신뢰성을 고려한 병열프로세서에서 구성

論 文
31～5～1

—혼선없는 데이타의 액세스와 메모리프로세서의 연결회로망—

Implimentation of Parallel Processor System with Reliability

—Conflict-free Access of Data and Memory Processor Interconnection Network—

高 明 三*・정 택 원**
(Myoung-Sam Ko・Taeg-Won Jeong)

Abstract

In numerical computation, it is desirable to access any row or column, the main diagonal, subarrays of a matrix without any conflict for successful parallel processing. To meet this requirement special storage scheme is used for conflict-free access of necessary data.

Interconnection network, which connects processing elements and processing element memory modules, is required to execute the necessary operations.

In this paper we discuss the skewing method for conflict-free access to various bit slices and single-stage interconnection networks.

1. Introduction

As the switching speed of the computing devices approach a limit, parallel processing has been considered to improve the computer throughput.

Each processor in parallel processing computer system must be processed independently for successful parallel processing. This requires that each processor can be able to accesss necessary data without any conflict. This requirement may be satisfied by using many memory modules. Each memory module operates independently so that accesses to different modules can be overlapped to improve the overall performance of the computer.

* 正 會 員：서울大 工大 計測制御工學科 敎授・工博
** 正 會 員：서울大 大學院 計測制御工學科
接受日字：1981年 11月 6日

To meet this requirement, Kuck[1] discussed skewing storage scheme of data which was proposed by Lawrie.[2] Lawrie considered skewing storage scheme when the number of processors is an even power of two.

The skewing method results data not in the order when fetched from memory modules. Thus, it is necessary to unscramble the data so that they are in the correct order.

Unscrambling of data means the realization of a permutation. Therefore an interconnection network is required to realize a permutation.

There are some interconnection networks. The traditional $N \times M$ crossbar switch can be used as interconnection network. But this network is too expensive to use in large systems.

Clos's network[3] is another one but control procedure for parallel processing is not known. This

network is a three stage network constructed from crossbar switches.

The other interconnection network is the rearrangeable switching network by Benes.[4] The time delay for this network is too long to use in parallel processing.

To satisfy the need for low cost and a little time delay, Lawrie[2] proposed a network called omega network. Lawrie's network is a generalization of shuffle-exchange network by Stone.[5][6]

Lang[7] modified Lawrie's network to perform some permutations in less steps than Lawrie's network.

Lang and Stone[8] modified Lawrie's network so that the modified network has destination tag of one bit per processor.

In this network, the control variables at step k are determined by a very simple Boolean function of the step at $k-1$. Thus, the control requires a single bit per datum as compared to $\log_2 N$ bits per datum in the original procedure.

All of these networks are multi-stage network and requires much time delay to realize the given permutation. Pradhan and Kodandapani[9] presented single-stage networks and multi-stage networks when the number of processors and the skewing distance are co-prime. In this paper we discuss on the following problems;

1. skewing storage scheme when the number of processors is an odd power of two
2. single-stage network for scrambling/unscrambling of t-ordered vector when the skewing distance t and the number of memory modules are not co-prime.

2. Computer Model

We assume that the computer model proposed in this paper is a single instruction multiple data stream(SIMD) machine and is given by Fig.1. The computer consists of an instruction decoding and control unit, N processing elements, M memory modules and an interconnection network.

The single instruction is read and decoded by the instruction decoding and control unit. Instructions for the control unit are executed there. And instructions for processing elements are sent to processing elements, and all processing elements execute the same instruction simultaneously.

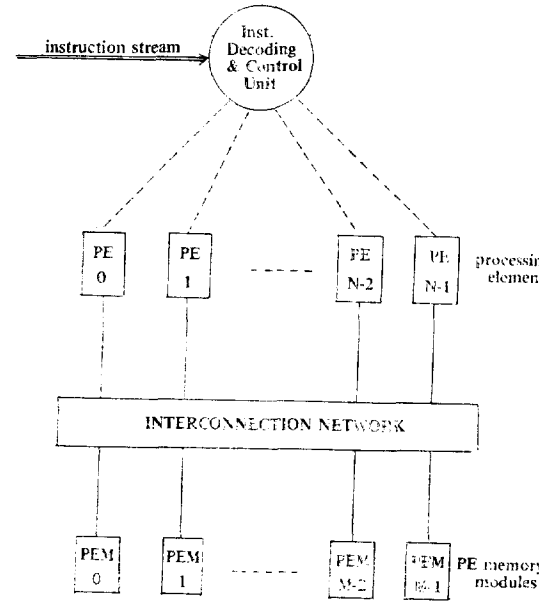

Fig. 1. Computer model

Each processing element has an index register and the address of the operand can be obtained by adding the content of this register to the address in the instruction. Therefore every processing element fetch a word from its own memory by a single load instruction. As a result is that a vector of data will be able to be fetched in one memory access.

3. Date Storage

It is assumed that data are processed in the form of N-vector. Since, it is usual to handle matrices by rows, columns, diagonals, or blocks for numerical applications. There fore much attention has been paid to stroage schemes that allow conflict-free access to these subarrays.[2][10] Thus, it is necessary to store all the elements in any row, column, diagonal, and block in different memory module.

Budnik and Kuck[10] used $2^{2L}+1$ memory modules with positive integer L to store arrays so that rows or columns, the main diagonal and square subarrays could all be fetched with a single memory access. Fig. 2 shows an example of Budnik and Kuck.

address	memory modules 0	1	2	3	4
0	$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	
1	$a_{1,3}$		$a_{1,0}$	$a_{1,1}$	$a_{1,2}$
2	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$		$a_{2,0}$
3		$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

Fig. 2. 4×4 matrix stored with $2L^{2L}+1$ memory modules where $L=1$

This example shows that any row or column, the main diagonal, and all 2×2 subarrays can be fetched with one memory access. When a column is fetched, the element of the column are not in the proper order. Elements that should be adjacent are two memories apart. This results a 2-ordered vector.

In the t-skewing by Kuck,[1] the data are stored t-memories apart from the previous data. This results a t-ordered vector when fetched from memory. Thus, the i-th element of the vector is stored in memory module numbered t_i mod M where M is the number of memory modules.

In Lawrie's[2] $(t_1, t_2, \cdots\cdots, t_i, \cdots\cdots, t_k)$ skewing, t_i is the skewing distance in the i-th dimension. Thus, for a two dimensional $N\times N$ matrix, rows are t_1-ordered, columns are t_2-ordered, the forward diagonals are (t_1+t_2)-ordered, and the backward diagonals are (t_1-t_2)-ordered.

By Lawrie[2], a sufficient condition for a t-ordered N-vector to be accessible without conflicts can be represented by equation.[1]

$$M\geq N \text{ gcd } (t, M) \tag{1}$$

where gcd (t, M) represents the greatest common divisor of t and M.

Lawrie considered $(N^{\frac{1}{2}}+1, 2)$ skewing where N represents the number of processor and N is an even power of two. This results conflict-free access of any row or column, the forward diagonal, the backward diagonal with $M=2N$. But the skewing distance by Lawrie is not unique. Using $(N^{\frac{1}{2}}+2, 1)$ skewing, we can get the same results as Lawrie's.

If N is an odd power of two, Lawrie's $(N^{\frac{1}{2}}+1, 2)$ skewing can not be applied because $N^{\frac{1}{2}}$ is not integer. We considered the case when N is an odd power of two.

Theorem 1.

Assume the number of processor N to be an odd power of two and the number of memory modules $M=2N$. If p is the maximum value in p' where p' satisfies the equation (2)

$$\text{gcd}(p', M)=2, p'<N^{\frac{1}{2}} \tag{2}$$

Then we have conflict-free access of rows or columns, the forward diagonals, the backward diagonals when stored using$(p, 1)$ skewing or$(p+1, 2)$ skewing.

proof; When$(p, 1)$ skewing is used, rows are p-ordered and columns are 1-ordered. And the forward diagonals are $(p+1)$-ordered, the backward diagonals are $(p-1)$-ordered.

gcd $(p, M)=2$,

gcd $(1, M)=$gcd $(p+1, M)=$gcd $(p-1, M)=1$.

Thus, the sufficient condition of equation (1) is satisfied.

Theorem 2.

Let p, M and N be the same as in Theorem 1. Then and $p\times p$ subarray of $N\times N$ matrix can be accessed in a single memory access.

proof; The elements of the $p\times p$ block $a_{i,j}, a_{i,j+1}, \cdots\cdots, a_{i,j+p-1}, a_{i+1,j}, a_{i+1,j+1}\cdots\cdots, a_{i+p-1,j}, a_{i+p-1,j+1}, \cdots, a_{i+p-1,j+p-1}$ will be stored in memory modules numbered $M_{x1,x2}=(i+x^1)p+(j+x^2)$ mod $M=(i+x^1)p+(j+x^2)$ mod $2N$ for $0\leq x^1, x^2<p$. We need only show that these memory modules are distinct. In other words it is necessary to show that

$$M_{x1,x2} \overline{2N} M_{y1,y2} \text{ iff } x^1=y^1 \text{ and } x^2=y^2$$

for $0\leq x^1, x^2, y^2, y^1<p$ where $a\underset{b}{=}c$ means that there exists an integer k such that $a=c+kb$.

Assume that $M_{x1,x2}\overline{2N}M_{y1,y2}$.

Rearranging gives

$$(x^1-y^1)p\overline{2N}y^2-x^2 \tag{3}$$

Without loss of generality we may assume that

$x^1-y^1\geq 0$

case 1. $x^1-y^1=0$

Since $2N>y^2-x^2>-2N$, the Equation (3) gives

$y^2-x^2=0$.

case 2. $x^1-y^1>0$

a) $y^2-x^2>0$

By using the relations $2N>(x^1-y^1)p>0$ and $2N>y^2-x^2$, Equation (3) gives

$(x^1-y^1)p=y^2-x^2$

Since that $(x^1-y^1)p-(y^2-x^2)>0$, this equation can not be satisfied.

b) $y^2-x^2<0$

By using the relations $2N>2N+(y^2-x^2)>0$, Equation (3) can be written

$(x^1-y^1)p=2N+(y^2-y^2)$

There are no x^1, x^2, y^1, y^2 satisfying this equation because of $(2N+y^2-x^2)-(x^1-y^1)p>0$.

c) $y^2-x^2=0$

above equation gives

$x^1-y^1=0$.

Proof is completed.

4. Interconnetion Network

As shown in section 3, data not in the order when fetched from the memory modules. This requires the reordering of data to execute the necessary operations. Therefore the interconnection network is very important in parallel processing.

In the following, we give a brief review of some interconnection networks.

(A) Lawrie's omega network

Lawrie proposed a network consisted of $\log_2 N$ identical shuffle exchange networks.

The permutation p is usually represented as $(i, p(i))$, where $p(i)$ represents the mapping of $i, 0\leq i\leq N-1$. The shuffle network performs the perfect shuffle permutation expressed in equation(4).

$$p(i)=2i \bmod N \tag{4}$$

And the exchange network exchanges two data if exchange operation is ordered by a control bit. Fig. 3 shows an 8×8 omega network.

Let (S, D) be the permutation to be realized. We say that S is the source tag and D the destination tag. Assume that $s_n s_{n-1}\cdots s_1$ and $d_n d_{n-1}\cdots d_1$ are the binary representation of S and D, respectively where $n=\log_2 N$

At first stage S is switched to $s_{n-1}s_{n-2}\cdots s_1 s_n$ by the shuffle network and $s_{n-1}s_{n-2}\cdots s_1 s_n$ is switched to the upper (lower) output of the exchange network if $d_n=0(d_n=1)$. After passing n shuffle exchange networks, input S is connected to output D. Fig. 4 shows the connection (011, 010) and (010, 100)

The omega network can not realize all permutat-

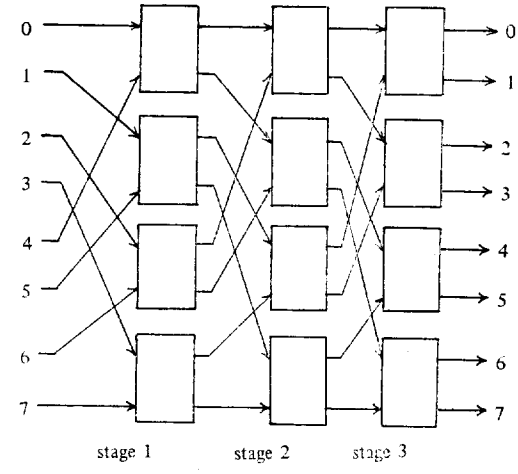

Fig. 3. 8×8 omega network

ions because of conflicts. Fig. 5 shows an example of conflicts. Lawrie showed that the omega network can produce uniform shifts and unscrambling of t-ordered vectors where t is odd. We will show that we can unscramble t-ordered vectors withe t even by single-stage network as follows.

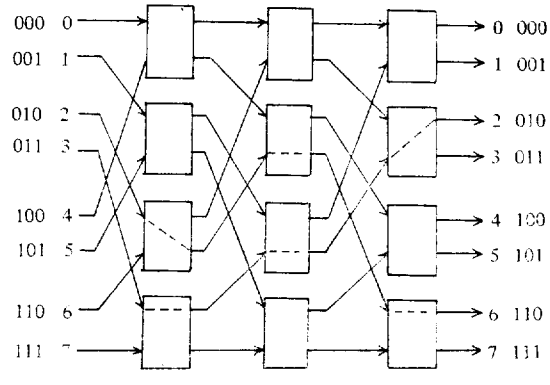

Fig. 4. Connection (011, 010) and (010, 110)

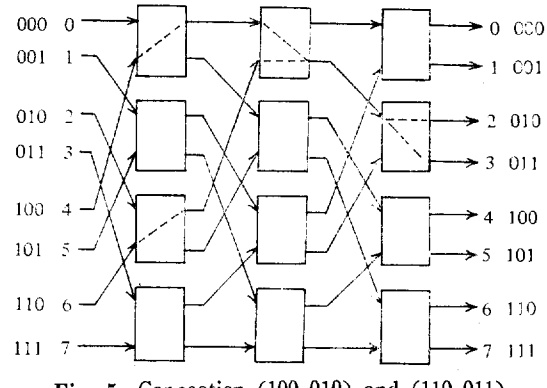

Fig. 5. Connection (100, 010) and (110, 011)

(B) Single-Stage Network

Multi-stage networks have much time delay. Pradhan and Kodandapani[9] proposed single stage networks to shorten the time delay. They considered perfect shuffle network, uniform shift network, plus-minus 2^i(PM2I) network and scrambling/unscrambling network.

The scrambling/unscrambling means the realization of a permutation of the type

$$p(i)=ti \bmod N.$$

Pradhan and Kodandpani discussed the case when t and the number of processors N are relatively prime. If we use $(p, 1)$ skewing in section 3, the result of Pradhan and Kodandapani can not be applied because the greatest common divisor of p and N is not 1. Hence we considered the case when p and the number of memory modules M are not relatively prime, which can be applied to the case when gcd (p, M) is not 1.

The Boolean difference of $g(x_n, x_{n-1}, \cdots, x_i, \cdots, x_1)$ with respect to x_i is written dg/dx_i and defined by the following equation:

$$dg/dx_i=g(x_n, \cdots, x_i=1, \cdots, x_1)\oplus g(x_n, \cdots, x_i=0, \cdots, x_1)$$

where $\oplus$ is the exclusive-OR operator.

In the following, $t_n t_{n-1}\cdots t_1$ is used for the binary representation of a number t. Any permutation $p(i)=j$ can be represented by n functions j_k, $1\leq k\leq n$, where $i=i_n i_{n-1}\cdots i_1$ and $j=j_n j_{n-1}\cdots j_1$. We give a theorem about switching functions when the greatest common divisor of t and M is not 1 where t is the skewing distance and M is the number of memory modules.

Theorem 3. Assume $\gcd(t, M)=2^m$. Then, the functions j_k, $1\leq k\leq m+n$, that represent the permutation expressed as

$$p(i)=ti \bmod M$$

can be written as

$$j_{n+m}=i_n\oplus g_{n+m}(i_n, i_{n-1}, \cdots, i_1)$$

$$j_k=0,\ 1\leq k\leq m$$

and the other functions $j_{n+m-1}, j_{n+m-2}, \cdots, j_{m+1}$ are related by the following recursive rule such that

If

$$j_k=i_{k-m}\oplus g_k(i_n, i_{n-1}, \cdots, i_1)$$

then

$$j_{k-1}=i_{k-m-1}\oplus dg_k/di_{k-m-1}\oplus t_{m+2}$$

where

$$t=t_n t_{n-1}\cdots t_1.$$

proof; Since gcd $(t, M)=2^m$, t can be expressed as

$$t=t_n t_{n-1} t_{n-2}\cdots t_{m+2}100\cdots 0.$$

From the multiplication table of Table 1, we get

$$j_{n+m}=t_{m+1}i_n\oplus t_{m+2}i_{n-1}\oplus\cdots\oplus t_n i_{m+1}\oplus C_{n+m}$$
$$=i_n\oplus g_{n+m}(i_n, i_{n-1}, \cdots, i_1)$$
$$j_{n+k}=t_{k+1}i_n\oplus t_{k+2}i_{n+1}\oplus\cdots\oplus C_{n+k}$$
$$=i_{n+k-m}\oplus t_{m+2}i_{n+k-m-1}\oplus\cdots\oplus t_n i_{k+1}\oplus C_{n+k}$$
$$=i_{n+k-m}\oplus g_{n+k}(i_n, i_{n-1}, \cdots, i_1) \quad (6)$$
$$j_{n+k-1}=t_k i_n\oplus t_{k+1}i_{n-1}\oplus\cdots\oplus t_n i_k\oplus C_{n+k-1}$$
$$=i_{n+k-m-1}\oplus t_{m+2}i_{n+k-m-2}\oplus\cdots\oplus t_n i_k\oplus C_{n+k-1} \quad (7)$$

Table 1. Multiplication table

		i_n	i_3	i_2	i_1
		t_n	t_3	t_2	t_1
		$t_1 i_n\cdots\cdots t_1 i_3$		$t_1 i_2$	$t_1 i_1$
$t_{m+1}i_n\cdots\cdots t_{k+1}i_n\cdots\cdots t_2 i_{n-1}\cdots\cdots t_2 i_2$				$t_2 i_1$	
$t_{m+2}i_{n-1}\cdots t_{k+2}i_{n-1}\cdots\ t_3 i_{n-2}\cdots\cdots t_3 i_1$					
$\cdot\cdots\cdots$	$\cdot\ \cdots\cdots$	$\cdot$	C_3		
$\cdot\cdots\cdots$	$\cdot\ \cdots\cdots$	$\cdot$			
$\cdot\cdots\cdots$	$\cdot\ \cdots\cdots$	$\cdot$			
$t_n i_{m+1}\cdots\cdots t_n i_{k+1}\cdots\cdots t_n i_1$					
$C_{n+m}\cdots\cdots C_{n+k}\cdots\cdots C_n$					
j_{n+m}	j_{n+k}	j_n	j_3	j_2	j_1

In Table 1, for $3\leq k\leq n+m$, C_k represent the carry bits generated during the summation of $k-1$ columns from right. Equation (6) gives

$$g_{n+k}=t_{m+2}i_{n+k-m-1}\oplus t_{m+3}i_{n+k-m-2}\oplus\cdots\oplus t_n i_{k+1}\oplus C_{n+k}$$

Hence,

$$dg_{n+k}/di_{n+k-m-1}=t_{m+2}\oplus dC_{n+k}/di_{n+k-m-1} \quad (8)$$

C_{n+k} can be expressed as

$$C_{n+k}=C^1{}_{n+k}\oplus C^2{}_{n+k}$$

where $C^1{}_{n+k}$ represents the carry bit generated in the addition of $(k-1)$ st column only from right and $C^2{}_{n+k}$ represents the EX-OR sum of the carry bits generated in the addition of j-th columns from right, $2\leq j\leq k-2$. Thus,

$$dC_{n+k}/di_{n+k-m-1}=dC^1{}_{n+k}/di_{n+k-m-1}$$

Table II shows that

$$dC^1{}_{n+k}/di_{n+k-m-1}=t_{m+2}i_{n+k-m-2}\oplus t_{m+3}i_{n+k-m-3}\oplus\cdots$$
$$\cdots\oplus t_n i_k\oplus C_{n+k-1} \quad (9)$$

Equations (7)(8) and (9) give

$$j_{n+k-1}=i_{n+k-m-1}\oplus dg_{n+k}/di_{n+k-m-1}\oplus t_{m+2}$$

Similarly, we can get the same result for j_k and j_{k-1}, $n \geq k \geq m+1$.

Table 2. The truth table for $dC^1_{n+k}/di_{n+k-m-1}$

p	$C^1_{n+k}(i_{n+k-1}=1)$	$C^1_{n+k}(i_{n+k-m-1}=0)$	$dC^1_{n+k}/di_{n+k-m-1}$
$4z$	0	0	0
$4z+1$	1	0	1
$4z+2$	1	1	0
$4z+3$	0	1	1

In Table 2, p *is* the number of 1's in $i_{n+k-m-1}$, $t_{m+2}i_{n+k-m-2}, \cdots, t_n i_k, C_{n+k-1}$. And z is any integer.

5. Conclusion

We showed that the skewing method yields conflict-free access of rows, columns, diagonals, and subarrays of a matrix when the number of processors is an odd power of two if the skewing distance is properly determined.

The switching functions representing the single-stage interconnection network for scrambling/unscrambling are developed when the skewing distance and the number of memory modules are not relatively prime.

Acknowledgements

This study was supported partly by the Hyundai's Research Fund in 1979

References

〔1〕 D.J. Kuck; "ILLIAC IV software and application programming", IEEE Trans. Comput., vol. C-17, pp. 758~770, Aug. 1968.

〔2〕 D.H. Lawrie; "Access and alignment of data in an array processor", IEEE Trans. Comput., vol. C-24, pp. 1145~1155, Dec. 1975.

〔3〕 C. Clos; "A study of nonblocking switching networks", Bell Syst. Tech. J., vol. 32, No. 2, pp. 406~424, 1953.

〔4〕 V.E. Benes; Mathematical theory of connecting networks and telephone traffic control, NY, Academic, 1965.

〔5〕 H.S. Stone; "Parallel processing with the perfect shuffle", IEEE Trans. Comput., vol. C-20, pp. 153~161, Feb. 1971.

〔6〕 H.S. Stone; "Dynamic memories with enhanced data access", IEEE Trans. Comput., vol. C-21, pp. 359~366, Apr. 1972.

〔7〕 T. Lang, "Interconnections between processors and memory modules using the shuffle-exchange network", IEEE Trans. Comput., vol. C-25, No. 5, pp. 496~503, May 1976.

〔8〕 T. Lang and H.S. Stone; "A shuffle-exchange network with simplified control", IEEE Trans. Comput., vol. C-25, pp. 55~65, Jan. 1976.

〔9〕 D.K. Pradhan and K.L. Kodandapani; "A framework for the study of permutation and applications to memory processor interconnection networks", Proc. 1979 Int. Conf. Parallel Processing, pp. 148~158, Aug. 1979,

〔10〕 P. Budnik and D.J. Kuck; "The organization and use of parallel memories", IEEE Trans. Comput., vol. C-20, pp. 1566~1569, Dec. 1971.