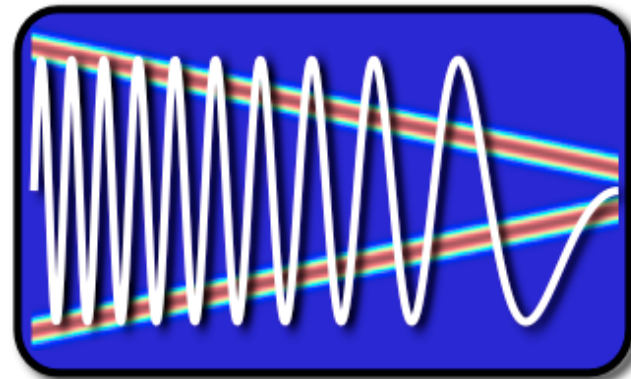


EE123



# Digital Signal Processing

## Lecture 6 Properties of DFT

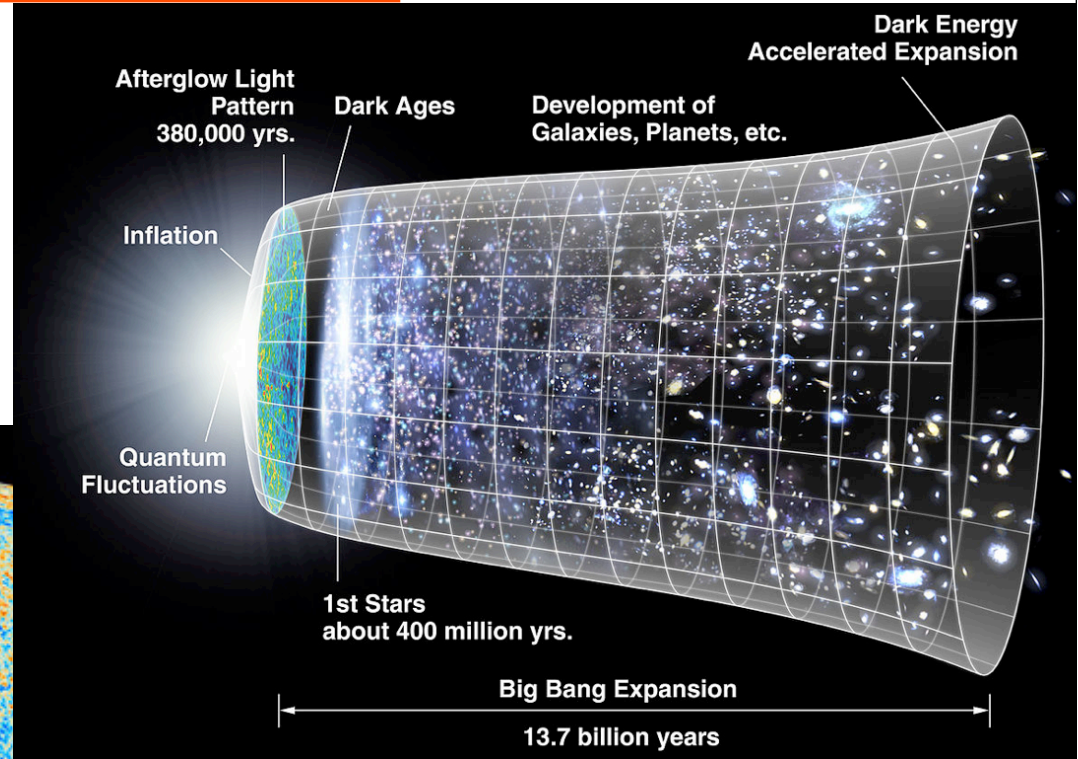
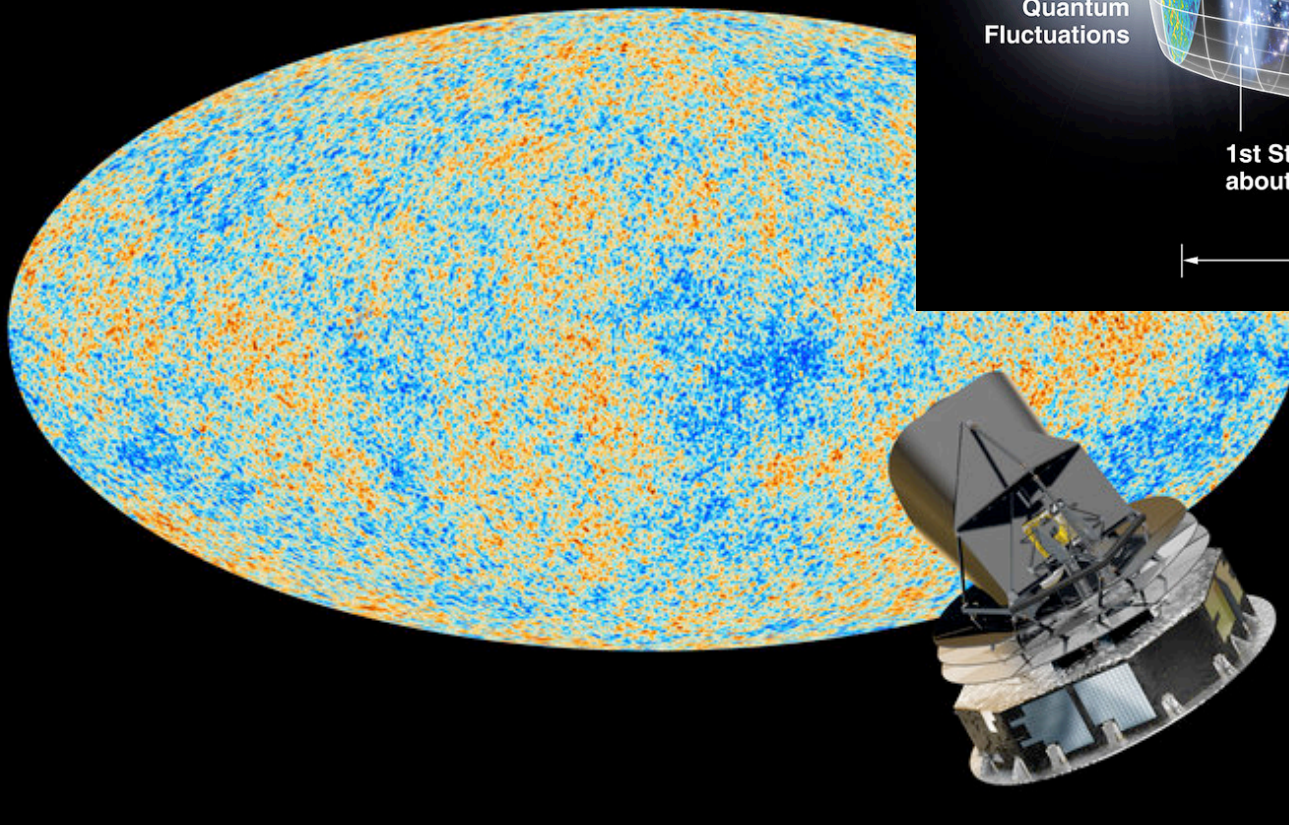
## Announcements

---

- HW1 solutions posted -- self grading due
- HW2 due Friday
  
- SDR give after GSI Wednesday
- Finish reading Ch. 8, start Ch. 9
  
- ham radio licensing lectures Tue  
6:30-8pm Cory 521

# Cool things DSP

- Cosmic Microwave Background radiation



## Last Time

---

- Discrete Fourier Transform
  - Similar to DFS
  - Sampling of the DTFT (subtitles....more later)
  - Properties of the DFT
  
- Today
  - Linear convolution with DFT
  - Fast Fourier Transform

## Properties of DFT

---

- Inherited from DFS (EE120/20) so no need to be proved

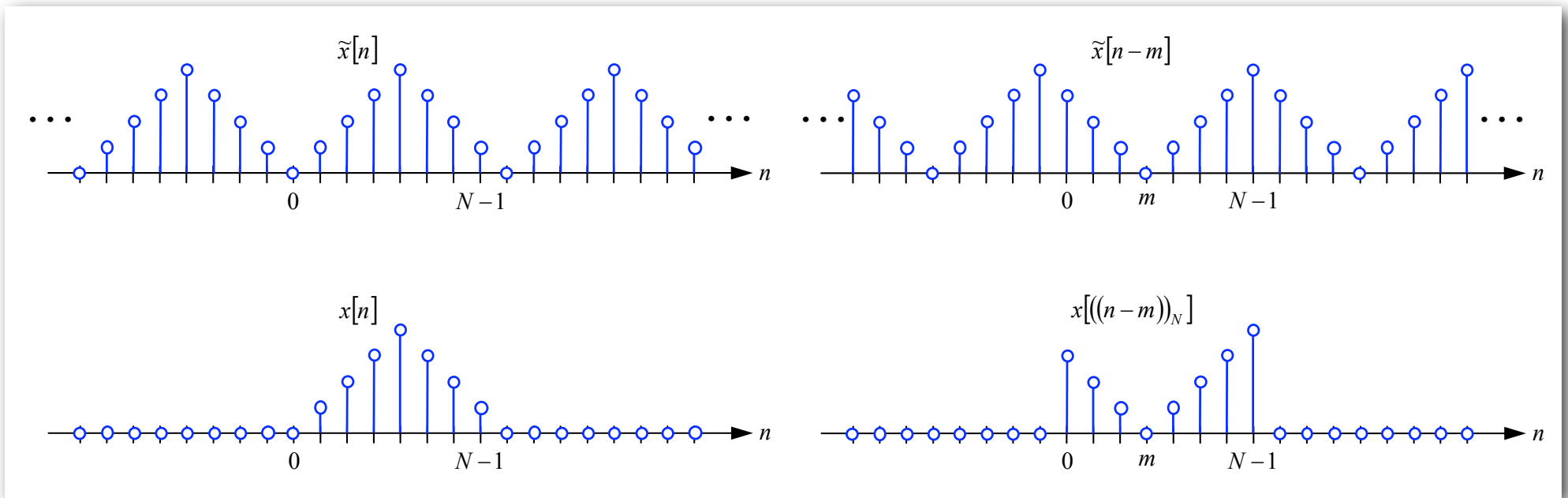
- Linearity

$$\alpha_1 x_1[n] + \alpha_2 x_2[n] \leftrightarrow \alpha_1 X_1[k] + \alpha_2 X_2[k]$$

- Circular Time Shift

$$x[((n - m))_N] \leftrightarrow X[k] e^{-j(2\pi/N)km} = X[k] W_N^{km}$$

# Circular shift



## Properties of DFT

---

- Circular frequency shift

$$x[n]e^{j(2\pi/N)nl} = x[n]W_N^{-nl} \leftrightarrow X[((k - l))_N]$$

- Complex Conjugation

$$x^*[n] \leftrightarrow X^*[((-k))_N]$$

- Conjugate Symmetry for Real Signals

$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Show....

# Properties of DFT

---

- Parseval's Identity

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$$

- Proof (in matrix notation)

$$\mathbf{x}^* \mathbf{x} = \left( \frac{1}{N} \mathbf{W}_N^* \mathbf{X} \right)^* \left( \frac{1}{N} \mathbf{W}_N^* \mathbf{X} \right) = \frac{1}{N^2} \mathbf{X}^* \underbrace{\mathbf{W}_N \mathbf{W}_N^*}_{N \cdot \mathbf{I}} \mathbf{X} = \frac{1}{N} \mathbf{X}^* \mathbf{X}$$



## Circular Convolution Sum

---

- Circular Convolution:

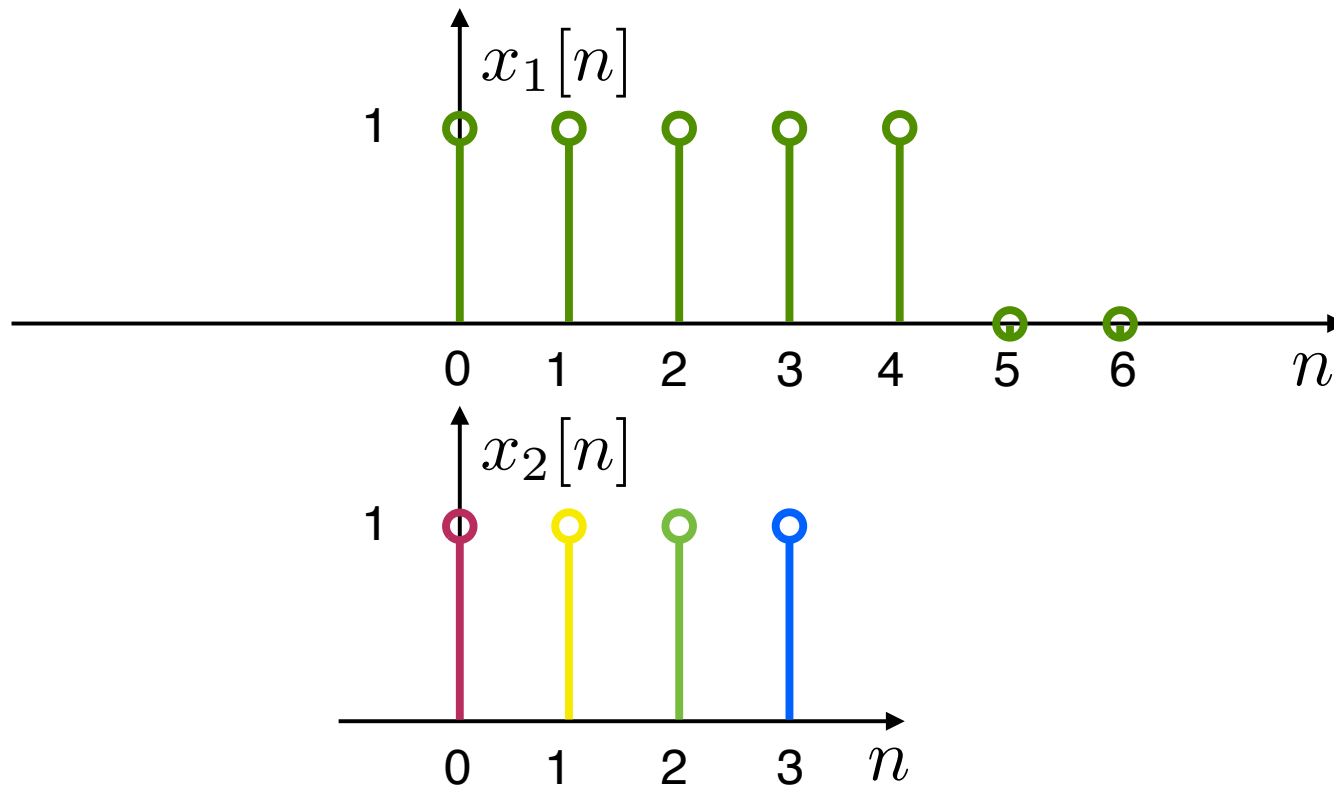
$$x_1[n] \circledast_N x_2[n] \triangleq \sum_{m=0}^{N-1} x_1[m] x_2[\left((n - m)\right)_N]$$

for two signals of length N

- Note: Circular convolution is commutative

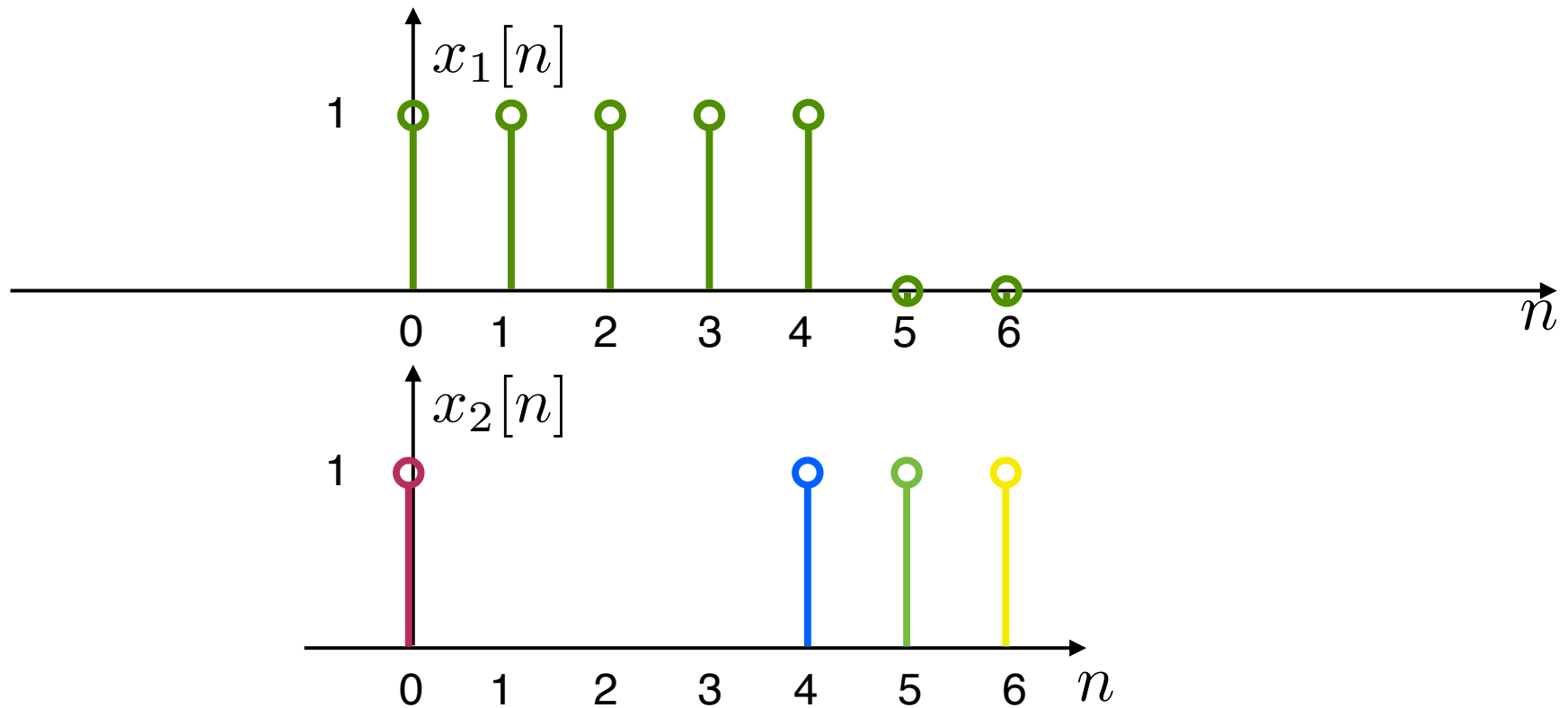
$$x_2[n] \circledast_N x_1[n] = x_1[n] \circledast_N x_2[n]$$

# Compute Circular Convolution Sum



$$y[n] = x_1[n] \textcircled{7} x_2[n] = ?$$

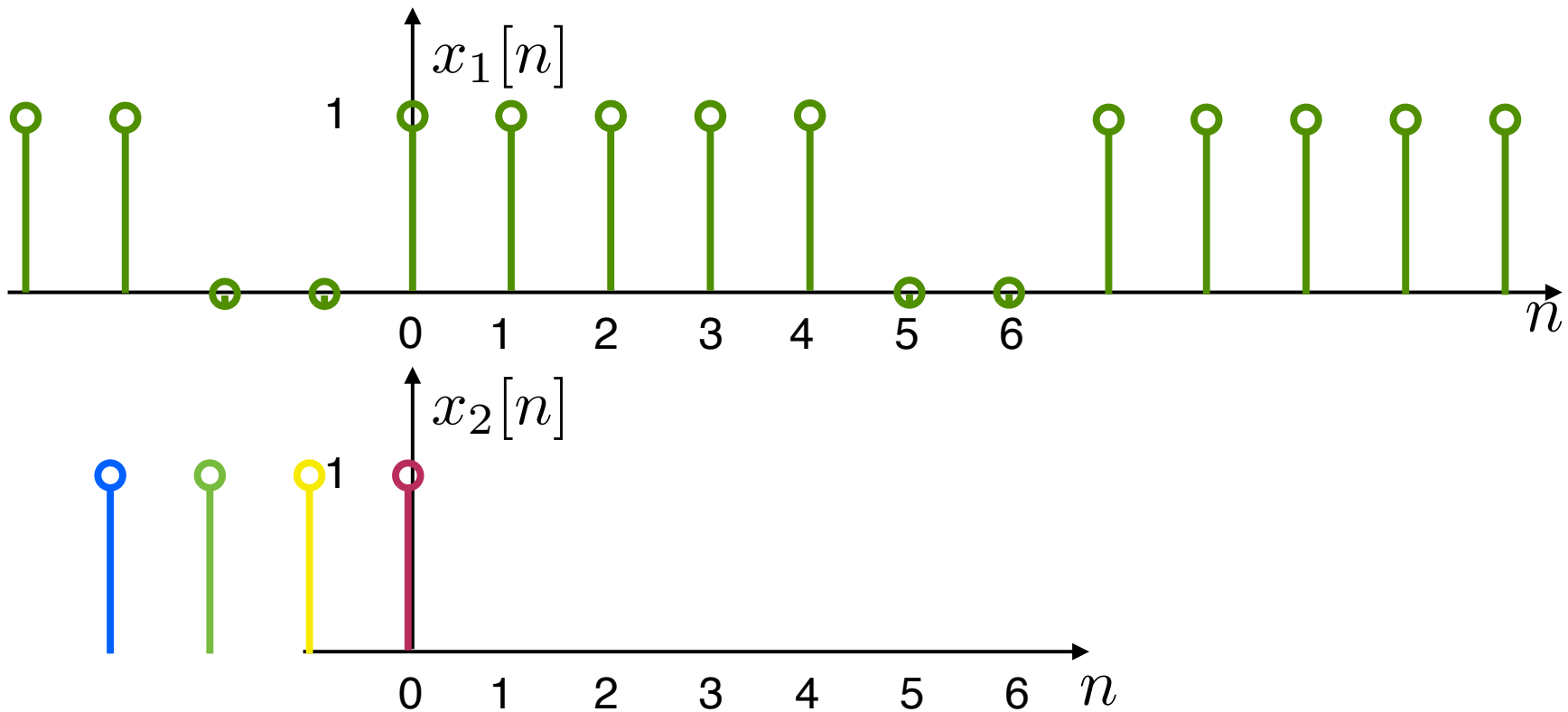
# Compute Circular Convolution Sum



Circular 'flip'  
multiply and add  
Here:  $y[0]$

$$y[n] = x_1[n] \textcircled{\gamma} x_2[n] = ?$$

# Compute Circular Convolution Sum



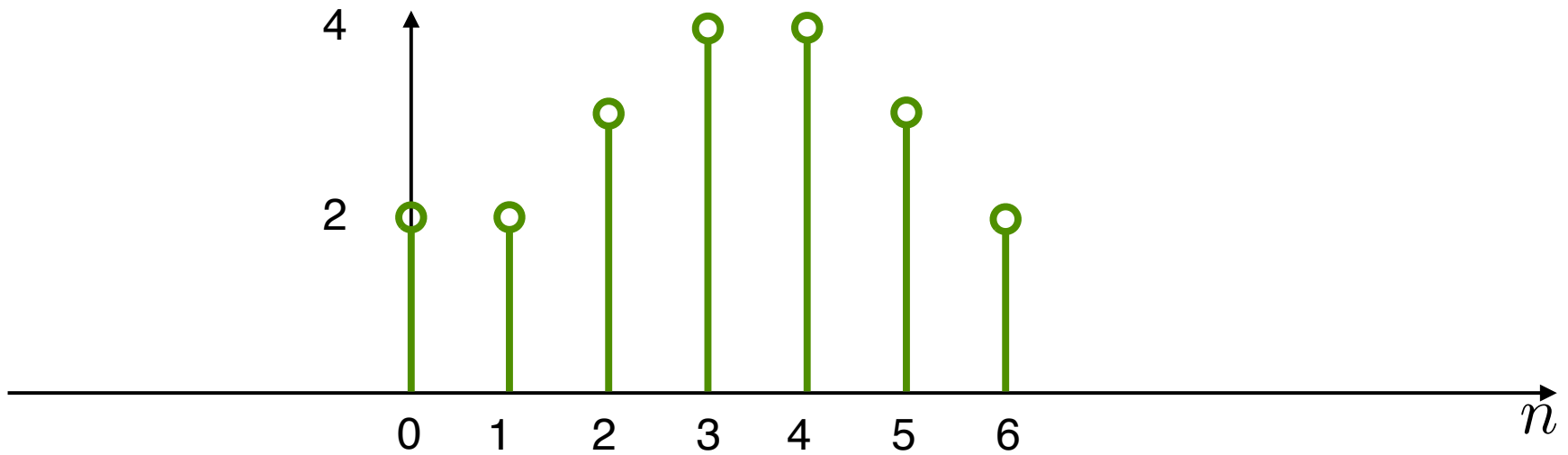
Equivalent periodic convolution over a period

$$y[n] = x_1[n] \circledast x_2[n] = ?$$

# Result

---

$$y[n] = x_1[n] \quad \textcircled{\gamma} \quad x_2[n] = ?$$



## Properties of DFT

---

- **Circular Convolution:** Let  $x_1[n]$ ,  $x_2[n]$  be length  $N$

$$x_1[n] \circledast x_2[n] \leftrightarrow X_1[k] \cdot X_2[k]$$

Very useful!!! ( for linear convolutions with DFT)

- **Multiplication:** Let  $x_1[n]$ ,  $x_2[n]$  be length  $N$

$$x_1[n] \cdot x_2[n] \leftrightarrow \frac{1}{N} X_1[k] \circledast X_2[k]$$

# Linear Convolution

---

- Next....
  - Using DFT, circular convolution is easy
  - But, **linear** convolution is useful, not circular
  - So, show how to perform linear convolution with circular convolution
  - Used DFT to do linear convolution

# Linear Convolution

---

- We start with two non-periodic sequences:

$$x[n] \quad 0 \leq n \leq L - 1$$

$$h[n] \quad 0 \leq n \leq P - 1$$

for example  $x[n]$  is a signal and  $h[n]$  an impulse response of a filter

- We want to compute the linear convolution:

$$y[n] = x[n] * h[n] = \sum_{m=0}^{L-1} x[m]h[n - m]$$

$y[n]$  is nonzero for  $0 \leq n \leq L+P-2$  with length  **$M=L+P-1$**

- Requires  $L \cdot P$  multiplications



## Linear Convolution via Circular Convolution

- Zero-pad  $x[n]$  by  $P-1$  zeros

$$x_{zp}[n] = \begin{cases} x[n] & 0 \leq n \leq L - 1 \\ 0 & L \leq n \leq L + P - 2 \end{cases}$$

- Zero-pad  $h[n]$  by  $L-1$  zeros

$$h_{zp}[n] = \begin{cases} h[n] & 0 \leq n \leq P - 1 \\ 0 & P \leq n \leq L + P - 2 \end{cases}$$

- Now, both sequences are of length  $M=L+P-1$

## Linear Convolution via Circular Convolution

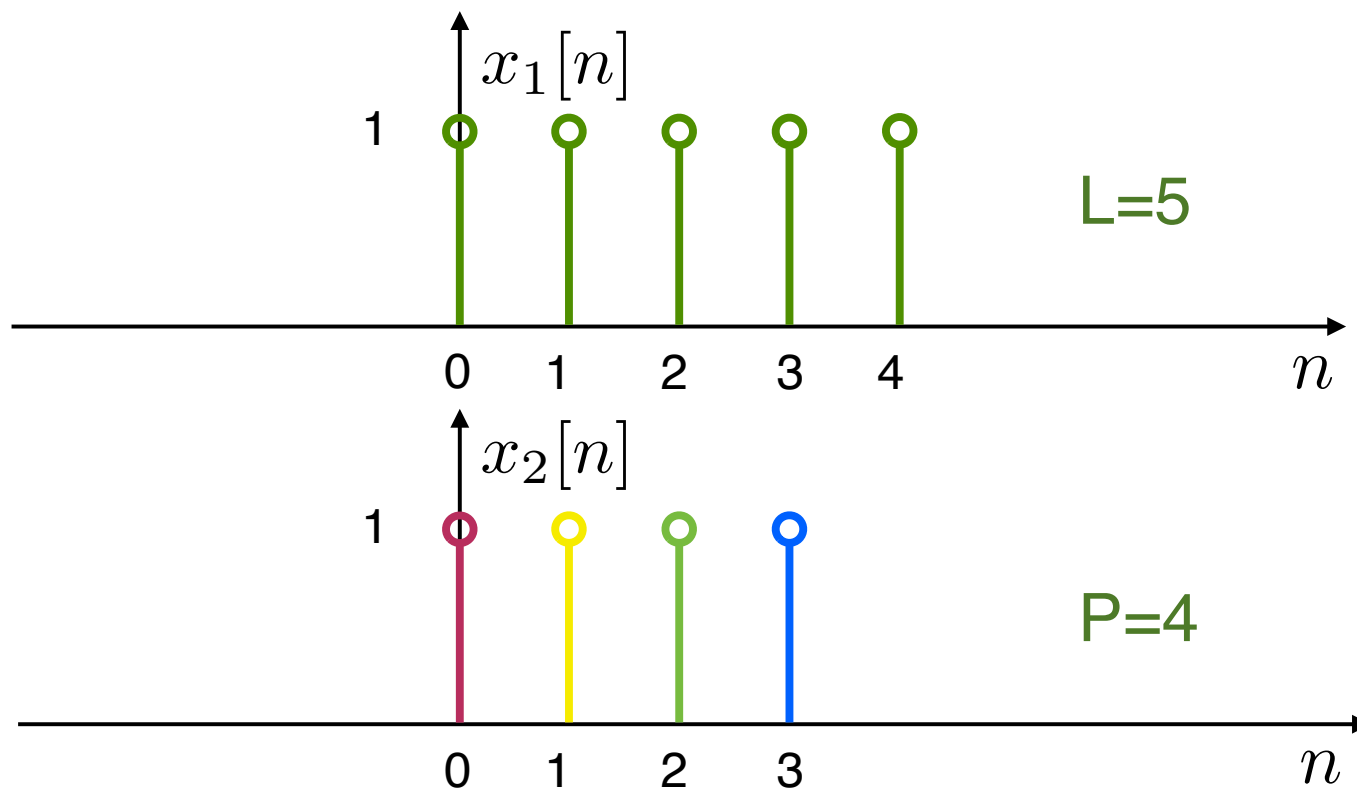
- Now, both sequences are of length  $M=L+P-1$
- We can now compute the linear convolution using a circular one with length  $M = L+P-1$

### Linear convolution via circular

$$y[n] = x[n] * y[n] = \begin{cases} x_{zp}[n] \circledast h_{zp}[n] & 0 \leq n \leq M - 1 \\ 0 & \text{otherwise} \end{cases}$$

# Example

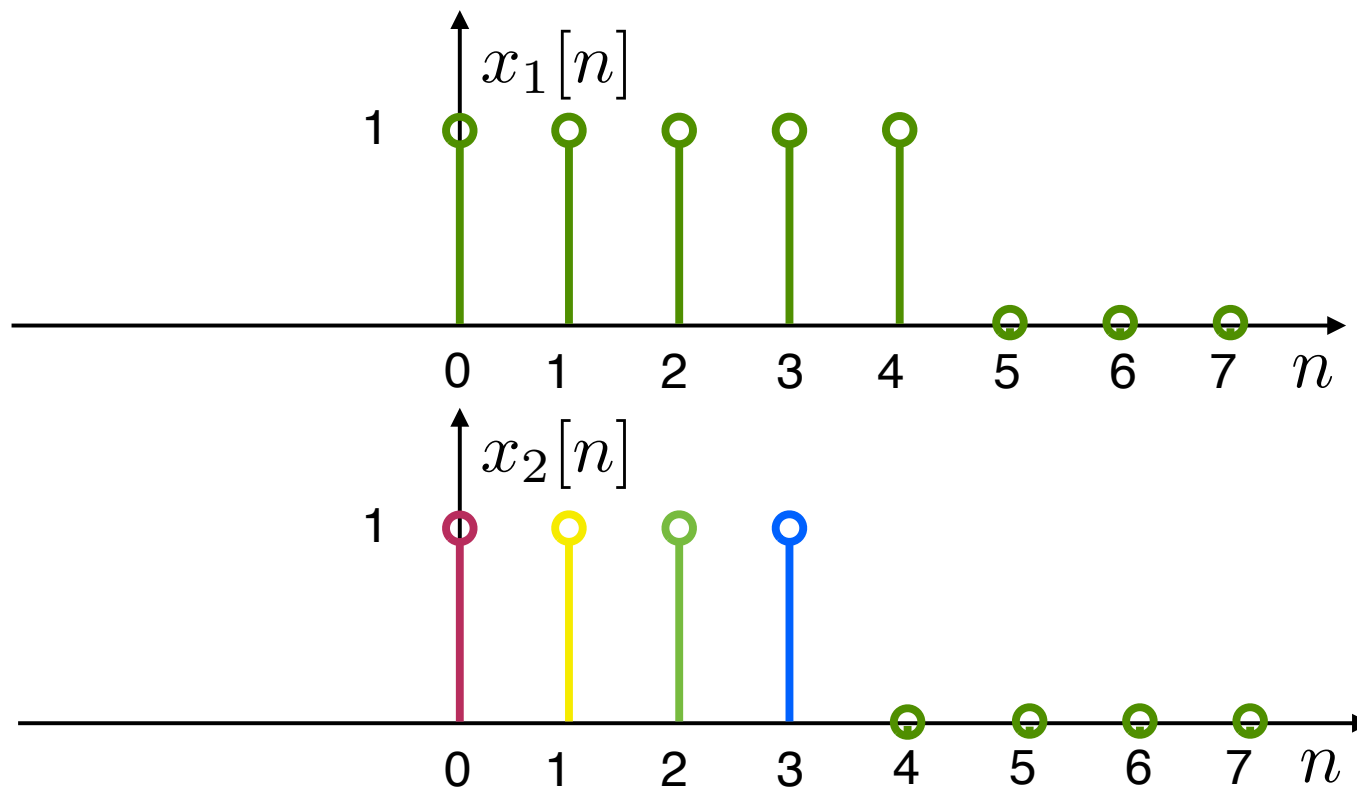
---



$$M = L + P - 1 = 8$$

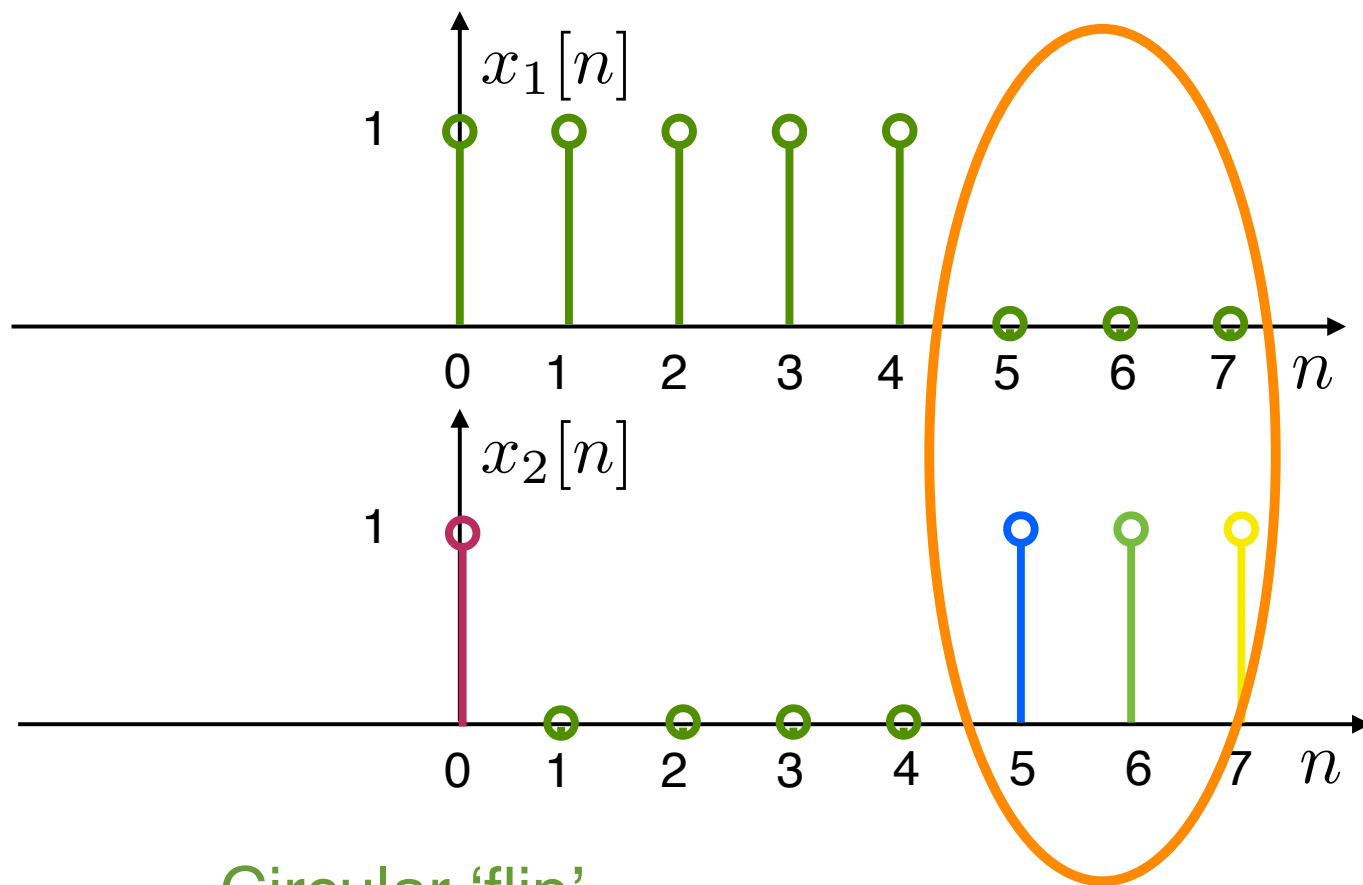
# Example

---



$$M = L + P - 1 = 8$$

# Example



Circular 'flip'

$$M = L + P - 1 = 8$$

$$y[n] = x_1[n] \textcircled{8} x_2[n] = x_1[n] * x_2[n]$$

## Linear Convolution using DFT

---

- In practice we can implement a circulant convolution using the DFT property:

$$\begin{aligned}x[n] * h[n] &= x_{zp}[n] \circledast h_{zp}[n] \\ &= \mathcal{DFT}^{-1} \{ \mathcal{DFT} \{ x_{zp}[n] \} \cdot \mathcal{DFT} \{ h_{zp}[n] \} \}\end{aligned}$$

for  $0 \leq n \leq M-1$ ,  $M=L+P-1$

- **Advantage:** DFT can be computed with  $N \log_2 N$  complexity (FFT algorithm later!)
- **Drawback:** Must wait for all the samples -- huge delay -- incompatible with real-time

# Block Convolution

---

- **Problem:**
  - An input signal  $x[n]$ , has very long length (could be considered infinite)
  - An impulse response  $h[n]$  has length  $P$
  - We want to take advantage of DFT/FFT and compute convolutions in blocks that are shorter than the signal
- **Approach:**
  - Break the signal into small blocks
  - Compute convolutions
  - Combine the results

# Block Convolution

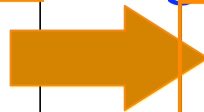
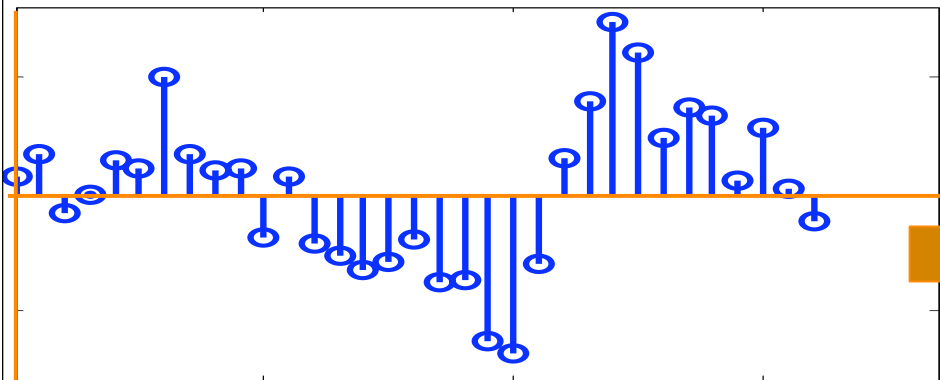
---

## Example:

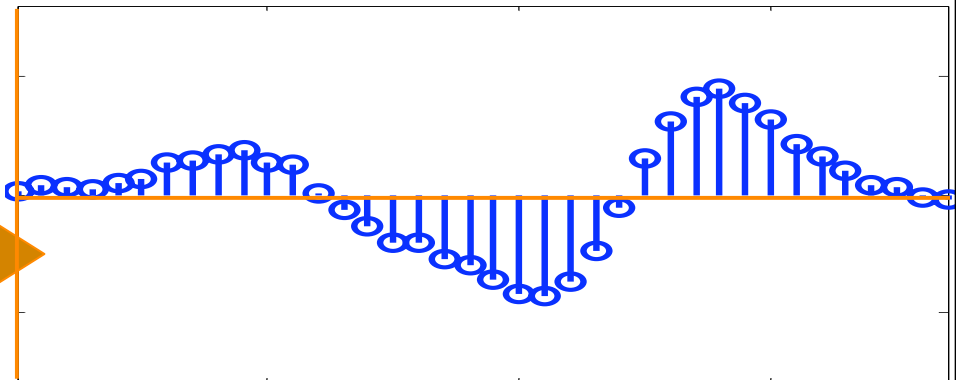
$h[n]$  Impulse response, Length  $P=6$



$x[n]$  Input Signal, Length  $P=33$



$y[n]$  Output Signal, Length  $P=38$





# Overlap-Add Method

We decompose the input signal  $x[n]$  into non-overlapping segments  $x_r[n]$  of length  $L$ :

$$x_r[n] = \begin{cases} x[n] & rL \leq n \leq (r+1)L - 1 \\ 0 & \text{otherwise} \end{cases}$$

The input signal is the sum of these input segments:

$$x[n] = \sum_{r=0}^{\infty} x_r[n]$$

The output signal is the sum of the output segments  $x_r[n] * h[n]$ :

$$y[n] = x[n] * h[n] = \sum_{r=0}^{\infty} x_r[n] * h[n] \quad (1)$$

Each of the output segments  $x_r[n] * h[n]$  is of length  $N = L + P - 1$ .

# Overlap-Add Method

We can compute each output segment  $x_r[n] * h[n]$  with linear convolution.

DFT-based circular convolution is usually more efficient:

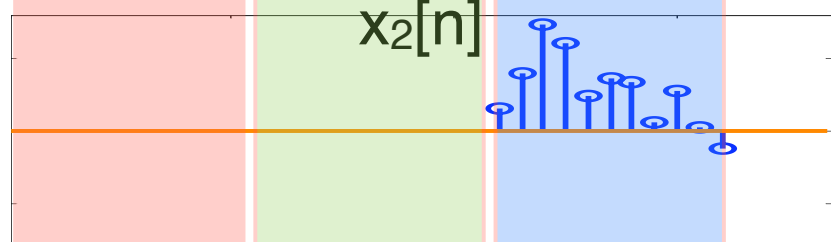
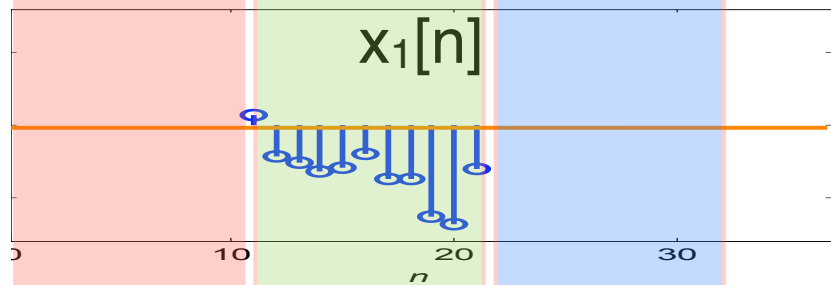
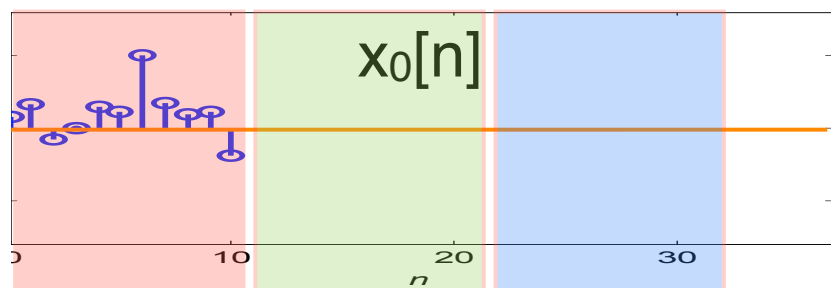
- Zero-pad input segment  $x_r[n]$  to obtain  $x_{r,zp}[n]$ , of length  $N$ .
- Zero-pad the impulse response  $h[n]$  to obtain  $h_{zp}[n]$ , of length  $N$  (this needs to be done only once).
- Compute each output segment using:

$$x_r[n] * h[n] = \mathcal{DFT}^{-1} \{ \mathcal{DFT} \{ x_{r,zp}[n] \} \cdot \mathcal{DFT} \{ h_{zp}[n] \} \}$$

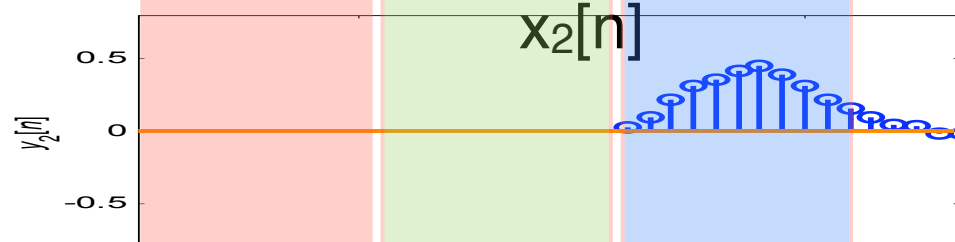
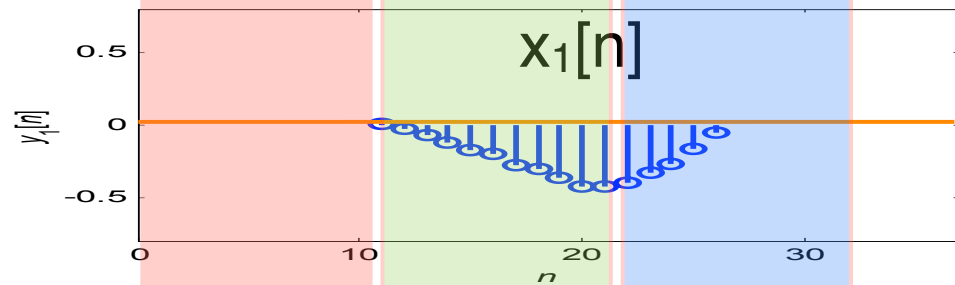
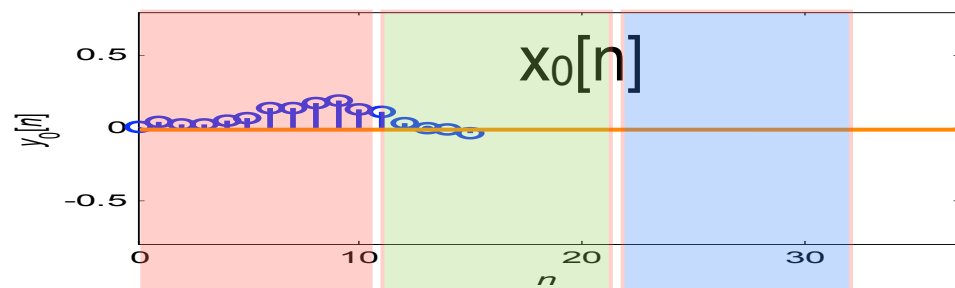
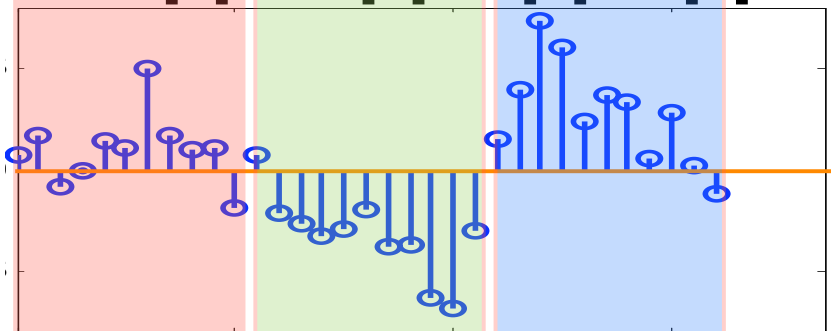
Since output segment  $x_r[n] * h[n]$  starts offset from its neighbor  $x_{r-1}[n] * h[n]$  by  $L$ , neighboring output segments overlap at  $P - 1$  points.

Finally, we just add up the output segments using (1) to obtain the output.

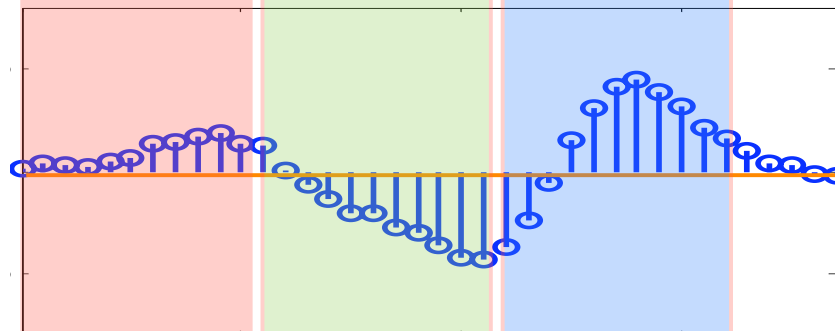
# Example of overlap and add:



$$x[n] = x_0[n] + x_1[n] + x_2[n]$$



$$y[n] = y_0[n] + y_1[n] + y_2[n]$$



# Overlap-Save Method

## *Basic Idea*

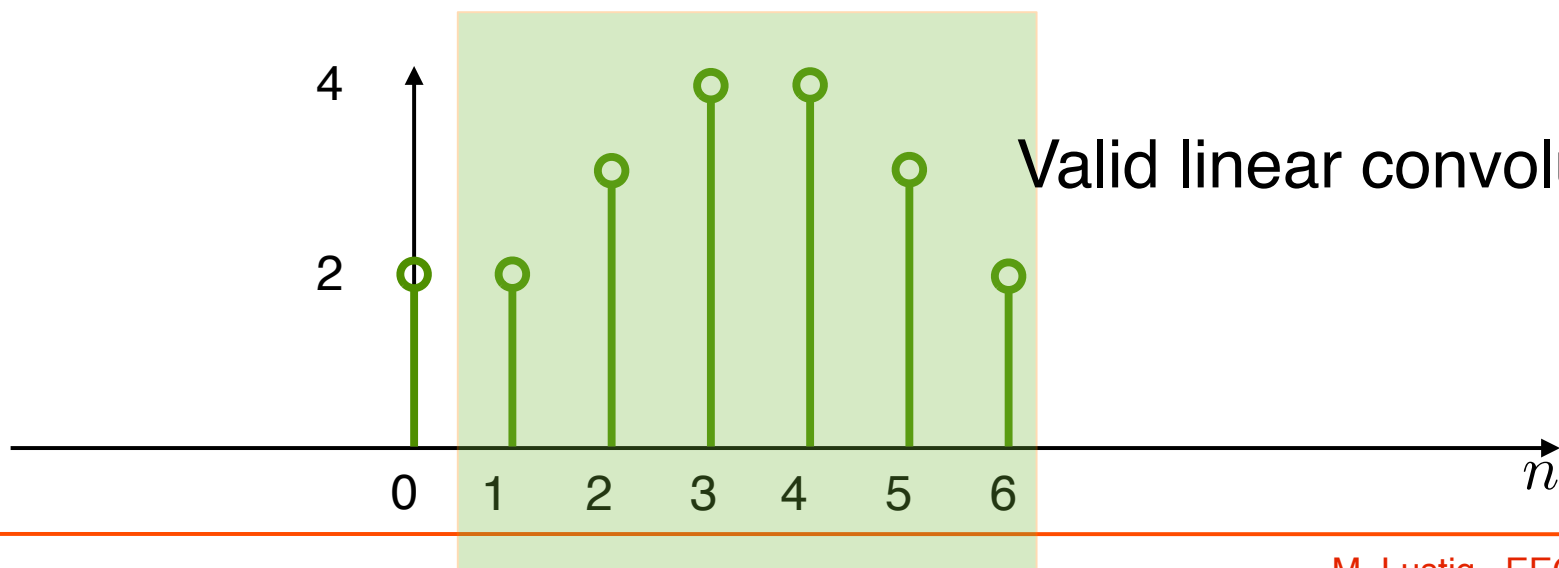
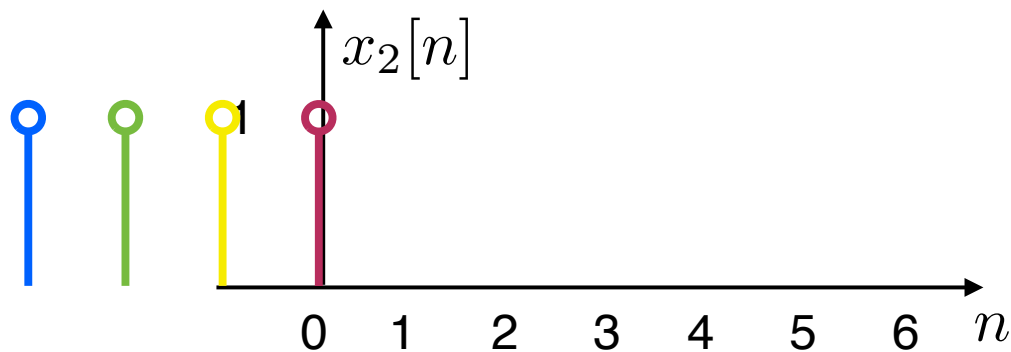
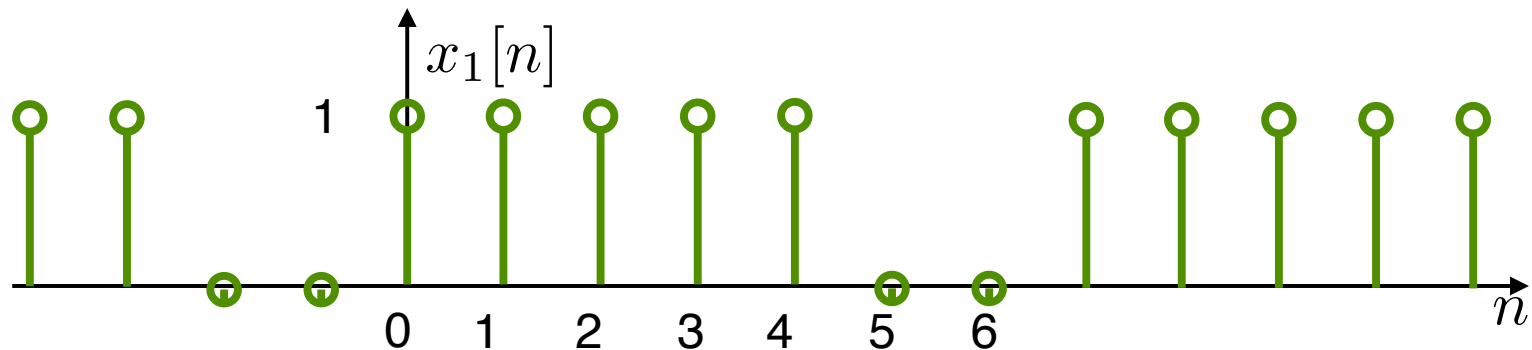
We split the input signal  $x[n]$  into overlapping segments  $x_r[n]$  of length  $L + P - 1$ .

Perform a circular convolution of each input segment  $x_r[n]$  with the impulse response  $h[n]$ , which is of length  $P$  using the DFT. Identify the  $L$ -sample portion of each circular convolution that corresponds to a linear convolution, and save it.

This is illustrated below where we have a block of  $L$  samples circularly convolved with a  $P$  sample filter.

# Recall:

---



# Example of overlap and save:

