A MASSIVELY PARALLEL PROCESSING SYSTEM
BASED ON A HYPER-CROSSBAR NETWORK

C. Chin and W. Lin
General Electric Company / CRD
Corporate Research and Development
Schenectady, NY 12301

ABSTRACT

Based on the advanced technologies, VLSI and HDI (High Density Interconnection), a parallel processing system consisting of 1024 processors is proposed. A special feature of this system is the reconfigurability of data communication channels between processors, achieved by using a hyper-crossbar interconnection network which facilitates a multiprocessor system to operate as a SIMD, MIMD, MSIMD, ... etc. Each processor possesses two communication channels, separately connected to a local crossbar network and to a global crossbar network (which are sub-networks of the hyper-crossbar network) for local communication and global communication, respectively. Processors connected to the same local network form a processor cluster for the execution of systolic-array-type algorithms. Primarily implemented by LINC chips, the global networks are able to programmably hold or delay operation data to synchronize the data flow for generic applications. With the operation speed of 20 MHz, the system can reach a peak performance of 40 billion operations per second.

Keywords: Computer Architecture, Parallel Processing, Crossbar Network, VLSI, HDI, Reconfigurability.

INTRODUCTION

Due to the computation demands of the modern applications, e.g. image processing, system simulation, real-time graphics display etc., the computation power required has reached billion operations per second or even higher. Given a general VLSI/CMOS operating clock speed, i.e. 10-40 MHz, it becomes necessary to develop a system with over thousand processors to achieve the required performance. Due to the communication demands from thousands of processors, one of the major bottle-necks existing in any multiprocessor system, interconnection communication networks for such systems have been recently focused and heavily studied (Ref. 1-2).

The most flexible and simplest solution is to have every processor connected to a global crossbar network which provides essentially a non-blocking communication link. Because of the technology limitation, this type of network is not feasible for a large scale system containing over thousand processors. In this paper, an alternative approach, i.e. a hyper-crossbar network based parallel processing system, is proposed. Because of the hyper-crossbar network, the proposed general-purpose multiprocessor system can be configured to operate as a SIMD, MIMD, MSIMD, ... etc. Furthermore, fault-tolerance capability can be enhanced by this approach.

SYSTEM ARCHITECTURE

The proposed parallel processing system has 1024 processors, a hyper-crossbar network, and a distributed main memory, as shown in Fig. 1. Processors are addressed by 10 bit binary codes $d_9d_8 \cdots d_1d_0$. Each processor consists of a node-level crossbar network, a processor controller, a multiplier, an ALU, a shifter, and a register file, as shown in Fig. 2. In the system, there are 1024 local memories which are individually attached to 1024 processors. The collection of all local memories forms the distributed main memory of the system. Through the node-level crossbar network, operands for the three operators, i.e. multiplier, ALU, and shifter, are provided by four resources: register file (two channels), operation results from previous clock cycle, local memory, and external (through the hyper-crossbar network). The operation results can be either temporarily saved in the register file, transmitted to other processors, or stored in the local memory.

Selectors in a processor provide an option for a half-word operation, while registers hold operands for a certain period. The controller which controls all operations in the processor receives global commands from the external and selectively
executes commands pre-stored in the local memory. The local memory is partitioned into two banks to facilitate the memory sharing and updating process. While one bank is being used to serve for the processor control, the other can be updated or read by other processors through the node-level crossbar network at the same time. There are no duplicated copies for information stored in the main memory to avoid memory incoherence problems (Ref. 2).

![Figure 2. The architecture of a node processor](image)

There is a hyper-crossbar interconnection network connecting all processors together. Instead of using a tightly-coupled connection, the hyper-crossbar network is an accumulation of many individual crossbar networks. The networks are partitioned into two groups, i.e. local network and global network. Processors with the same four leading address code d_4d_3d_2d_1 are connected to a 64-by-64 local crossbar network. Those processors which are connected to the same local crossbar network form a processor cluster. Processors with the same six trailing address code d_5d_4d_3d_2d_1d_0 are also connected together through a global 16-by-16 network which consists of 4 sets of LINC chips (Ref. 3) as shown in Fig. 3. Each set consists of four/eight LINC chips operating in parallel in order to provide 16-bit/32-bit wide communication channels. All the 16 local crossbar networks and 64 global networks are controlled by a host. The host determines the connection patterns, depending on the data flow specified in parallel computation algorithms.

The architecture of LINC chip is shown in Fig. 4. The chip has eight 4-bit data-paths consisting of an 8-by-8 crossbar network, either a FIFO or a programmable delay register for each of its inputs, and a pipeline register file for each of its outputs. The connection pattern between sources and destinations (including broadcasting) is determined by the control pattern register which is updated by the preloaded control pattern memory. Wider data channels can be achieved by combining more LINC chips together, while using the same control signals. With the reconfigurability and programmability, provided by LINC chips, data flow through a global network can be synchronized and redistributed.

![Figure 3. The hyper-crossbar network architecture](image)

![Figure 4. The LINC chip block diagram](image)

**IMPLEMENTATION CONSIDERATION**

The area complexity of a node processor has been studied. Using a 1.2 μm CMOS technology, a node processor (Local memory is not included) occupies an area of 340-by-340 mil² (Ref. 4). In other words, a 4-inch wafer can accommodate more than 80 node processors. Thus, it is sufficient to include one entire processor cluster on a single wafer. The associated 64-by-64 local crossbar network may also be incorporated on the same wafer to interconnect the node processors as shown in Fig. 5. By using the strategy of device redundancy, it is reasonable to assume that the network is fault free.

The area required by a crossbar network is due to switching circuits and data wiring. Although the area complexity of the crossbar switching circuits is known to be O(N²), where N is the number of network terminals, the actual area required for the circuits is almost negligible (only N transistors per channel as shown in Fig. 5). This is particularly true when the switching circuits are compared with 100 thousand transistors in a node processor. It is estimated that the wire routing area required for the proposed crossbar network is 2W·B·N·S, where W, B, N, and S are the pitch...
width, the channel bits, the network terminals, and the processor perimeter length, respectively. For the case of W = 4 um, B = 16 bits, N = 64, and S = 4 mm, the routing area is about 16 mm², which is much less than the 64-processor area, i.e. 64x4² mm². It can be also shown that the multi-stage interconnection network requires 2W·B·S·N·((logN)-1) routing area on a silicon. It is larger than the area of a crossbar network by a factor of log N.

Other implementation alternative is to use the high density interconnection (HDI) packaging technology (Ref. 5). HDI connects signal I/O pins between bare chips (unpackaged chips) with copper wires by using laser-patterned polymer layer overlays laminated the chips mounted on a silicon substrate. By using VLSI technology in conjunction with HDI, a very reliable processor cluster embedded in a distributed 64-by-64 crossbar network (as shown in Fig. 5) can be easily implemented in a package without any wire routing problem. Similarly a global network could also be realized as a network device. In this way, the system volume is reduced, and the system structure is modularized to facilitate the system expansion.

SYSTEM APPLICATIONS

In general, the system is designed as a general purpose machine. The data links between processors can be configured into almost any possible pattern for computation-intensive applications.

Image Processing

By programming the connection pattern in the hyper-crossbar network, the proposed system can be configured as a array-type multi-processor system to process matrix operation or pixel computation for image processing applications. In Fig. 6, an image processing algorithm is partitioned and distributed into different processor clusters (MSIMD). Clusters are pipelined together through global networks to simplify instruction flow and to enhance data throughput. The local network in a cluster can be programmed not only to emulate a mesh-connected network for neighborhood communication, but also to provide direct communication channels for global interactions within a cluster. Intermediate results, generated by each pipeline segment (i.e. processor cluster), can be re-directed, broadcasted, or delayed to maintain a synchronized operand flow by the global networks.

Parallelized Looping

Besides the mesh connected pattern, the processors can be configured as a multi-channel pipeline to compute multi-level loop instructions in parallel as shown in Fig. 7. The local networks provide communication links to pipeline the inner loop instructions, while the global networks support

![Figure 5. An implementation of a distributed local crossbar network](image)

![Figure 6. A pipeline configuration for image processing](image)

![Figure 7. A multi-channel pipeline to execute multi-level loop instructions](image)
data communication between different loops. Because of the programmable delay registers (PDR) and the pipeline register files equipped in LINC, which is the fundamental building block of the global networks, data synchronization and dependency problems can now be easily resolved. Other interconnection patterns or any combination of different patterns may also be established to execute MIMD, MSIMD, or other generic operations.

**Object Domain**  
**Space Domain**

![Diagram of Object Domain and Space Domain](image)

3-D Graphics Display System

An advanced real-time 3-D graphics display system has to manipulate numerous 3-D image objects in a display window. It usually requires a computation capability in the order of 10 billion instructions per second. Using parallel processing, as provided by the proposed system, is the only solution to enhance computation strength for satisfying the requirement. A special space partition algorithm (Ref. 6) to efficiently use multiprocessor elements for real-time graphics display is developed and shown in Fig. 8. The algorithm requires a flexible communication network to support its adjustable space partition and processor assignment, described in Fig. 9. Given the hyper-crossbar network of the proposed system, a flexible environment can be established to partition image space in various topologies which are required by high performance 3-D graphics display applications such as pilot training and mission rehearsal.

**Performance Evaluation**

The performance of the proposed system is evaluated in two ways, computation capability and communication capability. Operating at the speed of 20 MHz, each processor can perform 40 million operations per second. In total, the system can provide the peak performance of 40 billion operations in every second. It satisfies the speed requirement for most modern applications. The network capacity is determined by the number of channels, the channel width, and the network speed. Given a regular CMOS operating speed, i.e. 20 MHz, a 64-by-64 local crossbar network with 32 bits wide is able to transmit 40 billion bits in a second. In other words, each processor can transmit or receive over 320 million bits per second. Similarly, it can be easily estimated that a global network provides each processor with the same communication capability.

**Conclusion Remarks**

Operating at the speed of 20 MHz, the system can reach the peak performance of 40 billion operations per second. Because of the reconfigurability provided by the hyper-crossbar network, the average performance can be easily optimized to approach this peak performance. Systems with more processors can be achieved by either expanding the network size or introducing higher levels of crossbar networks. Furthermore, failed processors can be dynamically bypassed, without interrupting the system service.

**References**