Combined Method for Retiming and Folding SDFs

Anatolij Sergiyenko, Kostiantyn Glukhenko, Pavel Regida

National Technical University of Ukraine „KPI“, 37 Peremogy ave., Kyiv, Ukraine

aser@comsys.kpi.ua, glukhenko.kostiantyn@gmail.com, regidapavel@gmail.com

Abstract. *Methods for retiming the synchronous dataflow graphs (SDF), which are mapped into the pipelined datapaths, are considered. A method for retiming the spatial SDF is proposed that provides the minimum cost-performance ratio of the datapath. Comparison of different retiming methods applied to the IIR digital filter design shows the high efficiency of the proposed method.*

Keywords

Retiming, folding, datapath, VHDL, FPGA, SDF, DSP.

1 Introduction

The modern computer chips are operating at the clock frequencies which increase up to several GHz. This success is achieved due to the pipelining of computing and transfer of data. The methods of pipelined processor design consist in its structural synthesis, RTL description, and compiling them to the gate level. The most of methods are based on synchronous data flow graph (SDF) algorithm representation, and its transformations [1].

The methods of SDF optimization like retiming, folding, unfolding, and pipelining are well known [2]. Here a new method is proposed, which combines both retiming and folding.

2 SDF and its optimizations

SDF is a directed graph $G = (V,E)$. In the microelectronics SDF is considered as some computational structure, in which the node $v \in V$ represents some logic network with a delay of d units. The edge $e \in E$ represents a wire, which contains $w[e]$ marks. These marks mean FIFO registers, which delay the variables to w cycles of the algorithm. The algorithm cycle lasts T_A, and one period of the clock signal of the structure is equal to T_C. For the one-to-one mapping SDF to the structure these periods are equal to each other, i.e. $T_A = T_C$. The minimum value of T_C is equal to the critical path, which consists of a set of nodes connected by the edges e for which $w[e] = 0$.

The retiming consists in the exchange of the values $w[e]$ in edges, which does not infer the algorithm. A single retiming step consists in removing a group of marks in the input edges of the node v, and placing them in its output edges. The goal of retiming is the value T_C minimization. Another goal is minimizing the amount of registers. The different versions of this method are known like cut-set retiming, pipelining, and re-pipelining [3].

The method of SDF folding consists in the following. The sets of up to c nodes of SDF are mapped to the nodes of the folded SDF. The data flows in the folded SDF are arranged in the manner that the algorithm cycle is calculated for c clock cycles. To direct the data flows properly the artificial nodes of multiplexors are put in the folded SDF. The amount of registers in both SDFs is equal to each other. Really, the folded SDF is synthesized in three steps. At the first step the resources (nodes of the folded SDF) are selected, then the operator shedule is searched, and at the last step the operator assignment is searched. The method of c-slow retiming is a variant of SDF folding when c equal SDFs are glued together [3].

The method of SDF unfolding is based on unfolding the initial algorithm. Therefore, it provides the ratio $cT_A = T_C$. Moreover, it provides the theoretically minimum value of T_A at the cost of increased hardware [4].

The retiming method is very efficient, but it gives only the straight-forward solutions when $T_A = T_C$. The SDF folding method gives the scalable solutions when the algorithm period is equal to $T_A = cT_C$, and the clock period T_C is much less than one in the initial SDF [3]. But its complexity is very high. Besides, the steps of its synthesis have different goals. The first step minimizes the hardware at the cost of the speed, the second one minimizes the speed.

3 Method of spatial SDF optimization

When we combine the steps of resource selection, operator sheduling, and operator assignment in a single step then we can substantially improve the synthesis of the pipelined datapath. In the presentation the method which utilizes this idea is proposed. The operator scheduling for SDF means the assignment of time events for its nodes. The processor structure is derived by the homomorphic transform of SDF into the structure graph. Such a transform is implemented by gluing the nodes, which are mapped in a single processing unit (PU), and which are fired in different times [5].

The homomorphic transform of the graph is implemented by the assignment of tags of and operation types to the nodes. Then the nodes with the equal PU numbers are glued into a PU node. Therefore, each SDF node must have the tag with the parameters of operator event time (clock cycle), operator type, and PU number. The algorithm mapping consists in the assignment of different values to the node tags, and in calculating the optimization criterium. Such a tag set means a single structural solution of the processor. These tags can be represented by the vectors in the space $\mathbb{Z}^3$.

In [6] a method of pipelined datapath synthesis is described, in which SDF is represented in the space space $\mathbb{Z}^3$ as the algorithm configuration $\mathsf{K}_G = (K, D, A)$, where K is matrix of vectors $\boldsymbol{K}_i$ representing nodes, D is matrix of vectors $\boldsymbol{D}_j$ representing edges, A is the incidence matrix of SDF. In the vector $\boldsymbol{K}_i = (k_i,s_i,t_i)^T$ the coordinates k_i, s_i, t_i are equal to operator type, PU number, and clock cycle, respectively. Therefore, vectors $\boldsymbol{K}_i$ are the tags, which code the properties of SDF nodes, and such a graph is named as the spatial SDF.

The matrix D is derived from the equation $D = KA$. Therefore, the matrix K codes some structure solution. But this solution is correct only when the vectors $\boldsymbol{K}_i$ satisfy a set of the following conditions. The spatial SDF is correct if the matrix K contains none couple of equal vectors, i.e.

$$\forall\ \boldsymbol{K}_i,\boldsymbol{K}_j\ (\boldsymbol{K}_i \neq \boldsymbol{K}_j, i \neq j\ ). \tag{1}$$

The operator schedule is correct iff nodes, which are mapped in a single PU, are assigned to different clock cycles, i.e. taking into account the circular schedule

$$\forall\ \boldsymbol{K}_i,\boldsymbol{K}_j(k_i = k_j,\ s_i = s_j\ ) \Rightarrow t_i \not\equiv t_j \bmod c. \tag{2}$$

The operators of the same type are mapped into the same type PU, and its number is no higher than the folding factor c, i.e.

$$\boldsymbol{K}_i,\boldsymbol{K}_j \in K_{p,q}(k_i = k_j = p,\ s_i = s_j = q),\ |K_{p,q}| \le c, \tag{3}$$

where $K_{p,q}$ is a set of vectors of the p-th type mapped into the q-th PU of the p-th type (q =1,2,… $q^p_{\max}$).

If SDF has the closed cycles, i.e. the cycles of iteration dependencies, then the sum of vectors $\boldsymbol{D}_j$, which belong to such cycles must be equal to zero. Among them the edges of iteration dependencies $\boldsymbol{D}_{Dj}$ are considered for which $w[e] > 0$. Such a vector is equal to $\boldsymbol{D}_{Dj} = (0,0,-wc)^T$, and it means the delay of a variable to w iterations. This means that

$$\sum b_{i,j}\ \boldsymbol{D}_j = (0,0,0)^T\ , \tag{4}$$

where $b_{i,j}$ – the element of i-th row of the cyclomatic matrix of SDF.

The searching for the optimum spatial SDF consists in finding of the optimum matrix K. Firstly, the coordinates of the vectors $\boldsymbol{D}_j$ are given, which provide the minimum value of T_C. Then the timing coordinates of vectors $\boldsymbol{K}_i$ are derived from the equation $K = D_O A_O^{-1}$, where D_O is the matrix of $\boldsymbol{D}_j$, A_O is the incidence matrix of the maximum spanning tree of SDF. The rest of coordinates of vectors $\boldsymbol{K}_i$ are searched taking into account the relations (1) – (4).

The method of finding the optimum spatial SDF is implemented in two steps. The initial SDF is resynchronized for minimization of the edges with $w[e] > 0$. At the first step the SDF nodes and edges are placed in the three dimensional space as a set of vectors $\boldsymbol{K}_i$ and $\boldsymbol{D}_j$ satisfying the conditions (1) – (4). This is the initial spatial SDF forming. By this process the PU number is minimized by keeping the condition $|K_{p,q}| \to c$. The strategies and algorithms of this process are given in [5] – [7]. The results of this step are both operator schedule and initial structure of the processor, which provide the precise estimation of both speed and hardware volume.

At the second step the SDF delay compensation is implemented. For this process the acyclic subgraph of SDF is considered, which has none edge $\boldsymbol{D}_{Dj}$. The delay compensation consists in placing the intermediate nodes of registers in the edges, represented by long vectors $\boldsymbol{D}_j$. In the resulting compensated SDF all the vectors-edges except $\boldsymbol{D}_{Dj}$ are equal to $\boldsymbol{D}_j = (a_j,b_j,1)^T$, or $\boldsymbol{D}_j = (a_j,b_j,0)^T$. The edges of the compensated SDF form the stages with the spacing between them of a single clock cycle. The swapping the nodes in a single stage of the compensated SDF provides the minimization both register number and multiplexor complexity of the resulting structure.

The optimized SDF can be described by the VHDL language as a pipelined datapath module. This description can be translated into the gate level description by any proper synthesizer. Therefore, there is not need to find the structure and the time table of the derived module [8].

4 Experimental results

To compare the different methods of SDF optimization was the task selected of the low pass IIR filter structure synthesis, which has the following frequency response function

$$H(z) = \frac{b_1 + a_1 z^{-1} + z^{-2}}{1 + a_1 z^{-1} + b_1 z^{-2}} + \frac{b_2 + a_2 z^{-1} + z^{-2}}{1 + a_2 z^{-1} + b_2 z^{-2}} z^{-1}.$$

The retimed and pipelined SDF of the filter with the response $H(z)$ is shown in the Fig.1. The dotted edges represent the vectors of iteration dependencies $\boldsymbol{D}_{Dj}$. The red edges form the critical path. The length of the critical path is equal to $T_C = 2d_A + d_M$, where d_A, and d_M are equal to adder delay and multiplier delay, respectively. This critical path is part of the loop-back cycle, in which the number of registers is equal to one. This number could not be increased by any retiming, and therefore, the critical path could not be less than this one. This means that the number of registers in the loop-backs determines the retiming effectiveness.

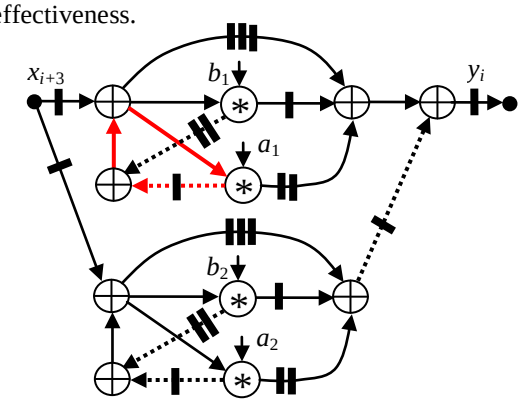

Fig. 1. Retimed SDF

The Fig.2 illustrates the respective spatial SDF with the factor $c = 2$ after the delay compensation step, and the Fig.3 the respective filter structure does. Due to this graph the clock cycle is equal to the minimum value $T_C = \max(d_A, d_M)$.

The SDF in the Fig.1, and SDFs, which were optimized by different methods, were described by VHDL, and configured in the Xilinx Kintex-7 FPGA using ISE tools. The results of these tasks are shown in the Table 1.

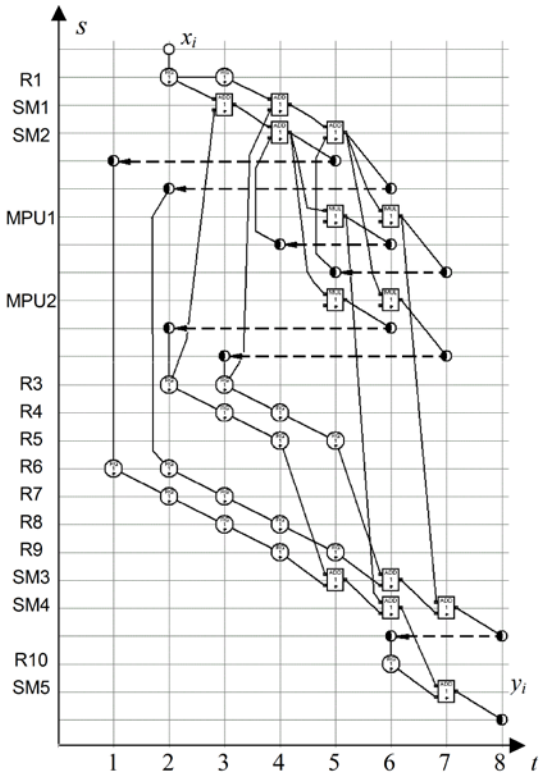

Fig. 2. Spatial SDF after delay compensation

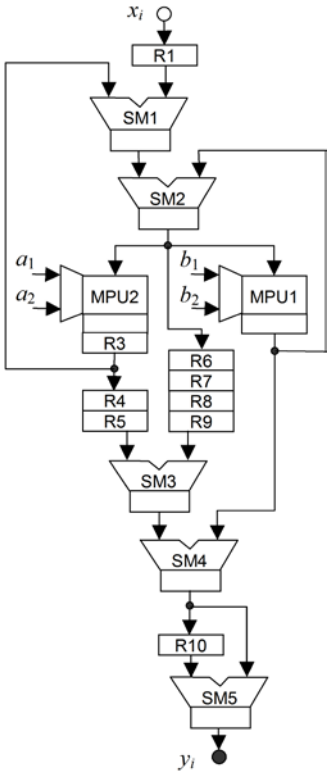

Fig. 3. Derived filter structure

Tab.1. Results of configuring the IIR filter projects

Method of optimization	S_L, LUT	S_M, MPU	T_C, ns	T_A, ns	Q_L, LUT/MHz	Q_M, MPU/MHz
Retiming (SDF in Fig.1)	212	4	5,78	5,78	1,23	0,023
Folding	184	2	5,43	10,86	1,99	0,022
Unfolding	510	14	6,94	3,47	1,77	0,049
Spatial SDF retiming	144	2	3,10	6,20	0,89	0,012

Here the hardware volume is measured in look-up tables (LUT) S_L, and in multiplier units (MPU) S_M. The effectiveness criteria are represented by the products $Q_L = S_L T_A$, and $Q_M = S_M T_A$. They mean the hardware volume, which is related to one megahertz of the sampling frequency. Therefore, the structure which has the less values of Q_L, and Q_M is prevalent.

It is obvious that the spatial SDF retiming is intended for the synthesis of structures with the slowing factor c. But comparing to the SDF folding method this method provides less hardware volume S_L, and less clock cycle T_C. In this testbench the method provides the best quality of the structural solution. We can compare the parameters of the folded, derived by this method, and of the structure after retiming. Then we find out that the speed of the folded structure is near the speed of the structure with the factor $c = 1$, but it has substantially less hardware volume than the last one. The parameters Q_L, and Q_M characterize the energy consumption efficiency as well. Therefore the processor designed by the proposed method consumes less energy.

5 Conclusion

The retiming methods are effectively used both in microelectronics and in programming for decades. Comparing the different methods of SDF retiming shows the high efficiency of the proposed method of spatial SDF retiming, which provides the minimum cost-performance ratio for the datapaths which implement the algorithms with feedbacks. This method is under implementation in the cloud-based CAD system in the Computer Engineering department of NTUU „KPI“.

References

[1] The Synthesis Approach to Digital System Design: P. Micheli, U.Lauther, P.Duzy – Ed-s. *Kluwer Academic Pub*. 415 p., 1992.

[2] Handbook of Algorithms for Physical Design Automation: C. J. Alpert, D. P. Mehta, S. S. Sapatnekar – Ed-s. *Auerbach Publ.* 1024 p., 2008.

[3] A. H. Shoab: Digital Design of Signal Processing Systems. A Practical Approach. *John Wiley & Sons*, 586 p., 2011.

[4] M. Potkonjak, j. M. Rabaey Maximally and Arbitrarily Fast Implementation of Linear and Feedback Linear Computations. *IEEE Trans. On Computer Aided Design of Integrated Circuits and Systems.* 19(1): 30-43, 2000.

[5] А. М. Сергиенко, В. П. Симоненко: Отображение периодических алгоритмов в программируемые логические интегральные схемы. *Электрон. Моделирование*. 29(2): 49–61, 2007.

[6] A. Sergiyenko, Ju. Kanevski, O. Maslennikov, R. Wyrzykowski: A Method for Mapping DSP Algorithms into Application Specific Structures: *24 th. EUROMICRO Conference (EUROMICRO'98).* 1: 10365-10371, 1998.

[7] А.М. Сергієнко: Досконалий кістяк графа алгоритму. *Вісник Національного технічного університету України. Сер. Інформатика і обчислювальна техніка.* 46: 62–67, 2007.

[8] А.М. Сергиенко: Методика проектирования цифровых фильтров с помощью VHDL. *Моделювання та інформаційні технології. Зб. наук. праць. Ін-т проблем моделювання в енергетиці ім. Г.Е. Пухова. НАНУ*. 12: 99–107, 2002.