A DISTRIBUTED PROCESSING ELEMENT INTERCONNECTION FOR TWO DIMENSIONAL PROCESSOR ARRAYS

Gordon L. DeMuth

IBM Federal Systems Division
Manassas, Virginia 22110

SUMMARY

Parallel or distributed processing has been proposed to obtain high computation rates in signal processing. One of the more significant problems associated with distributed processing is the method of interconnecting the distributed processing elements or cells. The interconnection provides the data transfer between cooperating cells.

Distributed processing elements connected in a two-dimensional arrangement use local, nearest neighbor signal data communication and they distribute processing by both pipelining and paralleling operations. This paper describes an approach for interconnecting the processing elements of a two-dimensional array in a manner that is expandable, partitionable and reconfigurable. The two-dimensional array of processing elements can be divided into autonomous partitions that are assigned independent tasks; if an element fails, the elements in the array can be reconfigured.

RESUME

Le traitement parallèle ou réparti est recommandé dans le but d'obtenir des taux de computation éléves dans le traitement des signaux. L'un des problèmes plus significatifs associés au traitement réparti est la méthode d'interconnecter les éléments de ce traitement ou cellules. L'interconnexion permet de transférer les données entre les cellules contributives.

Les éléments du traitement réparti connectés dans un dispositif bidimensionnel utilisent la communication des données par signaux voisins ou locaux; ils repartissent le traitement par pipelining et des operations parallèles. L'article suivant décrit la méthode proposée pour interconnecter les éléments de traitement d'un ensemble bidimensionnel de façon extensible, compartimenté et reconfigurable. Cet ensemble peut être divisé en partitions autonomes qui reçoivent des tâches indépendentes. Si l'un des éléments est en panne, les autres peuvent être reconfigures.

INTRODUCTION

Parallel processing provides a means of obtaining the high computation capability required for many problems that are now practical using VLSI circuit technology; the preprocessing, segmentation, feature extraction and feature classification of multiple images, each comprised of a large number of pixels, is an example. As the number of parallel processing elements becomes large, the bandwidth required for signal communication can exceed the capabilities of common path interconnections such as the data bus or ring network. Systolic arrays, pipeline networks, etc. have been proposed for these applications to distribute bandwidth and computation requirements (1,2,3). In these arrays of distributed processing elements, global communication is replaced by local or nearest neighbor communication. Figure 1 illustrates a two-dimensional array that has pipelined stages interconnected by switching networks; these switching networks will be referred to as matrix switches and are constructed using crossbar switches with relatively small dimensions.

A broadcast capability exists between distributed processing elements and each distributed processing element contains memory and control in addition to it's arithmetic capability. Each distributed processing element is a SIMD (Single Instruction Multiple Data) processor and the overall two-dimensional array represents a MIMD (Multiple Instruction Multiple Data) processor with distributed memory (4). There is no shared or global memory.

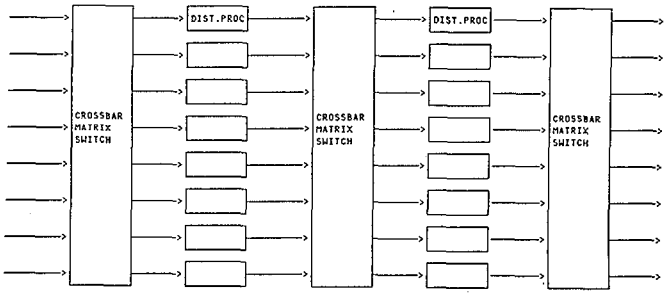


Figure 1. Distributed Processing Elements Interconnected via Matrix Switches

Each processing element (source) can be connected to a subset of processing elements (sinks) in the next column where the subsets are chosen in an overlapping manner. Similarly, each processing element (sink) can be connected to a subset of processing elements (sources) from the preceding column. This overlapping routing capability is referred to as overlapping connectivity windows.

CROSSBAR SWITCHES AND SYSTOLIC ARRAYS

When crossbar switches are used in the customary manner to globally connect distributed processing elements, the crossbar switch becomes both complex and expensive as the number of distributed processing elements increases. An N by N crossbar switch connecting N transmitters to N receivers grows in complexity as a function of N squared. Distributed processing systems, such as systolic arrays, do not require that each processor communicate with all other processors in the system; for these distributed processing systems, crossbar switches can be used to connect local groups of distributed processing elements in a way that is expandable and reconfigurable when components fail. This is the basis for the overlapping window connection network.

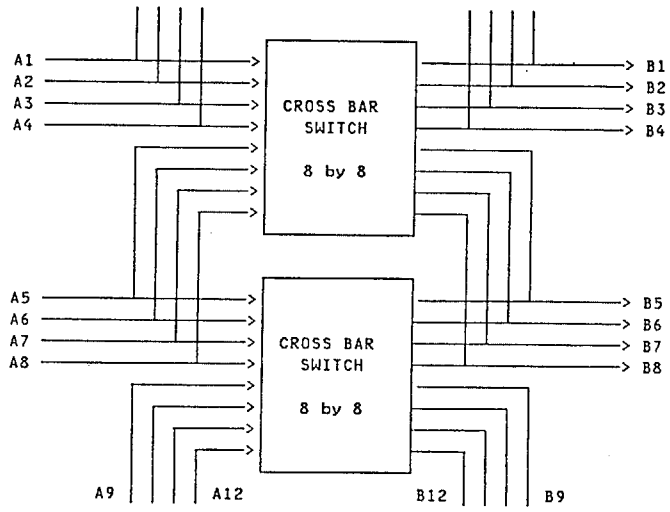
OVERLAPPING CONNECTIVITY WINDOWS

The matrix switch in the overlapping window connection network is configured using crossbar switches of moderate size that are interconnected to provide parallel, redundant paths. This network provides connectivity of each distributed processor to a group of nearby source (sending) distributed processors and to a group of nearby sink (receiving) distributed processors. Overlap in the connectivity field of adjacent processing elements within each column provides routing and assignment flexibility and the ability to configure parallel partitions of distributed processing elements; independent tasks can be carried out autonomously and concurrently in the parallel partitions of different size and structure (5). The redundant matrix switch data paths provide high system availability.

Very large scale integrated circuit (VLSI) technology has sufficient gate density an input-output capability to permit fabricating byte-wide crossbar switches with up to 8 by 8 connection capability. Crossbar switches such as these can be connected using the overlapping window type of network illustrated in Figure 2 Switch inputs from sending distributed processors are connected to several crossbar switches in an overlapping manner; the switch outputs to the receiving distributed processors are connected in a similar overlapping manner.

REDUNDANT, PARALLEL PATHS

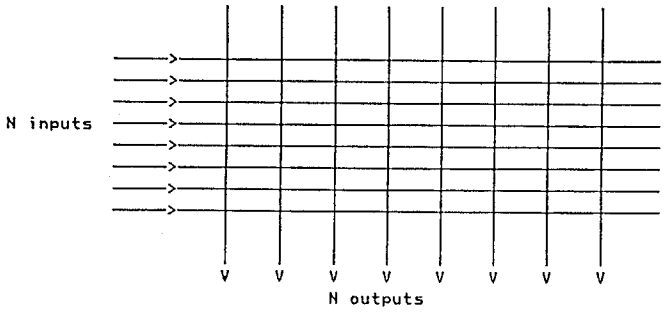
Individual crossbar switches (Figure 3) are programmable to either connect selected crosspoints or to leave all crosspoints unconnected so that an alternate crossbar switchpath can be used instead. Similarly, the crossbar switch is designed to dominantly fail in an unconnected state to permit use of alternate parallel paths in the event of component failure.



N = 8, crossbar dimension
P = 2, number of crossbars connected to a device

Note: Additional crossbar switches are added until the total number of distributed processing elements in the two stages of the systolic array being connected are accommodated. The loop is then closed by connecting these paths of the bottom crossbar switch to the paths of the top crossbar switch. A loop of overlapping connectivity windows is then created in the matrix switch.

Figure 2. A Switching Matrix Interconnection that provides Parallel Connection Paths and Overlapping Connectivity Windows



Note: Each output line is connected to the N input lines by a single pole, (N+1) throw switch; the (N+1)th. Position is "open" for alternate crossbar switch selection.

Figure 3. N by N Crossbar Switch

ORGANIZATION

The overlapping window connection of the crossbar switches implies the connection of each crossbar switch to a window or group of processors that is slightly displaced from the window or group of processors connected to the adjacent crossbar switch. For instance, crossbar switch (k) is connected to processors (n) through (n+7), switch (k+1) is connected to processors (n+4) through (n+11), and switch (k+2) is connected to processors (n+8) through (n+15). For this example, the following parameters apply:

N = 8, where N implies the size of the crossbar switch (N by N).

P = 2, the number of parallel switches connected to a processor.

C = 12, the connectivity of a distributed processor in terms of fan-in and fan-out.

R = 4,8, R is a measure of available path redundancy and indicates the number of adjacent processors connectable with P, P-1, ..., 1 path permutations.

For a given size crossbar switch (N) and a selected degree of parallelism (P), the connectivity (C) and redundancy (R) can be determined by the following equations:

C = N/P (2P-1), where P is greater than 1 and less than or equal to N.

R = N/P, 2N/P, 2N/P,....., indicates the number of adjacent processors accessible by a source/sink with P, P-1, ..., 1 path permutations.

The preceding equations are used in generating the connectivity and redundancy for several possible configurations of the overlapping window data transfer network.

N	P	C	R
8	2	12	4,8
8	4	14	2,4,4,4
8	8	15	1,2,2,2,2,2,2,
6	2	9	3,6
6	3	10	2,4,4
6	6	11	1,2,2,2,2,2
5	5	9	1,2,2,2,2
4	2	6	2,4
4	4	7	1,2,2,2
2	2	3	1,2

Figure 4 illustrates the overlapping window matrix switch for the configuration with N=8, P=2, C=12, and R=4,8. Figure 5 shows how the overlapping window matrix switch can be expanded to accommodate an increase in the number of distributed processing elements in the columns of the two-dimensional array.

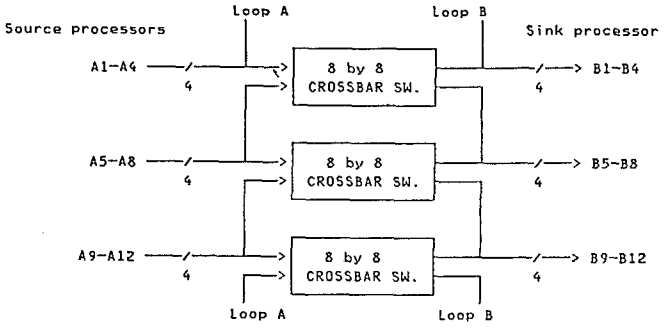


Figure 4. Redundant, Expandable Matrix Switch (12 Source Processors/ 12 Sink Processors)

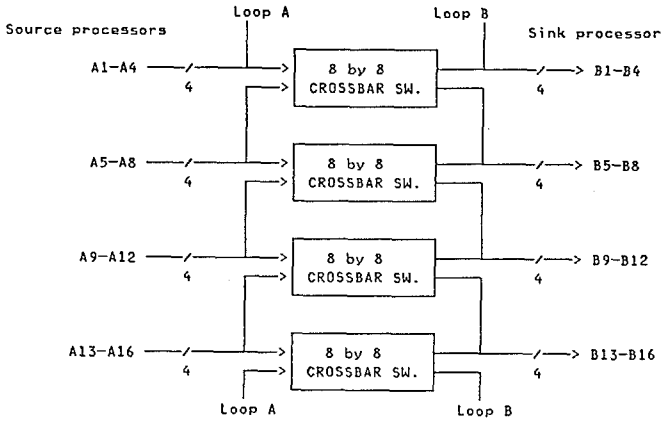


Figure 5. Redundant, Expandable Matrix Switch (16 Source Processors/ Sink Processors)

DIFFERENT NUMBERS OF PROCESSORS IN SUCCESSIVE STAGES

The overlapping window matrix switch is not constrained to processor arrays in which the number of distributed processing elements in successive stages is the same. This situation frequently arises when the processing elements are not of the same type in the succeeding stages of the processor array; the processing elements are homogeneous within a stage but are heterogeneous from stage to stage. The different number of processing elements required for the input to the matrix switch (transmitters) and for the output of the matrix switch (receivers) is accommodated by using a different value of parallelism, P, for the input and output connections.

The following expanded designations will be used for the parameters of the overlapping window matrix switch when the number of distributed processing elements is not the same from stage to stage.

PS = the number of parallel crossbar switches connected to a sending processor.

PR = the number of parallel crossbar switches connected to a receiving processor.

CS = the connectivity or fan-out of a sending processor.

CR = the connectivity or fan-in of a receiving processor.

RS = the path redundancy for the fan-out paths of a sending processor.

RR = the path redundancy for the fan-in paths of a receiving processor.

PM = the minimum value of PS or PR for the overlapping window matrix switch being evaluated; eg. the matrix switch between the (k)th and the (k+1)th pipeline processor stages.

Equations relating the connectivities (CS, CR) and the redundant paths (RS, RR) to the other parameters become:

$CS = (N/PR)(PS+PR-1)$

$CR = (N/PS)(PS+PR-1)$

RS = CS-(PM-1)(2N/PR), number of channels with PM path permutations.

= 2N/PR, number of channels with PM-1, PM-2, ..., 1 path permutations.

RR = CR-(PM-1)(2N/PS), number of channels with PM path permutations.

= 2N/PS, number of channels with PM-1, PM-2, ..., 1 path permutations.

Using these equations, the connectivities and redundancies for several overlapping window matrix switches connecting pipeline processing stages with different numbers of processing elements per stage (PS not equal to PR) are calculated:

N	PS	PR	CS	CR	RS	RR
8	2	4	10	20	6,4	12,8
8	4	2	20	10	12,8	6,4
8	2	8	9	36	7,2	28,8
8	4	8	11	22	5,2,2,2	10,4,4,4
6	2	3	8	12	4,4	6,6
6	2	6	7	21	5,2	15,6

An example for N = 8, PS = 2, and PR = 4 is illustrated in Figure 6.

CONCLUSION

The overlapping window matrix switch can be configured with modest size crossbar switches to interconnect adjacent stages of a two-dimensional array of distributed processing elements and it permits forming parallel partitions that are dedicated to independent processing tasks. The matrix switch is expandable by simply adding additional crossbar switches to accommodate an increased number of distributed processing elements and the processing element interconnection can be changed in the event of component failure within a distributed processing element. The overlapping parallel path structure of the matrix switch results in a reliable interconnection network for the two-dimensional array of distributed processing elements.

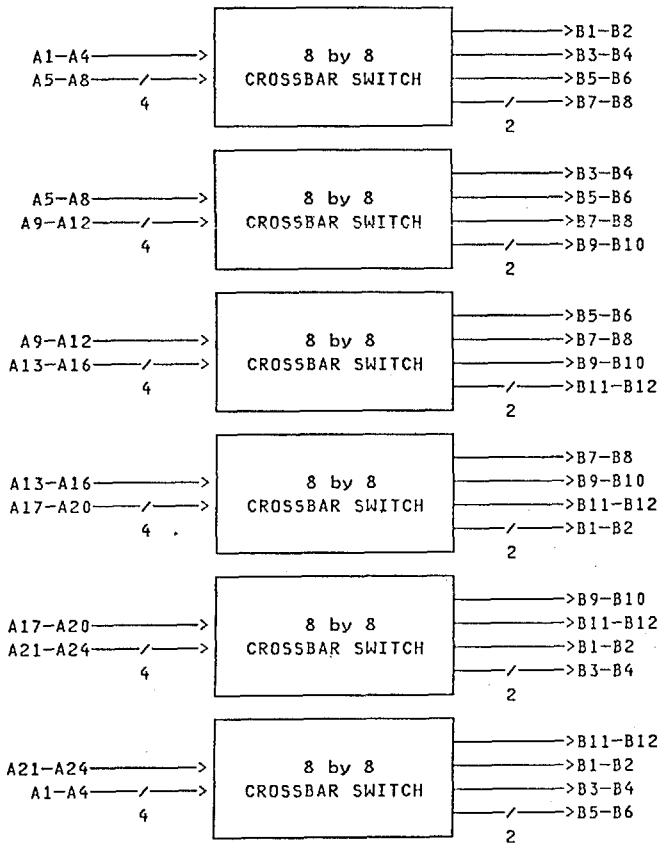


Figure 6. Overlapping Window Matrix Switch When the Number of Sending Processors is Different From the Number of Receiving Processors

REFERENCES

1. H. T. Kung, "Why Systolic Architectures?", IEEE Computer, pp. 37-46, Jan. 1982

2. A. V. Kulkarni and D. W. L. Yen, "Systolic Processing and an Implementation for Signal and Image Processing", IEEE Transactions on Computers, Vol. C-31, No 10, Oct. 1982

3. K. Hwang and Z. Xu, "Multipipeline Networking for Developing Multiprocessor Supercomputers", CRI-85-002, Computer Research Institute, University of Southern California, Los Angeles, July 1985

4. G. L. DeMuth, " A Partitionable Distributed Beamformer and Signal Conditioner", IEEE Proceedings of the 1986 International Conference on Acoustics, Speech and Signal Processing, April 1986

5. H. J. Siegel, L. J. Siegel, F. C. Kemmerer, P. T. Mueller, Jr., H. E. Smalley, Jr., and S. D. Smith, " PASM, A Partitionable SIMD/MIMD System for Image Processing and Pattern Recognition", IEEE Transactions on Computers, Vol. C-30, No. 12, pp. 934 - 947, Dec. 1981