V4.13

ON UNIFORM ONE-CHIP VLSI DESIGN CONSIDERATIONS FOR SOME DESCRETE ORTHOGONAL TRANSFORMS

KuoJuey R. Liu and K. Yao
Electrical Engineering Department
University of California, Los Angeles
Los Angeles, CA 90024-1594

Abstract - One-chip VLSI design consideration for area*time^2 optimal FFT shuffle-exchange architecture is considered and a systolic-network architecture for the computation of the FFT is presented. This architecture has the same asymptotically optimal theoretical $O(N^2\log^2N)$ area*time^2 complexity as the FFT shuffle-exchange architecture, but is more suitable for one-chip VLSI design. In this paper, architectures which are feasible for an one-chip FFT design, as well as for shuffle-exchange type fast discrete orthogonal transforms such as Generalized transform, Cosine transform and Slant transform are also discussed.

I. Introduction

Most signal processing techniques involve intensive arithmetic computations. The low-cost, high-density, fast VLSI devices can satisfy the ever-increasing demands of speed and performance in modern signal processing and make super-computing practrical. The traditional criterion of component count is no longer adequate to establish a scale of comparison among various solutions of a given problem. The number-of-elements criterion is substantially based on the fact that processing elements and their interconnections are realized by different media. This difference disappears in VLSI since all the elements are layouted with the same design and process technologies on the surface of silicon chip.

A great deal of work has been performed in recent years to establish bounds on the cost of VLSI structure [9]. For any given problem, it is of great interest to explore the tradeoffs between the area and time of a dedicated circuit developed to solve that problem. In general, two key parameters are used in evaluating such a cost: the time taken by the structure to solve a single problem; and the area occupied on the silicon chip. Limits on area-time performance, area*time^2, have been proved for a number of important problems, including sorting, matrix multiplication, decoding, binary multiplication, and fast Fourier transformation [1,3,6,7]. The fact that there is a theoretical limit to area*time^2 performance suggest that designs be evaluated in term of how closely they approach the limit. Even thougth such asymptolic analysis can give insight into performance evaluation and design; however, today's VLSI technologies still cannot support the one-chip design for a large number of sample points. The small constant factors that do not appeal in the asymptolic order are important when the sample points are small. Therefore, the asymptolic complexity cannot really reflect completely the performance, especially the area factor, for small sample points one-chip design.

This work was partially supported by the UC MICRO program.

The VLSI implementation of fast Fourier transiorm (FFT) has received considerable attention recently. Part of the reasons are that the FFT is the most well-known transform and is more frequently used in signal processing than any other transforms. Several architectures have been proposed and designed on VLSI chip [1,3,7]. For the N-element Fourier transform, it has been shown that no circuit can have a better area*time^2 performance than $O(N^2\log^2N)$ [1,9]. Nine designs were also compared and discussed in [1]. Some of the designs are very fast but occupy more area than that of the slower ones; some area-efficient designs are slower than those that need more area. The product of area and the square of the time needed to perform the FFT cannot be smaller than $O(N^2\log^2N)$. In [1], Thompson showed that, by using full parallel processing, the shuffle- exchange architecture (or FFT network) is the fastest one with time complexity O(logN) among the optimal structures and is relativeiy simple intuitively. Since it is an area*time^2 optimal structure with the smallest time complexity, the area complexity is, of course, the largest one with an area complexity $O(N^2)$. That is, it is a very area consuming structure. As in Fig. 1a, shuffle-exchange architecture has regular data flow and simple communication schemes (although not locally) which is very suitable for fast transforms. In the following sections, a systolic-network architecture for FFT which has the same optimal theoretical $O(N^2\log^2N)$ area*time^2 complexity as shuffle-exchange architecture asymptotically, but with smaller constant factor is presented. The generalization of this FFT systolic-network architecture to some shuffle-exchange type fast discrete orthogonal transforms such as Generalized transform, Cosine transform, and Slant transform are also discussed.

II. The Systolic-Network Architecture

A. System Description

In Fig. 1b the basic multiply-add cell of a shuffle-exchange FFT is shown. Each cell has three bit-series inputs w^k, x_0, and x_1. It produces two bit-series outputs

$$y_0 = x_0 + w^k x_1, \qquad y_1 = x_0 - w^k x_1 \qquad (1)$$

Such multiply-add cells perform the binary addition and multiplication functions. For an eight-point, N=8, FFT as in Fig. 1a, it requires twelve multiply-add cells. It is well known that the area needed for multiplication is much more than that of addition. For today's technology only about three or four eight by eight bits multipliers can be built into one 64-pin chip. The reduction of the multiplication is necessary if we wish to implement the FFT in one-chip with larger number of samples N.

CH2561-9/88/0000-2136 $1.00 © 1988 IEEE

A four-point FFT need no multiplication since the twiddle factors are 1, -1, i, and -i only. Based on that fact, the four-point

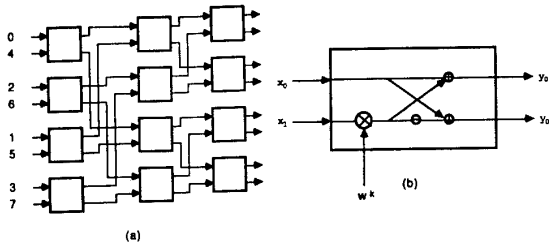

Fig. 1 The shuffle-exchange architecture for FFT and it's basic cell.

FFT can be built in a systolic complex number adder as shown in Fig. 2a. The basic function cell is shown in Fig. 2b. For a radix 2 eight-point FFT systolic-network architecture, we combine the multiply-add cell which is used in a shuffle-exchange FFT with the four-point systolic complex number adder as shown in Fig. 3a. There are three kinds of basic cells for systolic-network architecture.

- **the leaf cell** is a complex number adder which communicates with external data inputs and sum with the previous cell output and the twiddle factors sent from buffer memory.
- **the node cell** which is a multiply-add cell which performs equation (1) and the outputs are the FFT result.
- **the buffer memory** which is composed of shift register.

The input data are loaded in a skewed fashion. At each clock period, the data from the external and buffer memory are sent to the leaf cell and then the outputs of each leaf cells are pipelined from left to right. The outputs, which come out in order of X(0), X(4); X(1), X(5); X(2), X(6); X(3), X(7), from the node cell are the transformed result as we expected. The total area required is much less than that of shuffle-exchange architecture because of the reduction of multiplication. If the cells of the first two columns of the shuffle-exchange architecture are replaced with complex number adder, it still need four multiply-add cells to perform the multiplication. Because of the area limitation of the multiplication, only a eight-point FFT can be bulit in one chip using the shuffle-exchange architecture; even though it is optimal in the sense that the area*time^2 achieves the lowest bound. On the other hand, a 16-point FFT can be built by using the systolic-network architecture since only four multipliers are required. A 16-point FFT systolic-network is shown in Fig. 3b.

Since the number of leaf cells is $\log_2(N/4)$ and the number of node cells is $(N/8)*\log_2(N/4)$, when N is large, the system is dominated by the node cells in the shuffle-exchange architecture. Therefore, the area*time^2 complexity of the FFT systolic-network architecture approach that of FFT shuffle-exchange architecture asymptolically. That is, it also achieves the lowest bound of the area*time^2 complexity of order $O(N^2\log^2 N)$. When the number of sample points is small, the systolic-network architecture have a smaller chip area, but the tradeoff is that the throughput of the systolic-network architecture is one third that of shuffle-exchange FFT. If the FFT is implemented by using the systolic matrix multiplication only and is not combined with the shuffle-exchange architecture, the throughput will be reduced further and the trancation errors resulting from cascading of the multiply-add cells will become serious. Therefore, the combination of the four-point systolic complex number adder and shuffle-exchange architecture is

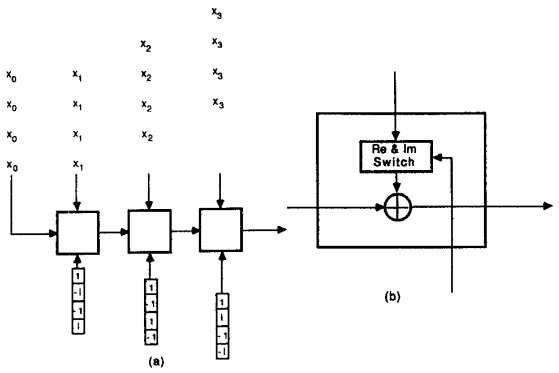

Fig. 2 A systolic complex number adder for 4-point FFT and it's basic cell.

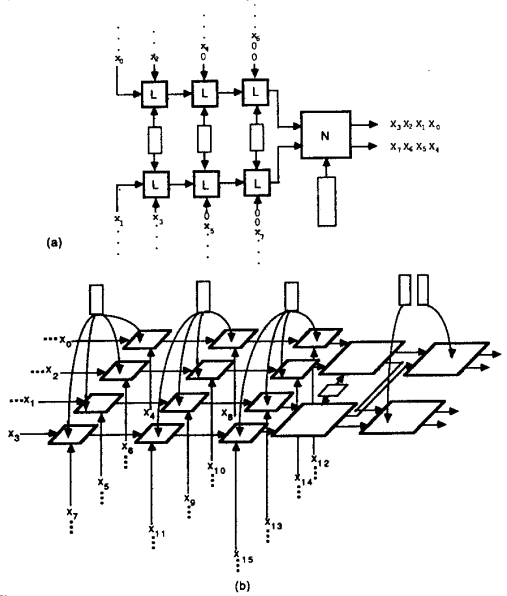

Fig. 3 8-point and 16 point FFT systolic-network architectures.

optimal in the sense that few multipliers are needed for small sample points and the area*time^2 complexity is optimal when the number of the sample points become large.

B. Computational Problems

One important question is that what kinds of computational problems can make good use of this architecture? A four by four DFT matrix requires 16 multiplications and 12 additions/subtractions. An decimation in time FFT matrix is obtained by exchanging the second and third columns. An decimation in frequency FFT may also be obtained by exchanging the second and third rows. Only eight multiplications and eight additions/subtractions are required for the computation of both matrices since the matrices can be recursively decomposed after rows or columns exchanges. We find that if the matrix, under either the exchanging of rows or cloumns (i.e., either decimate in time or in frequency), can be partitioned into four blocks which can also be partitioned recursively further, then it can make good use of this architecture. That is, a 2^m*2^m matrix [A(m)] can be decomposed into four $2^{m-1}*2^{m-1}$ matrices $[A_1(m-1)]$ and $[A_2(m-1)]$ as given by

$$[A(m)] = \begin{bmatrix} \pm[A_1(m-1)] & \pm[A_1(m-1)] \\ \pm[A_2(m-1)] & \pm[A_2(m-1)] \end{bmatrix}$$

or

$$[A(m)] = \begin{bmatrix} \pm[A_1(m-1)] & \pm[A_2(m-1)] \\ \pm[A_1(m-1)] & \pm[A_2(m-1)] \end{bmatrix}$$

The requirements that the $[A_1(m-1)]$ and $[A_2(m-1)]$ sub-matrices can be further decomposed recursively like [A(m)] are necessary conditions in order to apply the proposed systolic-network architecture. It should be noted that if the final four by four sub-matrix is not composed of only {0, 1, -1, i, -i}, then the shuffle-exchange architecture need to be used.

C. System Expansibility

A radix 2 16-point FFT can be built in one chip by using the proposed systolic-network architecture. To consider a 32 or more -points radix 2 FFT, we need to use some bridge multiply-add chips which contains four multiply-add cells in one chip.

III. Architectures of Some Discrete Orthogonal Transforms

Many discrete orthogonal transforms are frequently used in many transform coding, spectral estimation, and signal analysis problems [5]. The intensive arithmetic computations of these transforms are of major problem in implementing them in real-time applications. Some of these transforms are known to have fast computational algorithms which are very similar to FFT. If the shuffle-exchange or systolic-network architecture can solve these computational problems, then the host computer can change the twiddle factors in the buffers to perform different transformations. Such uniform solution provides more flexibility and simplicity than building different tramsform in different chips. In this section, some of the discrete orthogonal transforms which can be implemented in the shuffle-exchange or systolic-network architecture are presented.

A. Generalized Transform

Generalized transform is a class of transforms where their basis vectors are on the unit circle. The transform $(GT)_0$ yields the Walsh-Hadamard transform $(WHT)_h$, while $(GT)_{n-1}$, where $n=\log_2 N$ and N is the number of sample points of data sequence, yields the discrete Fourier Transform [5]. The transform matrices $[G_r(L)]$ can be generated recursively. For r=0,

$$[G_0(m)] = [H_h(m)] = \begin{bmatrix} [H_h(m-1)] & [H_h(m-1)] \\ [H_h(m-1)] & -[H_h(m-1] \end{bmatrix}$$

where $[H_h(0)]=1$ and $[H_h(m)]$ is the $(WHT)_h$ matrix of order $2^m * 2^m$.

Since such kinds of matrices can be decomposed recursively like the FFT and no multiplication is needed, the WHT can be implemented by using the systolic-network architecture with simpler basic cells than FFT. The basic cell is shown in Fig. 4a.

For r=1, 2, 3,..., n-2

$$[G_r(m)] = \begin{bmatrix} [G_r(m-1)] & [G_r(m-1)] \\ [A_r(m-1)] & -[A_r(m-1)] \end{bmatrix}$$

with $[G_r(0)]=1$ and $[G_r(1)]= [H_h(1)]$. Then $[A_r(k)]$ can be recursively generated by

$$[A_r(k)] = [B_r(r)] \otimes \begin{bmatrix} [H_h(k-r-1)] & [H_h(k-r-1)] \\ [H_h(k-r-1)] & -[H_h(k-r-1)] \end{bmatrix}$$

and $[B_r(r)]$ is a r-dependent matrix [5]. Therefore, the proposed systolic-network is suitable to the entire class of generalized transforms using the general basic cell shown in Fig. 4b.

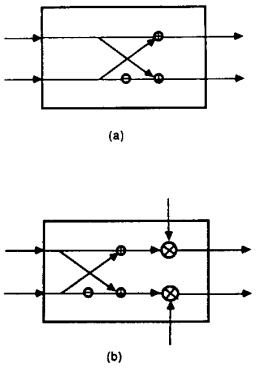

Fig. 4 Basic cells for WHT and Generalize transform.

B. Cosine trandform

Several VLSI architectures for implementation of the discrete Cosine transform (DCT) have been studied [4,10,11,12]. All of these structures were designed for special-purpose DCT chip. Makhoul [8] has shown that an N points DCT can be implemented by using a N points FFT. The procedure consists of starting with a given real sequence x(n), $0 \le n \le N-1$, perform:

(1) Form the sequence v(n), where v(n) is goven by

$$v(n) = \begin{cases} x(2n), & 0 \le n \le [\frac{N-1}{2}] \\ x(2N-2n-1), & [\frac{N+1}{2}] \le n \le N-1 \end{cases}$$

(2) Compute V(k), $0 \le k \le N-1$, the DFT of v(n);

(3) Multiply V(k) by $2\exp(-j\pi k/2N)$ and take the real part of it.

By using this algorithm, the DCT can be implemented in systolic-network architecture by reordering the sequence to fit the FFT computation.

C. Slant transform

Slant matrix can be generated recursively by decomposing the matrix as

$$[S(L)] = \frac{1}{\sqrt{2}} \left[\begin{array}{cc|c|cc|c} 1 & 0 & & 1 & 0 & \\ a_N & b_N & 0 & -a_N & b_N & 0 \\ \hline & 0 & I_{N/2-2} & & 0 & I_{N/2-2} \\ \hline 0 & 1 & & 0 & -1 & \\ -b_N & a_N & 0 & b_N & a_N & 0 \\ \hline & 0 & I_{N/2-2} & & 0 & -I_{N/2-2} \end{array}\right] \times (\mathrm{diag}[[S(L-1)], [S(L-1)]])$$

where $a_2=1$, $b_N=1/(1+4a_{N/2}^2)^{1/2}$, $a_N=2b_Na_{N/2}$, N=4, 8, 16, ... Computations similar to those for the FFT show that the shuffle-exchange and systolic-network architectures can also implement the Slant transform.

IV. Conclusion

Even thougth the area*time^2 complexity of an VLSI architecture for a given problem is optimal in the sense that the bound is achieved, the practical one-chip design of this optimal architecture is not necessarily optimal when N, the number of sample points, is small. Since the area*time^2 complexity has been considered from an asymptotic point of view, it does not show those factors that really affect the design when N is small. Until the wafer-scale-integration become practical, the one-chip design of an optimal architecture is still of considerable interest. In this paper we propose a systolic-network architecture suited for small sample points synthesis. The architecture is better than the shuffle-exchange architecture for an one-chip FFT design and it also achieves the optimal area*time^2 bound. But the tradeoff is that the throughput is slowed down by one third that of the shuffle-exchange architecture.

We can see also that many of the suboptimal discrete orthogonal transforms can be implemented by using the shuffle-exchange architecture or the systolic-network architecture. It is possible to connect such VLSI chip to a general computer system to perform the real time transform computation instead of running these transformations in software. In order to performance different transformations, the twiddle factors stored in the buffer memories need to be changed. The details of the communications between the transform chip and the computer are beyond the goal of this paper.

V. References

[1] C. D. Thompson, "Fourier transforms in VLSI," IEEE Trans. Comput., vol.C-32, pp.1047, Nov. 1983.

[2] H. T. Kung, "Why Systolic Architectures?," Computer, pp.37, Jan.1982.

[3] G. Bongiovanni, "Two VLSI structures for the discrete Fourier transform,"IEEE Trans. Comput. vol. C-32, pp.750, Aug. 1983.

[4] M. Vetterli & A. Ligtenberg, "A discrete Fourier-Cosine transform chip," IEEE Jour. select areas in Comm, vol. SAC-4, Jan. 1986.

[5] D. F. Elliott & K.R. Rao, "Fast transform - Algorithms, Analyses, applications," 1982.

[6] J. D. Ullman, "Computational aspects of VLSI," Computer Science Press, 1984.

[7] P. Bertolazzi & F. Luccio, " VLSI : Algorithms and Architectures," Elsevier Science Publishers B.V., Netherlands, 1985; G. Bilardi & M. Sarrafzadeh, "Optimal Discrete Fourier Transform in VLSI," pp.79.

[8] J. Makhoul,"A fast cosine transform in one and two dimensions," IEEE Trans. ASSP, pp.27, Feb. 1980.

[9] C. D. Thompson,"A complexity theory for VLSI," Ph.D. disseration, Carnegie-Mellon Univ. Aug. 1980.

[10] P. Duhamel & H. Hmida, "New lehgth-2^N DCT algorithm suitable for VLSI implementation," ICASSP, Dallas, Apr. 1987.

[11] N. Demassieux & F. Jutand, "An optimized VLSI architecture for a multiformat discrete cosine transform," ICASSP, Dallas, Apr. 1987.

[12] J. H. O'Neill & A. Ligtenberg, "A single chip solution for an 8 x 8 two-dimensional DCT," ISCAS, Philadelphia, Apr. 1987.