Digital Signal Processing
Transposed Forms

D. Richard Brown III

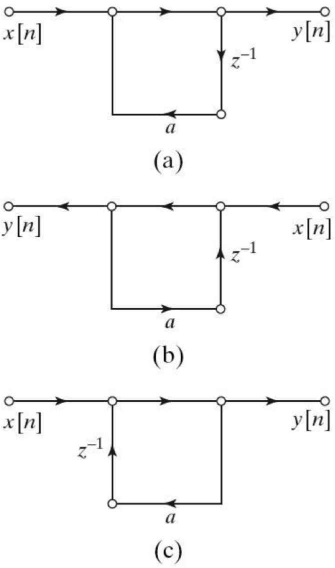
Structure Transposition

In general, for single-input single-output systems, we can form an equivalent structure by performing the **transpose operation** on any structure, which is defined as follows:

1. Reverse all paths
2. Replace “pick-off” nodes with accumulators, and vice-versa
3. Interchange input and output

Also called “flow graph reversal”.

Simple Example of Transposition

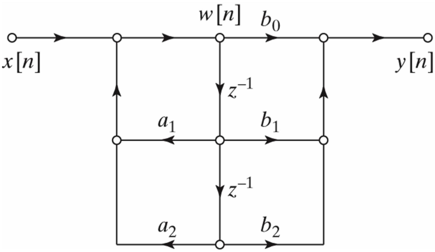


Original system

Reverse paths
Replace nodes with sums and vice versa
Interchange input and output

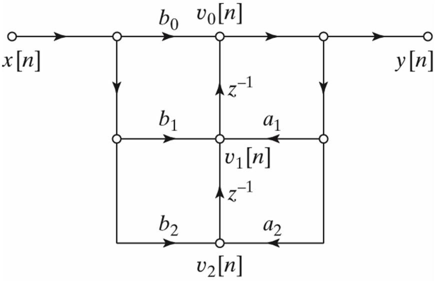
Redraw figure with input on left and output on right

Direct Form II Transposed IIR Structures



DF-II

$$W(z) = X(z) + a_1 z^{-1} W(z) + a_2 z^{-2} W(z)$$
$$Y(z) = b_0 W(z) + b_1 z^{-1} W(z) + b_2 z^{-2} W(z)$$



DF-II transposed

$$V_2(z) = b_2 X(z) + a_2 Y(z)$$
$$V_1(z) = z^{-1} V_2(z) + b_1 X(z) + a_1 Y(z)$$
$$V_0(z) = z^{-1} V_1(z) + b_0 X(z)$$
$$Y(z) = V_0(z)$$

Equivalence

For the DF-II structure, we have

$$W(z) = X(z) + a_1 z^{-1} W(z) + a_2 z^{-2} W(z)$$
$$Y(z) = b_0 W(z) + b_1 z^{-1} W(z) + b_2 z^{-2} W(z)$$

We can solve the first equation for $W(z)$ to get

$$W(z) = \frac{1}{1 - a_1 z^{-1} W(z) - a_2 z^{-2}} X(z)$$

and substitute into the second equation to get

$$Y(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} W(z) - a_2 z^{-2}} X(z)$$

Equivalence

For the DF-II structure, we have

$$W(z) = X(z) + a_1 z^{-1} W(z) + a_2 z^{-2} W(z)$$
$$Y(z) = b_0 W(z) + b_1 z^{-1} W(z) + b_2 z^{-2} W(z)$$

We can solve the first equation for $W(z)$ to get

$$W(z) = \frac{1}{1 - a_1 z^{-1} W(z) - a_2 z^{-2}} X(z)$$

and substitute into the second equation to get

$$Y(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} W(z) - a_2 z^{-2}} X(z)$$

For the DF-II transposed structure, we have

$$V_2(z) = b_2 X(z) + a_2 Y(z)$$
$$V_1(z) = z^{-1} V_2(z) + b_1 X(z) + a_1 Y(z)$$
$$V_0(z) = z^{-1} V_1(z) + b_0 X(z)$$
$$Y(z) = V_0(z)$$

We can write

$$\begin{aligned} Y(z) &= z^{-1} V_1(z) + b_0 X(z) \\ &= z^{-1} \left[z^{-1} V_2(z) + b_1 X(z) + \right. \\ &\quad \left. a_1 Y(z) \right] + b_0 X(z) \\ &= z^{-2} V_2(z) + (b_1 z^{-1} + b_0) X(z) + \\ &\quad a_1 z^{-1} Y(z) \\ &= (b_2 z^{-2} b_1 z^{-1} + b_0) X(z) + \\ &\quad (a_2 z^{-2} + a_1 z^{-1}) Y(z) \end{aligned}$$

which (after isolating $Y(z)$) is equivalent.

Pseudocode Implementation of IIR DF-II Transposed

```
<read in new input x[n]>

<compute new intermediate values, from top
    v[0] = b[0]*x[n] + v[1]
    v[1] = b[1]*x[n] + a[1]*y[n] + v[2]
    ...
    v[N-1] = b[N-1]*x[n] + a[N-1]*y[n] + v[N]
    v[N] = b[N]*x[n] + a[N]*y[n]
>


<compute new output
    y[n] = v[0]
>
```