# Multi-Parallel Convolvers

Luigi Dadda, Vincenzo Piuri, Renato Stefanelli

Department of Electronics and Information, Politecnico di Milano
Milano, Italy

## Abstract

The paper presents a new scheme for convolver design, called multi-parallel convolver: it is based on concurrent processing of **p** adjacent samples that are input simultaneously to the **p**-parallel convolver. The scheme is composed by **p** units; each of them receives the input samples and produces one convolution every **p** samples (we call them **p**-phase sub-convolvers). The detailed design of the **p**-phase sub-convolvers and of the whole **p**-parallel convolver is presented and discussed. The scheme can be used both for the bit-parallel input presentation of each sample and for the bit-serial one.

The input sample's rate of the **p**-parallel convolver is **p** times the sample's rate of a standard (**1**-parallel) convolver implemented by using the same integration technology. The number of components required by a **p**-parallel convolver is approximately **p** times the number of components required by a standard convolver.

## 1: Introduction

Convolution and spectral transforms (e.g., the Fourier transform) are basic digital signal processing (DSP) operations. They are used in a wide range of applications with different computational constraints, in particular concerning the sampling frequency.

Software convolution and transforms are suited in applications requiring low sampling rates (e.g., in acoustics), while hardware convolvers and transformers become mandatory in telecommunication (e.g., for filtering) and in radars (e.g., for SARs - synthetic aperture radars).

The recent advances in integration technologies allow to implement these operations very efficiently by using dedicated units. In particular, VLSI and WSI technologies have greatly expanded the application fields which can benefit from DSP, since they achieve both higher speed and complexity (e.g., the number of convolution terms or transform points).

In literature, serial-input and parallel-input architectures have been proposed and used [1,2, 3, 4, 5, 6] to deal with different sampling rates and sample lengths. Speed of several MSPS (Mega Samples Per Second) have been achieved by adopting parallel input schemes [6, 9].

Since even higher MSPS are increasingly necessary for advanced applications (e.g., for SARs), we need to consider and develop faster technologies (e.g., GaAs technologies) and highly-parallel architectures. As massive-computing structures, systolic arrays were widely appreciated and used [3]; in particular, their success grew with the advances in WSI technologies and in defect/fault tolerance.

The complete exploitation of high architectural parallelisms requires the definition of suitable parallel algorithms. A Fast Fourier Transform algorithm is intrinsecally highly parallel, since it operates on subsequent time windows containing quite a large number of samples (usually at least few tens). In practical applications, such a parallelism is never fully exploited: some samples serialization is always considered (e.g., 4-samples groups in pipelined radix-4 FFT schemes).

The convolution algorithm is inherently parallel. It operates on time windows: while the Fourier transform considers non-overlapped windows (overlapping is adopted for secondary reasons), two subsequent outputs of a N-samples convolution are generated from windows which are overlapped by N-1 terms.

This high parallelism allows to achieve a sampling rate which is equal to the time required to perform the complete convolution (or one step of the convolution algorithm if a pipelined scheme is adopted). On the other hand, the maximum sampling rate is constrained by the system clock rate: in particular, it is equal to the clock rate if the samples are input in the parallel form. The clock rate is then constrained by the adopted implementation technology and by the convolver architecture. The optimal architecture, as far as the

1063-6889/93 $03.00 © 1993 IEEE

sampling rate is concerned, is a pipelined structure where each convolution step is performed in one clock period. This time is the time required by the basic operation performed in each stage of the pipeline; in the fastest schemes, it is given by the computational delay of the full-adder plus the commutation time of a flip-flop.

To increase the sampling rate beyond these technological and architectural constraints, it is necessary to devise architectures which accept two or more samples in parallel, instead of the single sample.

This approach could appear greatly in contrast with the facts that the input samples are naturally in time sequence and that the convolution is a time-sequential operation performed on the input stream, sample by sample. However, at first we must remark that also the spectral transform is sequential by windows and parallel within each window (even if, within each window, it can be sequentially performed by sub-windows).

Then, we observe that convolution can be obtained by using a well-known parallel computational scheme on the input samples: at first we apply the Fourier transform to the sequence of input samples, the Fourier spectrum is multiplied by the vector of convolution weights and the result is restored in the sequential form by applying the inverse Fourier transform.

Finally, we point out that this paper deals with an algorithmic approach to process more than one input sample at a time and to produce more than one convolution at a time: the convolutions which are processed in the same group are delivered simultaneously, even if it is possible (through suitable delay circuits) to restore their correct timed sequence.

Although this approach has been developed to overcome the technological and the architectural constraints on the sampling rate in bit-parallel circuits, it can be used also to increase the low sample rate of the bit-serial structures.

In general, for a given technology, the circuits of the convolver handling **p** samples in parallel occupy an area which is approximately **p** times the area of the standard convolver: this is often reasonable if no better solutions are available to decrease the computational time.

In this paper, we do not assume any specific mode for data presentation at the convolver inputs and for result delivery at the convolver outputs. Parallel presentation is considered at samples' level (i.e., among the samples), regardless of the bit-serial or bit-parallel presentation of each sample.

This assumption implies that suitable conversions of the data presentations must be adopted to guarantee the connectivity of the convolver to the other components of the computing system in which the convolver itself is included. For example, if the bit-parallel input samples are available sequentially one after the other, they must be stored in a bank of **p** registers (**p** at a time), so that they can be processed in parallel when all of them arrived. If the bit-serial input samples arrive one after the other and we adopt a bit-serial architecture for the convolver, the samples must be stored in a bank of **p** shift registers; data loading is performed serially by considering the shift registers as a single cascaded shift register, while retrieval is executed by extracting serially every sample from each shift register synchronously.

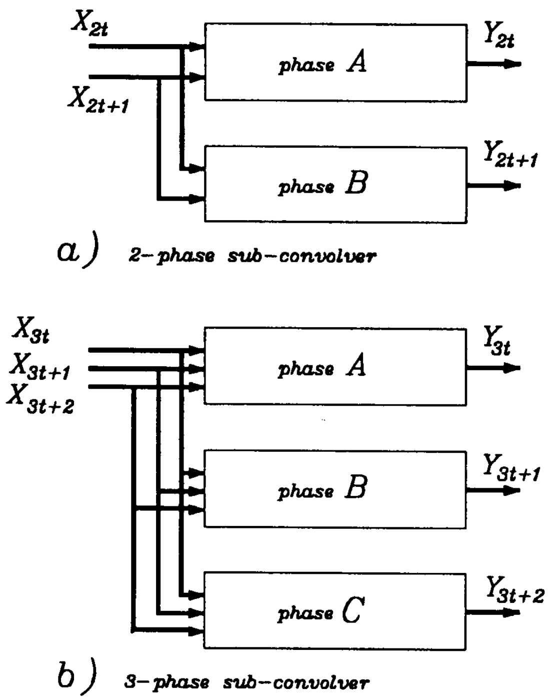
Similar conversions are required to regenerate the proper output presentation. However, since in the following we concentrate our attention on the core of the **p**-parallel convolver, we do not further consider this conversion problem.

The paper introduces the design methodology at first by considering the simple case of **2**-parallel convolvers. The architecture is derived from the analysis of the sequences of the basic convolution operations and, then, it is algebraically formalized. In section 3, the architecture and the method are extended to deal with **3**-parallel and **p**-parallel convolvers. Additional remarks concerning the architectural implementation and the fault-tolerance characteristics are discussed in section 4.

## 2: An Architecture for the 2-Parallel Convolver

A standard convolver is fed by a sequence of samples $\mathbf{X_i}$ (for $\mathbf{i}$ = 0, 1, ...) and delivers, for each $\mathbf{X_i}$, the convolution $\mathbf{Y_i} = \sum_{\mathbf{j=0}}^{\mathbf{N-1}} \mathbf{W_j X_{i-j}}$, where $\mathbf{W_j}$ are the coefficients (or weights) of the convolution. The simplest case of the **p**-parallel convolvers is the **2**-parallel one. In this convolver, a sequence of samples' pairs ($[\mathbf{X_{2t}}, \mathbf{X_{2t+1}}]$, for **t** = 0, 1, ...) is fed into the circuit, and two **N**-samples convolutions ($\mathbf{Y_{2t}}$ and $\mathbf{Y_{2t+1}}$) are generated for each pair of input samples. The convolver has therefore two input ports and two output ports.

The sequence of the $\mathbf{Y_{2t}}$ convolutions can be viewed as the output of a *2-phase* convolver [10]. A *phase* of a ***p**-phase* convolver is the circuit which computes one convolution every **p** samples. Similarly for $\mathbf{Y_{2t+1}}$. The general architecture of the **2**-parallel convolver can be obtained by using two **2**-phase convolvers (see fig. 1a): these convolvers work in parallel and are fed by the sequence of samples' pairs. In [10], design of **p**-phase convolvers with a single input samples' sequence is discussed. In this section we derive the design of a **2**-phase **2**-parallel-input convolver.



a) 2-phase sub-convolver

b) 3-phase sub-convolver

*Figure 1 - The general architecture of p-parallel convolvers: the 2-parallel convolver as a 2-phase 2-input convolver (a), the 3-parallel convolver as a 3-phase 3-input convolver (b).*

The operation of a 2-parallel convolver is described in Table A. For simplicity sake, we consider the case in which N is multiple of p, so that all the convolvers have to manage the same number of samples (i.e., have the same size); in our example we assume N=6.

Table A has been obtained as follows. The N=6 weights ($W_5$, $W_4$, $W_3$, $W_2$, $W_1$, $W_0$) are grouped in N/p=3 groups of p=2 weights: $w_1$=[$W_1$, $W_0$], $w_2$=[$W_3$, $W_2$], $w_3$=[$W_5$, $W_4$]. In the first N/p=3 rows (the time section **t** = 0), we place these groups in the first p=2 columns, which are characterized by the odd (**h mod 2** = 1) and the even (**h mod 2** = 0) values of the index **h**. For simplicity sake, only the indexes of the weights are written in the table.

Then, the first samples' pair (p-tuple) $x_0$=[$X_0$, $X_1$] is written in the next p=2 columns, characterized by even (**k mod 2** = 0) and odd (**k mod 2** = 1) values of the index **k**, respectively.

In the first row of the following 2 columns, we put the result of the vector product $w_1 \cdot x_0$= [$W_1$, $W_0$] $\cdot$ [$X_0$, $X_1$] = [$W_1X_0$, $W_0X_1$]. The columns are characterized by **h,k mod 2**, i.e., the residue modulo 2 of the index pair (**h,k**). In the adjacent 2 columns of the first row, we put the result of the vector product $w_1 \cdot x_0'$ = [$W_1$, $W_0$] $\cdot$ [$X_1$, $X_0$] = [$W_1X_1$, $W_0X_0$]. The other 2 rows of the time section **t** = 0 can be written by applying the same rules.

Similarly, we operate for the subsequent time section **t** = 1 in which the sample pair $x_1$= [$X_2$, $X_3$] is fed, and so on for all the other time sections.

| $t$ | $w_v$ | $W_h:h$ | | $X_k:k$ | | $W_hX_k$: $hk$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | $x_t$ | | $w_vx_t$ | | $w_vx_t'$ | |
| | | 1 mod2 | 0 mod2 | 0 mod2 | 1 mod2 | 1,0 mod2 | 0,1 mod2 | 1,1 mod2 | 0,0 mod2 |
| 0 | $w_1$ | 1 | 0 | 0 | 1 | 10 | 01 | 11 | 00 |
| | $w_2$ | 3 | 2 | | | 30 | 21 | 31 | 20 |
| | $w_3$ | 5 | 4 | | | 50 | 41 | 51 | 40 |
| 1 | | 1 | 0 | 2 | 3 | 12 | 03 | 13 | 02 |
| | | 3 | 2 | | | 32 | 23 | 33 | 22 |
| | | 5 | 4 | | | 52 | 43 | 53 | 42 |
| 2 | | 1 | 0 | 4 | 5 | 14 | 05 | 15 | 04 |
| | | 3 | 2 | | | 34 | 25 | 35 | 24 |
| | | 5 | 4 | | | 54 | 45 | 55 | 44 |
| 3 | | 1 | 0 | 6 | 7 | 16 | 07 | 17 | 06 |
| | | 3 | 2 | | | 36 | 27 | 37 | 26 |
| | | 5 | 4 | | | 56 | 47 | 57 | 46 |
| | | | | | | $Y_5$<br>$Y_{2t+1}$ | | $Y_6$<br>$Y_{2t}$ | |

*Table A - The operation of a 2-parallel convolver. The terms $W_h X_k$ are shown in the initial steps of a 6-samples convolution; the weights $W_h$ are given in the two leftmost columns (ordered by decreasing values of h), the samples $X_k$ are shown in the adjacent two columns (ordered by increasing values of k), and the products $W_h X_k$ are listed in the four rightmost columns (identified by the index pair hk).*

In the first time section **t** = 0, the index pairs are written by using different character heights. Small characters are adopted for those pairs whose index sum is smaller that 5 since they represent products which do not belong to any convolution. The first convolution is in fact $Y_5 = \sum_{k=0}^{5} W_k X_{5-k}$, since N=6. The product pairs belonging to $Y_5$ (i.e., those having sum equal to 5) lay in the columns **h,k mod 2 = 1,0** and **h,k mod 2 = 1,0**, and are circled. Similarly, the products belonging to the convolution $Y_6$ are circled in the two rightmost columns.

Consider the "mask" defined by the circles and the dashed links in the four rightmost columns of Table A: it points out the pairs of terms for $Y_5$ and $Y_6$, respectively. Note that the corresponding terms for the convolutions $Y_7$ and $Y_8$ can be identified by shifting such mask down by one time section, and so on for all the other convolution's pairs.

The convolver architecture can be directly derived from table A: different solutions may be considered. In a first solution, we observe that, at each time step, two products for each convolution are generated on $Y_{2t+1}$ (requiring N/2 time steps for each convolution): it is possible and it could be advantageous to design two "merged" multipliers which produce the sum of the products' pairs [1]. A different approach is required for $Y_{2t}$: the terms of the first product pairs (namely, 45 and 51) are produced in different time steps, and the same case occurs for all the subsequent pairs. Therefore, 4 time steps are required to generate $Y_{2t}$.

In a second solution, we can add separately the convolution terms on each column and, then, the two resulting sums may be added to generate the convolution. This implies that, in the column **h,k mod 2 = 1,0**, we convolve the even-indexed samples $X_{2t}$ with the odd-indexed weights $W_5$, $W_3$ and $W_1$; while, in the column **h,k mod 2 = 0,1**, the odd-indexed samples $X_{2t+1}$ are convolved with the even-indexed weights $W_4$, $W_2$ and $W_0$. The corresponding units are called sub-convolvers. Their outputs are then added to generate the $Y_{2t+1}$ convolutions (see the circuit on the top of fig.2).

For $Y_{2t}$, we note that the terms in the column **h,k mod 2 = 1,1** of Table A can be paired with the corresponding terms in the column **h,k mod 2 = 0,0** by delaying them by one time step (see the delay unit at the input of the third sub-convolver of fig. 2). The circuit producing $Y_{2t}$ is shown on the bottom of fig.2. With this circuit (see Table A), $Y_6$ is output in the same time step (**t** = 3) of $Y_7$; the proper synchronization of $Y_6$ with $Y_5$ can be obtained by means of a delay at the output of $Y_{2t+1}$ (see fig.2).

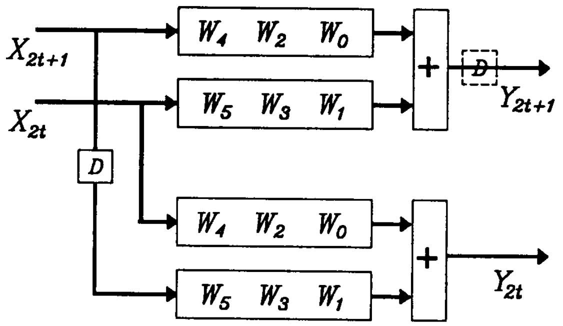


***Figure 2 - The 2-parallel convolver composed by 1-input sub-convolvers.***

The above analysis can be formalized by restructuring the mathematical definition of the convolution operation. First of all, we consider the odd convolution results (i.e., the outputs at the odd time steps in the standard architecture); they are identified by $Y_{2t+1}$ (for **t** = 0, 1, 2, ...). They can be rewritten as:

$$Y_{2t+1} = \sum_{j=0}^{N-1} W_j X_{2t+1-j}$$

We can separate the products containing the odd indexed X's from those containing the even indexed X's.

$$Y_{2t+1} = \sum_{s=0}^{N/2-1} W_{2s} X_{2t+1-2s} +$$
$$+\sum_{s=0}^{N/2-1} W_{2s+1} X_{2t+1-(2s+1)} = Y^{o}_{2t+1} + Y^{e}_{2t+1}$$

the sub-convolution $Y^{o}_{2t+1} = \sum_{s=0}^{N/2-1} W_{2s} X_{2t+1-2s}$, we obtain a convolution on N/2 samples by variable substitutions and by renumbering the subsequence of odd-indexed samples and the related weights:

$$\gamma^{oo}_{m} = \sum_{r=0}^{N/2-1} \omega^{e}_{r} \chi^{o}_{m-r}$$

where $Y_{2t+1}$ is replaced by $\gamma^{oo}_{m}$, **2s** by **r**, $W_{2s}$ by $\omega^{e}_{r}$, and $X_{2t+1-2s}$ by $\chi^{o}_{m-r}$.

From renumbering samples and weights and from the previous assumptions, we derive the relationship between

**m** and **t**. The first sample $\chi_0^o$ in the re-written subsequence $\chi_m^o$ is identified by **m** = 0; the second one $\chi_1^o$ by **m** = 1; and, in general, the *(m+1)*-th sample $\chi_m^o$ is identified by **m**. The first odd-indexed sample $X_1$ is identified in the input sequence $X_i$ by **t** = 0; the second one $X_3$ by **t** = 1; and, in general, the *(m+1)*-th odd-indexed sample $X_{2t+1}$ is identified by **t** = *m*, since, by our assumptions, the sample $X_{2t+1}$ corresponds to the sample $\chi_m^o$. Therefore, it is **m** = **t**.

For $Y_{2t+1}^e = \sum_{s=0}^{N/2-1} W_{2s+1} X_{2t+1-(2s+1)}$, we obtain similarly a convolution on N/2 samples:

$$\gamma_m^{oe} = \sum_{r=0}^{N/2-1} \omega_r^o \chi_{m-r}^e$$

where $Y_{2t+1}^e$ is replaced by $\gamma_m^{oe}$, **2s+1** by **r**, $W_{2s+1}$ by $\omega_r^o$, and $X_{2t+1-(2s+1)} = X_{2t-2s}$ by $\chi_{m-r}^e$. Also in this case, we can show that it is **m** = **t**.

Since $\chi_{m-r}^o$ and $\chi_{m-r}^e$ are available at the same time, the computation of the two sub-convolutions proceeds synchronously. Therefore, the original convolution (having odd index) can by computed by adding directly these sub-convolutions.

Consider now the convolution outputs $Y_{2t}$ (for **t** = 0, 1, 2, ...) at the even time steps in the standard architecture. They can be rewritten as:

$$Y_{2t} = \sum_{j=0}^{N-1} W_j X_{2t-j}$$

We separate the even-indexed samples from the odd-indexed ones:

$$Y_{2t} = \sum_{s=0}^{N/2-1} W_{2s} X_{2t-2s} + \sum_{s=0}^{N/2-1} W_{2s+1} X_{2t-(2s+1)} = Y_{2t}^e + Y_{2t}^o$$

From the expression of the sub-convolution $Y_{2t}^o = \sum_{s=0}^{N/2-1} W_{2s+1} X_{2t-(2s+1)}$, we obtain by variable substitution and by renumbering the subsequence of the odd-indexed samples and the related weights:

$$\gamma_m^{eo} = \sum_{r=0}^{N/2-1} \omega_r^o \chi_{m-r}^o$$

where $Y_{2t}^o$ is replaced by $\gamma_m^{eo}$, **2s+1** by **r**, $W_{2s+1}$ by $\omega_r^o$, and $X_{2t-(2s+1)}$ by $\chi_{m-r}^o$; from the previous assumptions and from renumbering samples and weights, it is **m** = **t-1**. The previous relationship is a convolution on N/2 samples (the odd-indexed samples of the original one).

From the expression of the sub-convolution $Y_{2t}^e = \sum_{j=0}^{N/2-1} W_{2j} X_{2t-2j}$, we obtain by variable substitutions and by renumbering:

$$\gamma_m^{ee} = \sum_{r=0}^{N/2-1} \omega_r^e \chi_{m-r}^e$$

where $Y_{2t}^e$ is replaced by $\gamma_m^{ee}$, **2s** by **r**, $W_{2s}$ by $\omega_r^e$, and $X_{2t-2j}$ by $\chi_{m-r}^e$; from renumbering and from the previous assumptions, it is **m** = **t**. Again, the above relationship defines a standard convolution on N/2 samples (the even-indexed samples of the original convolution).

Since **m** is equal to **t-1** in the first sub-convolution and by t in the second one, $\gamma_m^{eo}$ is generated when the (t-1)-th pair of samples has been input, while $\gamma_m^{ee}$ is produced only after the **t**-th pair arrived. This implies that $\gamma_m^{ee}$ is available one time step after $\gamma_m^{eo}$. To generate the original convolution (having even index), we must properly synchronize the sub-convolutions and, then, add them. Synchronization can be achieved either by introducing a delay at the output of $\gamma_m^{eo}$ before addition or by delaying the inputs of $\gamma_m^{eo}$ (see in fig. 2).

To evaluate the realizability and the effectiveness of a 2-parallel convolver, it is necessary to compare the *circuit complexity* of this convolver to the complexity of a standard convolver (in which the samples are presented one at a time).

Consider the architecture given in fig.2: in the case of N even, the number of basic convolver cells is twice the number cells for a standard scheme, since four sub-convolvers are required and each of them contains N/2 cells. We must consider also the two final adders and the two delay circuits.

The *latency time* for each convolution is about one half of latency time of the standard architecture. The small increase with respect to one half of the standard latency is due the computational time of the adder at the output of the sub-convolvers and to the delay circuits.

The *throughput* of the architecture is doubled with respect to the standard one.

## 3: The Convolvers for Three or More Parallel Samples

The design methodology for p-convolvers with p>2 is similar to the one discussed in the previous section for the case of p=2. The case of p=3 is considered in detail, then the results will be extended to the general case.

Table B lists all the relevant terms in the first convolution steps with N=9 samples (N is assumed to be multiple of p=3). Table B is obtained as for Table A, with the following additional remarks. The N=9 weights $W_8$, ..., $W_0$ are partitioned in N/p groups containing p=3 weights each: $w_1$=[$W_2$, $W_1$, $W_0$]; $w_2$=[$W_5$, $W_4$, $W_3$]; $w_3$=[$W_8$, $W_7$, $W_6$]. Samples are presented in subsequent triplets $x_0$=[$X_0$, $X_1$, $X_2$], $x_1$=[$X_3$, $X_4$, $X_5$], $x_2$=[$X_6$, $X_7$, $X_8$], ..., at the time steps t = 0, 1, 2, and so on. We place these groups and triplets in the leftmost 2 columns (W and X parts) of the first p adjacent rows (section t = 0).

| $t$ | $w_v$ | $W_h$: $h$ | | | $X_k$: $k$ | | | $W_hX_k$: | | | | $hk$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $x_t$ | | | $w_vx_t$ | | | $w_vx_t'$ | | | $w_vx_t''$ | |
| | | 2 mod3 | 1 mod3 | 0 mod3 | 0 mod3 | 1 mod3 | 2 mod3 | 2,0 mod3 | 1,1 mod3 | 0,2 mod3 | 2,1 mod3 | 1,2 mod3 | 0,0 mod3 | 2,2 mod3 | 1,0 mod3 | 0,1 mod3 |
| 0 | $w_1$ | 2 | 1 | 0 | 0 | 1 | 2 | 20 | 11 | 02 | 21 | 12 | 00 | 22 | 10 | 01 |
| | $w_2$ | 5 | 4 | 3 | | | | 50 | 41 | 42 | 51 | 42 | 30 | 52 | 40 | 31 |
| | $w_3$ | 8 | 7 | 6 | | | | 80 | 71 | 62 | 81 | 72 | 60 | 82 | 70 | 61 |
| 1 | | 2 | 1 | 0 | 3 | 4 | 5 | 23 | 14 | 05 | 24 | 15 | 03 | 25 | 13 | 04 |
| | | 5 | 4 | 3 | | | | 53 | 44 | 35 | 54 | 45 | 33 | 55 | 43 | 34 |
| | | 8 | 7 | 6 | | | | 83 | 74 | 65 | 84 | 75 | 63 | 85 | 73 | 64 |
| 2 | | 2 | 1 | 0 | 6 | 7 | 8 | 26 | 17 | 08 | 27 | 18 | 06 | 28 | 16 | 07 |
| | | 5 | 4 | 3 | | | | 56 | 47 | 38 | 57 | 48 | 36 | 58 | 46 | 37 |
| | | 8 | 7 | 6 | | | | 86 | 77 | 68 | 87 | 78 | 66 | 88 | 76 | 67 |
| 3 | | 2 | 1 | 0 | 9 | 10 | 11 | 29 | $1_{10}$ | $0_{11}$ | $2_{10}$ | $1_{11}$ | 09 | $2_{11}$ | 19 | $0_{10}$ |
| | | 5 | 4 | 3 | | | | 59 | $4_{10}$ | $3_{11}$ | $5_{10}$ | $4_{11}$ | 39 | $5_{11}$ | 49 | $3_{10}$ |
| | | 8 | 7 | 6 | | | | 89 | $7_{10}$ | $6_{11}$ | $8_{10}$ | $7_{11}$ | 69 | $8_{11}$ | 79 | $6_{10}$ |
| | | | | | | | | | $Y_8$ | | | $Y_9$ | | | $Y_{10}$ | |
| | | | | | | | | | $Y_{3t+2}$ | | | $Y_{3t}$ | | | $Y_{3t+1}$ | |

*Table B - The operation of a 3-parallel convolver. The terms $W_hX_k$ are shown in the initial steps of a 3-parallel convolver with N=9.*

In the 9 columns of the WX part, we place the index pairs hk of the products $W_hX_k$ according to the following rule. In the first p columns of the first row in the part WX, we place the components of the vector product $w_1 \cdot x_0$ (20, 11, 02). We perform then a circular permutation on $x_0$, obtaining $x_0'$=[$X_1$, $X_2$, $X_0$]; we multiply this new vector by $w_1$ and we place the result in the adjacent three columns of the first row in the part WX. We operate again a circular permutation on $x_0'$, obtaining $x_0''$=[$X_2$, $X_0$, $X_1$]; we multiply it by $w_1$ and we place the results in the rightmost three columns of the first row in the WX part. We repeat these operations also for the second and for the third rows of section t = 0.

Section t = 1 of Table B is obtained from the section t = 0 by replacing the 3-tuple ($X_0$, $X_1$, $X_2$) with the subsequent input 3-tuple ($X_3$, $X_4$, $X_5$) and by applying the same algorithm for generating the various products. The same algorithm is applied to the subsequent time sections.

Since a convolution $Y_i$ is composed by N products characterized by a sum of index factors equal to i, we can find the terms composing such convolution in Table B by selecting all the products $W_hX_k$ for which h+k=i. The first convolution for the case N=9 is $Y_8$: its terms have been circled in Table B. Note that all of them belong to the comlumns $x_t$ which generate the convolutions $Y_8$, $Y_{11}$, $Y_{14}$, ..., i.e., the convolution sequence $Y_{3t+2}$ (for t = 0, 1, 2, ...).

By using the terms belonging to the columns $x_t'$ we can compute the convolutions $Y_9$, $Y_{12}$, $Y_{15}$, ..., i.e., the convolution sequence $Y_{3t}$ (for t = 0, 1, 2, ...). Similarly, the columns $x_t''$ generate the convolution sequence $Y_{3t+1}$ (for t = 0, 1, 2, ...).

Each convolution sequence contains one convolution of the standard sequence every three. The units generating such sequences have been called *phases* [10] of a *three-phase convolver* (see fig. 1b. Design of a 3-phase, 3-parallel-input convolver can be performed on Table B by extending the method used in the previous section for the 2-phase, 2-parallel-input convolver.

One phase of the 3-phase, 3-parallel-input convolver can be composed by three standard (1-input) sub-convolvers (fig. 3), characterized by the weights given in columns h mod 3 = 2, h mod 3 = 1, and h mod 3 = 0 of Table B, respectively. The sub-convolvers' outputs are added together to generate $Y_{3t}$, $Y_{3t+1}$ and $Y_{3t+2}$.

As $Y_{3t+2}$ is concerned, for each triplet $X_{3t}$, $X_{3t+1}$, $X_{3t+2}$, the corresponding products belong to the same row. The three sub-convolvers are fed directly by the input triplets. For $Y_{3t}$, $X_{3t+1}$ and $X_{3t+2}$ (i.e., the inputs of sub-convolutions [$W_7$, $W_4$, $W_1$] and [$W_8$, $W_5$, $W_2$], respectively) must be delayed by one time step to guarantee the proper synchronization of input samples. For $Y_{3t+1}$, $X_{3t+2}$ must be delayed by one time step.

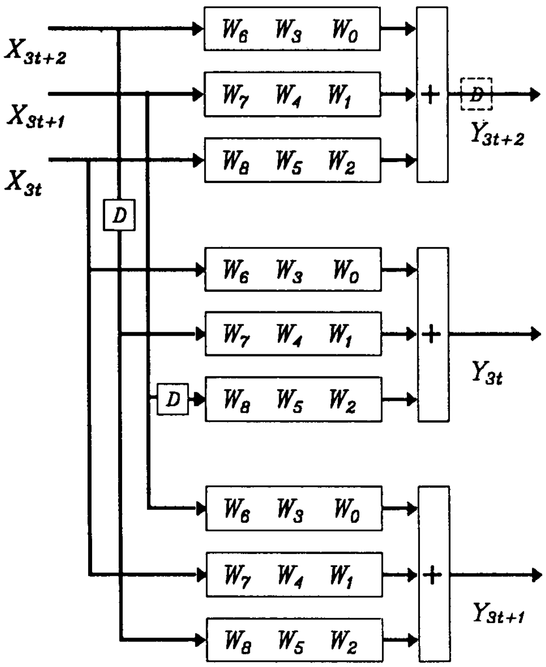


***Figure 3 - The 3-parallel convolver composed with 1-input sub-convolvers.***

The procedure and the formal analysis can be generalized for an arbitrary value of p. We note here that the $p^2$ columns in the part **WX** are partitioned into p groups $w_0$, $w_1$, ..., $w_{p-1}$, of p columns each. The corresponding products are found by vector multiplication of $w_i$ and $x_i$, which can be determined in the same way discussed for p=3. The input variables, that must be delayed to guarantee a proper synchronization of the sub-convolution outputs, are those that occur on the right of $X_0$ in each permutation $x_i$.

The formal analysis can be performed by partitioning the definition of the convolution into p sub-convolutions: each sub-convolutions is characterized by a unique value of the residue modulo p of the index for the considered samples.

The evaluation of our design methodology may be performed by considering the general case of p-parallel convolvers. The area $A_s$ occupied by the circuits of the standard architecture (i.e., the 1-parallel one) is basically given by $A_s = NA_c$, where $A_c$ is the area of each cell of the convolver. The *circuit complexity* (i.e., the area) of a p-parallel convolver is computed as $A_p = pNA_c + pA_a + pA_d$, where $A_a$ is the area of each adder at the output of the sub-convolvers, and $A_d$ is the area of each delay circuit. Since $A_a$ and $A_d$ are usually small with respect to the area of the convolver's cells, the area of the p-parallel convolver increases proportionally to p with respect to the area of the original convolver, i.e., $A_p \approx pA_s$.

The *throughput* of the p-parallel convolver is p times the throughput of the standard convolver.

Let $L_s$ be the *latency time* of the standard convolver, the latency time of the p-parallel convolver (for N multiple of p) is given by $L_p = L_s / p + t_a + t_d$, where $t_a$ is the delay of each adder at the output of the sub-convolvers, and $t_d$ is the delay introduced by the delay circuit (i.e., by definition, it is $t_d = L_s / N$). Also in this case, $t_a$ and $t_d$ are usually small with respect to $L_s / p$: the latency therefore is approximately proportional to 1/p with respect to the latency of the standard convolver, i.e., $L_p \approx L_s / p$.

## 4: Additional Design Considerations

In the previous sections, we assumed that N is multiple of p. If this is not true, the design methodology must be modified as follows.

The number of weights' group is $\lceil N / p \rceil$; for N=8 and p=3, it is $\lceil N / p \rceil$ =3, i.e., there are three groups $w_v$ as in the preceding example. The weights are partitioned in groups of p elements starting from $W_0$. The last group (containing $W_{N-1}$) contains less then p weights. The empty places in the last group are filled with zeros. The procedure continues as in the previous sections.

The circuit can be obtained by observing that, in each phase-convolver, the last sub-convolver contains only 2 stages, $W_7$ and $W_6$, with an additional delay at the sample input (or a third stage filled with zero weights).

In our design methodology, we can deal also with *fault tolerance*. Two levels of this problem can be considered. Fault tolerance may be taken into account at the sub-convolver level, i.e., within each sub-convolver. To such purpose, a number of approaches have been presented in literature [7, 8], according to the specific sub-convolver architecture and to data presentation.

A second architectural level of fault tolerance can be envisioned in our approach, as it is shown in fig. 4 for a 2-parallel convolver. One spare sub-convolver is introduced in the basic p-parallel architecture to deal with one faulty sub-convolver; a set of switches allows to confine the faulty sub-convolver and to replace it by using the spare one (see fig. 4b). In general, weight redistribution is required to guarantee the proper execution of the convolution operations.

In this architecture, we have some circuits which constitute the hard core for the fault tolerance of the system: namely, the switches, the output adders, and the delay circuits. However, we must point out that this hard core is usually small with respect to the whole convolver.

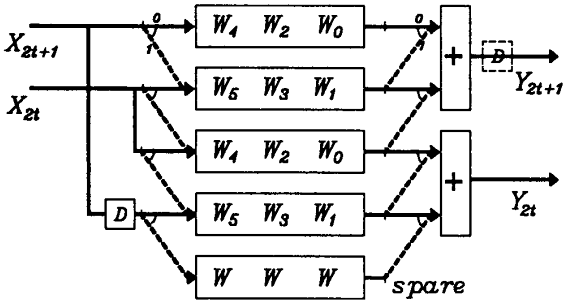


***Figure 4 - A fault-tolerant scheme for the 2-parallel convolver.***

A generalization of the scheme presented in fig. 4 can be envisioned to deal with more than one faulty sub-convolver. To such purpose, a switched bus architecture [12] can be adopted both for input distribution and for output collection: **p** input buses and **p** output buses are introduced. One position of the input switches allows to deliver the proper input data to the corresponding sub-convolver (i.e., to connect the proper input bus to the sub-convolver). One position of the output switches for each sub-convolver allows to collect the proper sub-convolution for the final addition. Reconfiguration can be performed by completely excluding one or more sub-convolvers from the output connections.

These techniques can be adopted also to introduce *defect-tolerance* capabilities, e.g., through laser cutting.

## 5: Conclusions

A new methodology for convolver design has been presented to increase the effective sampling rate beyond the technological and the architectural constraints. Our approach builds an enhanced convolver by using the available convolver schemes and the current technologies: the innovative idea is the partitioning of the computation onto a number of cooperating convolvers. Groups of **p** (>1) samples are handled in parallel by **p** sub-convolvers. The general procedure has been presented to design the **p**-parallel convolvers, for given values of the sample-parallelism degree **p** and the number **N** of convolution terms.

By considering the same implementation technology, the number of basic components for a **p**-convolver is slightly greater than **p** times the number of components for a standard convolver. Conversely, the throughput is greatly improved since it becomes **p** times the throughput of the standard convolver.

The computational time required to generate one convolution (latency) is the ratio between the computational time of the standard convolution and the parallelism degree **p**.

## References

[1] Swartzlander E.E. "Merged arithmetic for signal processing", IEEE Proc. Symp. Computer Arithmetic, Santa Monica, CA, 1978, pp. 239-244.

[2] Swartzlander E.E., Gilbert B.K., Reed I.S., "Inner product computers", IEEE Trans. on Computer, vol. C-27, 1978, pp. 21-31.

[3] Kung H.T., "Why systolic architectures?", IEEE Computer, vol. 15, 1982, pp.37-46.

[4] Danielsson P.E., "Serial-parallel convolvers", IEEE Trans. on Computers, vol. C-33, 1984, pp. 652-667.

[5] McCanny J.V., Phil D., McWhirter J.G., Wood K., "Optimized bit-level systolic array for convolution", Proc. IEE, vol. 131, 1984, pp. 632-637.

[6] Stearns C.C., Luthi D.A., Ruetz P.A., Ang P.H., "Design of a 20MHz 64 Tap transversal filter", IEEE Proc. ICCD'88, New York, pp. 574-577.

[7] Lasserre S., "A single-wafer bit-sliced convolver: an optimum statistical solution for the implementation of redundancy", Systolic Array Processors, Prentice Hall, 1989, pp. 514-524.

[8] Dadda L., Breveglieri L., "A modular bit-serial convolver", Proc. IFIP Workshop on Wafer Scale Integration, Como, 1990, Elsevier, pp. 279-289.

[9] Balboni A., Breveglieri L. Dadda L. Sciuto D., "A comparative evaluation of bit-serial convolvers", Proc. IFIP Workshop on Parallel Architectures on Silicon, Grenoble, 1989, pp. 309-326.

[10] Dadda L., "A Polyphase Architecture for serial-input convolvers", Journal of VLSI Signal Processing, vol. 2, 1990, pp.17-27.

[11] Dadda L., "Byte-serial Convolvers", Proc. Int'l Conf. Application Specific Array Processors, ASAP'90, Princeton, Sept. 1990, pp.530-541.

[12] Negrini R., Sami M., Stefanelli R., Fault tolerance through reconfiguration in VLSI and WSI arrays, MIT Press, Cambridge, MA, 1988