A PROBLEM OF OPTIMIZING SPACE PARAMETERS IN SYSTOLIC ARRAY DESIGN

I. Milovanović, E. Milovanović, T. Tokić, I. Milentijević, N. Stojanović

Faculty of Electronic Engineering, University of Niš,
Beogradska 14, 18000 Niš, Serbia,
igor@elfak.ni.ac.yu

Abstract: This paper addresses the problem of optimizing space parameters of systolic arrays for one class of nested loop algorithms in advance, before the systolic array synthesis. These parameters are determined only according to a given projection direction and loop boundaries.

1. Introduction

VLSI technology has made possible the integration of circuits with hundreds of thousands of components into a single silicon chip. This high level of integration opens the way for massive parallel computations. Systolic processing constitutes a feasible solution for massive parallel computations. Its principles are compatible with VLSI technology characteristics. Since systolic arrays are highly regular, only algorithms with repetitive computations perform well on them. Algorithms with nested loops fall into this category.

A number of systolic arrays (SA) can be designed that implement a given nested loop algorithm. They can differ in several aspects including array topology, number of the processing elements (PE), execution time, geometric and chip area, number of I/O pins etc. Therefore it might be important to determine these features in advance, before the systolic array synthesis. This enables the designer to adapt the synthesis process to the predefined requirements.

The objective of this paper is to optimize the number of processing elements, geometric and chip area of the systolic array for a given problem size, that implements a three nested loop algorithm, before SA synthesis. By taking into account some properties of the systolic algorithm and with appropriate choice of a transformation matrix can minimize space parameters of the SA.

2. Background

Suppose, without deterioring the generality, that the computations in the algorithm are performed according to the following loop nest:

for k:= 1 to N_3 do

for j:= 1 to N_2 do

for i:= 1 to N_1 do

The computational structure of the algorithm A is characterized by the index space P_{int}

$$P_{int} = \{(i, j, k) \mid 1 \le i \le N_1, 1 \le j \le N_2, 1 \le k \le N_3\}. \quad (1)$$

where data are used or computed, and a dependency matrix which consists of a set of constant dependency vectors, each of them corresponding to one of the variables. Let $D = \left[\vec{e}_1^3 \vec{e}_2^3 \vec{e}_3^3\right]$ be a data dependency matrix of order 3x3, of a given algorithm and $\vec{\mu} = \left[\mu_1 \ \mu_2 \ \mu_3\right]^T$, $|\mu_1| + |\mu_2| + |\mu_3| > 0$ allowable projection direction. Since only planar arrays are considered, we have that $\mu_i \in \{-1, 0, 1\}$, $i = 1, 2, 3$ (see for example [1]). The transformation matrix T, that maps systolic algorithm into two-dimensional (2D) planar systolic array (SA) that implements an algorithm A has the following form

$$T = \begin{bmatrix} \vec{\Pi} \\ S \end{bmatrix} = \begin{bmatrix} \vec{\Pi} \\ \vec{S}_1 \\ \vec{S}_2 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}, \quad (2)$$

where $\vec{\Pi}$ determines time schedule and S is the space mapping function determining PE locations and communication channels between them.

Systolic array that implements a given systolic algorithm is obtained according to the following mapping [1]-[4]:

$$T : (P_{int}, D) \mapsto (P, \Delta) \quad (3)$$

where D is a dependency matrix of a given algorithm and

$$\Delta = \left[\Delta_t \ \Delta_s\right]^T, \quad \Delta_t = \vec{\Pi} \cdot D, \quad \Delta_s = S \cdot D,$$

$$P = \{[t \ x \ y]^T\}, \ t = \vec{\Pi}[i \ j \ k]^T,$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = S \begin{bmatrix} i \\ j \\ k \end{bmatrix} \quad (4)$$

for all $[i \ j \ k]^T \in P_{int}$. The x-y positions of the PEs in the SA are determined according to (4), while communication links between them are determined by Δ_s.

The geometric and chip are of the systolic arrays are defined as follows.

Definition 1 Geometric area, g_a, of a 2D systolic array is the area of the smallest convex polygon which bounds the PEs in the x-y plane.

Denote by T_{1i}, i = 1, 2, 3 the (1, i) - cofactor of matrix T. Geometric area of the SA can be determined according to the following equality

$$g_a = (N_1 - 1)(N_2 - 1)\,|\,T_{13}\,| + (N_1 - 1)(N_3 - 1)\,|\,T_{12}\,| + (N_2 - 1)(N_3 - 1)\,|\,T_{11}\,|\,. \quad (5)$$

Definition 2 The chip area, c_a, of the, 2D systolic array is obtained according to

$c_a = (\max\{x\} - \min\{x\} + 1)(\max\{y\} - \min\{y\} + 1)$,

where x and y are defined by (4).

In other words, chip area is the smallest rectangle that bounds the obtained SA in the x-y plane. In practice, the chip area is determined from

$$c_a = \left(1 + \sum_{j=1}^{3} |t_{2j}|\,(N_j - 1)\right)\left(1 + \sum_{j=1}^{3} |t_{3j}|\,(N_j - 1)\right). \quad (6)$$

The number of processing elements (PE) in the 2D SA that implements the algorithm A which is obtained for some allowable projection direction $\vec{\mu} = [\mu_1\ \mu_2\ \mu_3]^T$ and transformation T can be obtained according to

$$\Omega = \begin{cases} N_1 N_2 N_3, \text{ if } a_i > N_i, \text{ for some } 1 \le i \le 3 \\ N_1 N_2 N_3 - (N_1 - a_1)(N_2 - a_2)(N_3 - a_3), \text{ otherwise} \end{cases} \quad (7)$$

where

$$ai = \left| \frac{T_{1i}}{\gcd(T_{11}, T_{12}, T_{13})} \right|.$$

In the above, $T_{1i,}$ i = 1, 2, 3, is the (1, i)-cofactor of matrix T, while gcd(T_{11}, T_{12}, T_{13}) denotes the greatest common divisor of the nonzero integers T_{11}, T_{12} and T_{13}.

According to (5), (6) and (7) one can see that g_a, c_a and number of PEs, Ω, depend on the bounds of P_{int}, which we cannot affect, and a transformation matrix T. To minimize these parameters, it is obvious that we have to deal with transformation T. However, under certain conditions, i.e. for some classes of algorithms, it is possible to accommodate P_{int} to a given projection direction $\vec{\mu}$, which in turn can influence on space parameters of the SA. This is discussed in the next section.

3. Determining valid transformations

Although we are not discussing time parameters of the SA, we will give the conditions for determining time schedule. Time schedule $\vec{\Pi} = [t_{11}\ t_{12}\ t_{13}]$ of transformation T defined by (2) is determined according to the following vector product (see for example [11)

$$\vec{\Pi} = \pm\left[(\vec{e}_2^3)^T - (\vec{e}_1^3)^T\right] \times \left[(\vec{e}_3^3)^T - (\vec{e}_1^3)^T\right].$$

The sign + or - is determined such that one of the following inequalities is satisfied

$$\vec{\Pi}\vec{e}_i^3 > 0 \text{ or } \vec{\Pi}\vec{e}_i^3 < 0$$

for all i = 1, 2,3. In this way $\vec{\Pi}$ is uniquely defined regardless to the projection direction vector $\mu = [\mu_1\ \mu_2\ \mu_3]^T$.

Space transformation matrix S, of transformation T, is determined from the following conditions:

1° Matrix T must be nonsingular, i.e.

$$\det T \neq 0.$$

This condition ensures that mapping (3) is regular.

2° Projection direction $\vec{\mu} = [\mu_1\ \mu_2\ \mu_3]^T$ is orthogonal to the projection plane, i.e.

$$\vec{S}_1 \cdot \vec{\mu} = 0 \text{ and } \vec{S}_2 \cdot \vec{\mu} = 0 \qquad (8)$$

3° The connections between the PEs in the 2D array must be of near-neighbor type and crossing is not allowed. This requirement means that elements of matrix $\Delta_s = S \cdot D$ have to be from the set {-1, 0,1}.

4° To avoid conflicts in the SA, the following two conditions must not be satisfied at the same time

$$\vec{\Pi}\vec{p}_1 = \vec{\Pi}\vec{p}_2 \text{ and } S\vec{p}_1 = S\vec{p}_2,$$

where $\vec{p}_1 \neq \vec{p}_2$ and $\vec{p}_1, \vec{p}_2 \in P_{int}$.

It is common known fact that for a given projection direction $\mu = [\mu_1\ \mu_2\ \mu_3]^T$ there are several valid transformations T that map systolic algorithm into SA implementation. Since space parameters of the SA directly depend on the transformation T, the goal is to reduce a set of valid transformations to those that yield to optimal or near optimal solution with respect to a given space parameter. To minimize geometric and chip area, we substitute condition 2° with slightly stronger condition

5° $\vec{\mu} = (\vec{S}_1 \times \vec{S}_2)^T$;

This condition ensures that cofactors $|T_{11}|$, $|T_{12}|$ and $|T_{13}|$ of matrix T take values only from the set {0,11}, which may minimize geometric and chip area.

We also introduce the following two conditions for planar arrays:

6° If $\mu_1 = 1$, then the following equality must be satisfied

$$t_{22}t_{32} + t_{23}t_{33} = 0,$$

7° If $\mu_2 = \pm 1$, then

$$t_{21}t_{31} + t_{23}t_{33} = 0.$$

must be fulfilled.

Note that conditions 6° and 7° cannot both be satisfied at the same time. Namely, if the projection direction $\vec{\mu}$ has the form $\vec{\mu} = [1 \pm 1 \mu_3]^T$ and if $N_1 > N_2$, then the condition 6° has to be satisfied, otherwise the condition 7° has to be fulfilled. If $N_1 = N_2$ either of the conditions can be used. Note that conditions 1°, 3°, 4°, 5° and 6° or 7° still do not determine transformation T uniquely, but the set of valid transformations has been reduced to those that give SAs with minimal space parameters for a given problem size.

Besides reducing the set of valid transformations {T}, it is also important to observe if the systolic algorithm has some features that can help us to minimize space parameters. Namely, under certain conditions, it is possible to map a given algorithm into an equivalent one which is accommodated to a given projection direction $\vec{\mu}$. This is discussed in the next section.

4. Accommodation of the systolic algorithm

Under certain conditions the number of PEs, and consequently geometric and chip area, of the SA can be reduced if the systolic algorithm can be accommodated to a given projection direction. The transformation of algorithm A is performed by accommodation of inner computation space P_{int} to the projection direction $\vec{\mu}$.. Then the accommodated index space $\overline{P}_{int}$ is mapped by transformation matrix T into systolic array. Essentially, we substitute mapping (3) by the following two mappings

$$H : P_{int} \rightarrow \overline{P}_{int} \text{ and } T : (\overline{P}_{int}, D) \rightarrow (P, \Delta), \qquad (9)$$

where mapping H performs the accommodation of P_{int} to the projection direction.

The question is when this accommodation is possible and how does H look like. The answer is given in the text that follows.

Let k be an iterative index variable in systolic algorithm A. Under this condition we involve the following definitions.

Definition 3 If for some fixed j the ordering of computations in algorithm A, may be performed over arbitrary permutations of index variables i and k, we say that A is an A(i, k) adaptable.

Definition 4 If for some fixed i the ordering of computations in algorithm A, can be performed over arbitrary permutations of index variables j and k, we say that A is an A(j, k) adaptable.

Remark 1 If a given algorithm A satisfies both Definition 3 and 4, we say that A is adaptable.

The transformation H for the above defined classes of algorithms is defined as follows.

Definition 5 Suppose that a given algorithm is of type A(j, k). If $\bar{\mu} = [1\ \mu_2\ \mu_3]^T$ is an allowable projection direction, the mapping H = (F, G) that performs accommodation of algorithm A to a given projection direction, is defined by

$$F = \begin{bmatrix} 1 & 0 & 0 \\ \mu_2 & 1 & 0 \\ \mu_3 & 0 & 1 \end{bmatrix}, \qquad G = \begin{bmatrix} 0 \\ g_2 \\ g_3 \end{bmatrix},$$

where g_2 and g_3 are the smallest integers determined such that for each $[i\ j\ k]^T \in P_{int}$, the following is valid

$$\mu_2 i + j + g_2 > 0, \text{ and } \mu_3 i + k + g_3 > 0.$$

The elements of Pint are obtained according to

$$[u\ v\ w]^T = F[i\ j\ k]^T + G. \tag{10}$$

Definition 6 Suppose that a given algorithm is of type A(i, k). If $\vec{\mu} = [\mu_1 \pm 1\mu_3]^T$ is an allowable projection direction, mapping H = (F, G) is defined by

$$F = \begin{bmatrix} 1 & \pm\mu_1 & 0 \\ 0 & 1 & 0 \\ 0 & \pm\mu_3 & 1 \end{bmatrix}, \qquad G = \begin{bmatrix} g_1 \\ 0 \\ g_3 \end{bmatrix},$$

where g_l and g_3 are the smallest integers determined such that for each $[i\ j\ k]^T \in P_{int}$ the following is valid

$$i \pm \mu_1 j + g_1 > 0 \quad \text{and} \quad k \pm \mu_3 j + g_3 > 0.$$

The elements of P_{int} are obtained according to (10)

The subject discussed in this section is explained on the example of matrix multiplication algorithm.

Matrix multiplication

Let $A = (a_{ik})_{N_1 \times N_3}$ and $B = (b_{kj})_{N_3 \times N_2}$ be a rectangular matrices. The systolic algorithm that computes $C = A \cdot B$ has the following form

Algorithm A

for k:= 1 to N_3 do

for j:= 1 to N_2 do

for i:= 1 to N_l do

a(i, j, k) := a(i, j - 1, k);

b(i, j, k) := b(i-1, j, k);

c(i, j, k) := c(i, j, k - 1) + a(i, j, k) * b(i, j, k);

It is not difficult to see that this algorithm is adaptable in the sense of Remark 1. In this example we take a direction $\mu = [1\ 1\ 1]^T$ which give the well known hexagonal systolic array (see for example [5]-[10]).

If we choose the following transformation matrix

$$T^{(1)} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 2 & -1 & -1 \end{bmatrix},$$

we conclude that $T^{(l)}$ satisfies the conditions 1°, 2°, 3° and 4°, but not the conditions 5°, 6° and 7°. According to (5), (6) and (7) we determine that the obtained hexagonal SA has the following features

$$\Omega = N_1N_2+N_1N_3+N_2N_3-N_1-N_2-N_3+1$$
$$g_a = 2[N_lN_2+N_1N_3+N_2N_3-2(N_l+N_2+N_3)+3] \qquad (11)$$
$$ca = (N2+N3-1)(2N1+N2+N3-3).$$

On the other hand, if we choose the following transformation matrix

$$T^{(2)} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix},$$

we conclude now that $T^{(2)}$ satisfies conditions 1°, 3°, 4°, 5° and 6°. Since the matrix multiplication algorithm is adaptable, we can apply the mapping (9). For matrix H = (F, G) we choose

$$F = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \text{ and } g = \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix}.$$

Now, the array obtained by mapping (9) has the following space parameters

$$\Omega = N_2 N_3$$
$$g_a = (N_2 - 1)(N_3 - 1)$$
$$c_a = N_2 N_3.$$

If we choose H = (F, G) as

$$F = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \text{ and } g = \begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix}, .$$

we obtain hexagonal SA with the following space parameters

$$\Omega = N_1 N_3$$
$$g_a = (N_1 - 1)(N_3 - 1)$$
$$c_a = N_1 N_3.$$

According to the previous, we conclude that with an appropriate choice of H and T we can synthesize hexagonal array which has the following space parameters

$$\Omega = N_3 \min\{N_1, N_2\}$$
$$g_a=(N_3 -1) \min\{N_1-1, N_2-1\} \quad (12)$$
$$c_a = N_3 \min\{N_1, N_2\}.$$

By comparing (11) and (12) it is not difficult to conclude that substantial minimization of the space parameters of the hexagonal systolic array that implements matrix multiplication algorithm has been achieved.

5. Conclusion

We have discussed a problem of optimizing space parameters of 2D systolic arrays in this paper. The minimization is achieved by introducing some stronger conditions for the transformation matrix that maps systolic algorithm into systolic array implementation and by accommodation of given algorithm to the projection direction vector. The described procedure has been explained on the example of matrix multiplication algorithm.

6. References

1. S. G. Sedukhin, The *designing and analysis of systolic algorithms and structures*, Programming, 2(1990), 20-40. (In Russian).
2. M. Chen, *A design methodology for synthesizing parallel algorithms and architectures*, J. Parallel Distributed Comput., (1986), 461-491.

3. C. Langauer, *A view of systolic design*, In: Proc. International Conference on Parallel Compution Technologies, (N. N. Mirenkov, ed), Novosibirsk'91, World Scientific, Singapore, 1991, 32-46.
4. D. I. Moldovan, *On the design of algorithms for VLSI systolic arrays*, IEE Proc., 71 (1983), 113-120.
5. M. O. Esonu, J. Al-Khalili, S. Haxiri, D. Al-Khalili, *Systolic arrays: How to chose them*, IEE Proc. Vo1139, 3 (1992), 179-188.
6. C. N. Zhang, J. H. Weston, Y.-F. Yan, *Determining object functions in systolic array designs*, IEEE Trans. Very Large Scale Integration (VLSI) Systems, Vol. 2, 3 (1994), 357-360.
7. T. I. Tokic, I. Z. Milovanovic, D. M. Randjelovic, E. I. Milovanovic, *Determining VLSI array size for one class of nested loop algorithms*, In: Advances in Computer and Information Sciences (U. Gudukbay, T. Dagax, A. Giirsay, E. Gelembe, eds.), IDS Press, 1998, 389-396.
8. H. T. Kung, C. E. Leiserson, *Systolic arrays (for VLSI*), Tech. Rep. CS-79-103, Carnegie Mellon University, Pitsburgh, PA, 1978.
9. I. Z. Milentijevic, I. Z. Milovanovic, E. I. Milovanovic, M. K. Stojcev, *The design of optimal planar systolic arrays for matrix multiplication*, Comput. Math. Applic., Vol. 33, 6 (1997), 17-35.
10. M. Gušev, D. J. Evans, *Nonplanar transformations of the matrix multiplication algorithm*, Inter. J. Comput. Math., Vol. 45 (1992), 1-21.