ZETA Networks

by

Setrag N. Khoshafian
Douglas M. Bates
David J. DeWitt

Computer Sciences Technical Report #509

August 1983

ZETA Networks

Setrag N. Khoshafian
Douglas M. Bates
David J. DeWitt

Computer Sciences Department
University of Wisconsin - Madison

This research was supported by the Department of Energy under contract #DE-AC02-81ER10920.

ABSTRACT

This paper introduces a multistage shuffle/exchange network, called the ZETA network, for processing all the pairs (X_i, X_j) of a given string of inputs $X_0, X_1, \ldots, X_{p-1}$. By using known results from algebra, it is shown that it is possible to determine a repetitive exchange pattern for a p−1 stage ZETA network, such that each pair (X_i, X_j) is the output from some processor at some stage. The main result is developed for values of p which are a power of 2. General suggestions are made to handle arbitrary values of p.

ZETA Networks

1. Introduction

In this paper we consider a type of interconnection network for an array of processors constructed so that the processors act on all possible pairs of p distinct input streams, $X_0, X_1, \ldots, X_{p-1}$. This type of network, the ZETA network, is a p−1 stage shuffle/exchange [STON71, CHUA81] where the pattern of exchanges is identical in each column. The processors at one stage process $\frac{p}{2}$ pairs and "pipe" their data to the processors at the next stage. Therefore, if there are several consecutive sets of p element input streams to be processed, the ZETA network acts as a pipeline, with consecutive stages processing pairs of consecutive input sets. All the processors will be executing the same operation but on different pairs of data (that is an SIMD [FLYN66] configuration).

The ZETA network was motivated by the need to evaluate the non-diagonal elements of the matrix X^TX where X is an n×p matrix of numerical values from a large statistical database. In such databases, n would typically be in the tens or hundreds of thousands while the p used in a particular problem would be much smaller, say between two and one hundred. This calculation is used in one method of analyzing the linear regression model

$$Y = Xb + \varepsilon \quad .$$

There have been systolic array [MEAD80] designs proposed for the parallel evaluation of the QR factorization of X [AHME82, GENT81, JOHN82, HELL83] which is another method of analyzing this regression model. Some discussion of the relative advantages of the X^TX method and the QR factorization method of analyzing a regression model can be found in [SEBE77] and [KENN80]. The ZETA network, however, is not restricted to this one problem. Since it is based on a shuffle/exchange it provides a simple and general interconnection method for any problem that requires the processing of all pairs of some inputs streams.

We first consider the special case where $p = 2^m$. The network can be pictured as having $\frac{p(p-1)}{2}$ processors arranged as $p-1$ columns with $\frac{p}{2}$ processors in each column. Each processor has two inputs and two outputs and the processors in one column are connected to the processors in the next column through a perfect shuffle. The exchange part of the shuffle/exchange comes from each processor being able to switch its inputs or send them straight through to its outputs. In a ZETA network, whether or not a processor switches its inputs is determined only by the row in which the processor is. That is, all the processors in a row switch their inputs, or all the processors in a row send their inputs straight through. This means that when $\frac{p(p-1)}{2}$ processors are not available, the network could be emulated by $\frac{p}{2}$ processors where successive stages are performed at successive times.

The foremost question to be asked of any interconnection network is whether all possible pairs of outputs are formed. There will certainly be $\frac{p(p-1)}{2}$ pairs formed but there may be repetitions and corresponding omissions. For example, a ZETA network where every processor did not switch inputs would repeat the entire pattern of pairings after m stages since it would be performing a perfect shuffle. (We are assuming that $p = 2^m$). The problem, therefore, is to determine an exchange pattern where all possible pairs are formed.

In Section 2 we show that such patterns can be constructed for any m. The formation of the patterns has an elegant algorithm which is related to error-detection codes and pseudo-random number generation. The general case of p not necessarily a power of 2 is considered in Section 3.

For the binary arithmetic operations, "." is binary multiplication (AND) and "⊕" is mod 2 addition (exclusive OR).

2. ZETA Networks

In this section we define ZETA networks and show that for a special class of ZETA networks we will be able to generate all the (X_i, X_j) pairs. As indicated earlier we assume that $p = 2^m$. The more general case will be discussed in Section 3.

Definition 1: A *ZETA* network is a p-1 stage shuffle/exchange network where, at each stage, the pattern of exchanges is the same.

Since the pattern of exchanges is the same at each stage, it is possible to emulate a ZETA network with just one stage of shuffle/exchange. At each stage of a ZETA network there are $\frac{p}{2}$-switching elements. We label these switching elements 0, 1, ..., $\frac{p}{2} - 1$. Each switching element has two outputs. At each stage, we label these outputs 0, 1, ..., p-1. Switching element P has outputs labeled 2P and 2P+1. An output labeled j issues from a switching element labeled $\left\lfloor \frac{j}{2} \right\rfloor$. Moreover, an output labeled j is some element X_i of the set $X_0, X_1, \cdots, X_{p-1}$. We shall call i the *index* of the output labeled j. At each stage of a ZETA network the outputs from the switching elements will form some permutation of this set or, equivalently, of 0, 1, ..., 2^m-1.

Definition 2: A ZETA network is *0-preserving* if the first switching element is not set (that is, it does not switch its inputs). This results in 0 being the index of the output labeled 0 at each stage.

Definition 3: A ZETA network is *i-complete* if i (that is X_i) is paired with each j (that is X_j) in some switching element.

Since our goal is to form all the (i, j) pairs, we are interested in ZETA networks which are i-complete for all i.

Definition 4: A ZETA network is *complete* if it is i-complete for all i=0, 1, ..., p-1.

Example: In Fig. 1 we have a 7 stage ZETA network. Switching elements 2 and 3 exchange their inputs. The ZETA network is 0-preserving and complete. At each stage we have labeled the switching elements, the outputs and indicated the binary representation of the indices. Note that, at each stage, the output indices constitute a permutation of 0, 1, ..., 7. For example, at stage 3 the outputs constitute the permutation $X_0, X_7, X_6, X_1, X_3, X_4, X_5, X_2$ of $X_0, X_1, \cdots, X_7$. This ZETA network has a very interesting property. At the bottom of each column of the switching elements we have given the binary representation of the index of of the element which is paired with 0. For example at stage 3, 0 is paired with 7 or 111. Now observe that if we take the exclusive OR of the binary representations of any other pair of indices at this stage we also get 111. This property is true for any stage of this ZETA network. In other words, at each stage of this ZETA network the binary representations of the indices of the outputs from the switching elements differ in the same bit positions.

Definition 5: A ZETA network is *bit difference preserving* if at each stage the binary representations of the indices of the outputs from the switching elements differ in the same bit positions.

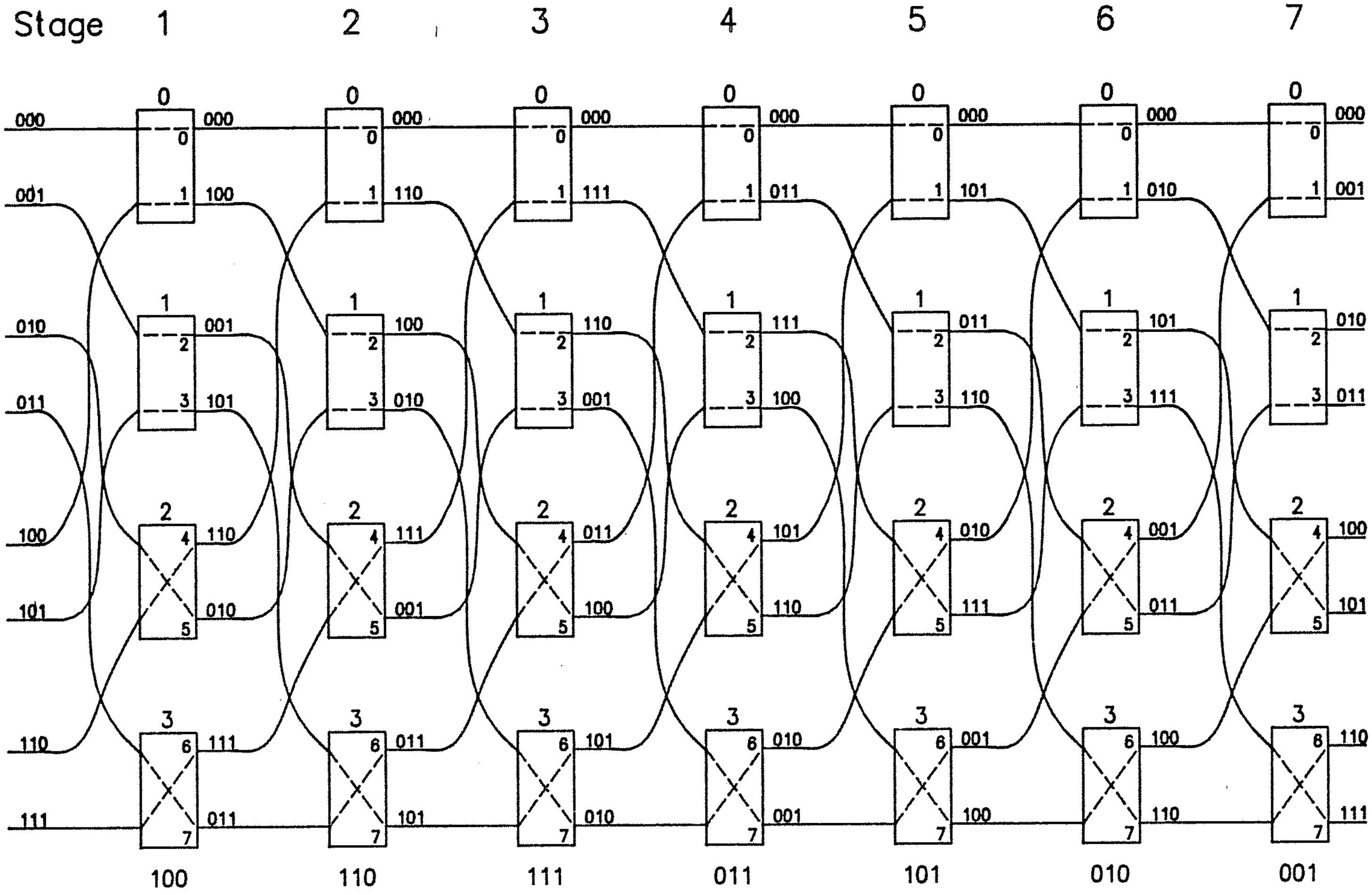
Stage
1
2
3
4
5
6
7
000
001
010
011
100
101
110
111
0
1
2
3
0
1
2
3
4
5
6
7
000
100
001
101
110
010
111
011
100
0
1
2
3
0
1
2
3
4
5
6
7
000
110
100
010
111
001
011
101
110
0
1
2
3
0
1
2
3
4
5
6
7
000
111
110
001
011
100
101
010
111
0
1
2
3
0
1
2
3
4
5
6
7
000
011
111
100
101
110
010
001
011
0
1
2
3
0
1
2
3
4
5
6
7
000
101
011
110
010
111
001
100
101
0
1
2
3
0
1
2
3
4
5
6
7
000
010
101
111
001
011
100
110
010
0
1
2
3
0
1
2
3
4
5
8
7
000
001
010
011
100
101
110
111
001

Fig. 1

If a ZETA network is bit difference preserving then for any pair (i, j) the binary representations of i and j will differ according to $d_{i,j} = i \oplus j$ and all the $d_{i,j}$'s at the same stage will be identical. But if the network is also 0-complete, any m-bit number will be paired with 0 at some stage. Hence any (i, j) will be the output of some switching element at some stage. Therefore:

Theorem 1: A bit difference preserving ZETA network is complete iff it is 0-complete.

Again consider the example in Fig.1. Note that the switching elements 0, 1, 2, 3 (or 00, 01, 10, 11) interchange their inputs iff $1.P_0 \oplus 0.P_1$ is 1 (P_0P_1 is the binary representation of the label). Therefore 10 (in binary) "determines" the exchange pattern. Moreover, consider the binary representations of the output indices from the switching elements, in any row of the network. It can be shown that, in the same row, the least significant two bits of a binary representation in a stage are obtained from the binary representation of the previous stage through a right shift. For example, at stage 3, 111 is paired with 0 and at stage 4, 011 is paired with 0. The least significant two bits of 011 are obtained from 111 through a right shift. The most significant bit of 011, however, is the complement of the least significant bit of 111. In fact it can be shown that for this ZETA network, the most significant bit of each output index is the exclusive OR of the least significant bit of the previous output index with the "inner product" of 10 (in binary) with the remaining bits of the previous output index. In our example the most significant bit 0 of 011 is equal to $1 \oplus 1.1 \oplus 1.0$.

Definition 6: Let $t=t_0t_1 \cdots t_{m-2}$ be any fixed (m-1) bit number. Then if the switching elements 0, ..., 2^{m-1}-1 switch their inputs when

$$t_0.P_0 \oplus t_1.P_1 \oplus \cdots \oplus t_{m-2}.P_{m-2} = 1 \qquad (2.1)$$

where $P = P_0P_1 \cdots P_{m-2}$ is the binary representation of switching element P, the ZETA network is said to be *t-determined.*

In what follows we shall show that for t-determined ZETA networks the binary representations of the permutation at stage k is determined from the binary representations of the permutation at stage k-1 through a right shift and the complementation of the most significant bit depending upon t and the remaining bits (see (2.2)). We shall also show that t-determined ZETA networks are bit difference preserving.

Theorem 2: Let $t=t_0t_1 \cdots t_{m-2}$ and consider the t-determined ZETA network. If $S_j^{(k)} = S_{0,j}^{(k)}S_{1,j}^{(k)} \cdots S_{m-1,j}^{(k)}$ is the binary representation of the index of the output labeled j at stage k, then:

$$S_{i,j}^{(k)} = S_{i-1,j}^{(k-1)} \text{ for } i = 1,\ldots,m-1$$

and (2.2)

$$S_{0,j}^{(k)} = S_{m-1,j}^{(k-1)} \oplus S_{0,j}^{(k-1)}.t_0 \oplus S_{1,j}^{(k-1)}.t_1 \oplus \cdots \oplus S_{m-2,j}^{(k-1)}.t_{m-2}$$

Proof: We shall prove by induction. Clearly in the first stage of the shuffle exchange network we have:

$$S_{i,j}^{(1)} = S_{i-1,j}^{(0)} \text{ for } i = 1,\ldots,m-1 \qquad (2.3)$$

and in fact (2.3) holds for any exchange pattern. We need to show the second equality of (2.2) holds for k=1. First note that $S_{0,j}^{(0)}S_{1,j}^{(0)} \cdots S_{m-2,j}^{(0)}$ is switching element $\left\lfloor \frac{j}{2} \right\rfloor$ and, since the network is t-determined, it is switched iff (2.1) holds. But switching corresponds to complementing $S_{m-1,j}^{(0)}$. Therefore (2.2) holds for k=1.

Assume (2.2) holds for $k \leq (n-1)$. Now there exists a r such that $S_j^{(n)}$ and $S_j^{(n-1)}$ are equal to $S_r^{(n-1)}$ and $S_r^{(n-2)}$ respectively: r is the index which, when shuffled and exchanged, gets mapped into j. But then, by the induction hypothesis, $S_r^{(n-1)}$ is obtained from $S_r^{(n-2)}$ through (2.2).

Q.E.D.

Theorem 3: The t-determined ZETA networks are bit difference preserving.

Proof: Again we shall prove our contention by induction. Now for k=1 (that is at the first stage), no matter what the exchange pattern is, the binary representations of the output indices from the switching elements differ only in the most significant bit position. Next assume for $k \leq n-1$ the output indices from the switching elements have the bit difference given by $S_1^{(k)}$. That is assume for the two outputs $S_{j_1}^{(k)}$ and $S_{j_2}^{(k)}$ from any switching element j

$$S_{i,j_1}^{(k)} \oplus S_{i,j_2}^{(k)} = S_{i,1}^{(k)} \quad \text{for } i = 0, \ldots, m-1 \tag{2.4}$$

holds for k=1,...,n-1. That (2.4) holds for k=n follows from the fact that $S_{j_1}^{(n)}$, $S_{j_2}^{(n)}$ and $S_1^{(n)}$ are obtained from $S_{j_1}^{(n-1)}$, $S_{j_2}^{(n-1)}$ and $S_1^{(n-1)}$ through (2.2).

Q.E.D.

Since t-determined ZETA networks are obviously 0-preserving and, as shown above, also bit difference preserving, to show that a t-determined ZETA network is complete we need to show it is 0-complete (Theorem 1). It is easy to see that not all t-determined ZETA networks are 0-complete. For example if t=00...0 we get the perfect shuffle (without any exchanges) at each stage and 0 will be paired, repeatedly, with 2^k for k=(m-1), (m-2),...0. To find t-determined ZETA networks which are 0-complete we need to determine the t's which, starting with 2^{m-1}, "generate" through (2.2) all the 2^m-1 non-zero m-bit numbers. This problem has been solved algebraically and 2^m-1 corresponds to the maximum period possible for a linear feedback shift register of m stages [GOLO67]. It has applications in generating m-bit pseudo random numbers with maximum periodicity [KNUT81].

In what follows we assume our reader has some familiarity with Galois fields GF(2^m) and the algebra of polynomials over a field (see [BIRK70, PETE72]). Below we give some preliminary definitions and two theorems which will guarantee the existence of 0-complete ZETA networks for any m. Since these results are well known in algebra and in the literature of error generating codes we just state them without proof.

Definition 7: The order of a nonzero element α of a multiplicative group G is the smallest positive integer r such that $\alpha^r = \alpha$. Note that $\alpha^{k+r} = \alpha^k$ for all k and the sequence $\{ \alpha^k \}_{k=0}^{\infty}$ is periodic of period r, with the first r elements all distinct.

Definition 8: An element α of GF(2^m) is called *primitive* if its order is 2^m-1. Every non-zero element of GF(2^m) can be expressed as a power of α and the multiplicative group of the non-zero element of GF(2^m) is cyclic.

Definition 9: An irreducible polynomial of degree m over GF(2) is called a *primitive polynomial* if it has a primitive element of GF(2^m) as a root.

Theorem 4 [KNUT81, BIRK70]: Over GF(2) there are $\frac{\varphi(2^m - 1)}{m}$ primitive polynomials of degree m, where φ is Euler's φ function.

Theorem 5 [BIRK70, PETE72]: Let

$$T(X) = 1 + t_0.X + t_1.X^2 + \cdots + t_{m-1}.X^m \tag{2.5}$$

be a primitive polynomial of degree m over GF(2) (note that $t_{m-1} = 1$). Then starting with any initial non-zero m-bit number $S^{(0)} = S_0^{(0)} S_1^{(0)} \cdots S_{m-1}^{(0)}$ the sequence $\{ S^{(i)} \}_{i=0}^{\infty}$ generated by

$$S_i^{(k)} = S_{i-1}^{(k-1)} \quad \text{for } i = 1, \ldots, m-1$$

$$\text{and} \tag{2.6}$$

$$S_0^{(k)} = S_{m-1}^{(k-1)} \oplus \sum_{i=0}^{m-2} S_i^{(k-1)}.t_i$$

is periodic of period $2^m - 1$.

Theorem 6: For any m we can always find a ZETA network of 2^m-1 stages which is complete.

Proof: For any m we can always find a primitive polynomial of degree m over GF(2) (Theorem 4). Let T(x) given by (2.5) be such a primitive polynomial. Consider the t-determined ZETA network, where $t = t_0 t_1 \cdots t_{m-2}$. By Theorem 3, any t-determined ZETA network is bit difference preserving. Moreover, Theorem 5 implies this t-determined ZETA network is also 0-complete. We are done by Theorem 1.

Q.E.D.

3. The General Case

In the previous section we showed that we can always find a ZETA network which will generate all the pairs (X_i, X_j) of $X_0, X_1, \cdots, X_{p-1}$, where $p = 2^m$. There are two basic problems in constructing a ZETA network for an arbitrary p: (1) it is not realistic to assume that the number of inputs will always be a power of two; (2) for large p it might be unreasonable to construct a ZETA network of p-1 stages (which requires $\frac{p(p-1)}{2}$ processors). In this section we propose a number of solutions for these two problems.

(a) *Introducing null elements:* perhaps the easiest way to handle the first case is through the introduction of null elements to the original input. That is find the smallest m such that $p \leq 2^m$ and append $2^m - p$ null elements to the original input. Note that with this scheme there are $\frac{2^m(2^m-1)}{2} - \frac{p(p-1)}{2}$ null operations. If p is $2^k + s$ for a relatively small s (for example s=1) this could degrade the throughput considerably.

(b) *Emulation (for large p):* if p is slightly less than a power of 2 but still very large, it might be unreasonable to construct the ZETA network. We mentioned earlier that the repetition of the exchange patterns allows us to emulate the ZETA network through just one stage of shuffle/exchange (by piping the data back to the switching elements a total of p-2 times).

If $\frac{p}{2}$ is still very large, it is possible to use quotient networks [FISH82] and emulate a shuffle/exchange of 2^{m+q-1} switching elements with a shuffle/exchange of 2^{m-1} switching elements. Each switching element will have 2^{q+1} inputs/outputs and must process 2^q pairs at each stage (for a total of 2^{m+q-1} stages).

(c) *Partitioning:* another way to handle the general case is by partitioning. Suppose $p = q.2^k$. There are two methods for partitioning:

(1) we can partition the p inputs into q subsets of 2^k elements each. Then a 2^k-1 stage ZETA network can be used to process the input pairs within each subset. The pairs across the subsets might be processed serially, through broadcasting etc.. The latter gives maximum efficiency if the ZETA network is emulated through a one stage shuffle/exchange. In this case if the switching elements contain blocks from one subset, the corresponding

blocks from another subset can be broadcast to all the switching elements, one by one. Each switching element will process two pairs across subsets and all the switching elements will be operating in parallel.

(2) we can partition the p inputs into 2^k subsets of q elements each. With this scheme, the ZETA network is used to process the pairs across the subsets and each switching element will process q^2 pairs at each stage. As with the previous scheme, there are several choices for processing the pairs within each subset. Unlike (1), however, the ZETA network is accessing all the input. Therefore, it is conceivable to uniformly distribute the amount of computation within each subset (that is processing $\frac{q(q-1)}{2}$ pairs), across the 2^k-1 stages of the network.

4. Summary and Conclusion

We have shown that we can determine a pattern of exchanges for a ZETA network, such that all the (X_i, X_j) pairs of a given set of inputs $X_0, X_1, \cdots, X_{p-1}$ are processed. We defined ZETA networks with this property "complete". We saw that primitive polynomials can determine complete ZETA networks, and since there are $\frac{\varphi(2^m-1)}{m}$ primitive polynomials (over GF(2)) for any m, it is always possible to construct a complete ZETA network for any m.

ZETA networks can be used very efficiently to evaluate the nondiagonal elements of X^TX. In the types of applications we are currently investigating, X is very large. In particular, X is much larger than the primary memory of the underlying system. Therefore, with a $p-1$ stage ZETA network, it is possible to process the matrix X in horizontal stripes. As we mentioned in Section 1, the ZETA network acts as a pipeline, with consecutive stages processing consecutive stripes of X. Therefore, if X is divided into N horizontal stripes and the time to access a horizontal stripe from secondary store is synchronized with the processing time needed at a stage of the ZETA network, X^TX can be evaluated in p+N time steps [1].

Finally, the repetitive exchange pattern allows the ZETA network to be emulated through just one stage of shuffle/exchange. This requires only $\frac{p}{2}$ processors. This emulation is one way to handle a large p. Emulation through quotient networks, introduction of null values and partitioning are some of the other methods that can be used to efficiently handle an arbitrary p (either very large or not equal to a power of two).

Acknowledgments

We wish to thank Dinna Bitton for her encouragement and helpful comments. We also like to thank Sam Bent for his suggestions on bit difference preservation and Jim Smith for his input and comments on an initial draft of this paper.

References:

[AHME82] Ahmed, H.M., Delomse, J.M., and, Morf, M. "Highly concurrent computing structures for matrix arithmetic and signal processing" Computers, 15, no.1 (Jan. 1982), pp. 65-82.

[BIRK70] Birkhoff, Garett, and Thomas, Bartee C., *Modern Applied Algebra*, McGraw-Hill, Inc., 1970.

[CHUA81] Chuan, WU, and Tse-Yun, Feng "The Universality of the Shuffle-Exchange Network" IEEE Transactions on Computers, vo. c-30, No. 5, pp.324-331.

[DONG79] Dongarra, J., Moler, C., Bunch, J., and G. Stewart, *Linpack Users' Guide*, SIAM, 1979.

[1] where a "time step" is the processor time needed at a stage of the network.

[FISH82] Fishburn, John P., and Finkel, Raphael A., "Quotient Networks," IEEE Transactions on Computers, Vol. c-31, No. 4, April 1982, pp.288-295.

[FLYN66] Flynn, J., "Very High-Speed Computing Systems," Proceedings of IEEE, pp. 1901-1906, 1966.

[GENT81] Gentleman, W.M., and Kung, H.T. "Matrix triangularization by systolic arrays" in Real Time Sugnal Processing IV: SPIE Proceedings vol. 298, 1981, pp. 329-336.

[GOLO67] Golomb, S.W., Shift-Register Sequences, Holden-Day, Inc., San Francisco, 1967.

[GOLU73] Golub, G. and G. Styan, "Numerical Computations for Univariate Linear Models," Journal Statistical Computing, Vol. 2, pp. 253-274, 1973.

[HELL83] Heller, Don E., and Ipsen, Isle C.F. "Systolic Networks for Orthogonal Decompositions," SIAM J. SCI STAT. COMPUT., Vol. 4, No. 2, June 1983, pp. 261-269.

[JOHN82] Johnsson, L., "A computational array for the QR-method" Proc. of Conference on Advanced Research in VLSI P. Penfield, ed., Artech House, Inc., Dedham, MA, 1982, pp. 123-129.

[KENN80] Kennedy, W. and J. Gentle, *Statistical Computing*, Marcel Dekker, Inc., 1980.

[KNUT81] Knuth, Donald E., *The Art of Computer Programming*, Volume 2, Seminumerical Algorithms Addison-Wesley, 1981.

[MEAD80] Mead, C.A., and Conway, L.A. *Introduction to VLSI Systems*, Addison-Wesley, Reading, Mass., 1980.

[PETE72] Peterson, Wesley W., and Weldon, E.J. Jr., *Error-Correcting Codes*, The MIT Press, 1972.

[SEBER77] Seber, G., *Linear Regression Analysis*, John Wiley & Sons, 1977.

[STON71] Stone, H. S. "Parallel processing with the perfect shuffle," IEEE Trans. Comput. C-20, pp.153-161, Feb 1971.