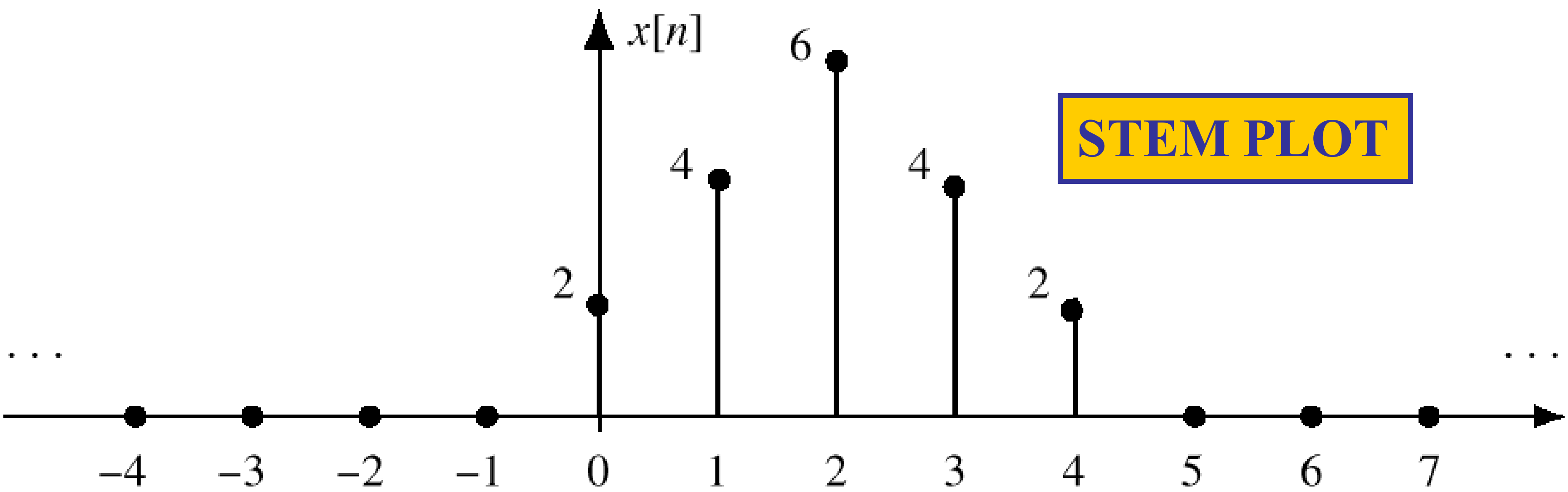

- OPERATE on x[n] to get y[n]
- WANT a **GENERAL** CLASS of SYSTEMS
  - **ANALYZE** the SYSTEM
    - TOOLS: TIME-DOMAIN & FREQUENCY-DOMAIN
  - **SYNTHESIZE** the SYSTEM

# D-T SYSTEM EXAMPLES


- EXAMPLES:
  - POINTWISE OPERATORS
    - SQUARING: $y[n] = (x[n])^2$
  - RUNNING AVERAGE
    - RULE: “the output at time n is the average of three consecutive input values”

# DISCRETE-TIME SIGNAL

- x[n] is a LIST of NUMBERS
  - INDEXED by “$n$”



# 3-PT AVERAGE SYSTEM

- ADD 3 CONSECUTIVE NUMBERS
  - Do this for each "$n$"

the following input–output equation

**Make a TABLE**

$$y[n] = \tfrac{1}{3}(x[n] + x[n+1] + x[n+2])$$

| $n$ | $n < -2$ | $-2$ | $-1$ | $0$ | $1$ | $2$ | $3$ | $4$ | $5$ | $n > 5$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $x[n]$ | 0 | 0 | 0 | 2 | 4 | **6** | **4** | **2** | 0 | 0 |
| $y[n]$ | 0 | $\frac{2}{3}$ | 2 | 4 | $\frac{14}{3}$ | **4** | 2 | $\frac{2}{3}$ | 0 | 0 |

$\boxed{n=0}$ $y[0] = \frac{1}{3}(x[0] + x[1] + x[2])$

$\boxed{n=1}$ $y[1] = \frac{1}{3}(x[1] + x[2] + x[3])$

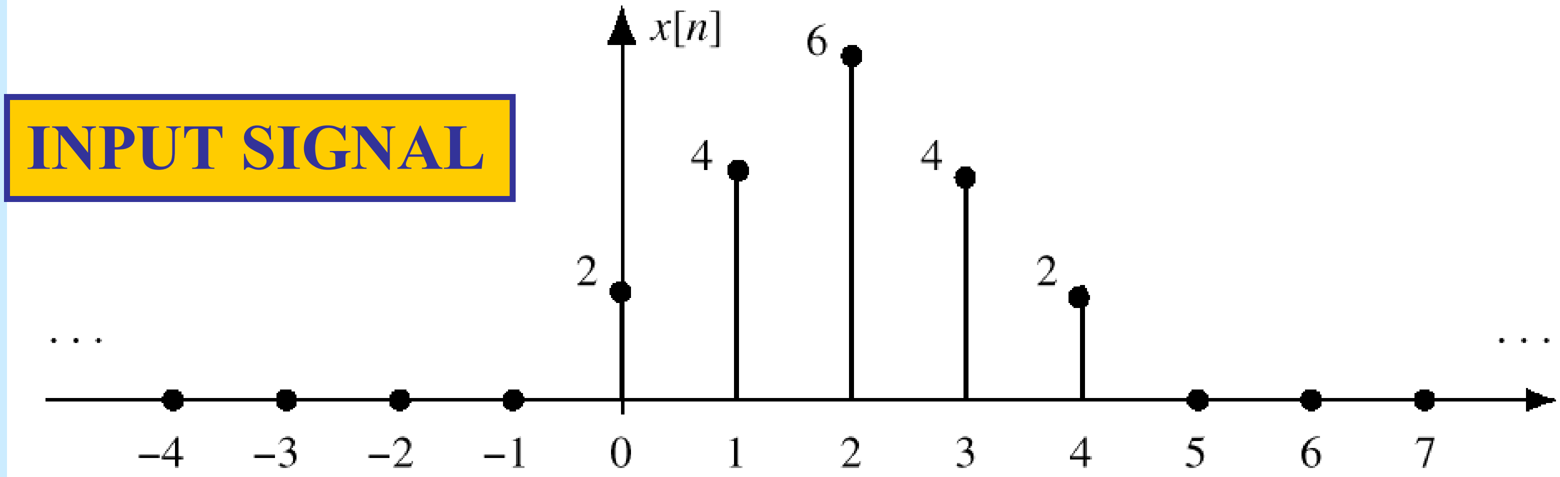



**Figure 5.2** Finite-length input signal, $x[n]$.

$$y[n]=\tfrac{1}{3}(x[n]+x[n+1]+x[n+2])$$

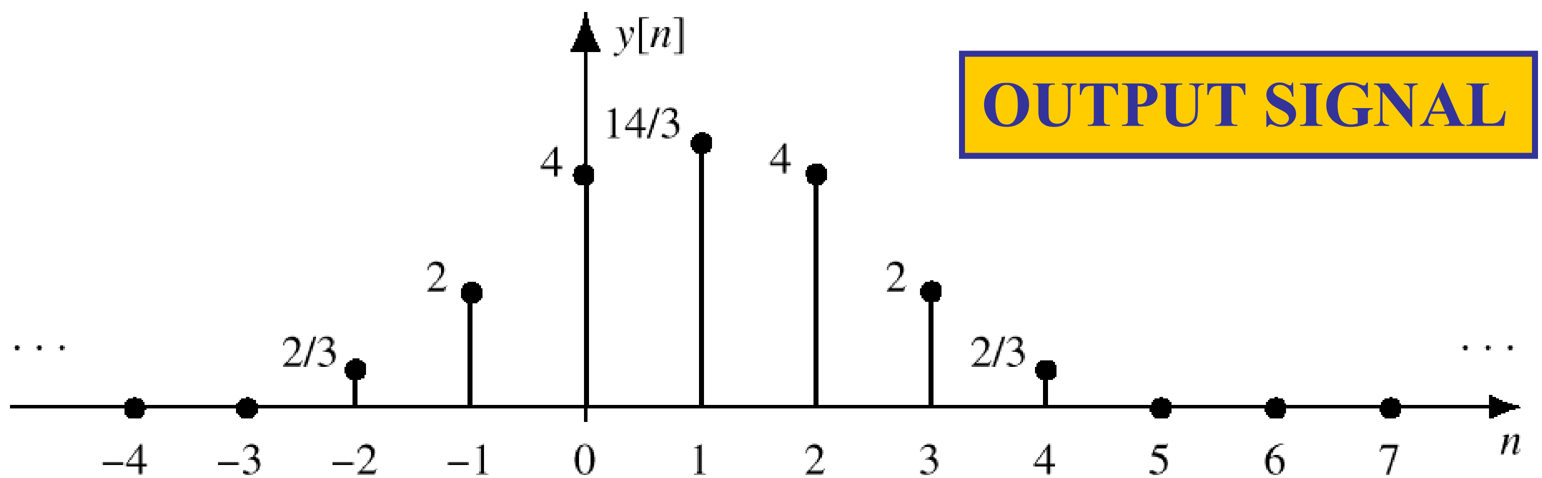



**Figure 5.3** Output of running average, $y[n]$.

# PAST, PRESENT, FUTURE



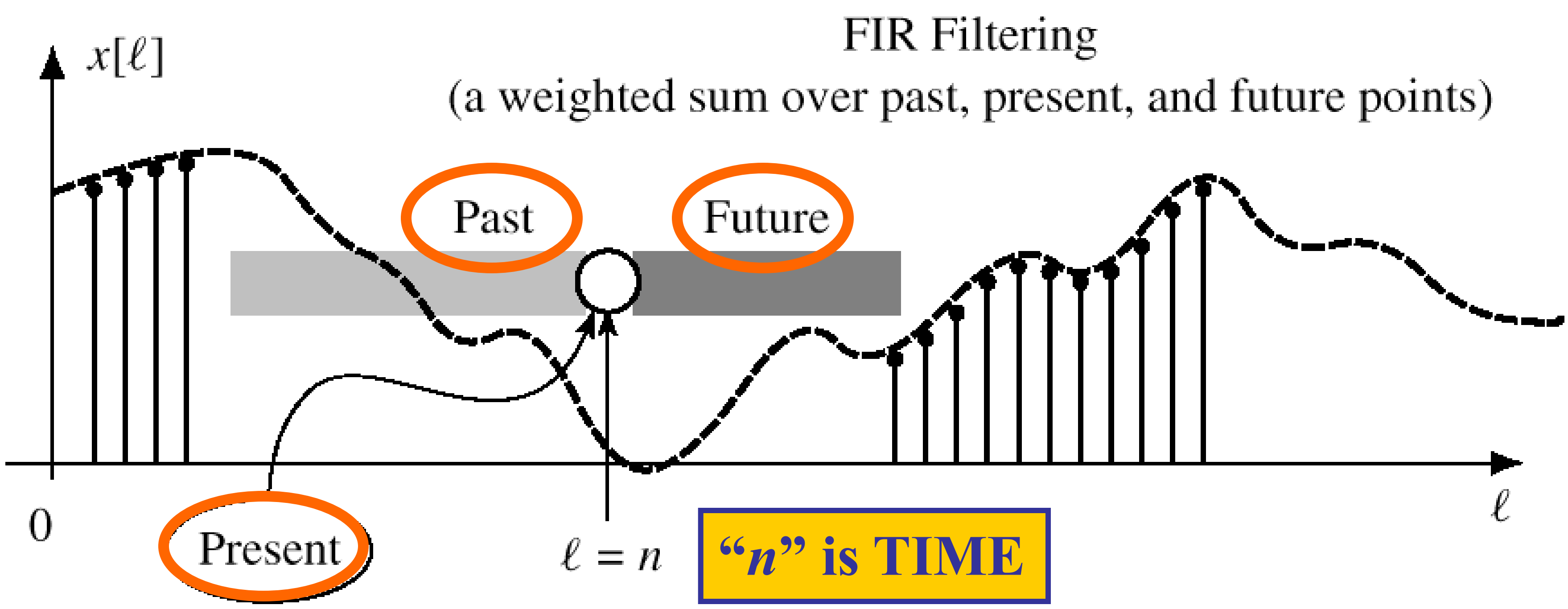



**Figure 5.4** The running-average filter calculation at time index $n$ uses values within a sliding window (shaded). Dark shading indicates the future ($\ell > n$); light shading, the past ($\ell < n$).

# ANOTHER 3-pt AVERAGER

- Uses “PAST” VALUES of x[n]
  - IMPORTANT IF “n” represents REAL TIME
    - WHEN x[n] & y[n] ARE STREAMS

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$$

| $n$ | $n < -2$ | $-2$ | $-1$ | $0$ | $1$ | $2$ | $3$ | $4$ | $5$ | $6$ | $7$ | $n > 7$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x[n]$ | 0 | 0 | 0 | 2 | 4 | **6** | **4** | **2** | 0 | 0 | 0 | 0 |
| $y[n]$ | 0 | 0 | 0 | $\frac{2}{3}$ | 2 | 4 | $\frac{14}{3}$ | **4** | 2 | $\frac{2}{3}$ | 0 | 0 |

# GENERAL FIR FILTER

- FILTER COEFFICIENTS $\{b_k\}$
  - DEFINE THE FILTER

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

  - For example, $\boxed{b_k = \{3, -1, 2, 1\}}$

$$\begin{aligned} y[n] &= \sum_{k=0}^{3} b_k x[n-k] \\ &= 3x[n] - x[n-1] + 2x[n-2] + x[n-3] \end{aligned}$$

# GENERAL FIR FILTER

- FILTER COEFFICIENTS $\{b_k\}$

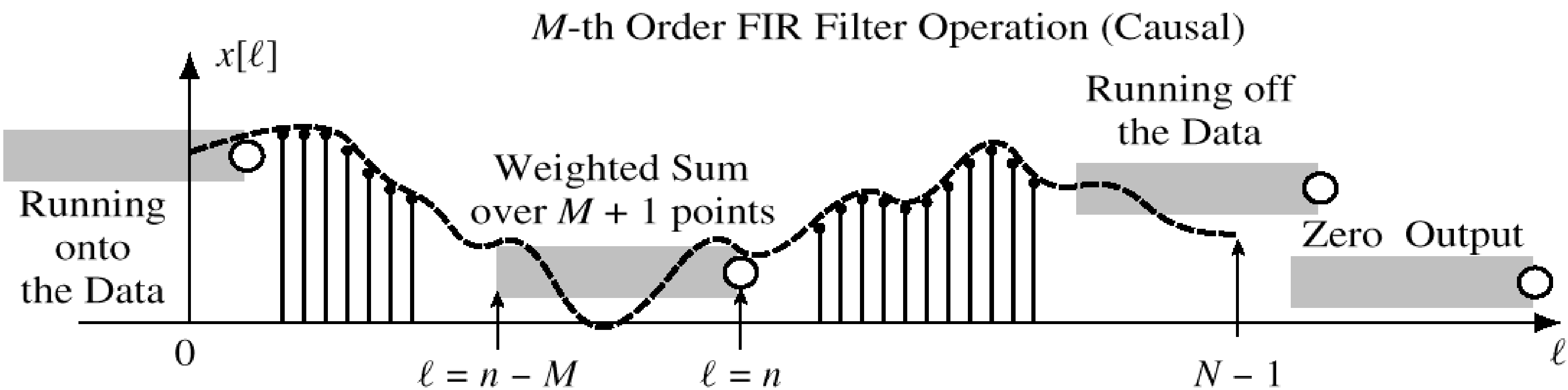
$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

- FILTER **ORDER** is M
- FILTER **LENGTH** is L = M+1
  - NUMBER of FILTER COEFFS is L

# GENERAL FIR FILTER

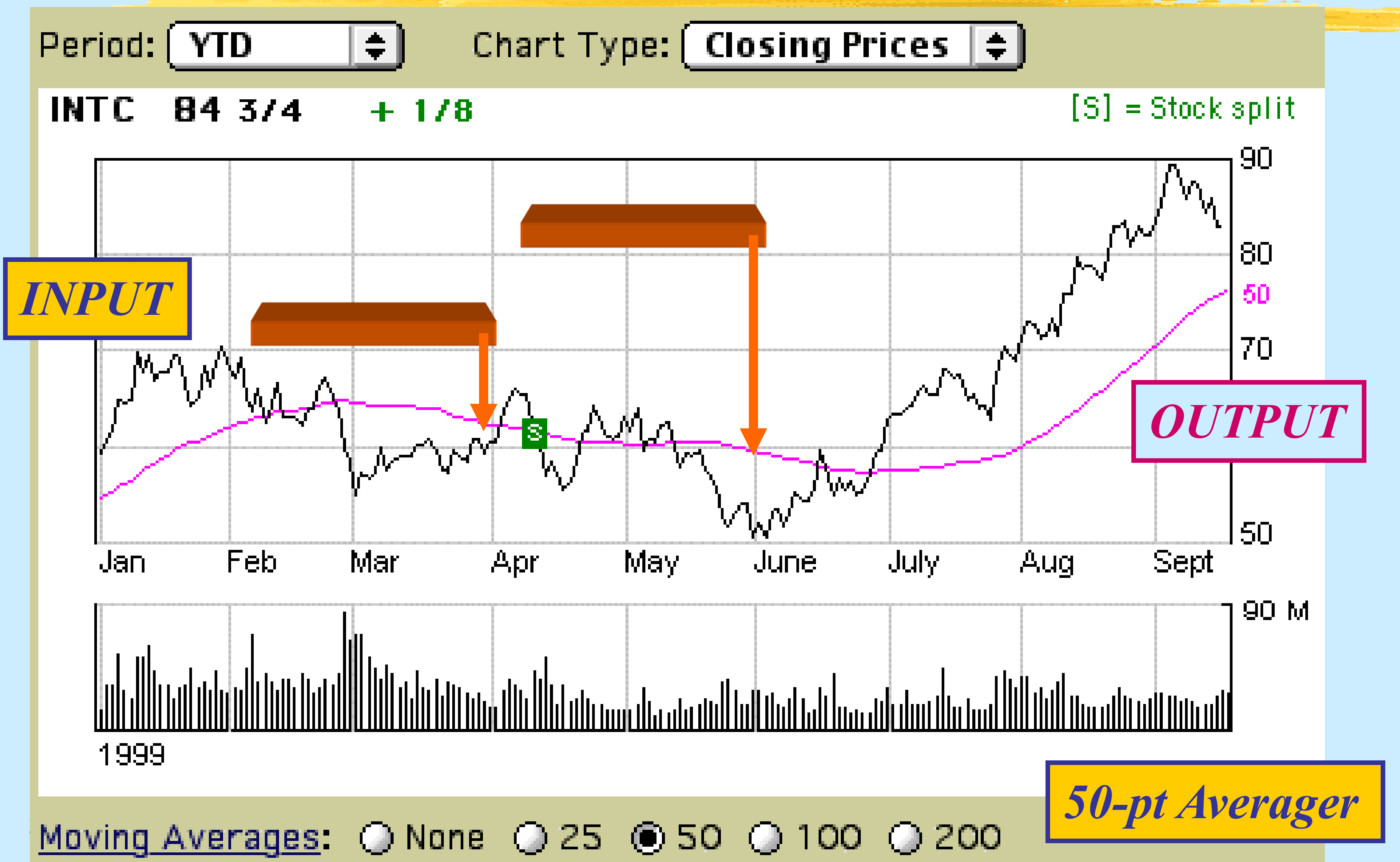
- SLIDE a WINDOW across x[n]

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$




$x[n\text{-}M]$ $x[n]$

# FILTERED STOCK SIGNAL



# SPECIAL INPUT SIGNALS

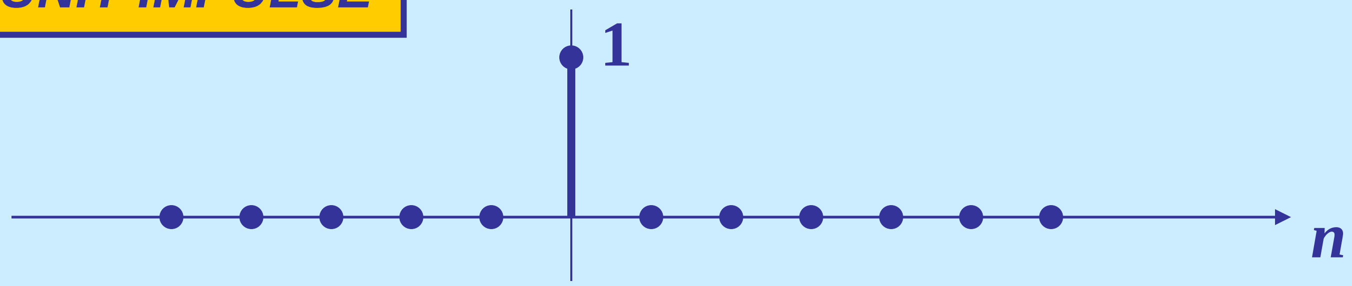
- x[n] = SINUSOID

**FREQUENCY RESPONSE (LATER)**

- x[n] has only one NON-ZERO VALUE

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

**UNIT-IMPULSE**



# UNIT IMPULSE SIGNAL δ[n]

| $n$ | ... | $-2$ | $-1$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\delta[n]$ | 0 | 0 | 0 | $\boxed{1}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\delta[n-3]$ | 0 | 0 | 0 | 0 | 0 | 0 | $\boxed{1}$ | 0 | 0 | 0 | 0 |

$n = 3$

**δ[n] is NON-ZERO When its argument is equal to ZERO**

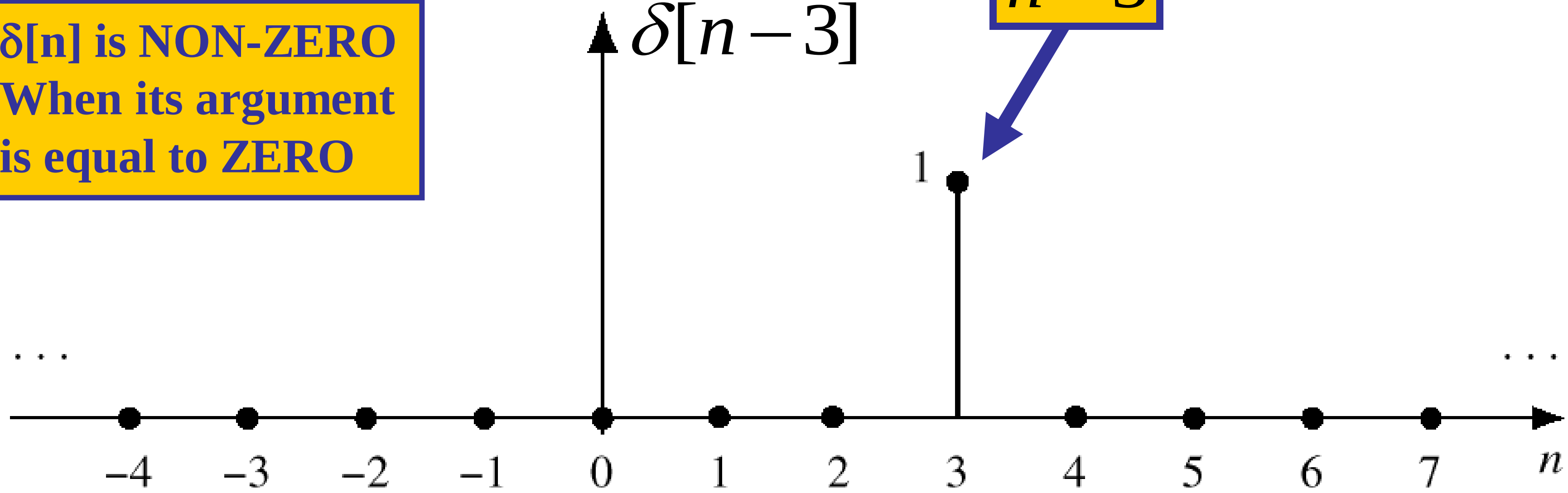
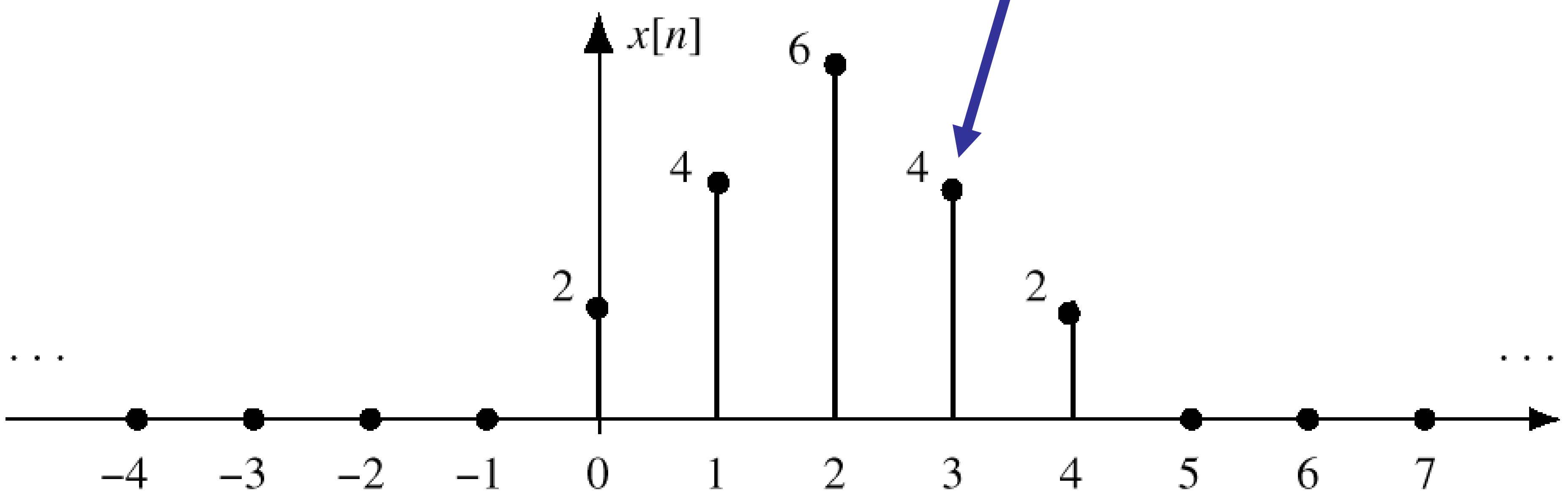



**Figure 5.7** Shifted impulse sequence, $\delta[n-3]$.

# MATH FORMULA for x[n]

- Use SHIFTED IMPULSES to write x[n]

$$x[n] = 2\delta[n] + 4\delta[n-1] + 6\delta[n-2] + 4\delta[n-3] + 2\delta[n-4]$$



# SUM of SHIFTED IMPULSES

| $n$ | $\ldots$ | $-2$ | $-1$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | $\ldots$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $2\delta[n]$ | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $4\delta[n-1]$ | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| $6\delta[n-2]$ | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 |
| $4\delta[n-3]$ | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 |
| $2\delta[n-4]$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| $x[n]$ | 0 | 0 | 0 | 2 | 4 | 6 | 4 | 2 | 0 | 0 | 0 |

$$x[n] = \sum_k x[k]\delta[n-k]$$

**This formula ALWAYS works**

$$= \ldots + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + \ldots \qquad (5.3.6)$$

# 4-pt AVERAGER

- CAUSAL SYSTEM: USE PAST VALUES

$$y[n] = \tfrac{1}{4}(x[n] + x[n-1] + x[n-2] + x[n-3])$$

- INPUT = UNIT IMPULSE SIGNAL = **δ[n]**

$$x[n] = \delta[n]$$

$$y[n] = \tfrac{1}{4}\delta[n] + \tfrac{1}{4}\delta[n-1] + \tfrac{1}{4}\delta[n-2] + \tfrac{1}{4}\delta[n-3]$$

- OUTPUT is called "IMPULSE RESPONSE"

$$h[n] = \{\ldots, 0, 0, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, 0, 0, \ldots\}$$

# 4-pt Avg Impulse Response

$$y[n] = \tfrac{1}{4}(x[n] + x[n-1] + x[n-2] + x[n-3])$$

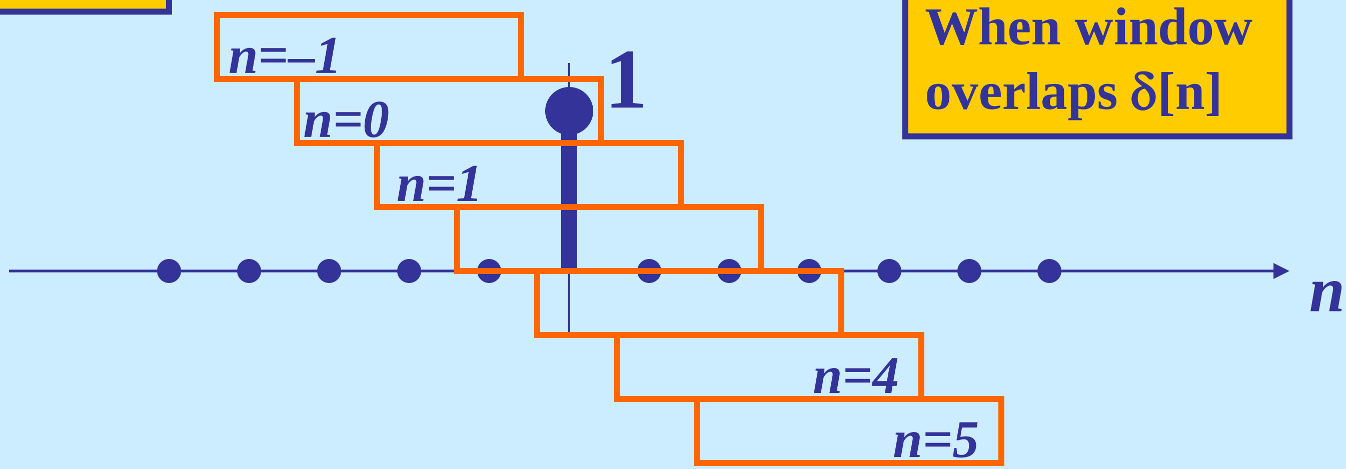
δ[n] "READS OUT" the FILTER COEFFICIENTS

$$h[n] = \{\ldots, 0, 0, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, 0, 0, \ldots\}$$

*n=0*

**"h" in *h*[n] denotes Impulse Response**

**NON-ZERO When window overlaps δ[n]**



# FIR IMPULSE RESPONSE

- Convolution = Filter Definition
  - Filter Coeffs = Impulse Response

| $n$ | $n<0$ | 0 | 1 | 2 | 3 | $\ldots$ | $M$ | $M+1$ | $n>M+1$ |
|---|---|---|---|---|---|---|---|---|---|
| $x[n]=\delta[n]$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $y[n]=h[n]$ | 0 | $b_0$ | $b_1$ | $b_2$ | $b_3$ | $\ldots$ | $b_M$ | 0 | 0 |

$$y[n]=\sum_{k=0}^{M} b_k x[n-k]$$

$$y[n]=\sum_{k=0}^{M} h[k]x[n-k]$$

**CONVOLUTION**

# FILTERING EXAMPLE

- 7-point AVERAGER
  - Smooth compared to 3-point Averager
    - By making its amplitude (A) smaller

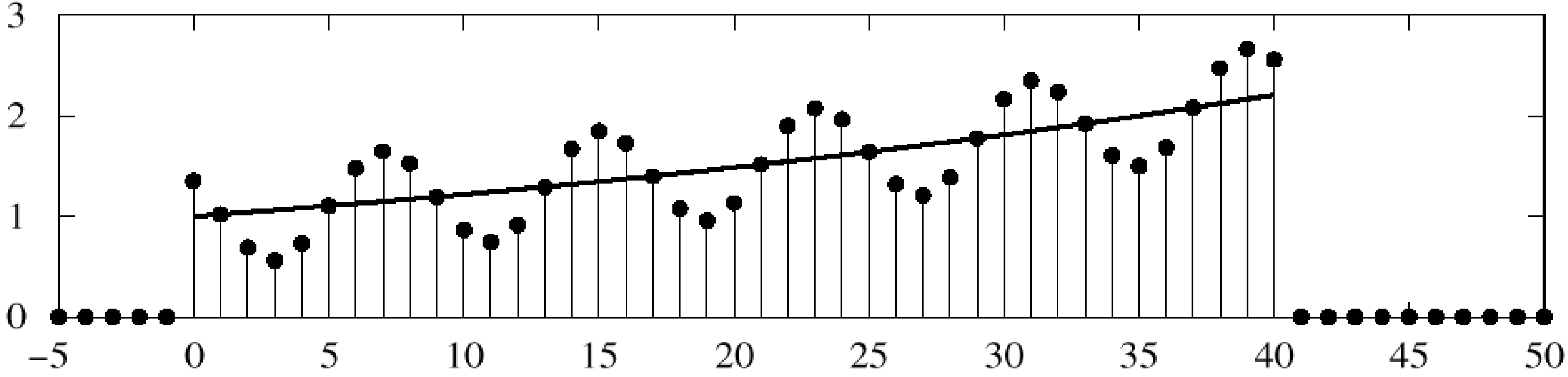
$$y_7[n] = \sum_{k=0}^{6} \left(\tfrac{1}{7}\right) x[n-k]$$

- 3-point AVERAGER
  - Changes A slightly

$$y_3[n] = \sum_{k=0}^{2} \left(\tfrac{1}{3}\right) x[n-k]$$

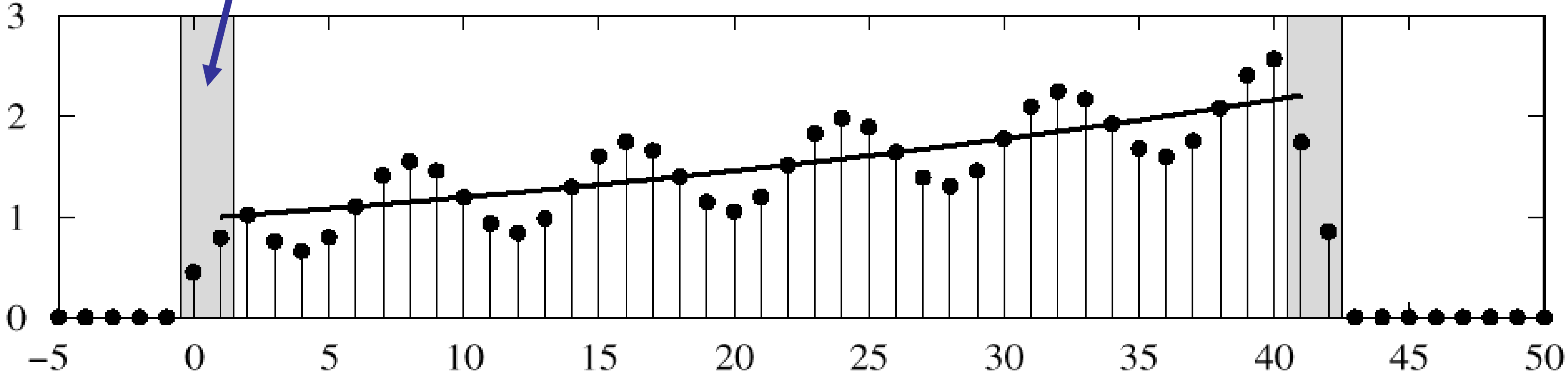
# 3-pt AVG EXAMPLE

$$\text{Input: } x[n] = (1.02)^n + \cos(2\pi n/8 + \pi/4) \quad \text{for } 0 \le n \le 40$$




USE PAST VALUES

Output of 3-Point Running-Average Filter



# 7-pt FIR EXAMPLE (AVG)

Input: $x[n] = (1.02)^n + \cos(2\pi n/8 + \pi/4) \quad \text{for } 0 \le n \le 40$