A Fault-Tolerant Computing System

James A. Katzman
Tandem Computers
19333 Vallco Parkway
Cupertino, California 95014

First Revision January, 1979

Abstract

A fault-tolerant computer architecture is examined that is commercially available today and installed in many industries. The hardware is examined in this paper and the software is examined in a companion paper [4].

Introduction

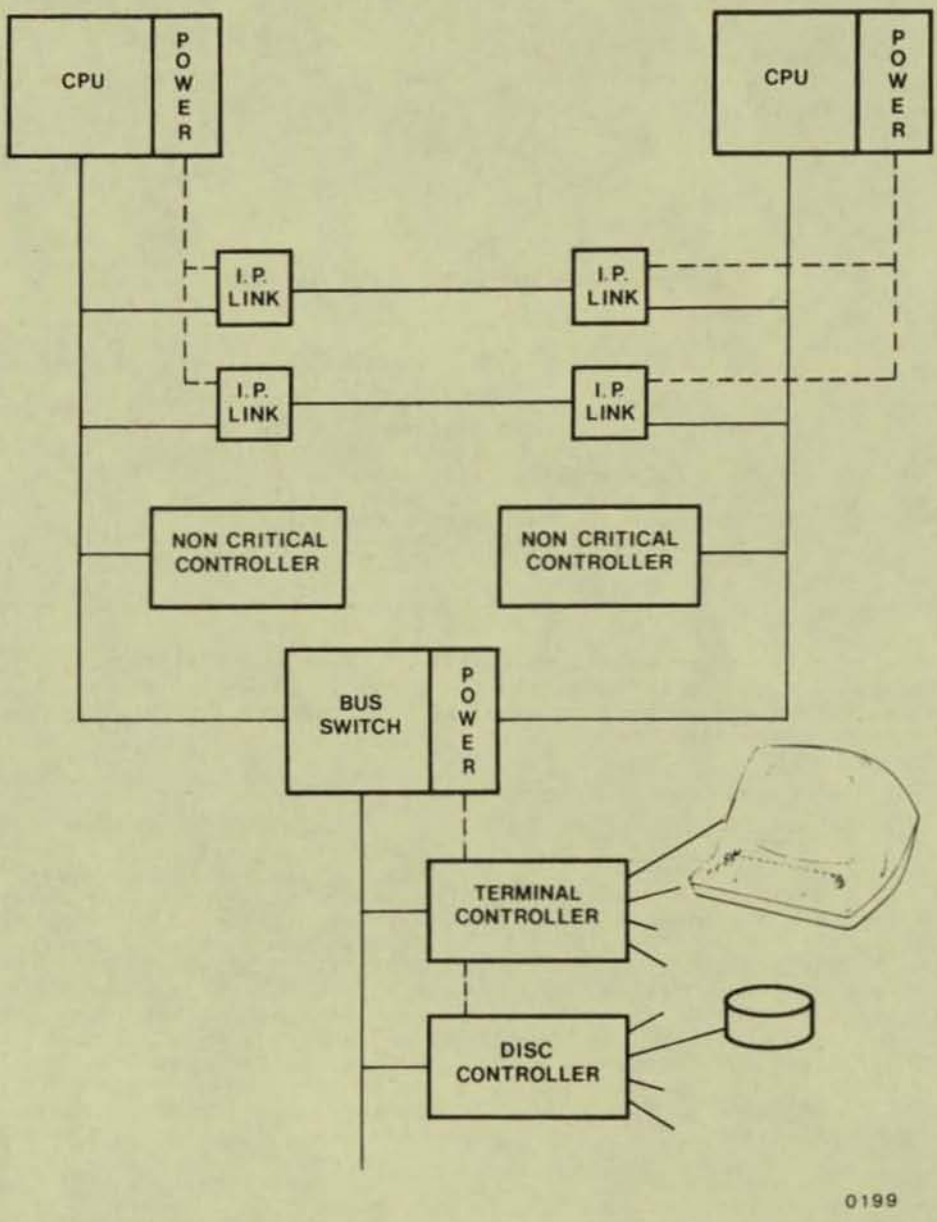
The increasing need for businesses to go on-line is stimulating a requirement for cost effective computer systems having continuous availability [1,2]. Certain applications such as automatic toll billing for telephone systems lose money each minute the system is down and the losses are irrecoverable. Systems commercially available today have met a necessary requirement of multiprocessing but not the sufficient conditions for fault-tolerant computing.

The greatest dollar volume spent on systems needing these fault-tolerant capabilities are in the commercial on-line, data base transaction, and terminal oriented applications. The design of the Tandem 16 NonStop* system was directed toward offering the commercial market an off-the-shelf, general purpose system with at least an order of magnitude better availability than existing off-the-shelf systems without charging a premium (see Appendix A). This was accomplished by using a top down system design approach, thus avoiding the shortcomings of the systems currently addressing the fault-tolerant market.

Except for some very expensive special systems developed by the military, universities, and some computer manufacturers in limited quantities, no commercially available systems have been designed for continuous availability. Some systems such as the ones designed by ROLM have been designed for high MTBF by "ruggedizing," but typically computers have been designed to be in a monolithic, single processor environment. As certain applications demanded continuous availability, manufacturers recognized that a multiprocessor system was necessary to meet the demands for availability. In order to preserve previous development effort and compatibility, manufacturers invented awkward devices such as I/O channel switches and interprocessor communication adapters to retrofit existing hardware. The basic flaw in this effort is that only multiprocessing was achieved. While that is necessary for continuously available systems, it is far from sufficient.

Single points of failure flourish in these past architectures (Fig. 1). A power supply failure in the I/O bus switch or a single integrated circuit (IC) package failure in any I/O controller on the I/O channel emanating from the I/O bus switch will cause the entire system to fail. Other architectures have used a common memory for interprocessor communications, creating another single point of failure. Typically such systems have not even approached the problem of on-line maintenance, redundant cooling, or a power

* NonStop is a trademark of Tandem Computers

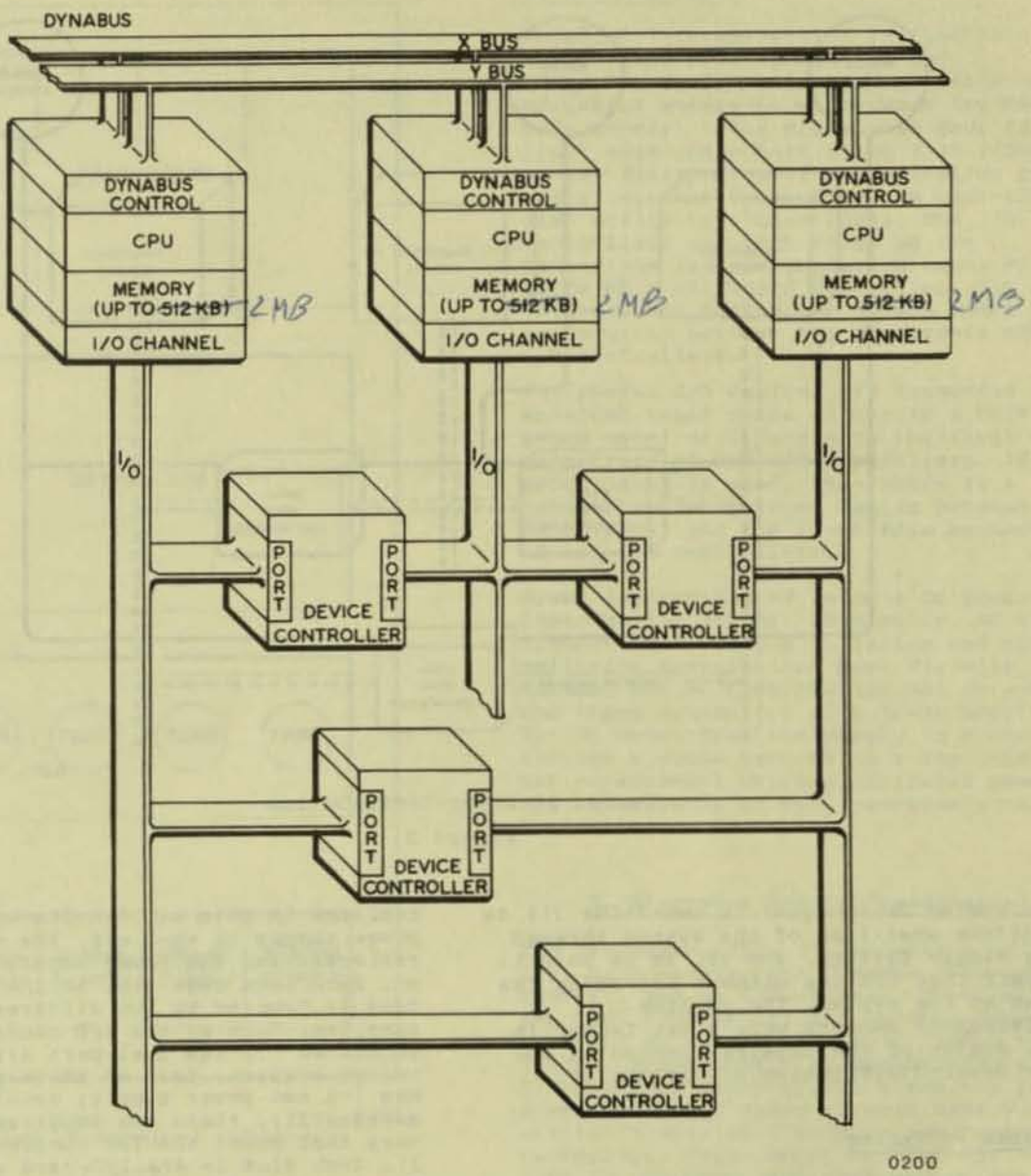


Example of Previous Fault-Tolerant Systems

Figure 1

distribution system that allows for brownout conditions. In today's marketplace, many of the applications of fault-tolerant systems do not allow any down time for repair.

Expansion of a system such as the one in figure 1 is prohibitively expensive. A three processor system, strongly connected in a redundant fashion, would require twelve interprocessor links on the I/O channels; five processors would need forty links; for n processors, 2n(n-1) links are required. These links often consist of 100-200 IC packages and require entire circuit boards priced between $6,000 and $10,000 each. Using the I/O channel in this manner limits the I/O capabilities as a further undesirable side effect. The resulting hardware changes for expansion, if undertaken, are typically dwarfed in magnitude by the software changes needed when applications are to be geographically changed or expanded.



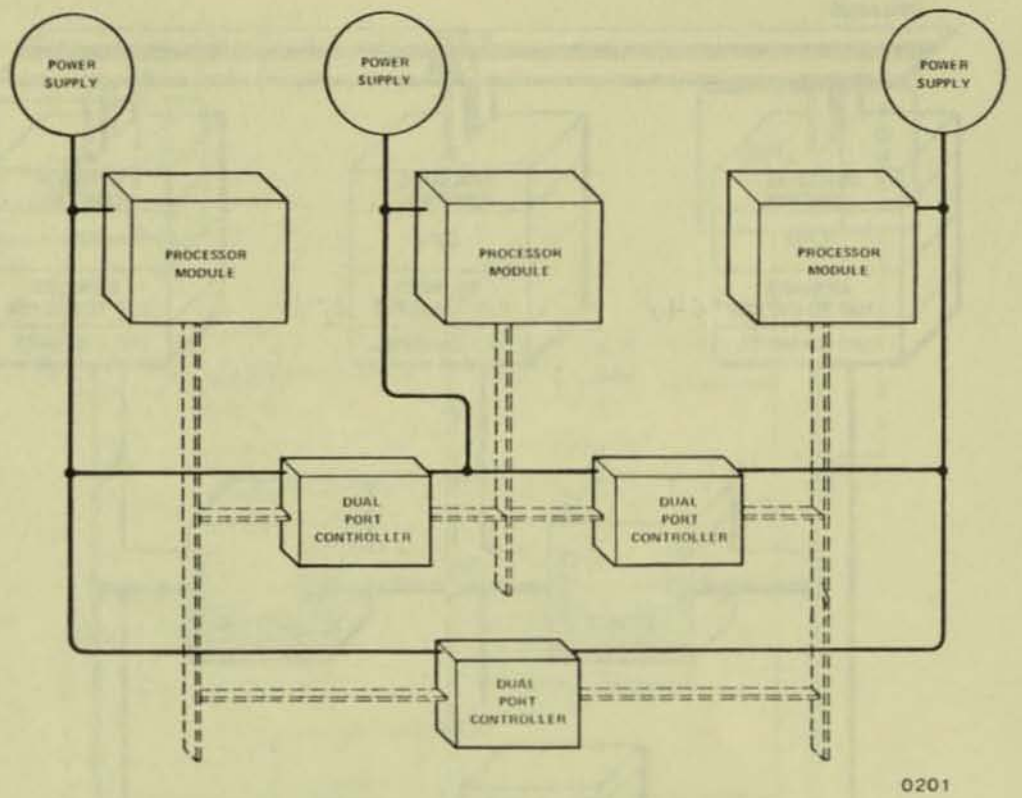
Tandem 16 System Architecture

Figure 2

This paper describes the Tandem 16 architecture at the lowest level (the hardware). Section 1 deals with the overall system organization and packaging. Section 2 explains the processor module organization and its attachment to the interprocessor communications system. Section 3 discusses the I/O system organization. Section 4 discusses power, packaging, and on-line maintenance aspects that are not covered elsewhere in the paper.

1. System Organization

The Tandem 16 NonStop system is organized around three basic elements: the processor module, dual-ported I/O controllers, and the DC power distribution system (Fig. 2,3). The processors are interconnected by a dual-interprocessor bus system: the Dynabus; the I/O controllers are each connected with two independent I/O channels, one to each port; and the power distribution system is integrated with the modular packaging of the system.



Tandem 16 Power Distribution

Figure 3

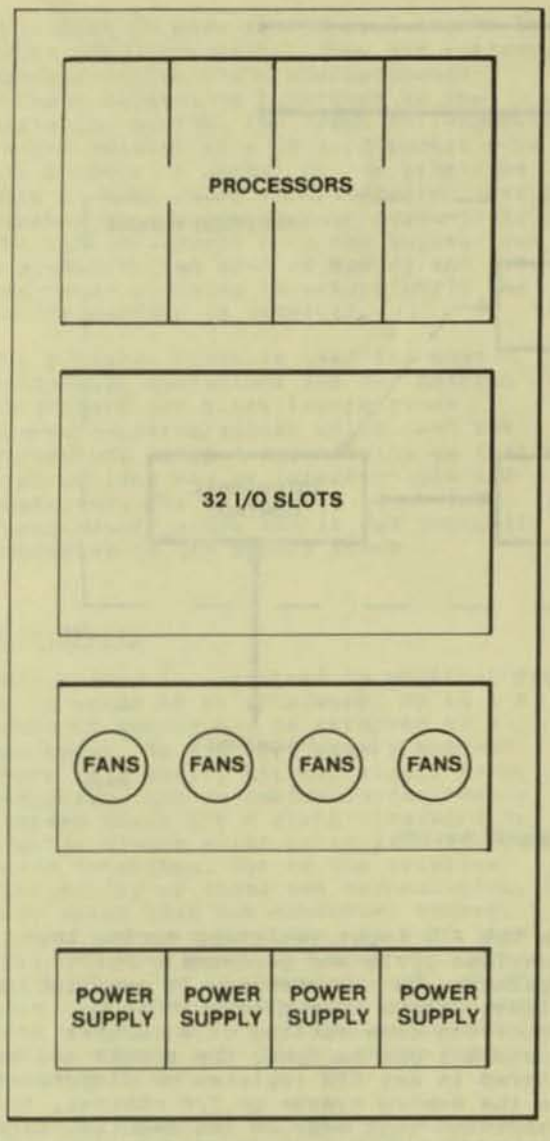
The system design goal is two-fold: (1) to continue operation of the system through any single failure, and (2) to be able to repair that failure without affecting the rest of the system. The on-line maintenance aspects were a key factor in the design of the physical packaging and the power-distribution of the system.

System Packaging

The cabinet (Fig. 4) is divided into 4 sections: the upper card cage, the lower card cage, cooling, and power supplies. The upper card cage contains up to 4 processors, each with up to 2 M bytes of independent main memory. The lower card cage contains up to 32 I/O controller printed circuit (PC) cards, where each controller consists of one to three PC cards. The cooling section consists of 4 fans and a plenum chamber that forces laminar air flow through the card cages. The power supply section contains up to 4 power supply modules. Multiple cabinets may be bolted together and the system has the capability to accommodate a maximum of 16 processors.

Each processor module, consisting of a CPU, memory, Dynabus control and I/O channel are powered by an associated power supply. If a failed module is to be replaced in this section its associated power supply is shut off, the module is replaced, and the power supply is turned on. Each card cage slot in the I/O card cage is powered by two different power supplies. Each of the I/O controllers is connected via its dual-port arrangement to two processors. Each of those processors has its own power supply; usually, but not necessarily, those two supplies are the ones that power the I/O controller (Fig. 3). Each slot in the I/O card cage can be powered down by a corresponding switch disconnecting power from the slot from both supplies without affecting power to the remainder of the system. Therefore, if a power supply fails, or if one is shut down to repair a processor, no I/O controllers are affected.

The dual-power sourcing to the I/O controllers was originally designed using relay switching. This plan was abandonded for several reasons: a) to contend with relay failure modes is difficult; b) the number of contact bounces on a switch-over is neither uniform nor predictable making it difficult for the operating system to handle power-on interrupts from the I/O controllers; and c) during the switch-over, controllers do lose power, and while most controllers are software-restartable, communications controllers hang up their communications



Tandem 16 Physical Cabinet

Figure 4

lines. We therefore devised a diode current sharing scheme whereby I/O controllers are constantly drawing current from two supplies simultaneously. If a power supply fails, all the current for a given controller is supplied by the second power supply. There is also circuitry to provide for a controlled ramping of current draw on turn-on and turn-off so there are no instantaneous power demands from a given supply causing a potential momentary dip in supply voltage.

Both fans and power supplies are electrically connected using quick disconnect connectors to speed replacement upon failure. No tools are required to replace a power supply. A screwdriver is all that is needed to replace a fan. Both replacements take less than 5 minutes.

Interconnections

Physical interconnection is done both using front edge connectors and back-planes. Communication within a processor module (e.g. between the CPU and main memory) takes place over four 50 pin front edge connectors using flat ribbon cable. Interprocessor communication takes place over the Dynabus on the back-plane also utilizing ribbon cable. The I/O controllers use etch trace on the back-plane for communication among PC cards of a multicard controller. The I/O channels are back-plane ribbon cable connections between the processors and the I/O controllers.

Peripheral I/O devices are connected via shielded round cable either to a bulk-head patch panel or directly to the front edge connectors of the I/O controllers. If a patch panel is used, then there is a connection using round cables between the patch panel and the front edge connectors of the I/O controllers.

Power is distributed using a DC power distribution scheme. Physically, AC is brought in through a filtering and phase splitting distribution box. Pigtails connect the AC distribution box to one of the input connectors of a power supply. The DC power from the supply is routed through a cable harness to a laminated bus bar arrangement which distributes power on the back-planes to both processors and I/O controllers.

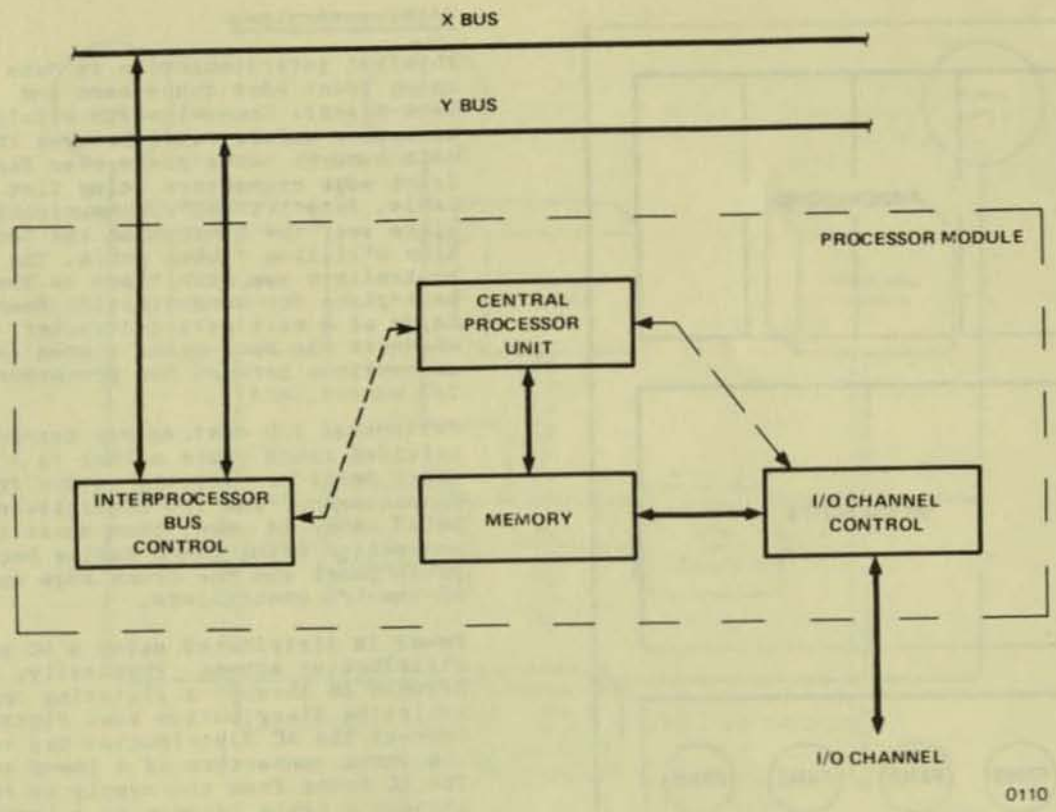
2. Processor Module Organization

The processor (Fig. 5) includes a 16 bit CPU, main memory, the Dynabus interface control, and an I/O channel. Physically the CPU, I/O channel and Dynabus control consists of two PC boards 16 inches by 18 inches, each containing approximately 300 IC packages. Schottky TTL circuitry is used. Up to 2 M bytes of main memory is available utilizing core or semiconductor technology. Core memory boards hold 32K or 128K 17-bit words and each occupy two card slots because of the height of the core stack. Semiconductor memory is implemented utilizing 16 pin, 4K or 16K dynamic RAMs. These memory boards contain 48K and 192K 22-bit words per board, respectively, and occupy only one card slot and are therefore 50% denser than core.

The processor module is viewed by the user as a 16-bit, stack-oriented processor, with a demand paging, virtual memory system capable of supporting multiprogramming.

The CPU

The CPU is a microprogrammed processor consisting of a bank of 8 registers which can be used as general purpose registers, as a LIFO register stack, or for indexing;



Tandem 16 Processor Organization

Figure 5

an ALU; a shifter; two memory stack management registers; program control registers (e.g. program counter, instruction register, environment or status register, and a next instruction register for instruction prefetching); scratch pad registers available only to the microprogrammer; and several other miscellaneous flags and counters for the microprogrammer.

The microprogram is stored in read-only memory and is organized in 512-word sectors of 32-bit words. The microinstruction has different formats for branching, sequential functions, and immediate operand operations. The Tandem 16 instruction set occupies 1024 words with the decimal arithmetic and the floating point options each occupying another 512 words. The address space for the microprogram is 4K words.

The microprocessor has a 100 ns cycle time and is a two stage pipelined microprocessor, i.e., all microinstructions take two cycles to execute but one completes each cycle. In the first stage of the pipeline any two operands are selected by two source fields in the microinstruction for loading into the ALU input registers. In the second stage of the pipeline the ALU performs a primitive operation on the operands placed in the ALU input registers during the previous cycle and performs a shift operation on the results. In parallel, a miscellaneous operation such as a condition code setting or a counter increment can be done, the result can be stored in any CPU register or dispatched to the memory system or I/O channel, and a condition test made on the results. Each of these parallel operations is controlled by a separate control field in the microinstruction.

The basic set of 173 machine instructions includes arithmetic operations (add, subtract, etc.), logical operations (and, or, exclusive or), bit deposit, block (multiple element) moves/compares/scans, procedure calls and exits, interprocessor SENDs, I/O operations, and operating system primitives. All instructions are 16 bits in length. The decimal instruction set provides an additional 32 instructions dealing with four-word operands while the floating point instruction set provides an additional 43 instructions.

The interrupt system has 16 major interrupt levels which include interprocessor bus data received, I/O transfer completion, memory error, interval timer, page fault, privileged instruction violation, etc.

Provision is made for several events to cause microinterrupts. They are entirely handled by the CPU's microprocessor without causing an interrupt to the operating system. One event for example, is the receipt of a 16 word packet over the Dynabus. A packet is the primitive unit of data which is transferred over the Dynabus for interprocessor communication. The microprocessor puts the information in a predetermined area of memory and does not cause a system interrupt until the entire message is received.

The register stack is used for most arithmetic operations and for holding parameters for block instructions (moves/compares/scans) which need the parameters updated dynamically so that the instructions may be interruptable and restarted. The 8-register stack is a "wraparound" stack and is not logically connected to the memory stack.

Main Memory

Main memory is organized in physical pages of 1K words of 16 bits/word. Up to 1 M words of memory may be attached to a processor. In the core memory systems there is a parity bit for single error detection, and in semiconductor memory systems there are 6 check bits/word to provide single error correction and double error detection. Due to the relative reliability of these two technologies, we have found that semiconductor memory, without error correction, is much less reliable than core, and that with error correction, it is somewhat more reliable than core. Battery backup provides short term non-volatility to the semiconductor memory system for utility power outage considerations.

It might be noted that there are some memory systems using a 21 bit error correction scheme (5 check bits on a 16 bit data word instead of 6). While 5 bits are enough to correct all single bit errors, it does not detect approximately 1/3 of the possible double bit error combinations. In these conditions, this 5 check bit scheme will incorrectly deduce that some bit (neither of the bits actually in error) is incorrect and correctable. The scheme will then correct this bit (actually causing 3 bits to be in error), and deliver it to the system as "good" reporting a correctable memory error.

Memory is logically divided into 4 address spaces (Fig. 6). These are the virtual address spaces of the machine; both the system and the user have a code space and a data space. The code space is unmodifiable and the data space can be viewed either as a stack or a random access memory, depending on the addressing mode used. Each of these virtual address spaces are 64K words long addressed by a 16 bit virtual address.

The physical memory address is 20 bits with conversion from the virtual address to physical address accomplished through a mapping scheme. Four maps are provided, one for each logical address space; each map consists of 64 entries one for each page in the virtual address space. The maps are implemented in 50 ns access bipolar static RAM. The map access and main memory error correction is included in the 500 ns cycle time for semiconductor memory systems.

The unmodifiable code area provides reentrant, recursive, and sharable code. The data space (Fig. 7) can be referenced relative to address 0 (global data or G+ addressing), or relative to the memory stack management registers in the CPU.

The lowest level language provided on the Tandem 16 system is T/TAL, a high-level, block-structured, ALGOL-like language which provides structures to get at the more efficient machine instructions. The basic program unit in T/TAL is the PROCEDURE. Unlike ALGOL, there is no outer block, but rather a main PROCEDURE. T/TAL has the ability to declare certain variables as global. PROCEDURES cannot be nested in T/TAL, but a SUBPROCEDURE can be nested in a PROCEDURE and only in a PROCEDURE. A SUBPROCEDURE is limited in local variable access capabilities.

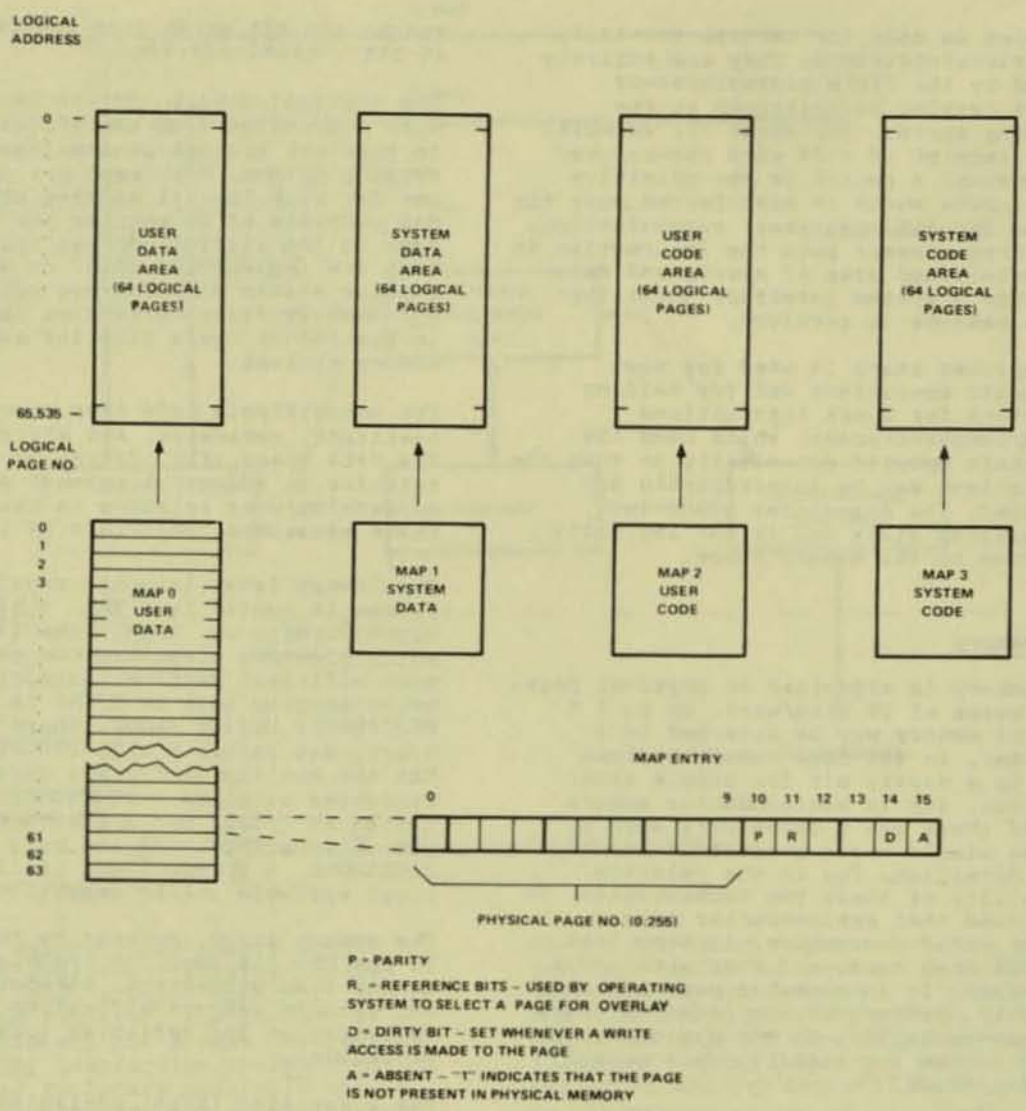
The memory stack, defined by two registers in the CPU, is used for efficient linkage to and from procedures, parameter passing, and dynamic storage allocation and deallocation for variables local to the procedure.

The L register (Local variables) points to the last stack marker placed on the stack. This marker contains return information about the caller such as the return address and the previous location of the L register. The contents of the L register are primarily changed by the procedure call and exit instructions.

Addressing relative to the L register provides access to parameters passed to a procedure (L-) and local variables of the procedure (L+). Parameters may be passed either by value (using direct addressing) or by reference (using indirect addressing).

The S register (stack top pointer) points to the last element placed on the stack. It is used for a SUBPROCEDURE's sublocal data area when S relative addressing (S-) is used.

There is a special mode of addressing used by the operating system, called System Global (SG+) addressing. It is used by the operating system while it is working in a



Tandem 16 Logical Memory Address Spaces

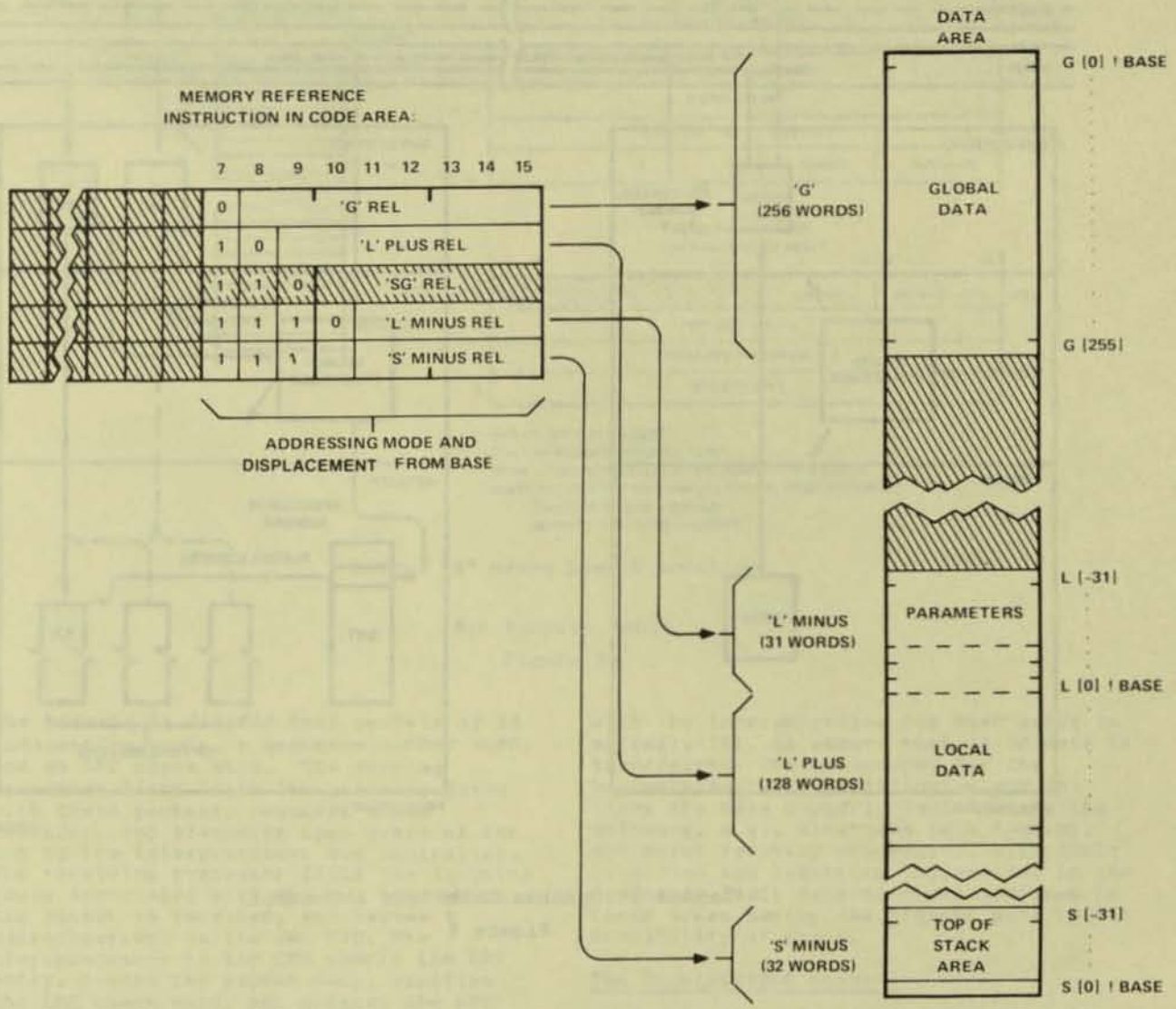
Figure 6

user's virtual data space (on his behalf) and needs to address the system data space. The system data space contains many resource tables and buffers and the need to access them quickly justifies the existence of this addressing mode.

There are three tables known to the operating system, the microprogram and the hardware: the system interrupt vector (SIV), the I/O Control (IOC) table, and the Bus Receive Table (BRT). These tables will be explained in later sections as appropriate.

The Dynabus

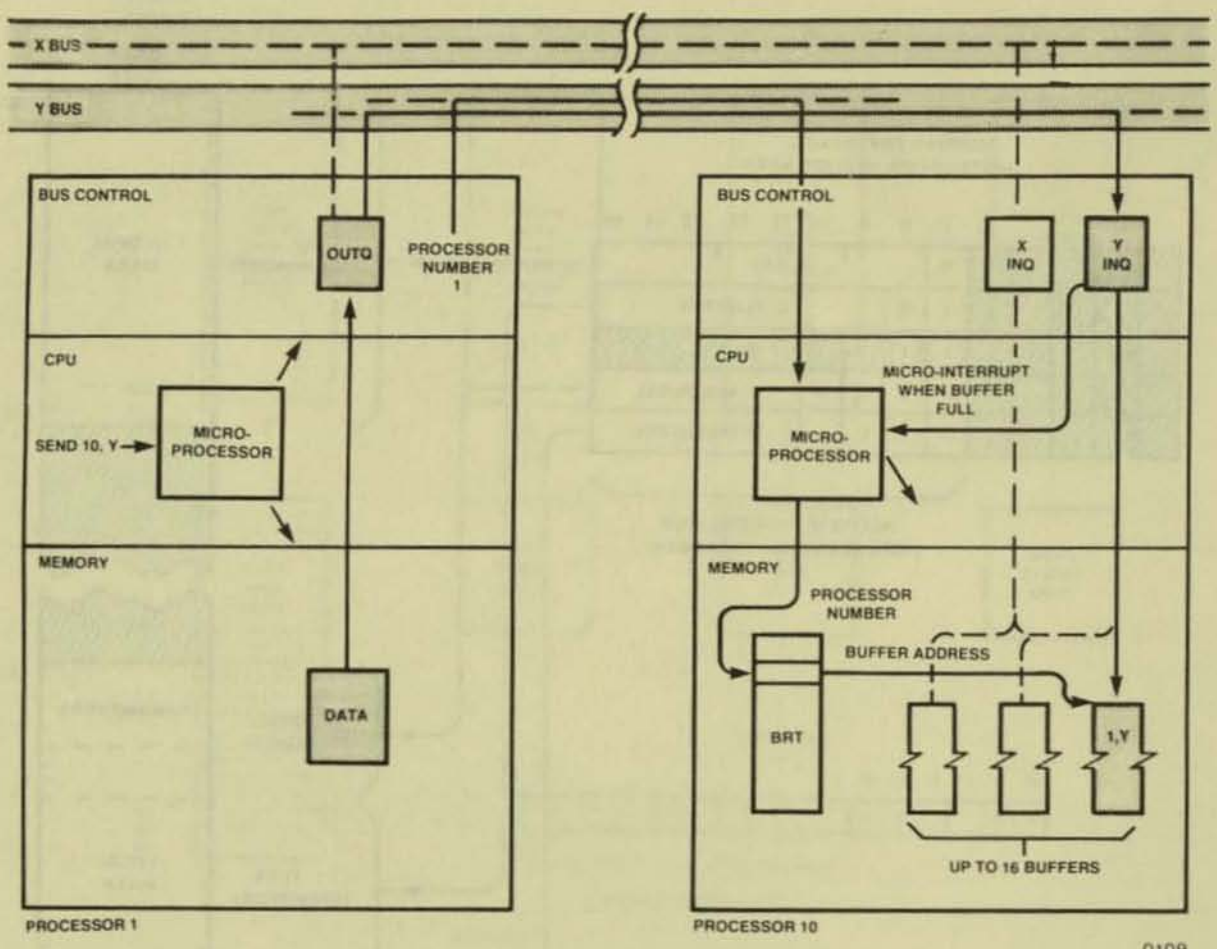
The Dynabus is a set of two independent interprocessor buses. Bus access is determined by two independent interprocessor bus controllers. Each of these controllers is dual-powered, in the same manner as an I/O controller. The Dynabus controllers are very small, approximately 30 IC packages, and are not associated with, nor physically a part of any processor. Each bus has a two byte data path and control lines associated with it. There are two sets of radial



Tandem 16 Data Space

Figure 7

connections from each interprocessor bus controller to each processor module. They distribute clocks for synchronous transmission over the bus and for transmission enable. Therefore, no failed processor can independently dominate Dynabus utilization upon failure since in order to electrically transmit onto the bus, the bus controller must agree that a given processor has the right to transmit. Each bus has a clock associated with it, running independently of the processor clocks and located on the associated bus controller. The clock rate is 150 ns on two to eight processor systems. The clock does need to be slowed down for the longer interprocessor buses of greater than eight processors. Therefore each bus on small systems transfers at the rate of 13.3M bytes/second and on the larger systems at 10 M bytes/second. Performance measurements have shown that under worst case test conditions the Dynabus is only 15% utilized in a ten processor system.

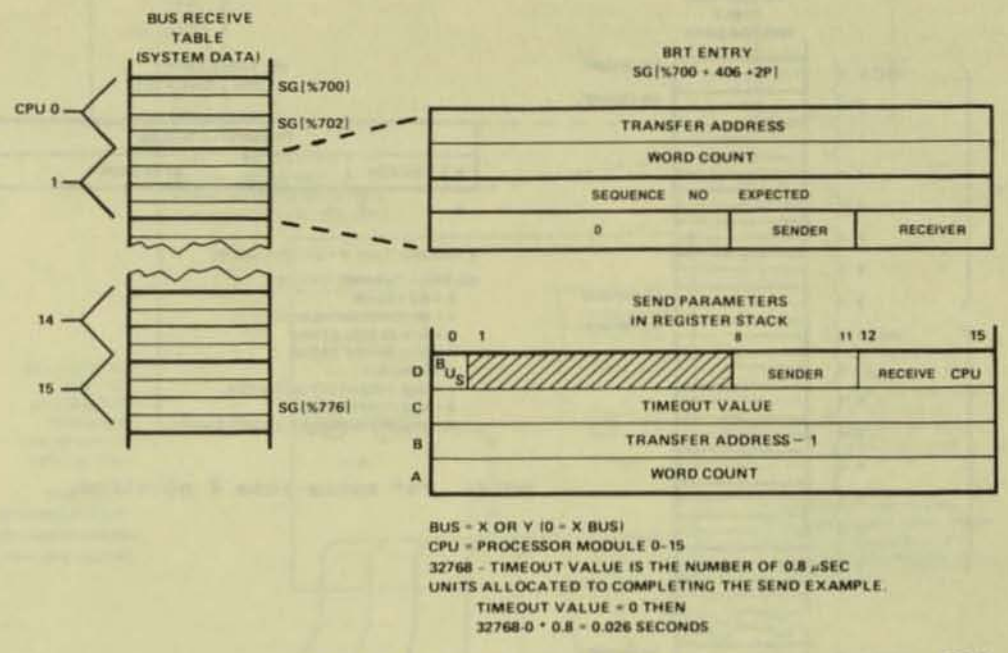


Tandem 16 Dynabus Interface & Control

Figure 8

Each processor in the system attaches to both interprocessor buses. The Dynabus interface control section (Fig. 8) consists of 3 high speed caches: an incoming queue associated with each interprocessor bus, and a single outgoing queue that can be switched to either of the buses. All caches are 16 words in length and all bus transfers are cache to cache. All components that attach to either of the buses are kept physically distinct, so that no single component failure can contaminate both buses simultaneously. Also in this section are clock synchronization and interlock circuitry. All processors communicate in a point to point manner using this redundant direct shared bus (DSB) configuration [3].

For any given interprocessor data transfer, one processor is the sender and the other the receiver. Before a processor can receive data over an interprocessor bus, the operating system must configure an entry in a table (Fig. 9) known as the Bus Receive Table (BRT). Each BRT entry contains the address where the incoming data is to be stored, the sequence number of the next packet, the processor number of the sender and receiver, and the number of words expected. To transfer data over a bus, a SEND instruction is executed in the sending processor, which specifies the bus to be used, the intended receiver, and the number of words to be sent. The sending processor's CPU stays in the SEND instruction until the data transfer is completed. Up to 65,535 words can be sent in a single SEND instruction. While the sending processor is executing the SEND instruction, the Dynabus interface control logic in the receiving processor is storing the data away according to the appropriate BRT entry. In the receiving processor this occurs simultaneously with program execution.



NOTE: "%" means base 8 notation. 0205

Bus Receive Table

Figure 9

The message is divided into packets of 14 information words, a sequence number word, and an LRC check word. The sending processor first fills its outgoing queue with these packets, requests a bus transfer, and transmits upon grant of the bus by the interprocessor bus controller. The receiving processor fills the incoming queue associated with the bus over which the packet is received, and issues a microinterrupt to its own CPU. The microprocessor of the CPU checks the BRT entry, stores the packet away, verifies the LRC check word, and updates the BRT entry accordingly. If the count is exhausted the currently executing program is interrupted, otherwise program execution continues.

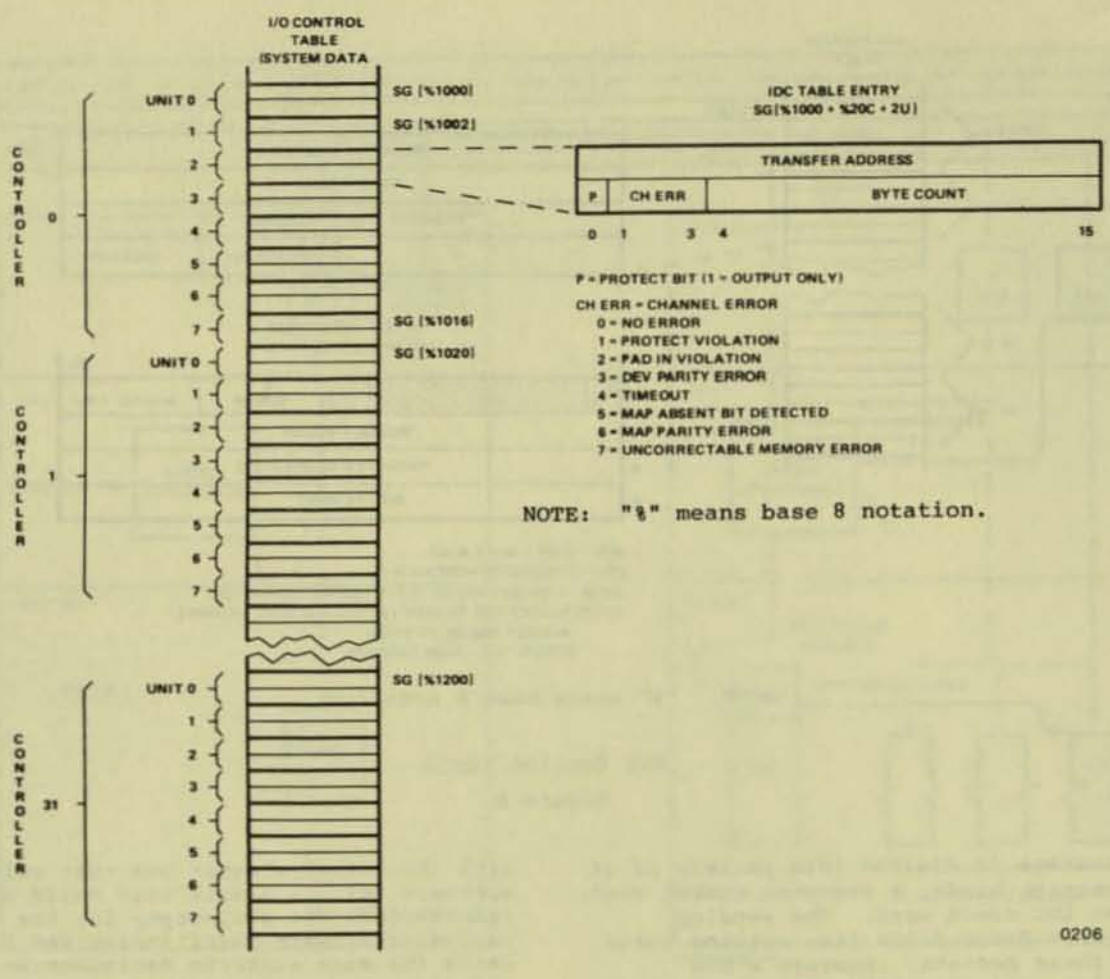
The BRT entries are four words that include a transfer count buffer address, sequence number expected and the sender and receiver CPU numbers. The SEND instruction has as parameters the designation of the bus to be used, the intended receiver, the data buffer address in the system data space, the word count to be transferred, and a timeout value. Error recovery action is to be taken in case the transfer is not completed within the timeout interval. These parameters are placed on the register stack and are dynamically updated so that the SEND instruction is interruptable on packet boundaries.

There are several levels of protocol, beyond the scope of this paper, dealing with the interprocessor bus that exist in software [4], to assure that valid data is transferred. The philosophy for the hardware/software partitioning was to leave the more esoteric decisions to the software, e.g., alternate path routing, and error recovery procedures, with fault detection and reporting implemented in the hardware. Fault detection was designed in those areas having the highest anticipated probability of error.

The Input/Output Channel

The heart of the Tandem 16 I/O System is the I/O channel. All I/O is done on a direct memory access (DMA) basis. The channel is a microprogrammed, block multiplexed channel with the block size determined by the individual controllers. All the controllers are buffered to some degree so that all transfers over the I/O channel are at memory speed (4 M Bytes/ Second) and never wait for mechanical motion since the transfers always come from a buffer in the controller, rather than from the actual I/O device.

There exists a table in the system data space of each processor called the IOC (I/O Control) table that contains a two word entry (Fig. 10) for each of the 256 possible I/O devices attached to the I/O channel. These entries contain a byte count and virtual address in the system data space for data transfers from the I/O system.



NOTE: "%" means base 8 notation.

I/O Control Table

Figure 10

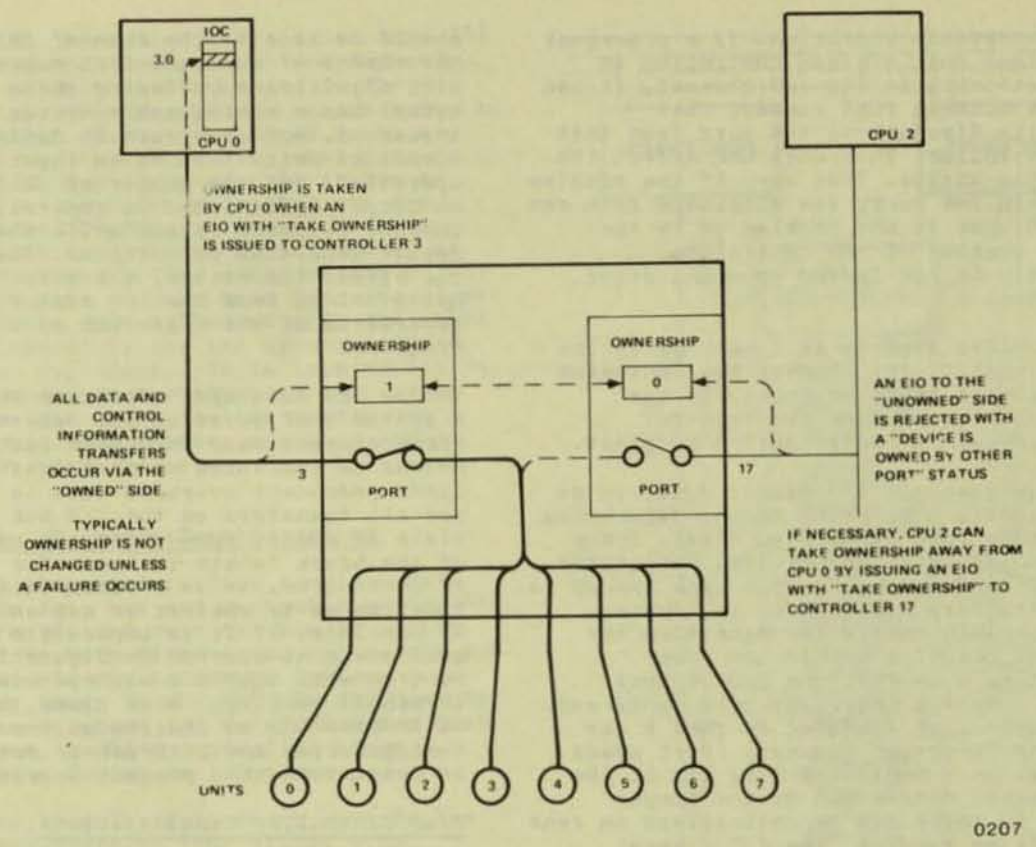
The I/O channel moves the IOC entry to active registers during connection of an I/O controller and restores the updated values to the IOC upon disconnection. The I/O channel alerts the I/O controller when the count has been exhausted and that causes the controller to interrupt the processor.

The channel does not execute channel programs as on many systems but it does do data transfer in parallel with program execution. The memory system priority always permits I/O accesses to be handled before CPU or Dynabus accesses (in an on-line, transaction oriented environment, it is rare that a system is not I/O bound). The maximum I/O transfer is 4K bytes.

3. I/O System Organizaion

The I/O system had a design goal of being very efficient in a transaction, on-line oriented environment. This environment has constraints different from those of a batch environment. The figure of merit in an on-line system is the number of transactions/second/dollar that can be handled by the system. We also wanted an I/O system that had low overhead, fast transfer rates, no overruns, and no interrupts to the system until a logical entity of work was completed (e.g., no character by character interrupts from the terminals). The resulting design satisfied these goals by implementing an I/O system that was extremely simple.

I/O controllers reconnect to the channel when their buffers are stressed past a configurable threshold, transfer data in a burst mode until their buffer stress is zero (buffer empty on input operations, full on output operations), and disconnect from the channel. When the transfer terminates the I/O controller interrupts the processor. Controllers may interrupt for other reasons than an exhausted byte count, e.g., a terminal controller receiving an end-of-page character from a page mode terminal, or I/O channel error condition, or a disc pack being mounted.



Ownership Circuitry

Figure 11

Dual-Port Controllers

The dual-ported I/O device controllers provide the interface between the Tandem 16 standard I/O channel and a variety of peripheral devices using distinct interfaces. While the I/O controllers are vastly different, there is a commonality among them that folds them into the Tandem 16 NonStop architecture.

Each controller contains two independent I/O channel ports implemented by IC packages which are physically separate from each other so that no interface chip can simultaneously cause failure of both ports. Each port of each controller has a 5-bit configurable controller number, and interrupt priority setting. These settings can be different on each port. The only requirement is that each port attached to an I/O channel must be assigned a controller number and priority distinct from controller numbers and priorities of other ports attached to the same I/O channel.

Each controller has a PON (power-on) circuit which clamps its output to ground whenever the controller's DC supply voltage is not within regulation. The PON circuit has hysteresis in it so that it will not oscillate if the power should hover near the limit of regulation. When the power is within regulation, the output of the PON circuit is at a TTL "1" level. A power-on condition causes a controller reset and also gives an interrupt to one of the two processors to which it is attached. The output of the PON circuit is also used to enable all the I/O channel bus transceivers so that a controller being powered down will not cause interference on the I/O channels during the power transient. This is possible because the PON circuit operates with the supply voltage as low as .2 volts and special transceivers are used which correctly stay in a high impedance state as long as the control enable is at a logical "0".

Logically only one of the two ports of an I/O controller is active and the other port is utilized only in the event of a path failure to the primary port. There is an "ownership" bit (Fig. 11) indicating to each port if it is the primary port or the alternate. Ownership is changed only by the operating system issuing a TAKE OWNERSHIP I/O command. Executing this special command causes the I/O controller to swap its primary and alternate port designation and to do a controller reset. Any attempt to use a controller which is not owned by a given processor will result

in an ownership violation. If a processor determines that a given controller is malfunctioning on its I/O channel, it can issue a DISABLE PORT command that logically disconnects the port from that I/O controller. This does not affect the ownership status. That way, if the problem is within the port, the alternate path can be used, but if the problem is in the common portion of the controller, ownership is not forced upon the other processor.

A controller signals an interrupt on the I/O channel if the channel has indicated an exhausted transfer count, if the controller terminates the transfer prematurely, or for attention purposes.

When simultaneous interrupts occur on an I/O channel, a priority scheme determines which interrupt is handled first. There are two levels of priorities, designated "rank 0" and "rank 1". Each rank has up to 16 controllers assigned to it. Jumper wires on each controller determine the rank and position within the rank (positions 0 to 15). The I/O channel issues a rank 0 interrupt poll cycle and each controller assigned to rank 0 can place an interrupt request, if it needs service, on a dedicated data bit of the I/O channel determined by the jumper wires. If there are no controllers on rank 0 requiring service, the I/O channel issues the interrupt poll cycle for rank 1. Note, only 32 controllers can be assigned to a given channel and each one has a unique rank and position designation. The highest priority controller is granted access to the interrupt system. Thus a radial polling technique allows the processor to resolve 32 different controller priorities in just two poll cycles. Each port of a controller has a separate set of configuration jumpers so that a controller can have different priorities on its primary and alternate path.

Controller Buffer Considerations

In the design of the Tandem 16 I/O system, a lot of attention was paid to the overrun problem. While overruns are possible on this system, they have been made a rare occurrence. Each I/O controller has 3 configurable settings: the I/O controller number, the interrupt priority, and buffer stress threshold reconnect setting.

Each I/O controller is buffered to some extent. The asynchronous terminal controller has 2 bytes of buffering, while the disc controller has 4K bytes of buffering. Considerations of device transfer rate, channel transfer rate, the individual controller's buffer depth, the controller's reconnect priority, and a given channel's I/O complement can be used to determine the buffer's depth (stress threshold) at which a reconnect request should be made to the channel to minimize the chance of overrun. Each controller with significant buffering (more than 32 bytes) has a configurable stress threshold. Buffer stress is defined as the number of cells full on an input operation, and the number of cells empty on output operations. In general, the I/O channel relieves stress while the I/O device generates more stress. Therefore the higher the stress, the more the buffer needs relief from the I/O channel, regardless of the direction of data transfer.

Tandem has developed a program which takes a system configuration and determines the appropriate stress threshold settings needed to guarantee no data overruns. Since reconnect overhead time is known, and all transfers on the I/O bus take place at memory speed, and the upper bound of the block length is known for each type of controller, it is a deterministic function as to whether or not an overrun is possible. If it is impossible to generate a no-overrun configuration, the program will output a minimum-overrun threshold settings. Most times, however, it is possible to iterate on the configuration until threshold settings can be determined that prevent overruns.

Disc Controller Considerations

The greatest fear that an on-line system user has is that "the data base is down" [5]. Many of these users are willing to pay the premium of having duplicated or "mirrored" data bases in case a disc drive fails. To meet this requirement, Tandem provides automatic mirroring of data bases.

A disc volume is a set of data contained on one spindle or one removable disc pack. A user may declare any of the disc volumes as mirrored pairs at system generation time (Fig. 12). The system then maintains these pairs so they always contain identical data. Thus protection is achieved for a single drive failure. Each disc drive in the system may be dual-ported. Each port of a disc drive is connected to an independent disc controller. Each of the disc controllers are also dual-ported and connected between two processors. A string of up to 8 drives (4 mirrored pairs) can be supported by a pair of controllers in this manner.

Note that in this configuration there are many paths to any given data and that data can be retrieved regardless of any single disc drive failure, disc controller failure, power supply failure, processor failure, or I/O channel failure.

The disc controller is buffered for a maximum length record which provides several features important in an on-line system. For example, the disc controller is absolutely immune to overruns.

This disc controller uses a Fire code [6] for burst error correction and detection. It can correct 11 bit bursts in the controller's buffer before transmission to the channel. Since overlapped seeks are allowed by the controller, when data is to be read from a mirrored pair it can be read from the drive which has its arm closest to the data cylinder. This is accomplished by using "split seeks," a SYSGEN parameter that requires one of the mirrored pair to only read from the first half of the disc cylinders with the other disc responsible for the second half of the disc cylinders. It is interesting to note that since the majority of transactions in an on-line system are reads, mirrored volumes actually can increase performance.

NonStop I/O System Considerations

The I/O channel interface consists of a two byte data bus and control signals. All data transferred over the bus is parity checked in both directions, and errors are reported via the interrupt system. A watchdog timer in the I/O channel detects if a non-existent I/O controller has been addressed, or if a controller stops responding during an I/O sequence.

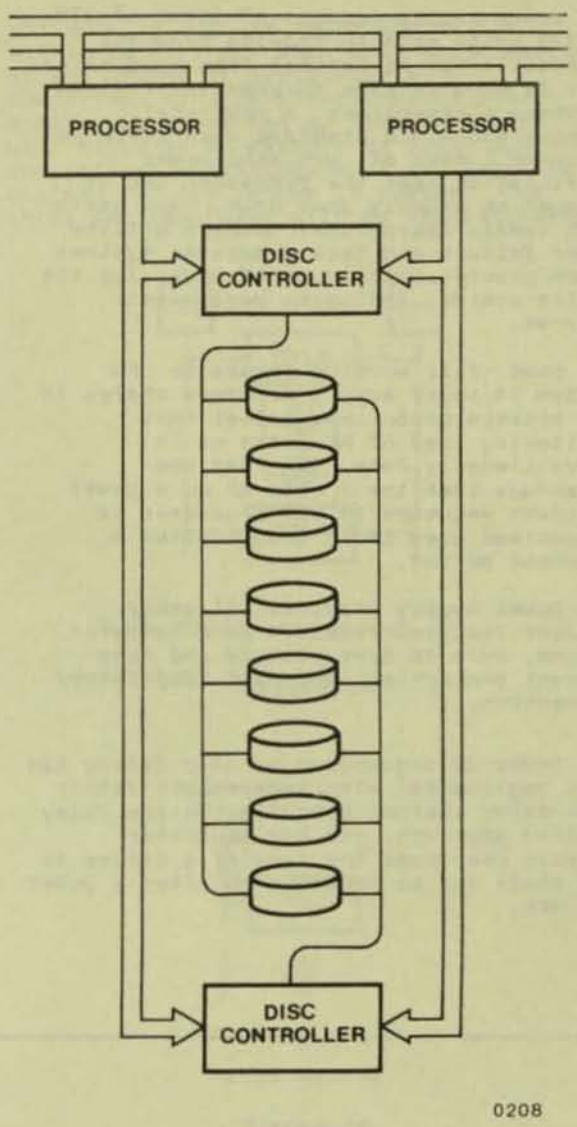
The data transfer byte count word in the IOC entry contains four status bits including a protect bit. When this bit is set to "1" only output transfers are permitted to this device.

Because I/O controllers are connected between two independent I/O channels, it is very important that word count, buffer address, and direction of transfer are controlled by the processor instead of within the controller. If that information were to be kept in the controller, a single failure could cause both processors to which it was attached to fail. Consider what would happen if a byte count register was located in the controller and was stuck in such a situation such that the count could not decrement to zero on an input transfer. It would be possible to overwrite the buffer and cause system tables to become meaningless. The error would propagate to the other processor upon discovery that the first processor was no longer operating.

Other error conditions that the channel checks for are violations of I/O protocol, attempts to transfer to absent pages (it is the operating system's responsibility to "tack down" the virtual pages used for I/O buffering), uncorrectable memory errors, and map parity errors.

4. Power, Packaging, On-line maintenance

The Tandem 16 power supply has 3 sections: a 5 volt interruptable section, a 5 volt uninterruptable section, and a 12-15 volt



Tandem 16 Disc Subsystem Organization

Figure 12

uninterruptable section. The interruptable section will stop supplying DC power when AC is lost while the uninterruptable sections will continue to supply DC power. The interruptable section powers I/O controllers and that portion of a processor which is not related to memory refresh operation. The uninterruptable sections provide power for the memory array and refresh circuitry. The 5 volt sections are switching regulated supplies while the 12-15 volt section is linearly regulated. The uninterruptable sections have a provision for a battery attachment so that in case of utility power failure, memory contents are kept for 1.5 to 4 hours, depending on the amount of memory attached to the supply.

The power supply accepts AC input of 110 or 220 volts ±20% to provide brownout insensitivity. At nominal line conditions, over 30 msec of ride through is provided by storage capacitors. A power-fail warning signal is provided when there is at least 5 msec of regulated power remaining so that the processor can go through an orderly shut down. Some users must remain operational through utility power failure and have generator systems which provide continuous AC power for the entire system, including peripheral devices.

The power-fail warning scheme in the Tandem 16 power supply monitors charge in the storage capacitors rather that monitoring loss of AC peaks as is conventionally done. This has the advantage that the 5 msec to do a power shutdown sequence in the processor is guaranteed even if it occurs after a brownout period.

The power supply provides all other prudent features required in a computer system, such as over voltage and over current protection, and over temperature protection.

The power-up sequencing on disc drives has been implemented with independent rather than daisy chained circuits. In the daisy chained approach, one bad sequencer circuit can cause the remaining drives in the chain not to sequence up after a power failure.

Further Packaging and On-line Maintenance Considerations

Modularity is a key concept in the Tandem 16 system. The maintenance philosophy is to make all repair by module replacement at the user site without making the system unavailable to the user. Therefore the backplanes, power supplies, fans, I/O channels, as well as the PC cards are modular and easily replaceable. Thumb screws are used when they can be so that a minimum of tools are needed for repair. The package is designed so that there is easy access to all modules.

Processors and I/O controllers not only can be replaced on-line, but added on-line without system interruption if expansion is planned, all without application software being changed.

Summary

The contribution of the Tandem 16 system lies in the synthesis of a system to directly address the need of the NonStop application marketplace. By avoiding the "onus of compatibility" to any previous system, an architecture could be designed from "scratch" that was "clean" and efficient.

The system goals have been met to a large degree. Systems have been installed containing two to twelve processors. Many application programs are on-line and running. They recover from failures, and stay up continuously.

Biography

James A. Katzman is a founder and Vice President of Marketing Support for Tandem Computers. From Tandem's start in November of 1974 through mid 1978, Mr. Katzman held the post of Vice President of Engineering and is one of the principal architects of the Tandem 16 NonStop system.

Previously Katzman was responsible for the design of the integrated I/O channel for the Amdahl 470V/6 system while at Amdahl Corp. from 1971 to 1974. While at Hewlett-Packard Company from 1968 to 1971 he was one of the principal architects of the H-P 3000 computer system.

Mr. Katzman holds patents on all of the above machines. He is a member of the ACM and IEEE. Academically he holds a BSEE from Purdue University and a MSEE from Stanford University. He is a member of Tau Beta Pi, Eta Kappa Nu, and Omicron Delta Kappa honorary societies. He is listed in the 1978-1979 edition of Who's Who in the West.

APPENDIX A

The Tandem 16 system provides its high availability through architecture. In the literature [7,8] we find that availability ranges between 0 and 1 and is defined as:

$$A = \frac{MTBF}{MTBF+MTTR} \qquad (1)$$

where

A = Availability
MTBF = Mean Time Between Failure
MTTR = Mean Time To Repair

The availability of two redundant systems where only one is required is represented by:

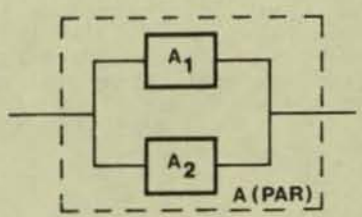


Figure 13

and the parallel system, A(PAR), has an availability of

$$A(PAR) = A_1 + A_2 - A_1 A_2 \qquad (2)$$

If $A_1 = A_2 = A$ then,

$$A(PAR) = 2A - A^2 \qquad (3)$$

When subsystems in series are required for operation, the system is represented by:

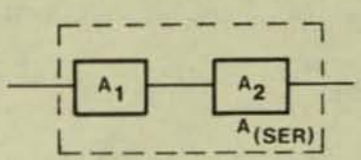
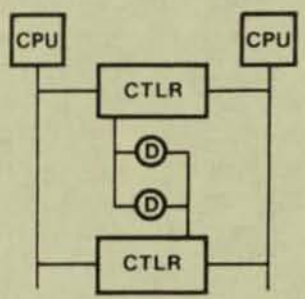


Figure 14

and the series systems, A(SER), has an availability of

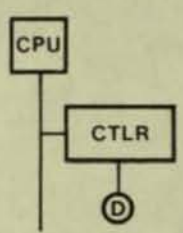
$$A(SER) = A_1 A_2 \qquad (4)$$

While it is not the intention of the author to give any more than these basics of the theory of Availability, a comparison of 3 architectures of disc subsystems connected to host computers will serve as an example to demonstrate the order-of-magnitude more availability claimed for the Tandem 16 system. The three architectures will be the following:



Tandem 16 System

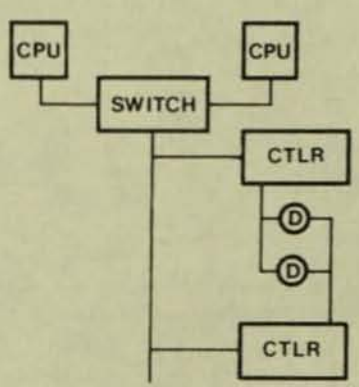
Figure 15



Batch System

Figure 16

and the typical "fault-tolerant" system



0209

"f-t" System

Figure 17

The availability models for the three systems are:

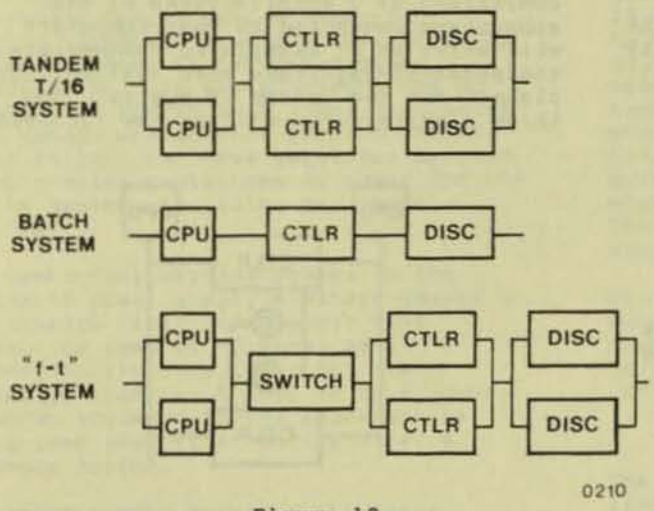


Figure 18

Assuming the MTBF of all similar components to be equal:

CPU	9,000 hours
CTLR	12,000 hours
DISC	4,000 hours
SWITCH	15,000 hours

and the MTTR for any failure to be a conservative 24 hours, the availability of these systems are:

One CPU	= .997340426	(5)
Parallel CPUs	= .999992927	(6)
One CTLR	= .998003992	(7)
Parallel CTLRs	= .999996016	(8)
One Disc	= .994035785	(9)
Parallel Discs	= .999964429	(10)
One Switch	= .998402556	(11)

Tandem 16 = (.999992927)(.999996016)(.999964229) (12)
= .999953172 (13)

Batch System = (.997340426)(.998003992)(.994035785) (14)
= .989413082 (15)

"f-t" System = (.999992927)(.998402556)(.99996016)(.999964229) (16)
= .998355803 (17)

Solving (1) for MTBF we get

$$MTBF = \frac{MTTR\ (A)}{1-A} \qquad (18)$$

Again assuming MTTR = 24 hours, the MTBF for the above systems are:

Tandem 16	Batch System	"f-t" system	
512,490	2,243	14,573	hours
= 21,353	= 93	= 607	days
= 58.4	= 0.25	= 1.66	years

The Tandem 16 architecture provides 35 times the MTBF of the typical "fault-tolerant" system architecture and 233 times that of the typical batch system. In this analysis it was assumed that dual controllers and dual ported discs were used, and that the two volumes were kept identical in each system but the batch system.

Tandem has completed extensive computer modeling of architectures. Empirical observations have substantiated our modeling data and product claim: the Tandem 16 architecture does, in fact, provide an order-of-magnitude more availability than any past commercially available systems. The results seen here in this appendix, however, would not be observed normally on any of the systems mentioned. There are assumptions made which make these calculations unrealistic: all faults are not independant as assumed, faults do go undedected for long periods of time, and so forth. What this exercise does prove is that this architecture does provide a vehicle for order-of magnitude improvement in availability which is empirically observable.

References

[1] Katzman J.A., "System Architecture for NonStop Computing," Compcon, February 1977, pp 77-80.

[2] Tandem Computers Inc., Tandem/16 System Description, 1976.

[3] Anderson, G.A.; and Jensen E.D.; "Computer Interconnection Structures: Taxonomy Characteristics and Examples," ACM Computing Surveys, December 1975, pp 197-215.

[4] Bartlett, J.F., "A 'NonStop' Operating System," Hawaii International Conference of System Sciences, January 1978, this volume.

[5] Dolotta, T.A.; Bernstein, M.I.; Dickson, R.S. Jr.; France, N.A.; Rosenblatt, B.A.; Smith, D.M.; and Steel, T.B. Jr.; Data Processing in 1980-1985, John Wiley & Sons, 1976.

[6] Peterson, W.W., Error Correcting Codes, The MIT Press, 1961, pp 183-199.

[7] U.S. Department of Defense, Military Standardization Handbook: Reliability Prediction of Electronic Equipment (MIL-HDBK-217B), 1965, Appendix A.

[8] Locks, M.O., Reliability, Maintainability, & Availability Assessment, Spartan Books/Hayden Book Company, 1973.