Some Space Considerations of Space-Time Mappings into Systolic Arrays

J.H. Weston, C.N. Zhang, Y.-F. Yan

Abstract: In this paper the space-time mapping of the dependency matrix of an algorithm is used to study spatial properties of a systolic array implementation of a 3-nested loop structure. Elementary expressions are developed for both the number of processing elements and the area of the array. These expressions involve only the space-time transformation and the lengths of the loops. As well, characterizations have been found for the form of the space-time transformation which produces a systolic array with the minimum number of processing elements, and one which has both the minimum number of processing elements and the smallest area.

1 Introduction

The mapping of algorithms, structured as nested loops, into systolic arrays has been the focus of considerable research since the introduction of systolic arrays in 1979 [2, 3, 4, 6, 8]. Many of the results reported are concerned with both the possibility of implementing such an algorithm in a systolic array, and optimization of the implementation with respect to a variety of criteria. The objective functions usually involve either time or space considerations. (A partial list is contained in reference [1].) The current paper examines two measures of space in finding an optimal systolic array implementation of an algorithm, the number of processing elements (PEs) and the area of the array. Since the number of PEs and the area are associated with both fabrication costs and communication time they are important parameters to be considered in choosing an implementation. Further if a particular algorithm is to be implemented in a fixed size systolic array then the number of

For algorithms which can be written in the form of nested loops, the description of an implementation in a systolic array can be cast in a mathematical framework involving a space-time transformation which maps the algorithm to a systolic array [3, 6]. This space-time transformation is used, in section 2, to find an explicit expressions for the number of PEs and, in section 3, the area required. Also a necessary and sufficient condition is derived for the form of a transformation which produces a systolic array consisting of the minimum possible number of PEs as well as occupying the smallest area.

A p-nested loop structure with constant data dependence vectors can be represented by a pair (D, C_D), in which D is the data dependency matrix [4, 5, 6], and $C_D = \{(i_1, i_2, \cdots i_p)^t : 1 \leq i_1 \leq l_1, \cdots, 1 \leq i_p \leq l_p\}$ is the index space (here t designates transpose). A systolic array implementation of such an algorithm may be obtained by a linear transformation ($p \times p$ matrix) $T = \begin{pmatrix} \pi \\ S \end{pmatrix}$ where π is a $1 \times p$ vector determining time scheduling, and S is a $(p-1) \times p$ matrix, the space transformation, which maps C_D onto an (p-1) - dimensional systolic array. T is called the space-time transformation. Let $C_S = TC_D$ and $C_S^{'} = SC_D$, then when $p = 3$ the columns of $C_S^{'}$ are the 2-space coordinates of the PEs in the systolic array implementation. To ensure a one to one mapping and causal time scheduling, T must be nonsingular and all the elements of the first row of $\Delta = TD$ negative (or positive depending on the convention). The latter two rows of Δ indicate the inter processor communications [5, 6]. The requirement of nearest neighbor connection is precisely that these latter two rows of Δ contain only 0, 1, or -1.

An example which has often been used to illustrate this approach is matrix multiplication. The algorithm, in normal form (without broadcast variables) is as follows:

Algorithm 1 (Matrix multiplication $C = A \times B$)

```
    for $i_1 := 1$ to $l_1$ do
    for $i_2 := 1$ to $l_2$ do
    for $i_3 := 1$ to $l_3$ do
        begin
            $a(i_1, i_2, i_3) := a(i_1, i_2 - 1, i_3);$
            $b(i_1, i_2, i_3) := b(i_1 - 1, i_2, i_3);$
            $c(i_1, i_2, i_3) := c(i_1, i_2, i_3 - 1) + a(i_1, i_2, i_3)b(i_1, i_2, i_3);$
        end;
```

where $a(i_1, 0, i_3) = a_{i_1,i_3}$, $b(0, i_2, i_3) = b_{i_3,i_2}$, $c(i_1, i_2, l_3) = c_{i_1,i_2}$ for all $i_1, i_2,$ and i_3.

For this algorithm, $D = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$ and $C_D = \{(1,1,1)^t, (1,1,2)^t, \cdots, (l_1, l_2, l_3)^t\}$.

In this example there are many valid space-time transformations, T, even if the systolic array is required to have only nearest neighbor connections. Three such valid transformations are $T_1 = \begin{pmatrix} \pi_1 \\ S_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \end{pmatrix}$, $T_2 = \begin{pmatrix} \pi_2 \\ S_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 0 & -1 & 1 \end{pmatrix}$, and $T_3 =$

$\begin{pmatrix} \pi_3 \\ S_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$. Here

$$\Delta_1 = T_1 D = \begin{pmatrix} -1 & -1 & -1 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix}, \Delta_2 = T_2 D = \begin{pmatrix} -1 & -1 & -1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix},$$

and $\Delta_3 = T_3 D = \begin{pmatrix} -1 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$. If $l_1 = l_2 = l_3 = 4$ then the resulting systolic arrays are shown in Figs. 1, 2 and 3.

In the following, the assumption will be made that p = 3 although the results in section 2 can be reworded to remain true for an arbitrary positive integer p.

2 Determining the number of processing elements

The space-time transformation gives a natural way to map a p-level nested loop algorithm to an (p-1)-dimensional systolic array. In algorithm 1 when $l_1 = l_2 = l_3 = 4$ the three transformations T_1, T_2, and T_3 give systolic arrays with 28, 28, and 16 PEs respectively, (Figs. 1, 2, and 3). The usual approach to determining the number of PEs arising from a particular T is to transform C_D by T and count the number of points in the image [1, 5, 6]. If the lengths of the loops are large this may take considerable time. Also, using this method to find a transformation with the minimum number of PEs, it is necessary to first determine the number of PEs required for each valid space-time transformation. Lemma 1 and Theorem 1 give an expression for determining the number of PEs directly from the transformation T and the lengths of the loops.

Notation If $T = (t_{i,j})$ is a 3×3 integer matrix let $T_{i,j}$ be the (i,j) cofactor of T, and for each $j = 1, 2, 3$ let $a_j = \frac{T_{1,j}}{gcd(|T_{1,1}|,|T_{1,2}|,|T_{1,3}|)}$. The expression $T(D, C_D) = (\Delta, C_S)$ will be used to indicate that T is a valid space-time transformation which maps the algorithm represented by (D, C_D) to the systolic array represented by (Δ, C_S).

Lemma 1 *Let $T(D, C_D) = (\Delta, C_S)$, $T = \begin{pmatrix} \pi \\ \\ S \end{pmatrix}$, $(i_1, i_2, i_3) \epsilon C_D$ and $\Delta i_1, \Delta i_2, \Delta i_3$ be integers, then $S(i_1, i_2, i_3)^t = S(i_1 + \Delta i_1, i_2 + \Delta i_2, i_3 + \Delta i_3)^t$ if and only if there is an integer K so that $(\Delta i_1, \Delta i_2, \Delta i_3) = K(a_1, a_2, a_3)$.*

Proof: Since S is a linear function $S(i_1, i_2, i_3)^t = S(i_1 + \Delta i_1, i_2 + \Delta i_2, i_3 + \Delta i_3)^t$ if and only if $S(\Delta i_1, \Delta i_2, \Delta i_3)^t = (0, 0)^t$. T is non-singular hence there is at least one j so that $T_{1,j} \neq 0$, suppose $T_{1,1} \neq 0$. Solving $S(\Delta i_1, \Delta i_2, \Delta i_3)^t = (0, 0)^t$ for Δi_2 and Δi_3 gives

$$\Delta i_2 = \frac{T_{1,2}}{T_{1,1}} \Delta i_1 \text{ and } \Delta i_3 = \frac{T_{1,3}}{T_{1,1}} \Delta i_1. \tag{1}$$

Since Δi_2 and Δi_3 are integers there are integers K_1 and K_2 so that

$$\Delta i_1 = K_1 \frac{T_{1,1}}{gcd(|T_{1,1}|, |T_{1,2}|)} = K_2 \frac{T_{1,1}}{gcd(|T_{1,1}|, |T_{1,3}|)}.$$

Thus there is an integer K so that

$$\Delta i_1 = K \frac{T_{1,1}}{gcd(|T_{1,1}|, |T_{1,2}|, |T_{1,3}|)} = K a_1.$$

Hence (1) gives $\Delta i_2 = K a_2$ and $\Delta i_3 = K a_3$.

A similar argument applies if $T_{1,2} \neq 0$ or $T_{1,3} \neq 0$.

Based on this description of the points in the index space C_D which collapse, under S, to the same point in $C_S^{'}$, the number of PEs can be determined by the following theorem.

Theorem 1 *If $T(D, C_D) = (\Delta, C_S)$ then the number of PEs is*

$l_1 l_2 l_3$ if $|a_j| \geq l_j$ for some $j = 1, 2, 3,$ and $l_1 l_2 l_3 - (l_1 - |a_1|)(l_2 - |a_2|)(l_3 - |a_3|)$ otherwise .

Proof: If $(i_1, i_2, i_3)^t \epsilon\, C_D$ then call $(i_1 + a_1, i_2 + a_2, i_3 + a_3)^t$ its *redundant sequel* if $(i_1 + a_1, i_2 + a_2, i_3 + a_3)^t \epsilon\, C_D$. Let (i_1^0, i_2^0, i_3^0) be defined by

$$i_j^0 = \begin{cases} 1 & a_j \geq 0 \\ l_j & a_j < 0 \end{cases}$$

Clearly if $(i_1^0, i_2^0, i_3^0)^t$ has no redundant sequel then no point in C_D has one. If $|a_j| \geq l_j$ for some $j = 1, 2, 3$ then $(i_1^0, i_2^0, i_3^0)^t$ has no redundant sequel and each of the $l_1 l_2 l_3$ points in C_D has a distinct image in $C_S^{'}$, thus the number of PEs is $l_1 l_2 l_3$.

If $|a_j| < l_j$ for each $j = 1, 2, 3$ then $(i_1^0 + a_1,\ i_2^0 + a_2,\ i_3^0 + a_3)^t$ is a redundant sequel. In this case $(i_1^0, i_2^0, r)^t$ has a redundant sequel if and only if $1 \leq r + a_3 \leq l_3$. In either case $a_3 \geq 0$ or $a_3 < 0$, there are $l_3 - |a_3|$ choices for r. For each such choice r_0 for r, $(i_1^0, q, r_0)^t$ has a redundant sequel if and only if $1 \leq q + a_2 \leq l_2$. As before there are $l_2 - |a_2|$ such choices for q. Now for each such choice r_0 and q_0 there are $l_1 - |a1|$ choices for p so that $(p, q_0, r_0)^t$ is a redundant sequel. Hence there are $(l_1 - |a_1|)(l_2 - |a_2|)(l_3 - |a_3|)$ redundant sequels in total. Removing these redundant sequels from C_D leaves $l_1 l_2 l_3 - (l_1 - |a_1|)(l_2 - |a_2|)(l_3 - |a_3|)$ points, each of which maps, under S, to a distinct PE in $C_S^{'}$.

In algorithm 1, for example, if $T = T_1 = \begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \end{pmatrix}$ and $l_1 = l_2 = l_3 =$

4, then $T_{1,1} = 0, T_{1,2} = 2, T_{1,3} = 2$ so $a_1 = 0, a_2 = 1, a_3 = 1$ and the number of PEs is 28, Fig. 1. In this situation the minimum number of PEs possible is 16, and T_3 is a transformation which achieves this minimum, Fig. 3. The following corollary to theorem 1 characterizes the form of a space-time transformation which results in a systolic array with the minimum number of PEs.

Corollary 1 *If* $T(D, C_D) = (\Delta, C_S)$ *and* $0 \leq l_{j_2}, l_{j_3} \leq l_{j_1}$ *then the minimum number of PEs is* $l_{j_2} l_{j_3}$ *and is realized by a space-time transformation* $T = \begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,1} & t_{3,2} & t_{3,3} \end{pmatrix}$ *if and only if* $t_{2,j_1} = t_{3,j_1} = 0$.

Proof: Since T is nonsingular there is a j_1 with $1 \leq j_1 \leq 3$ and $a_{j_1} \neq 0$. Thus $|a_{j_1}| \geq 1$ and $l_{j_1} - |a_{j_1}| \leq l_{j_1} - 1$, hence $(l_{j_1} - 1) l_{j_2} l_{j_3} \geq (l_{j_1} - |a_{j_1}|)(l_{j_2} - |a_{j_2}|)(l_{j_3} - |a_{j_3}|)$ where j_1, j_2, j_3 is $1, 2, 3$ in some order. The largest of these bounds is obtained when $l_{j_2}, l_{j_3} \leq l_{j_1}$ and $a_{j_1} = 1, a_{j_2} = a_{j_3} = 0$, i.e. $, T_{1,j_2} = T_{1,j_3} = 0$. But $T_{1,j_2} = 0$ if and only if the j_1 and j_3 columns of S are linearly independent, and $T_{1,j_3} = 0$ if and only if the j_1 and j_2 columns of S are linearly independent. Thus if the j_1 column of S is not $(0,0)^t$ then the j_2 and j_3 columns of S are also linearly independent and the determinant of T is 0. Hence the systolic array has the minimum possible number of PEs, $l_{j_2} l_{j_3}$, if and only if the j_1 column of S is $(0,0)^t$, i.e. $t_{2,j_1} = t_{3,j_1} = 0$.

3 Determining the area of the systolic array

The area of the array is another parameter which is associated with fabrication costs and lengths of data paths. In [1] the area of a systolic array refers to the ”Silicon area measure”. Here the area is defined as follows.

Definition 1 *The* bounding polygon *of a two-dimensional systolic array is the smallest convex polygon in the plane which contains the array. The* area *of a two-dimensional systolic array is the area of its bounding polygon.*

Figs. 4 and 5 show the bounding polygons, here parallelograms, for the systolic arrays given by T_1 and T_2 in algorithm 1, and indicate that two systolic array implementations of the same algorithm, with the same number of PEs may have different areas. Since the area of a parallelogram with vertices $(a_1, a_2), (b_1, b_2), (c_1, c_2),$ and (d_1, d_2) is the absolute value of the determinant $\begin{vmatrix} b_1 - a_1 & b_2 - a_2 \\ c_1 - a_1 & c_2 - a_2 \end{vmatrix}$ [7], the area of the bounding parallelogram required for T_1 is 36 (Fig. 4) and for T_2 is 18 (Fig. 5).

It is useful to determine which polygons can be bounding polygons for systolic arrays. Since the indexing set C_D is contained in a rectangular region in three dimensions, Fig. 6, and the space-time transformation is linear and non-singular, the systolic array is bounded by either a parallelogram, or a polygon constructed from six lines which are pairwise parallel and of equal length. For example, for algorithm 1, $T_4 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix}$ results in a systolic array with bounding polygon shown in Fig. 7. The area of such a bounding polygon is found using the following lemma.

Lemma 2 *If ABCDEF is a convex polygon bounded by six lines which are pairwise parallel and of the same length (see Fig 8), then the area of ABCDEF is twice the sum of the areas of the triangles ABF, BCD and DEF.*

Proof: The proof is an argument in elementary geometry. Construct a line through D parallel to BC and a line through B parallel to CD, and let P be their point of intersection.

Join P and F (see Fig 9). BCDP is then a parallelogram and hence CD, PB and AF are parallel and have the same length. Therefore ABPF, and similarly FPDE are parallelograms and the area of ABCDEF is the sum of the areas of these three parallelograms.

To apply this lemma suppose that, in Fig. 7, A, B and F are the images of $(1,1,1)^t$, $(l_1,1,1)^t$ and $(1,l_2,1)^t$ under $S = \begin{pmatrix} t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,1} & t_{3,2} & t_{3,3} \end{pmatrix}$ then

$S(1,1,1)^t - S(l_1,1,1)^t = (t_{2,1}(l_1-1), t_{3,1}(l_1-1))$ and

$S(1,1,1)^t - S(1,l_2,1)^t = (t_{2,2}(l_2-1), t_{3,2}(t_2-1))$. Thus the area of the parallelogram ABPF is

$$\left| \det \begin{pmatrix} t_{2,1}(l_1-1) & t_{3,1}(l_1-1) \\ t_{2,2}(l_2-1) & t_{3,2}(l_2-1) \end{pmatrix} \right| = (l_1-1)(l_2-1)|T_{1,3}|. \qquad (2)$$

The analogous results are true for the parallelograms CDPB and EFPD.

Theorem 2 *If $T(D,C_D) = (\Delta, C_S)$ then the area of the polygon bounding the systolic array is* $[(l_1-1)(l_2-1)|a_1| + (l_1-1)(l_3-1)|a_2| + (l_2-1)(l_3-1)|a_3|]gcd(|T_{1,1}|, |T_{1,2}|, |T_{1,3}|)$

$= (l_1-1)(l_2-1)|T_{1,1}| + (l_1-1)(l_3-1)|T_{1,2}| + (l_2-1)(l_3-1)|T_{1,3}|$

Proof: If the polygon is of the form of Fig. 8 then the result follows directly from lemma 2 and (2) above. If it is a parallelogram then either one or two of the cofactors is zero and expression (2) remains valid.

Referring again to algorithm 1 with $l_1 = l_2 = l_3 = 4$ the area of the bounding polygon resulting from T_4 is $3 \times 3 \times |1| + 3 \times 3 \times |-1| + 3 \times 3 \times |1| = 27$.

Corollary 2 *If $T(D,C_D) = (\Delta, C_S)$ and $l_{i_2}, l_{i_3} \leq l_{i_1}$, then this systolic implementation of (D,C_D) requires the minimum possible number of PEs as well as the smallest area if and* *only if* $t_{2,j_1} = t_{3,j_1} = 0$ *and* $T_{1,j_1} = \pm 1$.

4 Conclusions

In this paper we have used the space-time mapping T of the dependency matrix of an algorithm to study spatial properties of a systolic array implementation of a 3-nested loop structure. The 3×3 matrix T has been used to develop elementary expressions for both the number of PEs and the area of the systolic array, expressions which involve only T and the lengths of the loops and do not require evaluation of TC_D. As well characterizations have been developed for the form of T which produces a systolic array with the minimum number of PEs, and one which has the minimum number of PEs and the smallest area. These results have been presented for 2-dimensional systolic arrays, but may be easily be extended to arrays of arbitrary dimensions.

Implementing an algorithm in a systolic array can be accomplished by converting the dependency structure of the algorithm to a form which can be imbedded in the array. In this paper the conversion function is constrained to be a linear transformation, the space-time mapping, and this linearity is used in an essential way. It may be possible to use the strength of the linearity of this function to optimize various other criteria which are important in choosing a systolic array implementation of an algorithm.

References

[1] Esonu, M.O., Al-Khalili, A.J., Hariri, S., and AlKhalili, D.: 'Systolic arrays: how to choose them', *IEE Proceedings-E,* 1992, **139,** pp. 179-188

[2] Kung, H.T., and Leiserson, C.E.: 'Systolic Arrays for VLSI', Sparse Matrix Proc., 1978-79 (Academic Press, Orlando, FL, 1979), pp. 256-282

[3] Miranker, W.L., and Winkler, A.: 'Spacetime representations of computational structures', *J. Computing,* 1984, **32,** pp. 93-114

[4] Moldovan, D.I.: 'On the design of algorithms for VLSI systolic arrays', *Proc. IEEE,* 1983, **71,** pp. 113-120.

[5] Moldovan, D.I., and Fortes, J.A.B.: 'Partitioning and mapping algorithms into fixed size systolic arrays', *IEEE Trans. Comput.,* 1986, **c-35,** pp. 1-12

[6] Moldovan, D.I.: 'ADVIS: A software package for the design of systolic arrays', *IEEE Trans. Comput.,* 1987, **CAD-6,** pp. 33-40

[7] Strang, Gilbert: 'Linear Algebra and Its Applications', (Academic Press, New York, 1976)

[8] Wong, Y, and Delosme, J.-M.,'Optimization of Computation Time for Systolic Arrays',*IEEE Trans. Comput.*1992, **41,** pp. 159-177