

Synthesis of Coefficient Bit Reordering Module for Folded Bit-Plane Array

I. Milentijevic, V. Ciric

Faculty of Electronic Engineering, University of Nis
Beogradska 14, PO Box 73, 18000 Nis, Serbia and Montenegro
{milentijevic, vciric}@elfak.ni.ac.yu

Abstract: *The goal of this paper is synthesis of Coefficient Bit Reordering Module (CBRM) for folded bit-plane FIR filter. Basic functional requirements for CBRM are: bit-serial entering of coefficient bits (initialization mode), reordering in accordance with required mathematical modulo description (initialization mode) and cyclic-parallel feeding of architecture (run mode). Mathematical description of the problem along with the folding set assignment, as well as the solution for mapping of coefficient bits onto the hardware architecture by rigorous mathematical procedure are presented in this paper. Synthesis of efficient programmable hardware for CBRM and principle of operation are described in detail. Programmability of hardware assumes changing of both number of coefficients and coefficient length with aim to provide flexibility and wider application area for folded bit-plane arrays.*

Keywords: Mobile devices, adaptive FIR filtering, folded bit-plane, coefficient entering.

1. Introduction

Cellular-phone technology is changing rapidly. There is an increasing number of wireless-communications standards, including variants of the IEEE 802.11 wireless LAN specification, code-division multiple access, the global system for mobile communications, and emerging third-generation technologies. Traditionally, devices need a separate chip to work with each standard. However, as wireless technologies mature, service providers differentiate themselves by offering new features, such as multimedia capabilities. Providing each feature typically requires a separate chip, or essence, multiple circuitry systems physically joined on a piece of silicon [1]. The additional circuitry adds cost, takes up space, increases power usage in mobile devices, and increase product-design time. This problem can be solved using adaptive approach. With this approach, software can redraw a chip's physical circuitry on the fly, letting a single processor to perform multiple functions [1-4]. In adaptive FIR filtering, circuitry typically can be changed on the fly, as software instructions tell circuitry control logic to alternate the computation. Reconfiguration can occur within as little as a few clock cycles. Systems can use various mechanisms to input reconfiguration information [1].

Designing of folded bit-plane architectures for FIR filtering requires the involvement of specialized hardware module for feeding the architecture with coefficient bits. The module should be based on mathematical dependencies that are inherent to the folding dependencies between operations and functional units. The goal of this paper is synthesis of coefficient bit reordering module (CBRM) for folded bit-plane FIR filter. Basic functional requirements for the module are: bit-serial entering of coefficient bits (initialization mode), reordering in accordance with required mathematical modulo description (initialization mode) and cyclic-parallel feeding of architecture (run mode). In this paper, mathematical description of the problem is given as well as folding set assignment along with solution for mapping of coefficient bits onto the hardware architecture by rigorous mathematical procedure. The attention is concentrated on synthesis of efficient programmable hardware for CBRM. Programmability of hardware assumes changing of both number of coefficients and coefficient length with aim to provide flexibility and

wider application area for folded bit-plane arrays [5-8]. The principle of operation for CBRM is described in detail.

With aim to clarify synthesis process of CBRM for folded bit-plane array we give a brief review of folding transformation.

2. Basic elements of folding technique

The folding technique is introduced by K.K. Parhi and described in [3,4]. The synthesis of folded data path is explained in Fig. 1 a) and Fig. 1 b). Fig. 1 a) shows an edge $U \rightarrow V$ with $w(e)$ delays, while Fig. 1 b) depicts the corresponding folded data path. The data begin at the functional unit H_u , which has P_u pipelining stages, pass through

$$D_F(U \rightarrow V) = Nw(e) - P_u + v - u \quad (1)$$

delays, and are switched into the functional unit H_v at the time instances $Nl + v$, where N is the number of operations folded to a single functional unit (folding factor), while u and v are the folding orders of nodes U and V that satisfy $N-1 \geq u, v \geq 0$. A folding set, S , is defined as an ordered set of operations, which contains N entries, executed by the same functional unit. For a folded system to be realizable $D_F(U \rightarrow V) \geq 0$ must hold for all of the edges in the DFG. Once valid folding sets have been assigned, retiming can be used to satisfy this property or determine that the folding sets are not feasible [4].

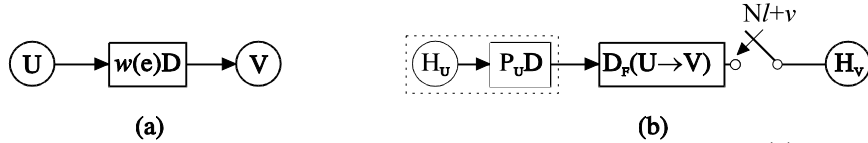


Fig. 1. The synthesis of folded data path. (a) An edge $U \rightarrow V$ with $w(e)$ delays; (b) The corresponding folded data path.

After this introduction to folding technique, we give the short description of folded bit-plane array with changeable coefficient number and coefficient length.

3. Folded bit-plane semi-systolic architecture

Output words $\{y_i\}$ FIR filter are computed as

$$y_i = c_0 x_i + c_1 x_{i-1} + \dots + c_{k-1} x_{i-k+1} \quad (2)$$

where c_0, c_1, \dots, c_{k-1} are coefficients while $\{x_i\}$ are input words. Computation (2) can be realized in different manners. When high performances are required systolic arrays are frequently used. Semi-systolic arrays share with systolic arrays desirable simplicity and regularity properties, in addition to their pipelining and multiprocessing schemes of operation. The only difference is that the broadcasting of data to many PEs in one time step is allowed in semi-systolic arrays, while systolic arrays are restricted to temporal locality of communication. Also, the existence of some additional connections can be allowed for semi-systolic architectures [9, 10].

The folded bit-plane architecture (FBPA) is semi-systolic architecture, obtained by application of folding technique onto semi-systolic bit-plane FIR filter architecture (BPA). The transformation of source DFG for the bit-plane architecture enables the synthesis of fully pipelined folded FIR filter architecture with changeable number of coefficients as well as the coefficient length and adjustable folding factor [5]. New assignment of folding sets for the application of folding technique to the BPA is proposed in [8]. The assignment supports the changing of operations in folding sets. Using of proposed folding set assignment different operations can be mapped onto

the different hardware units in the fixed structure array. Using register minimization, along with the folding transformation, not only the number of functional units has been reduced but also the area consumed by memory in the folded architecture has been kept to a minimum [11]. The derived architecture has kept desirable features of source architecture such as extensive pipelining, high regularity, truncation of LSBs of intermediate results without any loss of accuracy [12]. In the comparison to source BPA array, the folded array is restricted for the factor N . The number of basic cells is reduced to the number of basic cells in one plane of source architecture [12]. The FBPA is obtained by folding different operations onto a single functional unit as it is shown in Fig. 2. The following notation is adopted: k_c – number of coefficients, m_c – coefficient length, c_i^j – bit of coefficient c_i (with weight 2^i), N – folding factor, k – number of folding sets, p – position of operation within the DFG ($0 \leq p \leq L-1$).

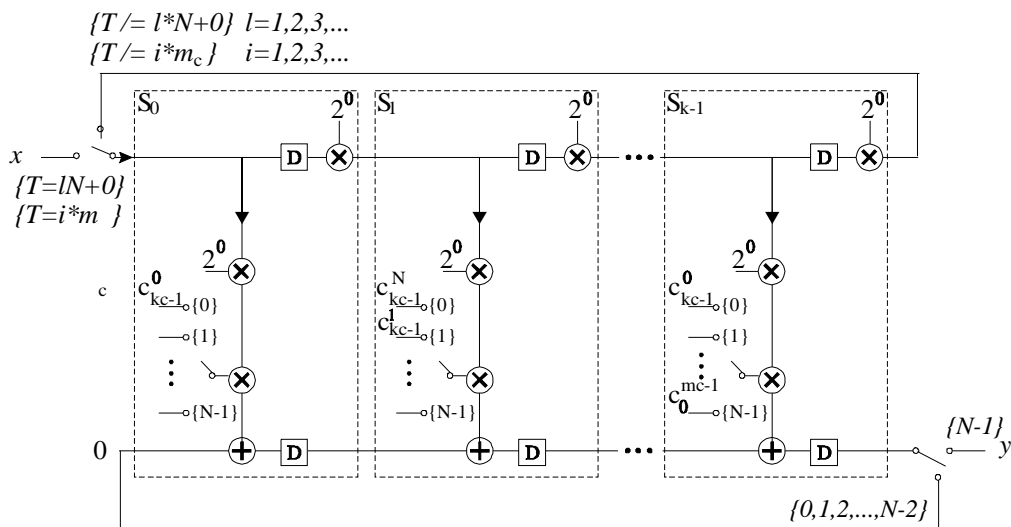


Fig. 2. The DFG (Data Flow Graph) of the FBPA with changeable number of coefficients and coefficient length.

The FBPA with $k=3$ and $m=4$ is shown in Fig. 3.

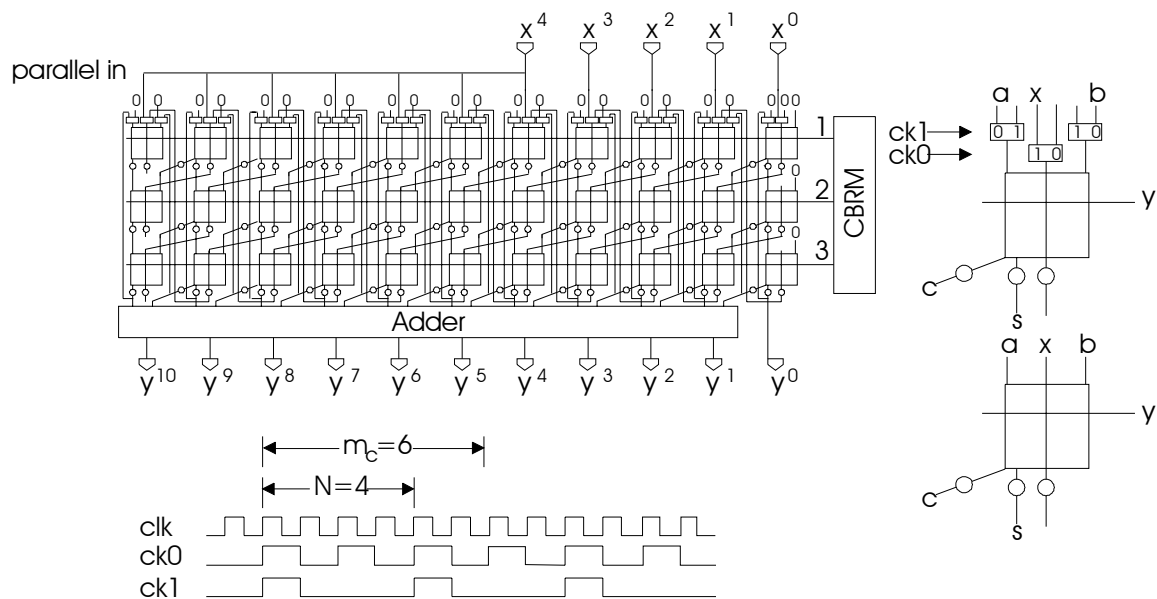


Fig. 3. The FBPA for $k=3$ and $m=4$.

As the spotlight of this paper is the synthesis of CBRM (Fig. 3.), let us introduce principles of adaptive FIR filtering onto the FBPA architecture.

4. Adaptive FIR filtering on the folded bit-plane array

There are k folding sets ($S_0, S_1, S_2, \dots, S_{k-1}$) where k is equal to the number of taps. Each folding set contains m_c operations, i.e. the folding factor N is equal to the coefficient length m_c . Folding sets S_0, S_1, \dots, S_{k-1} are shown in dashed boxes (Fig. 2). Each folding set contains N operations. In order to clarify the method of operation, the hardware section that performs the operation from set S_s ($s=0, 1, \dots, k-1$) is denoted with S_s , also. Initially, the computation starts in folding set S_0 where the product $2^0 \cdot c_{k_c-1}^0 \cdot x_0$ is obtained in the first clock cycle. In the next clock cycle folding set S_1 generates the partial product $2^1 \cdot c_{k_c-1}^1 \cdot x_0$ adding previously computed partial product from S_0 . Thus, the value $(2^0 \cdot c_{k_c-1}^0 \cdot x_0) + (2^1 \cdot c_{k_c-1}^1 \cdot x_0)$ is entered into the next section, which performs the operations from S_2 , in the third clock cycle. The next important time instance is $(k+1)st$ clock cycle. In that clock cycle both input data path and summation path are folded from section S_{k-1} to S_0 . In input data path product $2^k \cdot x_0$ is present at input of the section S_0 , while in the summation path $(2^0 \cdot c_{k_c-1}^0 \cdot x_0) + (2^1 \cdot c_{k_c-1}^1 \cdot x_0) + \dots + (2^{k-1} \cdot c_{k_c-1}^{k-1} \cdot x_0)$ enters the same section. S_0 adds $2^k \cdot c_{k_c-1}^k \cdot x_0$ to the entered sum. However, the computation for the coefficient c_{k_c-1} is not finished yet. The complete product $c_{k_c-1} \cdot x_0$ is obtained in the section $S_{(m_c-1) \bmod k}$ during clock cycle m_c . The computation of $c_{k_c-2} \cdot x_1$ starts in $(m_c+1)st$ clock cycle. The section $S_{m_c \bmod k}$ computes

$$\begin{aligned} & \{(2^0 \cdot c_{k_c-1}^0 \cdot x_0) + (2^1 \cdot c_{k_c-1}^1 \cdot x_0) + \dots + (2^{m_c-1} \cdot c_{k_c-1}^{m_c-1} \cdot x_0)\} + 2^0 \cdot c_{k_c-2}^0 \cdot x_1 \\ & = (c_{k_c-1} \cdot x_0) + (2^0 \cdot c_{k_c-2}^0 \cdot x_1). \end{aligned}$$

In order to explain method of operation we have started the explanation with computation of y_{k_c-1} that is the first result where all coefficients are included. Let us note that in the simultaneous process the architecture starts with computation of all results $y_{k_c-1}, y_{k_c-2}, \dots, y_0$ in reverse ordering. The first completely generated result at output is y_0 with latency of $2m_c - N$ clock cycles. New result y is generated every N clock cycles.

The example that illustrates the described algorithm is shown in Fig. 4. The data flow through the folded architecture, which looks like a nucleic chain, for case $k=3, N=4, k_c=2$ and $m_c=6$ is given in Fig. 4.

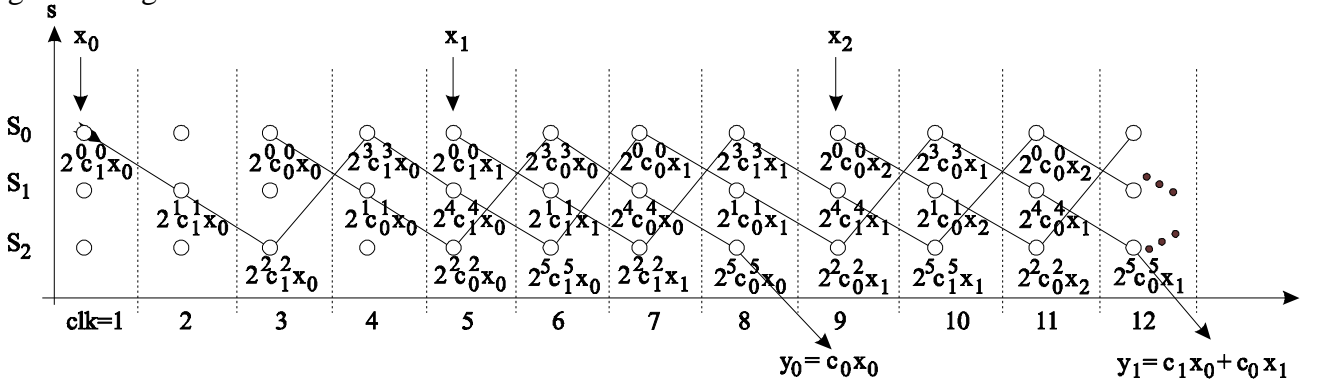


Fig. 4. Data flow for folded architecture for $k=3, N=4, k_c=2$ and $m_c=6$.

The proposed architecture supports the operation with changeable number of coefficients and coefficient length.

In order to make the FIR filtering on FBPA possible, the coefficient bits must be entered into architecture in proper order. First step to the synthesis of CBRM is finding dependencies between coefficient bits and time instances when specific DFG nodes use them.

5. Mapping of Coefficient Bits onto the DFG Nodes

Assignment of folding sets (S_s/r) proposed in [8], where s is index of folding set while r represents folding order, is involved with aim to enable the synthesis of the folded FIR filter architecture that will support the changing of both coefficient number and coefficient length. The assignment is described with

$$\begin{aligned} s &= p \bmod k \\ r &= p \bmod N. \end{aligned} \quad (3)$$

Folding set assignment (3) enables the changing of operations in folding sets. In other words the different operations can be mapped onto the different hardware units in fixed array structure [8, 12]. The equations (3) provide k folding sets where each folding set contains N operations. For the coefficients, k_c , and the coefficient length, m_c , the total number of operations, L , is:

$$L = k_c \cdot m_c = k \cdot N \quad (4)$$

Let us note that, the number of folding sets is not obligatory equal to the number of coefficients. In order to obtain mapping dependencies, unfolded DFG from Fig. 5 should be used, where each operation from DFG (Fig. 5) stands for multiplication of input data words by one coefficient bit from Fig. 2. Assignment of position numbers to operations in DFG (p) is done as follows: the leftmost operation is denoted with 0, while the rightmost operation is denoted with $L-1$ (Fig. 5).

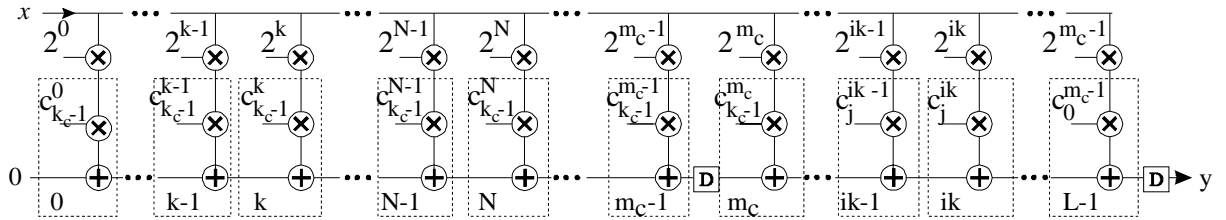


Fig. 5. Unfolded DFG

Assume that operation p ($0 \leq p \leq L-1$) from Fig. 5 performs multiplication of input data words by coefficient bit c_i^j . According to (3), operation p belongs to folding set $s = p \bmod k$, with folding order $r = p \bmod N$. In other words, folded architecture multiplies input data word (Fig. 2) by coefficient c_i^j on folding set s ($0 \leq s \leq k-1$) in time instance $\delta \cdot N + r$ ($0 \leq r \leq N-1$; $\delta = 0, 1, 2, \dots$). The operation that has position in DFG equal to p (Fig. 5), according to folding set assignment (3), can be described as

$$p = m_c \cdot (k_c - (j + 1)) + i. \quad (5)$$

The dependency between folding set s and folding order r of coefficient bit c_i with weight 2^j , using (3) and (5), is obtained as:

$$\begin{aligned} s &= (m_c \cdot (k_c - (j + 1)) + i) \bmod k \\ r &= (m_c \cdot (k_c - (j + 1)) + i) \bmod N \end{aligned} \quad (6)$$

Expression (6) describes the folding set s that performs multiplication by coefficient c_i^j in time instances $\delta \cdot N + r$ ($0 \leq r \leq N-1$; $\delta = 0, 1, 2, \dots$).

Inverse dependencies, denoted as $i = f(s, r)$ and $j = g(s, r)$, can be obtained by mapping position of operation p to matrix $A_{k \times N}$ in accordance with folding set assignment (3). Each column in matrix $A_{k \times N}$ represents one folding set S_s ($0 \leq s \leq k-1$) and each row stands for time

$$i = (((s + a \cdot k) - r) \cdot N + r) \bmod m_c, a = \begin{cases} 0, & s \geq r \\ 1, & s < r \end{cases},$$

$$j = k_c - 1 - \left(\frac{(((s + a \cdot k) - r) \cdot N + r)}{m_c} \right), a = \begin{cases} 0, & s \geq r \\ 1, & s < r \end{cases}.$$
(13)

According to (6) and (13) CBRM can be synthesized as two dimensional array which dimensions depends only on parameters k and N . Parameters k_c and m_c determine the reordering of coefficient bits only. In other words, it is obvious that size of the CBRM depends only on number of coefficients and coefficient length.

6. Synthesis of Coefficient Bit Reordering Module

In the respect to previous dependency-analysis (Eqs. 6 and 13), CBRM has two operational modes. First, initialization mode, when coefficient bits are entered into the CBRM, and the second, run mode, when CBRM is feeding array with coefficient bits.

The Coefficient Bit Supply Module (CBSM) has to provide the proper ordering of coefficient bits in accordance to (6) and (13). The internal structure of CBSM is given in Fig. 6a.

The CBSM is implemented as two-dimensional array of latches where each latch stores one bit of the coefficient (Fig. 6a). The number of rows is equal to the number of folding sets in FBSM (k) while the number of columns is equal to the FBSM's folding factor (N). Output from each row is feeding one folding set of the folded array with coefficient bits. Rows are implemented as shift registers, so during the run mode coefficients rotate through the rows from right to left, feeding each folding set of folded array with coefficient bits in correct order (solid arrows in Fig. 6a). Initial state of bits within CBSM is shown in Fig. 6b.

Problem of providing the correct bit order, during initialization mode, can be solved using the property of modulo dependence in (6). Due, coefficient bits are entered in bit serial manner (less significant bit first) starting from coefficient c_{k_c-1} through the latch denoted with [1,1] in Fig. 6a.

The coefficient bit is shifted in next time instance for one row up and one column to the left, while next bit enters CBSM. When the coefficient bit reach the first row (column) it is recycled back to the last row (column). The trace of the first coefficient bit $c_{k_c-1}^0$ can be described with mapping of time instances $t \in \{1, 2, \dots, k \cdot N\}$ onto the array position $[\alpha, \beta]$:

$$\alpha = ((t-1) \bmod k) + 1$$

$$\beta = ((t-1) \bmod N) + 1.$$
(14)

The number of clock cycles, required for initializing the structure, is $k \cdot N$.

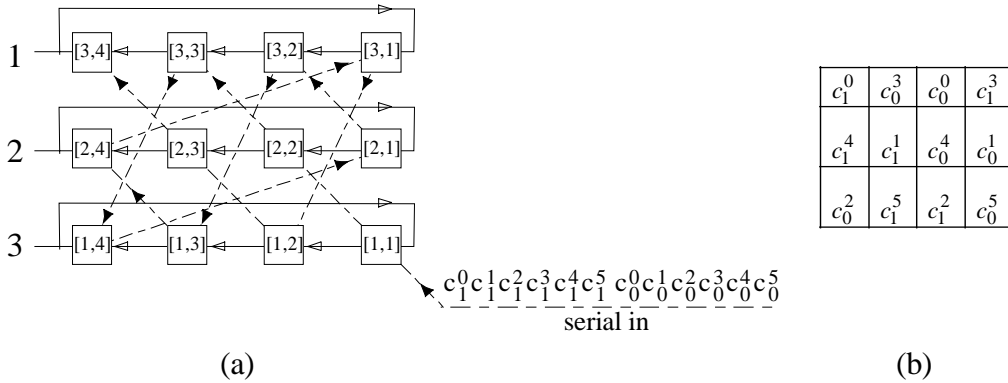


Fig. 6. a) CBRM for $k=3$, $N=4$, $k_c=2$ and $m_c=6$; b) layout of coefficient bits after initialization

The CBRM shown in Fig. 6, provides both reordering of coefficient bits during initialization mode and feeding the architecture with coefficient bits during the run mode.

7. Conclusion

The proposed coefficient bit reordering module enables feeding the folded bit-plane array with coefficient bits in proper order, in accordance with required mathematical modulo description of mapping of operations onto functional units in folded array. Functionality of CBRM assumes: bit-serial entering of coefficient bits (initialization mode), reordering of coefficient bits (initialization mode) and cyclic-parallel feeding of architecture (run mode). The initialization time depends on product of number of coefficients and coefficient length.

The module is synthesized using mathematical dependencies that are inherent to the folding dependencies between operations and functional units. Programmability of CBSM assumes changing of both number of coefficients and coefficient length with aim to provide flexibility and wider application area for folded bit-plane arrays. The synthesized module is able to handle feeding of folded array that performs computation with different number of coefficients and coefficient length. Simplicity of reordering array and regularity of its connections enables the area efficient implementation of coefficient bit reordering module.

8. References

1. L. Paulson, L. Garber, "Reconfiguring Wireless Phones with Adaptive Chips", *IEEE Computer*, Vol. 36, Number 9, September 2003, pp. 9-11.
2. D. Reuver, H. Klar, "A configurable Convolution Clup with Programmable Coefficients", *IEEE Journal of Solid State Circuits*, Vol. 27, No. 7, July 1992, pp. 1121 - 1123.
3. T. C. Denk, K. K. Parhi, "Synthesis of Folded Pipelined Architectures for Multirate DSP Algorithms", *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, Vol. 6, No. 4, Dec. 1998, pp. 595-607.
4. K. K. Parhi, "VLSI Digital Signal Processing Systems (Design and Implementation)", *John Wiley & Sons*, In., New York, 2000.
5. I. Milentijevic, V. Ciric, O. Vojinovic, T. Tokic, "Folded Semi-Systolic FIR Filter Architecture With Changeable Folding Factor", *Neural, Parallel & Scientific Computations, Dynamic Publishers*, Atlanta, Vol. 10, No 2, 2002, pp. 235-247.
6. I. Milentijevic, V. Ciric, T. Tokic and O. Vojinovic, "Folded Bit-Plane FIR Filter Architecture with Changeable Folding Factor", *DSD 2002, EUROMICRO - Digital System Design*, Dortmund, Germany, September 2002. pp. 45-52.
7. I. Z. Milentijevic, V. Ciric, T. Tokic and O. Vojinovic: "FPGA Implementation of Folded FIR Filter Architecture with Changeable Folding Factor". *Facta Universitatis, Ser. Electronics and Energetics*, Vol. 15, No 3, December 2002, pp. 451-464.
8. I. Milentijevic, V. Ciric, "Assignment of Folding Sets for Adaptive FIR Filtering on Folded Array", *WPS-DSD 2003, EUROMICRO*, Belek, Turkey, September 2003. pp. 21-22.
9. T. Noll, "Semi-systolic Maximum Rate Transversal Filters with Programmable coefficients", *Workshop of Systolic Architectures*, Oxford, 1986, pp. 103-112.
10. I. Milentijevic, M. Stojcev, D. Maksimovic, "Configurable Digit - Serial Convolver of Type F", *Microelectronics Journal*, Vol. 27. No. 6, Sep. 1996, pp. 559-566.
11. V. Ciric, I. Milentijevic, O. Vojinovic, "Retiming and Register Number Minimization for Adaptive FIR Filter Architecture", *Proceedings of a Workshop on Computational Intelligence and Information Technologies*, Nis, Serbia and Montenegro, October 2003, pp. 93-96.
12. V. Ciric, I. Milentijevic, "Configurable Folded Bit-Plane Architecture for FIR Filtering", *Proceedings of the WPS-DSD 2003, 29th EUROMICRO Conference*, Belek, Turkey, September 2003, pp. 23-24.