Polyphase Filters

Section 12.4 Porat

12.4 Polyphase Filters

Polyphase is a way of doing sampling-rate conversion that leads to very efficient implementations.

But more than that, it leads to very general viewpoints that are useful in building filter banks.

Before we delve into the math we can see a lot just by looking at the structure of the filtering….

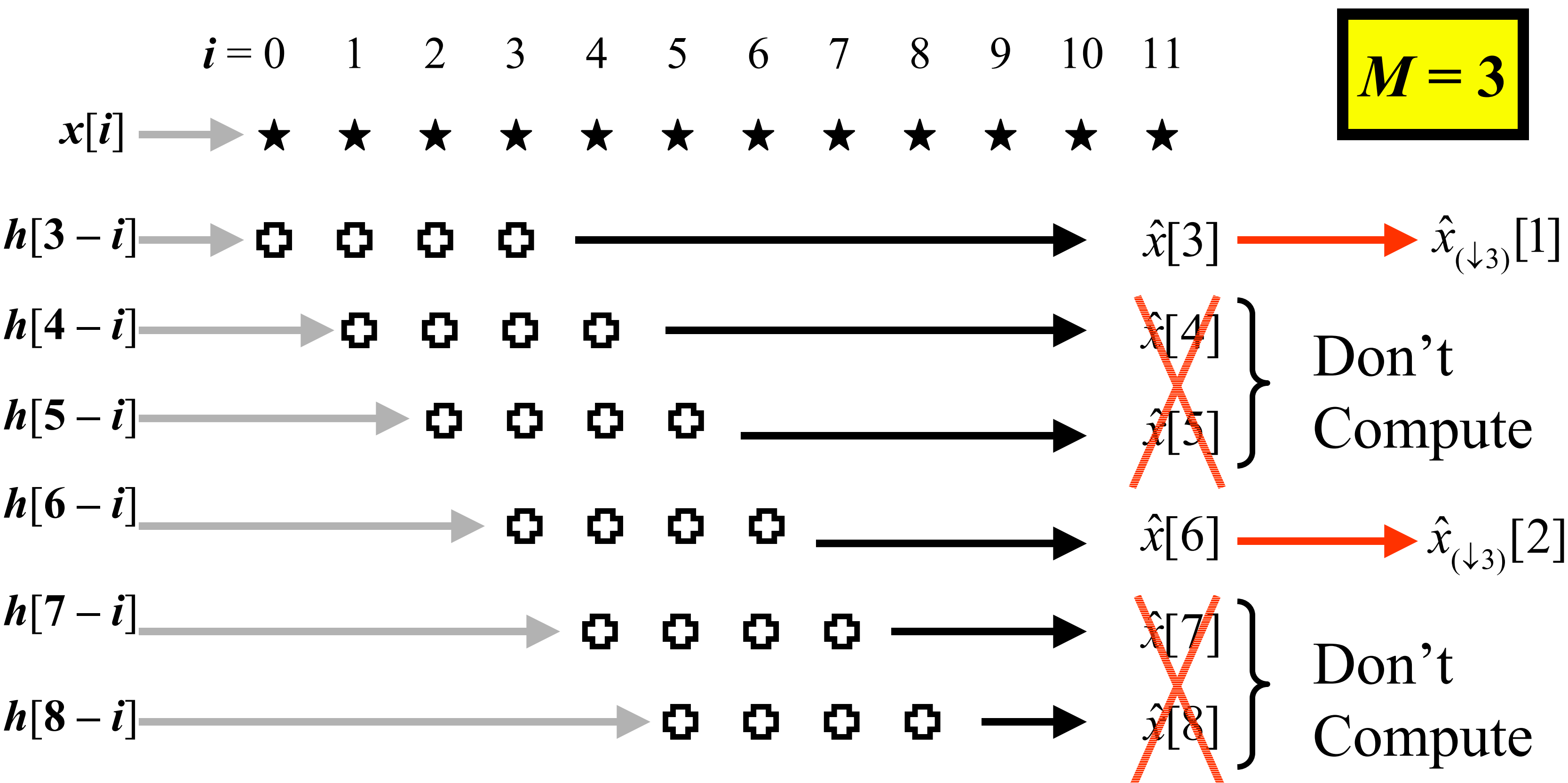
……… Of course, we **WILL** need to do the math, too, though.

Efficient FIR Filtering for Decimation

Filtering: $$\hat{x}[n] = \sum_i x[i]\, h[n-i]$$

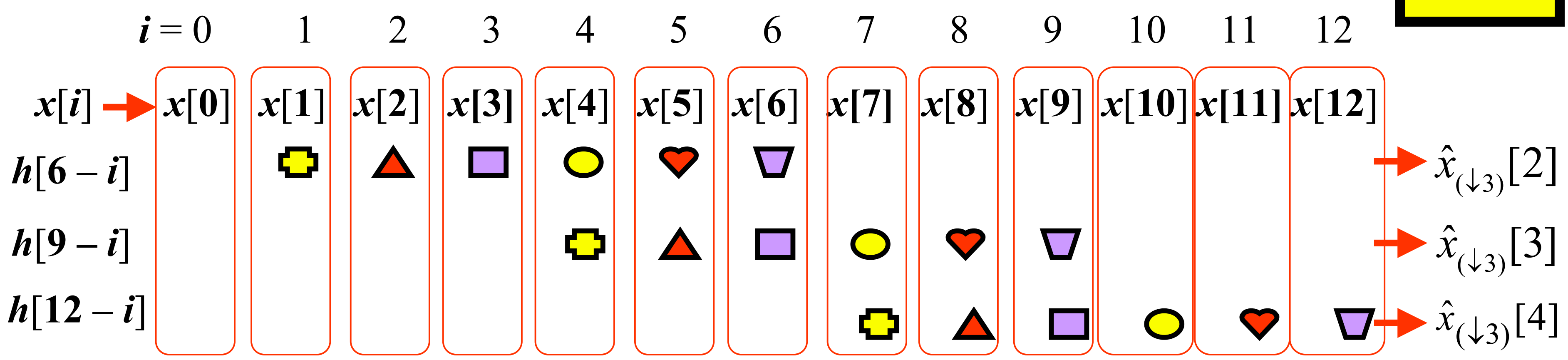
Decimation: $$\hat{x}_{(\downarrow M)}[n] = \hat{x}[nM]$$

$$= \sum_i x[i]\, h[nM - i]$$



Efficient FIR Filtering for Decimation

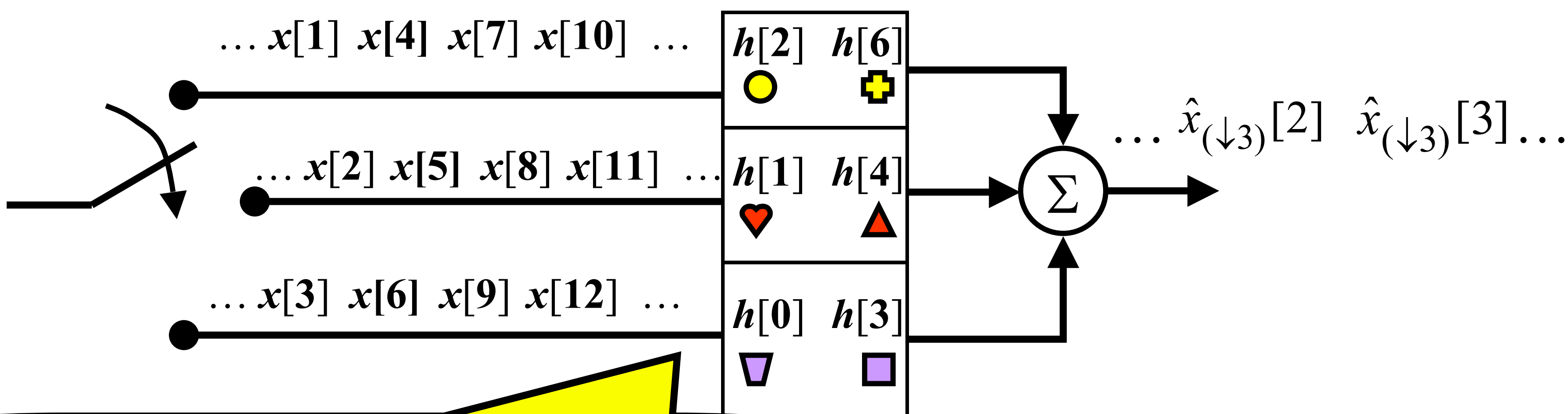
$M = 3$



$h[0]$ $h[1]$ $h[2]$ $h[3]$ $h[4]$ $h[5]$

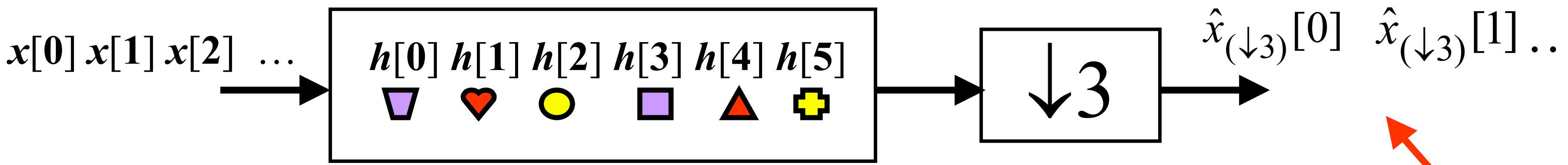
Original Filter… …gets split into M=3 subfilters:

Polyphase Form of FIR Decimation



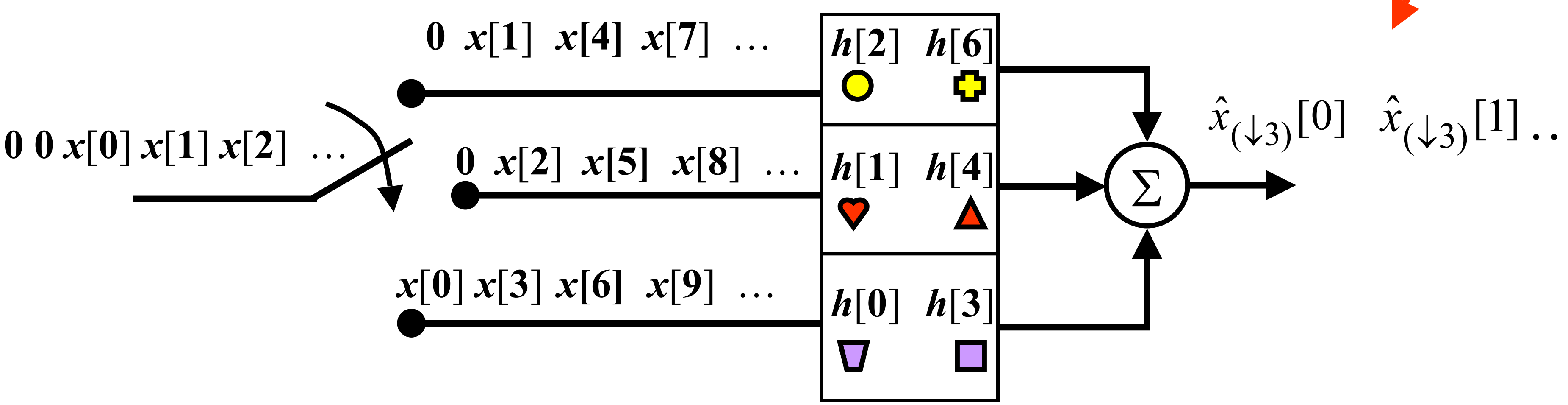
Advantage: "Decimate" then Filter

Inefficient **Direct Form of FIR Decimation**



Outputs are the Same

Efficient **Polyphase Form of FIR Decimation**



Example of Polyphase Filters for Decimation

Consider Length-10 Filter w/ M=4

i:	0	1	2	3	4	5	6	7	8	9	10	11	12
h[i]:	h[0]	h[1]	h[2]	h[3]	h[4]	h[5]	h[6]	h[7]	h[8]	h[9]	0	0	0 ….

Length of Polyphase Filters: ceil{length/M} = ceil{10/4} = 3

i':	0	1	2
$p_0[i']$:	h[0]	h[4]	h[8]
$p_1[i']$:	h[1]	h[5]	h[9]
$p_2[i']$:	h[2]	h[6]	0
$p_3[i']$:	h[3]	h[7]	0

$x_0[n]$:	x[0]	x[4]	x[8]	x[12]	x[16]	….
$x_1[n]$:	x[-1]	x[3]	x[7]	x[11]	x[15]	….
$x_2[n]$:	x[-2]	x[2]	x[6]	x[10]	x[14]	….
$x_3[n]$:	x[-3]	x[1]	x[5]	x[9]	x[13]	….

Example of Polyphase Filters for Decimation (pt. 2)

Matlab Code

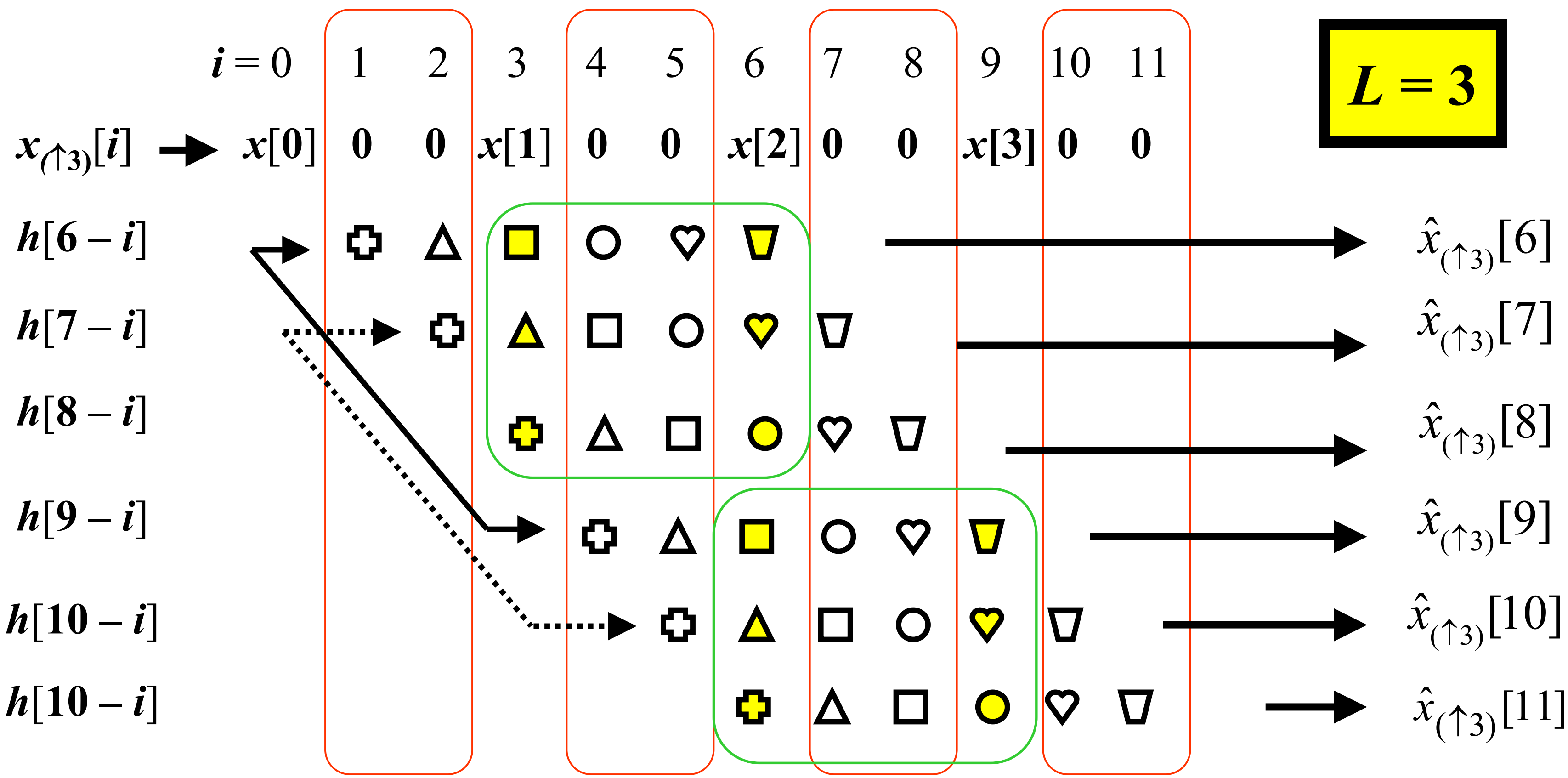
```
% Create input signal and filter
x=1:21;
h=[1 2 3 4 5 6 7 8 9 10 0 0];

% %%%%%%%   Direct Form (Inefficient)   %%%%%%%
y=filter(h,1,x);   % Compute  filter output
y_dec=y(1:4:end)  % Throw away unneeded output samples

% %%%%%%%   Polyphase Form (Efficient)   %%%%%%%
% Select polyphase filters
p0=h(1:4:end)
p1=h(2:4:end)
p2=h(3:4:end)
p3=h(4:4:end)
% Select polyphase signals
x0=x(1:4:end)
x1=[0 x(4:4:end)]
x2=[0 x(3:4:end)]
x3=[0 x(2:4:end)]
% filter each polyphase component and add together
y_poly_dec=filter(p0,1,x0)+filter(p1,1,x1)+filter(p2,1,x2)+filter(p3,1,x3)
```

Pad zeros to make length equal to integer multiple of *M*

Put a zero in front to provide the $x[-3]$, $x[-2]$, and $x[-1]$ terms

Efficient FIR Filtering for Interpolation

Interpolation : $$\hat{x}_{(\uparrow L)}[n] = \sum_i x_{(\uparrow L)}[i]\, h[n-i]$$



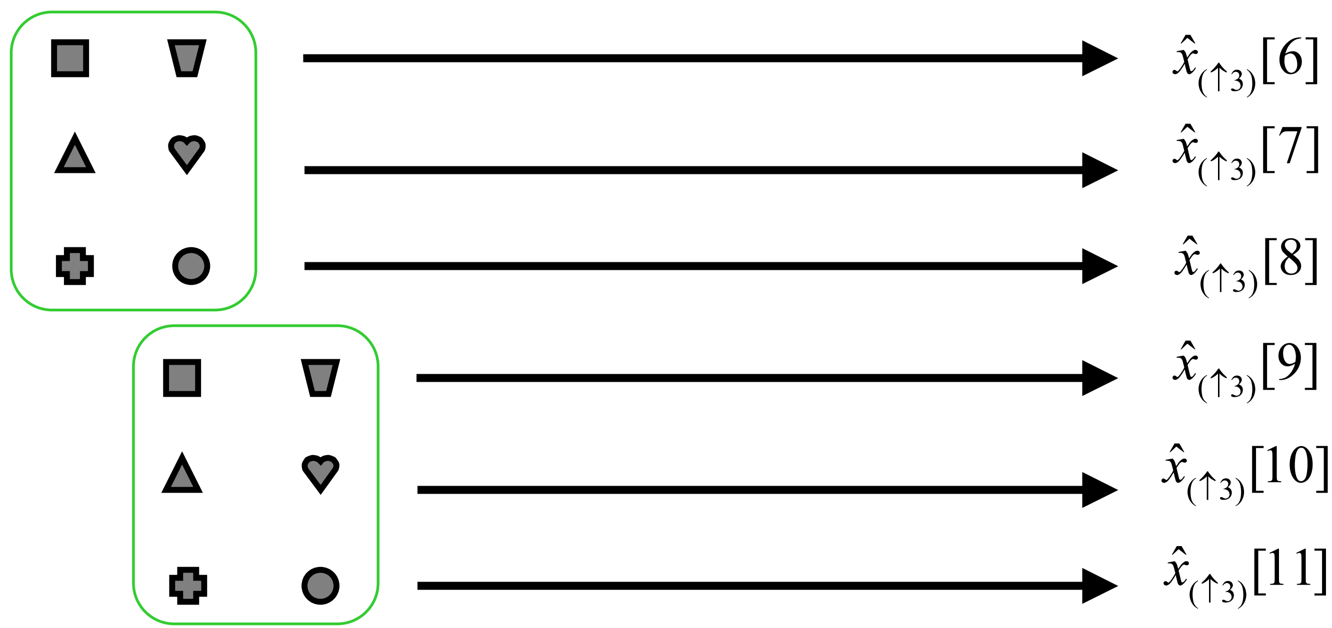
Efficient FIR Filtering for Interpolation

Interpolation : $$\hat{x}_{(\uparrow L)}[n] = \sum_i x[i]\, h[n - Li]$$

$L = 3$

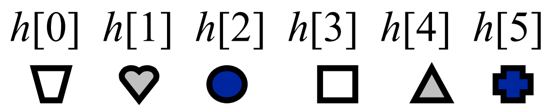
$\boldsymbol{i} = 0$ 1   2   3

$\boldsymbol{x[i]} \rightarrow \boldsymbol{x[0]}$ $\boldsymbol{x[1]}$ $\boldsymbol{x[2]}$ $\boldsymbol{x[3]}$



Efficient FIR Filtering for Interpolation

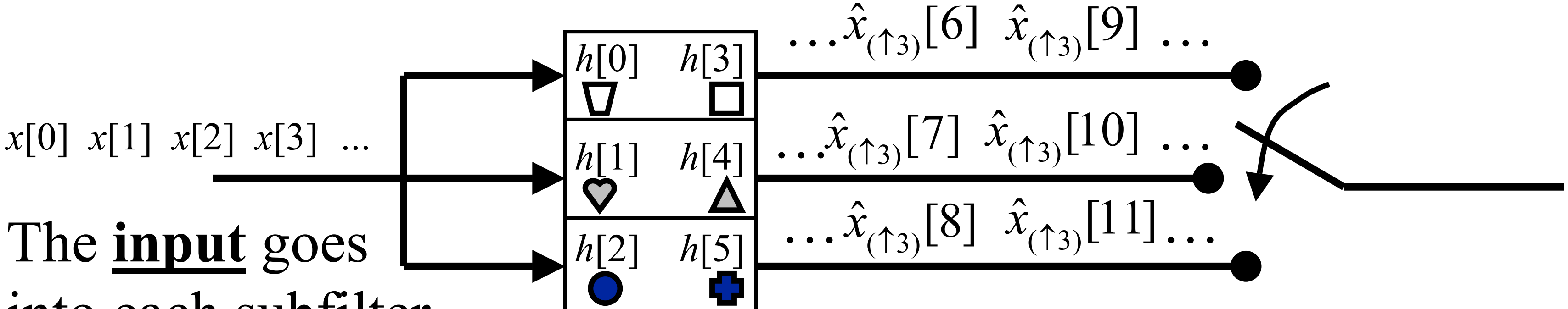
Original Filter…



$L = 3$

… gets split into $L = 3$ subfilters:

Polyphase Form of FIR Interpolation



The **input** goes into each subfilter

Advantage
Filter then
Interpolate

The **output** comes from alternating between the subfilter outputs

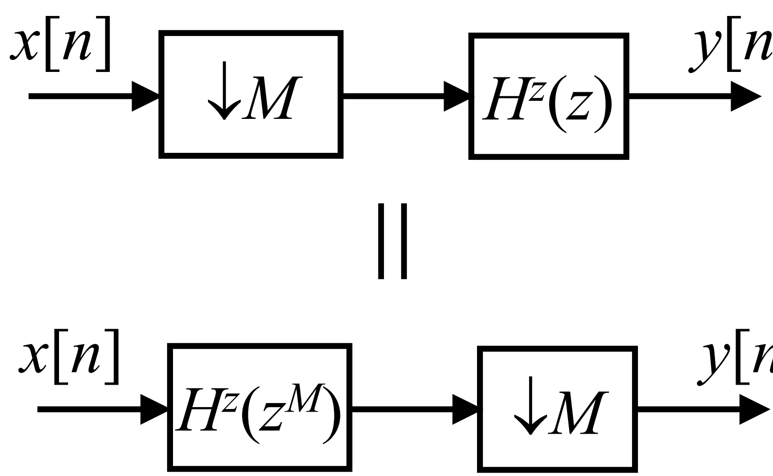
12.4.1 Multirate Identities

These provide analysis "tricks" useful when dealing with mathematical analysis of multirate systems.

The question in general is: How can we interchange the order of filtering w/ decimation/expansion?

Decimation Identity

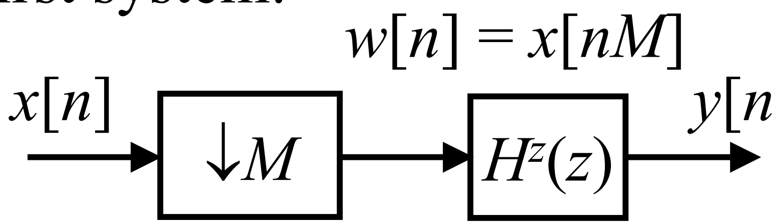
This identity asserts equality between the following 2 systems:



Can prove this either in the Time-Domain or Z-Domain

TD Proof of Decimation Identity

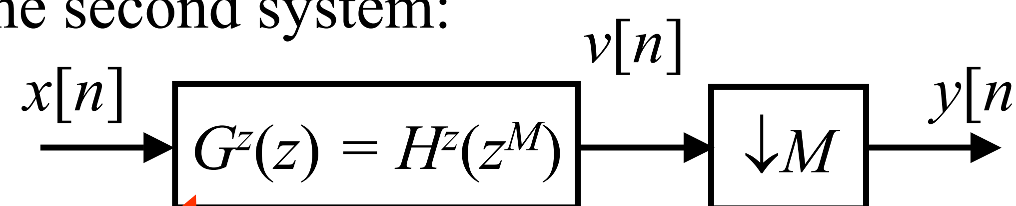
For the first system:



$$y[n] = w[n] * h[n] = \sum_k h[k]w[n-k]$$

$$= \sum_k h[k]x[(n-k)M]$$

For the second system:



$$g[n] = h_{(\uparrow M)}[n]$$

By Eq. (12.15)

$$= \begin{cases} h[n/M], & \text{if } n/M = \text{integer} \\ 0, & \text{otherwise} \end{cases} \qquad (\bigstar)$$

TD Proof of Decimation Identity (cont.)

Thus…

$$v[n] = x[n] * g[n] = \sum_{l} g[l]x[n-l]$$

$$= \sum_{k} h[k]x[n-kM]$$

Use (★)

Then…

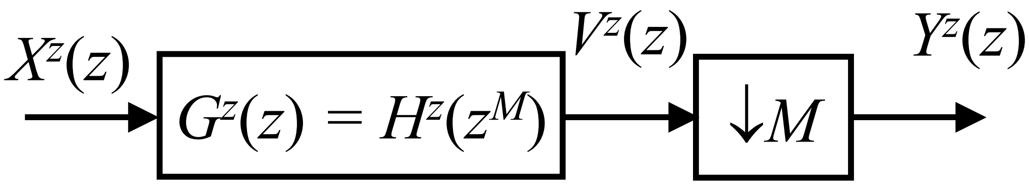
$$y[n] = v[nM]$$

$$= \sum_{k} h[k]x[(n-k)M]$$

$\boxed{\text{Same as for System \#1} \rightarrow \text{Proved!!!}}$

ZD Proof of Decimation Identity

For the second system:



where… $V^z(z) = X^z(z) H^z(z^M)$ (★★)

But… $Y^z(z) = \{V^z(z)\}_{(\downarrow M)}$

By ZT Result for Decimation

$$= \frac{1}{M} \sum_{m=0}^{M-1} V^z(z^{1/M} W_M^{-m})$$

Use (★★)

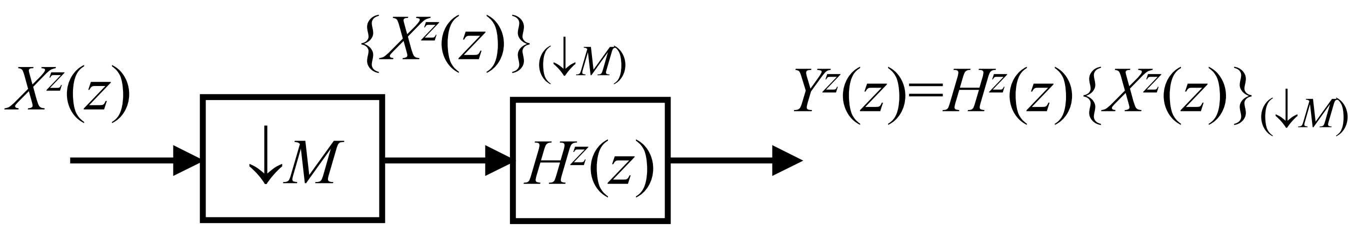
$$= \frac{1}{M} \sum_{m=0}^{M-1} X^z(z^{1/M} W_M^{-m}) H^z((z^{1/M} W_M^{-m})^M)$$

Now… $(z^{1/M} W_M^{-m})^M = z \underbrace{W_M^{-mM}}_{e^{j2\pi}=1} = z$

ZD Proof of Decimation Identity (cont.)

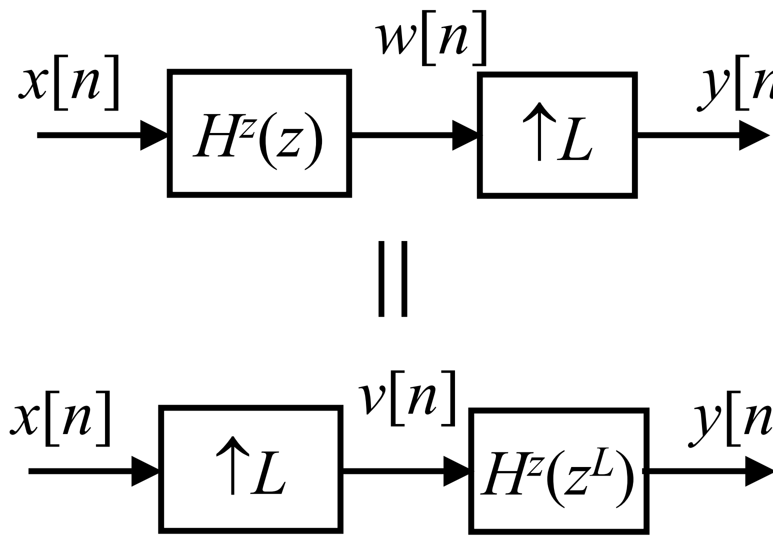
$$
\begin{aligned}
Y^z(z) &= \frac{1}{M}\sum_{m=0}^{M-1} X^z(z^{1/M}W_M^{-m})H^z(z) \\
&= H^z(z)\left[\frac{1}{M}\sum_{m=0}^{M-1} X^z(z^{1/M}W_M^{-m})\right] \\
&= H^z(z)\left\{X^z(z)\right\}_{(\downarrow M)}
\end{aligned}
$$

Which is clearly the same thing that the first system gives:



Expansion Identity

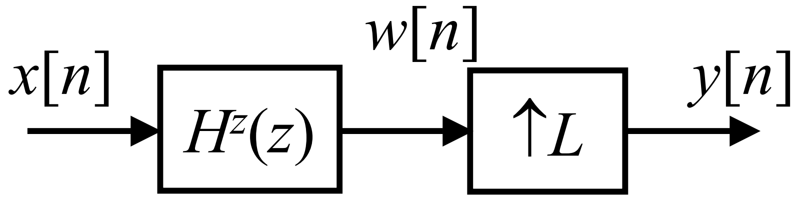
This identity asserts equality between the following 2 systems:



Will give only Z-Domain proof here.

ZD Proof of Expansion Identity

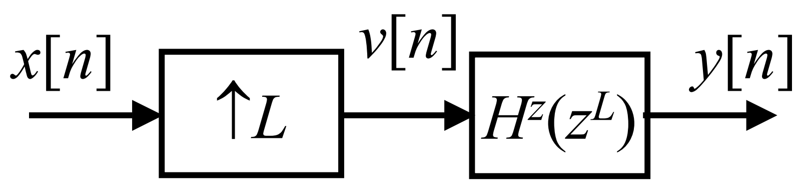
First system gives:



$$W^z(z) = X^z(z)H^z(z)$$

Then…
$$Y^z(z) = W^z_{(\uparrow L)}(z) = W^z(z^L)$$
$$= X^z(z^L)H^z(z^L)$$

Second system gives:



$$V^z(z) = X^z_{(\uparrow L)}(z) = X^z(z^L)$$

Then…
$$Y^z(z) = V^z(z)H^z(z^L)$$
$$= X^z(z^L)H^z(z^L)$$

Same!

12.4.2 Polyphase Representation of Decimation

Now we re-visit this topic and do it mathematically…

Basic Math Idea: Re-write convolution sum's index & manipulate to get "parallel" filters:

Recall Decimation: $x[n] \rightarrow \boxed{H^z(z)} \rightarrow \boxed{\downarrow M} \rightarrow y[n]$

Output given by (12.17) as… $$y[n] = \sum_i h[i]x[nM - i] \qquad (\star\star\star)$$

Write sum's index in "block form" – a common "trick":

$$i = i'M + m \qquad \begin{cases} i' = \text{integer} \\ 0 \le \text{m} \le M - 1 \end{cases}$$

M = Block Size

Counts Blocks

Counts Samples Inside a Block

12.4.2 Polyphase Rep of Dec (cont.)

Block-Based Indexing:

$$i = i'M + m \quad \begin{cases} i' = \text{integer} \\ 0 \le \mathrm{m} \le M-1 \end{cases}$$

Forward Indexing

$i' \backslash m$	0	1	2	$\cdots$	$M-1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	$-M$	$-M+1$	$\cdots$	-2	-1
0	0	1	2	$\cdots$	$M-1$
1	M	$M+1$	$M+2$	$\cdots$	$2M-1$
2	$2M$	$2M+1$	$2M+2$	$\cdots$	$3M-1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Each row is indexed forward

12.4.2 Polyphase Rep of Dec (cont.)

Use Block Indexing in (★★★):

$$y[n] = \sum_i h[i]x[nM - i]$$

$$= \sum_{i'} \sum_{m=0}^{M-1} h[i'M + m]x[\underbrace{nM - i'M - m}_{(n-i')M-m}]$$

- Sum up inside each block
- Sum up all Block Results

$$= \sum_{m=0}^{M-1} \underbrace{\sum_{i'} h[i'M + m]x[(n - i')M - m]}$$ (★★★★)

Sum all elements in the m^{th} position of each block

12.4.2 Polyphase Rep of Dec (cont.)

Now, let's interpret this:

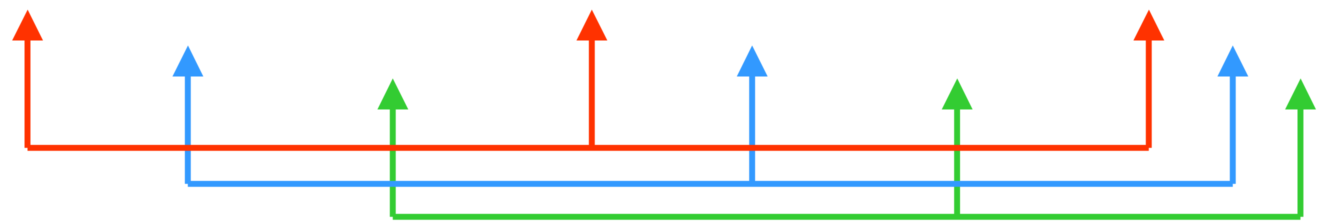
Define for each m, $0 \leq m \leq M$-1

$$p_m[i'] = h[i'M + m]$$

m^{th} Polyphase Component of h[n]

Example

n:	0	1	2	3	4	5	6		
$h[n]$:	1.2	4	0.5	7	1	1.7	2	0	0…



M=3

$$p_0[i'] = \{1.2, \quad 7, \quad 2\}$$

$$p_1[i'] = \{4, \quad 1, \quad 0\}$$

$$p_2[i'] = \{0.5, \quad 1.7, \quad 0\}$$

Each one is a decimated version of $h[n]$ & the versions are staggered

< See Fig. 12.15>

Fig. 12.15 from Porat's Book

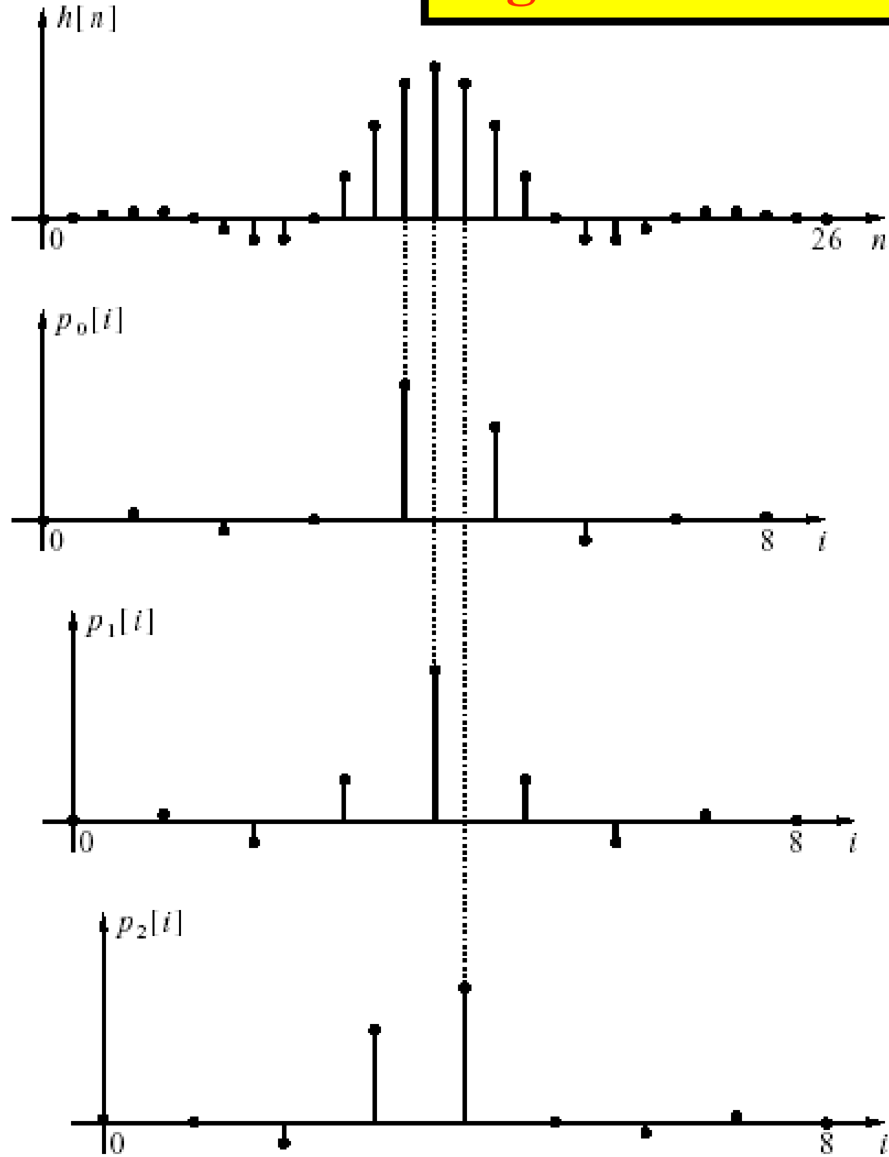


Figure 12.15 The polyphase components of an FIR filter, illustrated for $N = 26$ and $M = 3$.

12.4.2 Polyphase Rep of Dec (cont.)

What have we done?

Split up $h[n]$ into M subsequences – where the m^{th} subsequence is a decimated-by-M version of $h[n + m]$

Why the name "Polyphase"?

Recall: Time-Shift in TD $\leftrightarrow$ Phase-Shift in FD

$$h[n+m] \leftrightarrow e^{j\theta m} H^{\mathrm{f}}(\theta)$$

$\boxed{\rightarrow \text{“Poly}\underline{\text{phase}}\text{”}}$

12.4.2 Polyphase Rep of Dec (cont.)

Now… let's chop up the input <u>similarly</u>:

$$u_m[n] = x[nM - m]$$

Backward Indexing

$n \backslash m$	0	1	2	$\cdots$	$M-1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	$-M$	$-M-1$	$\cdots$		$-2M+1$
0	0	-1	-2	$\cdots$	$-M+1$
1	M	$M-1$	$M-2$	$\cdots$	1
2	$2M$	$\cdots$		$M+2$	$M+1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

<u>Differs From Before</u>: Each row is indexed backward

12.4.2 Polyphase Rep of Dec (cont.)

Now… back to the mathematical development.

Putting these re-indexed versions into (★★★★):

$$y[n] = \sum_{m=0}^{M-1} \sum_{i'} h[i'M + m] x[(n - i')M - m]$$

$$p_m[i'] = h[i'M + m]$$

$$u_m[n] = x[nM - m]$$

$$y[n] = \sum_{m=0}^{M-1} \left[\sum_{i'} p_m[i'] u_m[n - i'] \right]$$

$$= \sum_{m=0}^{M-1} \{ p_m * u_m \}[n]$$

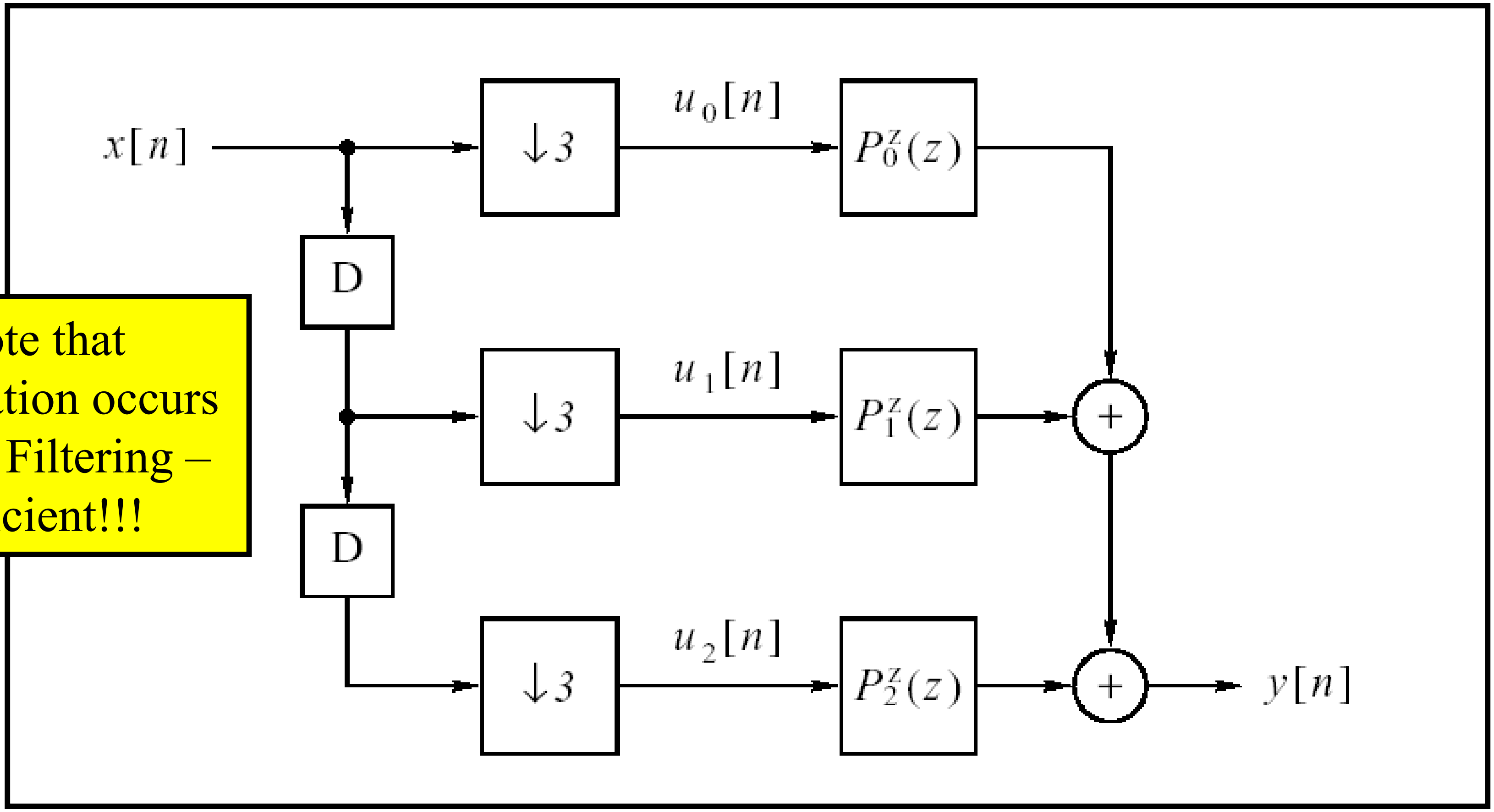
To Implement Polyphase Decimation

- Chop up filter into M sub-filters
- Chop up signal into M sub-signals
- Filter each sub-signal w/ a sub-filter
- Add outputs point-by-point

12.4.2 Polyphase Rep of Dec (cont.)

Two equivalent ways to think of this:

First Way (shown for *M*=3):

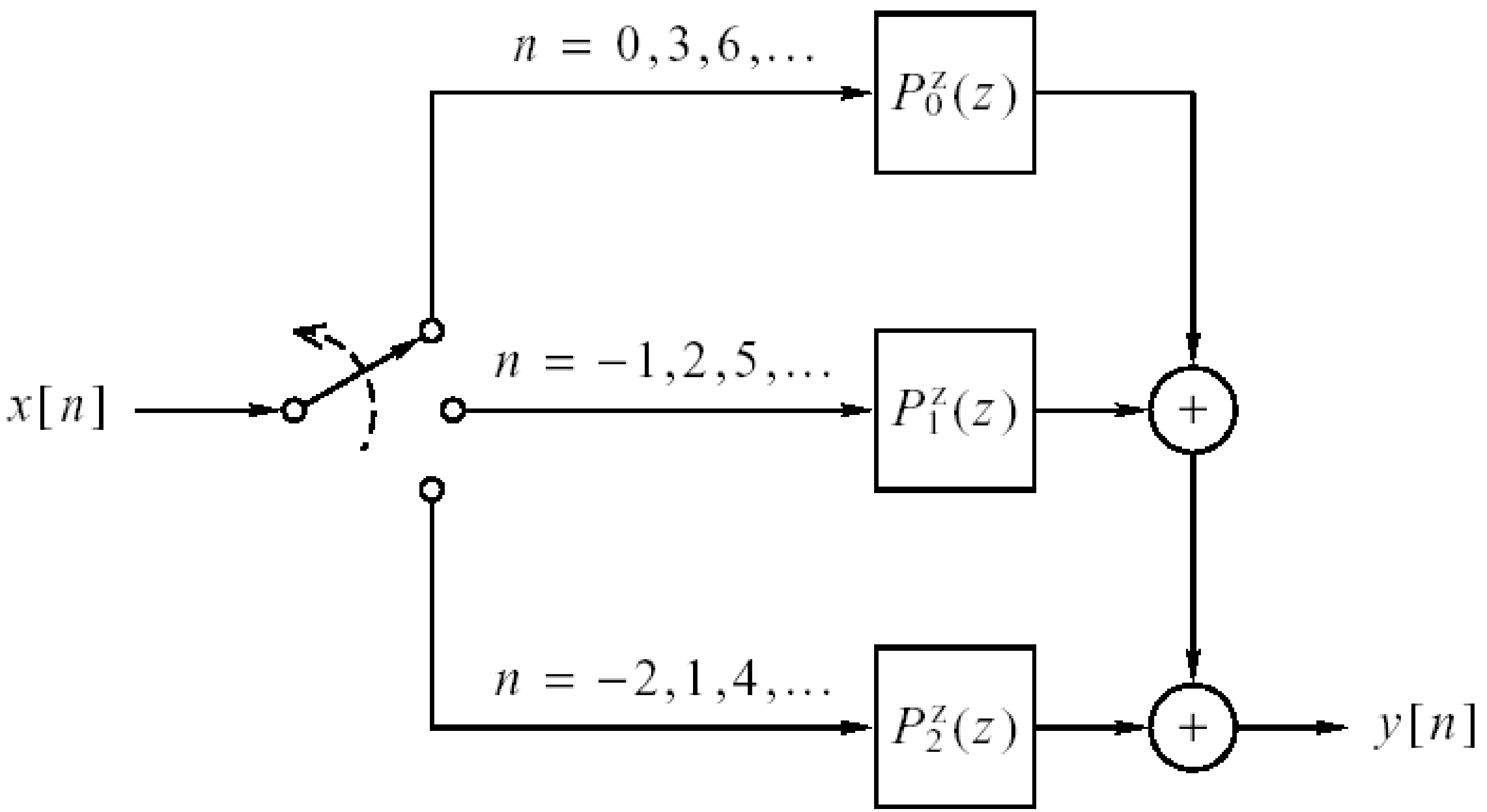


Note that Decimation occurs Before Filtering – Efficient!!!

<This is Fig. 12.16 from Porat's Book>

12.4.2 Polyphase Rep of Dec (cont.)

Second Way to View It (shown for *M*=3):



<This is Fig. 12.17 from Porat's Book>

12.4.2 Polyphase Rep of Dec (cont.)

Now we re-analyze this set-up, but in the Z-Domain….

Why? ….It provides further analysis insight.

Z-Domain results often provide insight into how to:

- Derive other results
- Design Polyphase Filters
- Etc.

12.4.2 Polyphase Rep of Dec (cont.)

First…. some time-domain trickery:

How do we get back $h[n]$ from the $p_m[n]$???

1. Insert M-1 zeros between each sample
2. "Line them up" using delays
3. Add them up

Expansion!

Recall Example:

$$\begin{aligned} p_0[i'] &= \{1.2, \quad 7, \quad 2\} \\ p_1[i'] &= \{4, \quad 1, \quad 0\} \\ p_0[i'] &= \{0.5, \quad 1.7, \quad 0\} \end{aligned}$$

$$\begin{array}{l} \{1.2, \; 0, \; 0, \; 7, \; 0, \; 0, \; 2, \; 0, \; 0\} \\ \{\;4, \; 0, \; 0, \; 1, \; 0, \; 0, \; 0, \; 0, \; 0\} \\ \{0.5, \; 0, \; 0, \; 1.7, \; 0, \; 0, \; 0, \; 0, \; 0\} \end{array}$$

$$\begin{array}{l} \{1.2, \; 0, \; 0, \; 7, \; 0, \; 0, \; 2, \; 0, \; 0\} \\ \{\;0, \; 4, \; 0, \; 0, \; 1, \; 0, \; 0, \; 0, \; 0\} \\ \{\;0, \; 0, \; 0.5, \; 0, \; 0, \; 1.7, \; 0, \; 0, \; 0\} \end{array}$$

$$h[n] = \{1.2, \quad 4, \quad 0.5, \quad 7, \quad 1, \quad 1.7, \quad 2, \quad 0, \quad 0\}$$

12.4.2 Polyphase Rep of Dec (cont.)

Thus….

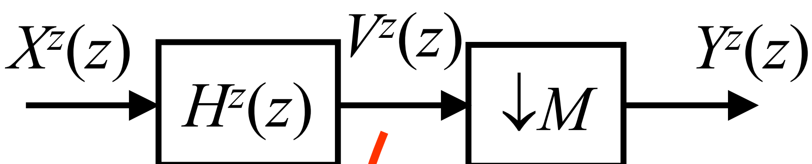
$$h[n] = \sum_{m=0}^{M-1} \{p_{m_{(\uparrow M)}}\}[n-m]$$

So…. in Z-Domain we have:

Delay Expand

$$H^z(z) = \sum_{m=0}^{M-1} z^{-m} P_m^z(z^M)$$

Now… filter/decimate looks like this:

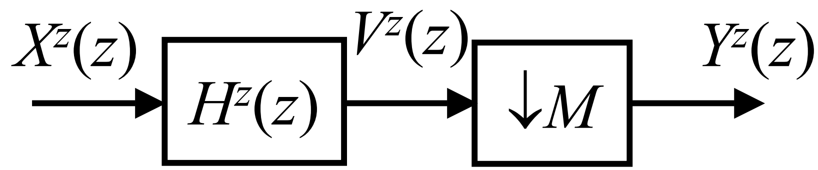


$$V^z(z) = X^z(z)H^z(z)$$

$$= \sum_{m=0}^{M-1} z^{-m} P_m^z(z^M) X^z(z)$$

12.4.2 Polyphase Rep of Dec (cont.)

… and after ↓M we get:



$$Y^z(z) = \{V^z(z)\}_{(\downarrow M)}$$

$$= \sum_{m=0}^{M-1} \underbrace{\{z^{-m} P_m^z(z^M) X^z(z)\}_{(\downarrow M)}}_{= P_m^z(z)\{z^{-m} X^z(z)\}_{(\downarrow M)}}$$

By the "Decimation Identity"

$$= \sum_{m=0}^{M-1} P_m^z(z) \underbrace{\{z^{-m} X^z(z)\}_{(\downarrow M)}}_{U_m^z(z)}$$

<u>By Definition</u> Signal's Polyphase Components

$$= \sum_{m=0}^{M-1} P_m^z(z) U_m^z(z)$$

….which is the Z-Domain Description of the polyphase decimation structure. We have now developed <u>two different derivations</u> of the polyphase structure.

12.4.3 Polyphase Rep of Expansion

Recall Expansion:

$$x[n] \longrightarrow \boxed{\uparrow L} \longrightarrow \boxed{H^z(z)} \longrightarrow y[n]$$

Output given by (12.19) as… $\displaystyle y[n] = \sum_i x[i]h[n - Li]$

Re-Index using: $n = n'L + \underbrace{(L-1) - l}_{\text{"backwards"}}$ $\begin{cases} n' & \text{integer} \\ 0 \le 1 \le L - 1 \end{cases}$

n' = Block Index
l = In-Block Index (indexes backward through block)

12.4.3 Polyphase Rep of Exp (cont.)

$$n = n'L + \underbrace{(L-1)-l}_{\text{"backwards"}} \qquad \begin{cases} n' & \text{integer} \\ 0 \le l \le L-1 \end{cases}$$

$n' \backslash l$	0	1	2	$\cdots$	$L-1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	-1	-2	$\cdots$		$-L$
0	$L-1$	$L-2$	$L-3$	$\cdots$	0
1	$2L-1$	$2L-2$	$2L-3$	$\cdots$	L
2	$3L-1$	$3L-2$	$3L-3$	$\cdots$	$2L$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Expansion Re-Index Table

12.4.3 Polyphase Rep of Exp (cont.)

Using this re-indexing gives…

$$y[n] = \sum_i x[i]h[n - Li]$$

$$y[\underbrace{n'L + (L-1) - l}_{n}] = \sum_i x[i]h[\underbrace{n'L + (L-1) - l}_{n} - Li]$$

$$= \sum_i x[i]h[(n' - i)L + (L-1) - l]$$

For each l such that $0 \leq l \leq L - 1$ we <u>define</u>:

$$q_l[n'] = h[n'L + (L-1) - l]$$

$$v_l[n'] = y[n'L + (L-1) - l]$$

$$\Rightarrow \quad v_l[n'] = \{x * q_l\}[n']$$

… for each l, this indexing just reads down a <u>column</u> of the "Expansion Re-Index Table"

12.4.3 Polyphase Rep of Exp (cont.)

To see this indexing structure, look at an example with $L = 3$:

$n' \backslash l$	0	1	2
	$v_0[n']$	$v_1[n']$	$v_2[n']$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	$y[-1]$	$y[-2]$	$y[-3]$
0	$y[2]$	$y[1]$	$y[0]$
1	$y[5]$	$y[4]$	$y[3]$
2	$y[8]$	$y[7]$	$y[6]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

12.4.3 Polyphase Rep of Exp (cont.)

Now… how do we get $y[n]$ from the v_l's??

If we interpolate each v_l sequence we get ($L = 3$)….

$$\cdots \quad y[-3] \quad 0 \quad 0 \quad y[0] \quad 0 \quad 0 \quad y[3] \quad 0 \quad 0 \quad y[6] \quad 0 \quad 0 \quad \cdots$$

$$\cdots \quad y[-2] \quad 0 \quad 0 \quad y[1] \quad 0 \quad 0 \quad y[4] \quad 0 \quad 0 \quad y[7] \quad 0 \quad 0 \quad \cdots$$

$$\cdots \quad y[-1] \quad 0 \quad 0 \quad y[2] \quad 0 \quad 0 \quad y[5] \quad 0 \quad 0 \quad y[8] \quad 0 \quad 0 \quad \cdots$$

Now delay these interpolated sequences…

$$\begin{array}{cccccccccccccc}
\cdots & y[-3] & 0 & 0 & y[0] & 0 & 0 & y[3] & 0 & 0 & y[6] & 0 & 0 & \cdots \\
\cdots & 0 & y[-2] & 0 & 0 & y[1] & 0 & 0 & y[4] & 0 & 0 & y[7] & 0 & \cdots \\
\cdots & 0 & 0 & y[-1] & 0 & 0 & y[2] & 0 & 0 & y[5] & 0 & 0 & y[8] & \cdots \\
\hline
\cdots & y[-3] & y[-2] & y[-1] & y[0] & y[1] & y[2] & y[3] & y[4] & y[5] & y[6] & y[7] & y[8] & \cdots
\end{array}$$

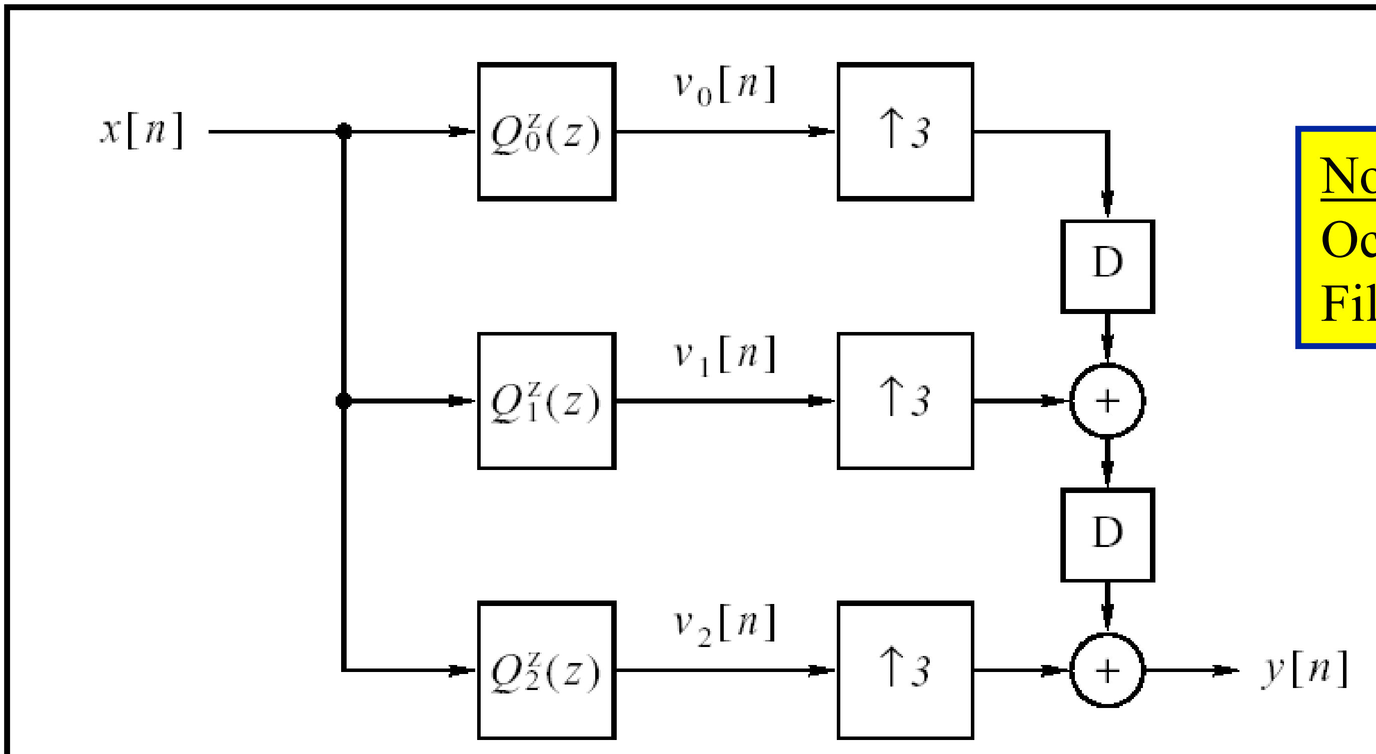
To get $y[n]$: add up the delayed, interpolated components!!

12.4.3 Polyphase Rep of Exp (cont.)

From this we see that we can write…

$$y[n] = \sum_{l=0}^{L-1} \{v_l\}_{(\uparrow L)}[n-(L-1)+l] \qquad \text{Recall: } v_l[n'] = \{x * q_l\}[n']$$

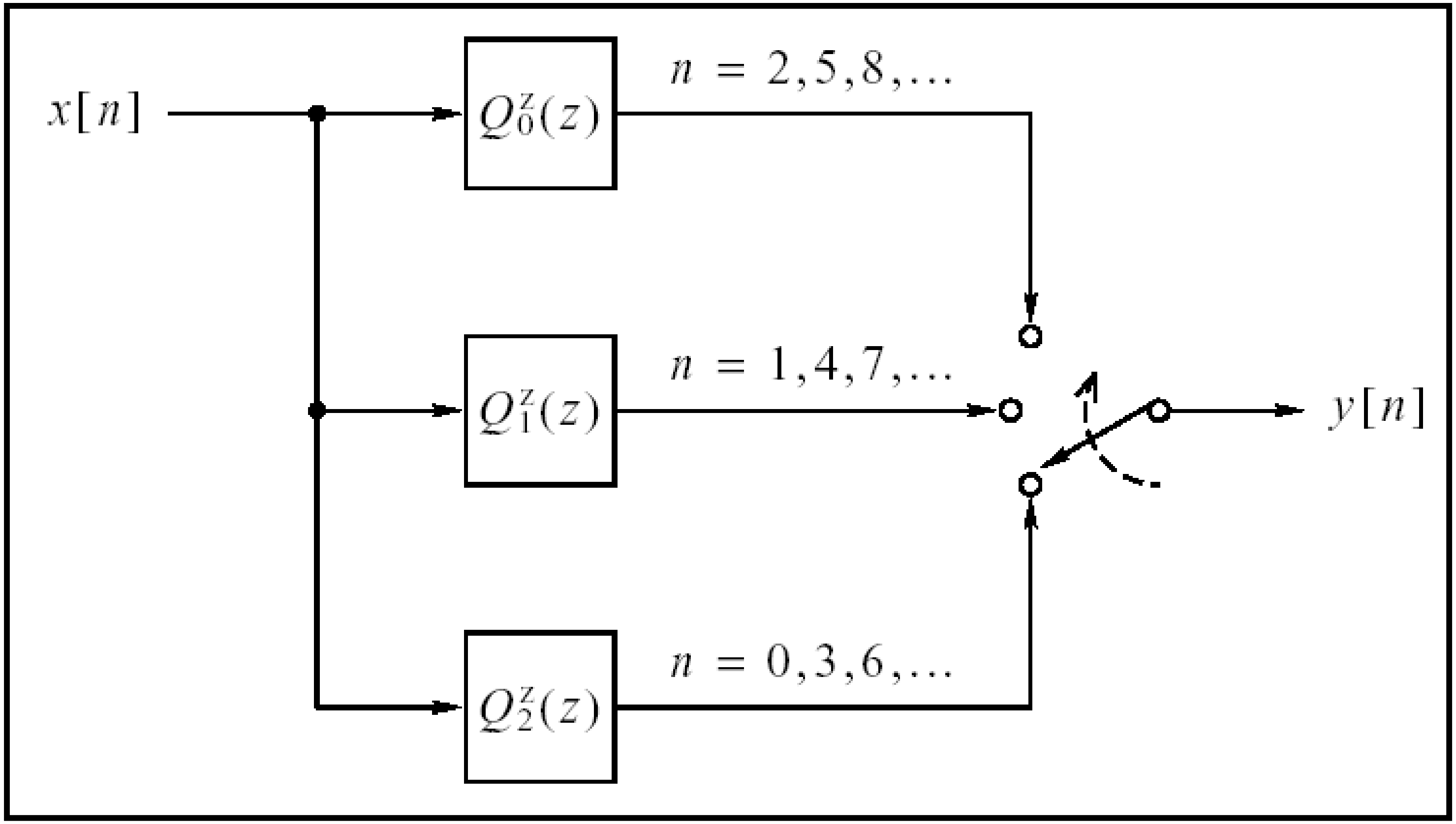
This leads to the following polyphase implementation for expansion:



<u>Note</u>: Expansion Occurs After Filtering – Efficient!!

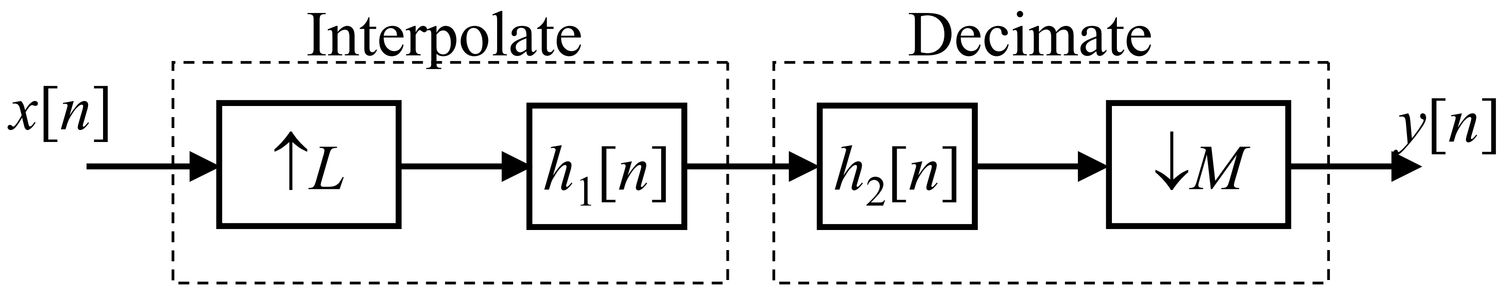
12.4.3 Polyphase Rep of Exp (cont.)

An equivalent alternate form of this processing is…

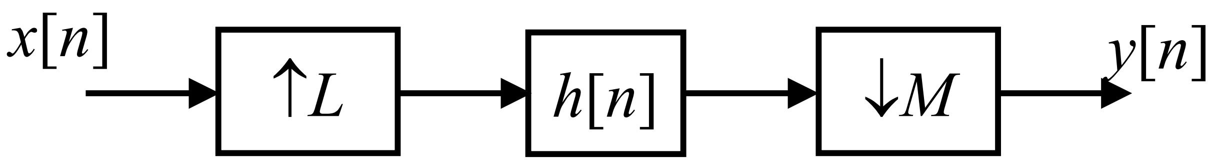


<u>Skip 12.4.4</u> Shows how to do polyphase method for <u>rational</u> rate change of L/M

But briefly… to change the rate by factor of L/M



which is equivalent to…



Q: How to implement this efficiently using polyphase ideas?
If interested: see Ch.3 of Oppenheim & Lim (on reserve)