UNIT-1

(Lecture-2)

Realization of Digital Systems: Block Diagram Representation

Raj Ranjan Prasad

Block Diagram Representation

- In the time domain, the input-output relations of an LTI digital filter is given by the convolution sum

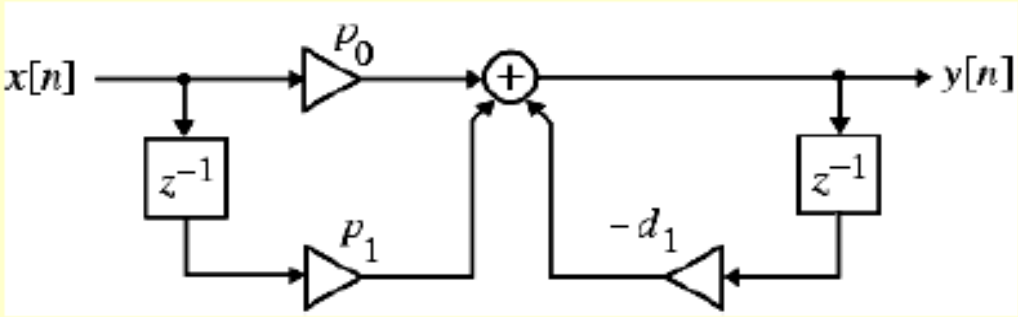
$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

or, by the linear constant coefficient difference equation

$$y[n] = -\sum_{k=1}^{N} d_k y[n-k] + \sum_{k=0}^{M} p_k x[n-k]$$

Block Diagram Representation

- For the implementation of an LTI digital filter, the input-output relationship must be described by a valid computational algorithm
- To illustrate what we mean by a computational algorithm, consider the causal first-order LTI digital filter shown below



Block Diagram Representation

- The filter is described by the difference equation

$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1]$$

- Using the above equation we can compute $y[n]$ for $n \geq 0$ knowing the initial condition $y[-1]$ and the input $x[n]$ for $n \geq -1$

Block Diagram Representation

$$y[0] = -d_1 y[-1] + p_0 x[0] + p_1 x[-1]$$

$$y[1] = -d_1 y[0] + p_0 x[1] + p_1 x[0]$$

$$y[2] = -d_1 y[1] + p_0 x[2] + p_1 x[1]$$

$$\vdots$$

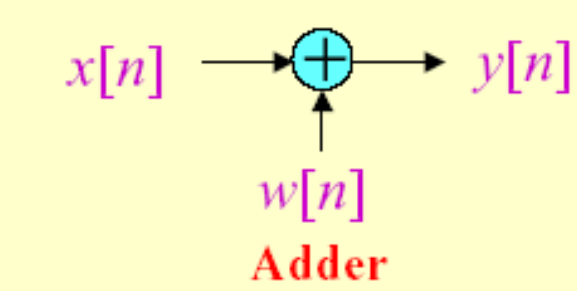
- We can continue this calculation for any value of the time index n we desire

Block Diagram Representation

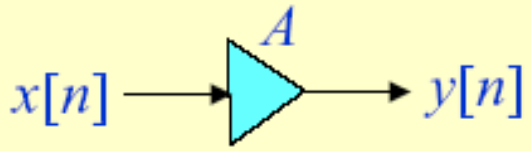
- Each step of the calculation requires a knowledge of the previously calculated value of the output sample (delayed value of the output), the present value of the input sample, and the previous value of the input sample (delayed value of the input)
- As a result, the first-order difference equation can be interpreted as a valid computational algorithm

Basic Building Blocks

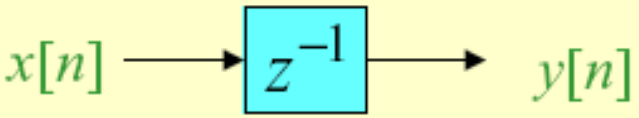
- The computational algorithm of an LTI digital filter can be conveniently represented in block diagram form using the basic building blocks shown below



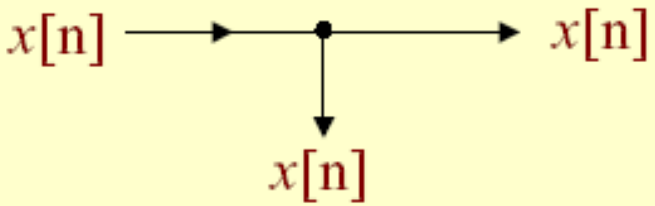
Adder



Multiplier



Unit delay



Pick-off node

Basic Building Blocks

Advantages of block diagrams:

1) Easy to write down the computational algorithm by inspection

2) Easy to analyze the block diagram to determine the explicit relation between the output and input

Basic Building Blocks

3) Easy to manipulate a block diagram to derive other "equivalent" block diagrams yielding different computational algorithms

4) Easy to determine the hardware requirements

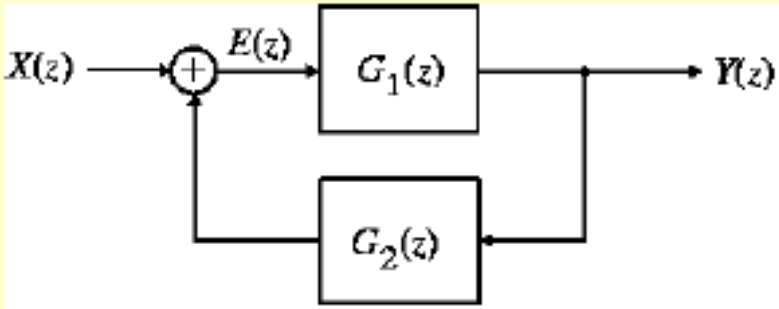
5) Easier to develop block diagram representations from the transfer function directly

Analysis of Block Diagrams

- Block diagrams can be analyzed by writing down the expressions for the output signals of each adder as a sum of its input signals, and developing a set of equations relating the filter input and output signals in terms of all internal signals
- Eliminating the unwanted internal variables then results in the expression for the output signal as a function of the input signal and the filter parameters that are the multiplier coefficients

Analysis of Block Diagrams

- Example: Consider the single-loop feedback structure shown below



- The output $E(z)$ of the adder is

$$E(z) = X(z) + G_2(z)Y(z)$$

- But from the figure, $Y(z) = G_1(z)E(z)$

Analysis of Block Diagrams

- Eliminating $E(z)$ from the previous two equations we arrive at

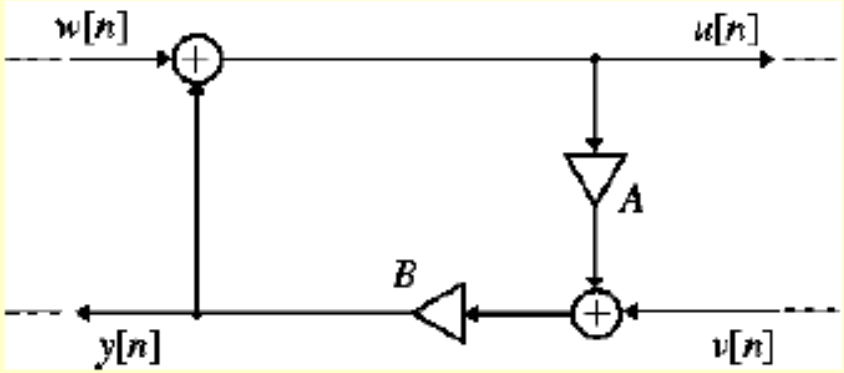
$$[1 - G_1(z)G_2(z)]Y(z) = G_1(z)X(z)$$

which leads to

$$H(z) = \frac{Y(z)}{X(z)} = \frac{G_1(z)}{1 - G_1(z)G_2(z)}$$

The Delay Free-Loop Problems

- For physical realizability of the digital filter structure, it is necessary that the block diagram contains no delay-free loops
- To illustrate the delay-free loop problem consider the structure below



The Delay Free-Loop Problems

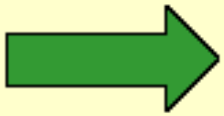
- Analysis of this structure yields

$$u[n] = w[n] + y[n]$$

$$y[n] = B(v[n] + Au[n])$$

which when combined results in

$$y[n] = B(v[n] + A(w[n] + y[n]))$$



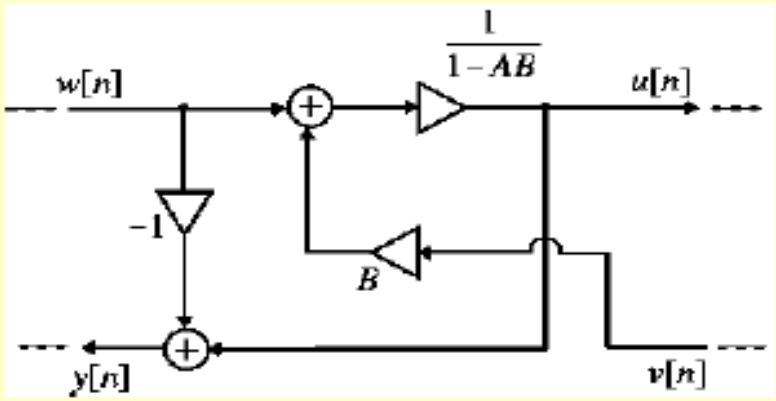
The determination of the current value of $y[n]$ requires the knowledge of the same value

The Delay Free-Loop Problems

- However, this is physically impossible to achieve due to the finite time required to carry out all arithmetic operations on a digital machine
- Method exists to detect the presence of delay-free loops in an arbitrary digital filter structure, along with methods to locate and remove these loops without altering the overall input-output relation

The Delay Free-Loop Problems

- Removal achieved by replacing the portion of the overall structure containing the delay-free loops by an equivalent realization with no delay-free loops
- Figure below shows such a realization of the example structure described earlier



Canonic and Noncanonic Structures

- A digital filter structure is said to be **canonic** if the number of delays in the block diagram representation is equal to the order of the transfer function
- Otherwise, it is a **noncanonic** structure

Canonic and Noncanonic Structures

- The structure shown below is noncanonic as it employs two delays to realize a first-order difference equation

$$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1]$$