

# THE HISTORY OF PARALLEL PROCESSING AT BURROUGHS

Richard Stokes and Robert Cantarella  
Federal and Special Systems Group  
Burroughs Corporation  
Paoli, PA 19301

## Introduction

Parallel processing in the context of the Burroughs experience has been synonymous with the development of the "supercomputer". While it is accurate to claim that, throughout the Burroughs standard product line, the application of parallel processing design is in ample evidence, the main stream of the work on supercomputers is centered in the Federal and Special Systems Group, Paoli, Pa. For almost two decades, the challenge of the parallel machine has been actively pursued without interruption. In that time a series of major systems have been developed, starting with ILLIAC IV, then PEPE, followed by BSP; and this paper describes the historical events in the development of these systems. A new parallel design currently under study for NASA called the Flow Model Processor (FMP) is not discussed here.

These machines as a group represent some of the most ambitious undertakings in the industry (Table 1). With the exception of the FMP, all have been completed in a fully working sense, and all substantially met their original design objectives.

As a group they are certainly a tribute to the designers whose skills harnessed enormous quantities of logic and memory circuits in concerted processing functions. Their contribution to computer science has been made, but perhaps not fully realized. The design rationale of these machines as a machine class (SIMD) provides the only demonstrable performance response for that class of large scientific applications that have vectorizable programs.

This 19-year history is intended as a synopsis of the plans, events and results of three major engineering experiences at the Burroughs Great Valley Laboratories. Unfortunately history, like art, is seen through the mind of the beholder and where serious omissions or errors occur they are certainly not intentional. The lessons learned and the experience derived from these endeavors are continuing to serve our engineering staff in the development of the FMP.

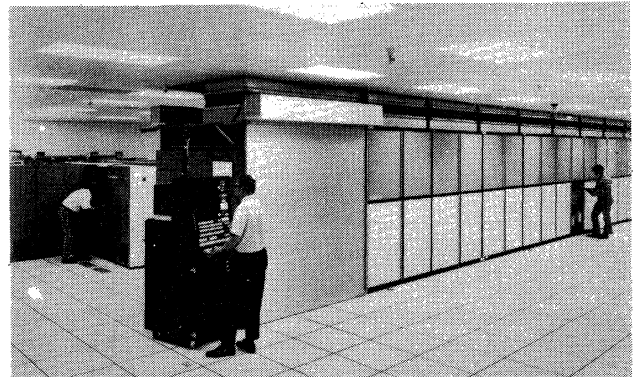
Table 1. Comparison of Parallel Processor Capabilities

	PEPE	ILLIAC IV	BSP
Data Word Size	32 bits	64 bits	48 bits
Instruction Word Size	32 bits	32 bits	24-48 bits
Backing Store	In host	Paged to PE	N-Mos RAM
Memory Cycle	100 ns	250 ns	160 ns
Number of Processing Elements	Up to 288	64	16
Processing Element	32-bit floating point accumulator oriented	66-bit floating point accumulator oriented	48-bit floating point memory oriented.
Microprogrammed	Yes	Yes	Yes
Processing Element Connections	Linear array	4 nearest neighbors	Cross Bar
Parallel Operation Within Arithmetic Unit	Yes	Yes	Yes
Associative Addressing	Yes	Pseudo	No
High Order Language	PFOR	GLYPNIR	FORTTRAN
Processing Speed			
Add	300 ns <sup>1</sup>	500 ns <sup>1,2</sup>	160
Multiply	1.9 us <sup>1</sup>	700 ns <sup>1,2</sup>	320

1. Time for one PE; all PEs may operate in parallel
2. Two operations may complete in this time
3. May be computed as  $N^2$  times 0.85 s, where each operand is assumed to consist of N bits.

## ILLIAC IV

The ILLIAC IV computer was a product of the mid-sixties, its original goals reflecting the prevailing optimism in the country and particularly in the young computer industry. It was the era of the "main frame houses" that continued to demonstrate Groche's Law with regular ease.



Illiac IV Installed at NASA Ames Research Center,  
Mountain View, California

The seeds of the ILLIAC IV program evolved from a project called Solomon developed at the Westinghouse Corporation in Baltimore, Maryland. The circumstance that marked the official beginning of the ILLIAC IV program was the move by Dr. Daniel Slotnick, a Solomon principal, from Westinghouse to the University of Illinois and the subsequent designation of that institution as the prime contractor by the Advanced Research Projects Agency of the Department of Defense.

The program plan was to have the University develop the system software and subcontract the hardware development on the basis of a competitive proposal. Study definition contracts awarded to Burroughs, Control Data Corporation and RCA resulted in three proposals in which Burroughs was awarded the hardware development contract in 1967.

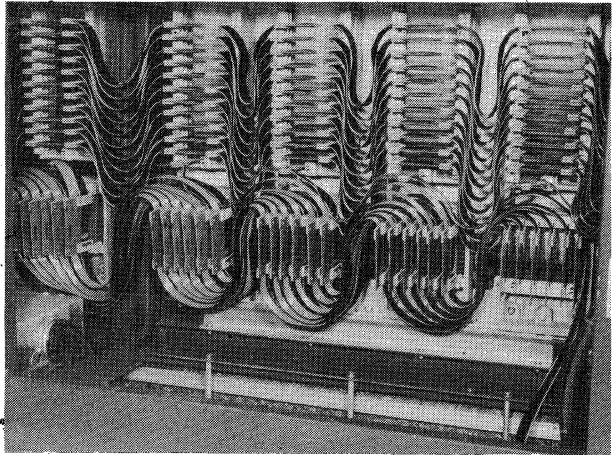
The central objective of the system was  $10^9$  operations per second. This, of course, placed considerable emphasis on hardware component speeds and parallel architectural design [1]. The proposed system contained 4 independent quadrants of 64 Processing Elements (PE) each, for a total of 256 PE's. Each PE contained an arithmetic element and a data memory and was interconnected to other PE's which were a distance of  $\pm 8$  and  $\pm 1$  in designated value. Thus in a  $8 \times 8$  array, a nearest neighbor connection pattern was realized.

Each quadrant was driven by a Control Unit decoding a single instruction stream and broadcasting the microstep for array instruction execution. The Control Unit has a program memory and a separate station for executing CV instructions concurrently with array instruction. ILLIAC IV was a classical SIMD design.

### The Hardware

The key components of the system design were: plainer thin film memories and multichip ECL logic circuit packages. Later events were to show that both choices were not realizable in the final system.

Thin film memories had been in development in Burroughs and elsewhere for several years prior to the start of ILLIAC IV. Thin film was considered the performance successor technology to magnetic cores and Burroughs was actively engaged in the process of moving this technology from the laboratory into production. Two factors conspired to preclude this expectation before production was realized: the tenacity of magnetic



ILLIAC IV Backplane

cores and the pace of semiconductor memories. When this situation became apparent, thin films were discontinued, as a product and, in turn, for ILLIAC IV.

Upon the demise of thin film memory at Burroughs, a contract was awarded to Fairchild Semiconductor for the PE memory system using a 64-bit bipolar component. This contract was one of the more successful projects of ILLIAC IV, calling for the design and production of 70 memory units, each with a capacity of 4K words. Considering the tight schedule and the new technology, many things that might have gone wrong did not: the memories were delivered on schedule and to specification.

The total capacity of 250K words, limited by cabinet volume, was a performance disadvantage for the growing application programs that were run on the system.

As part of the Burroughs proposal, Texas Instrument Corporation, acting as a subcontractor to Burroughs, agreed to provide the Processing Elements (PE) of the system, fully assembled and tested. A PE was a 64-bit floating point arithmetic [2]. The design was based upon a multichip package in which four (up chips) were mounted on a common substrate and interconnected by wire bonding. The circuit packages, 24-pin ceramic, were to be connected on a multilayer printed circuit board, one per PE.

The published reason for the termination of the multichip development by the contractor was low production yield. The design process contained the fundamental weakness of the multichip approach by postponing testing to a complexity level not justified by the value added and not repairable.

The fall-back position was the use of the more

conventional 14-pin DIP packaged ECL on smaller, 2-signal-layer, printed circuit boards, connected by a wired backplane. The logic circuits used were the TI2500 circuit family, implying that the fault of the initial design was the package scheme.

The foregoing component problems were the major ones and contributed to schedule delays and cost increases for redesign. In time, the program scope had to be reduced from four to one quadrant (256 PE's to 64 PE's) where the  $10^9$  operations per second would not be possible.

### The Software

The system software development was the responsibility of the University of Illinois, which undertook the development of a new Algol-like compiler called TRANQUIL [3]. In addition, an assembly language development called GLYPNIR [4] commenced at about the same time.

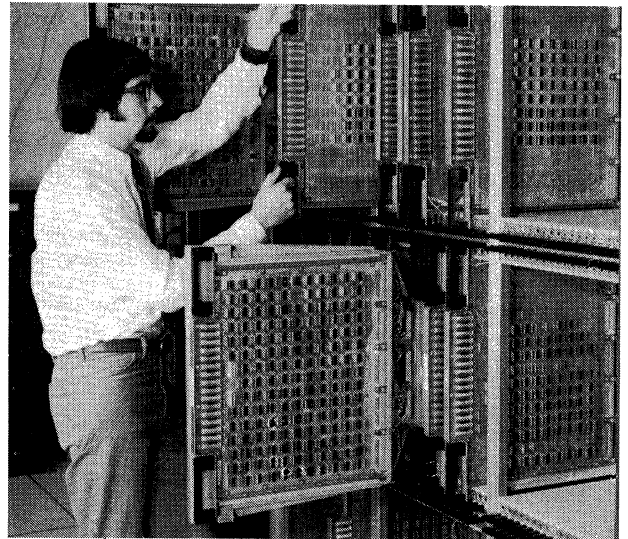
TRANQUIL was, of course, a major undertaking dealing with a parallel structure unlike any previous experience in compiler design. It contained language extensions to allow the users to identify parallel (vector) constructs and to manage the conditional states of the PE array. A preliminary version of TRANQUIL was completed and compared against the available GLYPNIR for object code performance.

The results were disappointing but not necessarily unreasonable for the early stage of the compiler. TRANQUIL, however, was discontinued and GLYPNIR became the principal language for programming ILLIAC IV. Later, after the system was installed at NASA Ames, another language emerged called CFDL (Computational Fluid Dynamic Language). CFDL was based on Fortran and supported the principal applications for that agency.

### The Completion

The ILLIAC IV system was shipped to NASA Ames in April 1972 and was accepted by the customer that December. The selection of the NASA site in lieu of the original one at the University of Illinois was due in part to the campus unrest of that era and the possible target the system presented. The system has been operational now for almost a decade and is considered an effective and productive resource in the mission of that agency.

To the people who designed and built the ILLIAC IV, it was certainly a triumph of skill and



ILLIAC IV Control Unit

determination. The size and complexity of the system (250 thousand, dual, in-line components) is a challenge by today's standard. ILLIAC IV also made its contribution to the science:

- a) It demonstrated that a SIMD architecture could be used effectively on some important applications.
- b) It showed that a system of that size and complexity could be used productively and reliably.
- c) It made the user community "vector conscious" and motivated the work toward vectorizing compilers and the inclusion of vector operations in later product designs.

A major drawback to a wider use of ILLIAC IV was the evolution in user environment. Modern compilers and operating systems removed the user from the hardware details of programming. The programming pioneering days were coming to a close.

### PEPE (Parallel Element Processing Ensemble)

The history of PEPE development discloses a number of different corporations that contributed in varying measure to the final delivered product. PEPE as an architectural concept began in the mid-sixties at Bell Laboratories, New Jersey, under the auspices of the Army Ballistic Missile Defense Agency (ABMDA). An early prototype was assembled there at the time AT&T decided to divest itself of military development contracts. As a result, the System Development Corporation

took charge of PEPE and, in turn, engaged Honeywell in support of the hardware design.

In March, 1973 Burroughs was awarded a contract by SDC to build a revised and enhanced version of PEPE for ABMDA, Huntsville, Alabama. The system Burroughs was contracted to build was specified in detail, focusing primarily on the problem of radar data processing for missile defense systems.

The execution of the contract by Burroughs is considered an industry paragon and Burroughs was singled out for an outstanding performance award by the U.S. Army for this achievement. The completed PEPE system was shipped from Burroughs Great Valley Laboratories, Paoli, Pa. to Huntsville in May 1970 and accepted by the customer by November of that year. The only significant change from the original contract was the reduction of the number of processing elements from 36 to 11 due to a program funding reduction.

### The Design

The PEPE design is considered special purpose because it is driven by the single application of radar target correlation and tracking. This application naturally lends itself to parallel processing since the processing functions are identical for multiple target returns and predictions. The PEPE is really three distinct linear arrays, each of which performs the parallel functions of correlation, tracking, and radar control, respectively. A Processing Element is a single orthogonal slice of these hardware elements, including a common memory and incorporating each of the three functions.

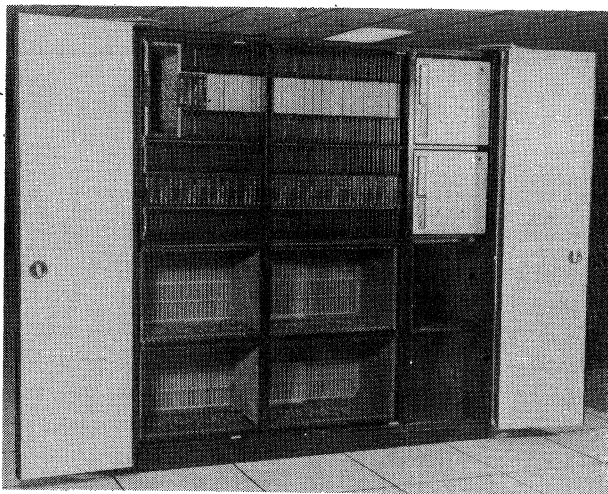
Another important aspect of the PEPE application is that there is no requirement for inter-PE communication. This permits the PE's to associate in a loosely coupled "ensemble," with a significant reliability advantage as a result. Multiple failures in PE would degrade but not fail the system. The system was packaged with 36 PEs in a cabinet and a maximum of 288 PEs was permitted.

The logic component family used in PEPE was the Motorola 10K ECL Family. MECL 10K was a mix of MSI and SSI completely packaged in ceramic DIPs. The memory was a 1K bipolar RAM produced by Fairchild Inc. The novel design of the printed circuit boards featured a combination of printed wiring and wrapped post wiring that avoided the problems of multilayer boards. This design, called the composite board, was used successfully on the BSP.

### The Epilog

The PEPE system was interfaced with a CDC 7600 host system in the Huntsville complex and used to develop application programs. Later the system was shipped to McDonnell-Douglas, Huntington Beach, California for its intensive benchmark testing. These activities are classified and the results cannot be published here. It can be reported, however, that the hardware performed exceedingly well and the system was returned to Huntsville.

The PEPE contribution might have been more formidable if the world political climate had warranted it so it may be assumed that it fulfilled a vital need. From an engineering viewpoint, it was simply a job well done.

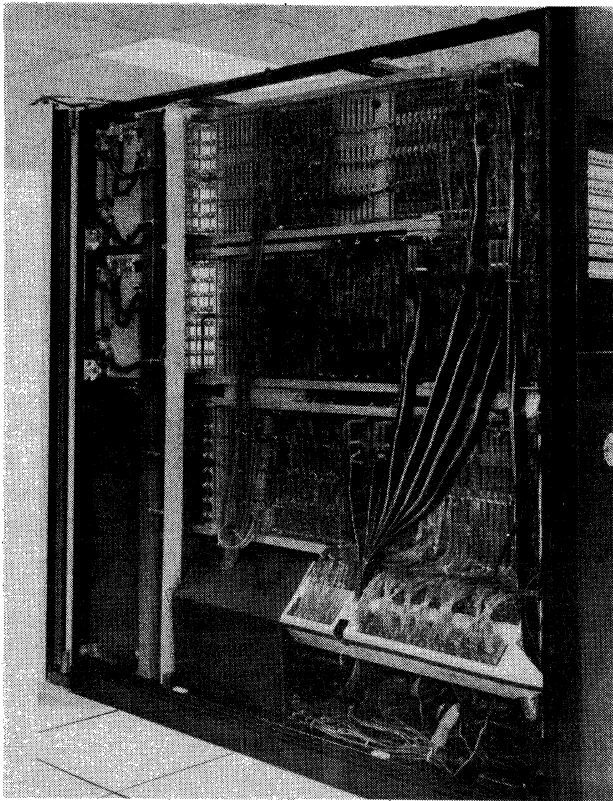


PEPE Cabinet, Front View

### BSP (Burroughs Scientific Processor)

The Burroughs Scientific Processor (BSP) was the result of an effort to develop a standard product supercomputer that would serve the scientific user community with massive computational requirements. This application requires machines with special architectures that can perform at levels beyond those achievable by circuit speed alone.

Fortunately, the programs often exhibit an internal structure in which the same operator can be applied to arrays or vectors of data. This had led to the development of several SIMD supercomputers of either an arithmetic pipelined or parallel processor design (e.g. ASC, STAR, and ILLIAC IV [1]). Both techniques had resulted in vector computers



PEPE Backplane

whose effective computational rates on suitable applications were one to two orders of magnitude greater than that of serial processors constructed of equivalent speed circuitry.

The generality of these machines was limited by restraints on the application programs. Due to pipeline start-up time, very long vectors of data were often required. A small scalar content could seriously degrade performance levels. Finally, they were difficult to program, often requiring assembly language coding and memory residency analysis in order that the speed of the machine be fully realized.

For these and other reasons, the only machines that had achieved commercial success by the early 1970's were the CDC 6600 and 7600 series which achieved their performance levels primarily by the use of very high speed circuitry and multiple function arithmetic processors.

Given the recently completed ILLIAC IV program and ongoing PEPE program, Burroughs had developed expertise in parallel processing which could be applied to developing a commercial super-computer. This, coupled with the Corporation's desire to field a FORTRAN processor to complement the product line and provide a test bed for a

new generation of high speed current-mode logic (BCML), provided the impetus for the development.

Although the BSP was not commercially successful, prototype and production models of the BSP were built, made operational, and in fact, met most of their design goals. The state of the computing art was advanced in several areas.

### Design Goals

The beginnings of the program can be traced to a feasibility study on repackaging ILLIAC IV which was conducted in 1972. A survey of the user community clearly showed that a more refined, easier to use machine was required. This led to the development of the set of design goals listed below.

**Standard Product.** The BSP was to be a standard product. This implied that it was to conform to the corporate standards for manufacturability, testability, reliability, maintainability, high level language programmability, ease of use and cost. It would be developed and manufactured by a standard M&E (Manufacturing and Engineering) plant. Corporate standard hardware technology was to be employed, providing a volume basis for material costs and manufacturing tooling.

**Attached Processor.** The BSP was to be attached to a large scale commercial computer system such as the B7700. This provided the capability to extend the FORTRAN performance of these machines and provided the user with access to the sophisticated system software developed for commercial large systems.

**Technology Driver.** The Corporation was currently engaged in the development of a high speed current mode logic family and its associated liquid cooled packaging technology, intended for use in Burroughs commercial plants. The BSP was to be a driver for this program. Thus it would provide schedule pressure on the components plants in advance of commercial requirements and be a test bed to shake down the technology.

**Programmability.** The BSP was to be efficiently programmable exclusively in a high order language. In practice, this meant that FORTRAN was the obvious choice. Any extensions were to be application oriented and machine independent. A vectorizer was to be provided as a means of efficiently executing existing codes.

**Ease of Use.** The machine was to be easy to use. This was motivated by users' desire to minimize the cost of developing and maintaining

application codes.

**Performance.** The BSP was to be capable of sustaining 20 to 40 MOPS on typical application codes in weather forecasting, nuclear reactor design, structural analysis, and other similar fields. This was to be measured on such standard benchmarks as the Livermore Loops.

In order to achieve these goals, several key technical problems had to be solved.

**Scalar Problem.** Some means had to be found to minimize the impact of scalar processing. This had been a bottleneck in then-current designs.

**Pipeline Start-up and Short Vector Performance.** A method had to be found for ameliorating the effect of pipe-start-up time so that high performance could be achieved on relatively short vectors.

**Memory Conflicts and Residency.** A memory structure had to be devised that would minimize the effect of memory conflicts which occurred when elements of operand vectors resided in the same memory bank. This structure could not require the user programmer to exhaustively study the application and specify special residency requirements.

**Automatic Bit Vector Control.** Bit vector control for data dependent branching and sparse vector operations had to be built into the machine and made easy to use.

**Generalized Parallel Processing.** The parallel processor had to be generalized so that it could be effectively employed in more applications. Research in parallel processing had resulted in many parallel algorithms for speeding up operations previously thought to be serial (e.g. linear recurrences [8]).

**Balanced I/O Structure.** High performance secondary store was required and had to be accessible without excessive operating system overhead.

**Self-checking and Fault Tolerance.** Extensive self-checking and fault tolerant mechanisms were to be built into the machine so that high reliability and trustworthiness could be achieved. This was to be done without seriously degrading the performance of the system.

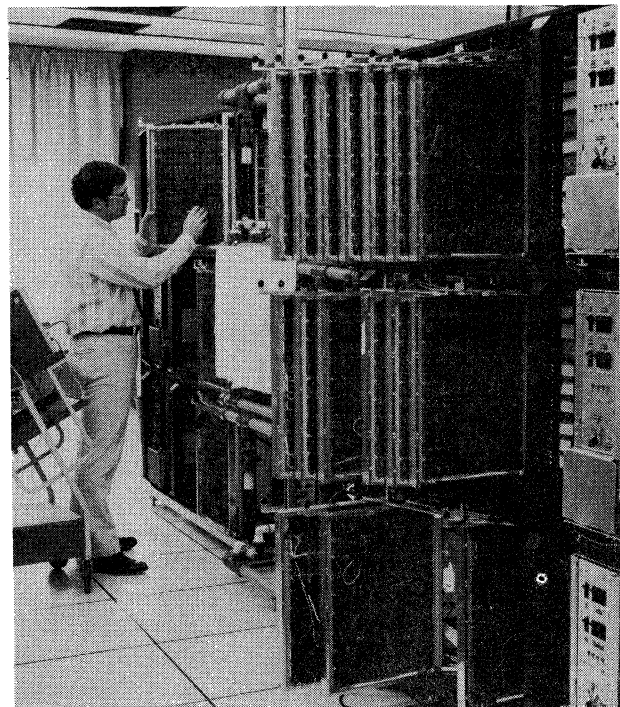
## Architectural Design

The solution of these problems was undertaken during the preparation of the PDA (Product Development Authorization — an internal proposal). This effort was completed in June, 1974.

The first issue to be decided was whether a pipelined or parallel processing approach would be taken. The latter was chosen because of the ease of implementing many of the sophisticated algorithms which had been discovered and the expertise which had developed during the ILLIAC IV program. Finally, the iterative nature of parallel processors made them more suitable for VLSI implementation in the future.

Once this had been decided, the memory conflict problem was then attacked. Although many skewing techniques were known for minimizing conflicts, none had the generality and uniformity that was desired. The result of this effort was a scheme [9] which offered conflict-free access to any linear vector whose skip distance was not a multiple of the prime number of memory banks. Even more importantly, the memory mapping was application independent.

The use of microprogramming was explored as a method of simplifying the programming of the machine and as a means of directly executing many common FORTRAN constructs such as nested DO loops with embedded assignment statements. This resulted in the development of the template concept, which allowed the overlapping of vector operations within the temporal pipeline of the parallel processor and solved the pipeline start-up problem. (Parallel processors do exhibit another



BSP Cabinet

start-up phenomenon in that full speed is not achieved until the vectors are at least as long as the width of the array.)

The scalar problem was attacked with an eye to minimizing the number of scalar operations and overlapping their execution with that of the parallel processor rather than relying solely upon raw circuit speed. Scalar operations were reduced by the application of parallel algorithms, automating memory indexing and parallel processor control operations in hardware, and off-loading I/O operations to a smart controller.

The remaining problems were solved in an exhilarating rush of discovery that culminated in a design which is remarkably similar to the final design documented in C. Jensen's paper [6]. The one major difference is that there were 67 slower dynamic memory banks which fetched vectors of length 64. The 16 arithmetic processors then executed the operation in 4 steps. Thus, the machine reached full speed at vectors of length 64. This allowed the use of low cost main memory.



BSP Demonstrating Class 6 Qualification

### Detailed Design

In the detailed design phase of the program (June, 1974 to August, 1976) the implementation of the concepts developed during the proposal was pursued. It had not been clear that the alignment network and automatic indexing hardware could be built out of a reasonable number of IC's or that there would not be a combinatorial explosion of microcode. These problems were overcome and the design had successfully incorporated the features of the architecture.

The applications group had found that length of vectors in many codes were shorter than 64. It

would be desirable to improve the short vector performance of the machine. The advent of low cost high speed static NMOS memories such as the 2147 made it possible to do this. The number of memory modules was reduced to 17 and the memory cycle time speeded up by a factor of 4. This allowed the parallel processor to come up to speed at vector lengths of 16 while providing the additional benefit of simplifying the design.

This had the result of throwing the design into imbalance. The scalar processor had to prepare descriptors four times as fast as before. The scalar unit had to be speeded up in order to fully take advantage of the faster parallel processor.

### The Turning Point

A related sequence of events occurring in 1977 had a large effect on the program. It had been observed that the scalar unit was, itself, functionally complete and could be offered as a lower cost Attached FORTRAN Processor (AFP). This product appeared to be relatively free and was adopted. However, it resulted in two releases, two sets of software, the development of a DISK version of the I/O system, and an interface to the B 6800. This represented a significant additional workload on the project.

The BCML development was very late and did not meet the original performance goals. A proposal to implement the first machine in the proven hardware of the PEPE system was rejected because the objective of driving the technology was deemed essential.

It was becoming clear that the performance of the scalar unit would not support application programs that did not contain a sufficiently high content of vector operations. The design of the scalar unit was straightforward, to minimize the overall development risks to the program. The performance on the Livermore Logics benchmarks (a scalar-vector mix) reinforced our strategy, but a broader product approach would require a performance enhancement of the unit. At this point, with limited time and resources, it was felt the problem could be addressed in a subsequent product upgrade after the production start of the present design.

### Making It Work

The machine was debugged during 1977 to 1980. There were many problems to overcome. Initially, late deliveries of circuits delayed the pro-

gram. When sufficient quantities were available, the hardware was built and put into system test.

The hardware technology was completely new, from the circuits to all three levels of packaging. In addition, the emerging CCD technology was to be employed for a second level store. Given the number of new items, it perhaps is not surprising that some design problems surfaced.

The first design of the sockets exhibited loose contacts, the proms speeds drifted, and there was a damaging latent fault in the zinc pillow blocks. These blocks were screwed in to hold the PWB assembly together and were under high pressure. They exhibited a cold flow phenomenon which caused the screws to slowly pull out. The assemblies were literally pulling themselves apart. A third of the machine had to be reworked in the midst of debugging. The CCD devices exhibited a high soft failure rate and were difficult to manufacture.

These problems were overcome and the production hardware was fully qualified, very reliable, and exceptionally stable. There were practically no electrical intermittents reported. The CCD memory was replaced by a dynamic RAM system. While this process of shaking down the hardware technology fulfilled one of the main objectives of the program, it delayed getting the machine into the marketplace at a critical time when CRAY was making deliveries for almost 2 years.

The software set was new and fully featured. The maturization of this amount of software took a long time and prevented us from routinely running customer benchmarks. This was aggravated by the temporary loss of all 7700's for customer shipments, which resulted in no system manager to debug the deliverable software (the alternate, but different, 6800 software was used instead). Nonetheless, by 1979, limited benchmarks could be run to measure the performance characteristics of the system.

**Performance Measurement and Marketing.** In the codes that were tested, the design lived up to its promise as an excellent vector processor. The Livermore loops ran at over 20 MOPS. In general, most comparisons showed that the machine was equivalent in performance to the CRAY I for many vectorizable codes. This was true even though the short vector performance of the parallel processor was only being partially realized and the hardware components were considerably slower.

Although the large main memory and fast secondary store was an advantage in large problems, users preferred the CRAY due to the guaranteed performance levels that could be achieved on existing non-vectorized and scalar codes.

**Conclusion.** The cancellation of the BCML and CCD programs, the attendant cost increases, the loss of an appropriate marketing window, and the lack of a dominant scalar speed led to the cancellation of the product. The design proved that it was possible to configure a parallel processor which was competitive in vector applications and considerably more general than those that had been designed in the past. This drive for generality is expected to continue into the next generation of MIMD architectures.

#### References

- (1) Barnes, G. et al. "ILLIAC IV Arithmetic Element," *IEEE Transactions on Computers* (August 1968), Vol. C 17, No. 8, pp. 746-757.
- (2) Davis, R. L. "ILLIAC IV Arithmetic Element," *IEEE Transactions on Computers* (September 1969), Vol. C-18, pp. 800-816.
- (3) Abel, N. et al. "TRANQUIL - A Language for an Array Processing Computer," *Proceedings AFIPS Joint Computer Conference*, Vol. 34, pp. 57-73.
- (4) Lawrie, D. "GLYPNIR - A Programming Language for ILLIAC IV," *Communications of ACM* (March 1975), Vol. 18, No. 3, pp. 157-164.
- (5) Stokes, R. A. "Burroughs Scientific Processor," *Proceedings of the Symposium on High Speed Computation*, University of Illinois.
- (6) Jensen, C. "Taking Another Approach to Supercomputing," *Datamation* (February 1978), pp. 159-172.
- (7) Kuck, D. J. "A Survey of Parallel Machine Organization and Programming," *ACM Computing Surveys* (March 1977), Vol. 9, No. 1, p. 29.
- (8) Chen, S. C. and Kuck, D. "Time and Parallel Processor Bounds for Linear Recurrence Systems," *IEEE Transactions on Computers* (July 1975), Vol. C 14, No. 7, pp. 701-717.
- (9) Lawrie, D. "Access and Alignment of Data in an Array Processor," *IEEE Transactions on Computers* (December 1975), Vol. C 24, No. 12, pp. 1145-1155.