PARALLEL STRUCTURES FOR EFFICIENT AND RELIABLE GENERAL PURPOSE COMPUTING

A. Blum

Advanced Technology, IBM Boeblingen Laboratory

Abstract

A general purpose parallel processing structure has been prototyped at the IBM Boeblingen Laboratory, facilitating an "any-to-any" intercommunication network for interconnecting up to 64 general purpose processors and their memories, based on a high-speed crosspoint switch. This prototype is operational at CERN and is used for real problem processing as well as for further application software development. The paper gives a short description of the hardware and software implementation of this IBM-CERN parallel processing prototype.

Furthermore, advances in the field of the interconnection related technology will be also described. This technology provides for further improving the communication among the Processing Elements as well as to the I/O resources, allowing for more efficient parallel software structures because of a reduced access time difference for accessing local and remote memories. Considering this enhanced hardware platform, software and hardware related designs for improved reliability will be also mentioned.

Introduction

Many parallel processing architectures have been invented during the last two decades and several comprehensive survey and tutorial papers have been published since, describing a variety of system hardware structures for concurrent program execution as well as parallel processing software operating systems suited for adaptation of parallel application programming paradigms [1, 2, 3, 4].

It is commonly understood that some specific application areas can be efficiently supported by certain specialized hardware architectures belonging, for example, to the class of Single Instruction, Multiple Data stream (SIMD) architectures such as the processor array arrangement, or to the class of systolic architectures.

Presently, most of the parallel processing architectures in practical applications are aimed at special purpose computations. Examples of such system implementations are all kinds of systolic arrays of special purpose processors, like Transputers, etc., employed basically for Image Processing. Another example is given by tree-like interconnection structures of processing elements, mainly aimed at database management applications. These types of interconnection structures provide for high speed communication only in preferred directions or only between certain locations within the system's topology. This, in turn, provides for sufficient intercommunication bandwidth only for specific applications, prohibiting a more general use of those parallel systems. Therefore, today's general purpose parallel processing systems are basically arranged around conventional types of system buses or hierarchies of system buses. Single system buses within a structure of many processing elements constitute, however, communication bottlenecks, because of their blocking nature for concurrent transfers.

The goal of the joint IBM-CERN prototype project, however, was to implement a general purpose concurrent processing platform capable to support a large variety of parallel processing applications [5]. Therefore, this paper restricts itself to discussing only a specific realization of a Multiple Instruction, Multiple Data string (MIMD) general purpose multicomputer which was delivered to CERN in 1990 [5]. The experience gained in the
course of the hardware and software design as well as during the application adaptation and performance evaluation of this Parallel Processing System (PPS) was of tremendous value for defining further enhancements of a general purpose parallel processing environment.

Structure of the Parallel Processing System - PPS

An example system can be envisaged, consisting of up to 64 microprocessors, for example of the type used in the IBM 9370 systems, each processor having its own local main memory. In the IBM-CERN prototype only one processor - the host - is equipped with its local I/O. However, all other processors could be provided with their local I/O as well, comprising the individual computers as the processing nodes in the more general sense.

The processing nodes are assumed to be interconnected by means of a high speed communication system based on a crosspoint switch hardware (Fig. 1) with 64 ports allowing 32 processing node pairs to intercommunicate simultaneously, in an any-to-any fashion [6, 7]. Fig. 2 shows the topology of a hardware path for exchanging information between two arbitrary processing units PU1 and PU2 via the crosspoint switch. Due to the packaging constraints of the chosen standard IBM 9370 computers, the crosspoint switch is interconnected to the memory bus of a particular IBM 9370 computer by means of a switch port adapter and an adapter to the PU-memory bus. The latter

---

Figure 1. Crosspoint switch interconnect structure of the PPS b), versus a central system bus a).

Figure 2. Intercommunications Hardware Topology. (Two Computers shown)
provides for synchronization of the individual PU clocks with the central clock of the switch complex.

A simple information exchange protocol via the crosspoint switch could be seen as exchanging short messages for internode control passing, such as for starting program execution in remote processing nodes or for program execution synchronization of programs running in remote nodes.

A typical message operation is composed of 3 basic parts: the operation initiation phase (a), the data transfer phase (b) and the operation end phase (c). The message operation between a pair of processing nodes can be significantly optimized by implementing a multilateral hardware control facility which is very efficient within the close locality of a multicomputer arrangement interconnected via a switch [8]. By passing the message operation control back and forth between the sending processing node, the switch control logic and the receiving processing node allows for significant optimization of the message information transfer latency and data rate. The latency can be improved even further when applying concepts of direct notification of the switch control logic with status information such as "ready"/"busy" or "expected ready" status of the respective processing nodes as proposed in [9]. All this indicates that the assumption made so often in the literature that a crosspoint switch is supposed to be too restricted in its efficiency is worth reconsidering because of the latest technological enhancements achieved so far and yet to come.

Conceptual generalization

It appears advantageous to complement a message intercommunication means between the processing nodes with a direct memory access facility between the local main memories of the communicating nodes. Such a direct memory access (DMA) operation can be quite efficiently implemented in hardware by basically starting a message operation with memory address setup within the initiation phase (a), and extending the scope of the data transfer phase (b), now carrying the DMA data used as transferred information files for exchanging large volumes of data and programs between the processing nodes. Yet further intercommunication enhancements will be discussed later such as introducing direct DMA operations between the local memory of a processing node and a DASD of a DASD pool accessible from all processing nodes, for example, via a second additional crosspoint switch connected to all processing nodes within the multicomputer system.

Using the described system structure the following Multiple Instruction, Multiple Data stream concurrent program execution schemes can be incorporated.

1. Distributed Local Main memories with shared distributed Global memory and distributed local DASD with distributed Extended Electronic memories (DASD - caches) (Fig. 3).

2. Distributed Local Main memories with shared centralized Global memory and shared distributed DASD (Fig. 4).

By extending the number of switch ports (or if providing a second crosspoint switch) for attaching the DASD (Fig. 3), both of the above schemes as well as combinations of them could be incorporated concurrently within a single multicomputer system with the capability of assigning the processing and DASD resources dynamically.

The sharing capability can be also extended to other I/O in a similar way as indicated for DASD. This allows the introduction of novel fault-
tolerance schemes into a local multicomputer system. Furthermore, virtually all other interconnect schemes, such as multiple buses, trees, hypercubes, systolies, and so on, can be mapped onto the crosspoint switch interconnect arrangement, basically by software emulation alone. This explains why a crosspoint switch, interconnecting multiple computers, or parts of a computer, such as I/O or memory, allows the execution of not only general purpose parallel software but also specialized parallel software, such as programs developed for systolic architectures, on a multicomputer system, provided the latency and the data rate of the underlying hardware switch is sufficient. Both can be even improved as the technology base will definitely be advanced with time.

Another important aspect of a multicomputer system is its ease of operation by a central system's operator with regard to software as well as hardware. Here, much can be learned from existing systems of concurrent and distributed processing such as Tandem, Stratus, or IBM's System Network Architecture (SNA) with the NetView software functions. The latter functions provide the means for the multitude of interconnected computers of the multicomputer system to appear as a "single system" to a centrally located system's human operator, allowing for initial program load as well as software monitoring of all individual computers from a single focal point.

More serious is the challenge of also maintaining a multicomputer "single system" image with respect to hardware initialization and diagnostics, especially if the individual processors are based on the LSSD-Design for fabrication testing and diagnostics and are conventionally monitored by dedicated service processors. By further enhancing the crosspoint switch control hardware, it becomes possible to perform LSSD-based shift-clock paced information shifting utilizing the switch hardware media for the initialization and diagnostics of the individual microprocessors according to the LSSD-related diagnostic concepts [10]. This allows to use a single centralized hardware-oriented service capability mapped on a separate processor or even on one of the regular multicomputer system processors. For fault-tolerance reasons, this centralized hardware service capability can be physically duplicated or triplicated while keeping a logical singularity with a possibility of transferring this service capability to different member computers for increased availability as well as for flexibility of operation or convenience reasons. It appears to be reasonable to integrate the system related as well as hardware initialization and diagnostic related information transfer functions on the same crosspoint switch media instead of separating them on two different transfer means, for example, if introducing a separate system diagnostic bus. This arrangement would allow even for diagnostics information transfer redundancy, at the same time avoiding a single point of error in the communication media if duplicating the crosspoint switch.

Further enhancements based on technology advances on chip and packaging level

There has been a considerable amount of research and development work conducted in the field of concurrently executing system software and application-related software. It is evident that the evolved concepts of parallel processing software are quite extensively influenced by the underlying hardware system, especially the communication structure. Several MIMD parallel processing systems such as Tandem's Guardian 90 are based on executing concurrently such entities of work as processes, individual transactions, or even tasks and jobs, more or less independently from each other. The parallel VM, a modification of the standard IBM VM, allows concurrent execution of parallel
Fortran applications on the joint IBM-CERN prototype. In a multicomputer with distributed local DASD, effective processing, load balancing and database partitioning are not trivial problems requiring quite extensive exchange of control information and/or data files via the interconnect. In a multicomputer system with a distributed but shared DASD pool the database distribution can be avoided. On the other hand, in such a system data access concurrency and coherence of buffered data have to be dealt with. This becomes even more complicated and may influence the system's overall processing performance, if information logging procedures for error recovery are overlaid for dealing with eventual error events. However, all this can be improved by providing special hardware means associated with the switch for supporting the related software-driven mechanisms. Examples for such hardware functions are locking support for shared data accessing or specific group-broadcast messages via the switch for buffer invalidation, as part of the data coherence control. Such conceivable specialized hardware support is becoming quite easy to be implemented not only because of reduced hardware costs and the virtually exploding amount of available circuits on a single VLSI chip.

Programmable logic gate arrays allow also for hardware design flexibility and function adaptability because of the possibility of programmable changes during the design phase. In general, it is worth mentioning, that even more efficient software concepts can be introduced when drastically increasing the speed and functional diversity of the intercomputer communication means within the multiprocessor complex.

This is suggested due to the expectation of further advances not only in the area of hardware circuit density on a single chip, but even to a greater extent, due to the advent of hybrid wafer scale integration (H-WSI) which will allow, firstly - for a much more closely packaging of the communicating processing and memory elements providing faster signal propagation, secondly - due to utilization of micromachining interconnection fabrication techniques for providing a drastic increase of the intra as well as inter wafer interconnection density [11, 12], thereby improving the communications data rate and simplifying the hardware design, despite increased functional complexity, and thirdly - due to introduction of WSI directed micromachined cooling techniques allowing to use faster high-power driver circuits for speeding up the intercommunication nets [13, 14, 15]. Such active cooling can be achieved by circulating coolant liquid by means of tiny electrostatic pump elements.

In order to improve, for example, the active liquid cooling with micromachined pump elements as suggested in [13], an increase in the pumping pressure of the coolant can be obtained by cascading a multitude of the electrostatic micropump elements with decreasing pump chamber volumes (Fig. 5a). If several hundred such miniaturized electrostatic micropump elements are cascaded, coolant pressures are obtained which are sufficient to operate low-temperature miniaturized "machineless" compressors, integrated in silicon based packaging [16]. This allows the heat to be Dissipated into the ambient environment also by specially designed facilities ("radiators") which are separated from the electronic circuit cluster and therefore are not exposed to such stringent miniaturization requirements as the performance critical integrated circuit cluster itself. The miniaturized compressors can be efficiently utilized for constructing, for example, Joule-Thompson type directed cooling facilities into the electronic chip packaging (Fig. 5b).

A survey of state of the art miniature cryo-cooler technology, lending itself, with modifications, also

![Figure 5. Integrated compression cascade a) and the machineless miniature cooler b).](image-url)
to computer chip packaging, is given in [17]. However, exploiting micromachining processes in silicon, more reliable "machineless" cooling techniques, avoiding mechanical wearing, should have preference.

Further miniaturized machineless cooling means, potentially constructable in silicon, can be based on the temperature separation effect of the Hartmann-Sprenger (HIS) acoustic resonance tube [18]. Detailed investigations of "machineless" gas-dynamic cooling facilities are described in [19]. However, an enhancement of such cryo-cooling apparatus is possible by additionally absorbing and removing the concentrated dynamic energy of the HIS resonance tube while converting it into electrical energy using miniaturized converters such as Piezo-elements located at the end of the resonance tube [20].

Additional intercommunication speed advantage can be gained from incorporating also ECL, or GaAs into the hybrid WSI [13]. Due to these general advances in technology, single stage cross-point switches with up to 512 ports and more can be expected to be realized cost-efficiently also for commercial applications in the foreseeable future. Because of all the conceivable functional and speed advantages, the crosspoint switch is obviously a very attractive means of interconnection for parallel processing, despite a somewhat negative reputation in the past, mainly due to its hardware costs. The evolving technology allows the further enrichment of the functionality of the crosspoint switch by such functions as immediate communication means between processor caches of different computers, between architectured processor internal registers as well as direct information transfer of information (data and instruction opcode) accessed from remote processor caches with latencies, comparable to those of local cache accesses. This opens the ground for the invention and incorporation of novel parallel software concepts with distributed global processor cache buffers complementing the very attractive solutions with distributed global main memories. By directly mapping process signaling and interrupt schemes onto the switch based hardware, the intercomputer synchronization speed can be also drastically improved. Taking into account the discussed technology enhancements, a physical embodiment of a local multicomputer system is conceivable as depicted in Fig. 6., where the overall processing electronics are concentrated for speed optimization as much as the packaging technology allows, even if the cooling means as well as the bulky data media (DASD) have to be separated from the processing electronics.

Summary and Conclusion

It has been outlined that with the introduction of an efficient multicomputer interconnect based on a flexible and fast crosspoint switch hardware means, the general purpose parallel (concurrent) computing is likely to experience additional impetus in the future. Furthermore, it has been shown that with forthcoming advances in chip and WSI packaging and interconnection as well in WSI cooling tech-
nology, the crosspoint switch based intercomputer communication means will gain drastic improvements in speed and functionality, enabling significant parallel processing software advances, further improving the applicational attractiveness of MIMD general purpose parallel computing.

Due to the miniaturization of the electronics complex not only a gain in performance and functionality can be achieved, such compact electronics can be packaged as a Parallel Processing Personal Computer (PPPC), introducing a new era of personal powerful workstations at reasonable costs.

In order to make the broad scale use of a performance efficient application oriented VLSI design a success, the introduction of enhanced miniature high density packaging technology has to be achieved. Within such a technological environment, however, a total system view during the design phase is important. In addition to improving the logic design tools, the electrical, mechanical and thermal properties of the VLSI oriented packaging have to be also considered and reflected in related design and simulation tools.

Recent encouraging results of microstructure research and modelling give reason to believe that the ultra-dense interconnection technology will become available for practical WSI designs in the near future.

Acknowledgement

I would like to express my sincere gratitude to Prof. Dr. W.G. Spruth, head of IBM Boeblingen Advanced Technology and initiator of IBM Boeblingen parallel processing efforts, for his contributions to this paper. Particular thanks are due also to the many colleagues from the IBM Boeblingen Laboratory and from other IBM locations, who have participated in and contributed to the joint IBM-CERN parallel processing prototype and other concurrent processing research projects, for their readiness to support this paper and for providing research material.

References


