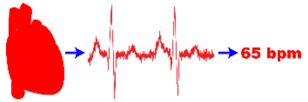
IBEHS 3A03: Biomedical Signals and Systems

Notes for Lecture #7

Tuesday, September 20, 2022

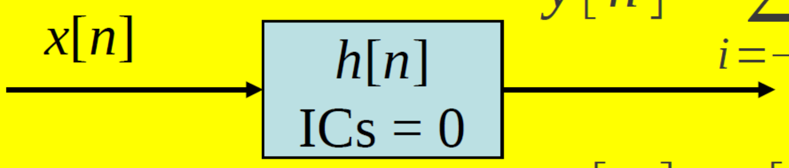
Kamen & Heck Ch. 2

Some slides adapted from material provided by Dr. T. Todd (McMaster) & Dr. M. Fowler (Binghamton Univ.)



Computing a convolution:

Recall from the previous lecture that the convolution operation was derived based on decomposing a signal $x[n]$ into individual scaled and time-shifted impulses $x[i]\,\delta[n-i]$ and passing them through an LTI system with the impulse response $h[n]$. Based on the additivity and homogeneity properties of LTI systems, the output $y[n]$ is the sum of all the scaled and shifted impulse responses $x[i]\,h[n-i]$:



$$y[n]=\sum_{i=-\infty}^{\infty} x[i]\,h[n-i]$$

CONVOLUTION!

$$y[n]=x[n]*h[n]$$

Notation for Convolution

Computing a convolution (cont.):

In cases where $h[n]$ and $x[n]$ are described by relatively simple mathematical expressions, it may be possible to simplify and solve the convolution sum directly.

In other cases, we may have to solve the convolution numerically. As discussed in the previous lecture, a difficulty with computing the convolution directly in the fashion described on slide 2 comes when the input signal $x[n]$ is fairly long, in which case we need to decompose this signal into many scaled and shifted impulse components and then compute the sum over all those components for each time step n.

Computing a convolution (cont.):

An alternative computation approach can be developed based on careful observation of the convolution equation:

$$y[n] = \sum_{i=-\infty}^{\infty} x[i]\, h[n-i]$$

If we reorder the indexing in the impulse response function, we obtain:

$$y[n] = \sum_{i=-\infty}^{\infty} x[i]\, h[-(i-n)]$$

We can now see that the output at each time step *n* is equal to an infinite sum of the input at each time step (indexed by *i* within the sum) multiplied by the “flipped” impulse response *h*[-*i*] shifted to the right by *n* time steps.

Computing a convolution (cont.):

We can perform this operation numerically/graphically with the following steps:

Step 1: Write both as functions of i: $x[i]$ & $h[i]$

Repeat for each n

Step 2: Flip $h[i]$ to get $h[-i]$ (The book calls this "fold")

Step 3: For each output index n value of interest, shift by n to get $h[n - i]$

(Note: positive n gives right shift!!!!)

Step 4: Form product $x[i]h[n - i]$ and sum its elements to get the number $y[n]$

Note that, although this general convolution equation allows for values of n and i from $-\infty$ to $+\infty$, in practice we will only need to consider time steps n and compute the sum over values of i for which the product $x[i]\,h[n-i]$ includes one or more non-zero values.

Computing a convolution (cont.):

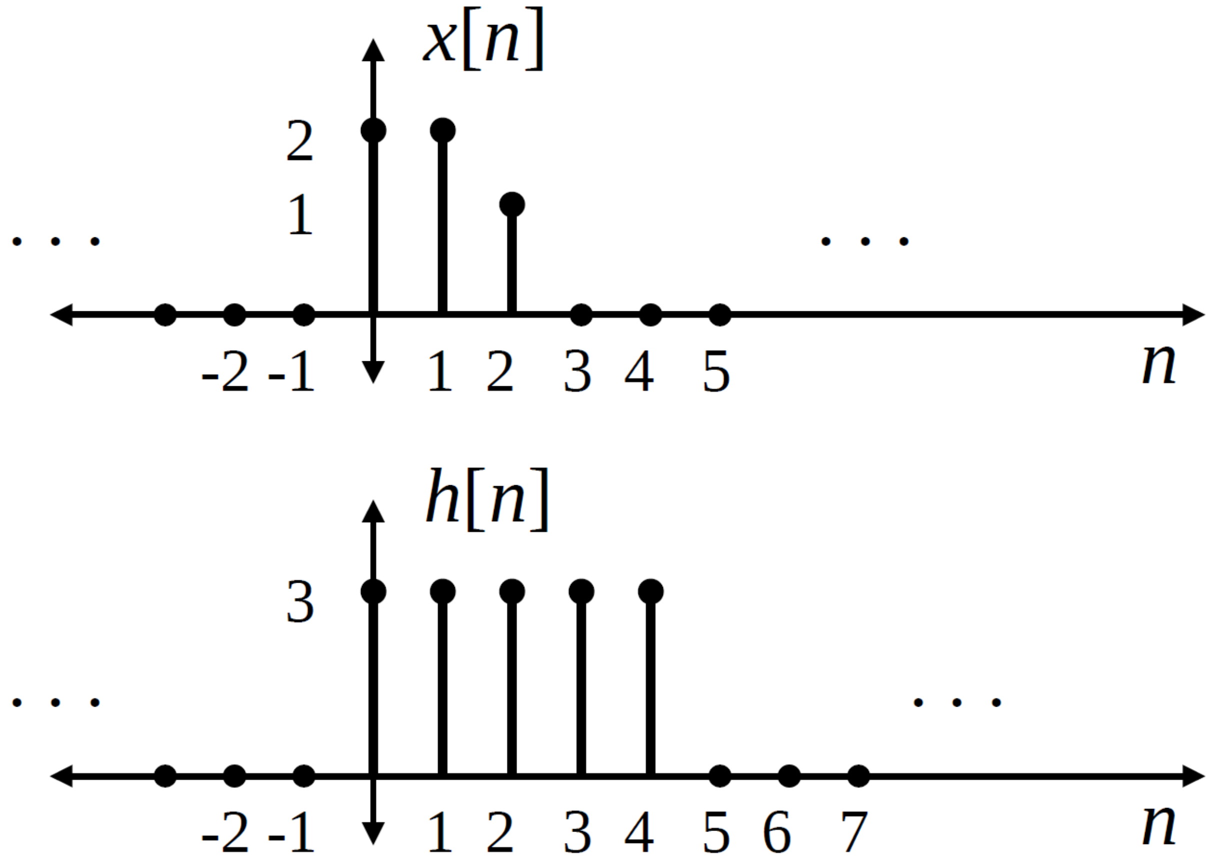
Note also that the convolution of *x*[*n*] with *h*[*n*] is equal to the convolution of *h*[*n*] with *x*[*n*], which is referred to as the **commutativity property** of convolution:

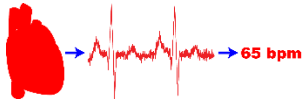
$$y[n]=\underbrace{\sum_{i=-\infty}^{\infty} x[i]h[n-i]}_{x[n]*h[n]}=\underbrace{\sum_{i=-\infty}^{\infty} h[i]x[n-i]}_{h[n]*x[n]}$$

Because of this property, we can actually apply the “flipping and shifting” to either the impulse response *h*[*n*] or the input signal *x*[*n*], whichever makes the graphical/numerical solution method easier.

Computing a convolution (cont.):

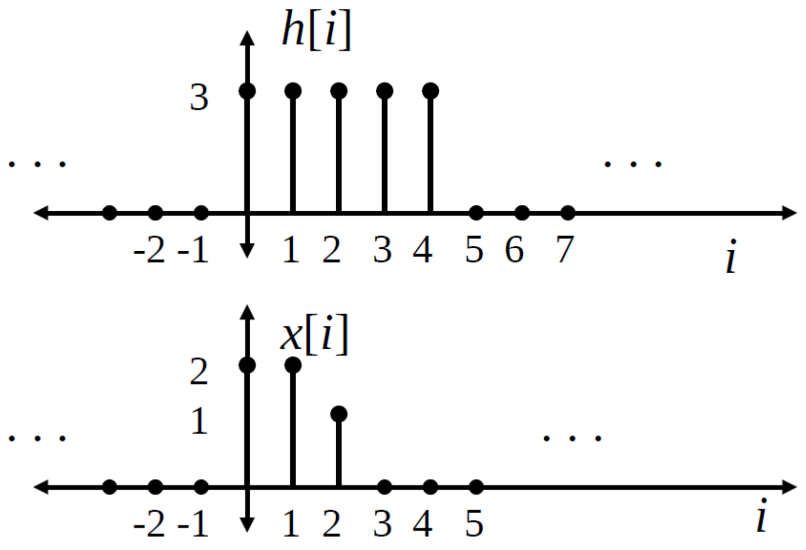
Consider the following graphical example for the given $x[n]$ and $h[n]$, where we will choose to flip and shift $x[n]$ rather than $h[n]$:





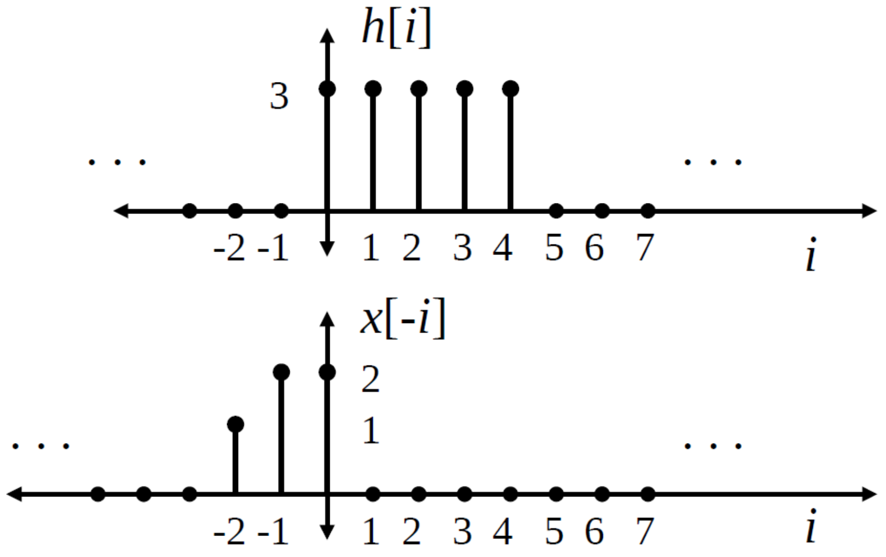
Step 1: Write both as functions of i: $x[i]$ & $h[i]$

$$y[n]=\sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$



Step 2: Flip $x[i]$ to get $x[-i]$

"Commutativity" says we can flip either $x[i]$ or $h[i]$ and get the same answer…
Here I flipped $x[i]$

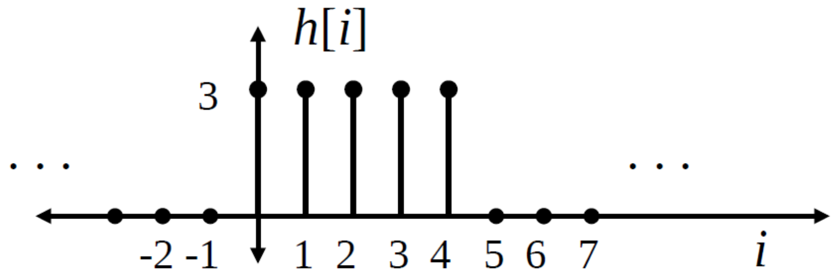


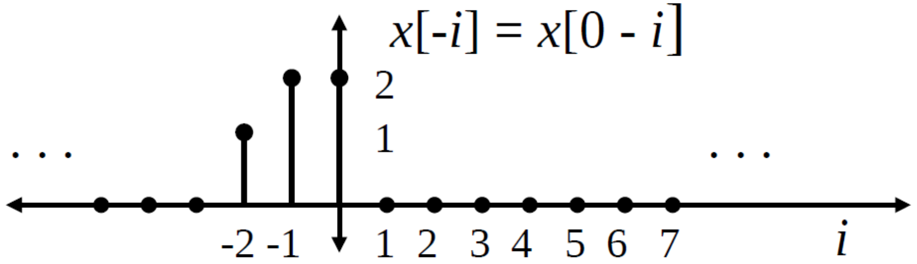
$$y[n]=\sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$

Step 3: For ***n* = 0**, shift by n to get $x[n - i]$

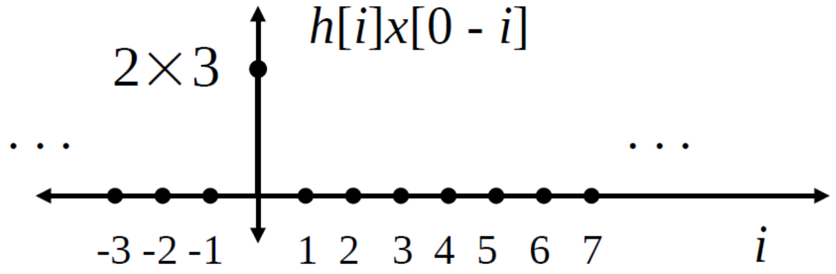
For $n = 0$ case there is no shift!
i.e.,

$$x[0 - i] = x[-i]$$





Step 4: For ***n* = 0**, Form the product $x[i]h[n – i]$ and sum its elements to give $y[n]$



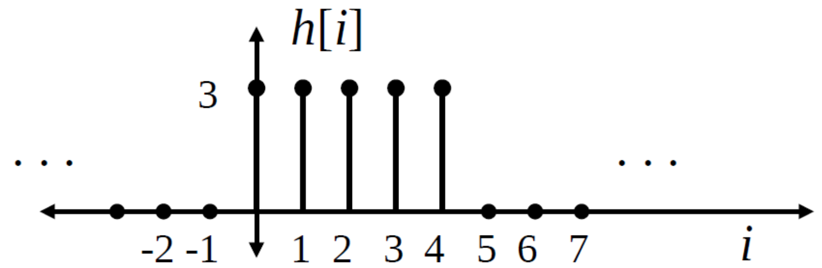
Sum over *i* ⇒ $\boxed{y[0] = 6}$

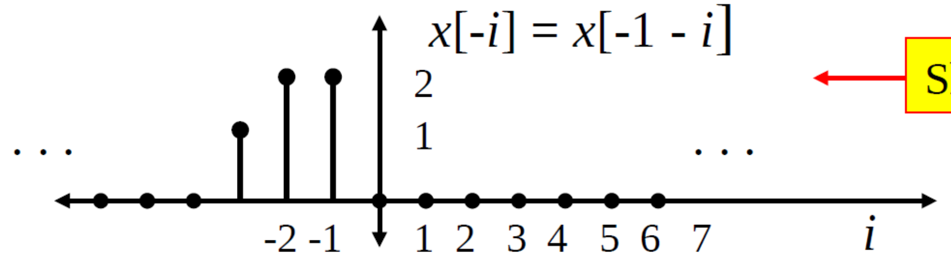
Steps 3&4 for all $n < 0$

$$y[n] = \sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$

Step 3: For $n < 0$, shift by n to get $x[n - i]$

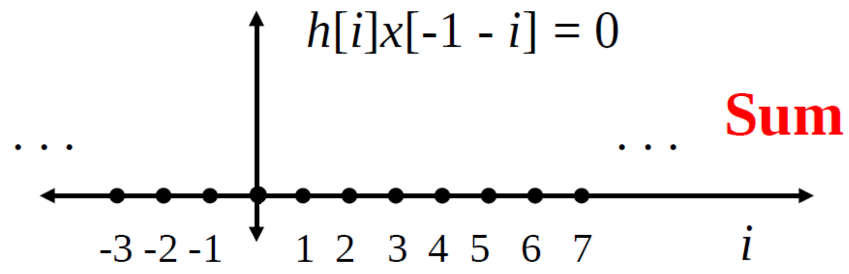
Negative n gives a left-shift





Shown here for n = -1

Step 4: For $n < 0$, Form the product $x[i]h[n - i]$ and sum its elements to give $y[n]$

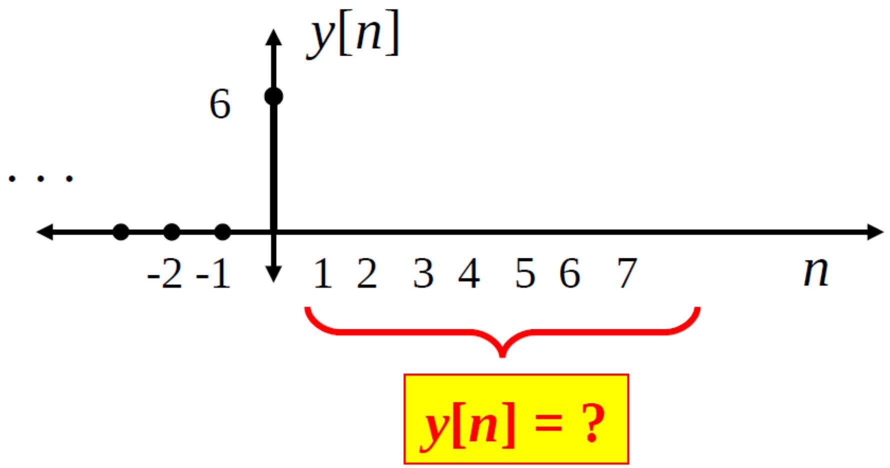


Sum over i ⇒ $y[n] = 0 \quad \forall n < 0$

65 bpm

$$y[n] = \sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$

So… what we know so far is that: $y[n] = \begin{cases} 0, & \forall\, n < 0 \\ 6, & n = 0 \end{cases}$



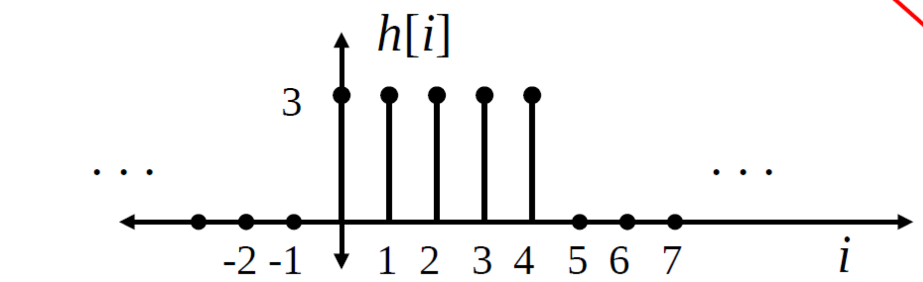
So now we have to do Steps 3 & 4 for $n > 0$…

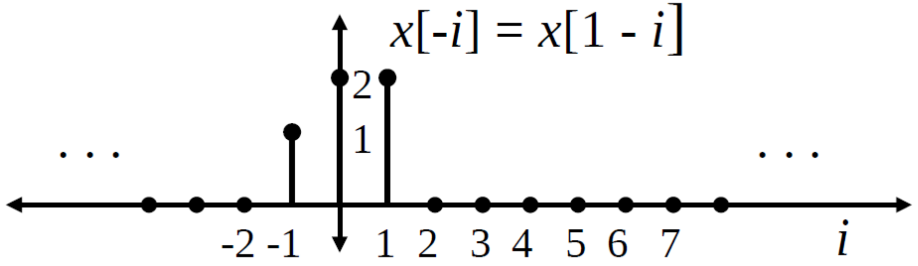
Steps 3&4 for $n = 1$

$$y[n] = \sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$

Step 3: For $n = 1$, shift by n to get $x[n - i]$

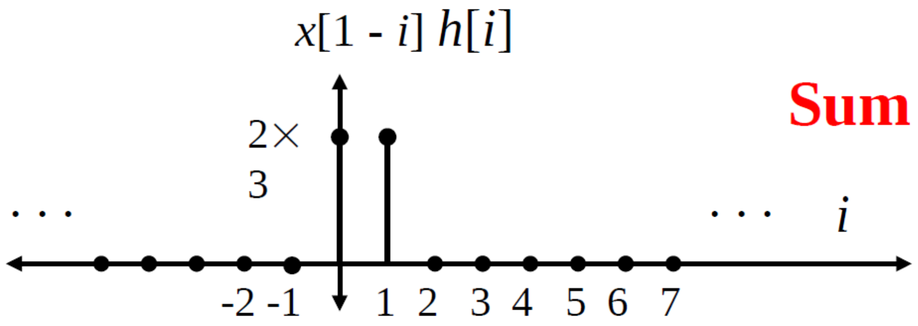
Positive n gives a Right-shift





shifted to the right by one

Step 4: For $n = 1$, Form the product $x[i]h[n - i]$ and sum its elements to give $y[n]$



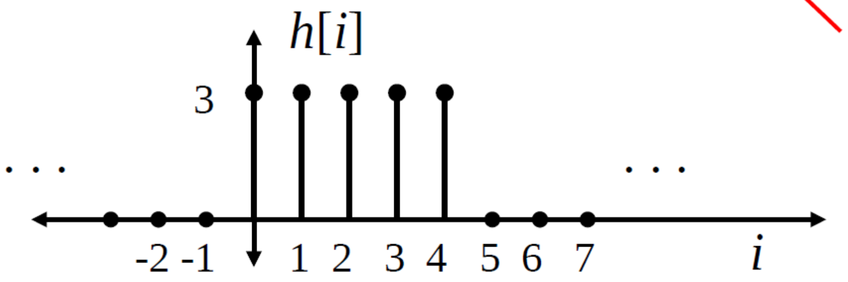
Sum over i ⇒ $y[1] = 6 + 6 = 12$

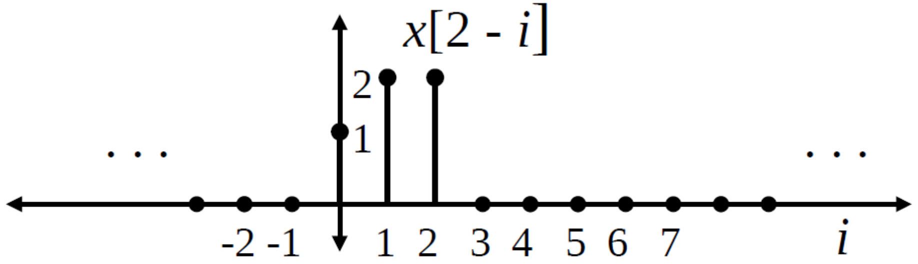
Steps 3&4 for $n = 2$

$$y[n] = \sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$

Step 3: For $n = 2$, shift by n to get $x[n - i]$

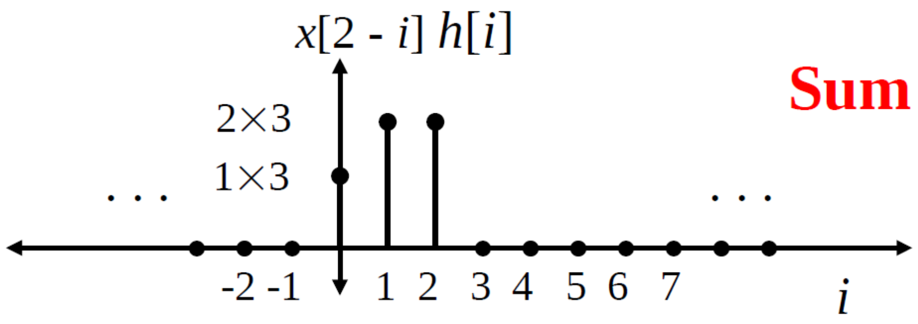
Positive n gives a Right-shift





shifted to the right by two

Step 4: For $n = 2$, Form the product $x[i]h[n - i]$ and sum its elements to give $y[n]$



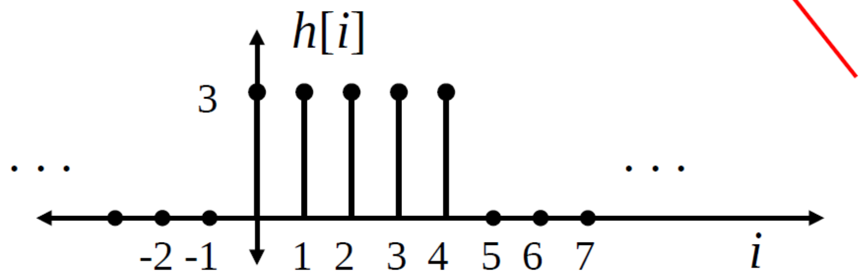
Sum over i ⇒ $y[2] = 3 + 6 + 6 = 15$

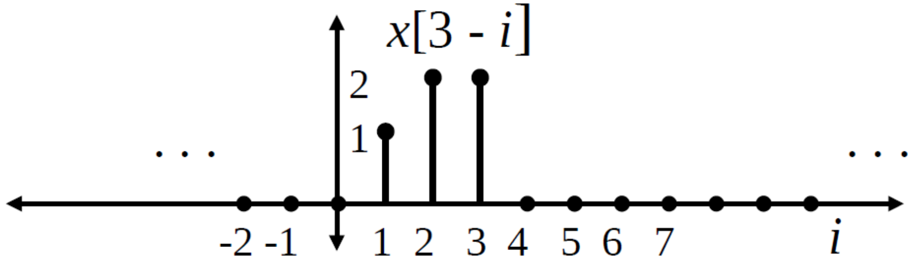
Steps 3&4 for $n = 3$

$$y[n]=\sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$

Step 3: For $n = 3$, shift by n to get $x[n - i]$

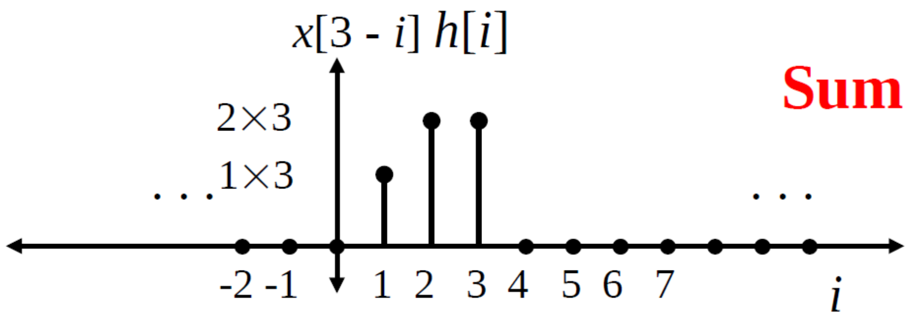
Positive n gives a Right-shift





shifted to the right by three

Step 4: For $n = 3$, Form the product $x[i]h[n – i]$ and sum its elements to give $y[n]$



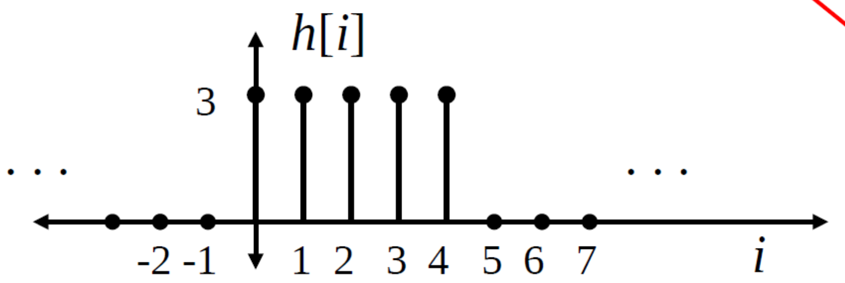
Sum over $i \Rightarrow$ $y[3] = 3 + 6 + 6 = 15$

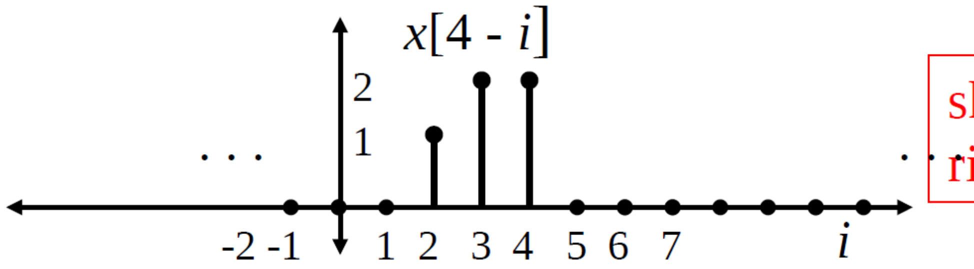
Steps 3&4 for n = 4

$$y[n] = \sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$

Step 3: For $n = 4$, shift by n to get $x[n - i]$

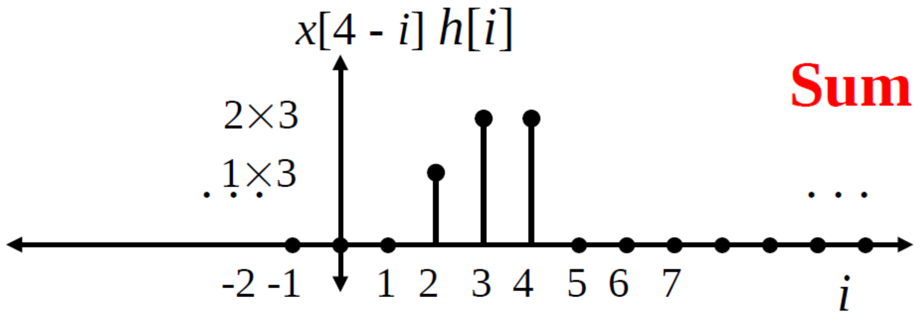
Positive n gives a Right-shift





shifted to the right by four

Step 4: For $n = 4$, Form the product $x[i]h[n - i]$ and sum its elements to give $y[n]$



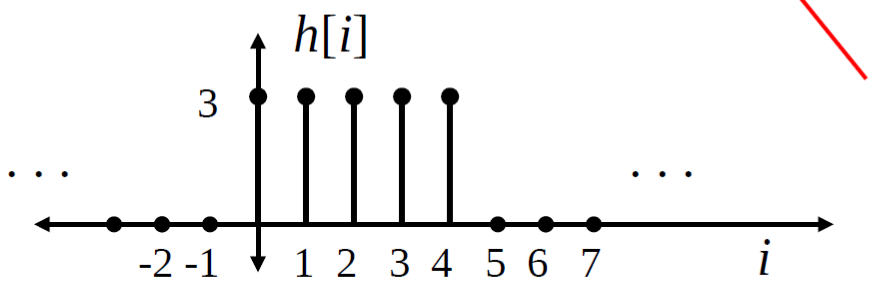
Sum over i ⇒ $y[4] = 3 + 6 + 6 = 15$

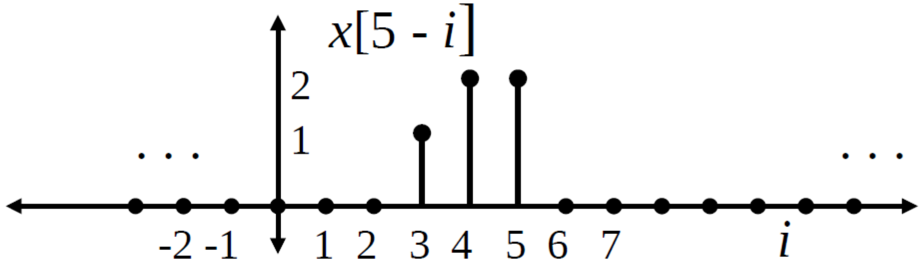
Steps 3&4 for $n = 5$

$$y[n] = \sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$

Step 3: For $n = 5$, shift by n to get $x[n - i]$

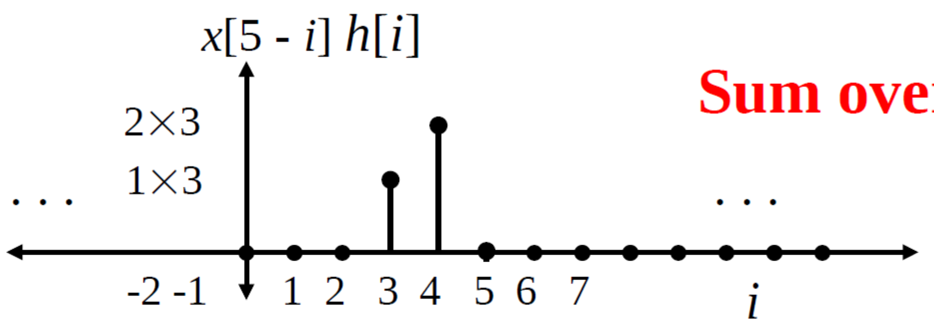
Positive n gives a Right-shift





shifted to the right by five

Step 4: For $n = 5$, Form the product $x[i]h[n - i]$ and sum its elements to give $y[n]$



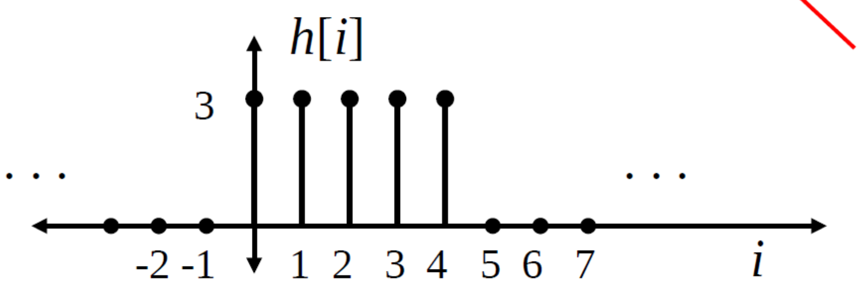
Sum over i ⇒ $y[5] = 3 + 6 = 9$

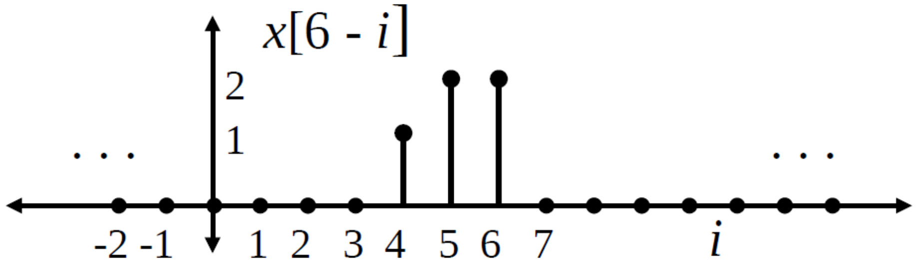
Steps 3&4 for $n = 6$

$$y[n] = \sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$

Step 3: For $n = 6$, shift by n to get $x[n - i]$

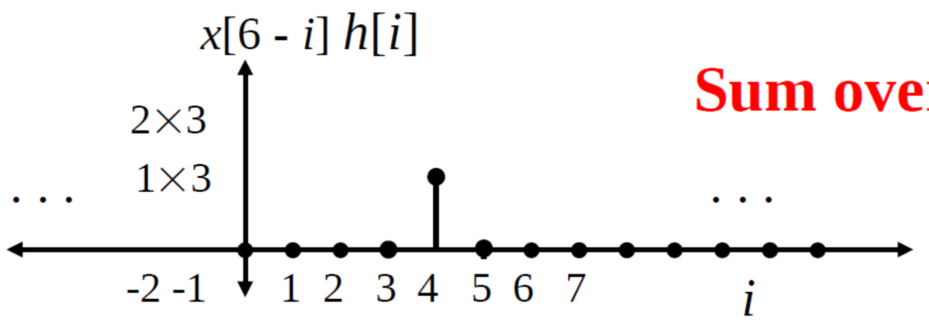
Positive n gives a Right-shift





shifted to the right by six

Step 4: For $n = 6$, Form the product $x[i]h[n - i]$ and sum its elements to give $y[n]$



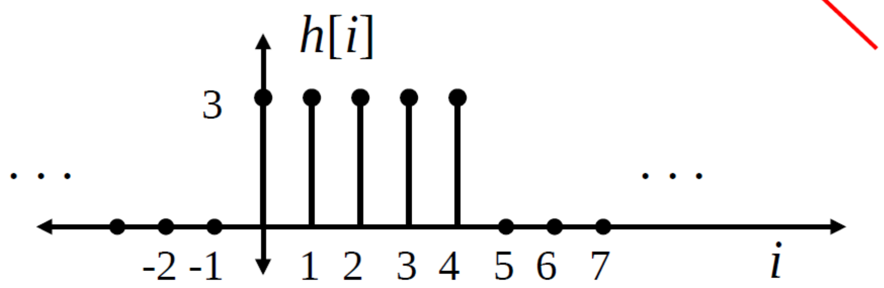
Sum over i ⇒ $y[6] = 3$

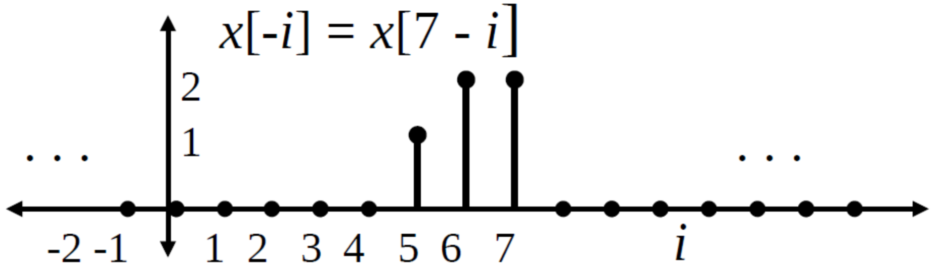
Steps 3&4 for all $n > 6$

$$y[n]=\sum_{i=-\infty}^{\infty} h[i]\,x[n-i]$$

Step 3: For $n > 6$, shift by n to get $x[n - i]$

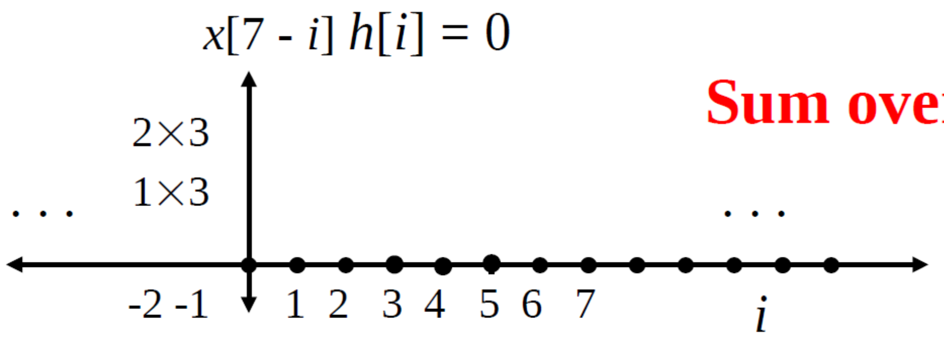
Positive n gives a Right-shift





shifted to the right by seven

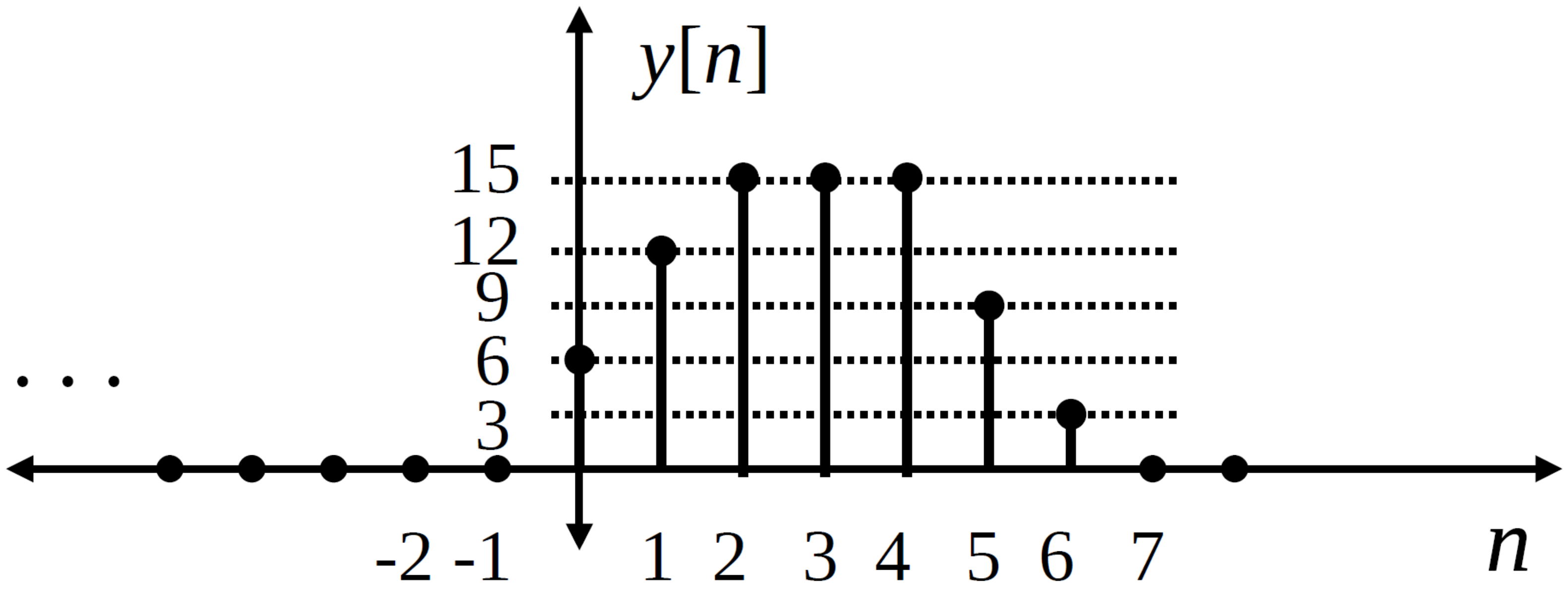
Step 4: For $n > 6$, Form the product $x[i]h[n - i]$ and sum its elements to give $y[n]$



Sum over i ⇒ $y[n]=0 \quad \forall\, n>6$

Computing a convolution (cont.):

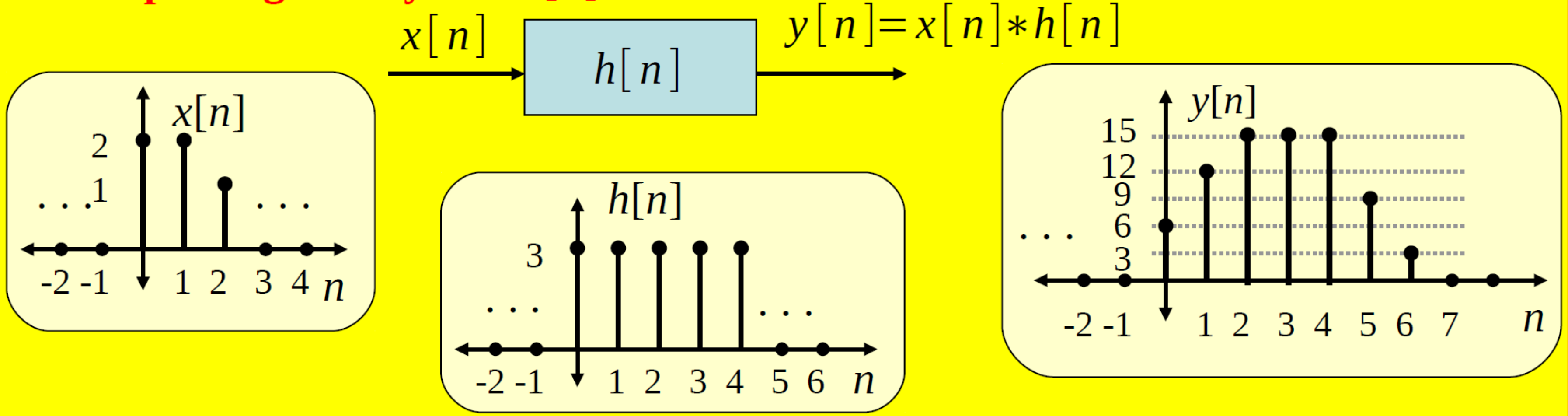
Now we can put all these values together to give:



Note that ramps up and down at the start and end of the output are quite common in convolutions and are referred to as *transients*.

Computing a convolution (cont.):

<u>BIG PICTURE</u>: So … what we have just done is found the zero-state output of a system having an impulse response given by this $h[n]$ when the input is given by this $x[n]$:



Computing a convolution (cont.):

Some important properties of the convolution operation are:

1. Commutativity $x[n]*h[n]=h[n]*x[n]$

$\Rightarrow$ You can choose which signal to "flip"

2. Associativity $x[n]*(v[n]*w[n])=(x[n]*v[n])*w[n]$

$\Rightarrow$ Can change order $\rightarrow$ sometimes one order is easier than another

3. Distributivity $x[n]*(h_1[n]+h_2[n])=x[n]*h_1[n]+x[n]*h_2[n]$

$\Rightarrow$ may be easier to split complicated system $h[n]$ into sum of simple ones

OR.. $\Rightarrow$ we can split complicated input into sum of simple ones

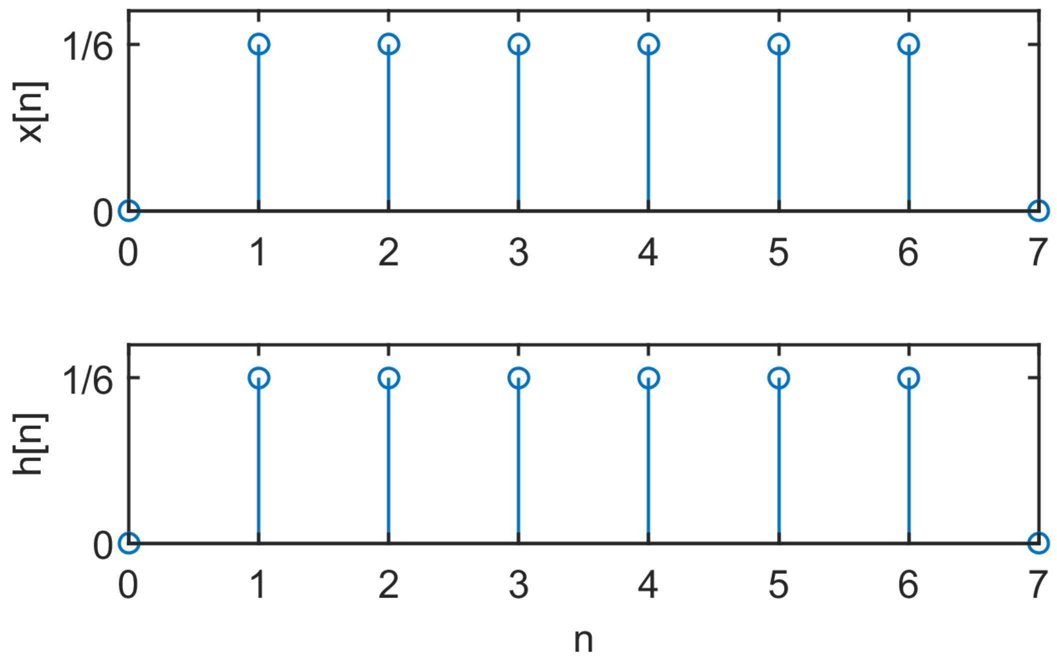
(nothing more than "linearity")

4. Convolution with impulses $x[n]*\delta[n-q]=x[n-q]$

65 bpm

Another D-T convolution example:

In the lecture I will demonstrate how to solve the convolution of the following signal $x[n]$ and impulse response $h[n]$ using this graphical/numerical method.



65 bpm