Implementation of real time partitioned convolution on a DSP board

Enrico Armelloni, Christian Giottoli, Angelo Farina.

Industrial Engineering Department - University of Parma
Parco Area delle Scienze 181/A, 43100 Parma – Italy
enrico.armelloni@unipr.it

Outline:

- Linear convolution;
- Overlap & Save method;
- Uniformly-partitioned Overlap & Save method;
- Software implementation on a DSP board;

Convolution (1):

Convolution of a continuous input signal *x(t)* with a linear filter characterized by an Impulse Response *h(t)* yields an output signal *y(t)*:

$$y(t) = x(t) \otimes h(t) = \int_{-\infty}^{\infty} x(t-\tau) \cdot h(\tau) \cdot d\tau$$

If the input signal and the Impulse Response are digitally sampled ($t = i \cdot \Delta t$) and the Impulse Response has finite length *N*, we can write:

$$y(i) = \sum_{j=0}^{N-1} x(i-j) \cdot h(j)$$

Convolution (2):

$$y(i) = \sum_{j=0}^{N-1} x(i-j) \cdot h(j)$$

Multiply and ACcumulate

```
y:=0;
FOR n:=0 TO N-1 DO
    y:= y + a[n]·x[n];
```

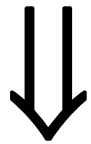
On a DSP board this instruction is performed in one cycle

- **Clock core = 100 MHz**
- **Sample frequency f_S = 48 KHz**

⇒ **Upper limit is 2000 MAC per sample**

Convolution (3):

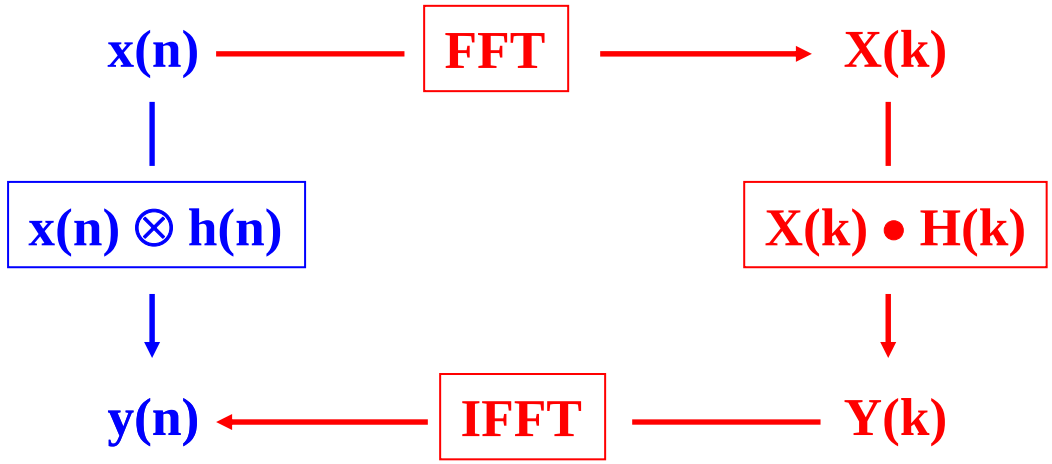
If Impulse Response is very long, i.e. 2 second or plus, like an IR measured inside of a theatre,



h(t) is length 96000 points or plus @ 48kHz.

Filtering in the frequency domain:

Could be better operate in the frequency domain



Problems ➔
- **Filtering can be performed only when all data are available**
- **Order of FFT is too high.**

Solution ➔
- **Overlap & Save algorithm.**

Overlap & Save algorithm (1):

It can be shown that multiplication of two DFTs corresponds to a circular convolution of their time domain sequences.

To implement a FIR filter, a linear convolution is required.

A procedure for converting a circular convolution into a linear convolution is the Overlap&Save algorithm.

Long duration signal sections $x_m(n)$ are overlapped of (Q-1) samples, where Q is the length of the Impulse Response $h(n)$.

Overlap & Save algorithm (2):

1. Perform N-point FFT of the IR *h(n)* and store it:

$$h(n) = \begin{cases} h(n) & n = 0,1,\ldots,Q-1 \\ 0 & n = Q, Q+1,\ldots,N-1 \end{cases}$$

2. Select N points from x(n) based on following expression:

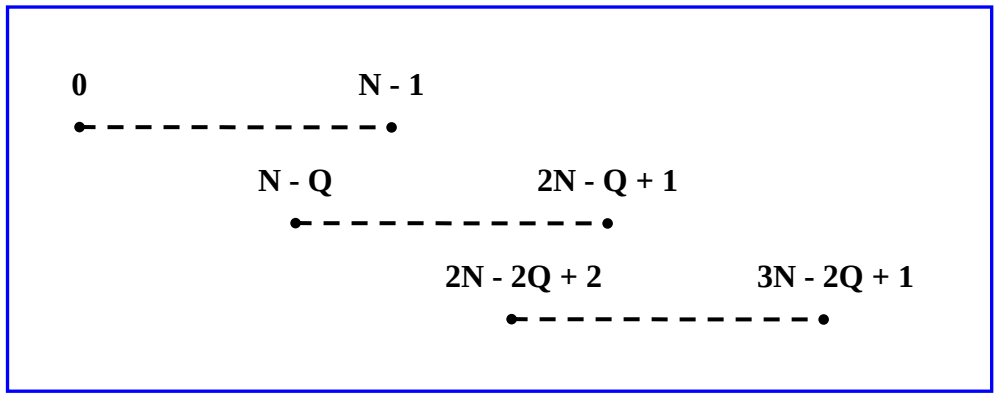
$$x_m(n) = x\left[n + (m-1)(N-Q+1) - (Q-1)\right]$$

where: n = 0,1,2,…,N-1

m = 1,2,3,…

N = FFT length

Q = IR length



Overlap & Save algorithm (3):

3. Multiply the stored frequency response of *h(n)* by the FFT of input signal batch *m*.

4. Perform an N-point IFFT of the product.

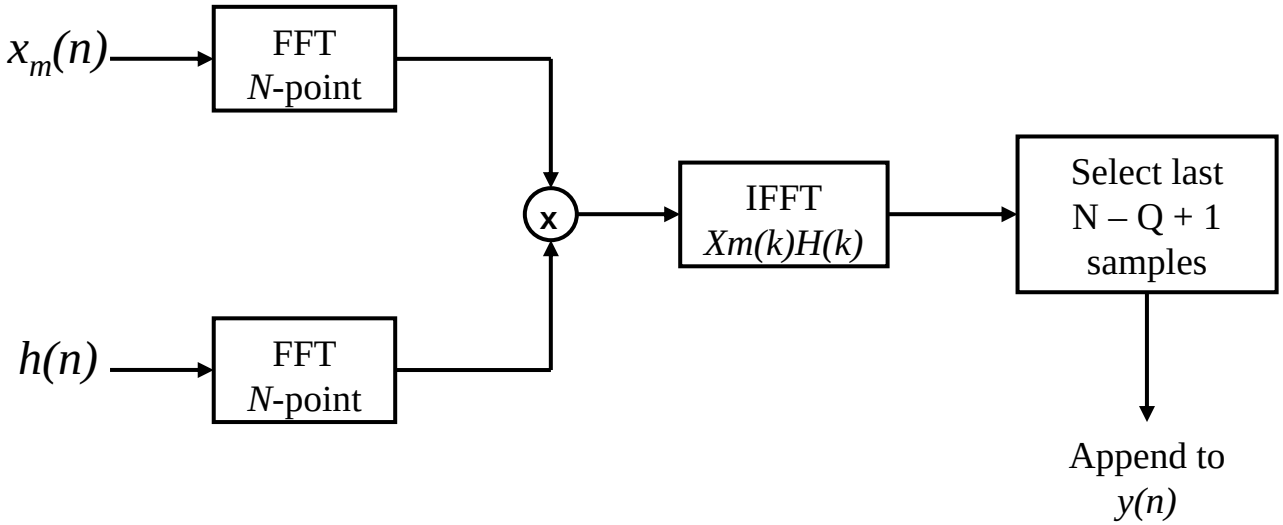
5. Discard the first (Q-1) points from each successive output of step 4, and

$y_1(n)$	$n = Q-1,\ldots,N-1$
$y_2[n-(N-Q+1)]$	$n = N,\ldots,2N-Q$
.	.
.	.
$y_m[n-(m-1)(N-Q+1)]$	$n = (m-1)(N-Q+1)+(Q-1),$
	$\ldots,(m-1)(N-Q+1)+(N-1)$
.	.
.	.

append the remaining outputs to *y(n)*: $y(n) = y_1(n), y_2(n),\ldots, y_m(n),\ldots$

Overlap & Save algorithm (4):

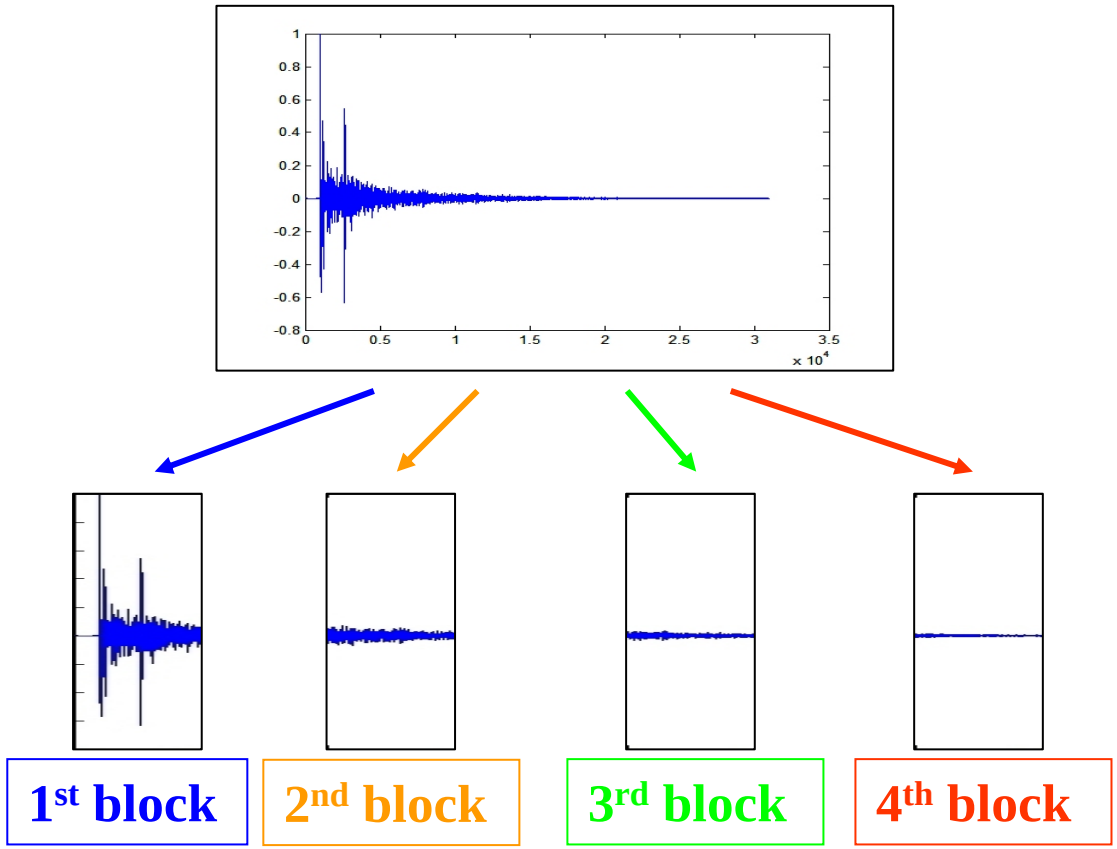
Overlap & Save convolution process:



Problems ➔
- **Latency between Input and Output data is too high.**
- **Management problem with internal memory of the DSP.**

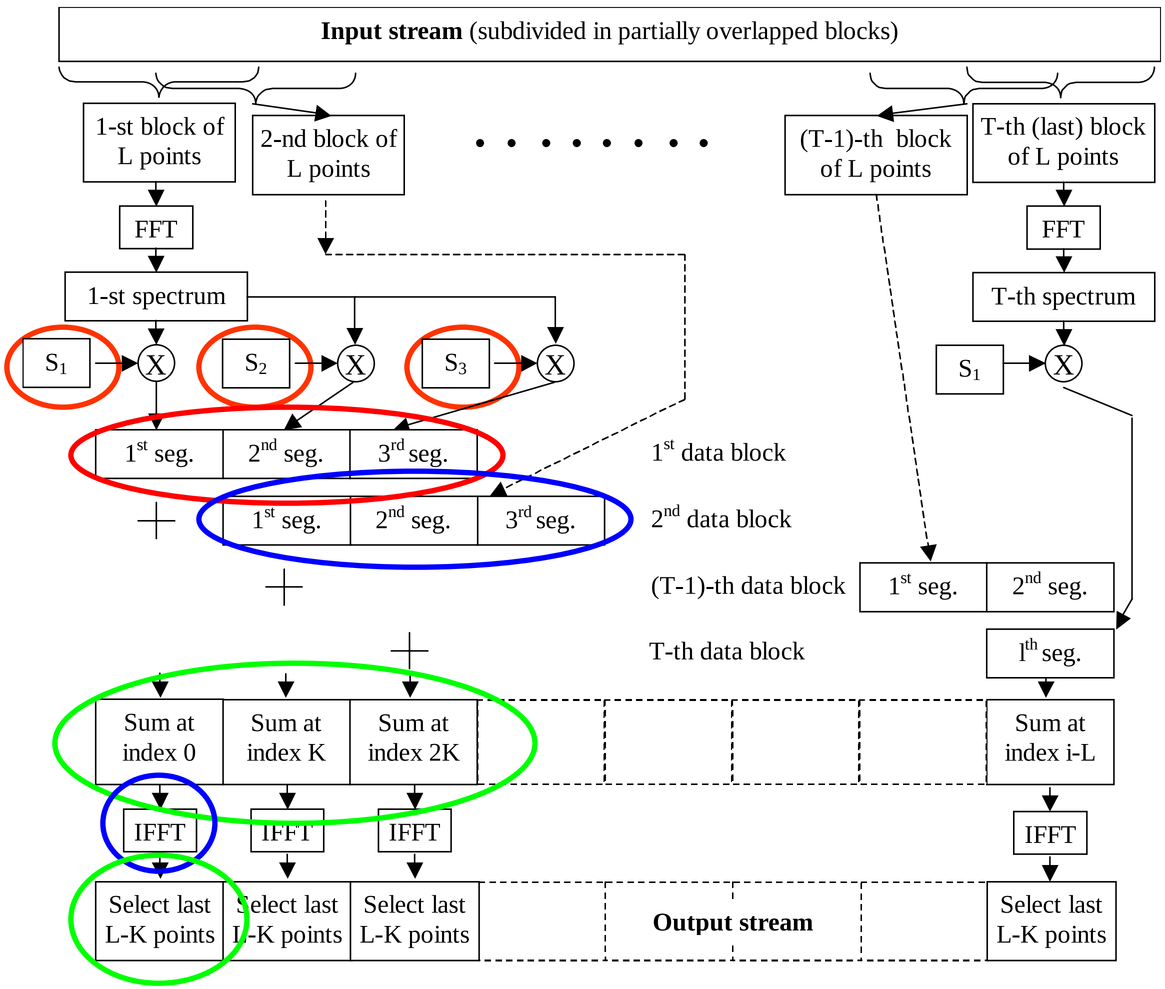
Solution ➔
- **Uniformly-partitioned Overlap & Save algorithm.**

Uniformly-partitioned O&S algorithm (1):



The impulse response $h(n)$ is partitioned in a reasonable number P of equally-sized blocks (i.e. P = 4), where each block is K points long.

Uniformly-partitioned O&S algorithm (2):



Each block is treated as a separate IR, zero padded to L and transformed with FFTs, in order to obtain a collection of P frequency domain filters S_i (i.e. $P = 3$).

Every filter S_i is convolved, using the Overlap&Save method, to L-point blocks of input data (each block begins $L - K$ points after the previous).

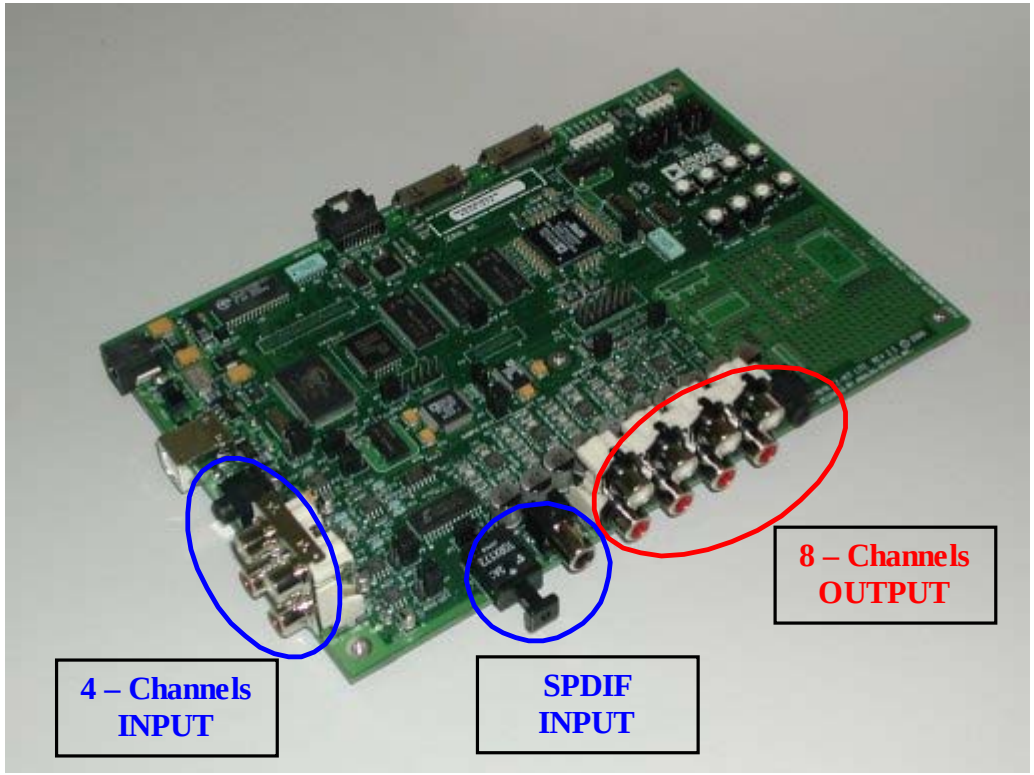
The results of the multiplications of the P filters S_i with the FFTs of the latest B input blocks are summed in frequency domain accumulators, and at the end an IFFT is done on the content of first accumulator for producing a block of output data. Only the latest L-K points of the block have to be kept.

Uniformly-partitioned O&S algorithm (3):

• ***Total number of FFTs is minimized***, in fact each block of input data needs to be FFT transformed and IFFT antitransformed just once, after frequency-domain summation.

•***Latency of the whole filtering processing is just L points instead of N***. It means that the I/O delay is kept to a low value, provided that the impulse response is partitioned in a sensible number of chunks (8 – 32).

Analog Devices DSP platform's features:



ADDS 21161N Ez-Kit Lite board

- 100 MHz (10 ns) SIMD SHARC DSP core.
- 600 MFLOPS (32-bit floating-point data).
- 600 MOPS (32-bit fixed-point data).
- Single-cycle instruction execution, including SIMD operation in two parallel computational units (ALUs).
- 4 channels INPUT, 8 channels OUTPUT.
- AD1836 and AD1852, 48 or 96 kHz sampling frequency, 24-bits audio converters

DSP's performances:

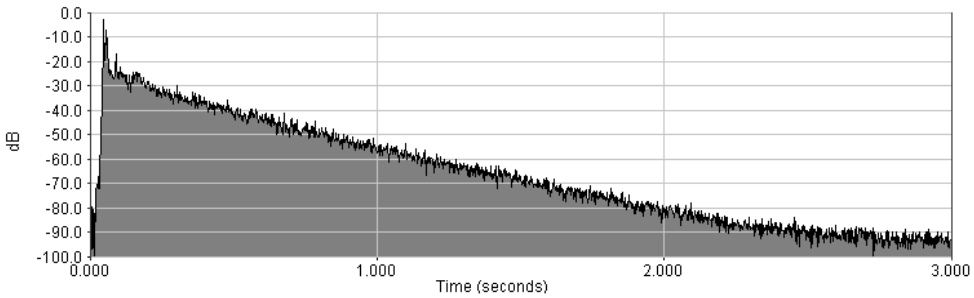
FIR filter implementation:

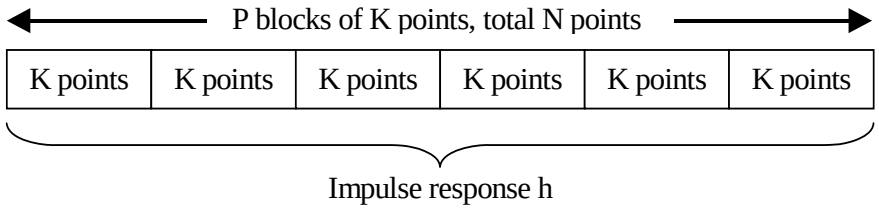
- Using a sampling frequency of 48 kHz, a 2000-taps direct-form FIR can be implemented

Maximun length of the Impulse Response is around ***40 ms*** !!

SIMD architecture of this processor (dual ALU) allows to implement a 2000-taps FIR filtering simultaneously on two independent data flows (stereo processing).

Impulse Response processing:



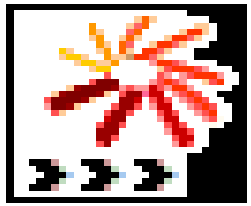


Impulse Response is:

- downloaded on DSP
- partitioned into *P* blocks, where each block is *K* points length (*K = 4096*)
- each block is zero-padded to a length of *L* points (*L = 8192*)
- transformed by standard FFT procedure supplied by Analog Devices and stored in the external memory.

I/O data stream processing:

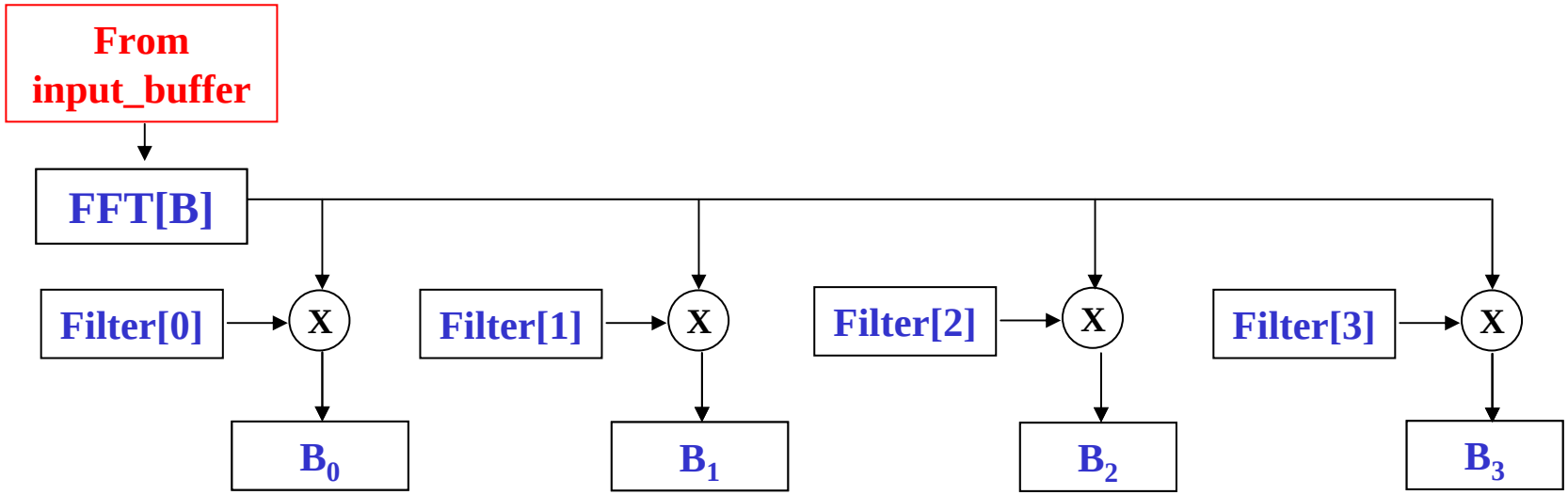
A ping-pong I/O buffer was used in the implementation of the algorithm:



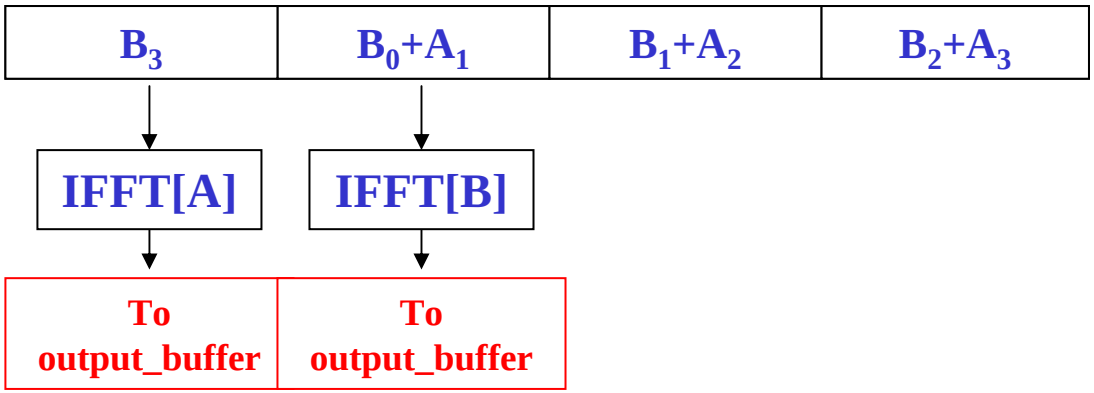
Ping-Pong I/O buffer animation

Filtering procedure:

Industrial Engineering Dept.
University of Parma – Italy



Computation circular buffer



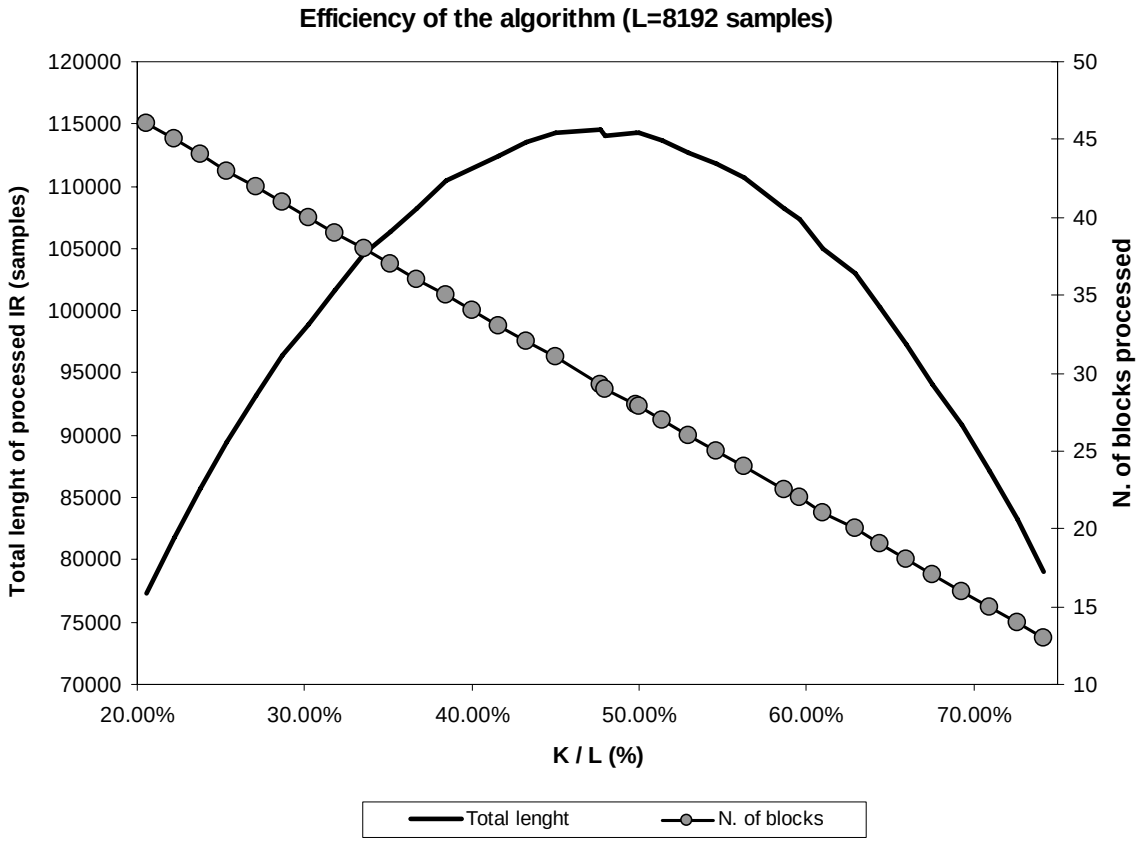
- **FFT[B]** = FFT of the processing stream.
- **Filter[i]** = *P* blocks containing FFT of the IR (i.e. *P* = 4)
- **IFFT[B]** = IFFT of the block $\mathbf{B_0 + A_1}$
- **Last *L-K*** of IFFT[B] are sent to Output_Buffer

Results (1):

Ch IN	Ch OUT	Number of block	IR length
1	1	27	**110592**
2	2	11	45056
2	4	5	20480
4	8	2	8192

- 110592 points @ 48 kHz $\Rightarrow$ IR length 2.3 second.
- 45056 points @ 48 kHz $\Rightarrow$ IR length 0.94 second.
- In 2x2 mode (4 filters) is far in excess than the requirements for good cross-talk canceling filters (typically 4096 taps).

Results (2):



- Tests performed demonstrated that the maximum efficiency is reached when the overlap between two input streams is around half of the FFT length, L.
- Using L = 8192 points, and a sampling frequency F_s = 48 kHz, latency between Input and Output is 170 ms.

Conclusion:

- Succesfull implementation of the real-time partitioned convolution on the ADDS 21161N Ez-Kit Lite board, operated from 1 to 4 input channels @ 48kHz.
- Impulse Responses of 110592 points were managed, with latency between Input and Output data limited to 170 ms.
- When it is required to implement a light, compact system and with little number of channels, DSP is a sensible solution, otherwise a PC provides a significantly better price/performance ratio.