A TIGHTLY COUPLED DISTRIBUTED SYSTEM FOR FLIGHT SIMULATORS

John R. Bocskor
Robert J. Cichon, Sr.
Gould Inc., Information Systems
Computer Systems Division
6901 West Sunrise Boulevard
Ft. Lauderdale, Florida 33313-4499

ABSTRACT

To attain the realism necessary for simulation today, higher and higher system fidelity is required. Initially, all simulation software was controlled and executed on a monolithic processor that had to complete execution of all software modules within a specified time frame. As simulation requirements increased, it became evident that portions of the simulation software could be executed in parallel. To meet the requirements for increased fidelity in simulators being designed today, the software has been divided into several cooperating modules. These modules generally load and execute in a number of computers connected by a portion of common physical memory referred to as shared memory. These conventional shared memory systems are typically used in cases where true parallel processing takes place. The shared memory system allows for high-speed coupling of computers which in turn allows higher frame rates thus better fidelity. A new method of tightly coupling multiple computer systems without the inherent deficiencies of conventional shared memory was needed. In addition, a new hardware implementation that utilizes gate array technology and a means of controlling such a system from a designated Host System are required.

INTRODUCTION

In the four sections that follow--Shared Memory Hardware, Computer System Hardware, Software Control, and Diagnostic Software--the authors will discuss the problems encountered and the solutions arrived at in developing a tightly coupled, distributed system for flight simulators.

In brief, the problems were threefold: 1)getting beyond the conventional shared memory approach, 2), "shrinking" the footprint of the system while enhancing its compute power, and 3), reducing the life-cycle cost.

SHARED MEMORY HARDWARE

A. Conventional Shared Memory
(see figure 1)

With this approach there is typically a Shared Memory Interface Controller connected to each node's main bus. In turn, each node is cabled to the Shared Memory Controller located in the shared memory chassis. The Shared Memory Controller actually arbitrates access to the shared memory bus on a cyclic rotating priority basis. Propagation delays on reads and writes of data in the shared memory area occur through each layer of hardware in this type of system. As distance between the shared memory chassis and the nodes is increased then additional propagation delays occur due to line delays.

Compute nodes are granted access to the Shared Memory Bus by order of priority. At the time of the start of the grant cycle only those nodes currently requesting shared data are serviced. Any other requests to the Shared Memory Bus have to wait until the next grant cycle, even though a higher priority node may be requesting the Shared Memory Bus. This type of arbitration is typical for most shared memory systems and ensures all nodes get access to the Shared Memory Bus.

In a conventional shared memory system all nodes connected to the Shared Memory Bus are contending for the same physical module of shared memory. In a heavily loaded memory intensive system the nodes are waiting for memory a high percentage of time. All nodes in the system are accessing this module during reads and writes of shared memory data. Contention could be reduced if the shared memory bandwidth was high enough to handle the aggregate demand of all users. These high bandwidth memories are not economically viable on todays computer systems.

Conventional shared memory systems are housed in a separate shared memory chassis and require a separate power source in addition to the other power supplies in the computer system. In some cases the shared memory chassis and power supplies can not be configured in the same cabinetry as the computer system. An additional cabinet has to be added to house the shared memory system. Spares requirements and system reliability of a conventional shared memory system are affected because the power source, the chassis and the shared memory module are typically uncommon with the rest of the computer systems hardware.

A sophisticated central clocking system is required to synchronize all nodes connected to the conventional shared memory system. A master clock is used to synchronize all the nodes' main busses as well as the Shared Memory Bus.

CONVENTIONAL SHARED MEMORY APPROACH

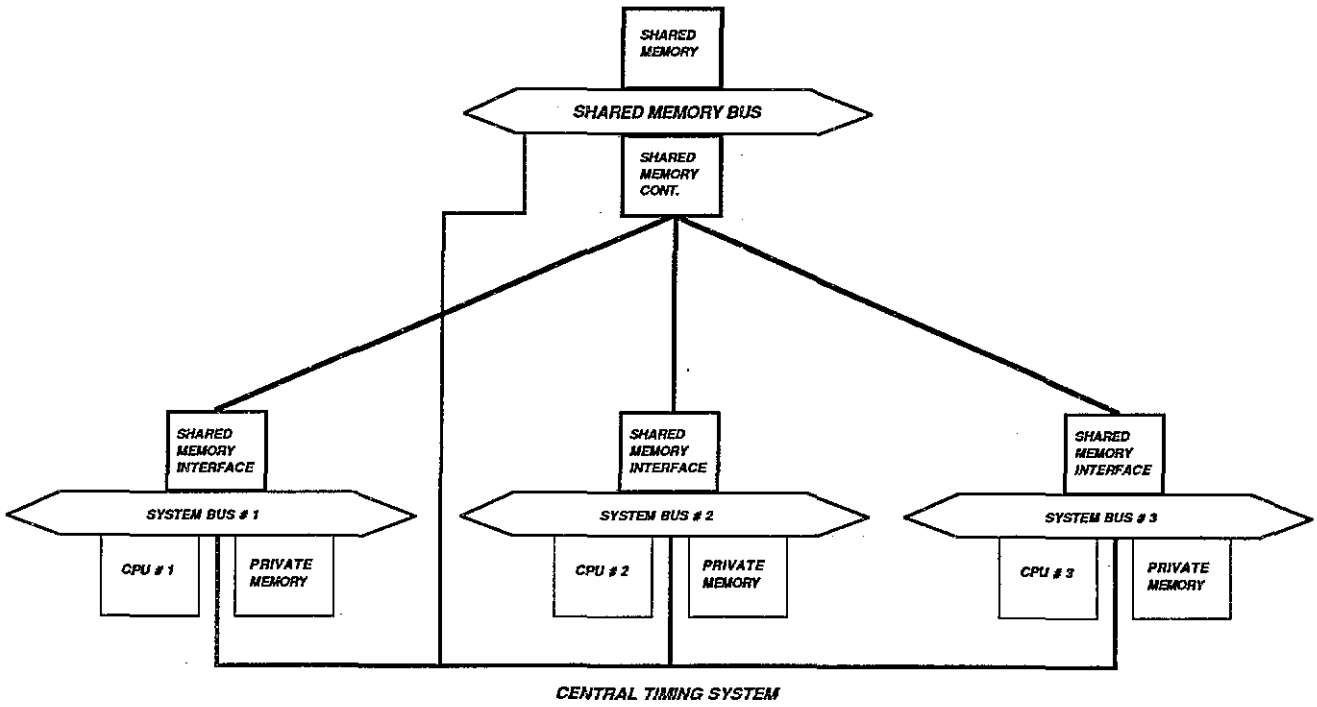


FIGURE 1

This clocking system is necessary because each node in the system sees and treats the shared memory module as an extension of its own main memory with full access privileges.

The conventional shared memory approach lends itself to many single points of failure. If a problem occurs with the shared memory power supplies, shared memory chassis, shared memory module, or the shared memory controller then, in effect all shared memory operations with all connected nodes are lost. New implementations of intercomputer shared memory systems must solve such problems.

B. Dual Ported Shared Memory Approach
(see figure 2)

All nodes are connected by a high speed 26 megabyte (MB)/sec intercomputer datalink. The intercomputer datalink consists of 32 bits of data, 24 bits of address and control lines. Only write data within specified address regions is transmitted out onto the intercomputer datalink. Using 40 foot cables, write transactions can occur on the intercomputer datalink every 150 nanoseconds (ns), a rate which decreases to 300ns using 80 foot cables. The intercomputer datalink is not accessed by read instructions. Each node in the system maintains its own version of the common database; therefore, all data reads are local to each node's main memory. The key ingredient to the success of the new shared memory approach is the fact that all read traffic has been removed from the shared memory system. Typical simulation code accesses the variable data set located in shared memory at a high rate per frame. There are very few writes into the variable data set but, many reads of the common data throughout each frame. The fact that the read traffic has been removed from the shared memory system has increased overall effective bandwidth of the computer system.

High speed dual ported integrated memory modules (DPIMM) are the key to the tightly coupled shared memory approach. DPIMM's are available in 2MB and 8MB sizes. The memory modules are internally two-way interleaved and can be externally interleaved for effective 4-way interleaving. Interleaving allows multiple memory accesses to occur before a memory busy signal is raised on the node. Port 0 (node main bus interface) operates a 26 MB/sec and port 1 (intercomputer datalink interface) operates at 26 MB/sec using 40 foot cables and

DUAL PORTED MEMORY APPROACH

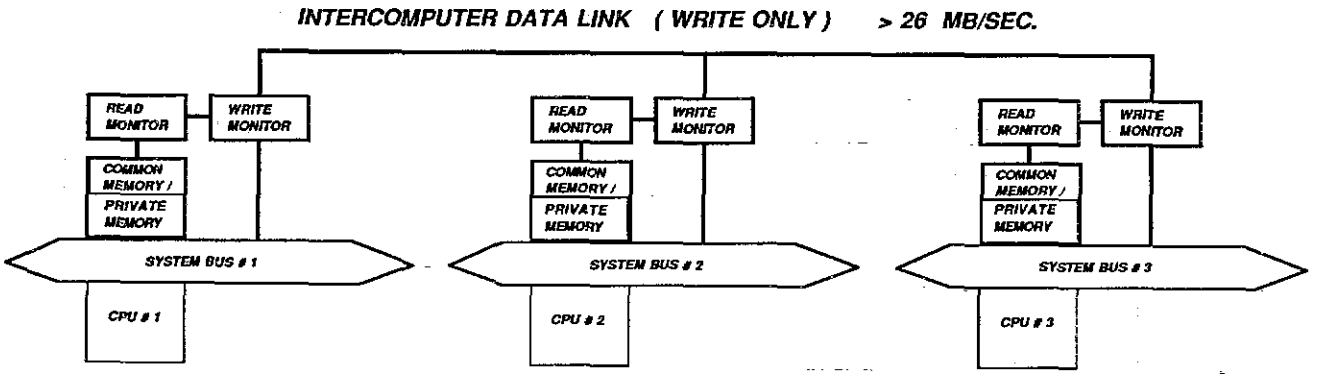


FIGURE 2

13MB/sec using 80 foot cables. Each dual ported memory module has a 4 deep request latch on port 0 which allows four memory requests to occur before the memory busy lines are raised. Port 1 has a 2 deep request latch.

Simultaneous accesses to the intercomputer datalink are arbitrated on a cyclic rotating priority basis. All requesting nodes are assured of getting access to the intercomputer datalink with a worst case wait of (8) clock cycles. This case would only occur if all nine nodes request the intercomputer datalink simultaneously. There are no cycles skipped if the requesting nodes are out of priority order (node 0 - 8). In the case of simultaneous requests, then node 0 is the highest priority and node 8 is the lowest priority during that grant cycle. A node is granted access to the intercomputer datalink every clock tic during the grant cycle by order of priority.

Write Monitor - The Write Monitor senses each node's main bus for address ranges to be transmitted on the intercomputer datalink. The Write Monitor senses writes on the node's main bus within a preselected address range. The address range is dynamically controlled by software and the Write Monitor transmits the data and address on to the intercomputer datalink only if a write occurs within the preselected address range. The Write Monitor contains a first-in first-out (FIFO) buffer that can queue up to 64 intercomputer datalink write requests. This allows the local node to continue processing and not wait for the grant to the intercomputer datalink. The Write Monitor does not occupy space in the node's main card cage. It connects to the system on the back side of the main bus. Each node in this dual ported shared memory system can be set to monitor a different set of shared address ranges by initializing each node's write control register to a different value.

Read Monitor - The Read Monitor senses the intercomputer datalink for shared data that falls within a preselected address range. The address range is dynamically controlled by software and the Read Monitor writes the data into the second port of the Dual Ported memory module only if the address is within the preselected address range. Each node in the system has its own copy or a subset of the shared data based on the Read Monitor address range. The amount of the shared data that is common with other nodes is software configurable and can contain overlap portions with other nodes in the system. The Read Monitor contains a first-in first-out (FIFO) buffer that can queue up 64 memory write requests. This allows the intercomputer datalink to continue processing other nodes requests. The Read Monitor does not occupy space in the node's main card cage. It connects to the node on the back side of the bus directly behind the DPIMM. This board actually interfaces directly into port 1 of the Dual ported Memory module.

Intercomputer Datalink - Up to nine nodes can be connected on one intercomputer datalink, each daisy chained by twisted pair, differential ended cables. Minimum node intercomputer port hardware includes one (1) 2MB dual ported memory module, one (1) write monitor, one (1) read monitor, and datalink cables. The intercomputer hardware is capable of generating external interrupts in the event of hardware failures. Parity errors, buffer overflow conditions, non-present memory and error correction code (ecc) errors all generate an external interrupt which can be monitored and serviced by application software. This capability allows the user to determine the

integrity of the intercomputer datalink during realtime execution. If the datalink is failing, then an orderly software shutdown can occur.

A high reliability option minimizes single point of failure in a tightly coupled dual ported shared memory system. All datalink termination circuitry as well as the link arbitration circuitry is housed in a separate unit. This allows power off maintenance for any node in the system without disrupting or causing data loss on the intercomputer datalink. The datalink is a passive link so that off-line repair of the intercomputer port hardware can be accomplished without affecting the other nodes in the system.

COMPUTER SYSTEM HARDWARE

With the division of the simulation software into multiple tasks and the ability to execute them in a distributed environment it is necessary to change the hardware from the monolithic processor approach to a multiple processor approach and still consider the standard requirements of simulation, such as smaller footprint, increased performance, and lower cost.

To get from here to there you must first identify the areas that don't fit the new approach. Starting with a basic system, as shown in figure 3, we see that it takes at least ten printed circuit boards plus memory and input/output (I/O) devices to run the simplest of jobs. The traditional architecture of the Central Processing Unit (CPU) and the Floating Point Accelerator (FPA), though solid and well established, required too much chassis real estate. New requirements such as increased reliability, reduced cost, and more performance in less space have led to a new design that requires only two chassis slots, one for the CPU and one for the FPA.

The remaining boards, all I/O processors, also needed to be incorporated into a single PC board. This goal was less conceivable, but nevertheless obtainable if the application was Distributed Processing using discless nodes or if I/O performance requirements were not a major factor. Because Distributed Processing was the target application, redesign was a reasonable goal. Reducing a ten board basic system to a three board basic system allowed for redesigning and repackaging the system cabinet. Memory was also addressed by designing some of the features of two standard memory boards into a single board memory and by also adding a second port that could be used by the new shared memory system.

Now comes the question: How do you squeeze a three board set, a two board set, and five I/O boards into three single boards? Following is a brief overview of the philosophy and technique used to develop system hardware to build a tightly coupled distributed processing system.

The CPU board set answer seemed to lie in Custom CMOS Gate Array Technology. With the reduction ratio of 3:1, development decided to start with the 2 micron size arrays and attempted to squeeze the new design onto a ten layer PC board, but unknowns such as timing problems that required a timing circuit external to the arrays ate up large pieces of board real estate. Thus the newer 1.2 micron technology was used in some areas to meet the physical limitations of the ten layer PC board. In the pre-development stages, additional functionality was added to the project. The new CPU would now contain the capability to run virtual memory operating systems as well as real-time memory operating systems. The result of the new design was a single ten-layer PC board with a mixture of 2 micron and 1.2 micron Custom CMOS Gate Arrays that runs both real-time and virtual memory operating systems, and uses about one-third the power of its predecessor.

In parallel with the CPU design the task of combining five I/O processors into a single PC board was undertaken. Rather than risk a whole new system on a single new form of technology, it was decided that the Multi-Function Processor (MFP) should be designed from existing, time-tested, standard technology that engineering was more familiar with than the Custom Gate Array technology, which was used on the CPU. Preliminary layout indicated physical limitations would require the design to reside on one and one-half PC boards. However with the availability of connector pins on the backplane that was being used, the one-half size card could be a Device Interface (DI) card which would plug onto the rear of the backplane in the same slot location as the full size card: thus, the two cards would only use one backplane slot. Utilizing AMD29116 technology for the System Bus Interface, Z80 and SCN2681 DUART technology for the asynchronous section, and Z80, NCR5386S and 8310 technology for the small computer system interface (S.C.S.I.) section resulted in a board set that is compatible with the System Bus, has two S.C.S.I. ports for disc and tape, seven asynchronous ports, one parallel printer port (Centronix or Data Products configurable), one console port, a real-time clock, an interval timer, and twelve external interrupts. All of this I/O was routed on ten layer P.C. boards that use only one backplane slot and consume only about half the power of the five processors that they replaced. The CMOS CPU and the Multi-Function Processor were designed in parallel and were started first because of their high degree of difficulty. After these projects were well under way, the redesign of the Floating Point Accelerator was started so

that its completion would coincide with the other projects.

The redesign of a two board Floating Point Accelerator to a single board Floating Point Accelerator utilized Surface Mount Technology (SMT) to simply reduce the size of the integrated circuits that were currently being used on the two board set. Use of SMT carried with it the prerequisite of a new P.C. board. So a new ten layer P.C. board with surface mounted chips was designed and built. The implementation of the SMT board was a one-for-one replacement of the previous floating point board set without attempting performance changes. With a one-for-one, gate-for-gate redesign the results would be more predictable and much simpler, plus the same arithmetic accuracy would be obtainable. The floating point redesign was completed in twelve months, which coincided with the completion of the CMOS CPU and the Multi-Function Processor. Now that the processors were completed it was time to add memory to the package to complete the board set.

To complete the hardware package a Dual Ported Integrated Memory Module (DPIMM) was added to the system. The DPIMM as its name implies, has two access ports: one port interfaces with the system bus and the other port interfaces with the new intercomputer memory system. Design features such as a four deep request latch on port 0 and two way on board interleaving give the Dual Ported Memory performance that is equal to or better than two standard memory modules. This memory board is discussed further in the "SHARED MEMORY HARDWARE" section.

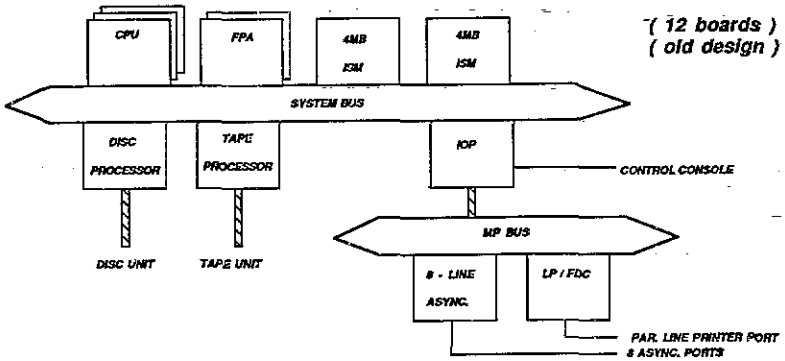
By reducing the system board count from ten to three and one half and including a dual ported memory board we now had the basic complement of boards required for a stand-alone computer system. This opened the door for a redesign of the system chassis size from a fourteen slot version to an eight slot version. The eight slot chassis houses the three and one half boards plus a dual ported memory board and still leaves fifty percent of the chassis for future expansion. Mounting the eight slot chassis vertically in a 22" wide by 55" high cabinet allowed up to four chassis to be mounted in a single cabinet. The power supplies were also mounted on a vertical plate next to the chassis and both the chassis and power supplies were engineered so that they could be easily unbolted and slid out for repair or replacement.

The modularity of the chassis and power supplies made the system easily configurable from one to four chassis in a single cabinet or from one to eight chassis if two cabinets were bolted together. This Electro-Mechanical package lent itself well to the closely coupled distributed processing system that was also currently under development. With the front to rear air flow and four chassis per cabinet a host cabinet and two multi-chassis cabinets could be bolted together to form a nine processor distributed system in a footprint of five feet two inches wide by three feet two inches deep and four feet seven inches high (5'4" X 3'2" X 4'7"). See figure 4. The new hardware packaging shrinks the footprint by 36% Sq. Ft. by reducing the number of cabinets from nine to three. The smaller board complement reduces lifetime sparing requirements --especially when used in a tightly coupled distributed system where all of the basic boards are the same. The lower power requirements of the basic board set will increase the overall reliability of the system by reducing frequency of failures due to excessive heat build-up. The modular approach will serve two needs: 1) the need to easily expand a system, and 2) the need to easily maintain and service the system. All of these improvements will be beneficial in the Simulation Marketplace.

SOFTWARE CONTROL

The connection of multiple computer systems using the Tightly Coupled, Dual Ported Memory approach requires a central point of system control. Figure 5 depicts a Host and two nodes fully configured utilizing the dual ported memory approach. I/O controllers would be added based on simulation requirements. Node 0 (the Host) has control over all other remote nodes in this distributed computer system. Each node in the system is remotely bootstrap-loaded and receives all control commands from Node 0. The operating system executing in each node must be the same revision level as the Host system and retains all the capabilities of the Host system. This requirement is being mandated by simulation Request for Proposals (RFP's). The remote nodes do not require any peripheral equipment to load and execute the operating system, application software, or diagnostics. Peripheral equipment is supported on any node in the system should the specific simulation task require it. The central point of control does away with a requirement for a console crt and a hard disc drive on each node in the system just to support the bootstrap process.

Operating system and simulation code loading takes place via the RS-232 Control Link and the Tightly Coupled Dual Ported Memory Link. All operating system features are available to each task in every node in the system. Each node supports and executes an independent copy of the operating system; therefore, all application code executing in a node is fully supported at the system service level. Full operating system support eliminates the need and overhead of intercomputer operating system message



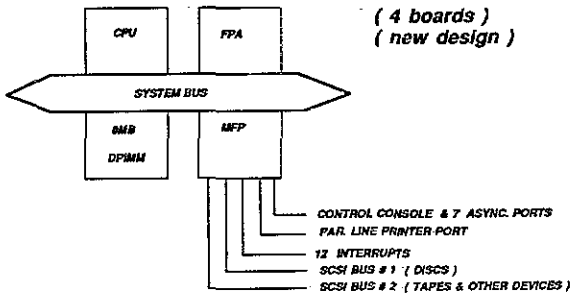


FIGURE 3

HOST & EIGHT NODES

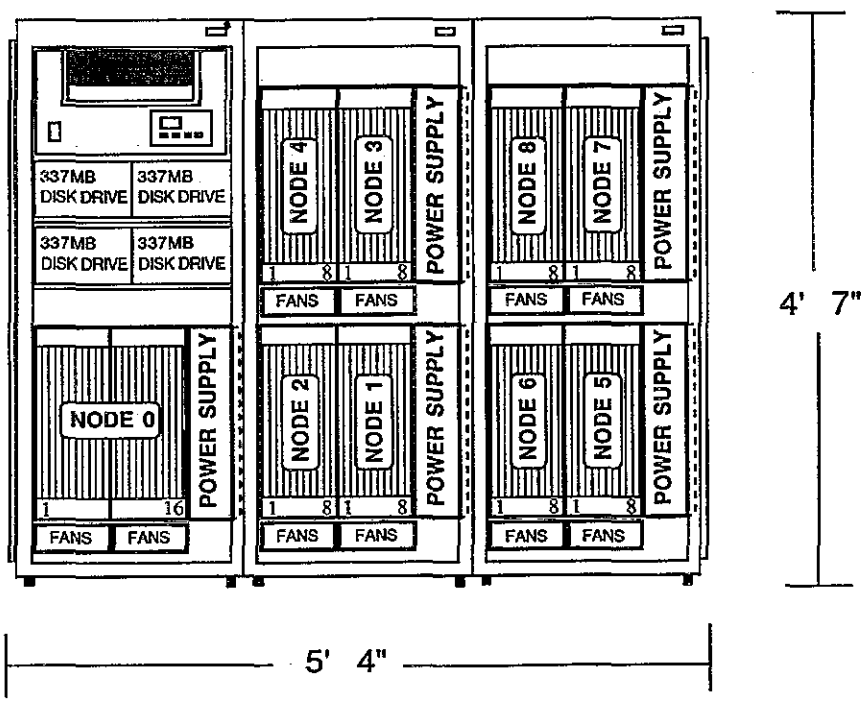


FIGURE 4

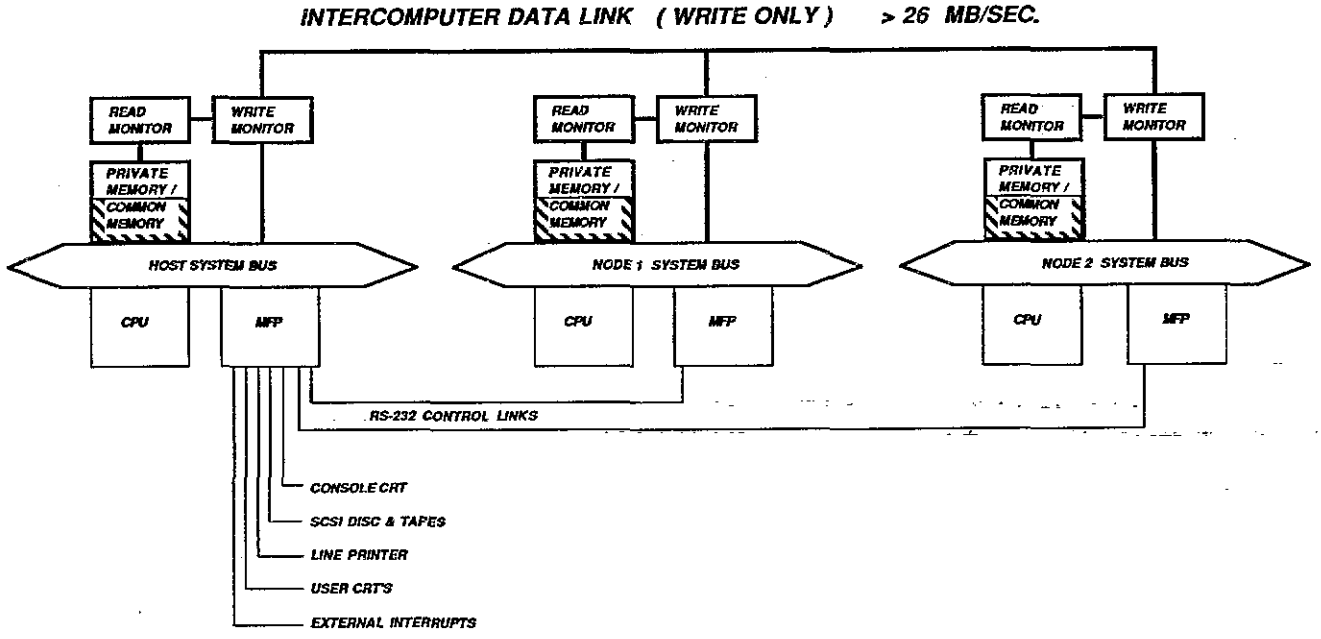


FIGURE 5

services over the Tightly Coupled Dual Ported Memory Link. The operating system that the application code executes under in real time must have the same support as the operating system that the application code was developed and debugged under. A full complement of hardware interfaces is supported by the operating system executing in each node. Each node in the system has the ability to perform all I/O independent of other nodes in the system. Application code has direct access to any configured hardware interface; therefore, the I/O load can be spread among all nodes in the system. Any I/O interface can have direct memory access into the area of memory being shared by all processors in the system thereby, providing all of the simulation tasks with access to the input buffer.

The software system gives the user the ability to preselect and build a remote nodes operating system and task load on the Host system. Once a remote node is loaded, the operating system can activate all simulation tasks at system start up. Memory disc support allows for high speed task activations and high speed disc I/O on a limited basis. A memory disc is defined to an operating system by configuring a portion of main memory to be formatted as a disc device. There are no rotational or head seek latencies associated with a ram memory disc. The memory disc is limited only by the amount of physical memory in a node. All access and use of the memory disc is transparent to the simulation code and the operating system sees and treats it like any other disc defined to the system. If the simulation requires hard discs on one or more remote nodes then the ability to bootstrap the node from the hard disc instead of the in-memory disc is also supported. In this case the memory disc would not be transferred to the node during the bootstrap process. Main memory that would normally be allocated by the memory disc would be available for other uses in that node.

A suite of software utilities is necessary for the central control of a Tightly Coupled Distributed Computer System. All control, downloading and monitoring of the entire system is from a single designated point -- the Host.

REMOTE BOOTSTRAP UTILITY

The Remote Bootstrap Utility is capable of bootstrapping from 1 to 8 nodes from the Host system (node 0). This utility is controlled by a master system definition file. This file defines the node configurations of the entire system. The file contains information defining each node's configuration to include: Node Operating System Image, Node Memory Disc Image, Control Link Address, Memory Disc Device Address, etc. The boot utility executes automatically when the Host system is initialized but can be inhibited from executing by setting a system flag before the Host system is activated. This feature prevents other active nodes in the system from being re-activated when the Host system is restarted. The bootstrap utility also has the ability to bootstrap

all nodes or selective groups of nodes in the system. Selective groups of nodes can be bootstrapped by using an alternate control file with the remote bootstrap utility. Selectively bootstrapping individual nodes does not effect the operation of other nodes in the system. The minimum hardware required to remotely bootstrap a node is a CPU, MFP (Multi Function Processor), a dual ported memory module and intercomputer datalink interface boards. Each remote node's operating system image is built, tested and saved on the Host system, using standard operating system utilities. All operating system initialization tasks as well as user application tasks are built into memory disc images for each target node in the system. A standard set of utilities is used to build and save the memory disc images into standard disc files on the Host system.

ERROR/MESSAGE MONITORING UTILITY

The Error/Message Monitoring Utility is automatically activated on the Host system by the Remote Bootstrap Utility after the first node in the system is initialized. The monitoring utility executes only on the Host system and continually monitors the console port of each active node for operating system and user generated console messages. Any system error or user generated message sent to a node's console port will be captured by the monitor and is retyped to the Host console crt along with the node I.D. of the message originator. In the event of Host system failure and restart the utility can be re-activated manually to continue monitoring all other active nodes on the system. The utility can detect Control Link I/O errors. If an RS-232 Control Link error occurs, then the node is marked offline to all utilities executing on the Host system. If a remote node halt condition is detected, then the monitor prints the appropriate halt message on the Host console crt and marks that node offline. To reestablish communications over the Control Link or to reactivate a node the Remote Bootstrap Utility must be executed.

FILE COPY UTILITY

The file copy utility executes on the Host system and allows disc file transfers from the Host to any configured and active node or from any node back to the Host file system. Disc files being transferred can reside in any volume and directory on the Host or nodes file system. The copy utility signals the target node via the RS-232 Control Link and transfers the disc files over the intercomputer datalink thru a static communications partition. The utility executes interactively or in command line mode. Interactively it is menu driven and prompts the user for all necessary input to complete the file transfer. In command file mode all necessary input is passed to the utility as parameters on the command line. If a parameter is incorrect, the utility will switch into interactive mode and begin prompting the user for the necessary input to complete the transfer. Simulation tasks or data files can be transferred or retrieved from the nodes by using this file copy utility.

REMOTE LOGIN UTILITY

The Remote Login Utility executes on the Host system and prompts the user initially for target node I.D.. It then establishes communications with the target node via the RS-232 Control Link. From any configured terminal on the Host system the user can remotely login into any active remote node in the system. The user can execute commands to determine the status of the node, directly execute file system commands as well as other system level commands.

Using the remote login utility there are two methods of debugging user written handlers that reside within the operating system. With the operating system debugger the user can set break points and stop execution of the operating system at any point. When the operating system halts at the set break point the user can begin displaying registers, changing memory, displaying queues to determine if his code is operating correctly. Additionally, the user can directly enter into the panel mode on the remote node to Halt the node and set instruction stops, write stops or read stops and then release the system to continue to execute normally. When the node encounters one of the set stop points, it halts and allows the user to single step from that point.

The Remote Login Utility has the ability to operate in batch mode so that a predetermined set of commands can be built and executed from a single command file on the Host system. With one command sequence the user can transfer an executable task to a node and get it activated by using the Remote Copy Utility and the Remote Login Utility.

REMOTE STATUS UTILITY

The Remote Status Utility is initially menu driven and executes on the Host system. It uses only the RS-232 Control Link for communications with the target node. It provides target node operating system status and task execution status at a preselected snapshot update rate. All status is displayed at the user terminal on the Host system. Any target node errors or user generated console messages that occur during the status update are captured and immediately redisplayed on the Host console crt. The remote status utility provides operating system status to include: % CPU availability, % IPU availability, % memory availability, O.S. image name, Time of day, etc. The task execution status returned includes: Taskname, Size, CPU & IPU accumulated time, state, etc. The status information displayed by the utility gives the user

detailed information about the condition and current loading of the system during realtime execution. It can also be useful in determining if a particular task is behaving correctly in the system or if the task is in an abnormal queue or if it's using an excessive amount of system resources.

REMOTE COMMUNICATIONS UTILITY

The Remote Communications Utility resides in memory of each active node in the system. It is automatically activated by the remote node's operating system at bootstrap time but, it can be optionally disabled. If disabled, the remote node status and remote file copy features are not available. It communicates with Host system utility programs using the RS-232 Control Link as well as the Tightly Coupled, Dual Ported Memory Link. It interprets opcodes it receives from the remote status and the remote copy utilities and interfaces with the remote node's operating system and file system.

DIAGNOSTIC SOFTWARE

All diagnostics including a diagnostic executive supports a multi-CPU tightly coupled environment. The diagnostics package down loads its executive as well as specific diagnostic programs into a target node. All input from the Host system is directed to the selected target node and all diagnostic output is redirected back to the Host system. The only peripheral devices required on a node are those required by the simulation tasks. No additional peripheral equipment is required to execute the diagnostics on any remote node in the system. All communications to and from the nodes is thru the RS-232 Control Link and the Tightly Coupled, Dual Ported Memory Link. Diagnostics can be executed on any standard supported hardware product that is properly configured on a node. Once a failure is detected, and the repair has taken place, then all that remains is to reinitialize the affected node using the remote bootstrap utility. At this point application software can be restarted and it can reestablish communications with other nodes in the system over the Tightly Coupled, Dual Ported Memory Link.

CONCLUSION

In the previous sections, the authors have shown the use of new methods and technologies for tightly coupling multiple computer systems for flight simulators. By using gate array and surface mount technologies the computer hardware system has been reduced to just a few boards. This has allowed smaller system packaging and will drastically reduce the life cycle costs of the computer system. The conventional shared memory hardware has been replaced with a high-speed dual ported common memory system. Central software control of this tightly coupled multiple computer system gives the simulator manufacturer the flexability required to configure todays simulator systems.

The solutions arrived at have made it possible to attain the high system fidelity required for flight simulation using a tightly coupled, distributed system.

ABOUT THE AUTHORS

Mr. Bocskor is currently product manager for the MPX-32™ real time operating system and the Multiprocessor Reflective Memory Software System, at Gould Inc. He holds a degree in computer science and has 15 years of hardware and software experience related to commercial and military flight simulators. Since joining Gould in 1981, he has served as a software consultant to a major simulator manufacturer and was involved with the design and development of the control software for the Tightly Coupled, Disributed Processing System.

Mr. Cichon is presently the hardware product manager for the CONCEPT/32™ product line, at Gould Inc. He has fifteen years experience in analog electronics and ten years experience in digital electronics. Twenty three years were spent in various technical and managerial functions in a customer services environment, and the last two years were spent as a hardware product manager, with the current responsibility of developing the system packaging and configurations for the Distributed Processing product.

CONCEPT/32™ is a trademark of Gould Inc.
MPX-32™ is a trademark of Gould Inc.