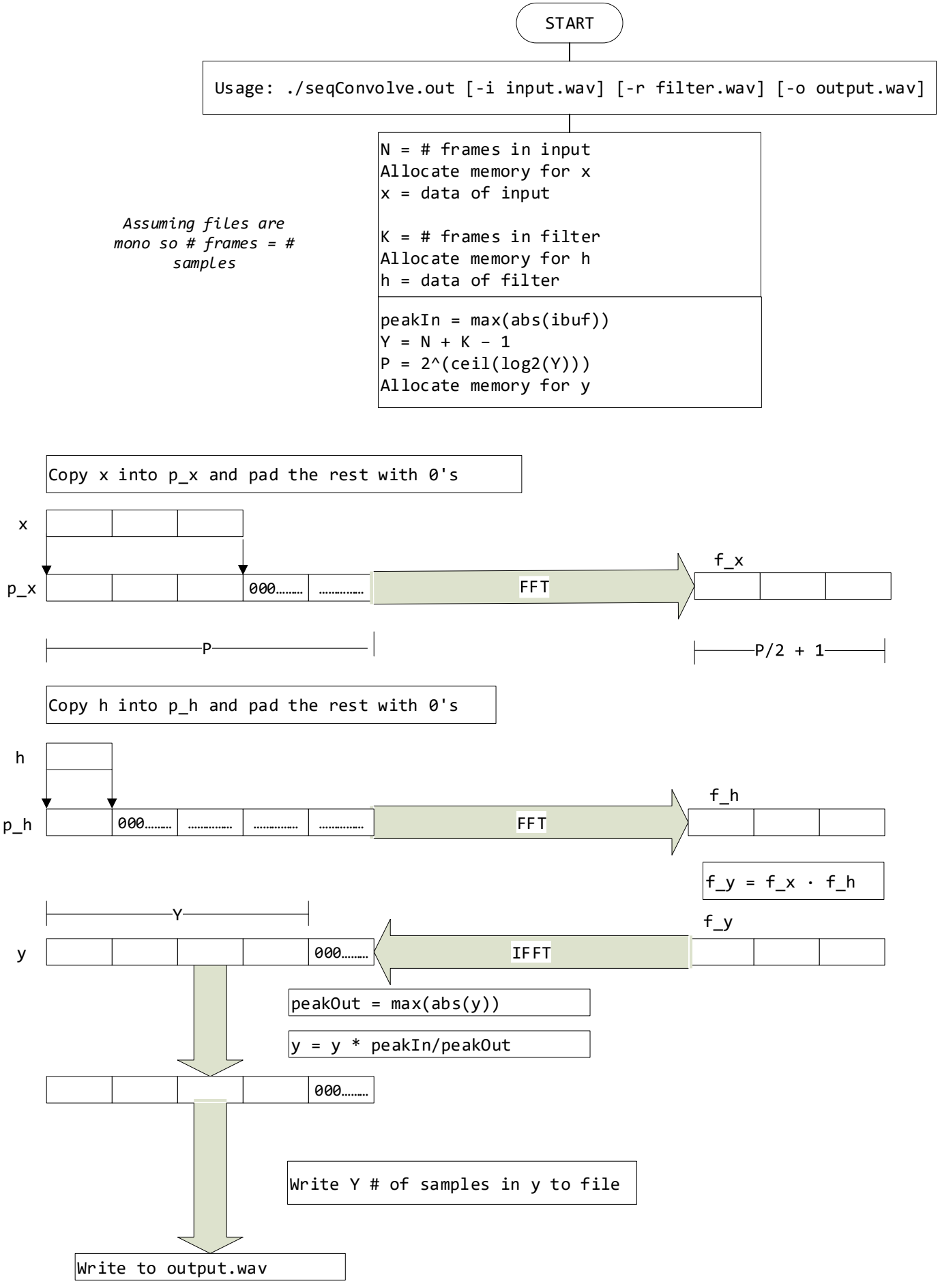

START
Usage: ./seqConvolve.out [-i input.wav] [-r filter.wav] [-o output.wav]
N = # frames in input
Allocate memory for x
x = data of input
K = # frames in filter
Allocate memory for h
h = data of filter
Assuming files are mono so # frames = # samples
peakIn = max(abs(ibuf))
Y = N + K – 1
P = 2^(ceil(log2(Y)))
Allocate memory for y
Copy x into p_x and pad the rest with 0's
x
p_x
000……
……
FFT
f_x
P
P/2 + 1
Copy h into p_h and pad the rest with 0's
h
p_h
000……
……
……
……
FFT
f_h
f_y = f_x · f_h
Y
y
000……
IFFT
f_y
peakOut = max(abs(y))
y = y * peakIn/peakOut
000……
Write Y # of samples in y to file
Write to output.wav

x = input
N = size of input

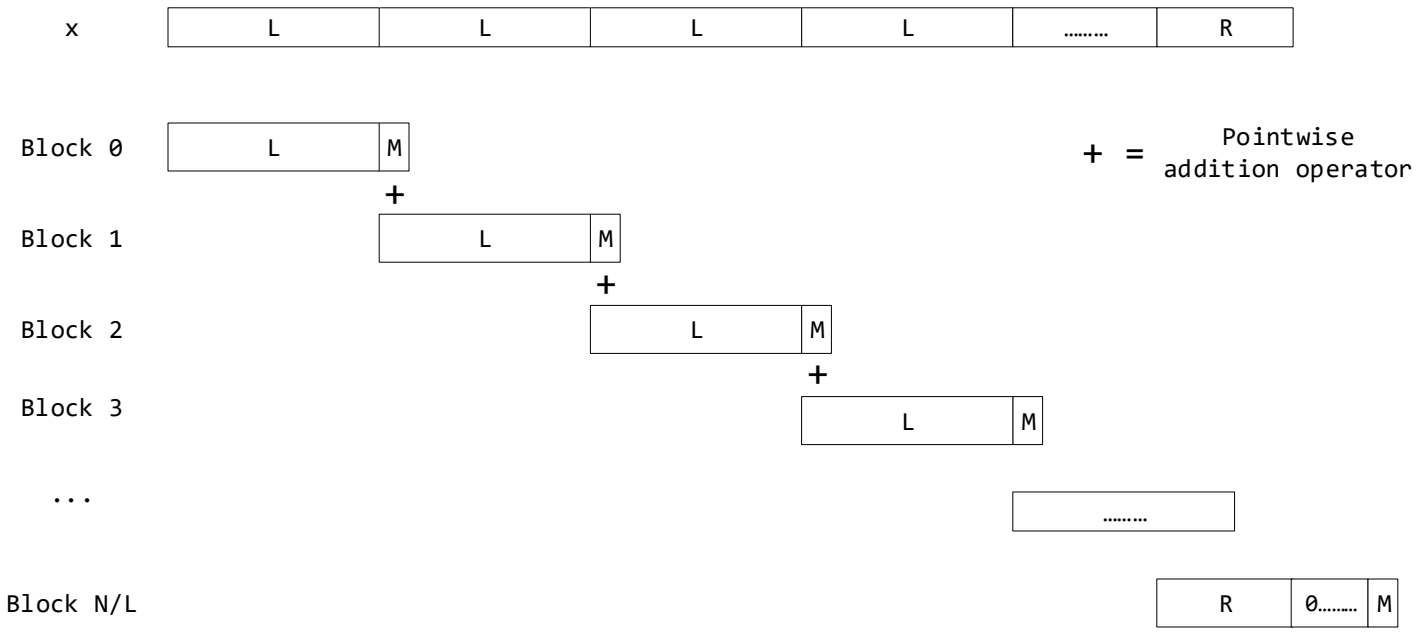
h = filter
K = size of filter

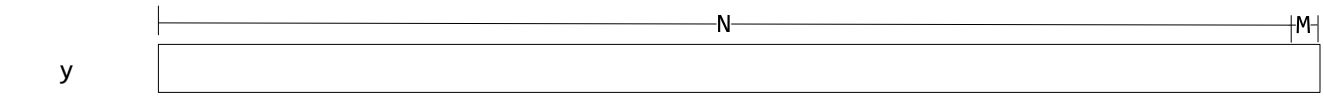
Y = N + K – 1 = length of output

L = input block amount
R = remainder = N % L

M = K – 1
Block Size = L + M

L can be any length and can change in size. For simplicity, here L is constant and L is chosen so that the block size is a factor of 2

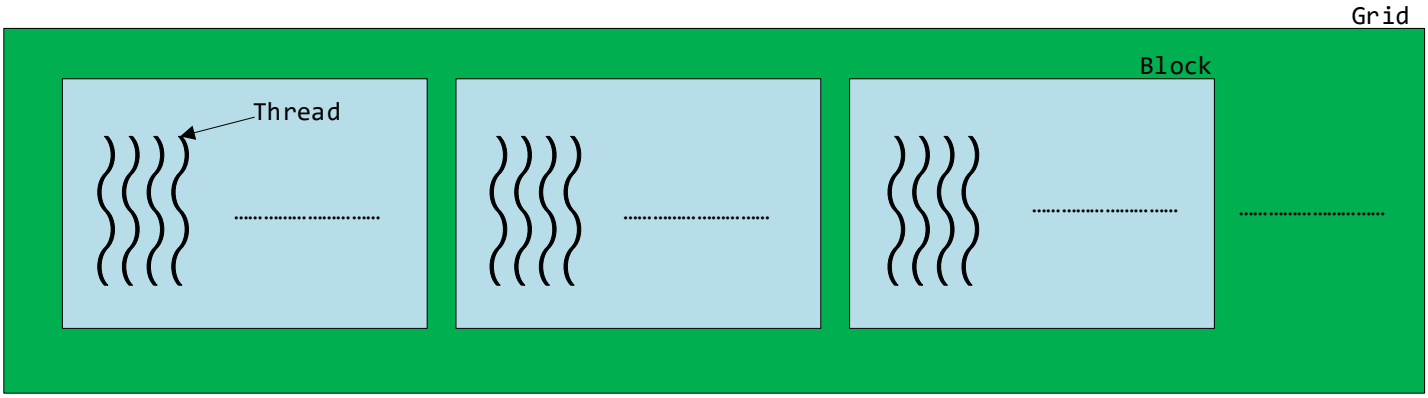


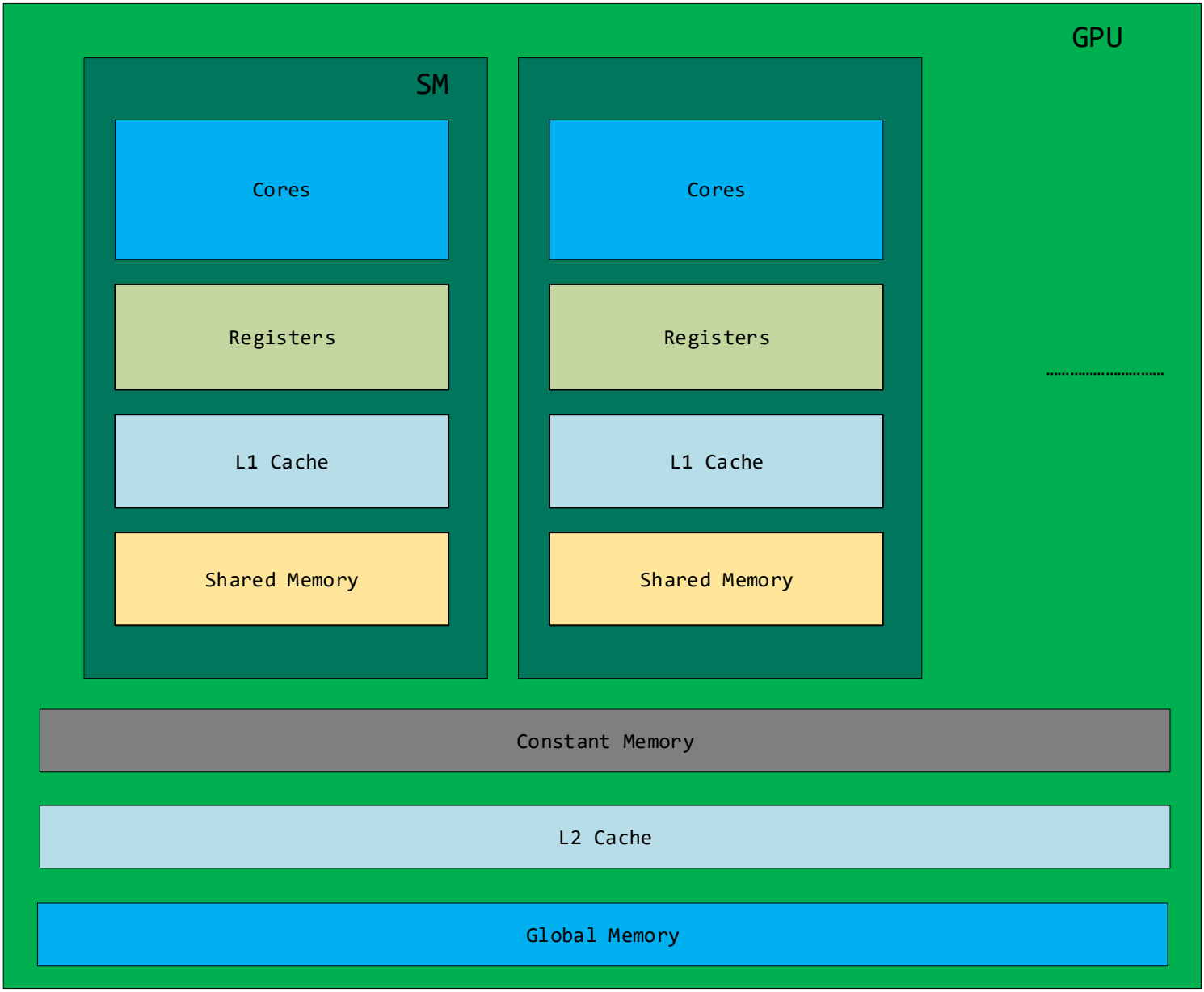


1 – Copy L elements from the input to the block
2 – Pad the block with M number of 0's
3 – Convolve the block
4 – Repeat for all blocks until the last block
5 – For the last block, copy R elements and pad the remainder of the block with 0's
6 – Reconstruct the output from the collection of blocks

Excluding the first and last block, M number of elements are pointwise added together at both the beginning and the end of each block.

For the first block, M elements at the end are pointwise added. For the last block, M elements at the beginning are pointwise added.


Grid
Block
Thread


GPU
SM
Cores
Registers
L1 Cache
Shared Memory
Cores
Registers
L1 Cache
Shared Memory
Constant Memory
L2 Cache
Global Memory

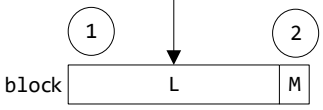
x = input
N = size of input

h = filter
K = size of filter

L = input block amount
R = remainder = N % L

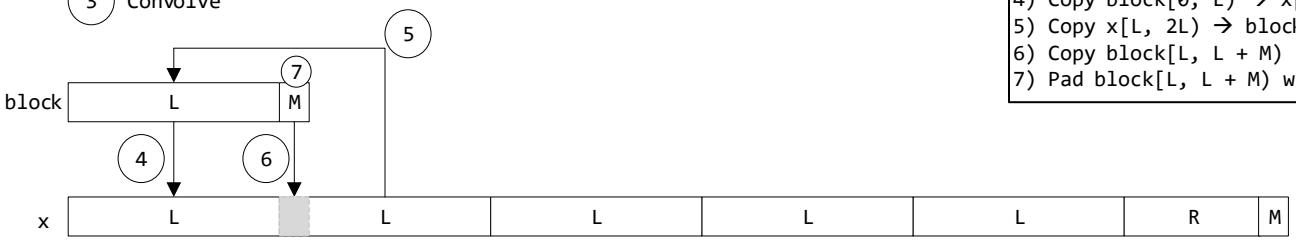
M = K – 1
Block Size = L + M

L can be any length and can change in size. For simplicity, here L is constant and L is chosen so that the block size is a factor of 2



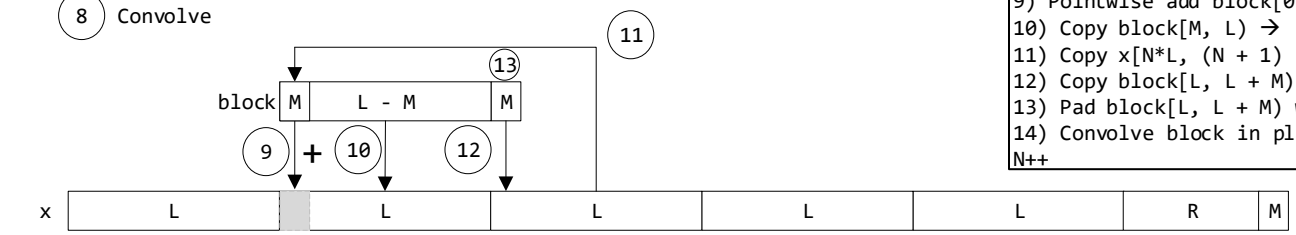
N = 0
1) Copy x[0,L) → block[0]
2) Pad block[L, L + M) with 0's
3) Convolve block in place with filter
4) Copy block[0, L) → x[0]
5) Copy x[L, 2L) → block[0]
6) Copy block[L, L + M) → x[L]
7) Pad block[L, L + M) with 0's

3 Convolve



N = 1
8) Convolve block in place with filter
9) Pointwise add block[0, M) + x[N*L)
10) Copy block[M, L) → x[N*L + M]
11) Copy x[N*L, (N + 1) *L) → block[0]
12) Copy block[L, L + M) → x[N*L]
13) Pad block[L, L + M) with 0's
14) Convolve block in place with filter
N++

8 Convolve

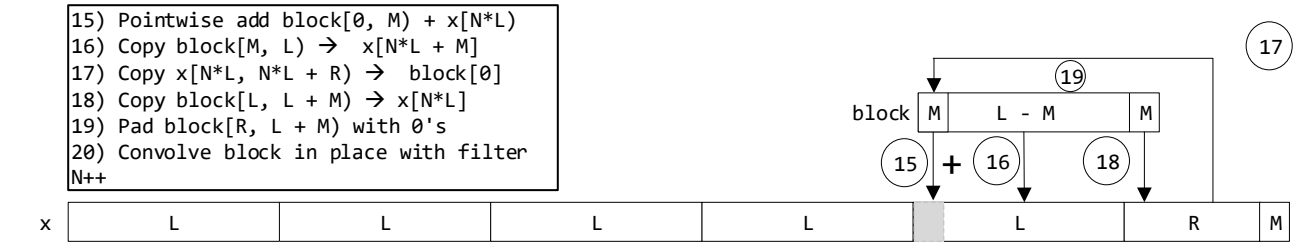


14 Convolve

Repeat steps 9 – 14 until no more whole blocks size L and increment N each time

.....

15) Pointwise add block[0, M) + x[N*L)
16) Copy block[M, L) → x[N*L + M]
17) Copy x[N*L, N*L + R) → block[0]
18) Copy block[L, L + M) → x[N*L]
19) Pad block[R, L + M) with 0's
20) Convolve block in place with filter
N++



20 Convolve

21) Pointwise add block[0, M) + x[N*L)
22) Copy block[M, R + M) à x[N*L + M]