POLYPHASE CONVOLVERS

Luigi Dadda

Department of Electronics - Politecnico di Milano
P.za Leonardo da Vinci 32 - 20133 Milano - ITALY

ABSTRACT

A general theory of serial-input serial-output polyphase convolvers based on modules (*phases*) producing one convolved output every p samples, is presented. Methods are exposed for designing such convolvers for arbitrarily assigned weights and samples lengths and number of convolution terms, under the assumption of zero, or assigned, interval between successive samples, minimum number of phases and minimum intervals between successive convolved output from each phase. Two types of solution are shown, the first based on the use of distinct serial- parallel multipliers, the second on multipliers partially shared among successive convolution terms. A structure based on bit-slices is presented, permitting a convolver with assigned parameters to be designed from a stack of slices.

1. INTRODUCTION

Convolution is one of the most important operation in digital signal processing, and a number of circuits have been proposed for it. It is beyond the scope of this paper to present a detailed survey, but it seems appropriate to underline the most notable development steps.

The need for fast operation (particularly, but not exclusively, in communication) has led to an extensive use of parallel architectures e.g. by Swartzlander, [4].

Specialized modular schemes have been proposed by Kung, [5], based on systolic architectures.

Fast serial-input serial-output multipliers were proposed as interesting alternatives to parallel schemes (see a survey by Dadda and Ferrari in [1]). Serial-input serial-output convolvers have received in recent years much attention, being particularly interesting for VLSI implementation, due to their low pin count and modularity. A survey, together with a new general systolic scheme, has been proposed by Danielsson [7], and a notable scheme systolic at bit level by McCanny et al.[8,9,11]. Circuits for implementing inner product have been proposed by Sips [6] for serial-input serial-output units to be used in parallel processors.

In this paper we present an extension of a scheme proposed in [13], based on the concept of polyphase architecture, in which a convolver is composed by p sub-convolvers, or *phases*, i.e. circuits producing one convolved output every p samples. The scheme is semi-systolic, requiring the broadcasting of samples bits.

The design method suggested in the above mentioned paper was essentially to examine what can be done with one, two, three, etc. phases, obtaining interesting schemes particularly for samples and weights of equal length, but not offering a general method of designing schemes having important properties, including: 1) arbitrarily assigned samples and weights length, 2) zero samples separation (i.e. the most significant bit of a sample being followed immediately by the least significant one of the following sample) thus offering the maximum possible sampling rate compatible with the bit rate determined by the adopted technology, 3) minimum logically possible number of phases and 4) minimum logically possible interval between the convolved output from each phase. These conditions are assumed as necessary for an optimum design of convolvers belonging to the polyphase family. Moreover, a bit-slice structure will be obtained, thus permitting an easy design and the adoption of fault tolerant schemes. In the proposed schemes the attainable bit rate is determined by the delays trough a flip flop and a full adder.

In the following it will be first shown that a canonical scheme can be drawn, obeying the first of the above conditions. It will then be shown how the canonical scheme can be reduced so that the remaining conditions are also fulfilled.

2. A CANONICAL SCHEME

Convolution is obtained by multiplying N regularly time-spaced samples $X_k, X_{k-1}, \cdots\cdots, X_{k-N+1}$ with the coefficients or weights $W_{N-1}, W_{N-2}, \cdots, W_0$:

$$Y_k = \sum_{i=0}^{N-1} W_i X_{i+k-N+1}$$

It will be assumed for the moment that both weights an samples are non-signed integers of n_w and n_x bits respectively. The case of two's complement numbers will be treated later.

Weights are assumed in parallel form, stored in static registers, so that each term in the convolution can be implemented by means of a serial (the sample)-parallel (the weight) multiplier, as shown in fig.1a. Note that n_x clock times after the first sample bits, the n_x bits long, least significant part of the product, will be stored in the shift register at the left of the multiplier output, while the most significant part will be stored in redundant form in the two, n_w bits long registers included in the multiplier. After n_w more clock-times this most significant part will also be in the shift registers along with the previous least significant part, for a total length of $n_w + n_x$ bits. Such multiplier-shift units will be represented in the following in a more compact way as shown in fig.1b or 1c.

Consider now an array of N such multiplier-shift units, each one displaced by n_x stages to the right of the preceding one, as

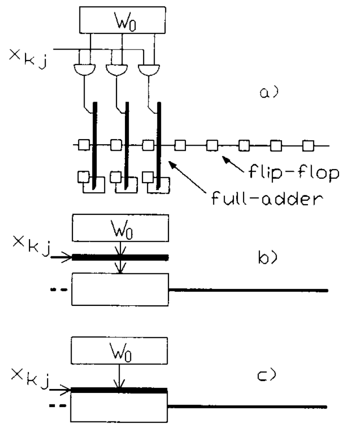


fig.1: a): The logical scheme of a serial parallel multiplier. b,c): A compact representation of it.

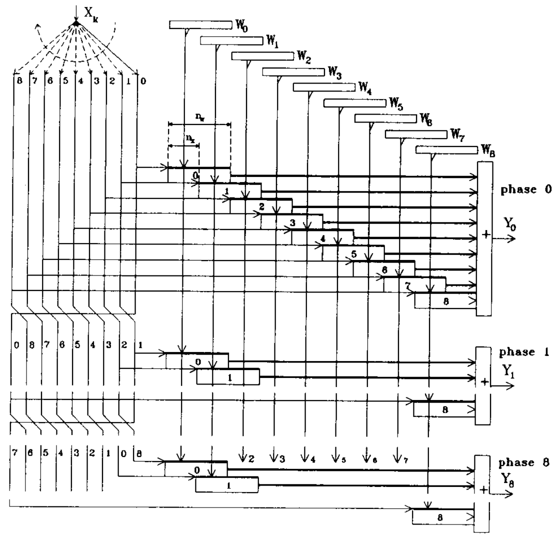


fig.2: A canonical polyphase convolver ($n_w = 4$; $n_x = 2$; $N = 9$; $n_{px} = 0$).

shown in fig.2 (upper part, phase 0) where the condition $n_w > n_x$ has been assumed, and suppose that the lengths of the shift registers is such that their rightmost stages are vertically aligned with the rightmost side of the bottom multiplier. The total length of the first, topmost unit is: $L_1 = (N-1)n_x + n_w$

The outputs of all shift registers (and of the bottom multiplier) feed a serial-input (N inputs) adder [2,3]. Assume also that the first sample X_0 is applied to unit 0, where the product W_0X_0 is generated (note that, in the following, X's indexes are: $k$mod$N$); the second sample is then applied to unit 1, generating W_1X_1 that is aligned with W_0X_0 in unit 0; and so on for the remaining units. A sample distributor is necessary to obtain the proper sample at the input of each unit, see fig.2. At the beginning of sample X_{N-1}, all products appear with their least significant bits at the input of the N-input adder, and, after the corresponding delay, the first, least significant bit of the first convolved, Y_{N-1}, is produced, followed by its successive bits.

If now the next sample, X_N, is applied to unit 0, and the following $X_{N+1}, \cdots, X_{2N-1}$ to the remaining units, the convolved Y_{2N-1} is generated. The circuit thus generates one convolved every N samples. Precisely, it produces all Y_k with $k$mod$N = 0$.

In order to obtain the convolved Y_k with $k$mod$N = 1$ an identical circuit can be used, whose inputs are: X_1 for unit 0, X_2 for $1, \cdots$, etc. For obtaining all convolved, N such circuits are necessary, each with N inputs, where the samples are cyclically permuted as shown in fig.2. In other words, the N circuits are fed at any given time, with the same sample, differently ***phased***. Each of the circuits described will be correspondingly called a phase, composed by N multiplier-shift units that will be called sub-phases. Phase i will produce all Y_k with $k$mod$N = i$.

Such convolver certainly implements the first condition, i.e., of working with zero sample's interval, but it appears rather expensive, requiring: N^2 multipliers (each n_w stages long), $1/2N(N-1)$ flip flops in the shift registers and a N inputs serial adder. It has, nevertheless, a very simple structure: we consider it as a ***canonical*** scheme that can be the starting point of a procedure leading to more economical circuits.

Before studying such procedure, let us consider some general properties of a generic serial input - serial output convolver having p output ports ($p \geq 1$) (and obviously one input port).

Having already called n_x and n_w the samples and the weights lengths respectively, we call n_{px} the samples separation, i.e. the number of clock times between the most significant bit of a sample and the least significant bit of the following sample; n_{py} is similarly the separation between successive convolved output from a given port. Note also that n_{px} and n_{py} bits can be given arbitrary values (e.g. zero, or sign-exstension, or whatever else is convenient).

The sampling rate is: $f_s = n_{tx}^{-1}$; $n_{tx} = n_x + n_{px}$ being the sampling period.

The length of convolved Y_k is: $n_y = n_x + n_w + n'$ where $n' = \lceil \log_2 N \rceil$. The convolved period is $n_{ty} = n_y + n_{py}$.

In a p output ports convolver, in a sampling period n_{tx} a total of pn_{tx} bits is output from the p ports, and, since one convolved is produced for each sample, it must be: $n_{ty} = pn_{tx}$.

A convolver with a single output is therefore conceivable only if $n_{tx} = n_{ty}$, i.e. if $n_{px} = n_{py} + n_w + n'$. For smaller n_{px} it is necessary to provide two or more output ports.

For an assigned n_{px} the minimum number of ports, p_m, can be determined by noting that the number of bits output from all ports for each sampling period, which is n_{tx} bits long, must be the minimum multiple of n_{tx} capable of accommodating one convolved n_y bits long. In other words, the minimum number of ports is: $p_m = \lceil n_y/n_{tx} \rceil$, i.e. the smallest multiple of n_{tx} not smaller that n_y.

Consequently, for n_y multiple of n_{tx}, $p_m = n_y/n_{tx}$, and the convolved separation n_{py} in each output will be zero.

For n_y not multiple of n_{tx}, it is necessary to assume a non zero convolved separation. Its minimum value, n_{pym}, is given by the smallest number to be added to n_y so that $n_y + n_{pym}$ is multiple of n_{tx}. This can be used as a procedure for evaluating $\lceil n_y/n_{tx} \rceil$:

$$\lceil n_y/n_{tx} \rceil \longrightarrow (n_y + n_{pym})/n_{tx} = p_m \qquad (A)$$

which obtains both p_m and n_{pym}.

Note therefore that p_m implies n_{pym} and viceversa.

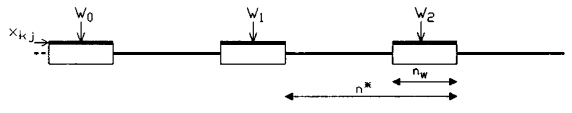


fig.3: A sub-phase scheme with regularly spaced multipliers.

Such results are independent from the architecture adopted, since they derive from purely logical global constraints. In case of polyphase architecture, p_m is also the minimum number of phases.

Note that the canonical scheme is composed by $p = N$ phases, in general larger than p_m.

The output ports can be used in ways different from the one described for the polyphase architecture (i.e. each convolved being entirely output from a given port).

As shown in [12], one can partition each convolved in portions whose length is equal to n_x: the least significant portions of the subsequent convolved will be output from a port; the following more significant portions from another port, and so on.

In a third way, the convolved could be expressed with a sequence of bytes, each composed by p bits.

In relation to the circuits fed by the convolver outputs, the convolved Y_k can be transformed accordingly, if necessary: it is easy to design a circuit for transforming any of the outputs formats into another, even different from those just described (e.g. into convolved in full parallel from, rounded to a prescribed length).

3. REDUCING THE CANONICAL SCHEME

It has been noticed that in the canonical scheme one convolved output every N samples is produced in each of the N phases, and that consequently successive convolved outputs from the same phase are separated by an interval $n_{py} = Nn_x - n_y$ (having assumed $n_{px} = 0$) usually larger than the minimum found in the preceding paragraph.

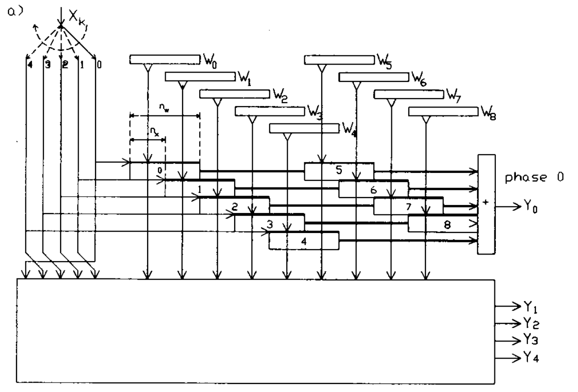
A first step in the reduction process consists then in designing a circuit affording p_m and n_{pym}.

Note that in the canonical scheme the number of phases N is equal to the number of sub-phases and to the length of the cycle in the X-inputs distributor. We first look at reducing such numbers from N to the smallest possible value, p_m.

In order to obtain this, let us consider the structure of a phase. We note first that, whatever is its internal architecture, all N weights have to be present, since the phase output is a convolved and therefore is the addition of products of samples with all weights.

A reduction of the sub-phases number can thus be obtained only if each sub-phase includes more than one multiplier. This can be obtained by structuring each sub-phase as shown in fig.3 scheme, i.e. if it is composed by cascading several serial-parallel multipliers and shift registers. Such a circuit has the following properties:

- if the X input to a multiplier is kept zero, and if the flip- flops internal to the multiplier are initially cleared, the multiplier works as a shift register;
- if the internal flip-flops of a multiplier contain data, and the X input is non-zero, the multiplier output is the sum of the initial internal content with the product W_iX_k;
- if two adjacent multipliers are spaced n^* stages apart, and they are operated synchronously (e.g. with the same X_k), a train of successive, distinct (i.e. non overlapping) numbers is generated and shifted to the right, where each number starts in the leftmost multiplier 0 as the product W_0X_k, then is incremented by W_1X_{k+1}, W_2X_{k+2}, etc. in the following multipliers, provided each number, a *partial* convolved, will never reach a length larger than n^*.



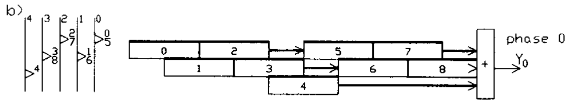


fig.4: a) A convolver with minimum number of phases.
b) Merging sub-phases.

Note that fig.3 scheme is the traditional systolic structure approach to convolution, requiring the successive samples to be separated by a sufficiently large interval n_{px} ($\geq n_w + n'$) as shown in the preceding paragraph. The same scheme can be used to implement polyphase structures, assuming $n* = n_{ty} = n_y + n_{pym}$, where n_{pym} is determined by the procedure (A) given in the preceding paragraph. (Note that in such procedure, the total length $n_y = n' + n_w + n_x$ of the convolved must be used, and not the smaller $n_w + n_x$ length of a single product, since in this case we could have interference between partial convolved during the successive accumulation of products, or in the final adder).

The scheme thus obtained for the same example as in fig.2 ($n_w = 4$, $n_x = 2$, $N = 9$, $n_{px} = 0$) is represented in fig 3a, where $n' = 4$, $n_y = 10$ and $p_m = \lceil n_y/n_x \rceil = 10/2 = 5$, so that five phases are necessary and a zero separation n_{pym} exists between successive convolved outputs.

The results given by the general procedure can also be directly obtained by reasoning on the canonical scheme, in order to obtain the sub-phases structure. The new sub-phases can be obtained by *merging* two or more canonical sub-phases, by applying the rule that two or more multipliers can be placed in the same sub-phase if their spacing is such that they can operate simultaneously (i.e. with the same X_k) on two distinct partial convolved. In order to do so, their spacing (see fig.4a) must be a multiple of n_x (since in the canonical scheme the multipliers spacing is n_x) not smaller than n_{ty}. This is just another way of telling the condition expressed by (A).

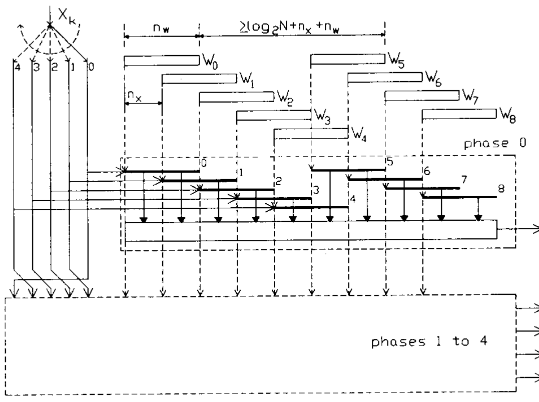


fig.5: A convolver with shared multipliers.

Note that the sub-phases in the reduced scheme can be considered as obtained by merging all sub-phases in the canonical scheme that are fed by samples X_k with the same $k \bmod p_m$ index. Note also that the adder in each phase requires a number of inputs (p_m) smaller than in the canonical scheme (N).

A further reduction can be obtained by observing that some of the new sub-phases can be merged again as shown in fig.4b. The merging can be performed between two (or more, in general) sub- phases whose multipliers do not overlap. This multipliers, however, must be fed by the same samples as in the original sub-phases. Since their distance in the new subphases is smaller that in the original ones (n_{ty}), they add to the convolved their respective terms before the convolved are completely output from the preceding multipliers. (Note also that in general the merging process offers various choices of sub-phases to be merged).

The final result of the described reduction process applied to a phase of fig.4a scheme is shown in fig.4b.

The case $n_w \leq n_x$ can be treated in a similar way. For brevity reason, and since the case $n_w > n_x$ is more common in applications, it will not be considered any further.

4. A SCHEME USING SHARED MULTIPLIERS

It will now be shown how the scheme of fig.4b, drawn for an example with $n_w > n_x$, can be further simplified. Note first that in such scheme each phase is composed by at least two sub-phases, due to the fact that multipliers in the canonical scheme are overlapped, and cannot therefore belong to the same sub-phase. Note however that whenever any multiplier is operating, all multipliers that are overlapped with it are inactive, since samples are cycled among sub-phases and consequently only one sub-phase can be active at any given time (i.e. it can have at least one multiplier operating, the other sub-phases simply shifting their content). It is then possible to adopt the circuit shown in fig.5, where each phase is composed by a continuous array of full adders, each one being fed by the logical sum of outputs from the gate arrays which are fed by the weight registers overlapping in that particular stage.

The overlapping multiplier stages can then be considered as shared among two or more multiplications, hence the denomination given to the new circuit.

The new circuit requires a smaller number of component and, moreover, it offers a more regular structure, as it appears in comparing fig.4 with fig.5. The latter property is particularly important for the design of modular and bit-sliced version, as it will be seen in a following paragraph.

As far as total *complexity* is concerned the two general schemes of fig.4b and fig.5 can be compared as follows. Note first that both are composed by the same number of phases. In addition, both schemes comprise a sample distributor and a set of weight registers, which, besides being identical, account for a small fraction of the total complexity, due mostly to the phases.

A phase in fig. 4b is composed by N multipliers, each of length (i.e. number of stages) n_w, by some shift registers and by the final adder (whose effect on cost is negligible). For simplicity only the multipliers will be considered. The total number of multipliers stages is: $L_A = n_w N$ and, considering the case of n_w multiple of n_x: $L_A = kn_x N$; (k integer). In a phase in fig.5 scheme , the total length L_B of the shared multipliers is: $L_B = (N-1)n_x + n_w$ or, for $n_w = kn_x$: $L_B = (N+k-1)n_x$. We have then: $\frac{L_A}{L_B} = \frac{k}{1+\frac{(k-1)}{N}}$. In practical cases k can be at most 3 or 4. If we consider the case of $N \gg 1$ (e.g. several tens),

$$\frac{L_A}{L_B} \simeq k.$$

For these reasons we will refer mostly to the shared multipliers scheme in the following.

The preceding schemes have been designed for non-signed numbers, and some modifications are necessary for samples and weights in two's-complement form.

Several methods can be used for two's-complement factors multiplication (see [10] for a survey). Among them, a most convenient method is the one already used in a similar convolver [12], based on a transformation of the product-array for obtaining an array composed with positive terms only, and by the addition of a negative constant to its value.

The array transformation consists in complementing all the array terms containing one and only one sign-bit (i.e. those terms having negative weight), and affects the gate-array producing the successive product array rows to be added in the serial-parallel multipliers. It requires a new clock (a "word-clock") given by a pulse corresponding to each sample sign-bit.

The constant to be added to each product is:

$$C_1 = -2^{n_x+n_w-2} + 2^{n_x-1} + 2^{n_w-1}$$

and can be accounted for with a single constant $C_N = NC_1$ to be added to the convolver output.

A similar method has been proposed by Agrawal [3], and used by Sips [6] in an inner product scheme, for adding simultaneously numbers in two's complement form.

The above modifications are not explicitly represented in the schemes.

5. SCHEMES FOR DIFFERENT OPTIMIZATION CRITERIA

The schemes discussed in the preceding paragraphs obtain the maximum sampling-rate for a given bit-rate, since samples follow each other with zero interval, $n_{px} = 0$. If such interval is allowed to be non zero, different schemes can be obtained with a smaller number of components. It is therefore interesting to study the trade-off between circuit complexity and samples interval (and consequently sampling-rate).

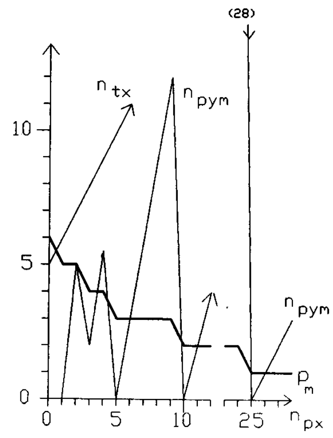


fig.6: Design parameters vs. samples separation for a convolver with $n_x = 5$; $n_w = 15$; $n' = 10$.

This can be easily done using procedure A with an assigned $n_p x \geq 0$. The minimum number of phases is then

$$p_m = \lceil n_y / n_{tx} \rceil$$

and, since $\lceil n_y/n_{tx} \rceil \leq \lceil n_y/n_x \rceil$, a number of phases smaller than in the case $n_{px} = 0$ can be obtained. The procedure:

$$\lceil n_y/n_{tx} \rceil = (n_y + n_{py})/(n_x + n_{px}) = p_m$$

can be used in various ways, e.g.:

- given n_y and n_{tx}, find p_m and n_{pym} (as suggested before);
- given n_{tx} and p_m, find n_{pym};
- given n_x, n_y and p_m, find n_{px} and n_{pym}.

As an example, fig.6 shows the results obtained for a convolver with $n_x = 5$; $n_w = 10$; $n' = 10$. The number of phases p_m and the convolved separation n_{pym} are plotted vs. the samples separation n_{px}. Note that the convolver cost can approximately be assumed as proportional to p_m.

It can be seen that the solutions corresponding to $n_{px} = 5$, or 10 or 25 are characterized by $n_{pym} = 0$, i.e. to the maximum convolved rate.

6. SCHEMES MODULARITY

Partitioning a scheme into modules is an important design step, particularly for VLSI implementation. The polyphase schemes described can be partitioned at different functional levels, thus offering a valuable flexibility.

At the *first, global level* the system appears as a linear array of phases, as can be seen in figures 2,4 and 5. The following remarks can be made:

- The samples distributor as drawn in the figures is a common part. It is in principle possible to include a distributor in each phase, in such a way that, with appropriate settings, their operation is equivalent to the single one (obtaining also to reduce the bus area).
- A second common part is the set of weight registers. They have been drawn in the figures as a part shared by all phases, since these use the same weight set. The connections to the multipliers are very regular, though rather numerous, and could be replaced by associating a weight register set to each phase.

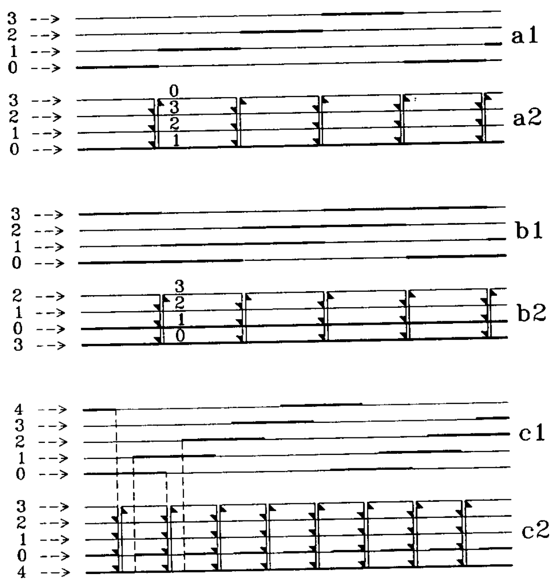


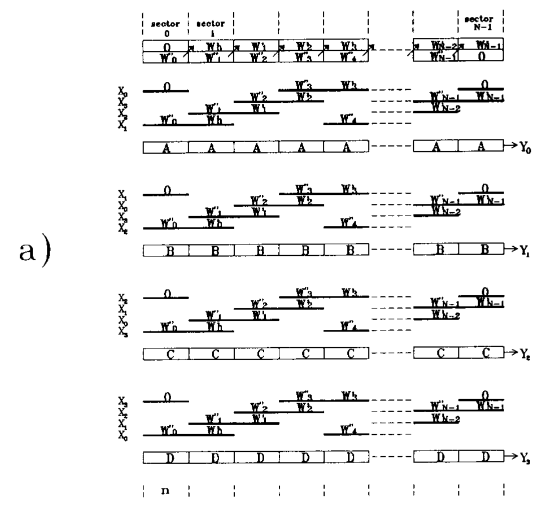
fig.7: Sample-busses with cyclic permutations.

A *second level* of modularity exists within each phase. Referring to fig.4 and 5 schemes, it can be seen that a periodicity p_m can be noticed within each phase if N is sufficiently large. Such periodicity permits to define each phase as a linear array of identical parts, each including p_m multipliers in fig.4, or p_m gate-arrays in fig.5.

Looking at the phase scheme in the same figures it can also be seen that they can be partitioned with vertical sections, defining *sectors* of equal width n_x.

The sectors structure is periodic with p_m period. Each sector can moreover be partitioned in identical bit-slices, which is the *third level* of modularity. Sectors become identical if the X-busses feeding the gate-arrays are cyclically permuted at the sectors interfaces, as shown in fig.7 examples. In cases where n_w is multiple of n_x (fig.7a,b) two types of slices are needed, having the same internal composition and differing only in the connection scheme with the following (or the preceding) slice. Note that the cyclic permutation is obtained at the sector's interfaces.

In fig.7b $n_w = 2n_x$, and two arrays overlap in each sector. Fig.7c represents a case in which n_w/n_x is not integer, requiring two slices with a different internal structure. It must however be noticed that the difference consists in having one or two gates, a)



b)

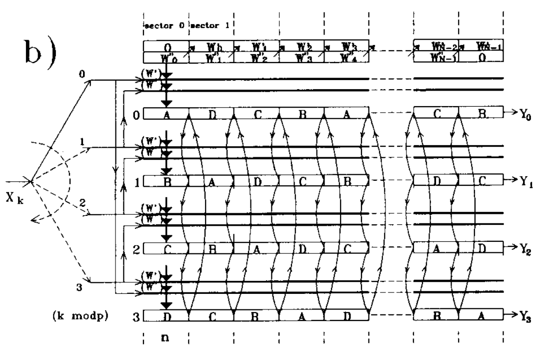


fig.8: a-A shared-multipliers scheme for $n_w/n_x = 2$; $n' = n_x$. b-Same scheme with cyclic phase permutations.

might be convenient to implement a single larger slice containing the same stages of all p_m phases (a ***convolver-slice*** vs. a ***phase-slice***).

There is therefore a choice to be made: implementing phases as units, obtaining the convolver by associating p_m phases, operated in parallel; or implementing modules composed by several convolver-slices (or by few ***convolver sectors***), and obtaining the convolver by cascading such modules. Note that in the same way also a single phase or a group of phases could be implemented.

Fig.5 scheme can be transformed into an equivalent one, provided the transformation does not affect the connections between: multiplier sectors of the same phase; weight registers and phase sectors, gate-arrays sectors and corresponding weight registers and samples.

Fig.8 shows two equivalent schemes for a convolver with: $n_w/n_x = 2$; $n' = n_x$. Fig.8a is drawn with the same rules of fig.5. For simplicity, the samples distributor has been omitted; moreover, each weight W_k is splitted in two parts, W'_k and $W"_k$, composed by the least, and respectively the most significant n_x bits. All W'_k and $W"_k$ can thus be accomodated in two registers. It can be seen in fig.8a that in sector 0-phase A only samples 0 and 1 are used (of course, at different times), and the same samples pair is used also in: sector 1-phase D, sector 2-phase C, sector 3-phase B, etc.

Similarly, samples 1 and 2 are common to sector 0-phase B, sector 1-phase A, sector 2-phase D, sector 3-phase C, etc.

Samples 2 and 3 are common to sector 0-phase C, sector 1-phase B, sector 2-phase A, sector 3-phase D, etc.

Samples 3 and 0 are common to sector 0-phase D, sector 1-phase C, sector 2-phase B, sector 3-phase A, etc.

All parts in sector 1 can be aligned with sector 0 parts so that parts on the same row are fed by the same sample-pairs, see fig.8b. The continuity between multipliers belonging to the same phase, is obtained by means of the connections shown in same figure, performing a cyclic permutation of phases. Note that the same connection scheme is used in all subsequent sectors interfaces.

The new scheme can conveniently be drawn on a cylindrical surface, where the gate-arrays will run in parallel to the cylinder's axis, while sectors of each phase will follow each other stepwise on a helicoidal line.

The new scheme offers some advantages, the most important being that the samples permutations among phases are not more necessary (they have been replaced by the phase permutations at the sectors interfaces). Moreover, only two (in general, $k = n_w/n_x$) samples are needed in each row (to be compared to four in fig.8a). Note also that weight registers are arranged in such a way that their vertical connections with the gate-arrays are uniform all over the scheme.

Fig.8b scheme has been shown as derived from fig.8a, but it can be drawn directly, with the following rules (for the case of n_w and n' multiple of n_x):

1. given n_w, n_x and n', determine p_m and n_{pym} (procedure A);
2. arrange p_m multipliers rows, each composed by N sectors n_x stages long, cyclically connected at each sector interface, as shown in fig.8b;
3. arrange the weights in $k = n_w/n_x$ registers, for W'_k, $W"_k$, $W"'_k \cdots$ (as shown in fig.8b for $k = 2$);
4. arrange similarly k gate arrays (each with $n_x N$ gates) for each multiplier rows, fed by a sample X_k and by W', $W"$ or $W"'$, $\cdots$ registers, respectively;
5. design a sample distributor for p_m outputs $X_0, X_1, \cdots, X_{pm-1}$ feeding the W' gates-arrays. For $k = 2$, the same p_m outputs feed the gates arrays for $W"$, one row earlier than for W'. For $k = 3$ the same p_m output of the sample distributor feeds the gates arrays for $W"$ and $W"'$ one row and respectively two rows in advance.

Note that in all figures the sample distributor has been placed at the left of the gate-arrays, so that X-pulses propagate from left to right, but it could alternatively be placed at their right. A comparison between the two schemes has been done in [12] for a similar case.

The operation of the new fig.8b circuit is the same of the fig.8a circuit from which it has been derived, since the transformation used does not affect the operation, as can be directly verified on fig.8b.

Assume the first sample as an X_0: it can be seen that a zero weight is input to sector 0-row 0 (phase A) multiplier, while

$W_0'X_0$ is added to sector 1-row 0 (phase D) and $W_0''X_0$ is added to sector 0-row 3 (phase D). At the end of sample X_0 the first term of convolved Y_0 will be: in sector 2-row 1 (the least significant n_x bits), in sector 1-row 0 (the following n_x bits), in sector 0-row 3 (the n_x most significant bits). The first term of convolved Y_0 has been produced in phase D.

Note that sample X_0 will be multiplied by all following weights: the corresponding results must however be disregarded since they do not contain the term W_0X_0. The first convolved will appear at the occurrence of sample X_{N-1}, at the output $(N-1)\text{mod}p_m$.

With the following sample X_1, the first term W_0X_1 of the convolved Y_1 is generated in phase A with the same procedure just described for Y_0 in phase D. Simultaneously, the second term W_1X_1 is added in phase D to the first term W_0X_0: the n_x least significant bits of W_1X_1 will be added to the n_x least significant bits of W_0X_0 in sector 2-row 1 and the n_x least significant bit of the result will be shifted in sector 3-row 2, while the 2 n_x more significant bits will be added to the 2 n_x most significant bits of W_0X_0 in sector 1-row 0 and in sector 2-row 1, the result being shifted in following sectors-rows of same phase D.

Note that the carry from the first addition is automatically taken into account, since each phase is composed by a continuous array through the cyclic permutation connections.

Note also that the total length allotted to a partial convolver in each phase is $4n_x$ bits, i.e. 4 sectors, 3 for each product, 1 for the overflows generated in the successive product accumulation. The 3 n_x least significant bits (i.e. three sectors) are filled by the first term, W_0X_0; at the end of X_1 the carry produced by the partial convolved $W_0X_0 + W_1X_1$ will be stored in the last n_x bits long sector, that will operate as a counter of the overflows produced by the successive accumulation of products.

As it can be seen from the figure, also the new scheme can be decomposed into modules.

In cascading modules, no matter in what scheme, a problem arises if these are in distinct chips (e.g. in WSI), in relation to the transmission delay in inter-chip connections. Such a problem has been briefly discussed in [13] in relation to the timing aspects in similar schemes and with the following conclusion. Samples and clock must be fed at the left side of the structure, in order that signals propagate in one direction only, left to right. It is consequently possible to interleave with the normal slices, at regular intervals, buffer slices, composed only by a flip-flop for each signal. Buffer slices provide reshaping and risynchronization of clock and signals. Moreover, since they do not include full-adders, they can be used at the interface between two cascaded chips as a buffer stage for the capacitance load of the connection, thus minimizing the effect on attainable bit rate.

In case of only one type of slice it becomes possible to implement convolvers which are reconfigurable, i.e. they can be programmed for arbitrarily assigned n_w, n_x and N. This possibility has already been discussed in [12], and can be obtained by means of gates for the selection of the inter-slice connection scheme, operated by a variable stored in a flip-flop.

These can be arranged in a *configuration control* register. A similar arrangement can be adopted for by-passing faulty slices, thus obtaining fault-repairing schemes.

A significant comparison can be made between the polyphase fig.8a scheme and the elementary monophase scheme of fig.3, in terms of hardware requirements and performance. Consider the case of $n_w = kn_x$ and $n' = k'n_x$. The numbers n_{FAp} of full adders and n_{FFp} of flip flops in the polyphase scheme can be shown to be:

$$n_{FAp} = (k + k' + 1)n_xN \quad ; \quad n_{FFp} = 2n_{FAp}$$

The corresponding numbers n_{FA1} and n_{FF1} in the monophase sheme are:

$$n_{FA1} = kn_xN$$

$$n_{FF1} = (2k + k' + 1)n_xN$$

so that:

$$n_{FAp}/n_{FA1} = (k + k' + 1)/k$$

$$n_{FFp}/n_{FF1} = 2(k + k' + 1)/(2k + k' + 1)$$

The ratio f_{sp}/f_{s1} between the sampling rates in the two circuits is:

$$f_{sp}/f_{s1} = k + k' + 1$$

As an example, consider the case: $n_w = 2n_x$; $n' = n_x$. We have then: $k = 2$; $k' = 1$; $n_y = 4n_x$; $p = 4$ and: $n_{FAp}/n_{FA1} = 2$; $n_{FFp}/n_{FF1} = 4/3$; $f_{sp}/f_{s1} = 4$ i.e. a four time increase in sampling rate with less than two time increase in hardware.

A comparison between polyphase convolvers and previously proposed serial input serial output convolvers has not yet been done. It can be noted that the latter do not generally offer the possibility to operate with zero samples separation, this condition requiring some sort of parallelism and redundancy, as seen in the proposed polyphase schemes.

7. CONCLUSION

It has been shown how serial-input convolvers can be designed on the basis of a polyphase architecture, using a method permitting to assign arbitrarily the weight length n_w, the samples length n_x and the number N of convolution terms, under the condition of zero or arbitrarily assigned separation between successive samples or between convolved. Two general schemes have been found, one based on the use of distinct serial-parallel multipliers, the other on multipliers which are partially shared among different multiplications, thus affording a simpler structure. The partition of such convolver schemes into modules in relation to an easier design and VLSI implementation has also been discussed.

REFERENCES

[1] Dadda,L., Ferrari.D., Digital multipliers: a unified approach, Alta Frequenza, vol.XXXVII, n.11, pp.321E-328E, nov. 1968.

[2] Dadda,L., Multiple addition of binary serial numbers, Proc. 4th IEEE Symp. Comput. Arithmetic, pp.140-148, oct. 1978.

[3] Agrawal,D.P., Rao,T.R., On multiple addition of signed binary numbers, IEEE Trans. Comput., vol C-27, n.11, pp.1068-1070, nov. 1978.

[4] Swartzlander,E.E., Gilbert,B.K., Arithmetic for ultra-high-speed tomography, IEEE Trans. Comput., vol.C-29, n.5, pp.341-353, may 1980.

[5] Kung,H.T., Why systolic architecture?, IEEE Computer, vol. 15, pp.37-46, jan. 1982.

[6] Sips,H.J., Bit sequential arithmetic for parallel processors, IEEE Trans. Comput., vol C-33, n.1, pp. 7-20, jan. 1984.

[7] Danielsson,P.E., Serial-parallel convolvers, IEEE Trans. Comput., vol.C-33, n.7,pp.652-667, july 1984.

[8] McCanny,J.V., McWhirter, J.G., Wood, K., Optimized bit level systolic array for convolution, IEE Proc. F, Commun.

Radar and Signal Processing, vol. 131, n.6, pp. 632-637, oct. 1984.

[9] Urquhart,R.B., Wood,D., Systolic matrix and vector multiplication methods for signal processing, ibid., pp.623-631, oct. 1984.

[10] Dadda,L., Fast multipliers for two's-complement numbers in serial form, Proc. IEEE Symp.Comput.Arithmetic,Urbana IL, june 1985.

[11] McCanny, J.V., Evans, R.A., McWhirter, J.G., Use of bi- directional data flow in bit-level systolic array chips, Electronic Letters, vol.22, n.10, may 1986.

[12] Dadda,L., Breveglieri,L., Serial-input serial-output bit sliced convolver, Proc.ICCD'88, pp.490-495, oct.1988.

[13] Dadda, L., Serial-input polyphase convolvers, International Conference on Systolic Arrays, Killarney, Ireland, 31st May-2nd June, 1989.