# SNS COLLEGE OF TECHNOLOGY

**An Autonomous Institution**

**Coimbatore-35**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

## DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

## 19ECB212 – DIGITAL SIGNAL PROCESSING

II YEAR/ IV SEMESTER

## UNIT 4 – FINITE WORD LENGTH EFFECTS

TOPIC – ROUNDOFF NOISE POWER, LIMIT CYCLES IN RECURSIVE SYSTEMS & OVERFLOW LIMIT CYCLE

# PRODUCT QUANTIZATION ERROR

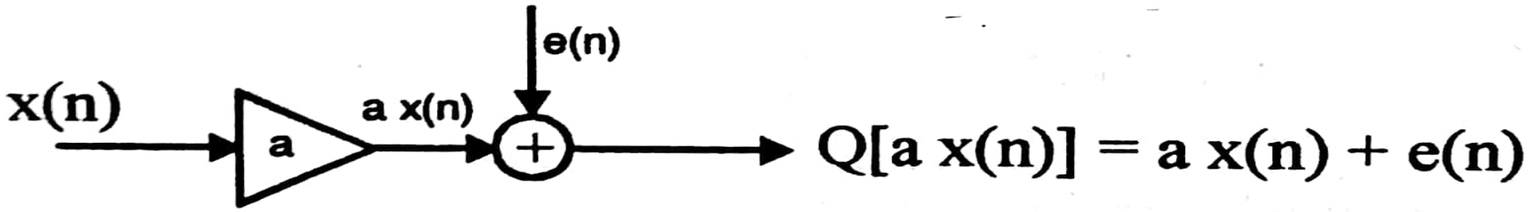
- In realization structures of IIR Systems, multipliers are used to multiply the signal by constants. The output of the multipliers i.e., the product are quantized to finite word length in order to store them in registers and to be used in subsequent calculations
- In fixed point arithmetic, the multiplication of two b-bit numbers results in a product of length 2b-bits. If the word length of the register used to store the result is b-bits then it is necessary to quantize the product (result) to b-bits. The error due to quantization of the output of multiplier is referred to as **Product Quantization Error**

# PRODUCT QUANTIZATION ERROR

- In digital system the product quantization is performed by rounding due to the following desirable characteristics of rounding
- In rounding the error signal is independent of the type of arithmetic employed
- The mean value of error signal due to rounding is zero
- The variance of error signal due to rounding is the least
- The analysis of product quantization error is similar to the analysis of quantization error due to A/D process
- But in product quantization error analysis it is necessary to define the noise transfer function, which depends on the structure of the digital network

# PRODUCT QUANTIZATION ERROR

- The Noise Transfer Function (NTF) is defined as the transfer function from the noise source to the filter output (i.e., NTF is the transfer function obtained by treating the noise source as actual input)
- The model of the multiplier of a digital network using fixed point arithmetic as shown. The multiplier is considered as an infinite precision multiplier. Using an adder the error signal is added to the output of the multiplier so that the output of adder is equal to the quantized product
- Therefore the output of finite word length multiplier can be expressed as



# PRODUCT QUANTIZATION ERROR

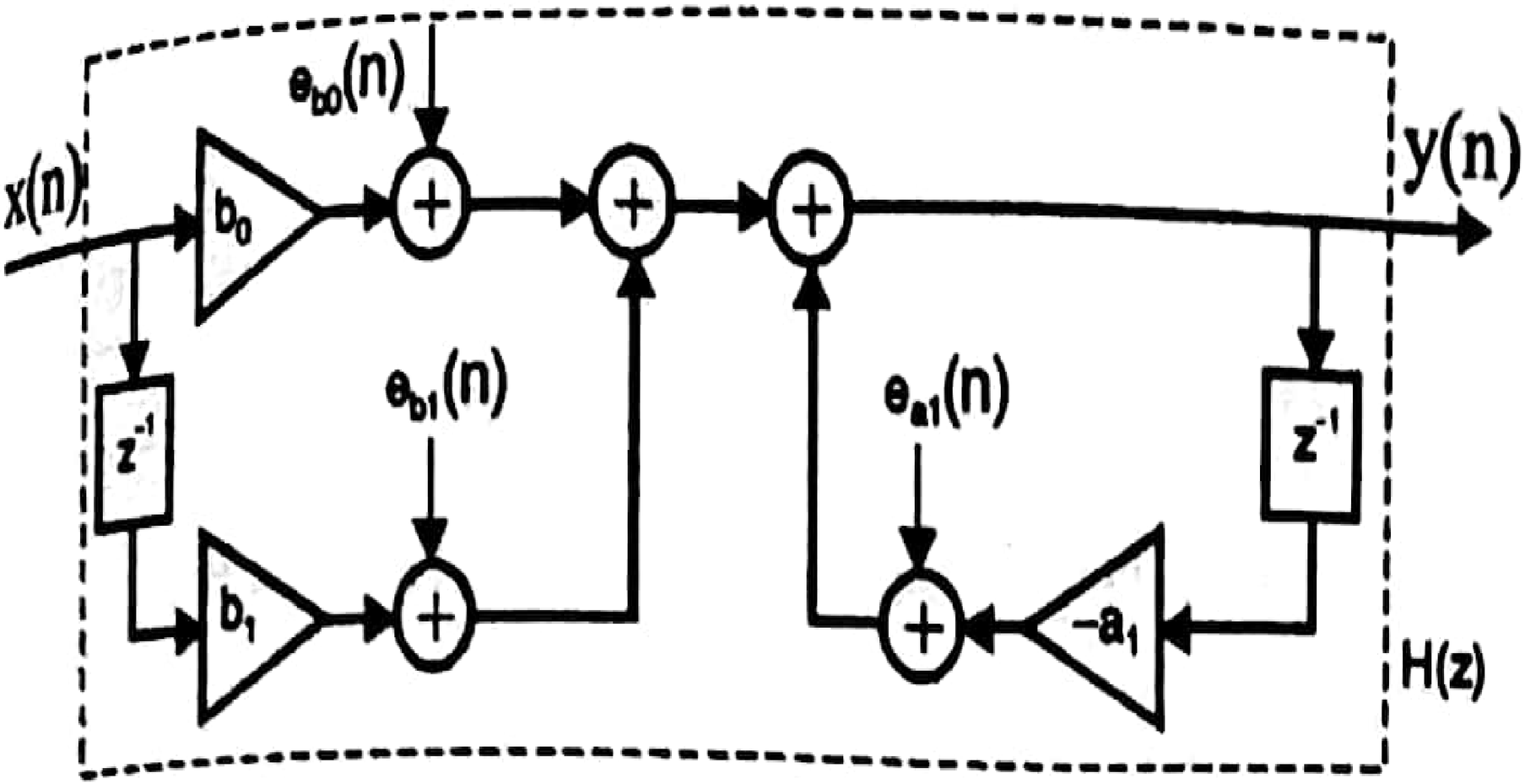
**Quantized Product = Q[a x(n)] = a x(n) + e(n)**

- Where a x(n) = Unquantized Product
- e(n) = Product quantization error signal
- The product quantization error signal is treated as a random process with uniform probability density function. The following assumptions are made regarding the statistical independence of the various noise sources in the digital filter
- Any two different samples from the same noise source are considered
- Any two different noise sources, When considered as random processes are uncorrelated
- Each noise source is uncorrelated with the input sequence

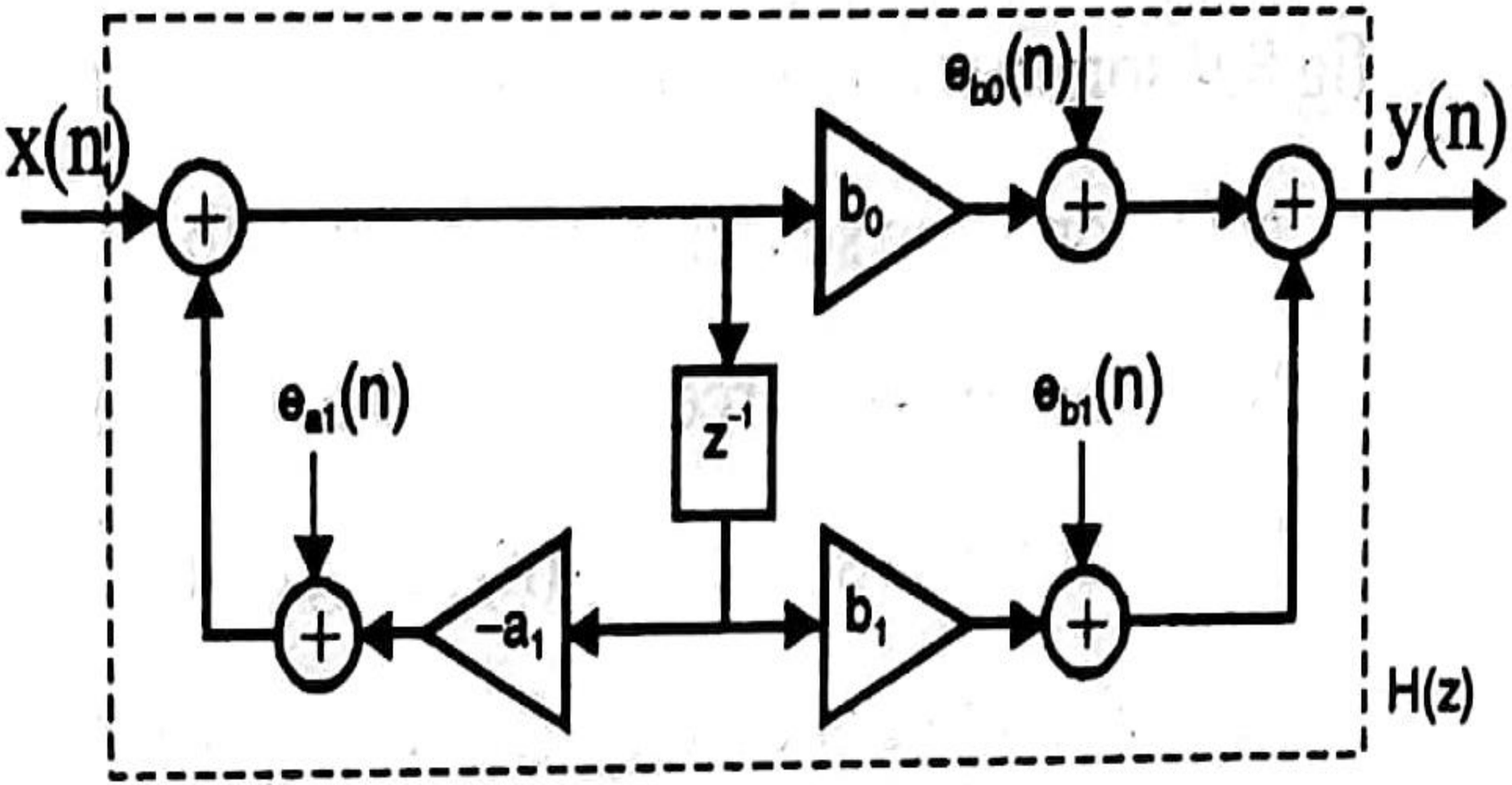
# PRODUCT QUANTIZATION ERROR

- The product quantization noise models for first-order and second-order IIR systems using direct form-I and direct form-II structures. In these models each finite precision multiplier is replaced by an ideal multiplier and an additive roundoff noise.
- The equations used for computing the steady state output noise variance (Power) due to quantization error in A/D conversion process can be used to compute the output noise variance due to product quantization, because in both cases the quantization is performed by rounding
- But the transfer function by each noise source is different. Therefore for each noise source, The Noise Transfer Function (NTF) has o be determined by treating the noise source as input (and the output of the system)

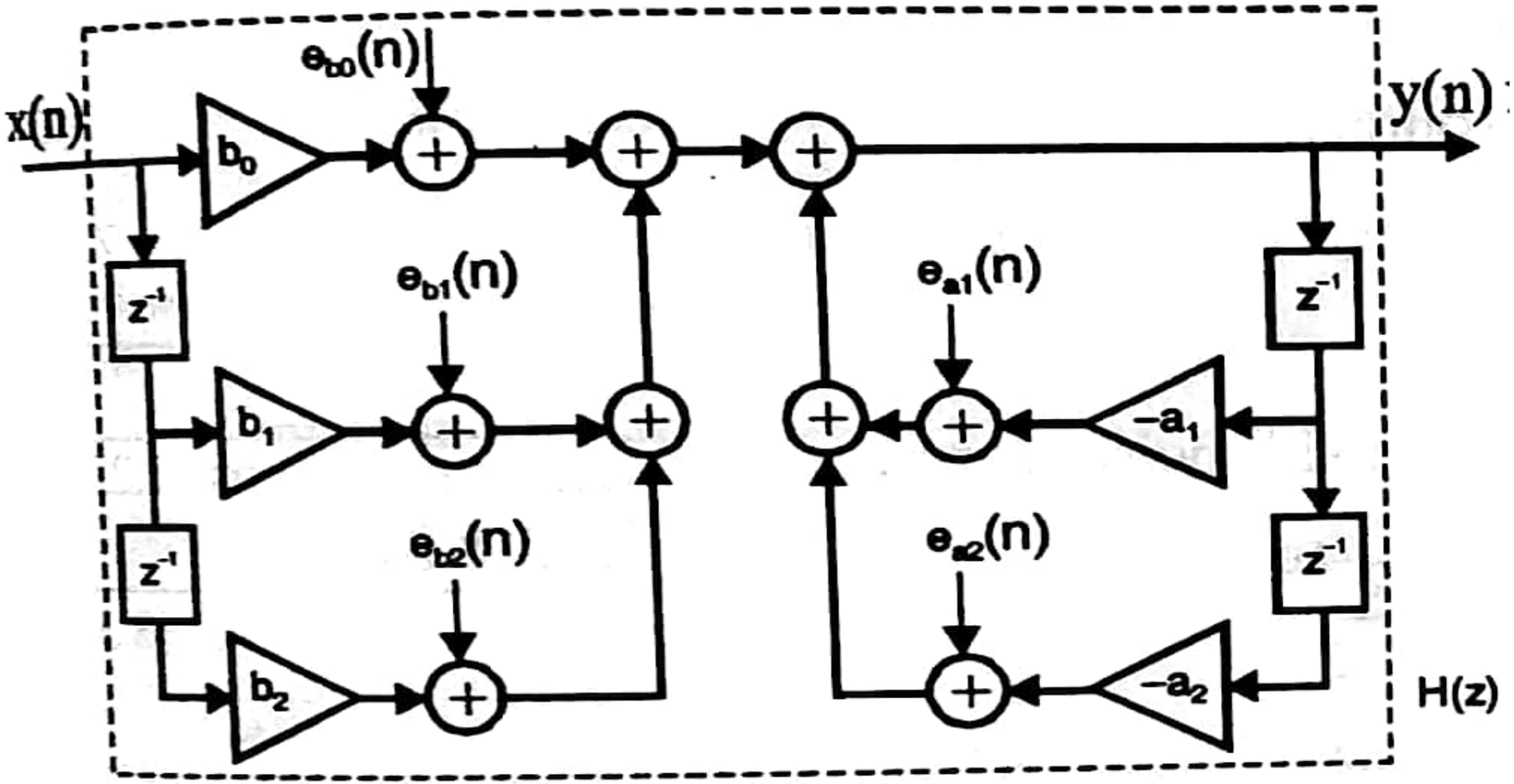
# FIRST ORDER DIRECT FORM -I



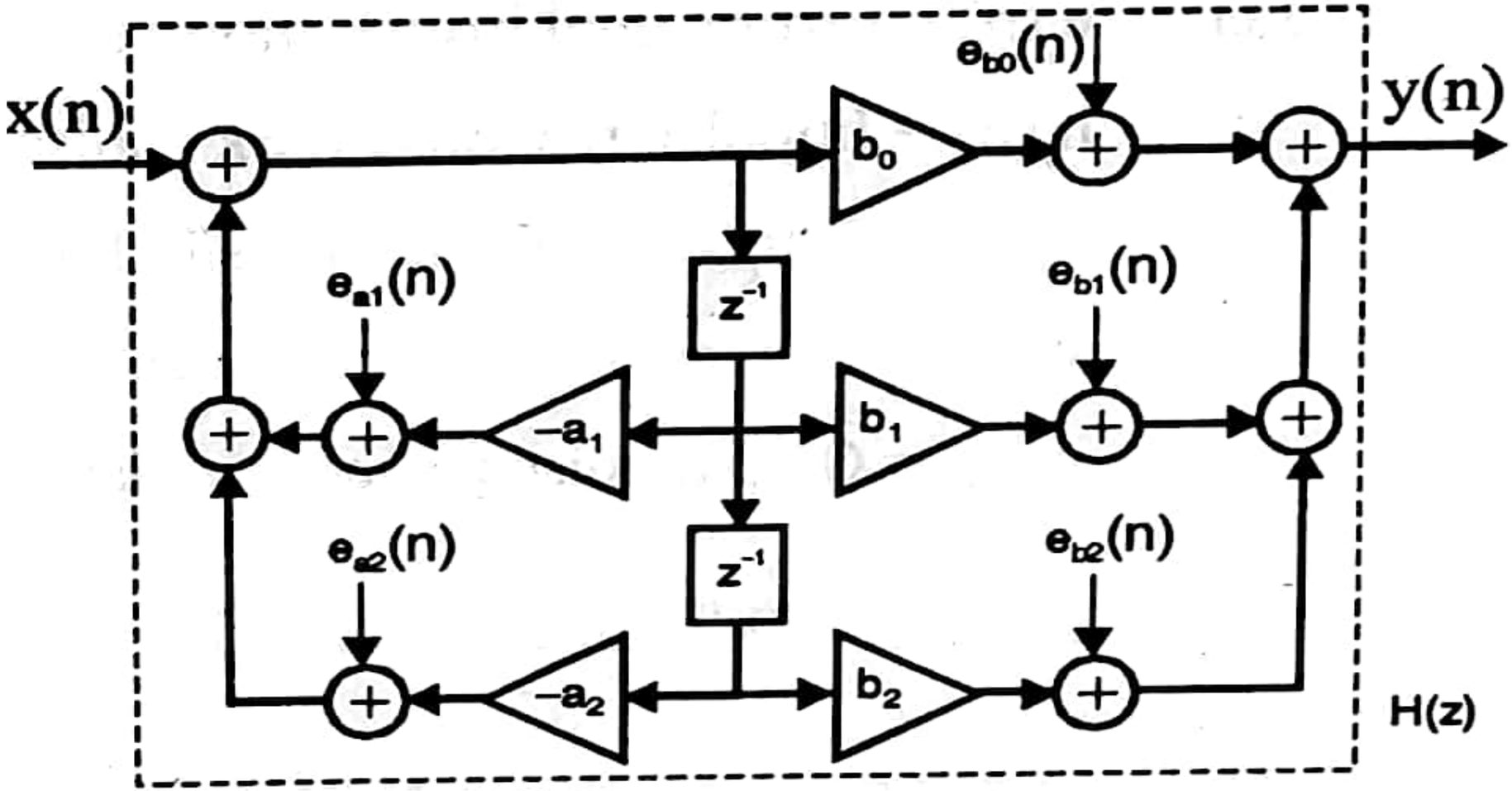
# FIRST ORDER DIRECT FORM -II



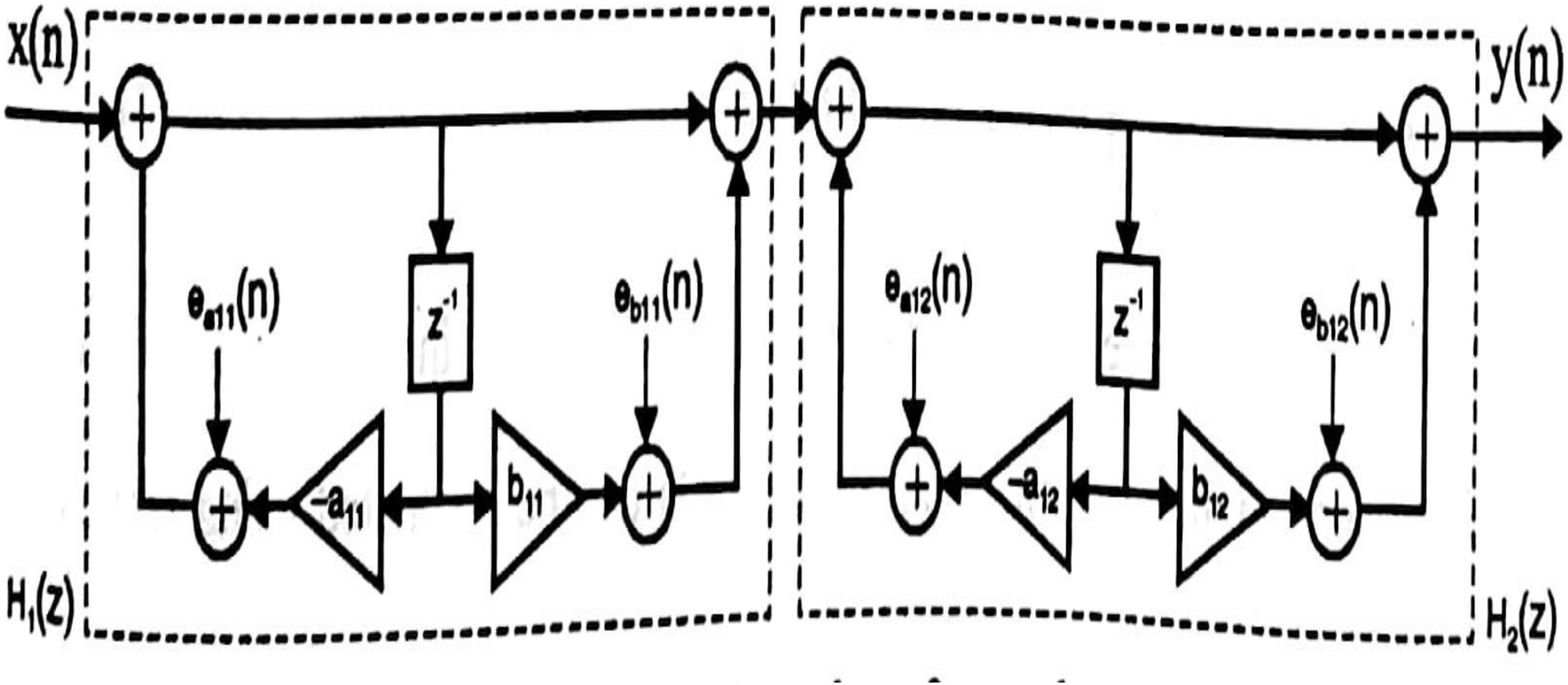
# SECOND ORDER DIRECT FORM -I



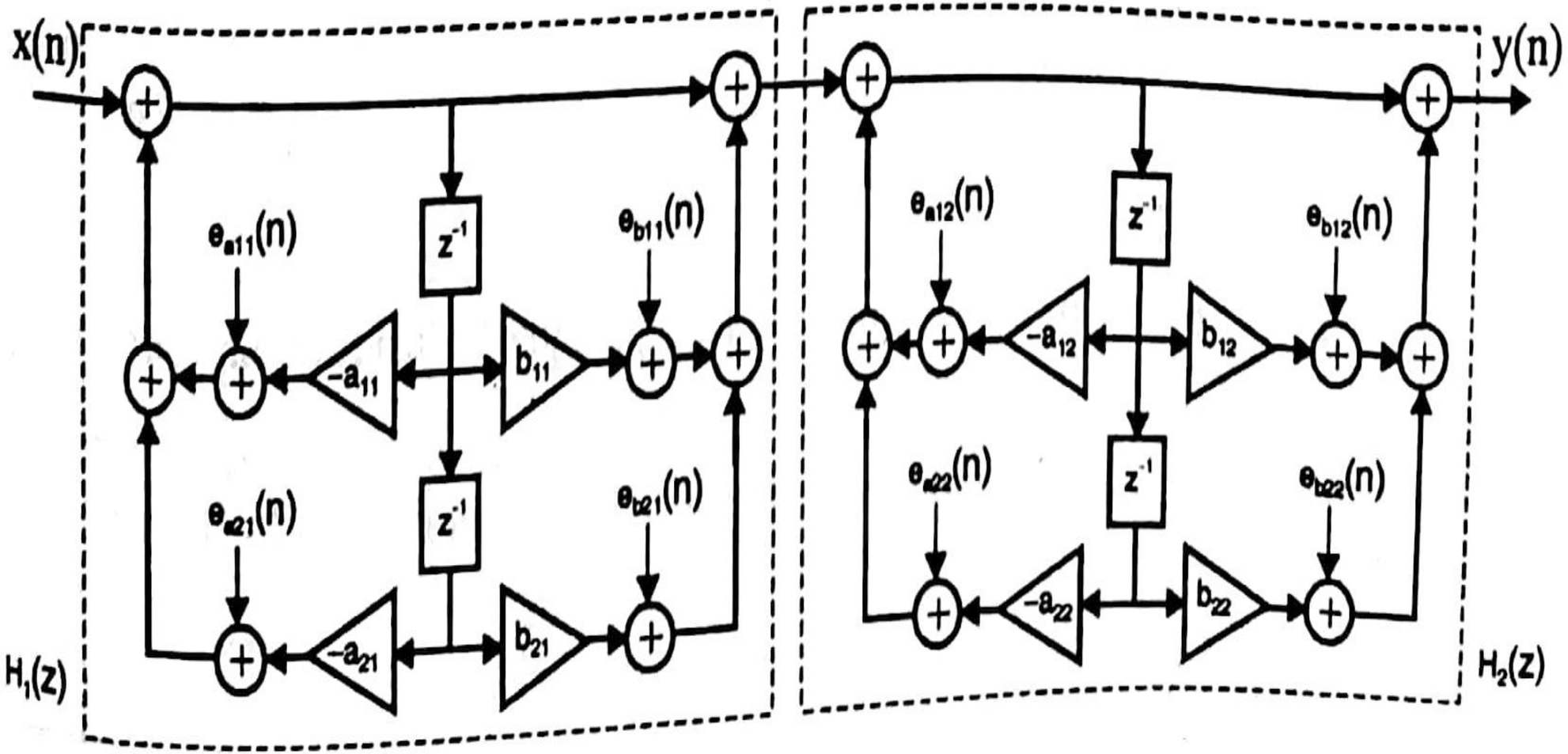
# SECOND ORDER DIRECT FORM -II



# CASCADING OF TWO FIRST – ORDER SECTIONS



# CASCADING OF TWO FIRST – ORDER SECTIONS



# OUTPUT NOISE POWER (ROUNDOFF NOISE POWER) DUE TO PRODUCT QUANTIZATION

Let, $e_k(n)$ = Error signal from $k^{th}$ noise source.

$h_k(n)$ = Impulse response for $k^{th}$ noise source.

$T_k(z) = \mathcal{Z}\{h_k(n)\}$ = Noise Transfer Function (NTF) for $k^{th}$ noise source.

$\sigma^2_{ek}$ = Variance of $k^{th}$ noise source.

$\sigma^2_{ekop}$ = Output noise power or variance due to $k^{th}$ noise source.

Now, Variance of $k^{th}$ noise source, $\sigma^2_{ek} = \frac{q^2}{12}$ or $\frac{2^{-2B}}{3}$

Now, Output noise power due to $k^{th}$ noise source, $\sigma^2_{ekop} = \sigma^2_{ek} \sum_{n=0}^{\infty} h_k^2(n)$

In equation the summation of $h_k(n)$ can be evaluated using Parseval's theorem.

$$\therefore \sigma^2_{ekop} = \sigma^2_{ek} \frac{1}{2\pi j} \oint_c T_k(z)\, T_k(z^{-1})\, z^{-1}\, dz$$

# OUTPUT NOISE POWER (ROUNDOFF NOISE POWER) DUE TO PRODUCT QUANTIZATION

where, $\oint_c$ denote integration around unit circle $|z| = 1$, in the anticlockwise direction.

The closed contour integration of equation can be evaluated using residue theorem of Z-transform as shown below.

$$\therefore \sigma^2_{ekop} = \sigma^2_{ek} \sum_{i=1}^{N} \text{Res}\left[T_k(z)\, T_k(z^{-1})\, z^{-1}\right]\Big|_{z=p_i}$$

$$= \sigma^2_{ek} \sum_{i=1}^{N} \left[(z-p_i)\, T_k(z)\, T_k(z^{-1})\, z^{-1}\right]\Big|_{z=p_i}$$

where $p_1, p_2, \ldots, p_N$ are poles of $T_k(z)\, T_k(z^{-1})\, z^{-1}$, that lie inside the unit circle in z-plane.

Let the number of noise sources in a digital system (or filter) be M. The total steady state noise variance at the output of the system due to product quantization errors is given by the sum of the output noise variances due to all the noise sources.

Let, $\sigma^2_{eTop}$ = Total output noise power due to product quantization error (or Total roundoff noise power)

$$\therefore \sigma^2_{eTop} = \sigma^2_{e1op} + \sigma^2_{e2op} + \ldots + \sigma^2_{eMop}$$

# LIMIT CYCLES IN RECURSIVE SYSTEMS

- **Zero Input Limit Cycles:** In recursive systems, when the input is zero or some nonzero constant value, the nonlinearities due to finite precision arithmetic operations may cause periodic oscillations in the output
- During periodic oscillations, the output y(n) of a system will oscillate between a finite positive and negative value for increasing n or the output will become constant for increasing n. Such oscillations are called **limit cycles.** These oscillations are due to round-off errors in multiplication and overflow in addition
- In recursive systems, if the system output enters a limit cycle, it will continue to remain in limit cycle even when the input is made zero. Hence these limit cycles are also called **Zero Input Limit Cycles**

# LIMIT CYCLES IN RECURSIVE SYSTEMS

- The system output remains in limit cycle until another input of sufficient magnitude is applied to drive the system out of limit cycle
- Consider the difference equation of first order system with only pole as shown

$$y(n) = a\,y(n-1) + x(n)$$

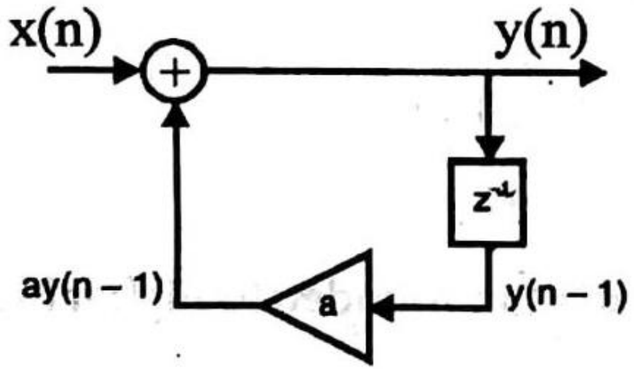
- The system has one product [a y(n-1)]. If the product is quantized to finite word length then the response y(n) will deviate from actual value. Let y'(n) be the response of the system when the product is quantized in each recursive realization

$$y'(n) = Q[a\,y'(n-1)] + x(n)$$

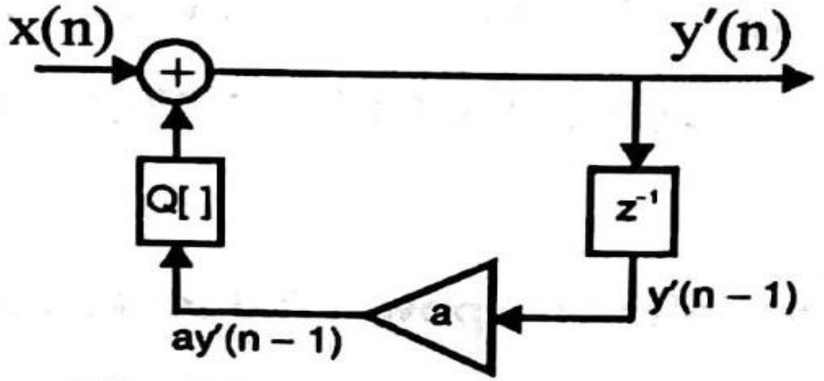
- Where, Q [ ] stands for quantization operation
- Q[a y'(n-1)] = Quantized value of the product a y'(n-1)

# LIMIT CYCLES IN RECURSIVE SYSTEMS

- In the first order system with only pole, the coefficient “a” will be the pole of the system. Let us examine the nature of response of first-order system for an impulse input and various values of poles
- Choose Sign-magnitude representation for binary product and response. Let the product be quantized to four bit binary (excluding sign bit) by upward rounding.




**Ideal System**




**Nonlinear system due to product quantization**

# LIMIT CYCLES IN RECURSIVE SYSTEMS

Let, $y'(n) = 0$ ; for $n < 0$

$$x(n) = \frac{15}{16} \quad ; \text{ for } n = 0$$
$$= 0 \quad ; \text{ for } n \neq 0$$

$$a = \frac{1}{2}$$

$$\text{When } n = 0,\ y'(n) = y'(0) = Q[a\,y'(n-1)] + x(n) = Q[a\,y'(-1)] + x(0)$$

$$= Q\left[\frac{1}{2} \times 0\right] + \frac{15}{16} = Q[0] + \frac{15}{16} = 0 + \frac{15}{16}$$

$$= 0.9375_{10} = +\frac{15}{16}$$

$+.9375_{10} \xrightarrow[\text{binary}]{\text{Convert to}} +.11110_2 \xrightarrow[\text{bit}]{\text{Add sign}} 0.11110_2 \xrightarrow[\text{to 4-bits}]{\text{Round magnitude}} 0.1111_2$

$0.1111_2 \xrightarrow[\text{sign bit}]{\text{Extract}} +.1111_2 \xrightarrow[\text{decimal}]{\text{Convert to}} +.9375_{10} = +\frac{15}{16}$

# LIMIT CYCLES IN RECURSIVE SYSTEMS

When $n = 1$, $y'(n) = y'(1) = Q[a\,y'(n-1)] + x(n) = Q[a\,y'(0)] + x(1)$

$$= Q\left[\frac{1}{2} \times \frac{15}{16}\right] + 0 = Q[0.46875] = 0.5_{10} = +\frac{8}{16}$$

$+.46875_{10} \xrightarrow[\text{binary}]{\text{Convert to}} +.01111_2 \xrightarrow[\text{bit}]{\text{Add sign}} 0.01111_2 \xrightarrow[\text{to 4-bits}]{\text{Round magnitude}} 0.1000_2$

$0.1000_2 \xrightarrow[\text{sign bit}]{\text{Extract}} +.1000_2 \xrightarrow[\text{decimal}]{\text{Convert to}} +.5_{10} = +\frac{1}{2} = +\frac{8}{16}$

When $n = 2$, $y'(n) = y'(2) = Q[a\,y'(n-1)] + x(n) = Q[a\,y'(1)] + x(2)$

$$= Q\left[\frac{1}{2} \times \frac{8}{16}\right] + 0 = Q[0.25] = 0.25_{10} = +\frac{4}{16}$$

$+.25_{10} \xrightarrow[\text{binary}]{\text{Convert to}} +.01000_2 \xrightarrow[\text{bit}]{\text{Add sign}} 0.01000_2 \xrightarrow[\text{to 4-bits}]{\text{Round magnitude}} 0.0100_2$

$0.0100_2 \xrightarrow[\text{sign bit}]{\text{Extract}} +.0100_2 \xrightarrow[\text{decimal}]{\text{Convert to}} +.25_{10} = +\frac{1}{4} = +\frac{4}{16}$

# LIMIT CYCLES IN RECURSIVE SYSTEMS

Similarly, the y'(n) can be calculated for other values of n.

Here, when $n=3$, $y'(n)=y'(3)=\frac{2}{16}=0.0010_2$

when $n=4$, $y'(n)=y'(4)=\frac{1}{16}=0.0001_2$

when $n=5$, $y'(n)=y'(5)=\frac{1}{16}=0.0001_2$

For all values of $n \geq 4$, the $y'(n)=1/16=0.0001_2$

Hence the system output becomes constant for $n \geq 4$. Also for $n \geq 4$, the input x(n) is zero. Therefore the system enters a limit cycle for $n \geq 4$ even though the input becomes zero.

# LIMIT CYCLES OF FIRST-ORDER SYSTEM

| n | x(n) | y'(n) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | a = 1/2 | | a = –1/2 | | a = 3/4 | | a = –3/4 | |
| | | Binary | Decimal | Binary | Decimal | Binary | Decimal | Binary | Decimal |
| 0 | 15/16 | 0.1111 | $\frac{15}{16}$ | 0.1111 | $+\frac{15}{16}$ | 0.1011 | $\frac{11}{16}$ | 0.1011 | $+\frac{11}{16}$ |
| 1 | 0 | 0.1000 | $\frac{8}{16}$ | 1.1000 | $-\frac{8}{16}$ | 0.1000 | $\frac{8}{16}$ | 1.1000 | $-\frac{8}{16}$ |
| 2 | 0 | 0.0100 | $\frac{4}{16}$ | 0.0100 | $+\frac{4}{16}$ | 0.0110 | $\frac{6}{16}$ | 0.0110 | $+\frac{6}{16}$ |
| 3 | 0 | 0.0010 | $\frac{2}{16}$ | 1.0010 | $-\frac{2}{16}$ | 0.0101 | $\frac{5}{16}$ | 1.0101 | $-\frac{5}{16}$ |
| 4 | 0 | 0.0001 | $\frac{1}{16}$ | 0.0001 | $+\frac{1}{16}$ | 0.0100 | $\frac{4}{16}$ | 0.0100 | $+\frac{4}{16}$ |
| 5 | 0 | 0.0001 | $\frac{1}{16}$ | 1.0001 | $-\frac{1}{16}$ | 0.0011 | $\frac{3}{16}$ | 1.0011 | $-\frac{3}{16}$ |
| 6 | 0 | 0.0001 | $\frac{1}{16}$ | 0.0001 | $+\frac{1}{16}$ | 0.0010 | $\frac{2}{16}$ | 0.0010 | $+\frac{2}{16}$ |
| 7 | 0 | 0.0001 | $\frac{1}{16}$ | 1.0001 | $-\frac{1}{16}$ | 0.0010 | $\frac{2}{16}$ | 1.0010 | $-\frac{2}{16}$ |
| 8 | 0 | 0.0001 | $\frac{1}{16}$ | 0.0001 | $+\frac{1}{16}$ | 0.0010 | $\frac{2}{16}$ | 0.0010 | $+\frac{2}{16}$ |

# LIMIT CYCLES IN RECURSIVE SYSTEMS

- The limit cycles shown in table are due to quantization of the product by rounding. It can be shown that most of the limit cycles can be eliminated if quantization is performed by truncation, but truncation is not preferred in product quantization, due to the biased errors it may introduce in the output
- In a limit cycle the amplitudes of the output are confined to a range of values, which is called the dead band of the filter
- For a first-order system described by the equation, y(n) = ay(n-1)+x(n), the dead band is given by

$$\text{Dead band} = \pm\frac{2^{-B}}{1-|a|} = -\frac{2^{-B}}{1-|a|} \text{ to } +\frac{2^{-B}}{1-|a|}$$

# OVERFLOW LIMIT CYCLE

- In fixed point addition of two binary numbers the overflow occurs when the sum exceeds the finite word length of the register used to store the sum. The overflow in addition may lead to oscillations in the output which is referred to as **Overflow limit cycles**
- The overflow occurs when the sum exceeds the dynamic range of number system. When binary fraction format is used for computing, the dynamic range is (-1,1). The overflow is explained by considering 4-bit binary fraction number in two's complement representation
- The actual sum of + 3/8 and + 5/8 is +1 but due to overflow it becomes -1.

# OVERFLOW LIMIT CYCLE

Let us add $+\frac{3}{8}$ and $+\frac{5}{8}$ in two's complement addition

$$+\frac{3}{8} \Rightarrow 0.011$$

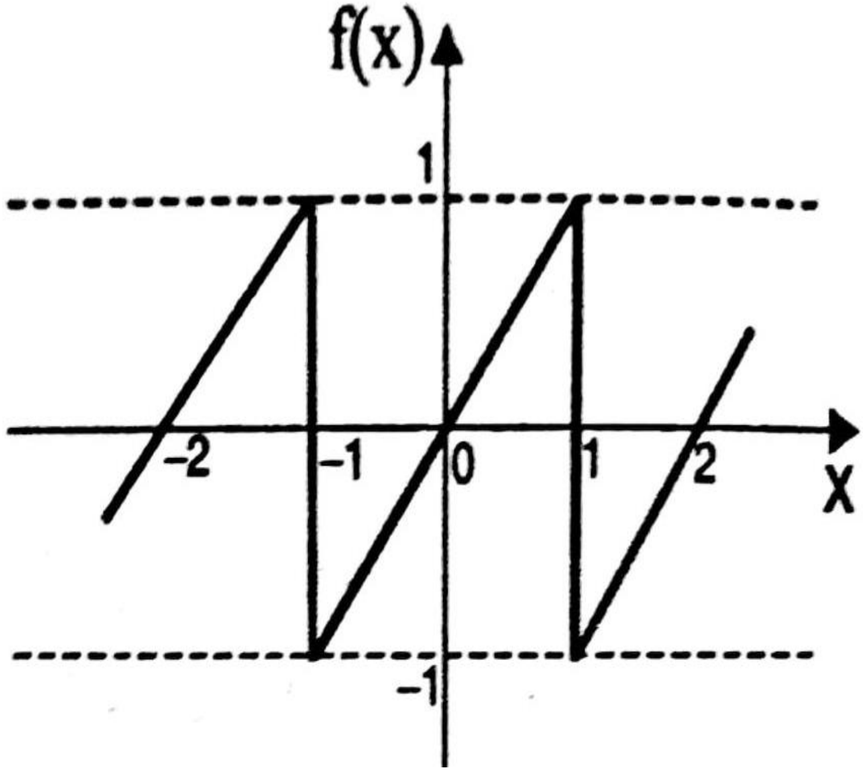
$$+\frac{5}{8} \Rightarrow 0.101$$

$$\frac{3}{8} + \frac{5}{8} \Rightarrow \underline{1.000} \Rightarrow -\frac{8}{8} = -1$$

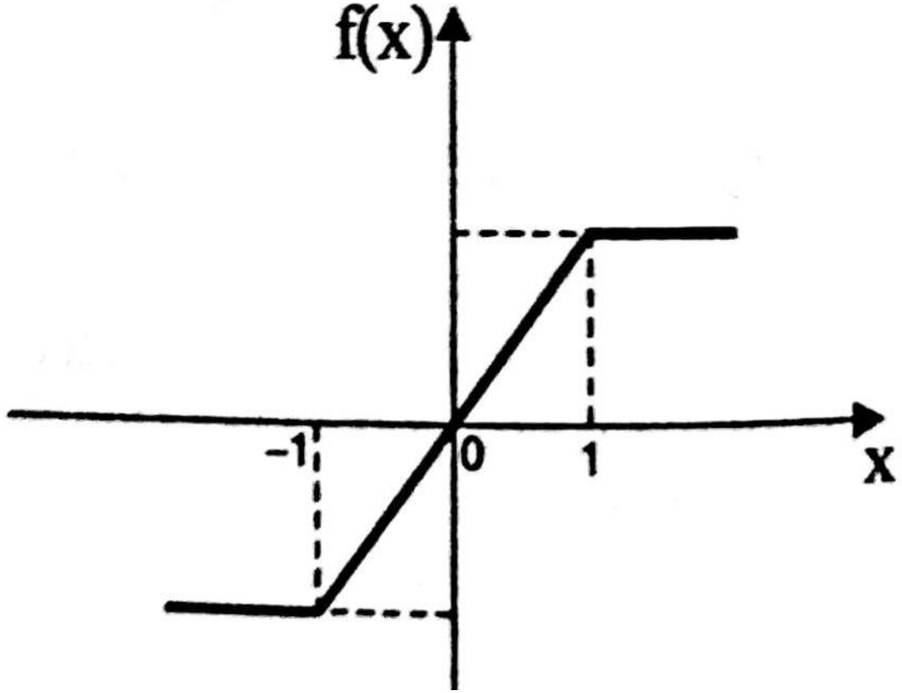
# FOUR –BIT TWO'S COMPLEMENT NUMBER

| Binary | Two's complement |
|---|---|
| 0 | 0.000 |
| 1/8 | 0.001 |
| 2/8 | 0.010 |
| 3/8 | 0.011 |
| 4/8 | 0.100 |
| 5/8 | 0.101 |
| 6/8 | 0.110 |
| 7/8 | 0.111 |
| –1 = –8/8 | 1.000 |
| –7/8 | 1.001 |
| –6/8 | 1.010 |
| –5/8 | 1.011 |
| –4/8 | 1.100 |
| –3/8 | 1.101 |
| –2/8 | 1.110 |
| –1/8 | 1.111 |

# OVERFLOW LIMIT CYCLE




**I/O Characteristics of two's complement adder**




**Characteristics of Saturation adder**

# OVERFLOW LIMIT CYCLE

- The overflow oscillations can be eliminated if saturation arithmetic is performed. In saturation arithmetic, when an overflow is sensed, the output (sum) is set equal to maximum allowable value and when an overflow is sensed, the output (sum) is set equal to minimum allowable value
- The saturation arithmetic introduce nonlinearity in the adder and the signal distortion due to this nonlinearity is small if the saturation occurs infrequently
- **Scaling to Prevent Overflow:** The two methods of preventing overflow are saturation arithmetic and scaling the input signal to the adder. In saturation arithmetic, undesirable distortion is introduced. Inorder to limit the signal distortion due to frequent overflows, the input signal to the adder can be scaled

# SCALING TO PREVENT OVERFLOW

- Let x(n) = Input to the system
- $h_k(n)$ = Impulse response between the input and output of node-k
- $Y_k(n)$ = response of the system at node-k

$$y_k(n) = h_k(n) * x(n) = \sum_{m=-\infty}^{+\infty} h_k(m)\, x(n-m)$$

- On taking absolute value of above equation, we get

$$|y_k(n)| = \left| \sum_{m=-\infty}^{+\infty} h_k(m)\, x(n-m) \right| = \sum_{m=-\infty}^{+\infty} |h_k(m)|\,|x(n-m)|$$

# ASSESSMENT

1. What are limit cycles?
2. The two types of limit cycles are ---------- and ----------
3. Define Zero input limit cycle.
4. In a limit cycle the amplitudes of the output are confined to a range of value and this range of value is called ---------- of the filter.
5. What is saturation arithmetic?
6. Define overflow limit cycle.
7. The overflow limit cycles can be eliminated either by using ---------- or by ----------
8. What is the drawback in saturation arithmetic?

# THANK YOU