A Parallel Computer Approach for Processing Space Station Telemetry Packets

John T. Polson
New Mexico State University

ABSTRACT

In the Space Station Era, the amount of required telemetry data will be enormous. NASA has proposed a space based network that may ultimately have peak data rates up to 1.2 billion bits per second. There are several levels of processing for the data once it is on the ground. The level zero processing involves reordering of packets, error correction, on line storage, and simple conversion to engineering units. Once the level zero processing is complete the data will be routed over conventional networks to the end users for further processing. The level zero processing will be done by the Data Handling Service, DHS, in real time.

This paper discusses a research effort at New Mexico State University to design and simulate the DHS function using a Global Memory Message Passing, GMMP, parallel computer architecture under development in the Electrical and Computer Engineering Department. This GMMP computer is capable of moving data into and out of main memory at the peak rate. The processing is partitioned by virtual channel number. This proposed implementation does not add much latency to the network. It appears that the entire GMMP computer can be built by cleverly using existing technology.

INTRODUCTION

The Electrical and Computer Engineering Department at New Mexico State University in Las Cruces, New Mexico has been studying advanced space data systems for NASA under the Advanced Telemetry Processing Pilot Program since October 1988. The ATP^3 research project is examining many different aspects of NASA's space based network. Commercially available simulation packages are being used to model telemetry processing, to model network protocols, and to verify some previous contract studies done for NASA by other

organizations. In 1989, Computer Sciences Corporation (CSC) suggested a set of functional requirements for the Data Handling Service (2,3). This paper implements CSC's requirements using a Virtual Port Memory machine, and thus, shows a design for the Data Handling Service using a parallel computer.

THE DATA HANDLING SERVICE IN NASA S SPACE BASED NETWORK

The Data Handing Service is a small piece of a much larger system. The DHS must be reliable and flexible. The Data Interface Facility produces the data which is passed to the DHS. The DHS processes this data and passes it along to the end users. The Virtual Port Memory machine can satisfy the requirements of the DHS suggested by Computer Sciences Corporation.

NASA s Spaced Based Network Requirements

All space communications are routed through the TDRSS, Tracking Data Relay Satellite System. There will be two TDRS in the east and two in the west. All four of the 300 Mbps channels are directed to NASA's ground terminal at the White Sands Test Facility near Las Cruces, New Mexico. The ground terminal feeds a facility called the Data Interface Facility, DIF. The processing done at the DIF has not been finalized. However, the data rate into and out of the DIF is physically limited 1200 Mbps as 4 independent, serial 300 Mbps channels. That is 3.33 ns between bits on each of the four links.

The data produced by the DIF is sent to the Data Handling Service, DHS. Level zero processing is done at the DHS before the data is finally relayed over land based networks or domestic satellite networks to the owners of the data. Level zero processing involves reordering of packets, conversion of some data from encoded form to engineering units, reversing the order of priority playback data, and archiving data for up to two years(1). Figure 1 shows how the space based network components are connected.

The three major data components are video, audio, and experimental data. The video data is high priority and high rate. Audio data is low rate and high priority. These two data types can tolerate some errors. The final data type is experiment data. This data can tolerate very few errors but does not have to get to the user immediately only correctly. This errorless requirement may mean longer download times. However, there will probably be great pains taken to ensure the data's integrity

Virtual Channel Access Protocol Data Unit

The packets sent to the DHS are Virtual Channel Access Protocol Data Units, VCA-PDU, that are encased in Channel Access Data Units, CADIJ, or Coded Channel Access Data Units, CCADU. A CADU is a VCA-PDU with a 32 bit synchronization header. The VCA-PDUs are the data exchanged between the Data Link Layer of the ISO OSI mode, and there is a header and trailer associated with each packet. One of the physical layers in the network is a RF link between a Tracking Data Relay Satellite and the ground terminal. For ease of synchronization, the length of the CADU and CCADU packets are fixed for a given physical channel(1). Considering the 32 bit synchronization header, this allows the length of a VCA-PDU to vary between 992 bits and 10200 bits for different physical channels(1).

The header of the VCA-PDU is made up of 64 bits in 6 fields. In order, the fields are version number 2 bits, spacecraft identification number 8 bits virtual channel number 6 bits, virtual channel data unit counter 24 bits, reserved spares 8 bits, and VCDU header error control 16 bits. The VCDU header error control field is optional. The version number indicates which kind of packet has been encased in the CADU. Currently there are telemetry transfer frames and CADUs. The spacecraft identification number is assigned by the Secretariat of the CCSDS and is fixed for each spacecraft. There can only be 64 virtual channels active at one time for a given spacecraft. The spacecraft identification number and the virtual channel number make up the VCDU identifier. The counter is simply the number of packets sent or the virtual channel modulo 16,277,216. The reserved spare field is set to all zeros until their function is defined. The VCDU header error control field is used as a shortened Reed-Solomon error correcting scheme for the header only. This will help ensure the validity of the header(1).

The trailer in the CADU consists of three optional fields. They are the operational control field 32 bits, the VCDU error control field 16 bits, and the Reed-Solomon check symbol field variable size. The operational control field will allow a mixture of CCSDS system packets along witth advanced orbiting system packets by defining a command link control word. The VCDU error control field is a 16 bit cyclic redundancy code used for error detection. The Reed-Solomon check symbol is used in conjunction with the CCADU (1). The VCA-PDU header and trailer are shown in Figure 2.

The last part of the VCA-PDU is the VCDU or data unit zone. This lies between the header and trailer and is the place for the higher network layers to place their data. There are many protocols that may be embedded in this field. However, the format is set by the VCDU identifier, the spacecraft identification number and the virtual channel number, when the virtual channel is established. The data zone is often filled with CCSDS packets which are small, variable length pieces of data created at the nodes of the space network by certain applications(1).

Data Handling Service Functional Requirements

The DHS performs two major procedures on a VCA-PDU. Packet processing is done by levels 3-7 n the ISO OSI network model and the information is finally passed on to the end users over land based or domestic satellite networks(2). Computer Science Corporation specifies the transformations to be performed by the DHS in a two part report. The titles of the reports are: *Space Station Era Telemetry Processing Identification Report and Space Station Era Processing Function Architecture Matching Report*(3,2).

The packet processing is broken up into 5 major steps. Sequentially, the steps are accept input data, demultiplex by virtue channel number, demultiplex by spacecraft identification number and application identification number, reassemble packets, and select storage method. The packet reassembly is monitored and statistics are kept for the DHS operators(2). The accept input data step is simply taking the data from the transmission media or physical layer and buffer it in memory for processing. The rate at which CADUs arrive is enormous. A single processor can not handle all of the data. Thus, the synchronization markers are removed and the VCA-PDUs are divided among many processors by viritual channel number and spacecraft id number. This reduces the rate at which VCA-PDUs must be processed by a single processor. The VCA-PDUs are the product of the Data Link Layer encapsulation service, They are smaller pieces of large blocks of information, or they are small blocks of information with some filler. The pieces of large blocks must be ordered and assembled. This is done by the reassemble packet function. Finally, some of the data must be archived before being passed onto the end users. This is done in the select storage phase.

There are two steps for distributing data. They are retrieve data from the database and distribute data over the networks. This procedure is also monitored and statistics kept for the operators(2). This seems very simple.

However, there will probably be a conversion between network protocols at this step when the information is taken from the space based network and put on a domestic network.

There are several functions at the DHS that are not in the data path. System information is maintained for accounting purposes and user purposes. Also, the throughput of the overall system is monitored. There are also commands which the system operators can use to monitor the operations. These operator commands have 3 steps: a validation transform, a request for services and an accept response transform. These are not in the data path and can be performed as low priority tasks(2).

In the architecture matching report, Computer Sciences Corporation suggests five different models that satisfy the above requirements. Architecture 1 is shown in Figure 3.

Virtual Port Memory Multiprocessor Architecture

A Virtual Port Memory Multiprocessor Architecture is a Global Memory Message Passing, GMMP, machine. VPM machines are described by Johnson (4) and are general purpose computing machines. The VPM machine is defined as follows:

> A virtual port memory multiprocessor architecture provides each process of a computation or concurrent system with a private virtual address space and pass by value message passing primitives, based upon an underlying hardware structure consisting of a shared memory, equally accessible to all processors, and a pass by reference message network.

The example machine which Johnson describes may have a total of 256 processing elements, I/O controllers, and user interface processors. A four processor prototype of this machine is under development in the Electrical and Computer Engineering Department, New Mexico State University in Las Cruces, New Mexico.

The prototype of the VPM architecture at NMSU is shown in Figure 4. The Interprocessor Message Bus, IMB, is a relatively low bandwidth bus that messages are passed over. Messages are passed between processes. The message unit at the Processing Element (PE), Input Output Controller (IOC), or User Interface Processor (UIP) catches all messages intended for a process that is running on the attached PE, IOC, or UIP. The global memory in the VPM Prototype is a paged segmentation scheme, A pointer to a segment of memory

is a common message. The receiving processes' segment table is updated to show a new segment, and the page frames in the global memory are marked as "copy on write which effectively gives both the sending process and the receiving process its own copy of the data. If one of the processes writes to one of the common pages a new copy is created for that process and the two pages are not marked as copy on write. Similarly, multiple processes can see the same physical pages of memory as long as the copy or write status is maintained(4).

Each PE board is made up of a message unit, a processor, and a large cache memory(4). The message unit is responsible for sending and receiving messages over the IMB. The large cache memory is used so that the PE can run as long as possible without using the shared Data Transfer Bus (DTB). The width of the DTB is the same as the length of a line in the cache for ease of implementation. The UIP boards run user's interface programs such as UNIX-like shells. The IOC boards handle all Input and Output devices such as line printers, disk drives, and ethernet connections. There are many types of IOCs that are required.

When a processor requests a memory reference that can not be handled by the local cache, a request is made to the global memory for the appropriate location. First, the Transaction Request Bus, TRB, is acquired for the request and the appropriate address is placed on the bus. This reaches the Address Translation cache, ATran, which performs the virtual to physical address translation. This usually takes a single clock cycle to complete. However, if the ATran cache can not handle the translation, a software exception is made to update the ATran cache appropriately and then the translation is completed. Finally, the memory request is queued up in a FIFO register at a memory bank, and the memory cycle is performed. The data is passed back to the requesting processor over the DTB.

DESIGN OF THE DATA HANDLING SERVICE USING THE VIRTUAL PORT MEMORY ARCHITECTURE

Software and Process Communication

There are three pieces of hardware a packet, VCA-PDU, encounters on its way through the Data Handling Service. In order they are an input IOC, one or more PE, and an output IOC. An input IOC is the piece of hardware that will accept input data, and demultiplex by spacecraft identification number and virtual

channel number. The spacecraft identification number and virtual channel number make up a segmented address space for virtual channels. This segmented address space, VCDU identifier in the VCA-PDU header, is used for the demultiplexing operation. Data frames that consist of fill data are thrown away at the input IOC. Specifically, fill frames are not put into main memory. Once the VCA-PDU is in main memory, a message is sent over the IMB to the process which is responsible for processing that frame. This message contains the segment number where the VCA-PDU is located in memory. Once this message is sent, the IOC releases the segment from its segment table. This effectively gives ownership of the memory that holds the transfer frame to the process that handles the frame. If there is no process responsible for handling the newly arrived frame, a message is sent to an error handling process which checks the header, if possible, and issues appropriate messages to a Data Handling Service error process or to the Data Interface Facility.

The processes that are responsible for handling transfer frames execute on the processing elements in the VPM architecture. There are many identical processors. However, there are not as many processors as there are segmented virtual channels. Thus, many processes may execute on a given processor. Each process handles a specific virtual channel. These processes are different for each virtual channel. However, there are some common functions among processes. Each process will prepare the date for archiving and distribution. However, some data may not be archived, and some data may not be distributed. A few virtual channels will be devoted to priority playback data, and these processes will have to reverse the order of the data bits. The variety of functions performed at this stage is limited only by the number of payloads. When processing is complete, a message is sent to an output IOC. This message, in general, contains a segment that holds a list of segments that have prepared data in them. These segments are given to the IOC as described above.

An output IOC is responsible for archiving and distributing the transformed data. The first segment is read into memory. This segment is examined for instructions as to where to send the data, how to archive the data, and where the data is located in memory. The data may be relayed to multiple users over networks and could be archived in multiple locations. However, there should be only one read of data from the VPM’s main memory. Just like the input IOC, the output IOC does simple operations in the same manner each time.

Global Memory

The global memory in the VPM architecture will be used to hold packets as they are processed, to hold all code, and to hold all operating system data structures. This memory requires a very high bandwidth in and an equally high bandwidth out.

The main memory in the Virtual Port Memory Multiprocessor Architecture is made from VRAM, video random access memory. VRAM is a special type of multi-ported, dynamic memory. The first port on the memory is identical to a standard DRAM array. Each cell contains 4 bits. The other access port(s) are serial in nature and have an enormous band width. There is a static ram shift register that is the same length as a row in the DRAM array. This shift register may be loaded from the DRAM array with an entire row in one memory cycle. The contents may be removed from the shift register through four pins on the chip. The column address is used as an index into the shift register for the starting location. These shift registers can also be filled through the serial port, and when full, an entire row of the DRAM array may be written in a single memory cycle.

The currently available chips are usually arranged as a 512 X 512 X 4 array(5). There is usually a single shift register, 4 bits wide and 512 long, that can be used as an input or output shift register. VRAM with multiple serial ports are produced. However, they are currently not in great demand and the shift registers are shorter than a full row. The technology exists for creating the VRAM with 2 serial ports shown in Figure 5 and a single random access port. In the next few years, these parts will come into production.

Input IOC

Four input IOCs are required to put the transfer frames into the global memory and pass a message to the process that is handling a particular frame's virtual channel number. The transfer frames arrive by a serial link. This link may be an RF link or it may be a fiber optic link, but this data must be converted to a parallel form for storage into memory, Also, a 16 bit CRC code must be generated for each packet and stored after it in memory.

The conversion from serial to parallel form can be done by a pair of high speed shift registers in a ping-pong configuration and a controller. Figure 6 shows these shift registers and the CRC generation data paths. The first register is

filled from the 300 Mbps line until full. The second shift register is filled while the first is emptied, in parallel, to the VRAM shift registers. When the second register is filled the 300 Mbps line is directed at the first register and the second shift register is emptied. A new bit arrives every 3.3 ns. The high speed shift registers are the same length as the memory banks are wide. Specifically, they are 128 bits in length. Thus, it requires 427 ns to fill a shift register. Data rates in excess of 1000 Mbps can be handled by ECL shift registers.

When a packet has arrived the contents of the VRAM shift register must be moved into the DRAM array. This takes one memory cycle or 80 ns. The 427 ns required to fill up a high speed shift register provides plenty of time for this memory cycle.

Output IOC

The Interface to the output IOC is not clearly defined. However, the second serial port on the VRAM will be used to removed the data from main memory when necessary. This data will be buffered locally in RAM by the IOC until two functions are complete. These are the archive and distribute data functions described earlier.

It is possible to build a disk drive that has an input and output bandwidth in excess of 300 Mbps. This will be used to buffer the data because a disk drive's memory is non-volatile. Eventually, the data will be moved from this disk drive to some other disk drive. This is very similar to the memory hierarchy used in modern workstations. While the archiving is being performed, the data will be put on the domestic networks destined for the end user.

Processing Time Models

CACI's NETWORK II.5 simulation package is used to determine the significant parameters for the Virtual Port Memory Multiprocessor Architecture. The width of the DTB, the width of the IMB, the number of PEs, and the average processing time are found. Figure 7 shows the basic block diagram of the simulation. More detailed models are under development for examining the specifics of the large system. An in depth description of the simulation can be found in (6). A discussion of NETWORK II.5 can be found in (7) and (8).

NETWORK II.5 simulates *processing elements*, *transfer devices* and *storage devices*. Modules represent processes executing on the processing elements. Processing elements are defined by their instructions. Transfer devices are described by their physical attributes and protocol. Storage devices are described by their access time, resident files, size limitations, and number of access ports. Modules can be triggered or started by sending a message to a processing element

The simulation model provides the ability to thoroughly analyze the VPM design. There are many design requirements that need to be evaluated. Analysis indicated that the following criteria needed to be evaluated first to determine if the VPM architecture was capable of meeting CSC's functional requirements for the Data Handling Service.

* Width of the Data Transfer Bus
* Width of the Interprocessor Message Bus
* Number of Processing Elements
* Average Processing Time per Packet
* Resident Time in Memory

Additional simulations will be constantly updated to reflect changes in DHS requirements, hardware specifications, and software specifications.

The simulation was used to determine the width of the DTB The width of the DTB was increased until its simulated utilization was less than 50%. Only powers of 2 were used for the number of bits retrieved and the frequency of operation was fixed at 16 MHz. The low frequency is a requirement because the DTB runs across a backplane bus. It is important to note that on average 1.5 bytes of packet data is accessed for each byte of packet data. This simulates the manipulation of pointers to data and writing them back to the global memory. A DTB running at 16 MHz should be 128 bits wide.

Then the processing time simulation was used to determine the width of the IMB. The messages passed over the IMB were simulated at 128 bits because this is the required length of a message that passes a segment number. An IMB running at 16 MHz should be 16 bits wide.

The simulation was used to determine the number of Processing Elements required to perform a minimum average processing time of 300 microseconds,

approximately 1 instruction per byte of data plus message overhead times using a 25 MHz CISC processor. There should be 64 Processing Elements in the DHS.

Finally, the average processing time was increased so that al PEs were near 100% utilized. This average processing time was found to be approximately 425 microseconds. For a 25 MHz CISC processor this is approximately 1625 instructions or 1.6 instructions per byte of data. Considering the quantity of video data and how little processing is done to it, video data is simply passed on to the output IOC by the PE, the average is very small. The experiment data will have plenty of time for level zero processing. Also, when the data rate is not running at its peak rate, each packet can take a little longer.

According to the simulation, the maximum time a single packet spent in memory was 1200 microseconds. Each packet requires a whole page frame of memory which is currently 2 Kbytes, and according to Little's law the number of packets in memory does not exceed 176. Thus, approximately 360 Kbytes of memory is all that is required to buffer packets in memory.

The 128 bit wide DTB implies memory banks that are 128 bits wide. Using the described VRAM chips, each bank has 4 Mbytes of memory. Thus, 8 banks of memory are used to give the machine 32 Mbytes of global memory. This allows each input IOC 2 banks to use for placing packets in memory. The remaining memory is required for storing code and data structures for the VPM machine

CONCLUSIONS AND FUTURE EFFORTS

A VPM machine with 64 processors, a 128 bit DTB, and a 16 bit IMB can implement the functional requirements for the DHS suggested by Computer Sciences Corporation. All the technology required to implement the DHS with a Virtual Port Memory Mutiprocessor machine is available today. This is an efficient implementation. The PE boards are inexpensive to replace and modular. If a new technology for transmitting serial data is chosen, the IOC boards can be replaced with more modern parts. The software is flexible and easily changed. The processing done to a virtual channel's packets can be specified by the experimenter.

There are many aspects of the space based network that have not been finalized. The format for archiving the data passing through the DHS has not been defined in any way. Defining this format will enable the output IOC

structure to be defined. Also, the type of domestic networks the DHS will route data over has not been specified. This will also help define the structure of the output IOCs. It is not clear how many output IOCs will be required or what they will look like.

Failure analysis for the DHS requires further investigation. Possible points of failure include a PE, a memory bank, an input IOC, an output IOC, power source, and software failures. Some of these points are easily addressed. If a PE fails, its processes will have to be re-started on a different PE. If a memory bank fails, it's page frames will be marked invalid until it can be replaced. If an input IOC fails the packets sent to it will never arrive, and eventually a request for retransmission will be made and the packet can arrive via a different link. If an output IOC fails, a different IOC can be used for awhile. If the power fails, it will be important to know how the machine will come back on line. If a software process fails, the destination node will eventually complain about not getting any data and the process can be restarted. It would be advantageous to know the time required for these steps.

ACKNOWLEDGEMENTS

Thanks are extended to Dr. Javin Taylor for reviewing this document and for directing my research, Dr. Eric Johnson for his endless help during my efforts, and Dr. Stephen Horan for his recommendations. The examples they have set for me at New Mexico State University have been inspiring.

REFERENCES

1. Consultative Committee for Space Data Systems, Advanced Orbiting Systems Networks and Data Links: Architectural Specification Volume A (Red Book), CCSDS, January 1989.

2. Computer Sciences Corporation, Space Station Era Processing Function Architecture Matching Report, Computer Sciences Corporation, July 1989.

3. Computer Sciences Corporation, Space Station Era Telemetry Processing Identification Report, Computer Sciences Corporation, March 1989.

4. Johnson, Eric “The Virtual Port Memory Multiprocessor Architecture.” Ph.D. diss , New Mexico State University, 1987,

5. Toshiba, Toshiba MOS Memory Products Data Book, Toshiba, 1989.

6. Poison, John T. “Design of Advanced Space Data Handling Service Using Virtual Port Memory Multiprocessor Architecture” Ms thesis, New Mexico State University, 1990.

7. CACI, NETWORK II.5 User’s Manual, CACI, 1988,

8. Cheung, Shun et al, Introduction to Simulation Using Network II.5, CACI, 1988.

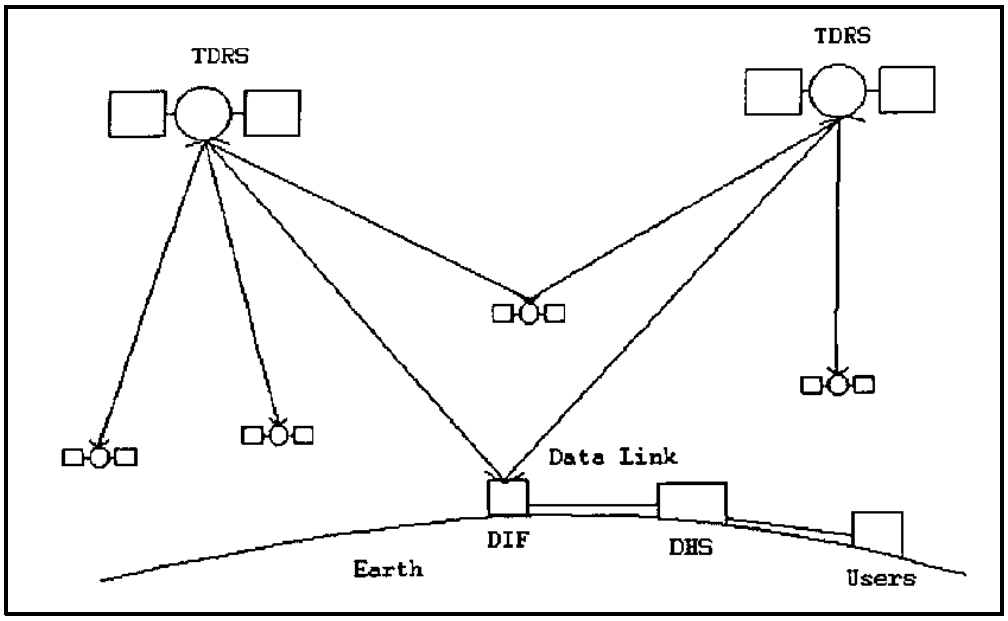


Figure 1 - Space Based Network

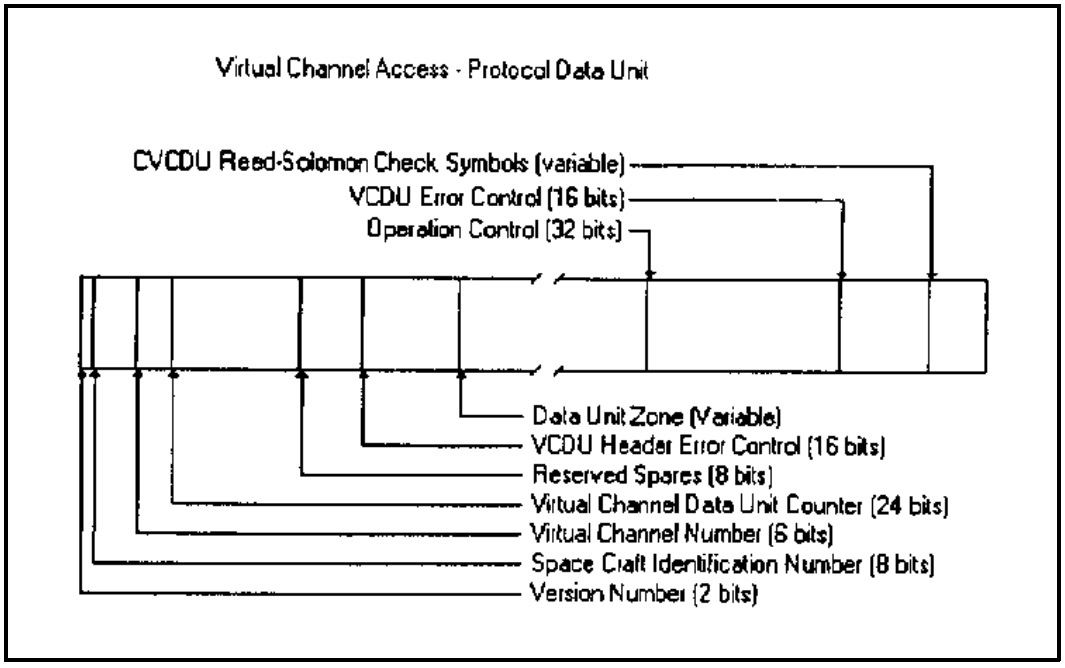


Figure 2 - Virtual Channel Access Protocol Data Unit

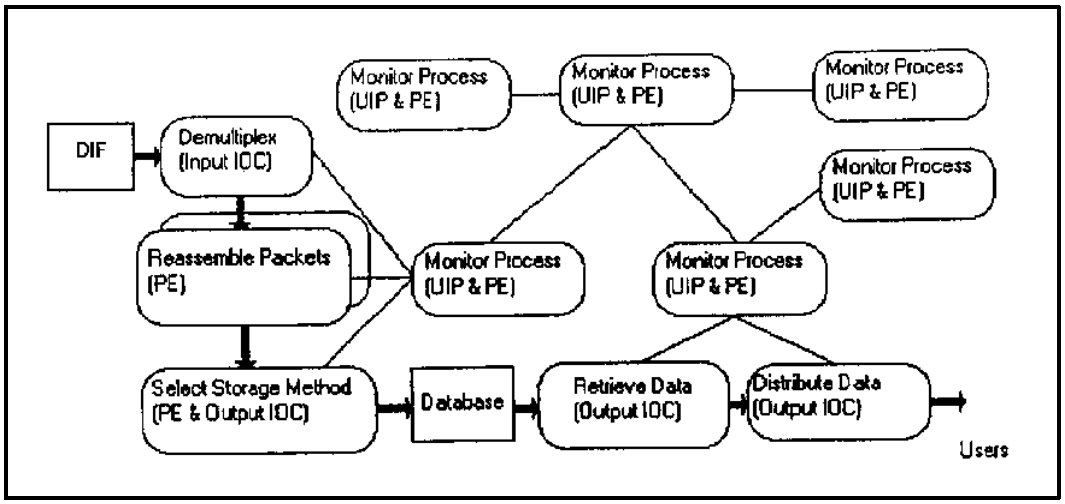
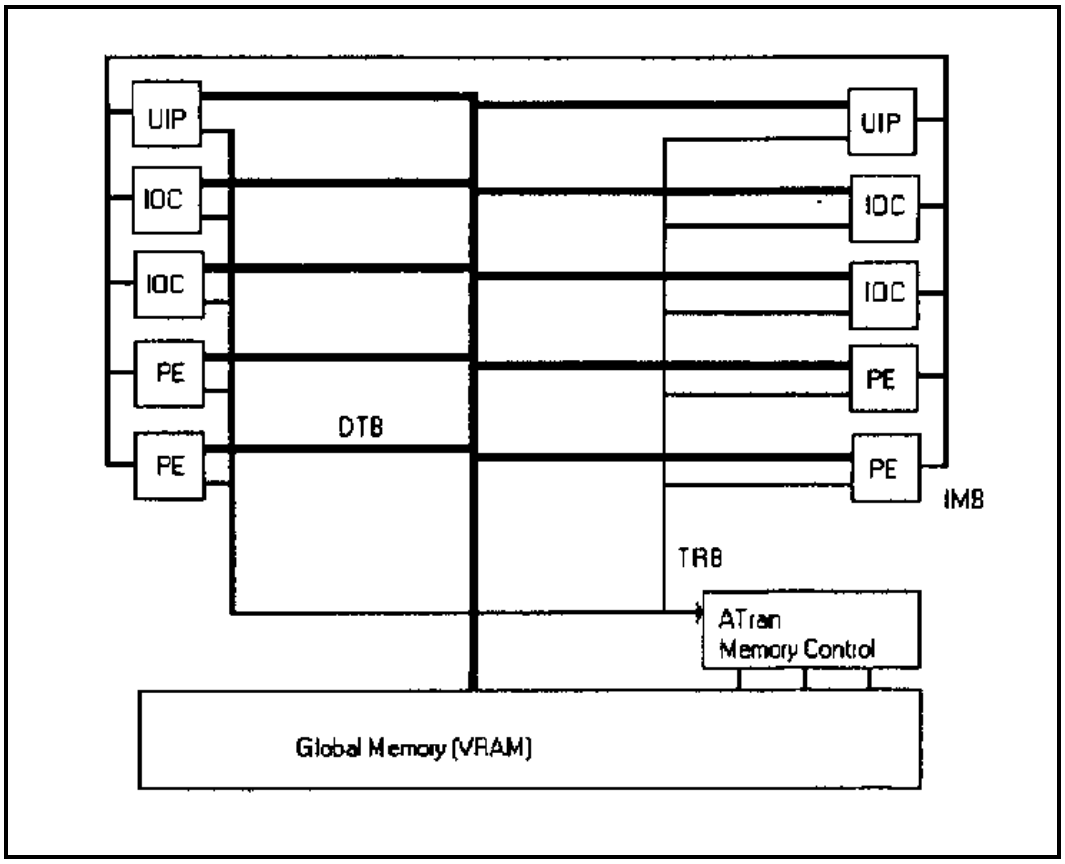


Figure 3 - Computer Sciences Corporation Architecture 1



igure 4 - Virtual Port Memory Architecture Diagram

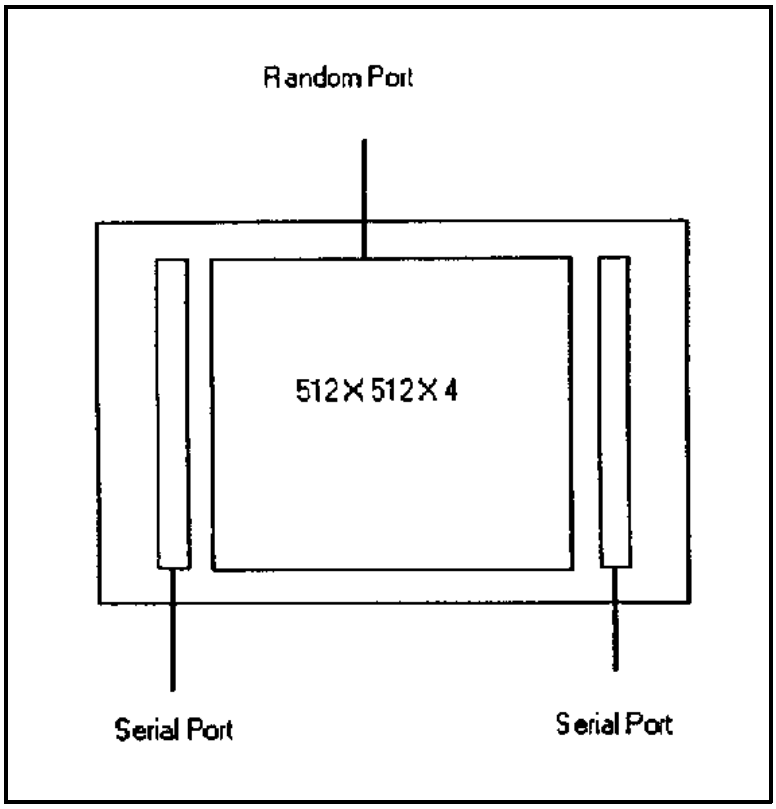


Figure 5 - Video Random Access Memory (VRAM)

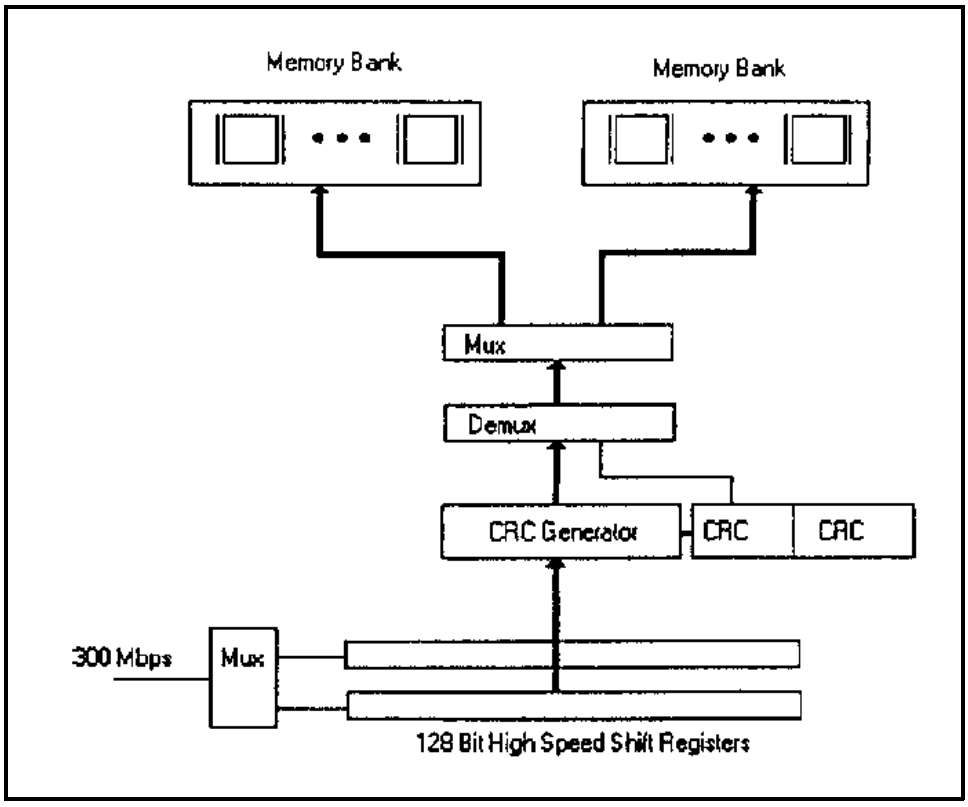


Figure 6 - Input IOC Block Diagram

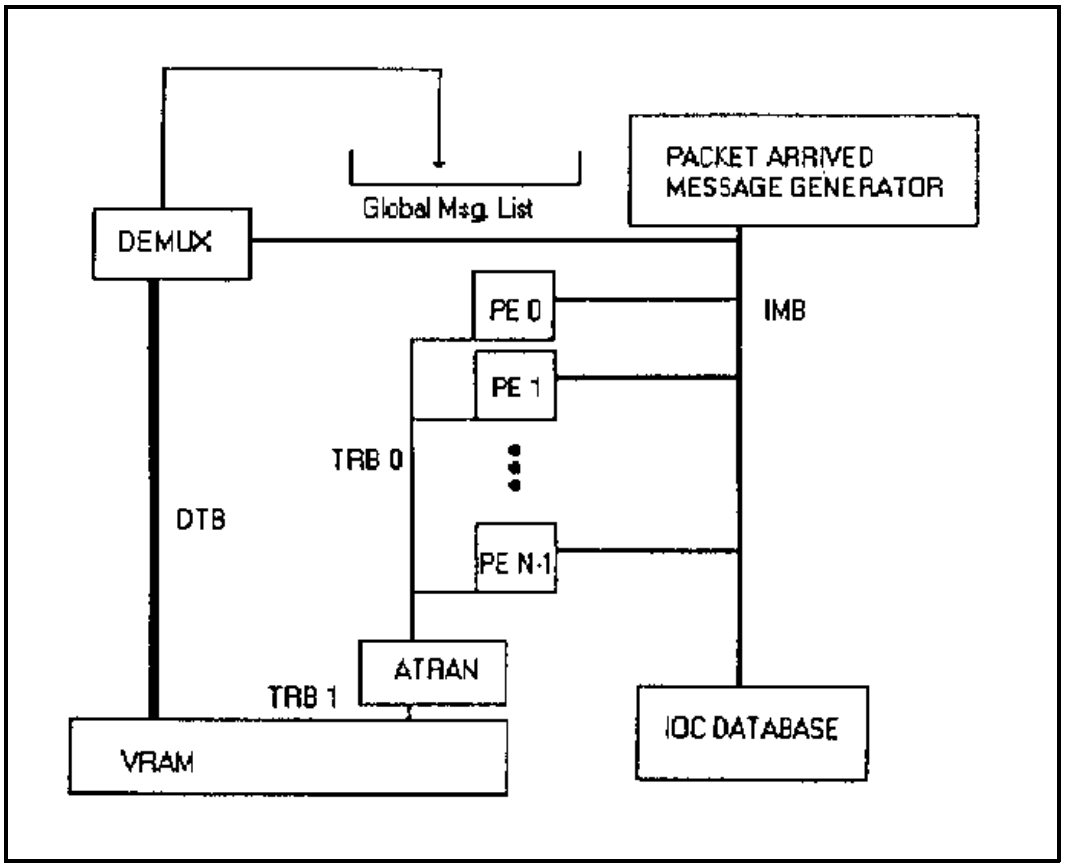


Figure 7 - Processing Time Simulation Block Diagram