# 11.10.1 Uniform DFT Filter Banks

# Uniform DFT Filter Banks

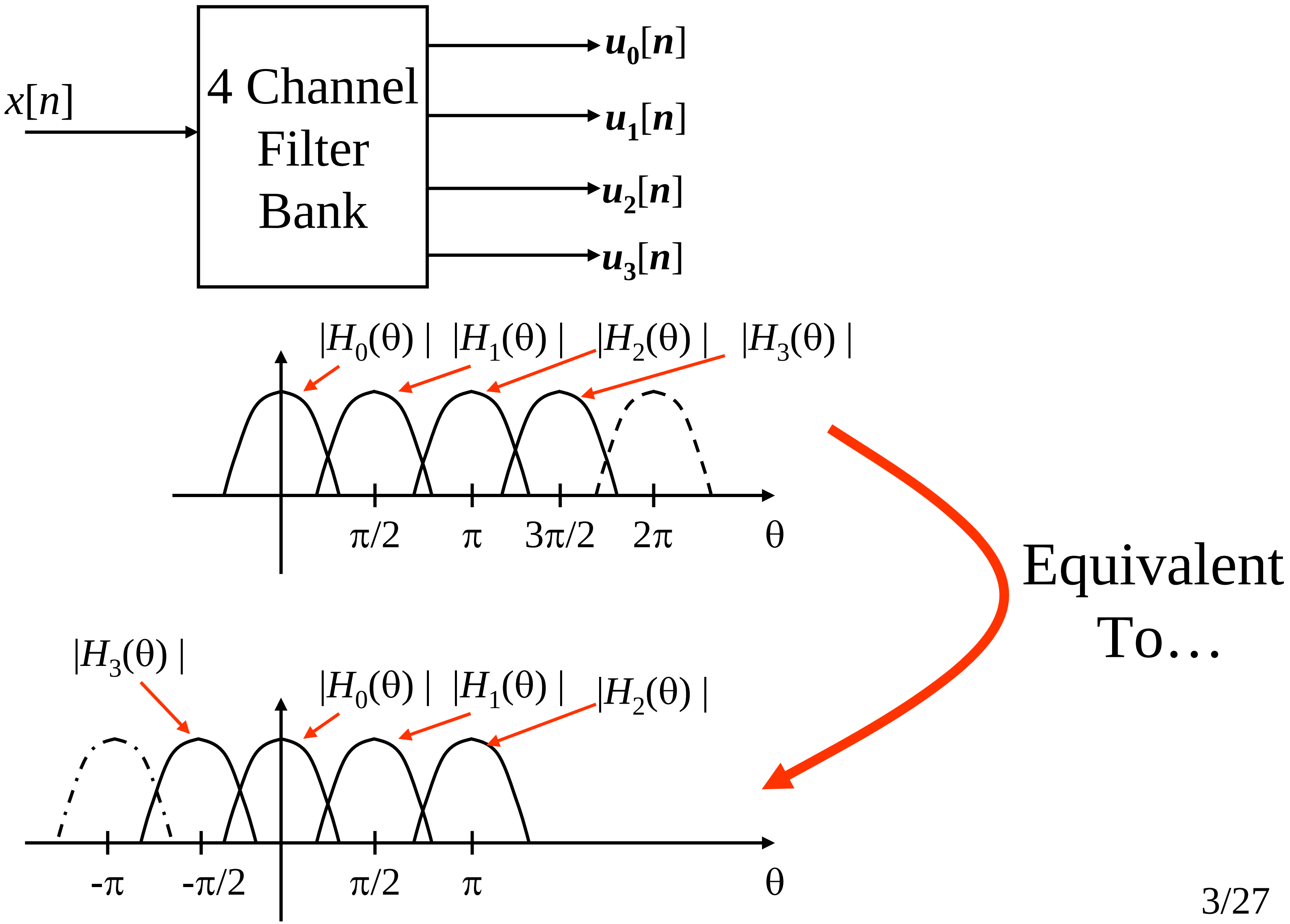
We'll look at 5 versions of DFT-based filter banks – all but the last two have serious limitations and aren't practical. But… they give a nice transition to the last two versions – which ARE useful and practical methods.

| | | | |
|---|---|---|---|
| **Version #1** (Not in P&M) | Undecimated | Rect. Window (Filter Size = # Channels) | Sliding DFT |
| **Version #2** (Not in P&M) | Decimated | Rect. Window (Filter Size = # Channels) | Sliding DFT |
| **Version #3** (Not in P&M) | Decimated | Non-Rect Window (Filter Size = # Channels) | Sliding DFT |
| **Version #4** (Not in P&M) | Decimated | Arbitrary Window (Filter Size Arbitrary) | Sliding DFT |
| **Version #5** (11.10.1) | Decimated | Polyphase Filter (Filter Size Arbitrary) | DFT |

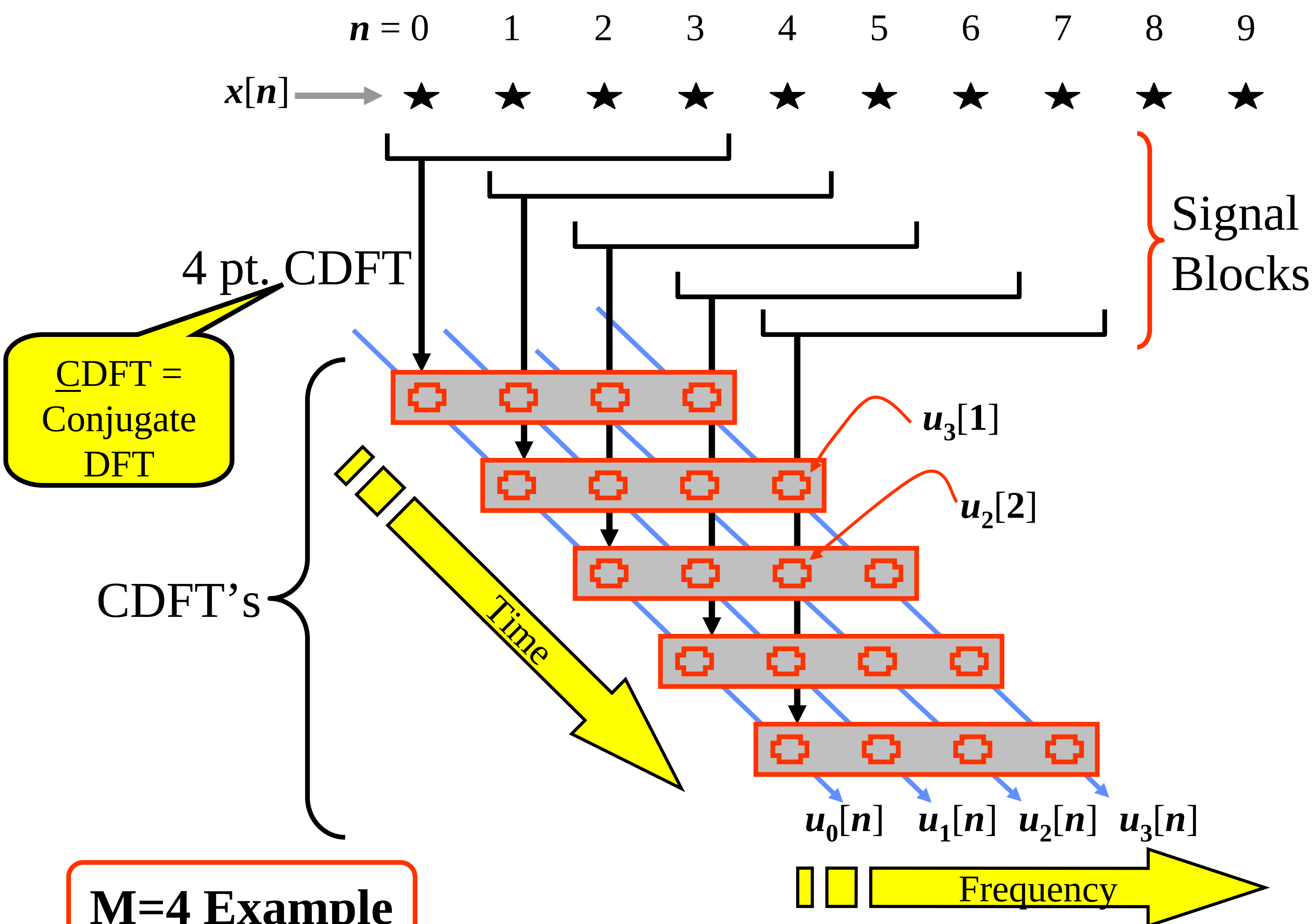
**Only Versions 4 & 5 are Practical Methods**

# Setting for Versions 1, 2, & 3

We will illustrate with a four channel case:

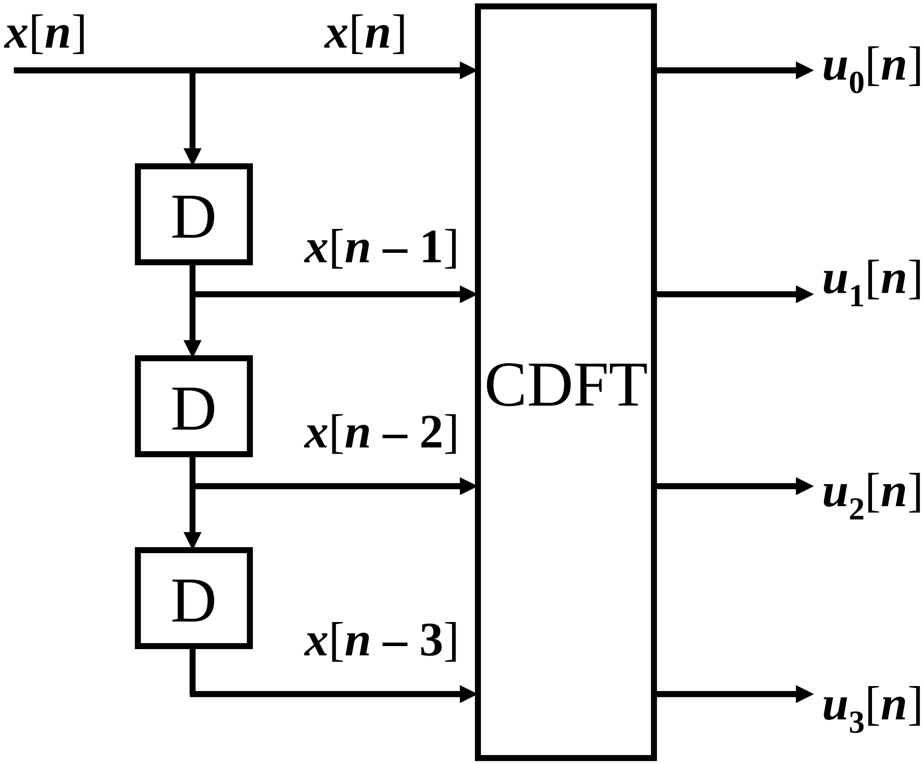


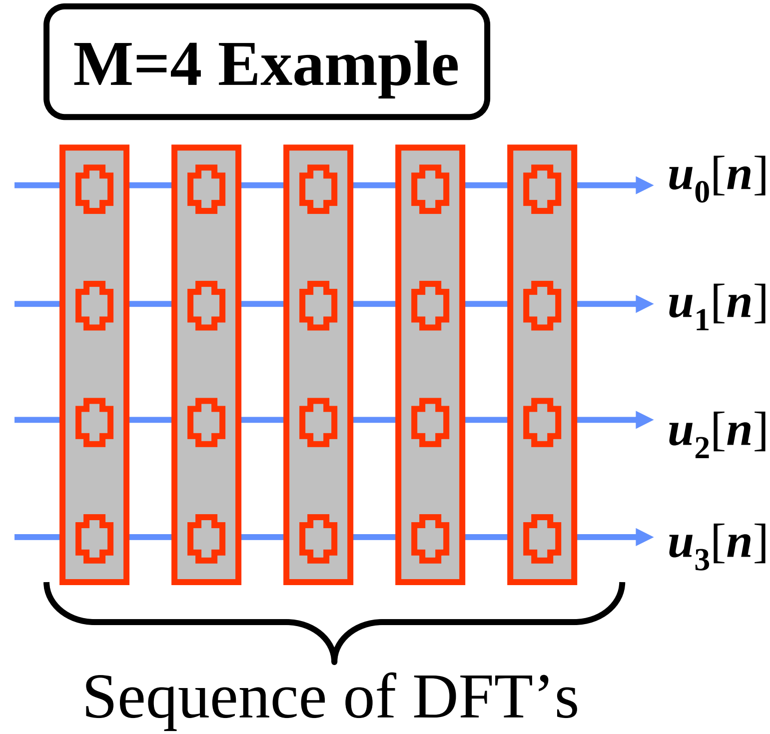
# Version #1: Sliding DFT Filter Bank




M=4 Example

# Different View of Version #1




## Math View

$$u_m[n] = \sum_{i=0}^{M-1} x[n-i] W_M^{im} \qquad m = 0, 1, \ldots, M-1$$

Fix $n$, Then Compute M-pt. CDFT

Note: Conjugate DFT Form

# Math Shows this DOES give Filter Bank

Output of this structure is:

$$u_m[n] = \sum_{i=0}^{M-1} x[n-i] W_M^{im}$$

$$= \{x * g_m\}[n] \qquad \text{where } \; g_m[n] = W_M^{nm} \quad 0 \le n \le M-1$$

Thus, the $m^{th}$ output signal is the linear convolution of the input signal with the impulse response $g_m[n]$.

Q: What is the $m^{th}$ filter's Transfer Function?

$$G_m^z(z) = \sum_{n=0}^{M-1} W_M^{nm} z^{-n}$$

$$= \frac{1-(zW_M^{-m})^{-M}}{1-(zW_M^{-m})^{-1}}$$

Use Geom. Sum Result

$$\sum_{n=N_1}^{N_2-1} a^n = \frac{a^{N_1} - a^{N_2}}{1-a}$$

# Math Shows … (con.t)

Q: What is the $m^{th}$ filter's Frequency Response?

$$G_m^f(\theta) = G_m^z(z)\Big|_{z=e^{j\theta}}$$

$$= \frac{1 - e^{-jM(\theta - 2\pi m/M)}}{1 - e^{-j(\theta - 2\pi m/M)}}$$

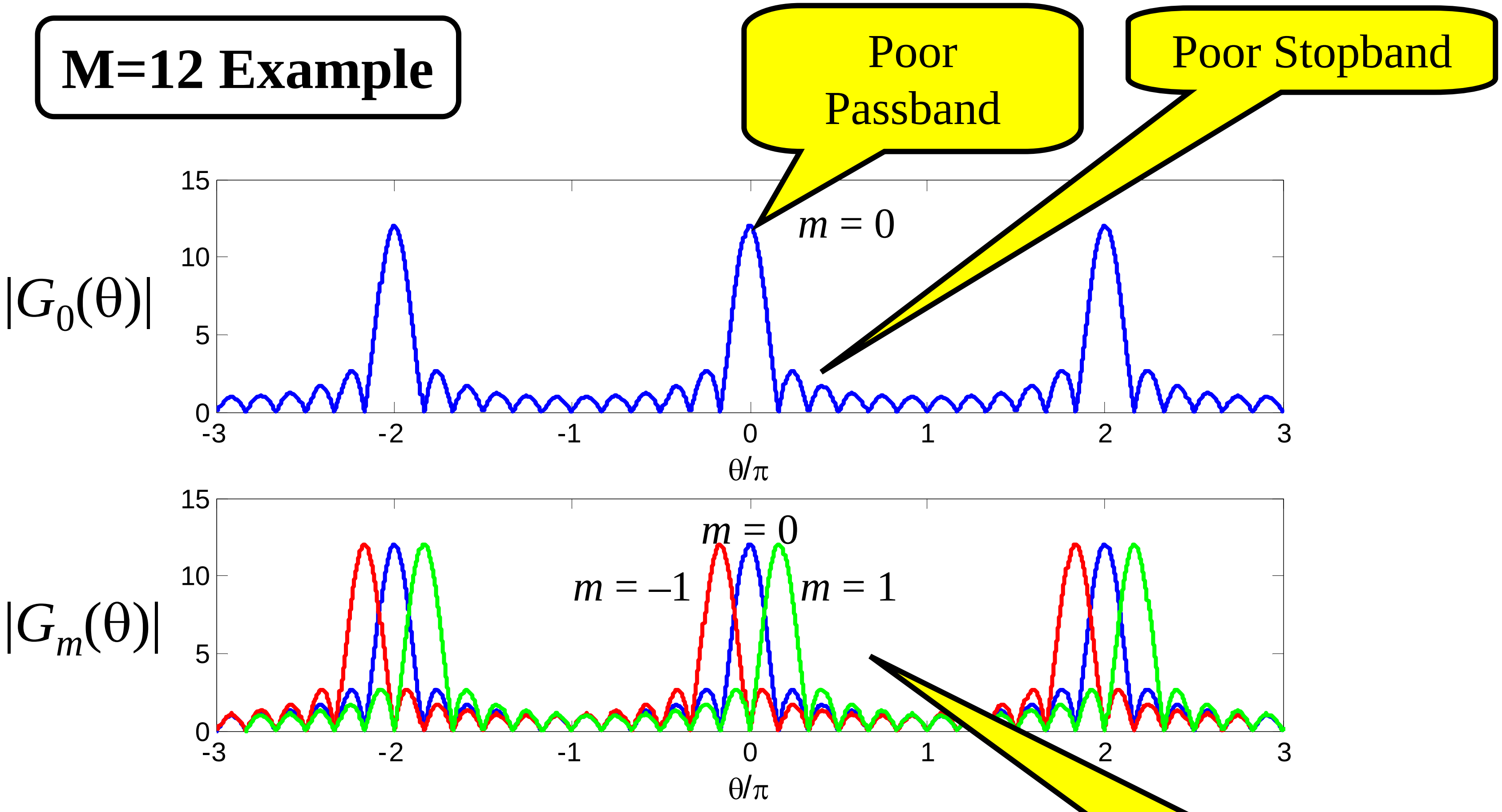
$$= \frac{\sin[0.5M(\theta - 2\pi m/M)]}{\sin[0.5(\theta - 2\pi m/M)]} e^{-j0.5(\theta - 2\pi m/M)(M-1)}$$

- Looks sort of like sinc function: Dirichlet Kernel
- Centered at $\theta = 2\pi m/M$ rad/sample

Note: The window determines the shape of the frequency response. The rectangular window used here makes a poor filter!!!

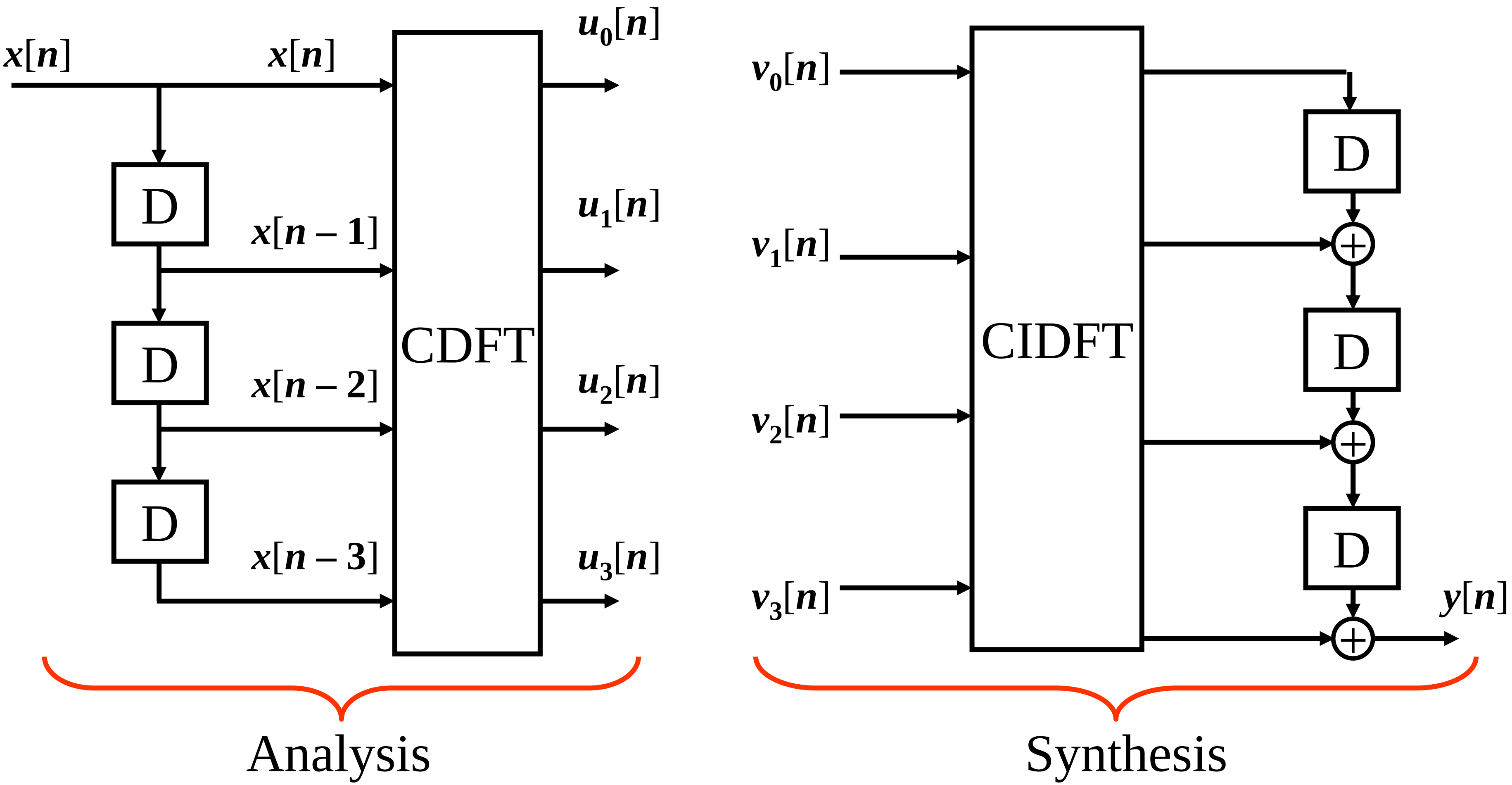
# Frequency Response of Version #1 Filterbank

**M=12 Example**



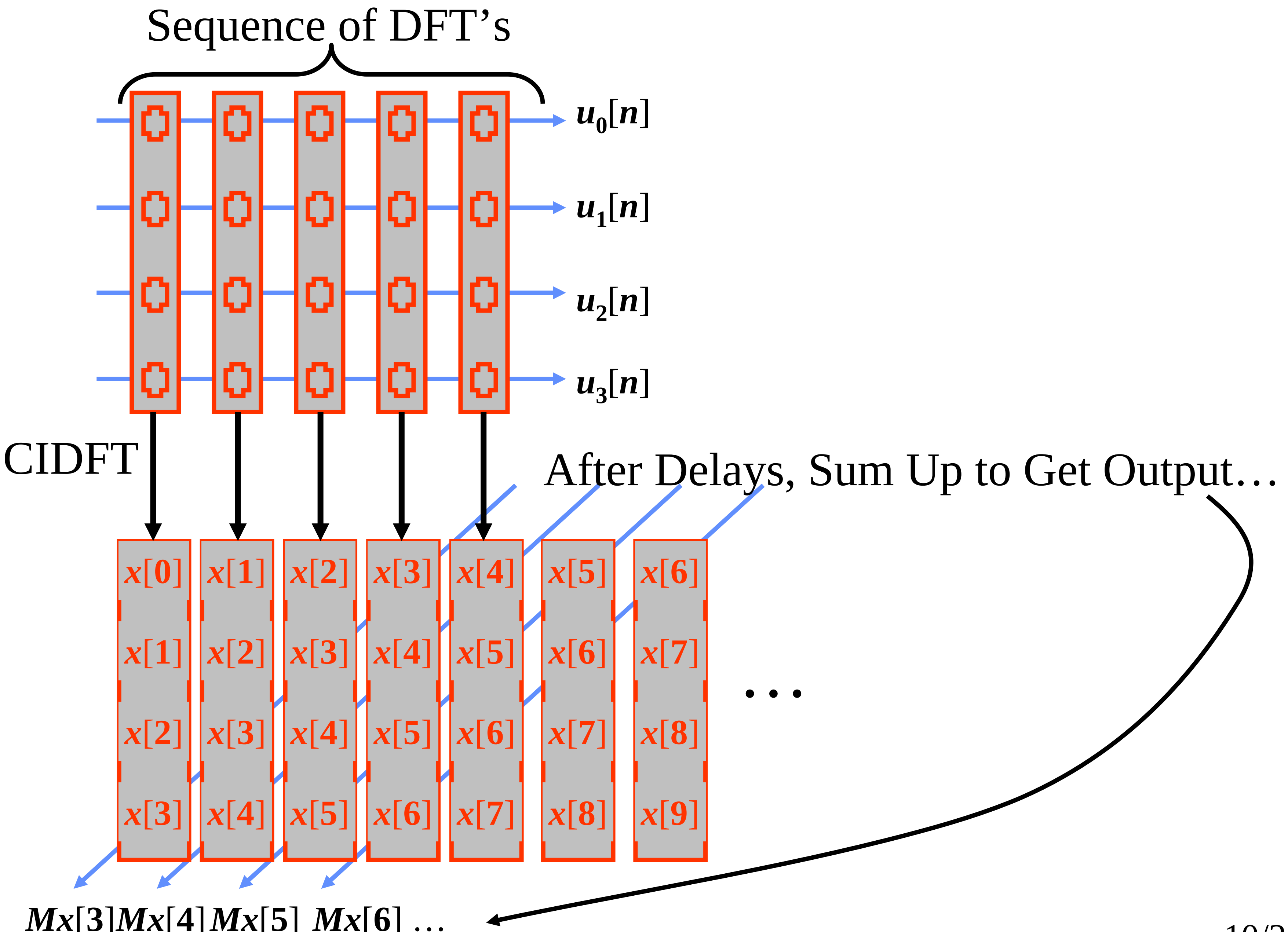

Region of Interest
$-\pi \leq \theta \leq \pi$

# Synthesis Bank for Version #1 Filterbank




CIDFT: Includes 1/M term (not in book!)

# Synthesis Bank for Version #1 (cont.)

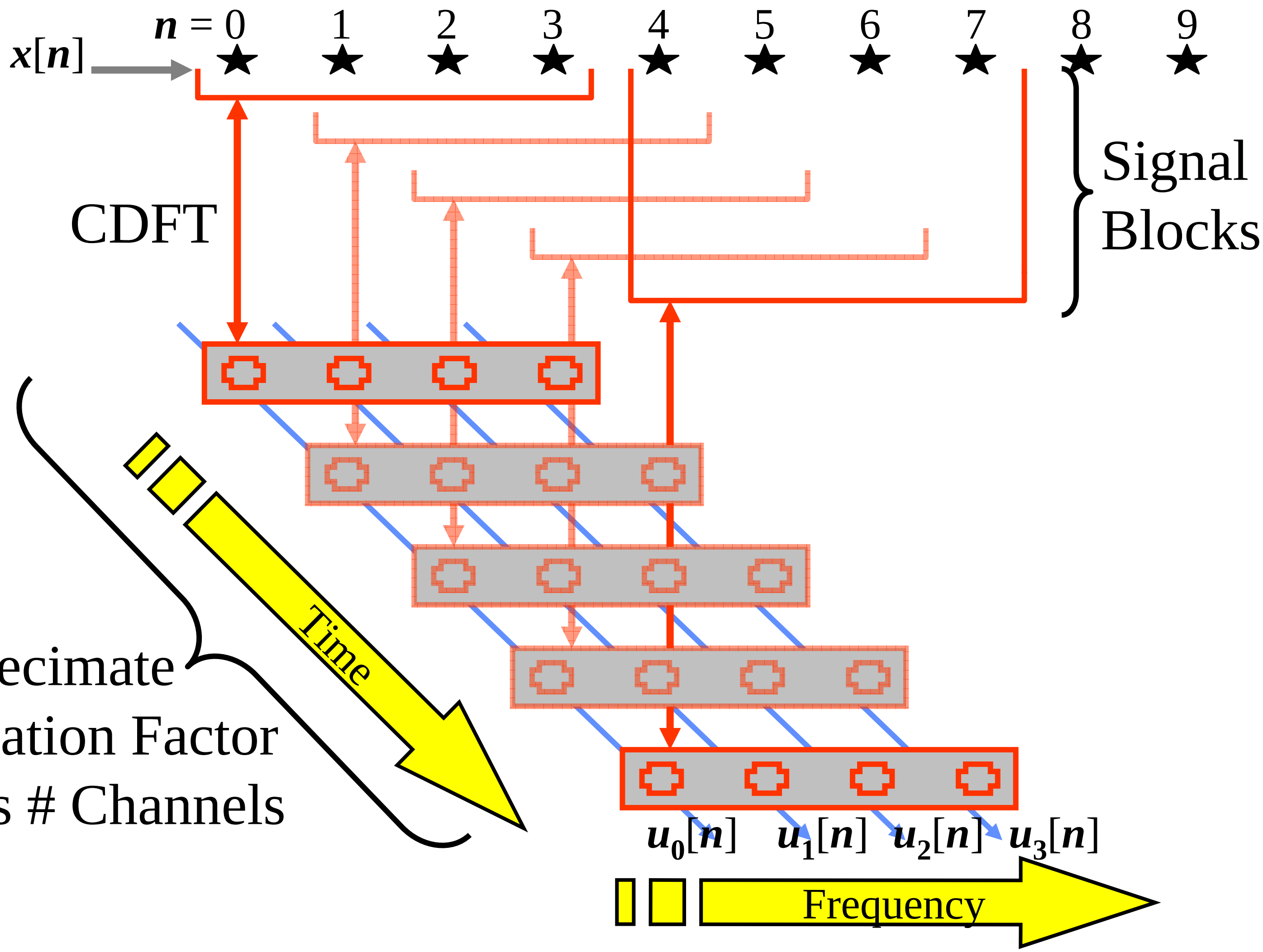


- Total sample rate out of analysis bank is M times input
  - This is redundant and is detrimental in applications like data compression
  - Fixed by decimating (Version #2 – #5)
- Frequency Response is Very Poor
  - DTFT of Rectangular Window
  - Thus, stopband attenuation is very bad and passband falls off
  - Fixed by using non-rectangular window (Versions #3 – #5)
- Filters MUST have same length as number of channels
  - Fixed in Versions #4 & #5
    - Use DSP trick in Version #4
    - Use Polyphase Structure in Version #5

# Version #2: Decimate Output

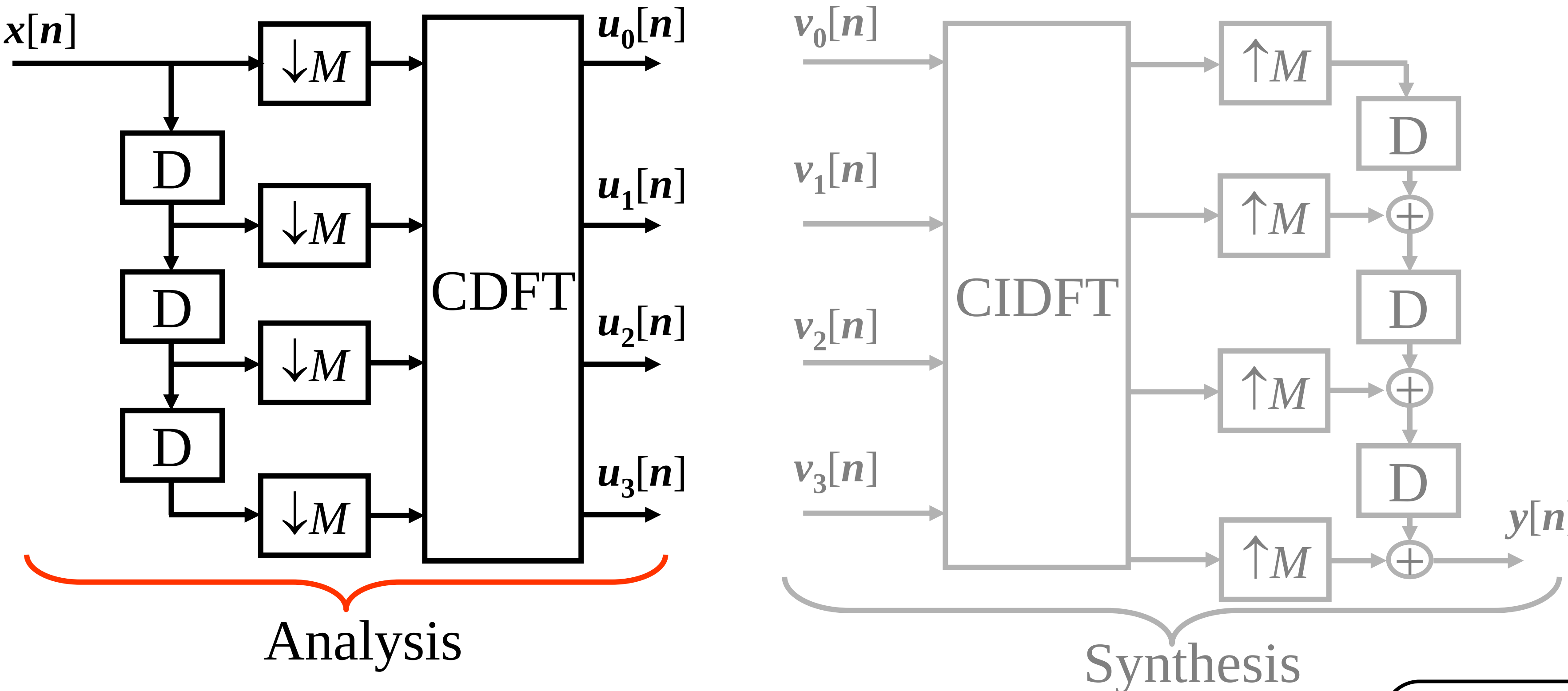
Q: Can we decimate each channel's output and still be able to get back the original signal after synthesis?

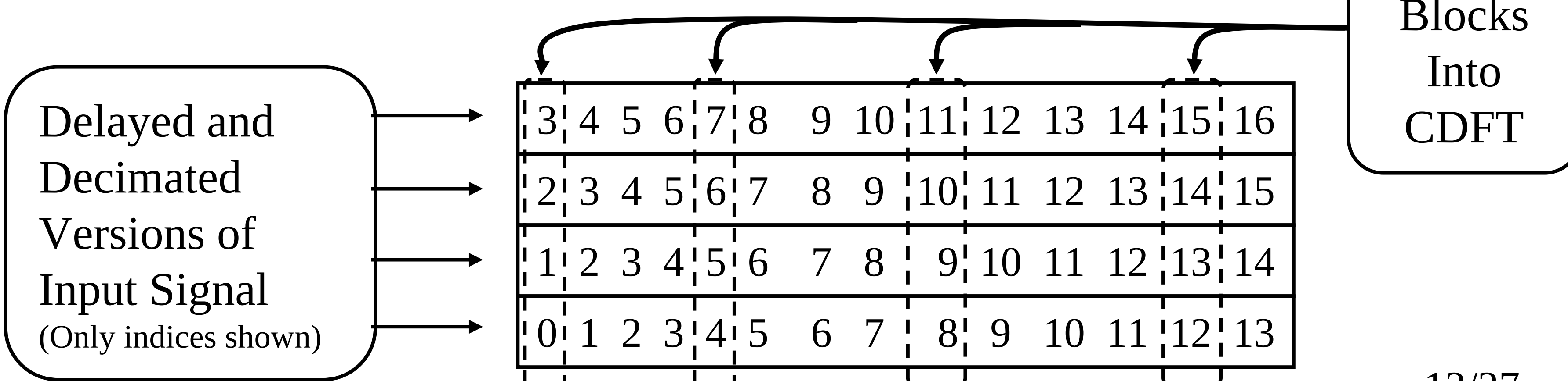
A: Yes… overlapping of the DFT windows is excessive!!!



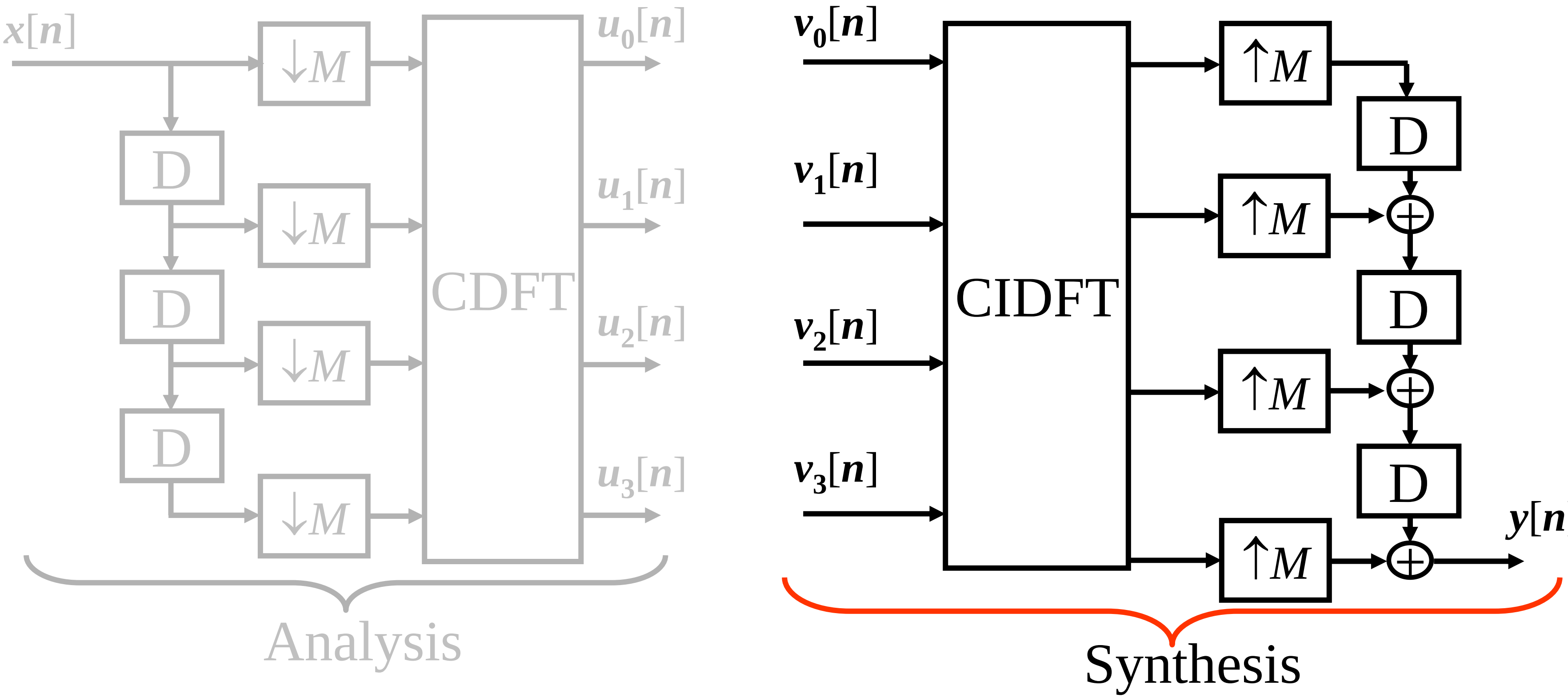

- Can Decimate
- Decimation Factor Equals # Channels

# Different View of Version #2



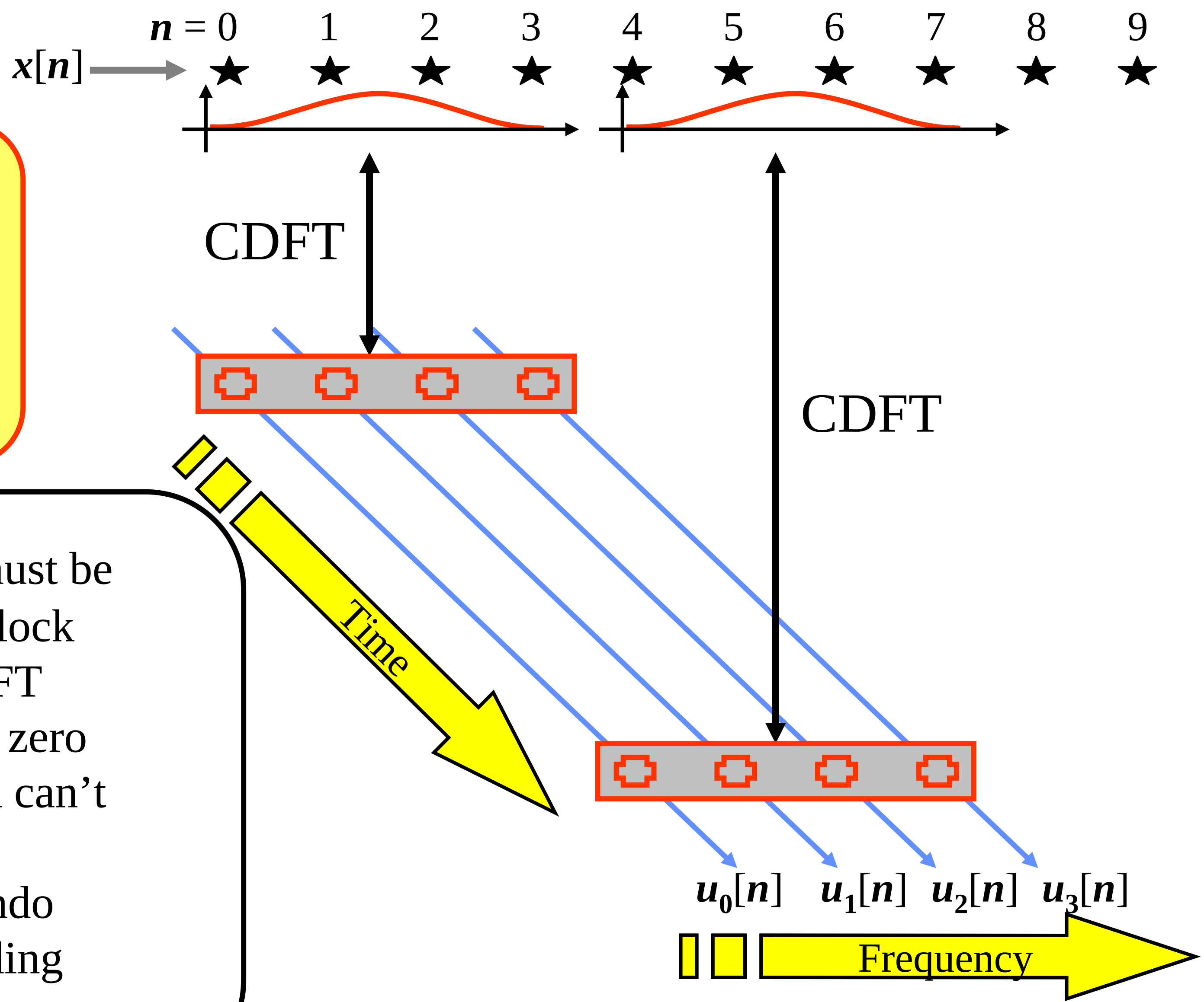
# Different View of Version #2 (cont.)




Up-shifted and Delayed Versions of CIDFT Output (Only indices shown)

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 3 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 11 | 0 | 0 | 0 | 15 | 0 |
| 0 | 0 | 2 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 14 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 13 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 12 | 0 | 0 | 0 | 16 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

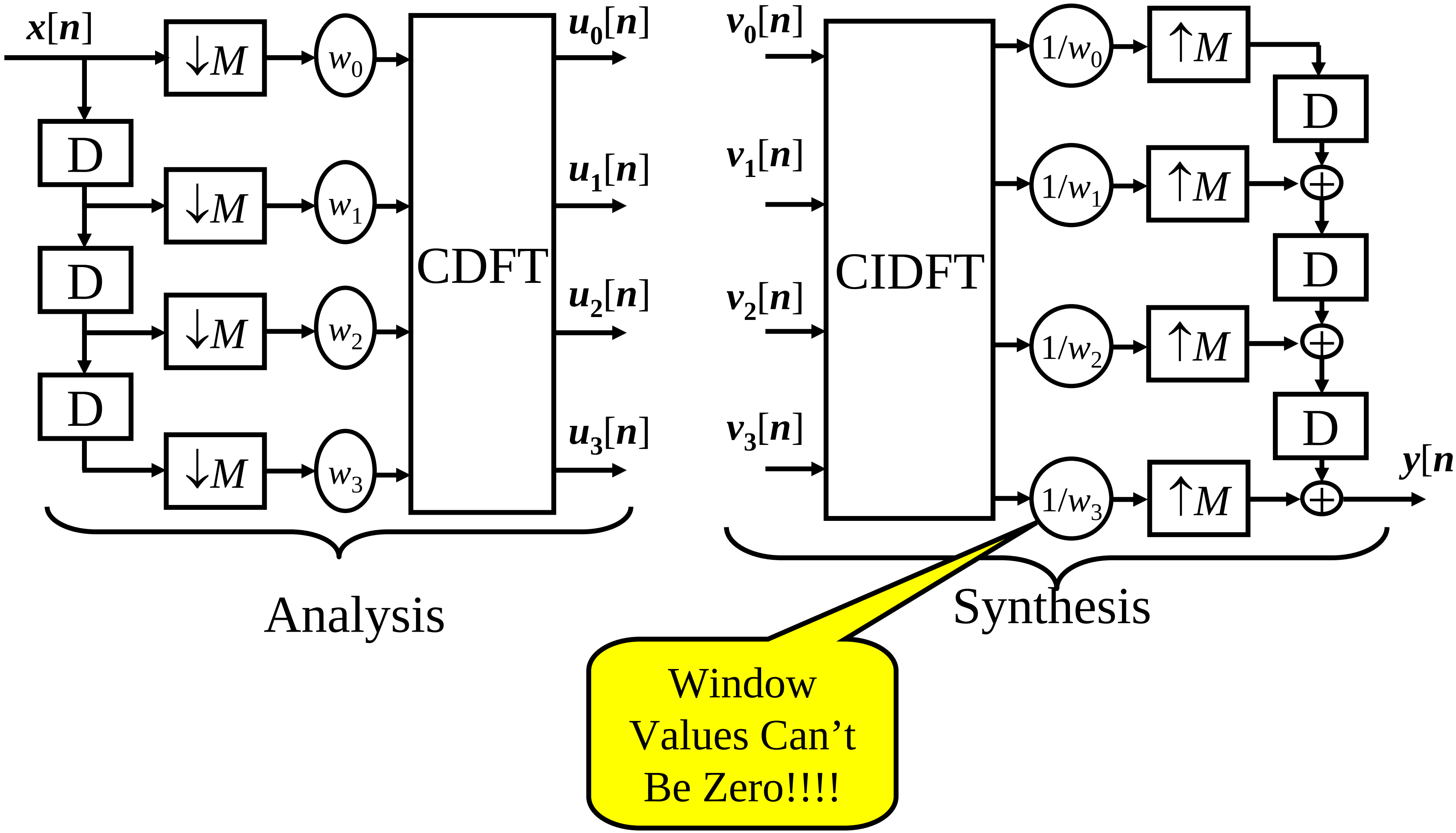
# Version #3: Non-Rectangular Window




**Must Have**
Window Length
||
# of Channels
||
Decimation Factor

- Window $w[n]$ must be non-zero over block
- Otherwise, ICDFT will give back a zero signal value and can't reconstruct
- After ICDFT, undo window by dividing by window values

# Different View of Version #3



# Ver. #4: Arbitrary Size Wind., Sliding DFT

- Version #3 Has Severe Limitation:
  - Window size is set by number of channels desired
  - May force a short window (filter) size
  - But… long filters are often needed to get desired frequency response
- To see how to remove this limitation, back to the Math View:

**Recall Math View of Version #1**

$$u_m[n] = \sum_{i=0}^{M-1} x[n-i] W_M^{im} \quad m = 0, 1, \ldots, M-1$$

***M*-Pt. DFT**

**Math View of Version #3**

$$u_m[n] = \sum_{i=0}^{M-1} x[nM-i]\, w[i]\, e^{j2\pi mi/M} \quad m = 0, 1, \ldots, M-1$$

***M*-Pt. DFT**

Window Length = $M$ (# Channels)

Dec. Factor = $M$ (Non-Overlapped Blocks)

# Version #4 (cont.)

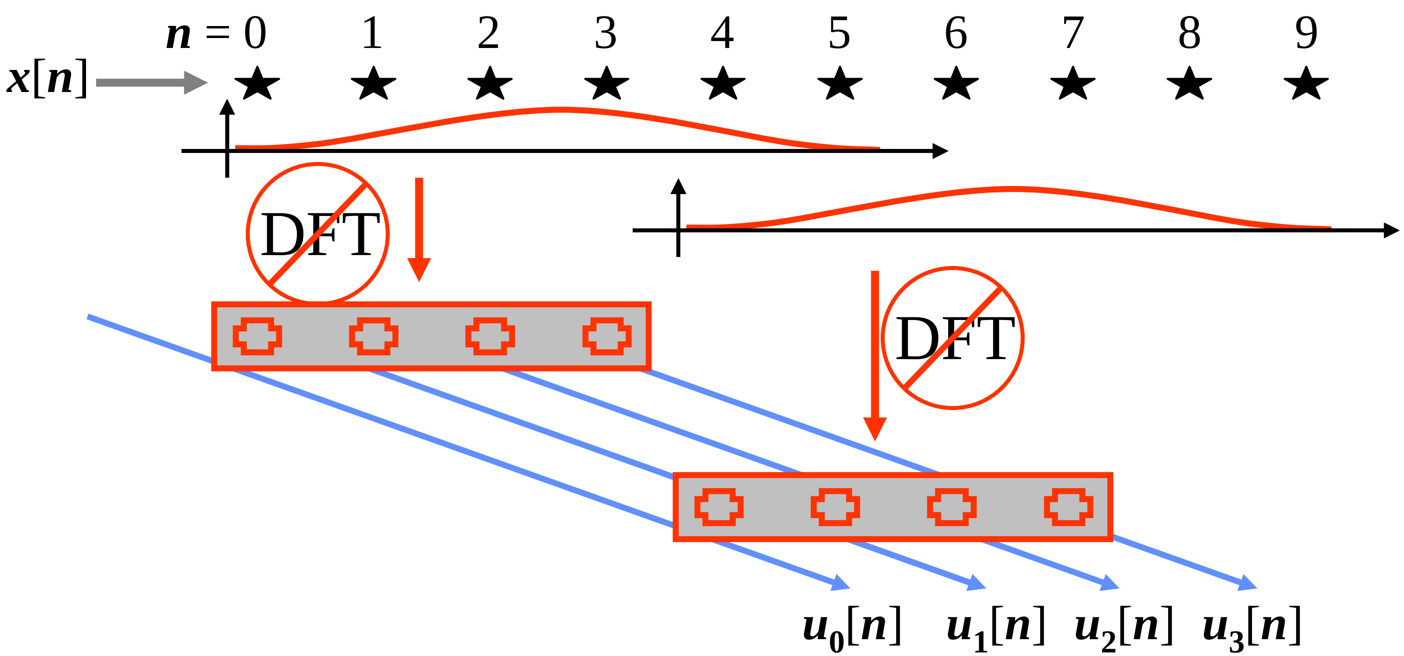
## Math View of Version #4

$$u_m[n] = \sum_{i=0}^{L-1} x[nD - i]\, w[i]\, e^{-j2\pi mi/M} \qquad m = 0, 1, \ldots, M-1$$

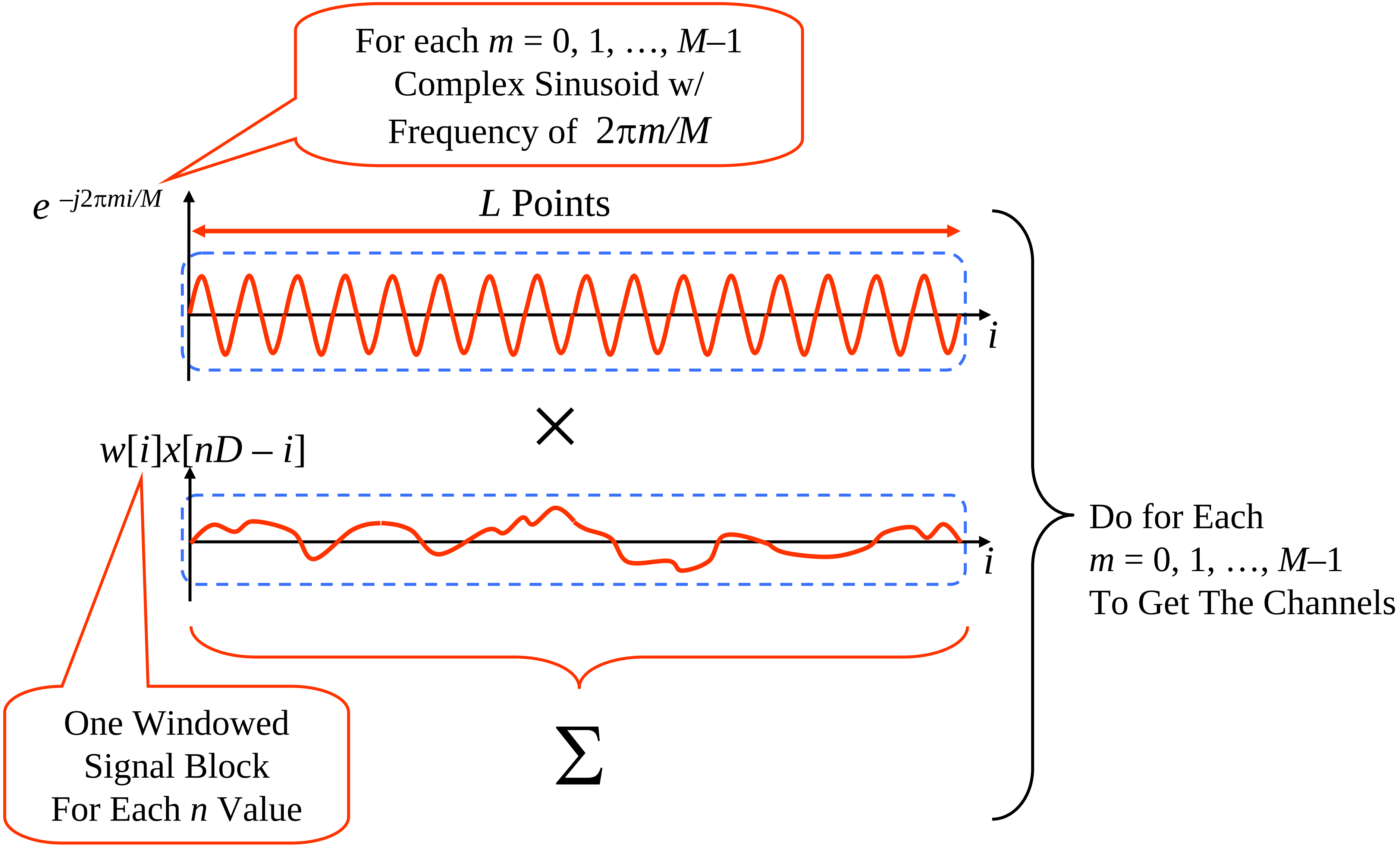
**NOT a DFT!!**
**(L-pt sum but.. M Freq. Pts)**

$D$ = Dec. Factor   $M$ = # Channels   $L$ = Window Length

$$D \le M < L$$



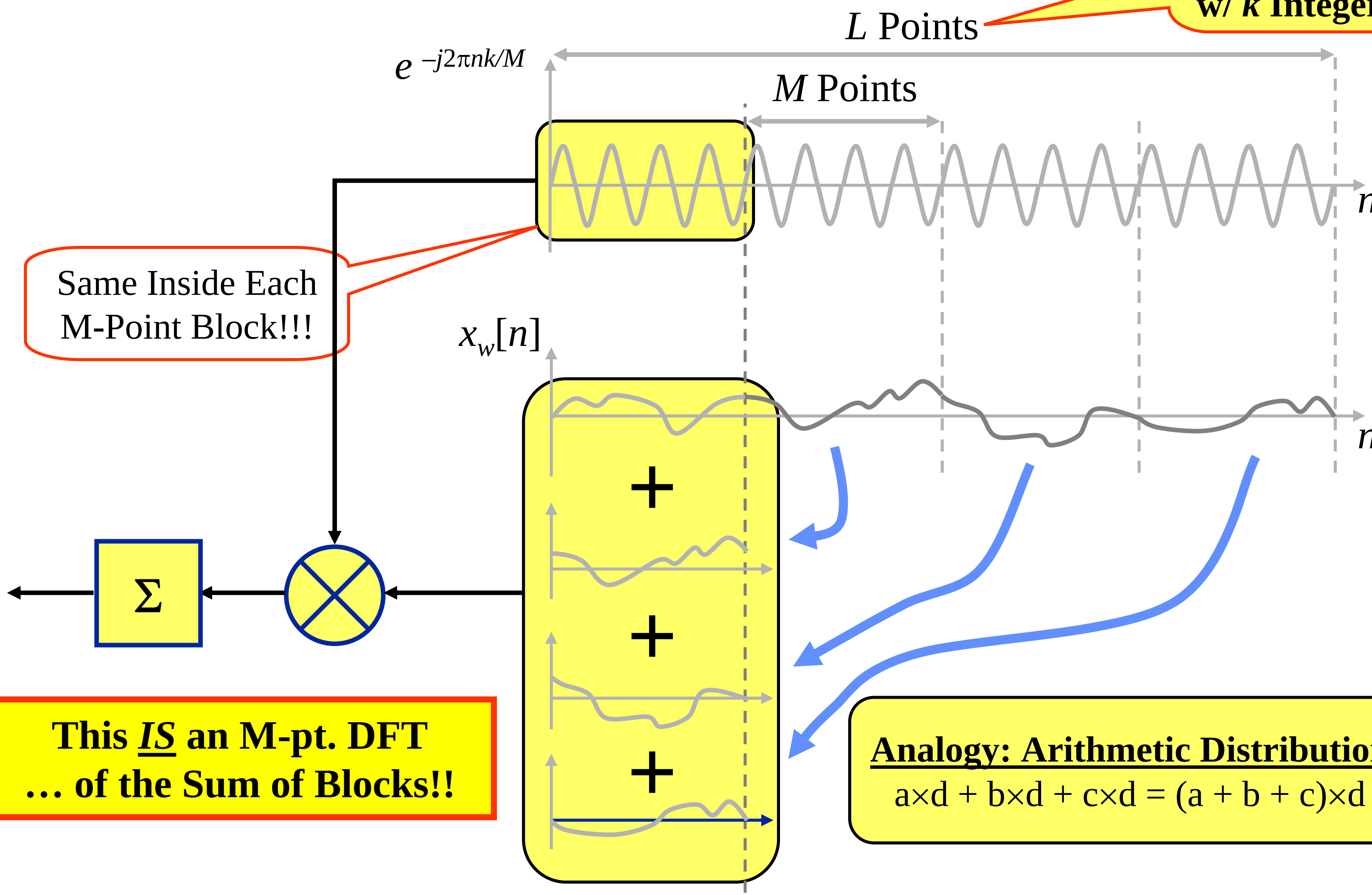
# If it is *NOT* a DFT, What *IS* it??!!




**OK, … But How To Compute This Efficiently ??!!**

**Here is How To Compute This Non-DFT Efficiently !!**
**A DSP TRICK!!!!**

**Must have $L = kM$ w/ $k$ Integer**




**This *IS* an M-pt. DFT … of the Sum of Blocks!!**

**Analogy: Arithmetic Distribution**
a×d + b×d + c×d = (a + b + c)×d

# Version #4: Summary

**Called "Overlap & Add DFT-Based Filter Bank"**

See copied pages posted on Blackboard

- Design of Filter Bank
  - Assume that # of Channels, $M$, has been specified
    - Usually pick $M$ as power of two to allow use of FFT
  - Choose Window Shape and Window Length, $L$, to give desired passband and stopband characteristics
    - To enable good filter, pick $L > M$; also pick $L$ as (integer)×M
  - Choose Decimation Factor, $D$, as large as possible ($D \leq M$) without generating excessive inter-band aliasing
- Algorithm Implementation
  - Apply $L$-pt window to current signal block
  - Break windowed $L$-pt block into $M$-pt sub-blocks
  - Add all the $M$-pt sub-blocks together to get a single $M$-pt block
  - Compute the $M$-pt DFT (using FFT algorithm)
    - Each DFT coefficient is the current output of a channel
  - Move the $L$-pt window ahead $D$ points

**For Synthesis: Crochiere & Rabiner, *Multirate Digital Signal Processing*, Prentice Hall, 1983.**

# Ver. #5: Arb. Size Wind., Polyphase, DFT

**Recall Math View of Version #4**

$$u_m[n] = \sum_{i=0}^{L-1} x[nM - i]\, \underbrace{g_0[i]\, e^{j2\pi mi/M}}_{\triangleq g_m[i]}, \quad m = 0, 1, \ldots, M-1$$

$$= \{x * g_m\}_{(\downarrow M)}[n]$$

Minor Changes
$w[n] \to g_0[n]$
$D \to M$

View: Each channel of FB consists of filter $g_m[n]$ that is a frequency-shifted version of a prototype lowpass filter $g_0[n]$. (All the uniform FBs we've looked at can be viewed this way.)

In the Frequency & Z Domains this is:

$$g_m[n] = g_0[n]\, \underbrace{e^{j2\pi mn/M}}_{W_M^{mn}} \quad \leftrightarrow \quad G_m^{\mathrm{f}}(\theta) = G_0^{\mathrm{f}}(\theta - 2\pi m / M)$$

$$\leftrightarrow \quad G_m^{\mathrm{z}}(z) = G_0^{\mathrm{z}}(z W_M^{-m})$$

Frequency Shift

# Ver. #5: Development

Approach:
1. Write the prototype LPF in its polyphase terms
2. Modulate result to get the channel filters
3. Use result to write pre-decimation channel output
4. Write post-decimation channel output

**Step #1**

$$G_0^z(z) = \sum_{i=0}^{M-1} z^{-i} P_i^z(z^M)$$

**Step #2**

$$G_m^z(z) = G_0^z(zW_M^{-m})$$

$$= \sum_{i=0}^{M-1} (zW_M^{-m})^{-i} P_i^z(z^M \underbrace{W_M^{-mM}}_{=1})$$

$$= \sum_{i=0}^{M-1} W_M^{im}\, z^{-i}\, P_i^z(z^M)$$

**Step #3**

$$U_m^z(z) = G_m^z(z) X^z(z)$$

$$U_m^z(z) = \underbrace{\sum_{i=0}^{M-1} W_M^{im}\, P_i^z(z^M)\, \underbrace{z^{-i} X^z(z)}_{\text{view as input}}}_{\text{apply decimation identity}}$$

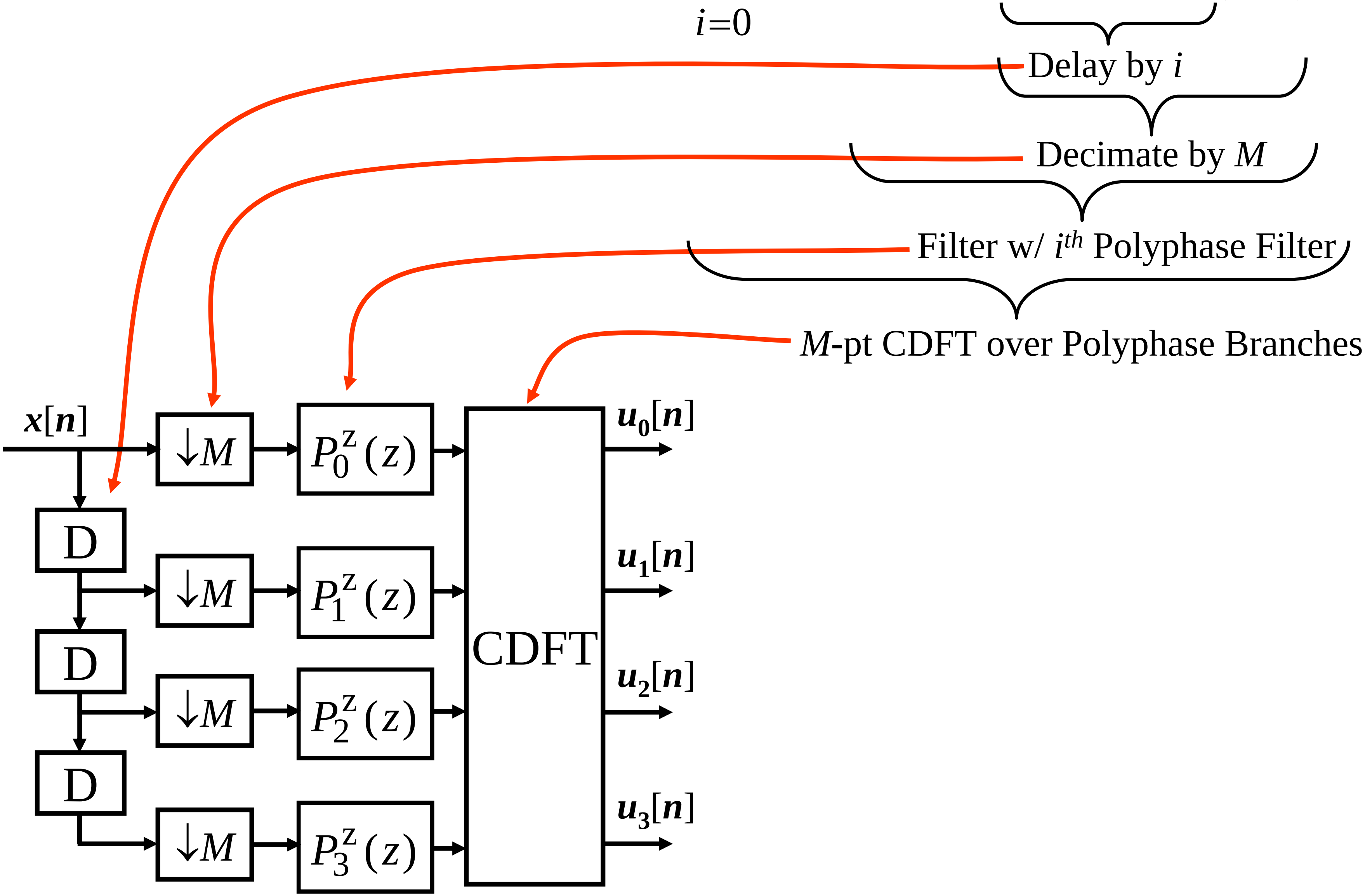
**Step #4**

$$\left\{U_m^z(z)\right\}_{(\downarrow M)} = \sum_{i=0}^{M-1} W_M^{im} P_i^z(z) \left\{z^{-i} X^z(z)\right\}_{(\downarrow M)}$$

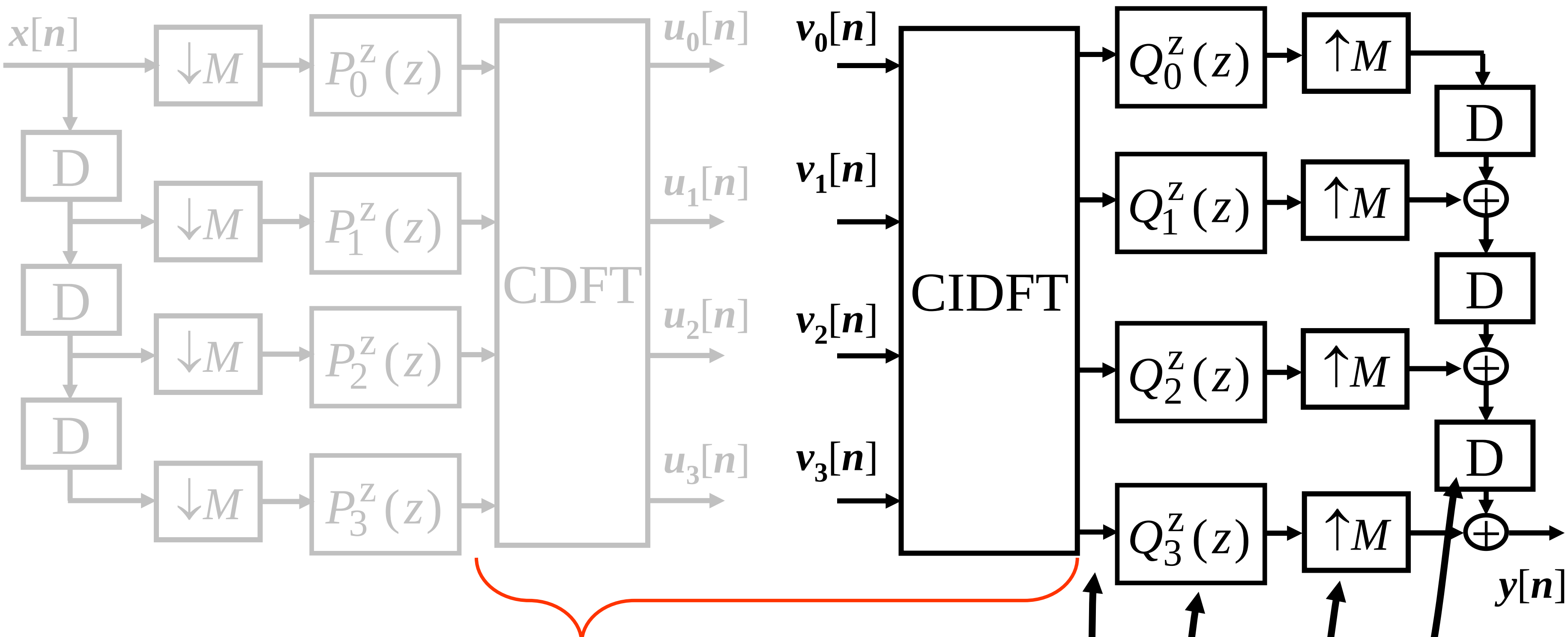
Filter then Dec → Dec then Filter

# Ver. #5: Interpret

$$U_m^z(z) = \sum_{i=0}^{M-1} W_M^{im} P_i^z(z) \left\{ z^{-i} X^z(z) \right\}_{(\downarrow M)}$$



# Ver. #5: Synthesis




**Cancel Each Other**

**➔ Like $P_i(z)$ connects to $Q_i(z)$**

$$Y^Z(z) = \sum_{i=0}^{M-1} \left\{ \left\{ z^{-i} X^Z(z) \right\}_{(\downarrow M)} P_i^Z(z) Q_i^Z(z) \right\}_{(\uparrow M)} z^{-(M-1-i)}$$

# Ver. #5: Synthesis – Does it Work?

**Here's where we were on the last slide:**

$$Y^{z}(z) = \sum_{i=0}^{M-1} \left\{ \left\{ z^{-i} X^{z}(z) \right\}_{(\downarrow M)} P_i^{z}(z) Q_i^{z}(z) \right\}_{(\uparrow M)} z^{-(M-1-i)}$$

**Use Z-Domain result for ↓*M* operation:**

$$Y^{z}(z) = \sum_{i=0}^{M-1} \left\{ \left[ \frac{1}{M} \sum_{m=0}^{M-1} (z^{1/M} W_M^{-m})^{-i} X^{z}(z^{1/M} W_M^{-m}) \right] P_i^{z}(z) Q_i^{z}(z) \right\}_{(\uparrow M)} z^{-(M-1-i)}$$

**Use Z-Domain result for ↑*M* operation:**

$$Y^{z}(z) = \sum_{i=0}^{M-1} \left[ \frac{1}{M} \sum_{m=0}^{M-1} (z W_M^{-m})^{-i} X^{z}(z W_M^{-m}) \right] P_i^{z}(z^M) Q_i^{z}(z^M) z^{-(M-1-i)}$$

$$= z^{-(M-1)} \underbrace{\sum_{m=0}^{M-1} X^{z}(z W_M^{-m}) \underbrace{\left[ \frac{1}{M} \sum_{i=0}^{M-1} W_M^{im} P_i^{z}(z^M) Q_i^{z}(z^M) \right]}_{\text{Want} = c z^{-l} \delta[m]}}_{\text{... to get this} = c z^{-l} X^{z}(z)}$$

This gives "Perfect Reconstruction"

Requirement for Perfect Reconstruction

# Ver. #5: Perfect Recon Criteria

**Look at what we saw on the last slide:**

$$\underbrace{\frac{1}{M}\sum_{i=0}^{M-1} W_M^{im} P_i^z(z^M) Q_i^z(z^M)}_{\text{IDFT of } P_i^z(z^M) Q_i^z(z^M)} = cz^{-l}\delta[m]$$

**Taking DFT of each side gives an <u>Equivalent PR Criteria</u>:**

$$P_i^z(z^M) Q_i^z(z^M) = cz^{-l}, \quad 0 \le i \le M-1$$

- General Filter Designs to Meet This are HARD!!! (We Won't Cover It)
  - <u>Special Cases</u>:
    - <u>Version #2</u> is…. $P_i(z) = Q_i(z) = 1, \quad 0 \le i \le M{-}1$
    - <u>Version #3</u> is…. $P_i(z) = w[i] \quad \& \quad Q_i(z) = 1/\, w[i], \quad 0 \le i \le M{-}1$