第六章：离散时间系统结构

6.0 引言

具有有理系统函数的LTI系统,其输入输出序列满足线性常系数差分方程

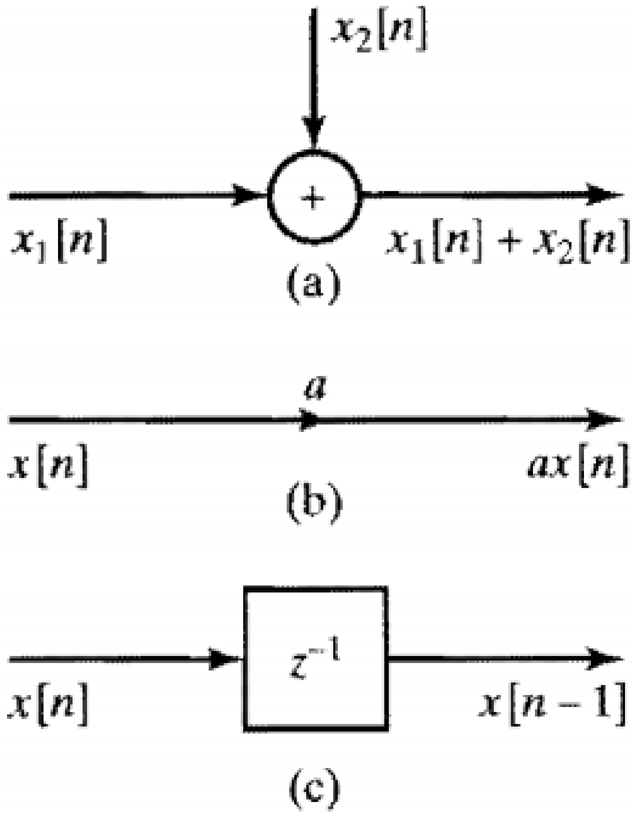
输入输出满足的线性常系数差分方程可由系统函数确定

离散时间系统采用硬件实现时，系统差分方程或系统函数需转换为技术可实现的算法或结构

差分方程描述的系统可由加、乘、和延迟等基本运算及互联构成

将系统用加、乘、和延迟等基本运算及其互连的方式表示称为方框图表示或信号流图（信流图）表示。

- 基本运算的方框图表示



两个序列相加（矢量加）

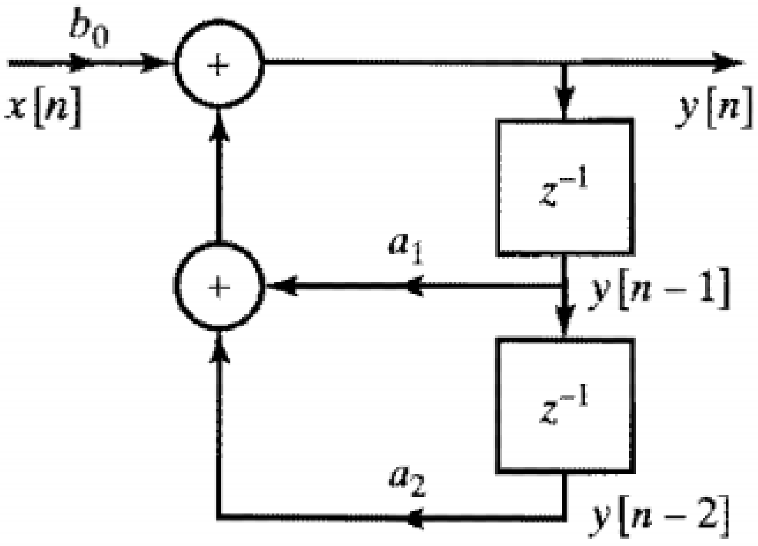
序列乘常数（标量乘）

序列单位延迟

例6.1 差分方程的方框图表示

系统差分方程为

$$y[n]=a_1y[n-1]+a_2y[n-2]+b_0x[n]$$



调整输出输入的顺序

$$y[n]-a_1y[n-1]-a_2y[n-2]=b_0x[n]$$

$$Y(z)-a_1Y(z)z^{-1}-a_2Y(z)z^{-2}=b_0X(z)$$

相应的系统函数为

$$H(z)=\frac{Y(z)}{X(z)}=\frac{b_0}{1-a_1z^{-1}-a_2z^{-2}}$$

◆ 直接I型实现

系统差分方程为

$$y[n]-\sum_{k=1}^{N}a_k y[n-k]=\sum_{k=0}^{M}b_k x[n-k]$$

相应的系统函数为

$$H(z)=\sum_{k=0}^{M}b_k z^{-k}\Big/\left(1-\sum_{k=1}^{N}a_k z^{-k}\right)$$

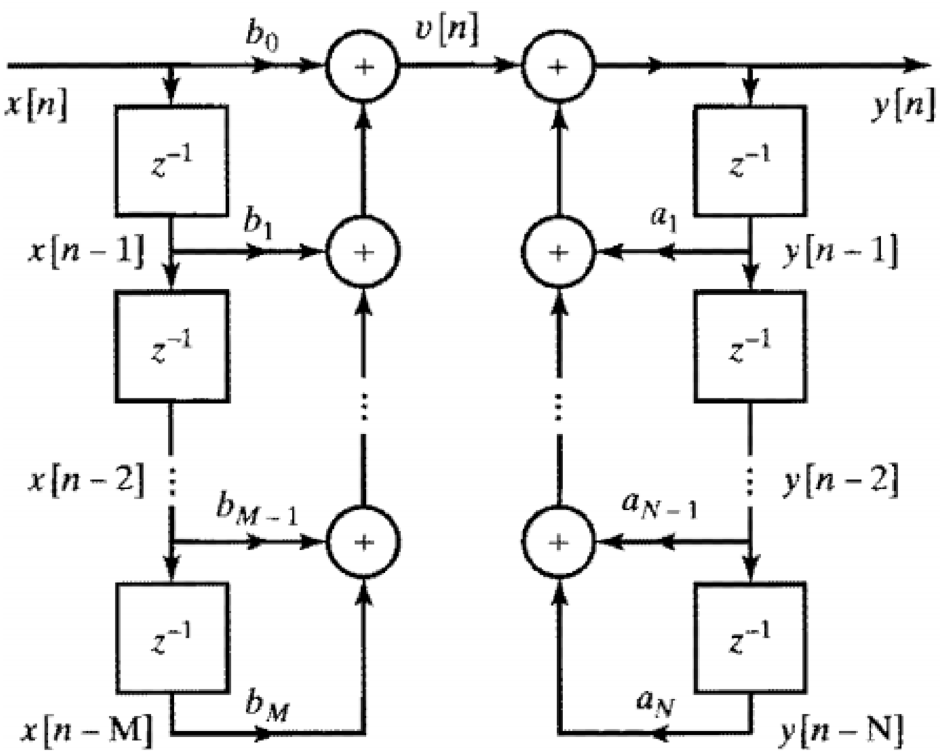
系统差分方程可改为

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

进一步可拆分为（并联关系）

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

$$y[n]=\sum_{k=1}^{N}a_k y[n-k]+v[n]$$



N阶差分方程的直接I型实现

◆ 直接II型实现

系统函数可分解为

$$H(z)=\frac{1}{1-\sum_{k=1}^{N}a_kz^{-k}}\sum_{k=0}^{M}b_kz^{-k}=H_2(z)H_1(z)$$

或等效为（级联关系）

$$V(z)=H_1(z)X(z)=\left(\sum_{k=0}^{M}b_kz^{-k}\right)X(z)$$

$$Y(z)=H_2(z)V(z)=\frac{1}{1-\sum_{k=1}^{N}a_kz^{-k}}V(z)$$

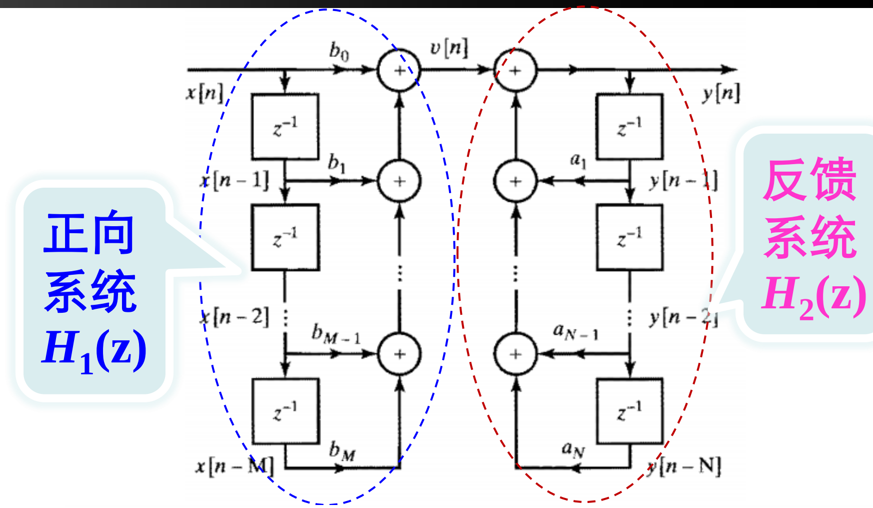
系统函数改写为

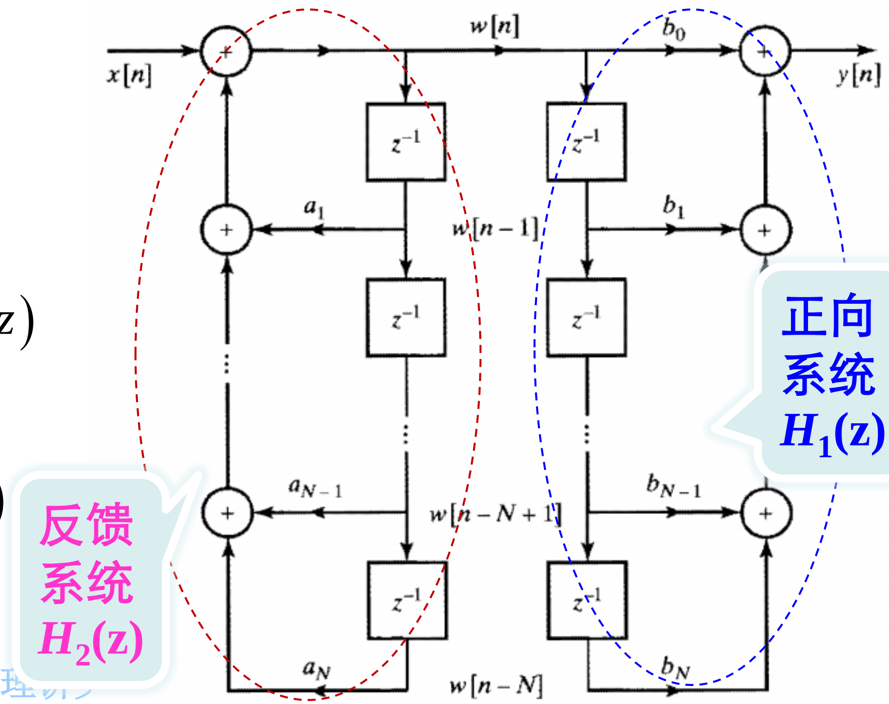
$$H(z)=\sum_{k=0}^{M}b_kz^{-k}\left(\frac{1}{1-\sum_{k=1}^{N}a_kz^{-k}}\right)=H_1(z)H_2(z)$$

或等效为

$$W(z)=H_2(z)X(z)=\frac{1}{1-\sum_{k=1}^{N}a_kz^{-k}}X(z)$$

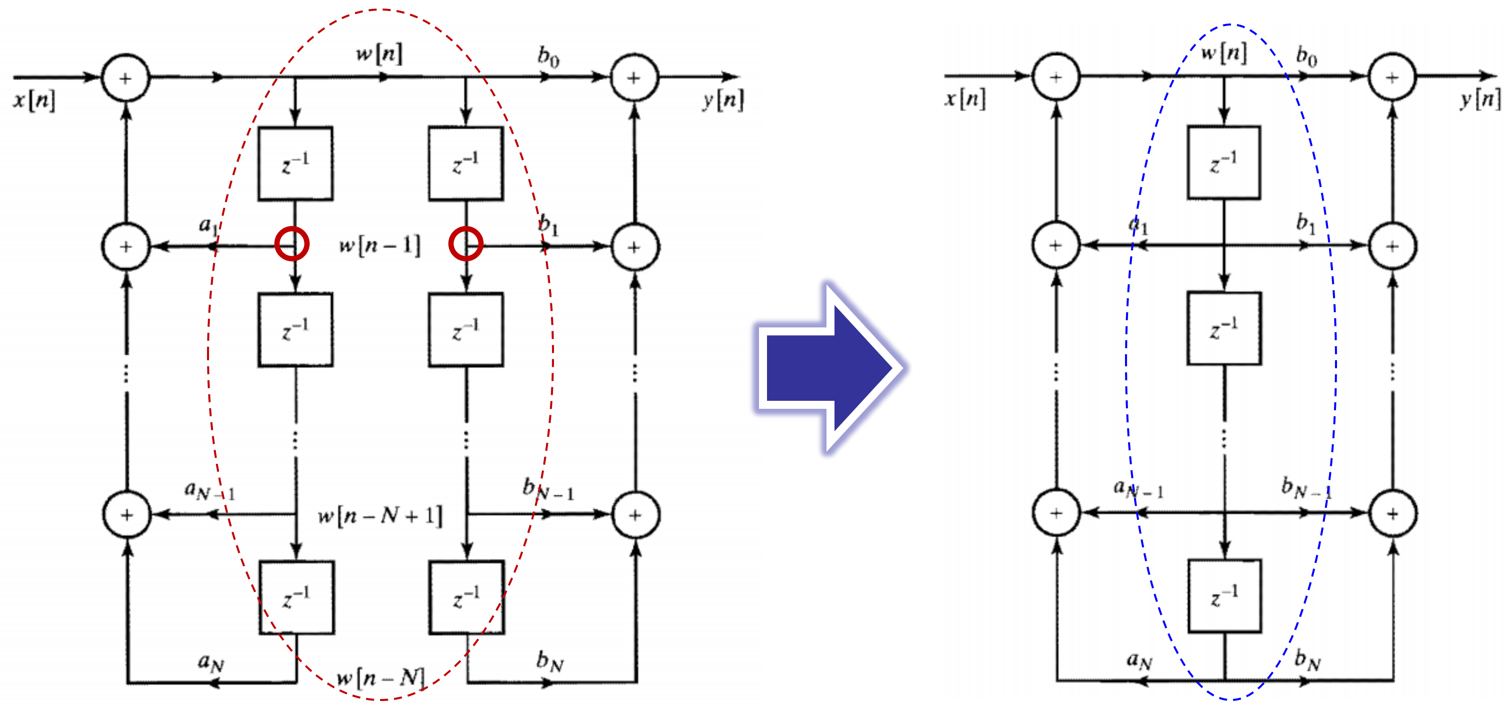
$$Y(z)=H_1(z)W(z)=\left(\sum_{k=0}^{M}b_kz^{-k}\right)W(z)$$





6.1 线性常系数差分方程的方框图表示

◆ 直接II型实现（续）



*N*阶差分方程的直接II型实现
或规范型实现

直接I型实现需要延迟单元数：***M*+*N***

直接II型实现需要延迟单元数：**max（*M*，*N*）**

例6.2 LTI系统的直接I和II型实现

系统函数为

$$H(z)=\frac{1+2z^{-1}}{1-1.5z^{-1}+0.9z^{-2}}$$

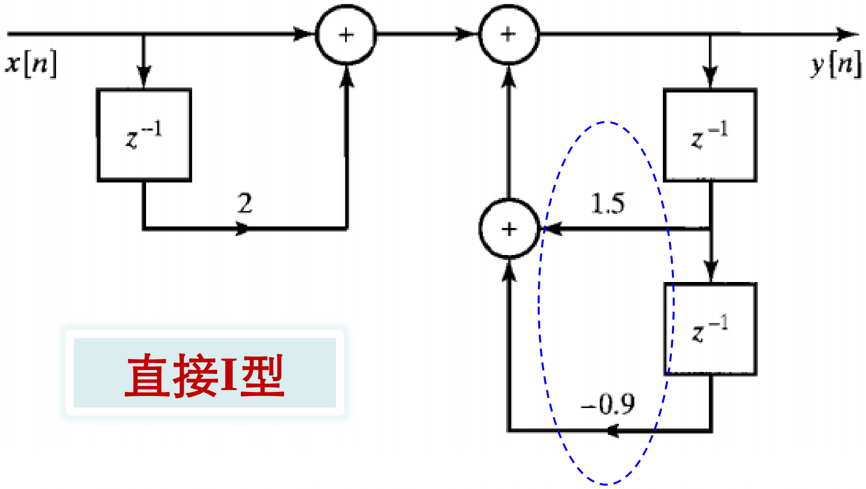
可得系数 $b_0=1$，$b_1=2$，以及

$a_1=1.5$，$a_2=-0.9$

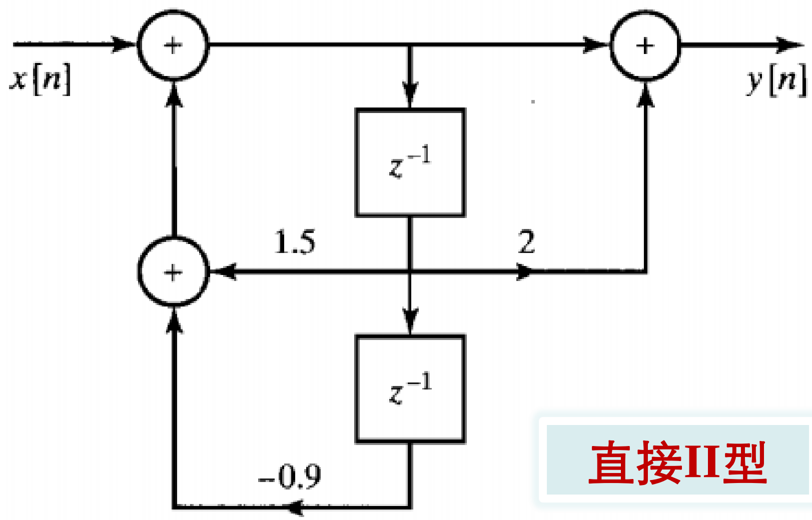
直接I型实现：系统函数的分子系数对应输入处理部分，分母系数对应输出处理部分

直接II型实现：系统函数的分子系数对应输出处理部分，分母系数对应输入处理部分

反馈回路的系数正负号与系统函数分母多项式中的符号相反



直接I型

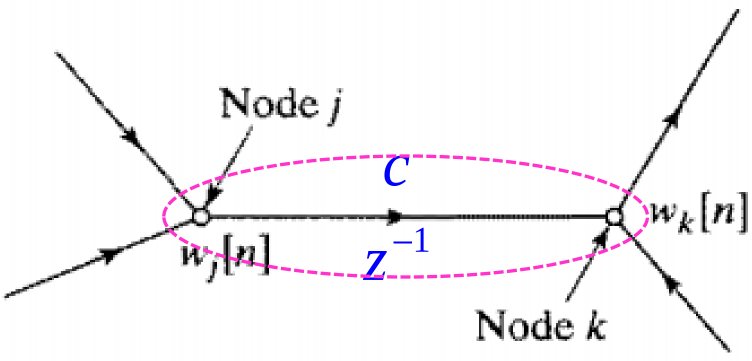


直接II型

6.2 线性常系数差分方程的信号流图表示

信号流图为一个将节点互联的有向支路的网络

- 每个节点与一个变量或节点值关联，第 k 个节点的节点值记为 $w_k[n]$
- 每个节点的节点值为进入该节点的全部支路的输出之和
- 由节点 j 出发，到节点 k 终止的支路记为支路 (j,k)



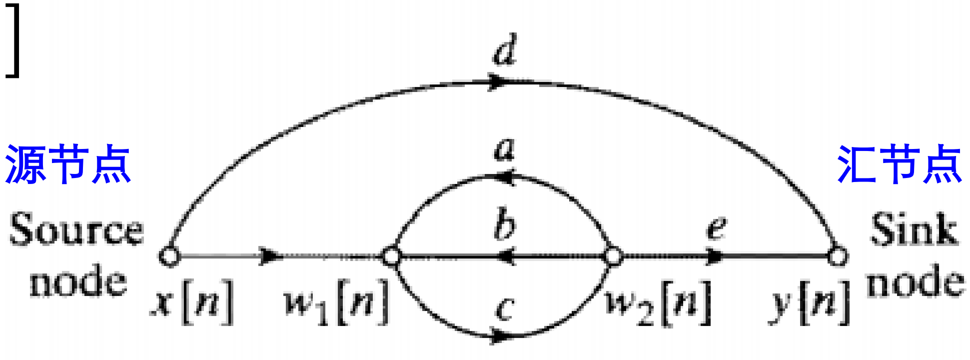
某一支路的输出为对该支路输入的线性变换（如常数乘、延迟等），该线性变换在支路上靠近指明方向的箭头处给出运算标记

如果支路未给出支路运算标记，则说明是一条传输为1的支路，或该支路实现对其输入的恒等变换。

6.2 线性常系数差分方程的信号流图表示

◆线性方程的信流图表示示例

$$w_1[n] = x[n] + aw_2[n] + bw_2[n]$$
$$w_2[n] = cw_1[n]$$
$$y[n] = dx[n] + ew_2[n]$$



- 信流图代表一个差分方程组
- 每个节点值都对应一个方程

源节点：没有流进支路的节点

汇节点：没有流出支路的节点

◆方框图与信流图等效表示

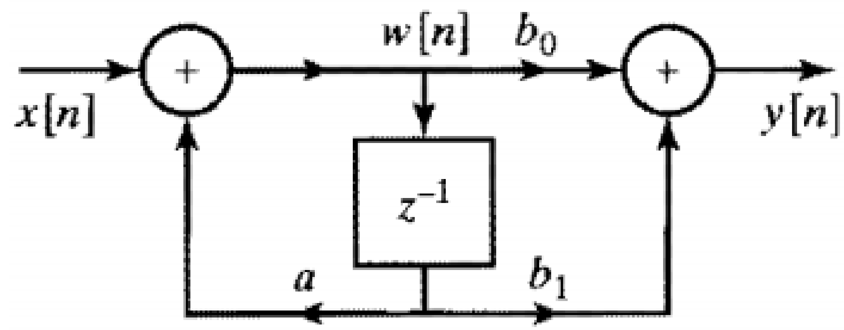
$$w_1[n]=x[n]+aw_4[n]$$

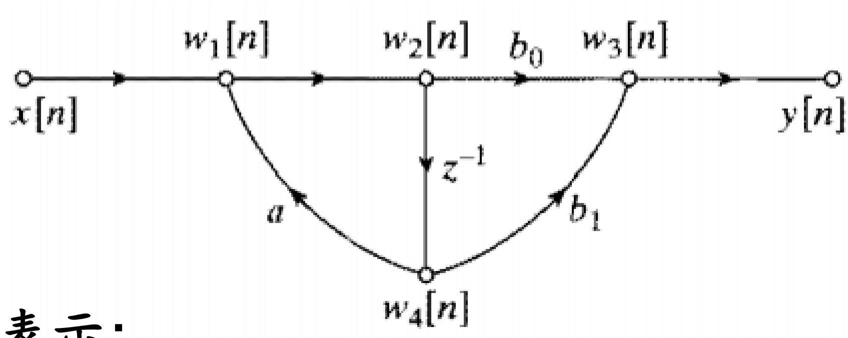
$$w_2[n]=w_1[n]$$

$$w_3[n]=b_0w_2[n]+b_1w_4[n]$$

$$w_4[n]=w_2[n-1]$$

$$y[n]=w_3[n]$$





◆方框图与信流图差异

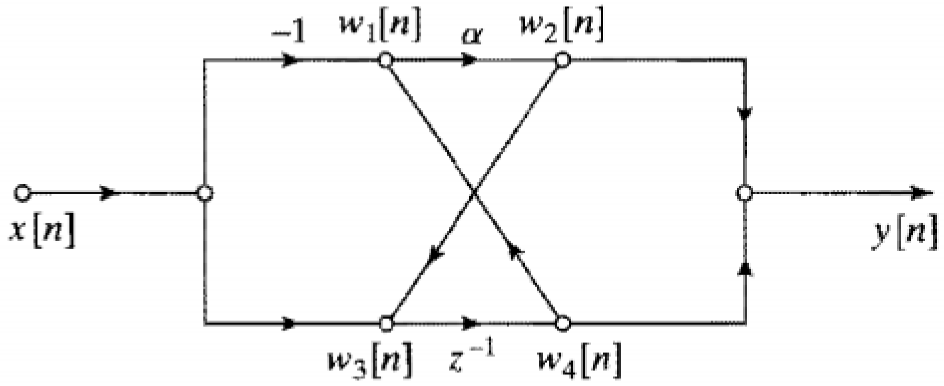
✓方框图的加法器和延迟用专用符号表示；

信流图的节点既代表分支点，又代表加法器；

✓方框图比较直观；

信流图更为简洁

◆由信流图确定系统函数



1）由信号流图获得差分方程

$$w_1[n]=w_4[n]-x[n]$$
$$w_2[n]=aw_1[n]$$
$$w_3[n]=w_2[n]+x[n]$$
$$w_4[n]=w_3[n-1]$$
$$y[n]=w_2[n]+w_4[n]$$

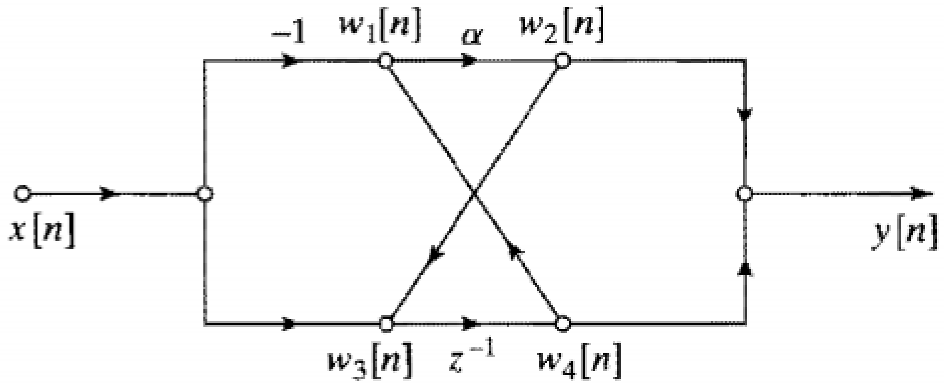
2）由差分方程获得其z变换

$$W_1(z)=W_4(z)-X(z)$$
$$W_2(z)=aW_1(z)$$
$$W_3(z)=W_2(z)+X(z)$$
$$W_4(z)=z^{-1}W_3(z)$$
$$Y(z)=W_2(z)+W_4(z)$$

6.2 线性常系数差分方程的信号流图表示

◆由信流图确定系统函数（续）

3）解代数方程得



$$W_2(z)=\frac{a\left(z^{-1}-1\right)}{1-az^{-1}}X(z)$$

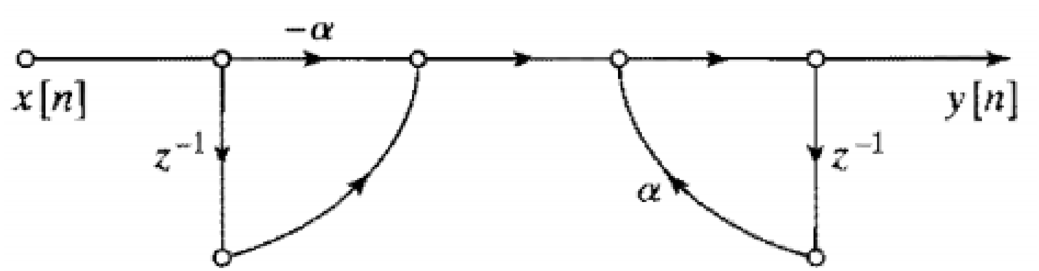
$$W_4(z)=\frac{z^{-1}(1-a)}{1-az^{-1}}X(z)$$

即

$$Y(z)=\left\{\frac{a\left(z^{-1}-1\right)}{1-az^{-1}}+\frac{z^{-1}(1-a)}{1-az^{-1}}\right\}X(z)=\frac{z^{-1}-a}{1-az^{-1}}X(z)$$

4）系统函数可表示为

$$H(z)=\frac{Y(z)}{X(z)}=\frac{z^{-1}-a}{1-az^{-1}}$$



6.3.1 直接型IIR系统的结构

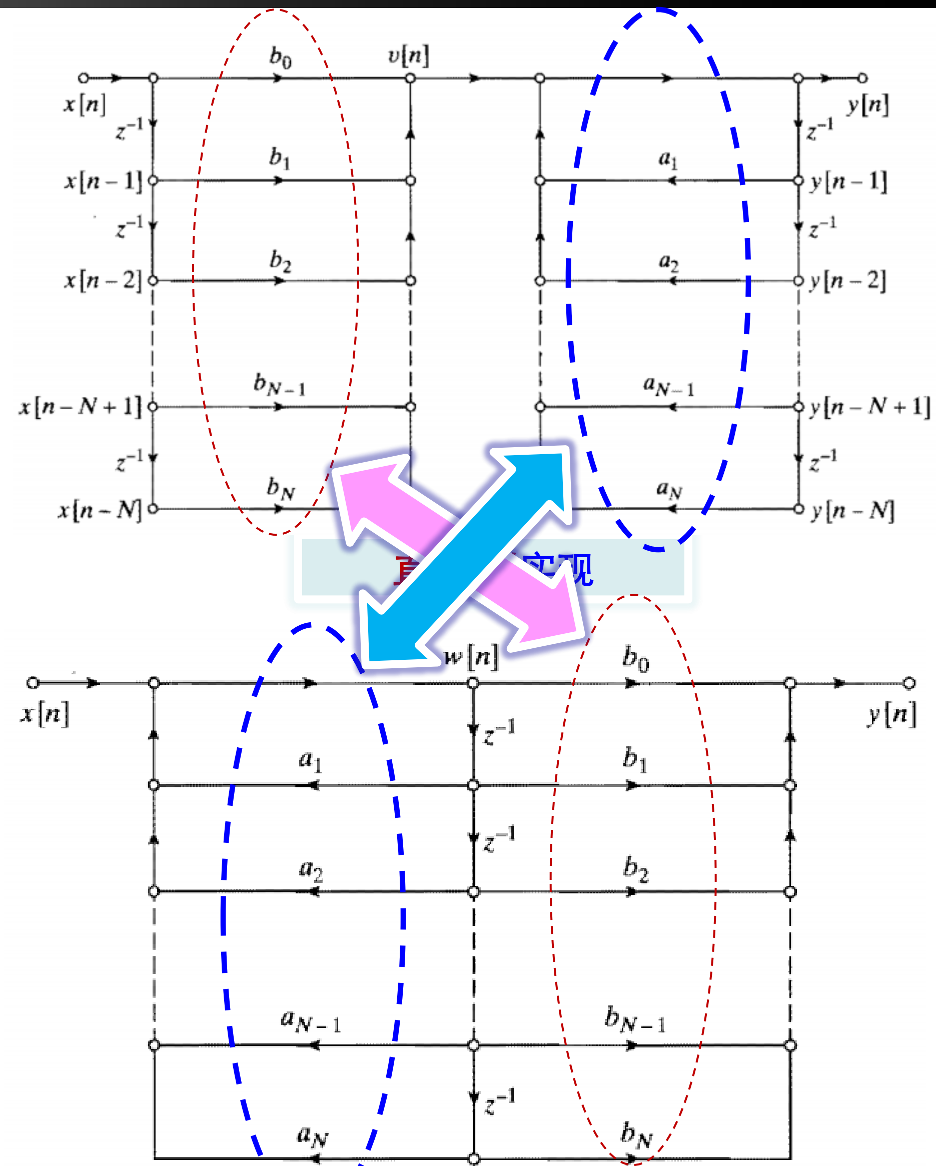
◆ IIR系统的直接型结构

系统的输入输出满足差分方程

$$y[n]-\sum_{k=1}^{N}a_k y[n-k]=\sum_{k=0}^{M}b_k x[n-k]$$

相应的系统函数为

$$H(z)=\frac{\sum_{k=0}^{M}b_k z^{-k}}{1-\sum_{k=1}^{N}a_k z^{-k}}$$



直接II型实现

6.3.2 级联型IIR系统的结构

◆ IIR系统的级联型结构

系统函数采用多项式因式乘积形式表示为

$$H(z)=A\frac{\prod_{k=1}^{M_1}\left(1-f_k z^{-1}\right)\prod_{k=1}^{M_2}\left(1-g_k z^{-1}\right)\left(1-g_k^* z^{-1}\right)}{\prod_{k=1}^{N_1}\left(1-c_k z^{-1}\right)\prod_{k=1}^{N_2}\left(1-d_k z^{-1}\right)\left(1-d_k^* z^{-1}\right)}$$

此结构为一种由一阶和二阶因子构成的通用实现结构。

式中 $M_1+2M_2=M$ ，$N_1+2N_2=N$；f_k 和 c_k 分别表示实零点和实极点；g_k 和 g_k^* 表示共轭零点， d_k 和 d_k^* 表示共轭极点。

一种标准实现结构是将上述系统函数中的每对实因子和每对复数共轭因子都配成二阶因子，系统函数可表示为

$$H(z)=\prod_{k=1}^{N_s}\frac{b_{0k}+b_{1k}z^{-1}+b_{2k}z^{-2}}{1-a_{1k}z^{-1}+a_{2k}z^{-2}}$$

式中 $N_s=\lfloor (N+1)/2 \rfloor$，为 $<(N+1)/2$ 的最大整数，并假设$M<N$。

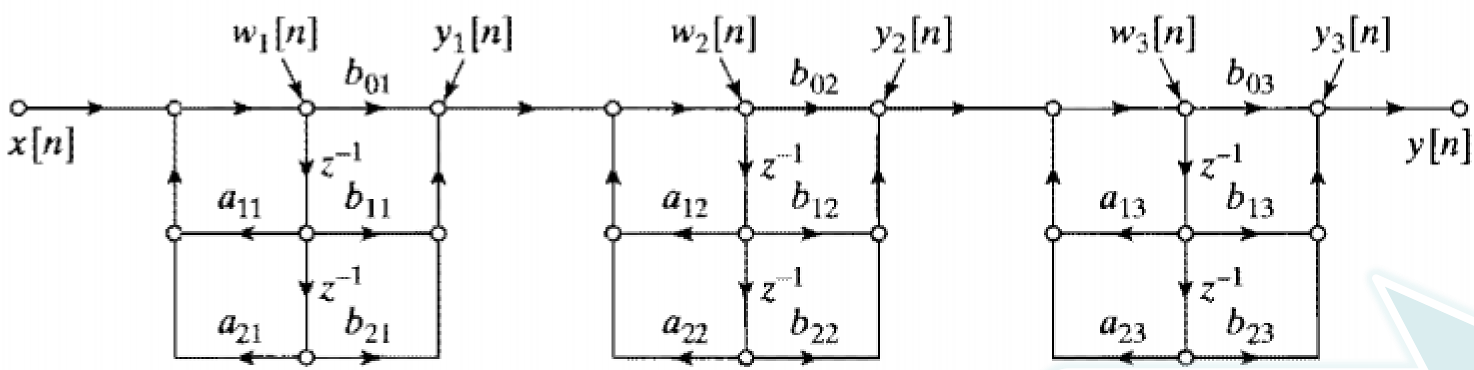
6.3.2 级联型IIR系统的结构

◆ IIR系统的级联型结构（续）

基于二阶因式表示的系统函数

$$H(z)=\prod_{k=1}^{N_s}\frac{b_{0k}+b_{1k}z^{-1}+b_{2k}z^{-2}}{1-a_{1k}z^{-1}+a_{2k}z^{-2}}$$

系统的信流图可表示为直接II型实现的2阶网段（section）级联



相应的差分方程为

$$y_0[n]=x[n]$$

$$w_k[n]=a_{1k}w_k[n-1]+a_{2k}w_k[n-2]+y_{k-1}[n],\quad k=1,2,\ldots,N_s$$

$$y_k[n]=b_{0k}w_k[n]+b_{1k}w_k[n-1]+b_{2k}w_k[n-2],\quad k=1,2,\ldots,N_s$$

$$y[n]=y_{N_s}[n]$$

为降低延迟单元开销，每个网段采用直接II型结构

6.3.3 并联型IIR系统的结构

◆ IIR系统的并联型结构

将有理系统函数进行部分分式展开

$$H(z)=\sum_{k=0}^{N_p} C_k z^{-k}+\sum_{k=1}^{N_1}\frac{A_k}{1-c_k z^{-1}}+\sum_{k=1}^{N_2}\frac{B_k\left(1-e_k z^{-1}\right)}{\left(1-d_k z^{-1}\right)\left(1-d_k^* z^{-1}\right)}$$

N_p+1个幅度加权/延迟支路（FIR系统）

N_1个一阶IIR系统

N_2个二阶IIR系统

式中分母因子总数 $N=N_1+2N_2$；若分子因子总数 $M\geq N$，则 $N_p=M-N$。

若系统仅有实极点，且将实极点成对组合，则系统函数可表示为

$$H(z)=\sum_{k=0}^{N_p} C_k z^{-k}+\sum_{k=1}^{N_s}\frac{e_{0k}+e_{1k}z^{-1}}{1-a_{1k}z^{-1}-a_{2k}z^{-2}}$$

式中 $N_s=\lfloor (N+1)/2\rfloor=\lfloor (N_1+1)/2\rfloor$。

6.3.3 并联型IIR系统的结构

◆ IIR系统的并联型结构（续）

基于二阶因式表示的系统函数

$$H(z)=\sum_{k=0}^{N_p} C_k z^{-k}+\sum_{k=1}^{N_s}\frac{e_{0k}+e_{1k}z^{-1}}{1-a_{1k}z^{-1}-a_{2k}z^{-2}}$$

例： $N=M=6$系统

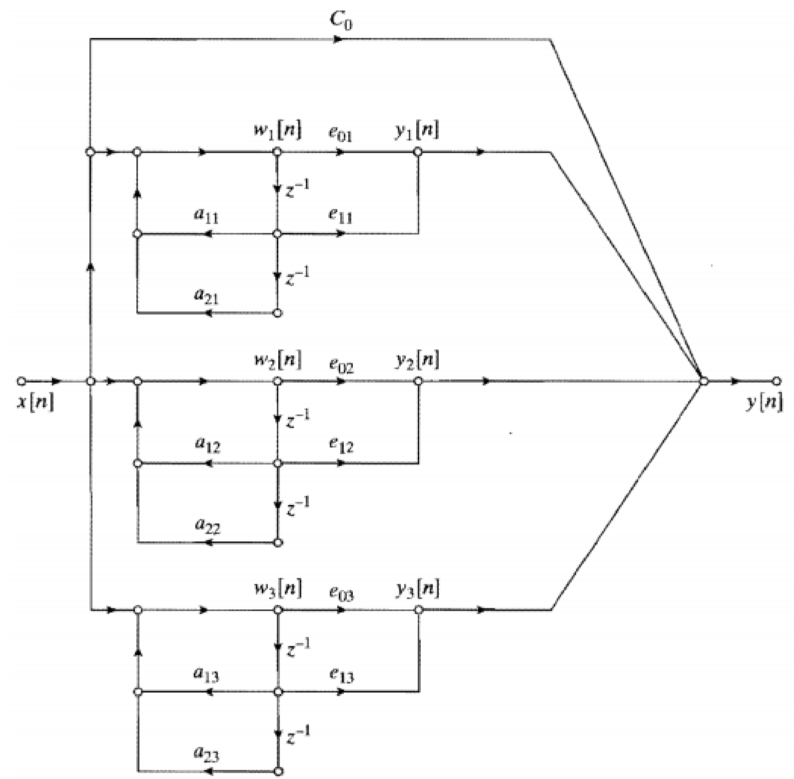
$$N_s=3 \quad N_p=0$$

相应的差分方程为

$$w_k[n]=a_{1k}w_k[n-1]+a_{2k}w_k[n-2]+x[n],\quad k=1,2,3$$

$$y_k[n]=e_{0k}w_k[n]+e_{1k}w_k[n-1],\quad k=1,2,3$$

$$y[n]=C_0x[n]+\sum_{k=1}^{3}y_k[n]$$



6.3.3 并联型IIR系统的结构

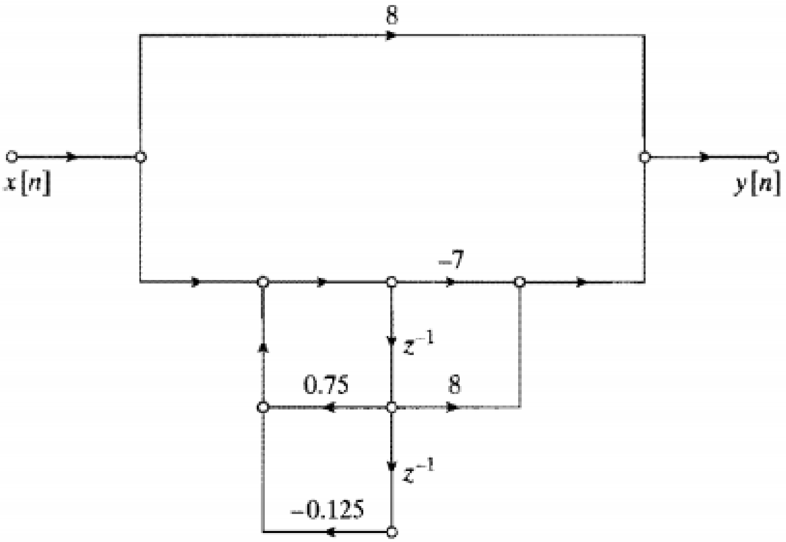
例6.6：并联型结构实现

系统函数为

$$H(z)=\frac{1+2z^{-1}+z^{-2}}{1-0.75z^{-1}+0.125z^{-2}}$$

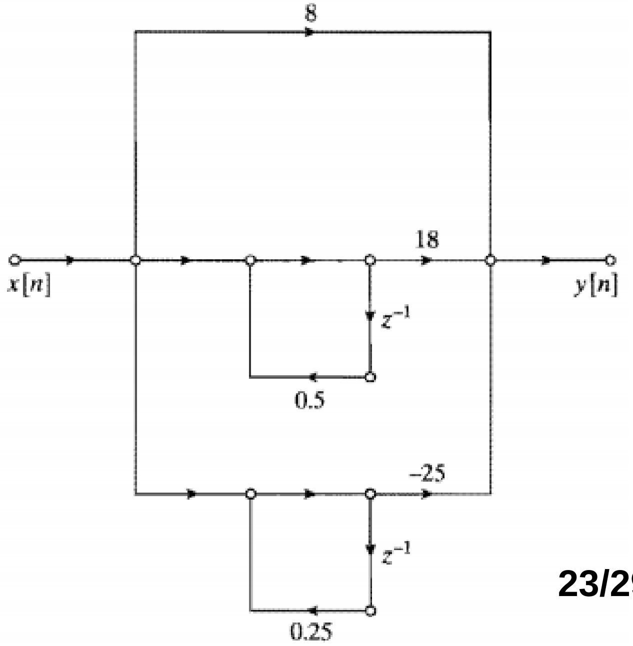
- 采用二阶网段并联实现

$$H(z)=8+\frac{-7+8z^{-1}}{1-0.75z^{-1}+0.125z^{-2}}$$



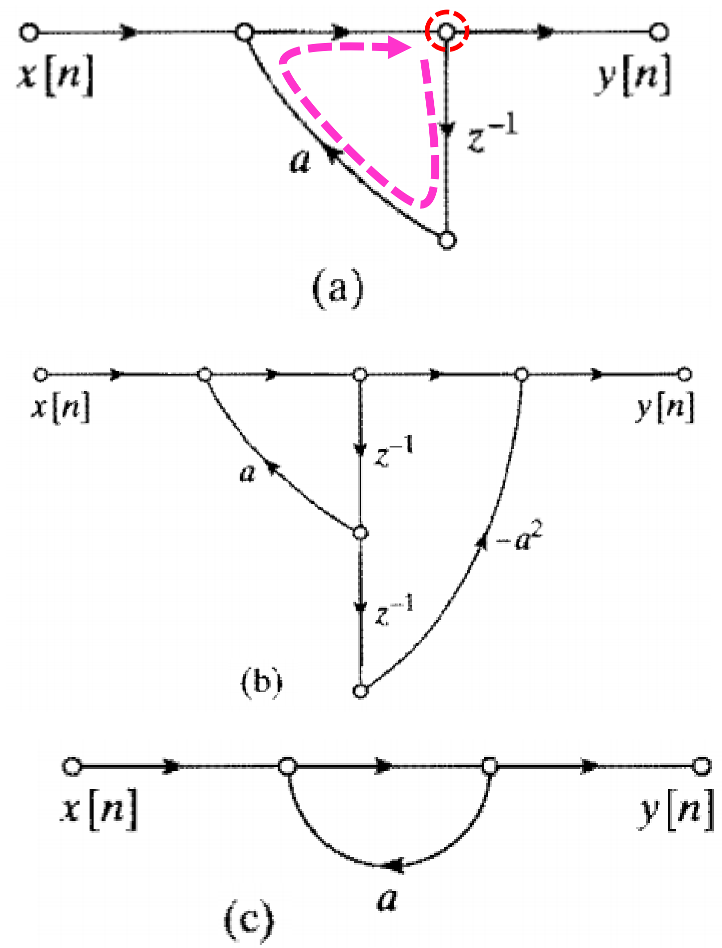
- 采用一阶网段并联实现

$$H(z)=8+\frac{18}{1-0.5z^{-1}}-\frac{25}{1-0.25z^{-1}}$$



6.3.4 IIR系统的反馈

反馈回路：是一条闭合路径，该路径从某一节点出发，以箭头方向穿过某些支路又回到那个节点。



- 系统存在反馈回路是系统产生无限脉冲响应的必要条件。
- 有些系统存在反馈回路但系统却有有限长的脉冲响应。

如系统

$$H(z)=\frac{1-a^2z^{-2}}{1-az^{-1}}$$

不可计算网络：网络中的节点变量不可以依次计算获得。

$$y[n]=ay[n]+x[n]$$

$$y[n]=x[n]/(1-a)$$

第六章：离散时间系统结构

6.4 转置形式

流图转置：将网络中所有支路的**方向颠倒**，但支路**增益不变**，并将输入和输出颠倒。

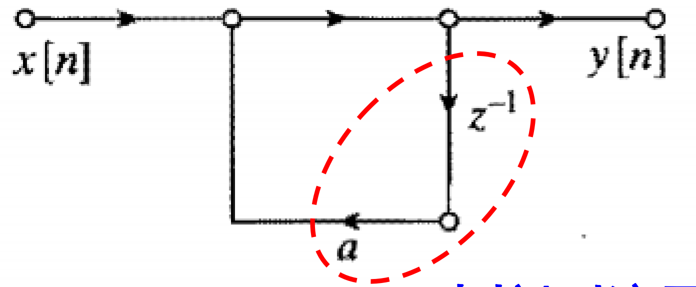
流图转置所得流图与原流图具有**相同的系统函数**。

例6.7 单极点一阶系统的转置型

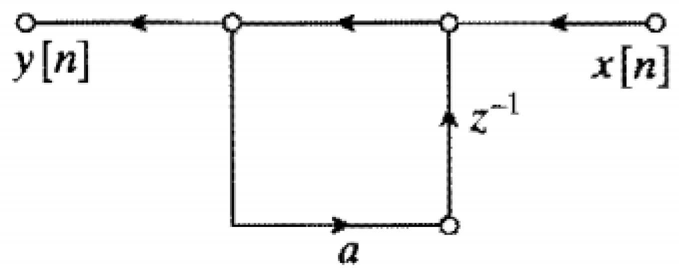
一阶系统函数为

$$H(z)=\frac{1}{1-az^{-1}}$$

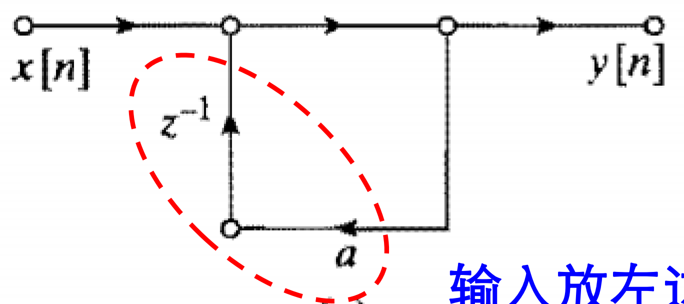
流图（a）与（c）系统函数相同



(a) 直接I型流图



(b) 转置形式



(c) 输入放左边

6.4 转置形式

例6.8 基本二阶网段的转置型

右图二阶系统的差分方程为

$$w[n]=a_1w[n-1]+a_2w[n-2]+x[n]$$

$$y[n]=b_0w[n]+b_1w[n-1]+b_2w[n-2]$$

其转置流图的差分方程为

$$y[n]=v_0[n]$$

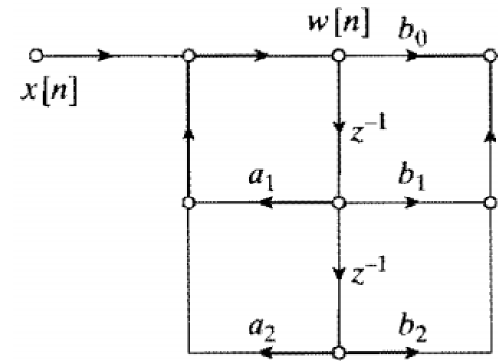
$$v_0[n]=b_0x[n]+v_1[n-1]$$

$$v_1[n]=a_1y[n]+b_1x[n]+v_2[n-1]$$

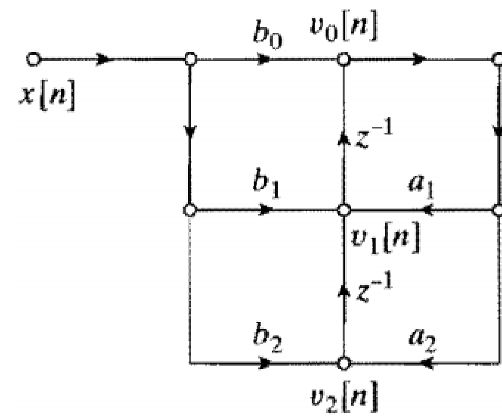
$$v_2[n]=a_2y[n]+b_2x[n]$$

转置型差分方程可表示为

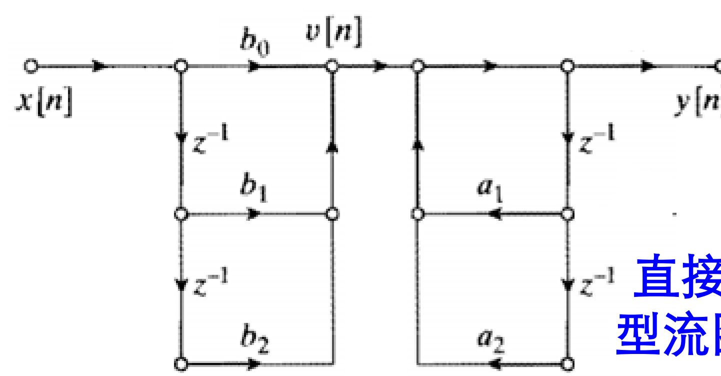
$$y[n]=a_1y[n-1]+a_2y[n-2]+$$
$$b_0x[n]+b_1x[n-1]+b_2x[n-2]$$



直接II型流图



转置型流图

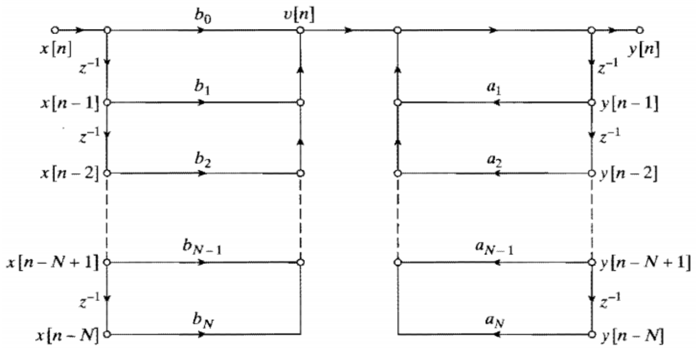


直接I型流图

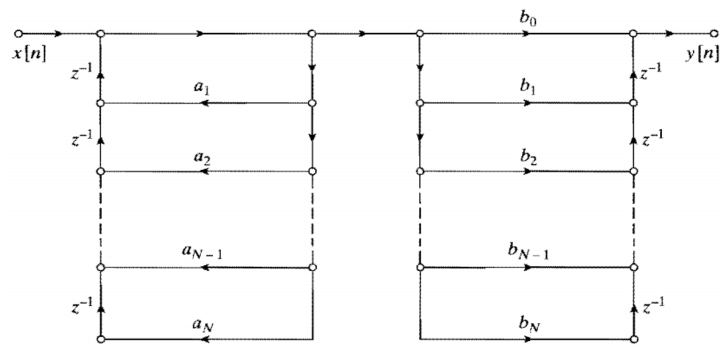
6.4 转置形式

系统差分方程为 $$y[n]-\sum_{k=1}^{N}a_k y[n-k]=\sum_{k=0}^{N}b_k x[n-k]$$

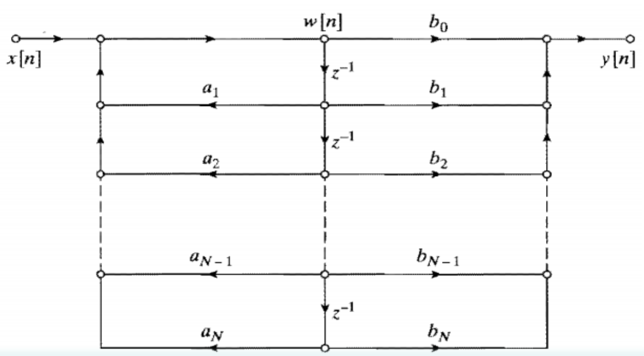
原流图形式



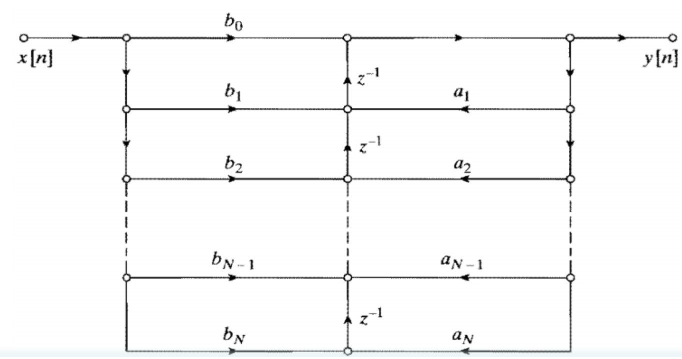
直接I型



转置形式



直接II型



转置型与原流图有相同的支路和系数数量，不改变流图框架

转置型相对原流图改变了零极点实现顺序，会影响系统性能

转置型为产生新的结构给出了一种简便的方法。