

 **TANDEM** COMPUTERS

# **The Hardware Architecture and Linear Expansion of Tandem NonStop Systems**

Robert Horst  
Tim Chou

Technical Report 85.3  
April 1985  
PN87610



The Hardware Architecture and Linear Expansion of  
Tandem NonStop Systems

Robert Horst and Tim Chou

April 1985

Tandem Technical Report 85.3



The Hardware Architecture and Linear Expansion of  
Tandem NonStop Systems

Robert Horst and Tim Chou

April 1985

ABSTRACT

The Tandem NonStop TXP is a commercially available multiple processor system that delivers mainframe class performance for transaction processing applications. Several sixteen-processor systems may be configured in a ring structure using fiber optics. This structure allows from two to over two hundred processors to be applied to a single online application. Benchmark results are presented to demonstrate the linear growth of system performance as processors are added.

-----  
A version of this paper appears in: Proceedings of 12th International Symposium on Computer Architecture, June 1985.  
-----

(TM) Tandem, NonStop, NonStop II, NonStop TXP and FOX are trademarks of Tandem Computers Incorporated.



TABLE OF CONTENTS

Tandem NonStop Architecture Evolution: 1976-1981.....1  
Tandem NonStop Architecture Evolution: 1981-1985.....4  
    Fiber Optic Extension (FOX).....4  
    TXP Processor Design Rationale.....8  
    The NonStop TXP Processor.....11  
Performance Benchmarks.....15  
    Banking Benchmark.....15  
    Retailing Benchmark.....18  
    Summary.....20  
Conclusions.....21  
Acknowledgments.....22  
References.....23





## TANDEM HARDWARE ARCHITECTURE EVOLUTION: 1976-1981

The market for high volume transaction processing has increased rapidly. In the 60's, only large airlines required large on-line transaction processing (OLTP) systems. These requirements were filled by centralized mainframe computers running specialized, highly tuned applications. They suffered from limited expandability, a costly applications environment, and the requirement to program in assembly language [5].

Today, many other industries are taking advantage of OLTP systems. Some of these applications include on-line banking, credit authorization, debit cards, teletext, telephone billing, electronic mail, medical information systems and paperless factories. These markets have similar systems requirements: the system must continue to operate despite hardware failures, it must be expandable, and it must be capable of high transaction throughput.

In 1976 Tandem introduced a new system architecture specifically designed to address the problems of OLTP. Designated the NonStop I, this system consisted of from two to sixteen loosely coupled processors which communicated with each other over dual high speed busses [2,10]. The Tandem hardware architecture is illustrated in Figure 1. This loosely coupled architecture has proven to be effective for transaction processing by supporting incremental expansion, high availability, and high performance [8]. The loose coupling does not limit performance since transaction processing,

unlike most scientific processing, is easily partitioned into multiple relatively independent processes.

Briefly, each CPU runs at one to two mips, memory is from 1 to 16 megabytes, each IO channel runs at 5 megabytes/second, and each interprocessor bus (Dynabus) runs at a peak rate of 16 megabytes per second.

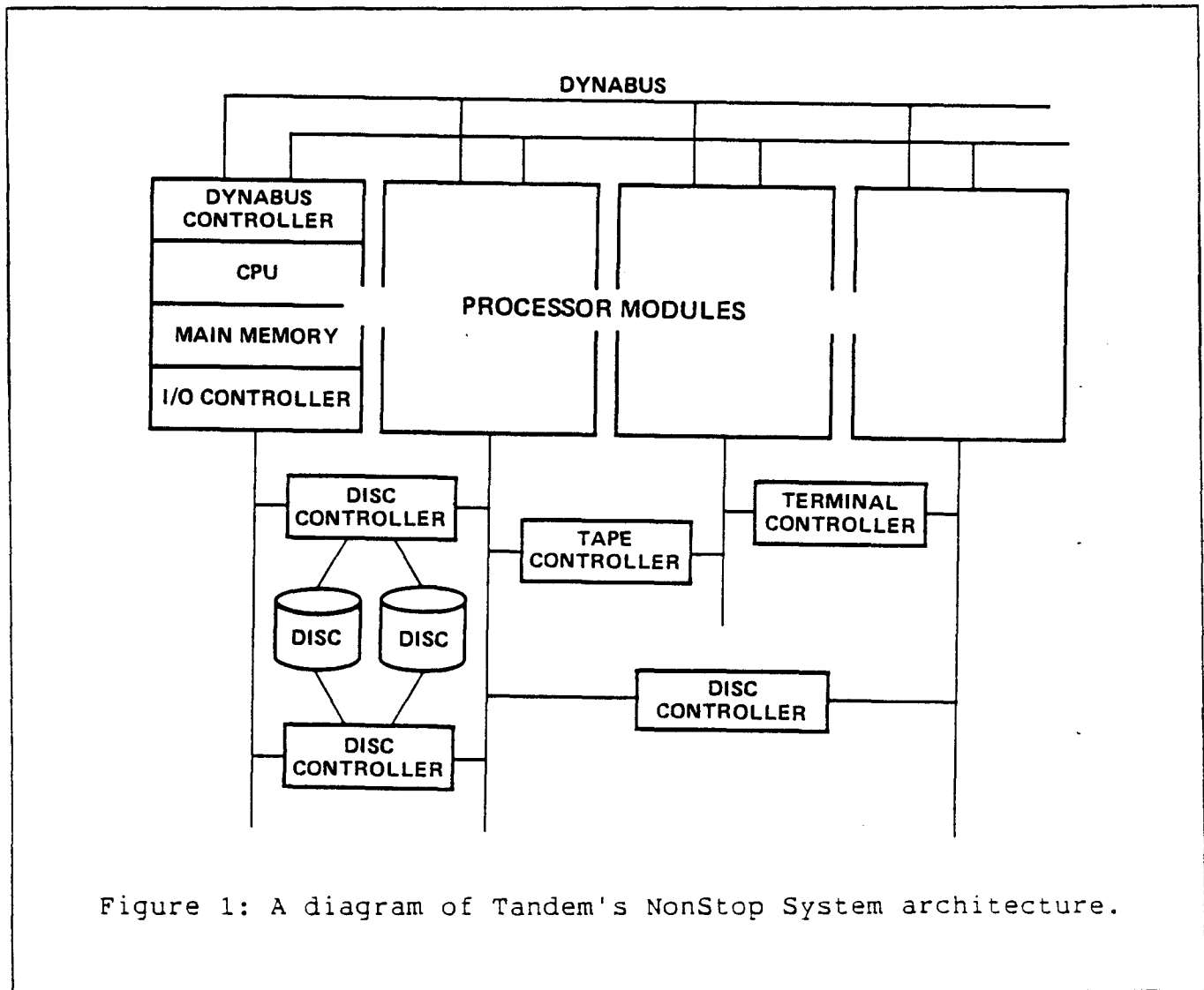


Figure 1: A diagram of Tandem's NonStop System architecture.

In 1981, the NonStop II was introduced to remove addressing limitations of the 16-bit NonStop I. Many new software features have been added to the basic offering. The system was expanded into a network of up to 255 systems, a relational data base management system as well as transaction management software [1,3,4] were also added. These products allowed users to develop distributed fault tolerant OLTP applications without worrying about the underlying system implementation.

## TANDEM HARDWARE ARCHITECTURE EVOLUTION: 1981-1985

The Tandem system as it stood in 1981 solved all of the requirements for OLTP, but the performance required for some large applications was beyond the reach of a 16-processor system. The network provided a way to apply more than one system to a single application without reprogramming; however, the relatively slow speed of data communications lines and software overheads of long-haul communication protocols proved to be a bottleneck in high volume applications. It was apparent that there was a need for systems able to support high volume transaction processing. The need was addressed on two fronts. A project was started to expand the number of processors which could be applied to a single application, and another project was started to develop a higher performance processor.

### FIBER OPTIC EXTENSION (FOX)

The most obvious way to increase the number of processors in a Tandem system is to extend the high-speed interprocessor busses to handle more processors. While this is not difficult from a hardware design standpoint, there are drawbacks. All processors would be required to be in close physical proximity in order to keep from degrading the bus performance. This would cause a physical space problem in some computer rooms -- the associated discs and communications gear require considerable space. In addition, allowing a single system to expand to more processors does not help existing customers who already have several systems requiring higher bandwidth communications.

An alternative approach to adding processors in the same system is to use a high speed connection between systems. This effectively adds another level of interconnection hierarchy between the 26 Mbytes/sec inter-CPU links, and the 56 Kbyte/sec network links. Figure 2 illustrates the Tandem solution which uses fiber-optics to link up to fourteen systems.

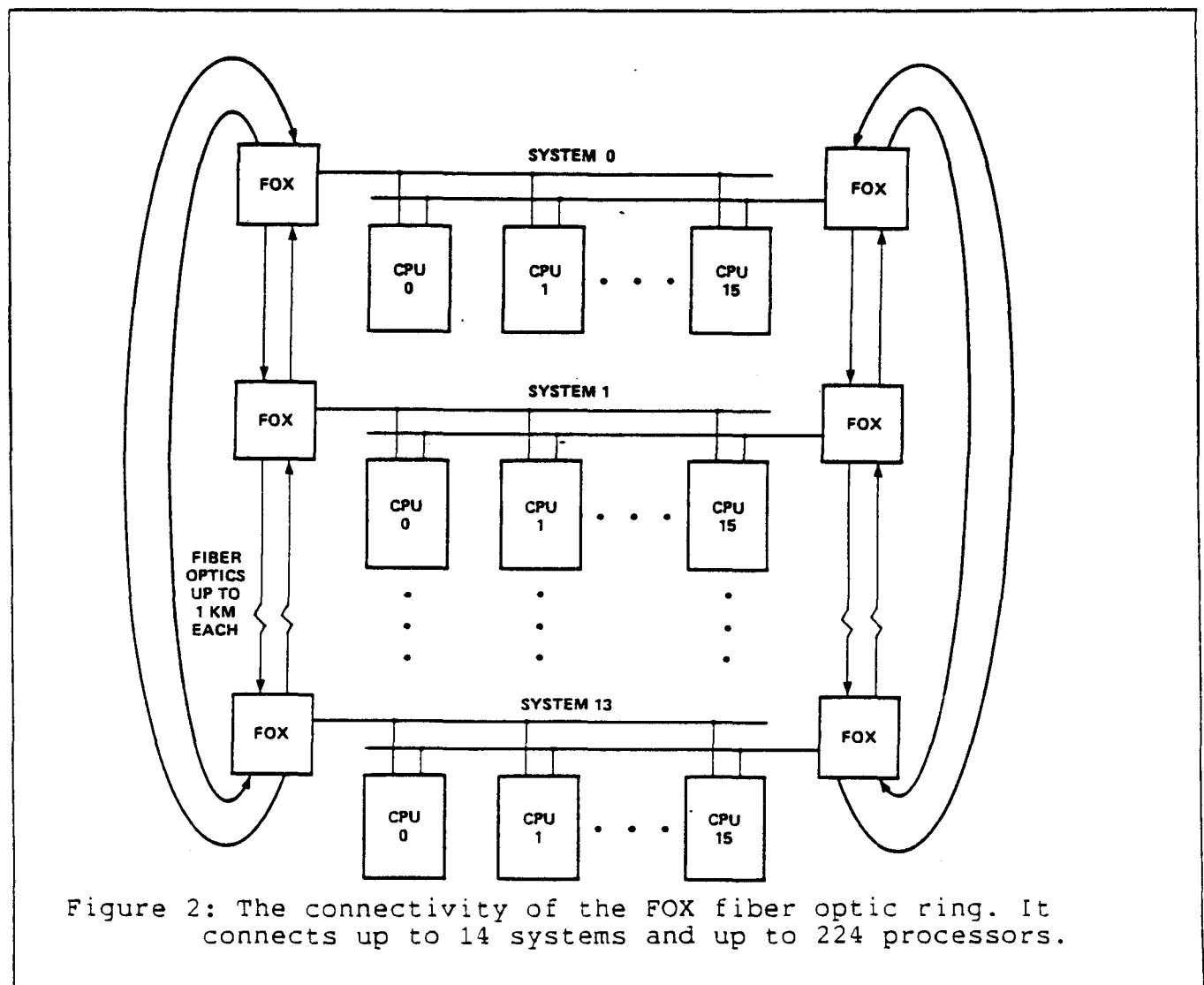


Figure 2: The connectivity of the FOX fiber optic ring. It connects up to 14 systems and up to 224 processors.

Each node can accept or send up to 4 Mbytes/sec. With this additional bandwidth, a cluster of up to 224 CPU's can be configured to handle a single on-line transaction processing application.

The topology of the FOX connection is a store-and-forward ring structure. Four fibers are connected between a system and each of its two neighbors. Each interprocessor bus is extended by a pair of fibers, which allows messages to be sent in either direction around the ring. The four paths provided between any pair of systems assures that communication is not lost if a system is disabled (due to a power-fail perhaps) or if an entire four fiber bundle is severed. The ring topology also has advantages over a star in that there is no central switch which could be a single point of failure. In addition, cable routing is easier with a ring than with a star.

In a ring structure, bandwidth increases as additional nodes are added. The total bandwidth available in a FOX network depends on the amount of passthrough traffic. In a 14 node FOX ring if each node sends to all other nodes with equal probability the network has a usable bandwidth of 10 Mbytes/sec. With no passthrough traffic the bandwidth increases to 24 Mbytes/sec. Theoretically an application generating 3K bytes of traffic per transaction, at 1,000 transactions per second, would require a FOX ring bandwidth of only 3 Mbytes/second. This would put total utilization of the FOX network at less than 30% of the total available bandwidth.

Fiber optic links were chosen both to solve technical and practical problems in configuring large clusters. Since fiber optics are not susceptible to electromagnetic interference, they provide a reliable connection even in noisy environments. They also provide high bandwidth communications over fairly large distances (1 km). This eases the congestion in the computer room and allows many computers in the same or nearby buildings to be linked. Fiber optic cables are also flexible and of small diameter, thus easing installation. Figure 3 provides additional details on FOX.

- o Ring-configured fiber optic inter-system link
- o Links up to 14 systems of 16 processors
- o Up to 1km between any two adjacent nodes
- o Data rate: 4 mbyte/sec per node, max 28 mbyte/sec per ring
- o Light source: Light emitting diodes
- o Receiver: PIN (Positive Intrinsic Negative) photodiodes
- o Cable: Glass fiber optics, attenuation 6dB/km max

Figure 3. Fox fact sheet.

## TXP PROCESSOR DESIGN RATIONALE

Once a system architecture can apply a large number of processors to a single application, there is still a question of what the characteristics of the processor should be. The possibilities span a large range of performance levels. At the low end of the performance range are microprocessor based designs. These may be based on one chip microprocessors such as the Motorola 68000 family, the Intel 8086 family or the National 32000. In the midrange are designs based on medium-scale integration or gate arrays. Such designs are typified by mini/supermini computers such as the Digital VAX series, Hewlett-Packard 3000 and the IBM 4300 series. At the high end are designs based on the most aggressive IC, packaging, and cooling technologies. Examples of this type of design are mainframes such as the IBM 3090, Amdahl 580, and high end machines from Sperry, Burroughs, CDC and Cray.

Many factors need to be analyzed in order to decide whether the micro, mini, or mainframe approach should be preferred for the processors in a high volume transaction processing system. These factors include cost-performance, granularity of fault tolerance, granularity of adding performance, and ease of managing the system.

In a system which is modularly expandable, cost-performance is the driving force in the development of a new processor. Improving cost-performance by lowering costs is difficult. Even using a microprocessor, which may be nearly free, may not significantly reduce



costs over a minicomputer-style design due to the many fixed costs which make up a system. The cost of main memory, packaging, power and cabling are not reduced in proportion to CPU cost reductions.

It is easier to improve cost-performance by increasing the performance of the processor. This seems to favor a mainframe as the best choice for a multiprocessor system. There are several reasons why this is not the case. The complexity of a mainframe design requires a much larger development cost and longer development time. In addition, in trying to improve uniprocessor performance some nonlinearities are encountered. For instance, the jump from air to liquid cooling constitutes a large cost increment.

Other factors influence the choice of processors. If the processor is too small, the number of processors required to perform a large application can become so large as to be unmanageable. In addition, if the system is not carefully designed, performance improvements can cease to be linear beyond some number of processors.

On the other hand, if the processor is so powerful that the application can be handled by a single processor, fault tolerance can suffer. Today the only fault tolerant configurations of mainframes require an expensive duplicated hot standby. Even if these machines were incorporated into a Tandem-like structure which would allow the second machine to do useful work, the failure of one of the mainframes would remove half the computing power from the system. In extremely critical applications, it may not be tolerable to degrade performance

while a hardware failure is being repaired. If the application requires only one processor to handle the peak load, a second processor is needed in case of failure, for a 100% overhead. In contrast, if four less powerful processors are used to handle the same workload, only one extra processor is needed in case of failure for a 25% overhead.

## THE NONSTOP TXP PROCESSOR

Based on the above rationale, Tandem introduced the NonStop TXP processor in 1983. Its primary design objectives were to improve both cost-performance and absolute performance over the NonStop II [7]. The initial pricing of the NonStop TXP offered a 50% improvement in price-performance over the NonStop II at about 2.5 times its performance.

In the NonStop TXP design, the emphasis on cost-performance extended all the way to the component level. One of the first decisions to be made was the selection of static RAM's to be used in the cache memory and control store. The most advanced RAM's at that time were organized as 4Kx4 and 16Kx1 bits with access times of 45ns. These were four times the density of and 10 ns faster than the RAM's used in the NonStop II.

To implement logic functions, advanced MSI Schottky technology and programmable array logic were chosen. An extensive analysis of LSI alternatives available in 1981 showed that a gate array machine would have been about the same performance level, higher cost, and would have required a much longer development cycle.

Once a technology was chosen, the next challenge was to develop an architecture which could utilize the improved components to improve performance and cost-performance. One such area was the cache memory design. Although extensive academic research in cache memories was

available during the NonStop TXP design [11], most of the studies did not anticipate the impact of large RAM's on cache organizations. Using 16K static RAM's, a 64K byte cache requires only 32 components (not including parity or the tag comparison RAM's or logic). This makes it much more economical to design a large "dumb" cache (direct mapped) than a smaller "smart" cache (set associative). After performing some real time emulation of different cache organizations, the final cache design for the NonStop TXP was chosen. It is a 64K byte direct mapped virtual cache with hashing of some of the address bits. Hit ratios for the cache have been measured between 96% and 99% while performing transaction processing workloads.

Other tradeoffs were also made in the interest of cost performance. In order for the NonStop TXP to be plug-compatible with the NonStop II processor, the CPU was required to fit on four circuit boards. Had it overflowed those boards, a large jump in cost would have occurred. For this reason, the NonStop TXP relies on microcode to perform some of the functions done in hardware on many machines. For instance, after every cache access, the microcode must perform a conditional branch to test for a cache miss. If a miss occurred, the microcode fetches the block from memory and refills the cache block. Performance could have been improved a few percent by providing additional control and data path hardware to perform the cache refill directly. However, since this hardware would have required an additional logic board, it would have adversely affected cost-performance.

Many of the tradeoffs made in the NonStop TXP design were based on detailed measurements of the NonStop II performance. A complex performance analyzer, named XPLOR, was designed and built solely for that purpose. XPLOR was used to perform the cache emulation experiments. In addition, it provided data on instruction frequencies, percent time spent in each instruction, and the memory reference activity of each instruction. This allowed hardware to be saved in the support of less frequent instructions and applied to accelerating the more frequent instructions. XPLOR also provided data which enabled the microcode to be optimized for the more frequent paths through complex instructions.

The final result is a processor which has a 83.3 ns microcycle time and can execute simple instructions in only two cycles. In typical applications, the NonStop TXP executes about 2 million instructions per second. This new processor has not uncovered any bottlenecks in the I/O or message system; hence, the improvements in CPU performance have been directly translated into transaction processing performance. Figure 4 summarizes the TXP features.

- o DYNABUS : 26Mbytes/sec
- o 2 MIP's per processor
- o 83.3 nsec microcycle time
- o Three stage microinstruction pipelining
- o Three stage macroinstruction pipelining
- o Dual data paths and dual arithmetic-logic units
- o Two level control store - 8K x 40 bits and 4K x 84 bits
- o Extensive parity and selfchecking
- o 64 Kbyte cache memory - 96% to 99% hit ratio
- o 64-bit access of main memory
- o 2-8 Mbytes physical memory (64K DRAMs)
- o 1 Gbytes virtual memory addressing
- o Basic instruction set of 220 instructions
- o Optional decimal and floating point instruction sets
- o 3 - 12 hour battery backup of main memory
- o Plug-compatible with NonStop II processor
- o 5 Mbyte/sec burst-multiplexed channel per processor
- o Up to 32 I/O controllers per processor
- o Up to 256 I/O devices per processor
- o Data communications:
  - Byte synchronous - 80kbps
  - Bit synchronous - 56kbps
  - Asynchronous - 50bps to 19.2kbps

Figure 4. TXP Fact sheet.

## PERFORMANCE BENCHMARKS

Recently several customer benchmarks have been run which demonstrate the capability of the Tandem architecture to support high volume, high performance on-line transaction processing. Two of these benchmarks are described below.

### BANKING BENCHMARK

In the summer of 1984 a major European bank benchmarked a Tandem system to support a bank card and electronic funds transfer application. The data base for the application consisted of the following files:

Card	: 1.2 million records of credit cards
Account	: 2.4 million records of account balances
Log	: History of updates to Accounts
Customer	: 4 million records about customers

The application was described by five major types of transactions. The two most frequent transactions were the DEBIT and LOOKUP transactions.

DEBIT Transaction Profile:

X.25 Message in;  
Read random Card;  
Read random Account;  
X.25 Message out;  
X.25 Message in;  
Read random with lock same Account;  
Rewrite Account;  
Sequential write to log of Account update;  
X.25 Message out.

LOOKUP Transaction Profile:

X.25 Message in;  
Read random Card;  
Read random Account;  
X.25 Message out;  
X.25 Message in;  
Read random Customer;  
X.25 Message out;

In this benchmark, 50% of the transactions were DEBIT, 40% were LOOKUP, and 10% were other transactions.

The benchmark was done on a 4, 8, 12, and 16 processor single system as well as 1, 2, 3 and 4 4-CPU systems FOXed together. The results are shown in Figure 5. Note that the transaction rate grows linearly in the case where the processors are interconnected via the Dynabus. This is also true when processors are interconnected via FOX. In addition, FOX introduces almost no throughput degradation.



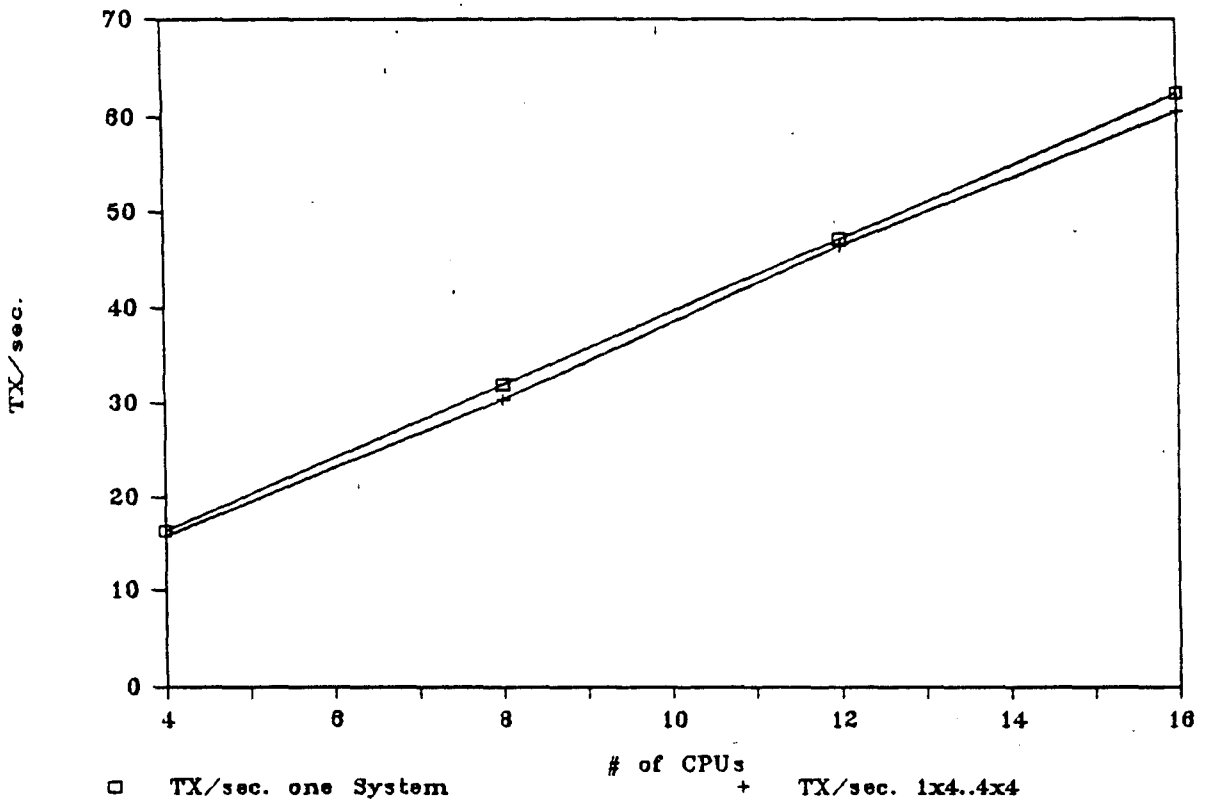


Figure 5. An benchmark to demonstrate Dynabus and Fox linearity. Transaction throughput vs the number of cpus on the DynaBus, and vs the number of cpus on a FOX ring.

## RETAILING BENCHMARK

In the fall of 1984 a major American retailer benchmarked a retail credit authorization application. The requirements were for three individual sites, each site providing 100+ authorization transactions per second for its area. Each site will be connected through a large SNA network and can communicate directly with the other two when necessary for processing out-of-area authorizations. The data base for the application consisted of the following files:

Authorization	: 20 million records
Bankcard Negative	: 10 million records
Out-of-Area Index	: 10 million records
Transaction Journal	: Large entry sequenced file

The application was composed largely of one transaction, MAIN, which was nearly 75% of all transactions.

### MAIN Transaction Profile:

- SNA Message in;
- Five in-memory table lookups;
- Random Read with Lock of Authorization file;
- Write Authorization file;
- Write Transaction Journal
- SNA Message out;

As with the banking benchmark, the retail benchmark showed linear growth in transaction throughput as processors and discs are added. By doubling the number of cpus and discs, the benchmark got twice the throughput for the same response time -- less than 1.7 seconds for 90% of the transactions. This is illustrated in Figure 6.

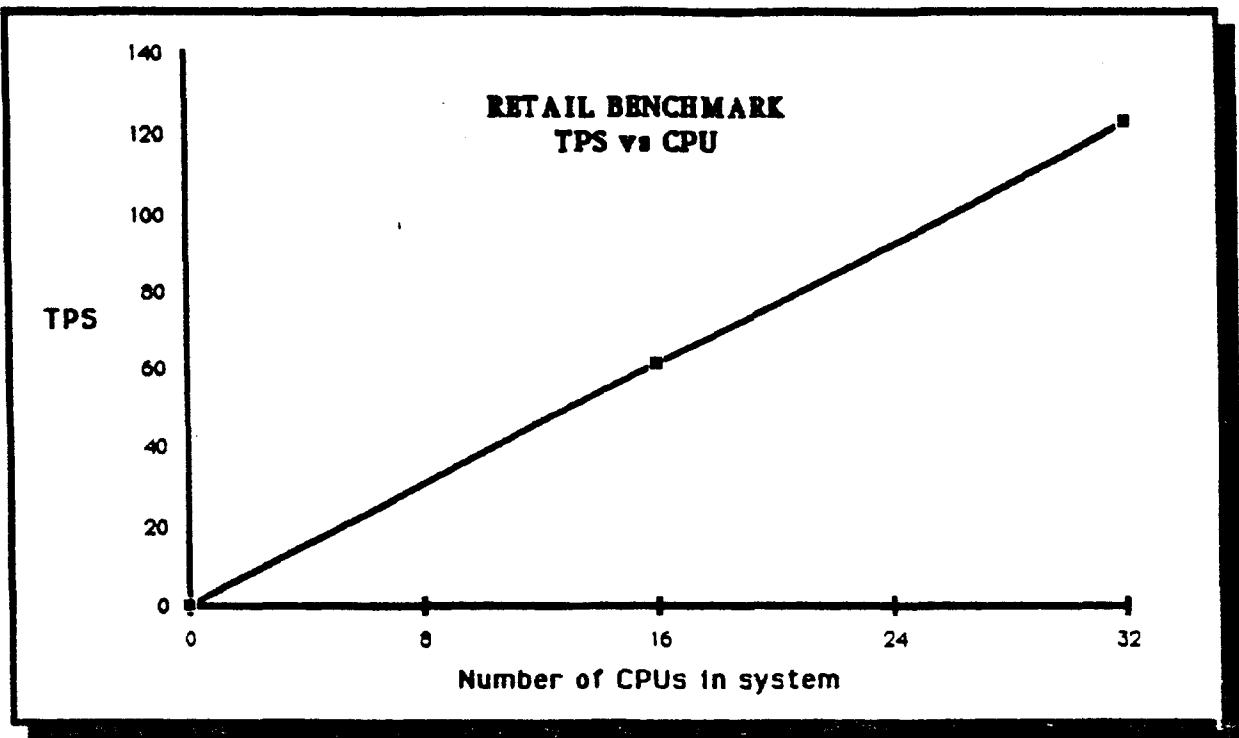


Figure 6. Linearity of transaction throughput for a retailing benchmark.

## SUMMARY

Both of these benchmarks demonstrate the capability for this system architecture to provide high volume transaction processing. The actual transaction throughputs are shown in Figure 6. Response times in both benchmarks averaged 1 second. Figures 5 and 6 graphically illustrate both the linearity and processing power in the region between 2 and 32 processors.

These graphs do not contain a wealth of data points because each benchmark is expensive in both time and equipment. However, they do represent real measurements as opposed to theoretical modeling results.

There is a great temptation to extrapolate these curves to 224 processors (the total number in a FOX ring). Assuming linear growth the transaction rate that hypothetically could be supported is somewhere over 800 transaction per second. In reality this may or may not be achievable; however, we hope at some point in time to be able to ascertain this experimentally. Once again, it should be noted that building a 224 processor system and benchmarking it is extremely costly in both time and equipment.

## CONCLUSIONS

For a number of years there has been academic interest and hypotheses [9] that a number of small processors could be tied together in some way and provide the computing power of a larger machine. While this may not be true in general, this paper illustrates that it is possible in on-line transaction processing.

Ordinary OLTP systems bottleneck at 50 transactions per second while high performance OLTP systems achieve 200 transactions per second. Beyond these rates, users have been forced to build their applications using ad hoc techniques rather than general purpose systems. Even using these ad hoc techniques, the highest transaction throughput claimed by any mainframe manufacturer is somewhere near 800 transactions per second [6]. Our experimental results to date indicate that the Tandem architecture, a general purpose multiprocessor, is capable of supporting high volume transaction processing.

## ACKNOWLEDGEMENTS

The authors would like to thank Jim Gray, Harald Sammer, Eric Chow, Joan Zimmerman, and Hoke Johnson for providing many helpful remarks and suggestions. The reported benchmarks were done by Harald Sammer's Frankfurt High Performance Research Center and by Tom Houy's Large Systems Support group at Tandem.

## REFERENCES

- [1] Arceneaux, G. et al., "A Closer Look at Pathway," Tandem Computers, SEDS-003, June 1982.
- [2] Bartlett, J., "A NonStop Kernel," Proceedings of the Eighth Symposium on Operating System Principles, pp. 22-29, December 1981.
- [3] Borr, A., "Transaction Monitoring in ENCOMPASS," Proceedings of the Seventh International Conference on Very Large Databases, September 1981. Also Tandem Computers TR 81.2.
- [4] Borr, A., "Robustness to Crash in a Distributed Database: A Non Shared-Memory Multi-processor Approach," Proceedings of the Tenth International Conference on Very Large Databases, September 1984.
- [5] Gifford, D., Spector, A., "The TWA Reservation System," Communications of the ACM, vol. 27, no. 7, pp. 650-665, July 1984.
- [6] Gray, J. et al., "One Thousand Transactions Per Second", in the Proceedings of the IEEE COMPCON-85, San Francisco. Also Tandem Computers TR 85.1.
- [7] Horst, R. and Metz, S., "New System Manages Hundreds of Transactions/Second," Electronics, pp. 147-151, April 19, 1984. Also Tandem Computers TR 84.1.
- [8] Kim, W., "Highly Available Systems for Database Applications," Computing Surveys, vol. 16, no. 1, pp. 71-98, March 1984.
- [9] Satyanarayanan, M., "Multiprocessing: An Annotated Bibliography," Computer, pp. 101-116, May 1980.
- [10] Siewiorek, D., Bell, C.G., Newell, A., Computer Structures: Principles and Examples. McGraw Hill Book Company, 1982.
- [11] Smith, A. J., "Cache Memories," Computing Surveys, vol. 14, no. 3, pp. 473-530, Sept. 1984.





Distributed by  
 **TANDEM** COMPUTERS  
Corporate Information Center  
19333 Vallco Parkway MS3-07  
Cupertino, CA 95014-2599

