NASA Contractor Report 181828

Avionics Architecture Studies for the Entry Research Vehicle

M. J. Dzwonczyk, M. F. McKinney, S. J. Adams, and R. J. Gauthier

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

May 1989

Contract NAS1-18061

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

FOREWORD

This report is the culmination of a year-long investigation of the avionics architecture for an Entry Research Vehicle (ERV) . The work was performed under the direction of NASA's Langley Research Center by the Charles Stark Draper Laboratory, Inc. (CSDL). Messrs. Howard Stone and Charles Meissner were the the NASA contract monitors for the program. Funding was provided under contract NAS1-18061, Task Assignment 7.

The Entry Research Vehicle is conceived to be an unmanned, autonomous spacecraft to be deployed from the Shuttle. It will perform various aerodynamic and propulsive maneuvers in orbit and land at Edwards AFB after a 5 - 10 hour mission. Preliminary study of the vehicle's avionic architecture was performed by CSDL under Task Assignment Number 1 of the same contract in the Spring of 1986. This follow-on study detailed the design and furthered the analysis of the architecture. This report describes in detail the avionics architecture proposed by CSDL and its subsequent reliability analysis.

Much of the work reported here has been informally released as part of the ERV memo series. The presentation of those documents here supercedes all previous releases of them.

ACKNOWLEDGEMENTS

The work documented in these volumes was performed by the Fault-Tolerant Systems Division of CSDL. More specifically, Mark Dzwonczyk served as the program manager and principal investigator; Martin McKinney studied hardware design issues such as component selection and device failure rates; Stuart Adams performed the reliability analyses for the candidate architecture; and Robert Gauthier devised the technology development plan for vehicle deployment. Mention is also due to Dr. Jay Lala, who as the Division Leader, provided valuable insights concerning the design and development of the fault-tolerant computer architecture. The authors would also like to acknowledge the help of George Chen, John Esielionis, Jake Martin, Jean Brady, and Florie Fabiano for their efforts in completing the work.

This report is dedicated to Bernie Kriegsman whose contributions of technical expertise and enthusiastic advice enlightened the authors with encouraging recommendations for solving the difficult problems associated with vehicle guidance and navigation.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION

The Entry Research Vehicle (ERV) concept was designed to be an experimental spacecraft operating in the mid-1990s. It would be a small, unmanned, autonomous vehicle deployed from the Space Shuttle. As such, the ERV would be used to demonstrate the operational capabilities of future NASA and USAF reusable orbital transportation systems (including Shuttle II). To meet these goals, the ERV mission plans include various propulsive and aerodynamic maneuvers during deorbit and reorbit phases.

After deployment from the Shuttle, a typical ERV mission will involve two to three Earth orbits during which the flight experiments will be performed. Mission durations are expected to be on the order of five hours. Missions are terminated with a reentry phase to glide landing at a dry lake bed in Southern California (Edwards AFB). Approximately five to ten missions for the single vehicle are proposed.

To perform its myriad of flight maneuvers, the ERV will require an extremely accurate guidance, navigation, and control (GN&C) system. Because of the hazards involved in the ERV flight (Shuttle proximity, autonomous landing near populated areas) and the high cost of vehicle (upwards of $100 million), a second principal requirement of the GN&C system is that of extremely low probability of failure resulting in vehicle loss. For worst case missions, the required probability of vehicle/mission loss due to avionic failure must be less than 10^{-6}.

In the Spring of 1986, the Charles Stark Draper Laboratory (CSDL) was contracted by NASA's Langley Research Center to perform preliminary avionic design studies for the ERV. That three-month initiative resulted in a strawman architecture for the ERV flight computer and associated inputs and outputs (I/O). The architecture was based upon the Advanced Information Processing System (AIPS) methodology for developing ultra-reliable computing machinery. AIPS, which is also being developed at CSDL and sponsored by NASA, approaches ultra-reliability requirements as a *fault-tolerance* problem, i.e., designing a system which can detect, isolate, and reconfigure itself around failed elements.

The proposed design for the ERV avionics consisted of a centralized AIPS Fault-Tolerant Processor with triply redundant distributed MIL-STD-1553 buses serving as the primary I/O ports to the vehicle subsystems. The initial studies completed that spring included performance and packaging specifications for the ERV avionics; software specifications for the guidance, navigation, and control of the vehicle; as well as guidance and navi-

gation algorithm design and simulation. The final report from that work was published in April, 1986 [Kriegsman et al].

In August of 1986, NASA/Langley awarded CSDL this follow-on contract to refine the ERV avionics suite design. Specifically, two major issues in the ERV architecture have been studied:

1. System design and implementation to module level,
2. Reliability analysis of the resulting architecture.

This report describes in detail the design and analysis of the ERV flight computer. The second section of this volume gives a a brief overview of the ERV avionics design and discusses key aspects of the Advanced Information Processing System (AIPS) architecture, upon which the ERV design was based. Section 3 describes, at a functional level, the buses that link the various parts of the flight computer. It also proposes possible implementations of the buses and their Interface Units (BIUs) at a physical level. In the fourth section, several microprocessors are examined, and one is chosen for the central processing unit in the ERV flight computer. A conceptual packaging design and sizing for the flight computer are included in Section 5, and Section 6 describes the interfaces between the avionics devices external to the flight computer and the Input/Output buses from the flight computer. In Section 7, the method and results of a reliability analysis which was performed on the ERV avionics are described. The study conclusions are presented in Section 8.

2.0 AVIONICS OVERVIEW

The reliability of a system, R(t), is the probability that the system will function properly and continuously over the time period [0, t]. The complement of a system's reliability is the probability of the system loss. If R(t) is the reliability of a vehicle, then the Probability of Vehicle Loss (P_{VL}) is equal to 1-R(t). A P_{VL} of 10^{-6} due to the vehicle's avionics was set as the requirement for the ERV design. Using the AIPS philosophy in the design process, this number was easily attained. The AIPS fault tolerant processor (FTP) architecture allows the system to suffer malicious failures while maintaining its integrity. Redundant hardware masks out faults as they occur in realtime while system software reconfigures the architecture into operational states. Literature on AIPS [Lala 84] and the FTP [Smith 84 and Lala et al] is available.

The ERV flight computer consists of three sets (channels) of identical hardware and is, thus, termed a triplex FTP. All channels run the same processes concurrently, exchanging and comparing data to assure each has a congruent copy. Each channel is divided into two Fault Containment Regions (FCRs): a *processor* and an *interstage*. An FCR is a module from which no faults can propagate into or out of. Each FCR must be electrically and physically isolated from the others and provide its own independent power and clock mechanisms. Although faults will not propagate from one FCR to another, errors may and must be masked out.

The architecture of the FTP is designed to be resilient to *Byzantine* failures. A Byzantine failure is an arbitrarily malicious failure; for instance, one that causes a module to transmit inconsistent data to differing parts of the system. It has been shown that in order for a system to tolerate *f* Byzantine failures, it must contain $3f+1$ FCRs, connect the FCRs by $2f+1$ links, and communicate data $f+1$ rounds through the FCRs [Lamport, Shostak, Marshall], [Dolev]. In addition, the FCRs must be synchronized to within a well-defined skew. A triplex FTP is resilient to all first Byzantine failures and thus provides 100% coverage for all first faults. A triplex FTP also provides 95% coverage for all second faults.

Figure 2-1 shows a block diagram of the triplex FTP and its six FCRs. The processors are drawn again on the right to show the interstage-to-processor connections. The gray lines in the interconnecting network refer to links that are not present for all types of communications (clock and data topologies differ).

The links connecting the processors to each other are data communication paths. These are bidirectional and are used for single source data exchanges. A single source data exchange is required to assure data congruency among all channels when the data source is only connected to a single channel.

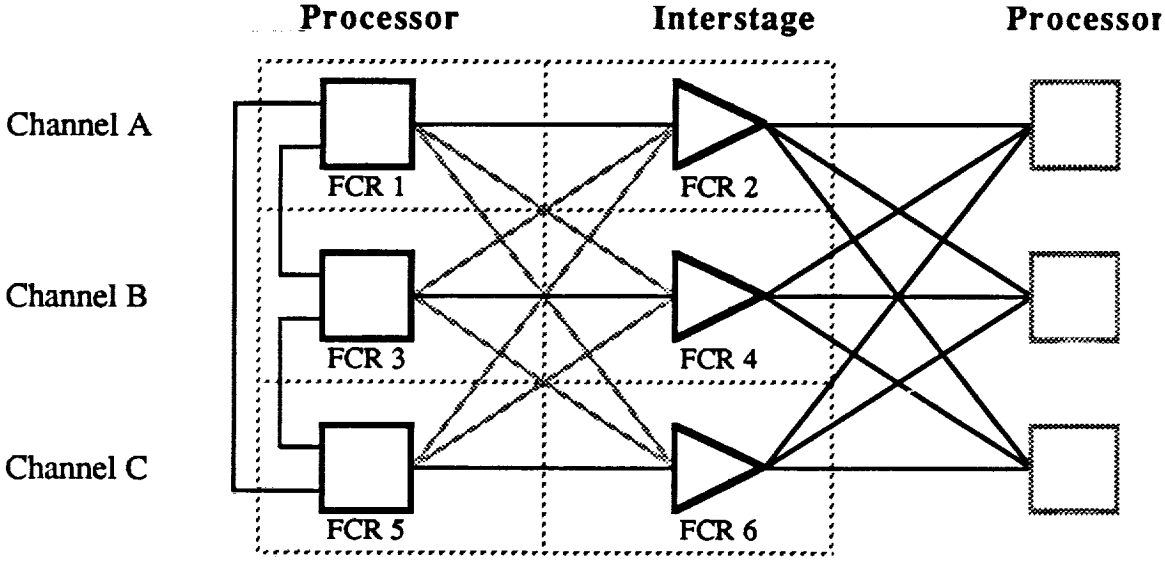

Figure 2-1. *FTP Block Diagram*

Once all processors have received the source data, the network passes the data from each processor to their respective interstage and then back to *all* of the processors. The data is voted in the hardware at the processor after it has gone through the exchange network. Error flagging is done in the hardware, relieving the system software from the task.

In addition to providing data congruency, the interchannel links provide a topology for synchronization via a Fault Tolerant Clock (FTC). The FTC synchronizes the channels to within a well defined bound. Each channel generates its own version of the system FTC and exchanges it with the other channels for comparison. It then adjusts its own clock as a result of the comparison, in effect, implementing a digital phase-lock loop. Since each FCR needs independent clocking, the FTC is voted at both the interstages and the processors. Hence, each processor broadcasts its FTC to all interstages where the signals are voted and then passed back to the processors for a second vote.

Section 3 discusses the interchannel communication in detail. A variety of FTP-related papers have been published which describe the architecture [Lala 86], [Lala et al], [Alger], [Gauthier].

Each "processor" FCR of the FTP will contain two 32-bit microprocessors running at 25 MHz accompanied by a floating-point coprocessor. 64 Kbytes of RAM and 1 Mbyte of program ROM will be available to each microprocessing pair. Although the final implementation has not yet been decided, the likely use of each of the processing pairs is one for computation-intensive operations and the other for transfers from/to I/O devices. In addition, each "processor" will contain some shared hardware, including the I/O bus interfaces and the interchannel communication.

Several microprocessors have been considered for the ERV flight computer including some MIL-STD-1750A[1] architectures and many 32-bit architectures. Microprocessor selection is based on a number of factors. These characteristics include performance, memory and I/O capability, reliability, power dissipation, size, and software support. Comparing these characteristics with the application's requirements determines whether the microprocessor will meet the application's needs. The main characteristics considered in the microprocessor selection for the ERV flight computer were performance, memory capability, and reliability. After careful comparison of benchmark reports, and the other characteristics, the Motorola 68020-68881 tandem was chosen as the microprocessor-coprocessor pair for the ERV flight computer. Although the 68020's performance is matched by a few other microprocessors, its reliability and support are much better. The microprocessor selection is discussed more fully in Section 4.

The FTP I/O network to the rest of the avionics will consist of three MIL-STD-1553[2] buses and three custom I/O buses. The custom I/O buses will be connected (one to each channel of the FTP) and will interface the FTP to the I/O devices that are not 1553 compatible. Each custom I/O bus will be connected to two of the three 1553 buses to provide a redundant cross-strapped network. The dual cross-strapped architecture provides tolerance of avionics failures (external to the FTP) without full triple replication of the hardware.

A block diagram of the FTP and its I/O buses to the rest of the avionics is shown in Figure 2-2.

---

1 The Air Force standard instruction set architecture for microprocessors used in spacecraft flight control.

2 This is the Military Standard Digital Time Division Command/Response Multiplex Data Bus and will conform to the latest MIL-STD-1553 specification. Currently, this is MIL-STD-1553B.

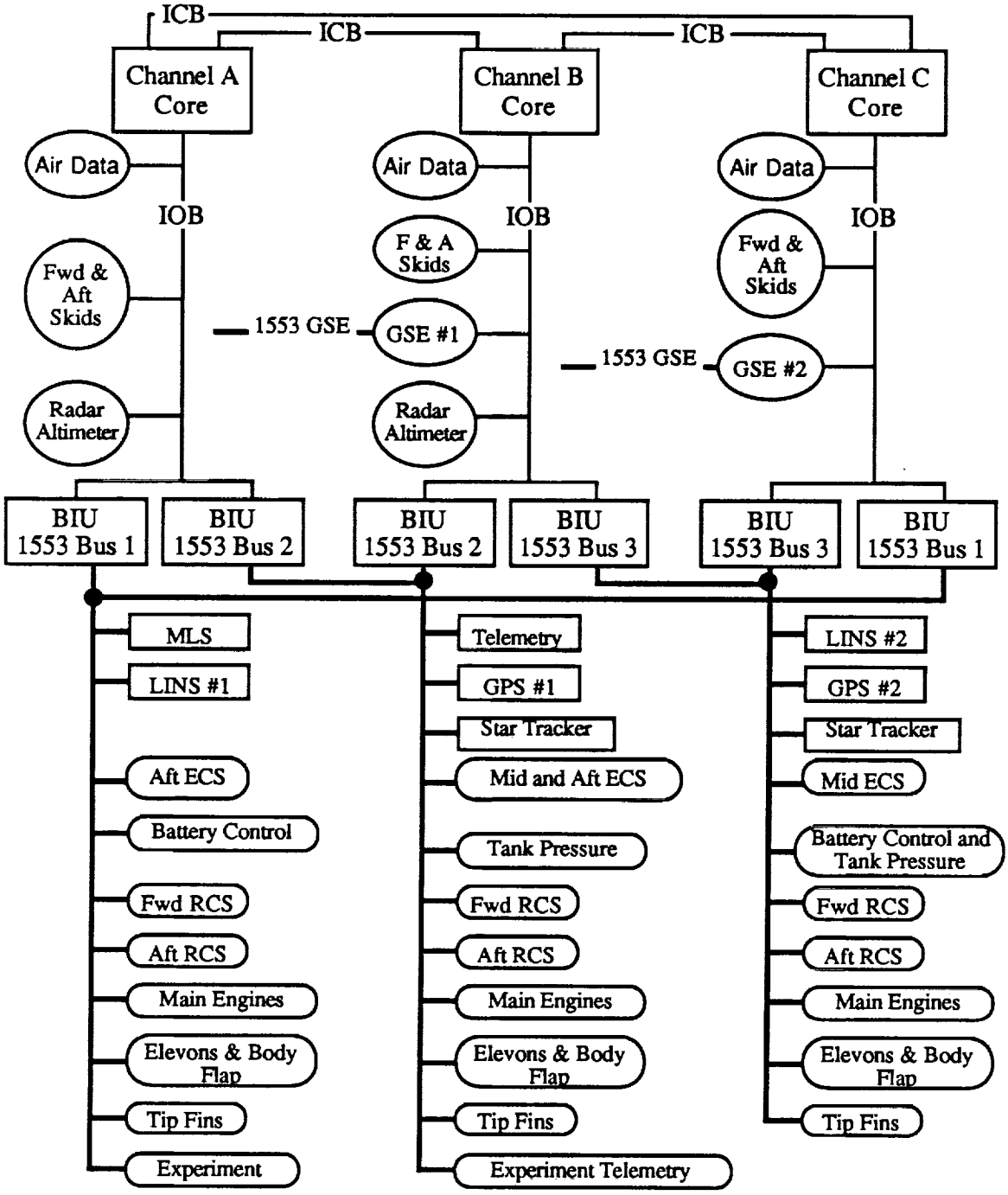

Figure 2-2. ERV Avionics Block Diagram

3.0 BUS STRUCTURES

The implementation of the FTP for ERV requires the definition of several buses which link various portions of the computer. This section describes at a functional level the structures of these buses. It also proposes possible implementations of the buses and their Interface Units (BIUs) at a physical level. Because the ERV avionics is still at an early design stage, however, this section does not attempt to provide detailed hardware characteristics of the buses or their BIUs.

3.1 OVERVIEW OF BUSES IN THE FLIGHT COMPUTER

The ERV FTP will employ four major buses to link modules within its triplex Core and from the Core to the flight sensors, actuators, and navaids. The buses are the Channel Bus (CB), the InterChannel Bus (ICB), the Input/Output Bus (IOB), and the MIL-STD-1553 Bus (1553). Figure 3-1 illustrates these buses as components of the flight computer.

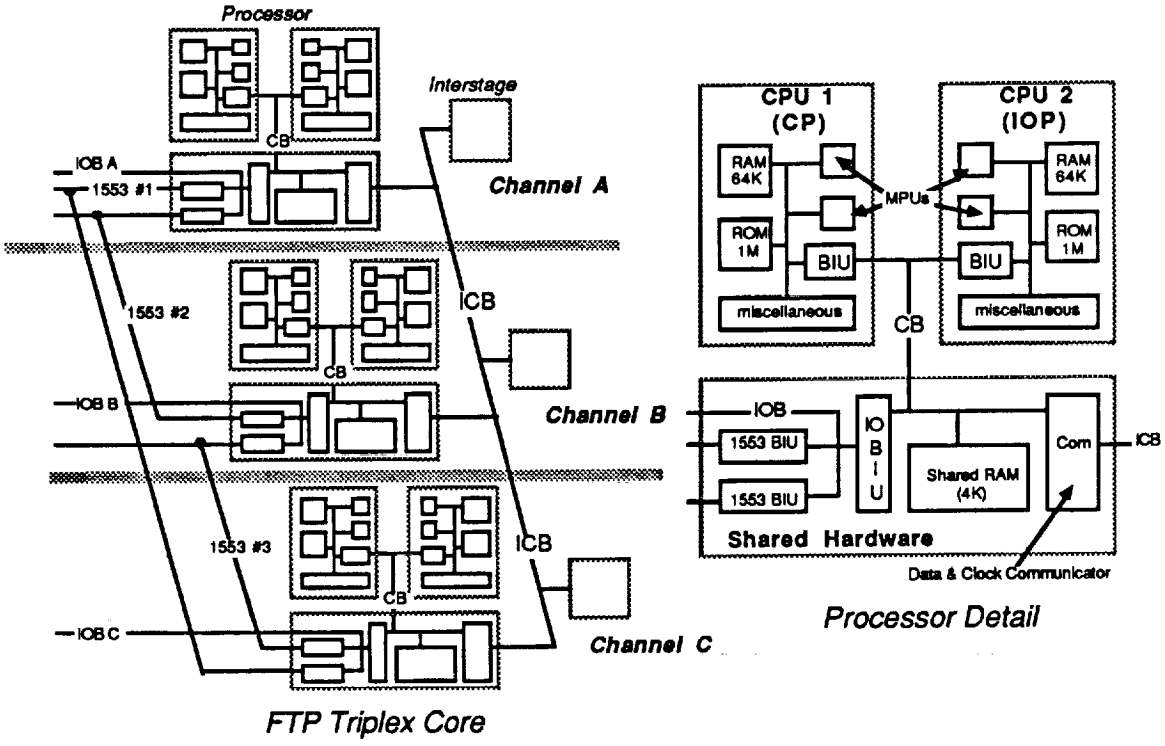

Figure 3-1. ERV Flight Computer

The Channel Bus serves to link the two CPUs in each processor to shared hardware within that module. Thus, there will be three identical CBs (one per processor) in the FTP.

The InterChannel Bus is used in the management of the hardware redundancy by connecting fault-containment regions (processors and interstages).

The Input/Output Bus provides access for each processor to all avionics external to the flight computer. Each processor will have its own identical IOB.

There will be three MIL-STD-1553 buses in the vehicle. All will be doubly cross-strapped with access to two processors. This provides tolerance of avionics failures (external to the Core) without full triple replication of hardware.

In the following sections, each of the buses will be described as to its function and suggested implementation. Any necessary discussion of its BIU will also be included. A diagram encapsulating critical information about the bus concludes each section.

An underlying theme to the implementation of all buses is hardware simplicity. This is especially true when considering (physical) bus sizing. Since a major contributor to system unreliability is connector failure, all buses have been designed to minimize the number of physical links in each bus.

3.2 THE CHANNEL BUS

The Channel Bus links the two central processing units (CPUs) in each channel to the shared hardware in that channel. Since all transactions external to the processor must use the shared hardware, the CB will be the bus used most frequently in the system.

A CPU is defined as the collection of components which could form a stand-alone microcomputer. This collection is shown in Figure 3-1 and includes the microprocessor, its coprocessor, and the volatile and non-volatile memory. The CPUs will for the most part be identical, but their functions will differ. (It is planned to appoint all I/O transactions to one CPU, allowing the other to remain computationally intensive.)

The CB will begin at the CPU boundary, as defined by Figure 3-1. Any bus internal to the CPU will not be included in the CB definition. (This "bus" will be trivial in comparison to the ones discussed here.)

3.2.1 FUNCTION

The CB will function as most general-purpose microprocessing buses with the

additional constraint of providing FTP-required signals for process synchrony. Thus, it will serve to link the two (master) CPUs to the (slave) shared hardware. It must provide for a contention or arbitration scheme between the CPUs. It also must allow all operations on it to be executed in a clock-deterministic manner in order that processors remain synchronized to within some known and well-bounded skew.

3.2.2 STANDARD BUS FEATURES

As part of the Technical Plan for this Task Assignment, it was suggested that the Channel Bus could possibly be a standard bus used by other avionics systems. The two primary military buses (MIL-STD-1553B and PI-bus) were examined as candidates for the CB. Both were found inappropriate for use based upon two major factors: 1) their general-purpose natures resulted in an overabundance of hardware complexity; and 2) they lacked capabilities to easily implement necessary mechanisms for FTP synchrony. In particular, it would be difficult to design *clock-deterministic* hardware in spec with those bus standards. (See Section 3.2.3 below for a discussion of clock-deterministic hardware.) Since these were the two prime candidates for the CB and did not meet the criteria for implementation, a non-standard bus specification was developed.

It was also suggested that upward compatibility be a feature of CB so that this architecture could serve in a larger, more complex avionics system. Since this advisory does not propose the detailed design for the CB, it also does not limit its potential. The general specification described here is useful for a wide variety of future applications.

The CB will, however, possess features standard to most microprocessing buses: slave devices will be selected by either of the CPUs by an address, data will be transferred on its own dedicated path, and control signals at the Master will govern the transaction. The following paragraphs describe these features in more detail.

A 12-bit address bus will be provided on the CB. This allows up to 4K slave devices to be individually selected. While most *commercial* microprocessing buses provide for up to 4G individually addressable slave devices, this is far above the required amount on the CB, which has only 2K of addressable memory. At the same time, 12 address lines allow for a more flexible memory-mapped scheme to access I/O than do either the 1553 or PI buses (only 32 slave devices each). Thus the lower bits of the CB address may be directly driven (or interpreted by the BIU) to the IOB.

A 16-bit data bus will transfer the data to and from the CPUs on the CB. Since most I/O will never be greater than 16-bits and the data exchange mechanism will only transfer 16-bit packets at one time (see Section 3.3.2), it is not necessary to make the bus any wider than 16-bits in width.

Address Stable (AS) and Read-Write (R-W) control lines will be provided by the master and will take on their conventional meanings.

Notably missing from this bus is a Data Transfer Complete (DTC) signal, which signifies to the master that the slave has completed its portion of the transaction. This is discussed in the next section.

Also, the CB will not have a data sizing signal: all transactions will be word (16-bits) accesses. The shared RAM, and all memory mapped registers (including memory-mapped I/O) in the shared hardware will thus be word-oriented.

3.2.3 PROCESSOR SYNCHRONY ON THE CB: CLOCK DETERMINISTIC OPERATION

A major requirement of FTP operation is that all instructions in each replicated processor follow these guidelines:

- all initial data be identical
- all processors execute the same instruction
- all instructions are synchronized to within some known and bounded skew

If these three rules are followed then all non-faulty processors will produce the same result within the skew time.

While other FTP hardware guarantees the adherence to items 1 and 2 above, the CB must implement a scheme which assures processor synchronization. Processors are microframe synchronized by the FTP fault-tolerant clocking mechanism, but the CB must provide lock-step synchronization. That is, the CB must assure each processor executes an instruction in a predetermined number of CPU clock cycles. This is termed *clock deterministic* operation.

Several previous implementations of the FTP use an inherently asynchronous bus

structure for the equivalent of the ERV FTP's CB. In the current implementation, this bus "synchronizes" its associated processors by enforcing clock deterministic operation of slave devices. Thus, while the same I/O devices in an asynchronous environment may have taken 5, 6, or 7 "wait-states" to respond to a transaction, an FTP mechanism will respond for the device in a predetermined number of wait-states. This mechanism is called a *wait-state generator*. The worst-case number, here 7, is taken as the response time to assure valid data transfer.

The wait-state generator (WSG) is initiated upon the CPU Address Stable (AS) signal. The WSG then counts a predetermined number of CPU clocks before returning a DTC to the microprocessor. This number is determined by the specific slave device that is being accessed (i.e., its address).

The CB will employ a WSG to provide clock-deterministic operation of the FTP CPUs. One WSG on each CPU will be utilized. They will be preprogrammed (hard-wired) with the appropriate number of CPU clock counts required for all CB transactions. The WSG will also be utilized for transactions internal to the CPU to maintain clock determinism of the hardware. (It should be noted that all instructions internal to the MPU chips are clock-deterministic.)

3.2.4 CPU CONTENTION FOR THE CB

As with any multiprocessor configuration utilizing shared resources, the CB must have a well-defined bus contention scheme to arbitrate requests between the CPUs for bus mastership. Similar to the clock-deterministic requirements for data transfer, the implementation of CB arbitration must assure processor synchrony.

There are several methods which could be used to arbitrate between CPUs while still providing for processor synchrony. A simple method is to base arbiter decisions on the fault-tolerant clock (FTC) - a system-wide, synchronized signal which oscillates at a frequency lower than the CPU clock (1 MHz v. 25 MHz, see Section 3.3.1). With two CPUs per channel, each phase of the FTC can be used to coordinate bus mastership on the CB: when the clock is high, one CPU is allowed CB access; when FTC is low, the other CPU can take mastership. More complex schemes can be implemented using the basic premise that the FTC is a longer period (than the CPU clock), synchronized, system-wide clock.

For purposes of this study, it shall be assumed that the CPU contention on the CB is arbitrated using the simple technique of FTC phase state. For the final ERV implementation, this may not be feasible, since the CPU may require use of the bus for time lengths greater than the FTC period. More complex schemes may use a multiple of the FTC for arbitration, a weighted scheme which gives preference to the one CPU (the IOP), or less stringent requirements on time for taking bus mastership (perhaps end of FTC phases would be legitimate if the CB is free). Nevertheless, the scheme must assure processor synchrony, and the FTC is an obvious means of reaching that end.

The arbiter should reside in the shared hardware. Four signals will be used to facilitate arbitration: two request lines and two acknowledge lines, one per CPU. While a simpler means could probably be instituted (using only the FTC signal and allowing CPUs to resolve contention themselves?), four signals allow for flexibility in this stage of the design.

3.2.5 CPU CLOCKING ON THE CB

A final requirement of the CB is to provide a common clock to each CPU, termed appropriately, CPUCLK. The CPUCLK will be the high-frequency system clock which controls the operation of the CPUs, including especially, the microprocessing units (MPUs). Each CPU can be thought of as a state-machine, where next states are determined solely by present state conditions. The CPUs are clocked at regular intervals to execute all operations.

The CPUCLK will be controlled by the FTC mechanism. The FTC is a divided-down version of the CPUCLK. A sequencer in the FTC synchronizes processors by regulating the CPUCLKs in order to digitally phase-lock the common FTC signal. This is done by "deleting" or "delaying" pulses in the CPUCLK stream to assure that the same number of CPUCLK edges are received by all processors during one FTC period. This process is described further in Section 3.3.1. Thus the CPUCLK will be a high-frequency regulated pulse train, as shown in Figure 3-2.

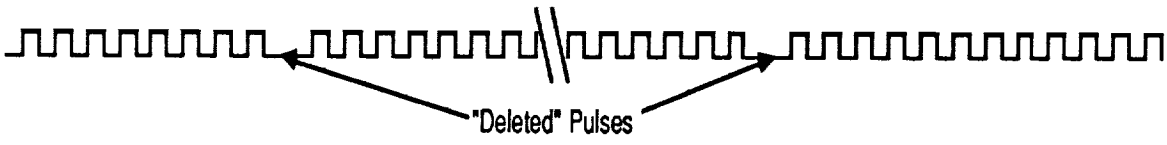

Figure 3-2. Pulse Train of CPUCLK

3.2.6 BIU FOR THE CHANNEL BUS

Bus Interface Units (BIUs) must be provided from the CPUs to the CB. Minimally, each BIU must properly drive, terminate, and receive the CB signals. The BIU must also perform some low-level tasks, such as bus contention decoding. The extent of the complexity of the BIUs rests with the final avionics design.

3.2.7 CHANNEL BUS IMPLEMENTATION SUMMARY

Figure 3-3 depicts the CB implementation suggested in this section. It shows that there are a total of 34 signals on the bus: 16 data, 12 address, AS, R-W, and 2 each of Bus Requests and Bus Grants. (Note that each CPU only transmits one request and receives only one grant.)

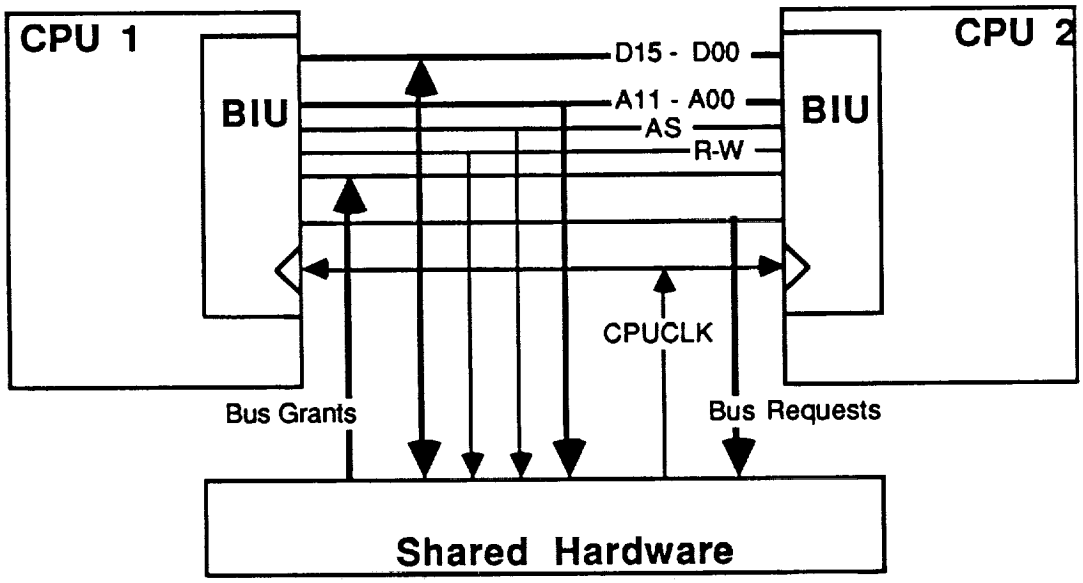

Figure 3-3 Channel Bus Signals

3.3 THE INTERCHANNEL BUS

The InterChannel Bus (ICB) serves to connect all fault containment regions (FCRs) in the ERV flight computer. As in all FTP architectures, each channel contains two FCRs: a *processor* and an *interstage*. The mechanism in each processor which interacts with other processors and interstages is termed a *communicator*. The interstage's sole function, on the other hand, is to interact with processors.

The ERV FTP channels must communicate data and clock information in order to maintain a fault-tolerant system. The two pieces of information are treated separately in the ERV FTP to avoid providing an overabundance of hardware and interconnections. That is, the data and clock exchange mechanisms have two separate topologies.

3.3.1 CLOCK EXCHANGE MECHANISM

The ERV flight computer redundant channels will be tightly synchronized at the instruction level. This FTP technique bounds the skew between processors - one of the requirements of FTP operation. Discrete processor clocks are synchronized indirectly by using a lower frequency, system-wide *fault-tolerant clock* (FTC). Processor clocks are synchronized using a digital phase-lock loop technique which allows each processor to "adjust" its own clock based upon its position relative to the FTC.

The FTC period is generally an order of magnitude longer than processor clocks. The exact relationship is application-specific and depends upon the data exchange rate required by the system. For example, a 200 KHz FTC synchronizes the 8 MHz processor clocks in the AIRLAB FTP [Gauthier].

The method of processor clock synchronization is illustrated by Figure 3-4 and can be briefly described as follows: Each processor creates a version of the FTC by dividing its processor clock (CPUCLK) by a predetermined factor. (For the AIRLAB FTP, the factor is 40: [8 MHz] ÷ 40 = [200 KHz].) FTCs are broadcasted from processors and voted at each interstage. A "voted" copy of the clock is then transmitted by each interstage to all processors. At the processors, a voted copy of the clock is again created. The resultant signal is compared to the FTC created by the processor (CPUCLK divided down). A processor then regulates its FTC (and CPUCLK) based upon the results of the comparison. This implementation is the classic method designed by Smith and McKenna.

In the figure, all hardware in the *processor* - except for the CPUs - is located in the shared hardware and is collectively termed the *FTC Communicator*. The very brief description given in this section can be supplemented by Smith's [Smith 81] work on fault-tolerant clocking schemes.

The configuration of the interconnection for the FTC will be the same as implemented in the NASA/Langley AIRLAB FTP. This fully connected network is illustrated in Figure 3-5.

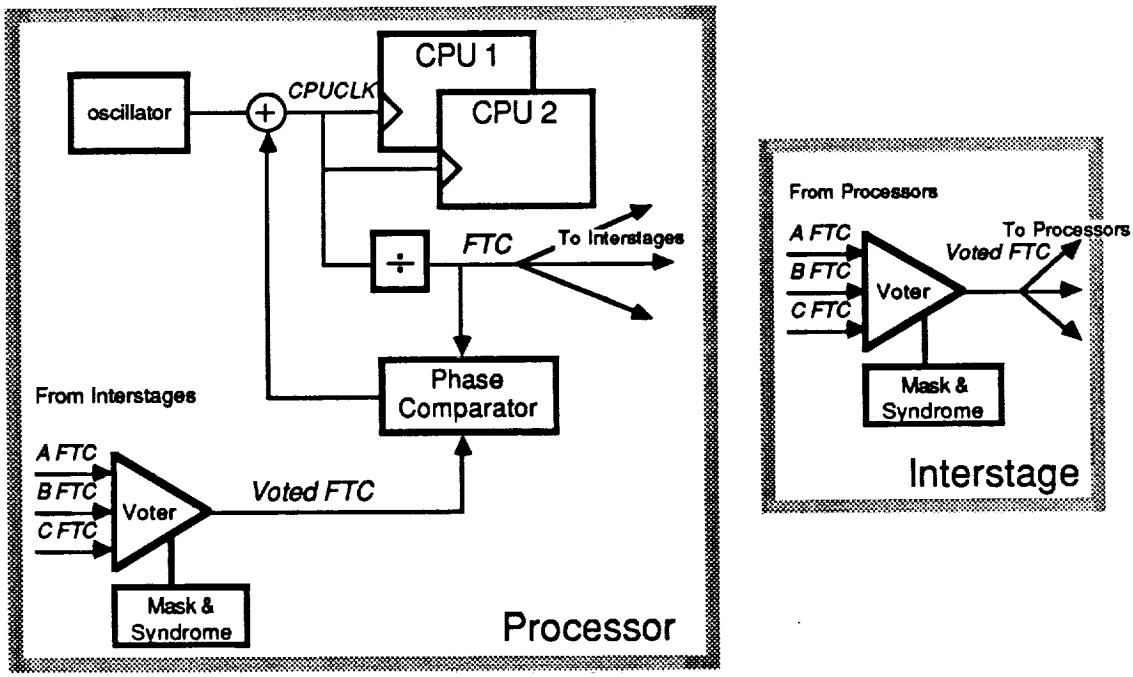

Figure 3-4. Digital Phase-Lock Loop Implementation of the Fault-Tolerant Clock

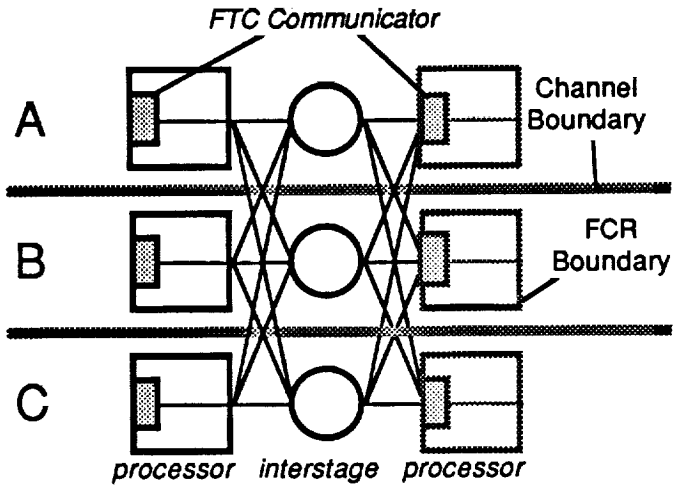

Figure 3-5. ERV FTP Fault-Tolerant Clock Network Topology

3.3.1.1 FAULT-TOLERANT CLOCK SPEED

The frequency of the FTC depends upon the required data exchange rate for the application. In order to distribute a congruent copy of data to all non-faulty processors (an FTP requirement), the processors must be synchronized within some known skew. Since all (non-faulty) processor clocks are bounded within some skew of the FTC, this signal is used to synchronize data exchanges.

All input to and output from the FTP must be exchanged between channels. In the ERV, an extremely high I/O rate will be demanded of the flight computer. Thus, the data exchange rate - i.e., the FTC frequency - should be as high as feasibly possible.

It has been determined that the FTC period will be 1 μs. This 1 MHz frequency will adequately support I/O rates required by the vehicle. In addition, it can provide phase-locking of CPUCLKs on the order of 20-30 MHz without a large percentage skew. It appears that the processor CPUs will be running at frequencies in this range (see Section 4.0.)

3.3.1.2 FTC MASKING AND SYNDROME INFORMATION

One requirement for the clocking mechanism in the FTP is the provision for mask and syndrome information at both clock voters (processor and interstage). This allows the hardware to isolate and detect clock failures. It also allows the FTP Fault Detection, Isolation, and Reconfiguration (FDIR) software to reconfigure the hardware in order to permanently mask failed elements in the system. Thus, the mask and syndrome information must be available to the CPUs.

At the processor voter, providing such hardware does not present a problem to the design. Readable and writable registers can be provided at the FTC Communicator. These registers are easily mapped as part of the shared hardware. The CPU performing FDIR can access them at will.

The process is not as simple for the interstage mask and syndrome registers. *In order to assure resilience to Byzantine failures, clock masks at the interstage cannot be received from a single processor.* Should that processor be faulty, it could pass an incorrect mask to the interstage voter, causing the interstage to have a value dissimilar to the other interstages' values which, in turn, causes that interstage's voter to behave incorrectly. Thus, a failure in one fault-containment region could propagate to another region - an unacceptable consequence for fault-tolerant systems.

One method to assure congruent masks is to provide a voter for them. All processors, then, should broadcast the three mask values (ENABLE_A, ENABLE_B, ENABLE_C) to all interstages. Prior to voting the clock values, the clock masks must be voted.

Syndrome (suspected error) information need not be as complicated. Each interstage voter can broadcast its syndrome (on command) to only its own processor without endangering system integrity. However, this data must be exchanged between all channels using the data communication network before it is analyzed.

The resulting required FTC mask and syndrome configuration can thus be described by Figures 3-6 (Topology), Figure 3-7 (Interstage Hardware), and Figure 3-8 (FTC Communicator Hardware).

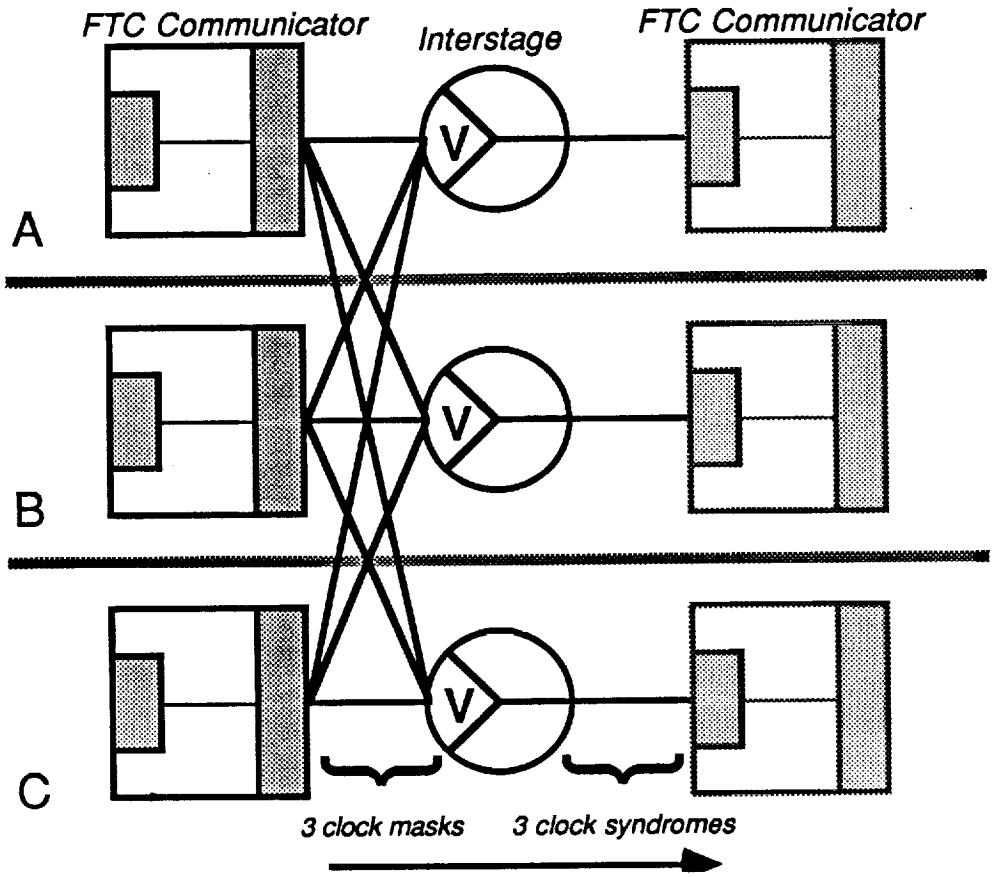

Figure 3-6. FTC Mask and Syndrome Topology

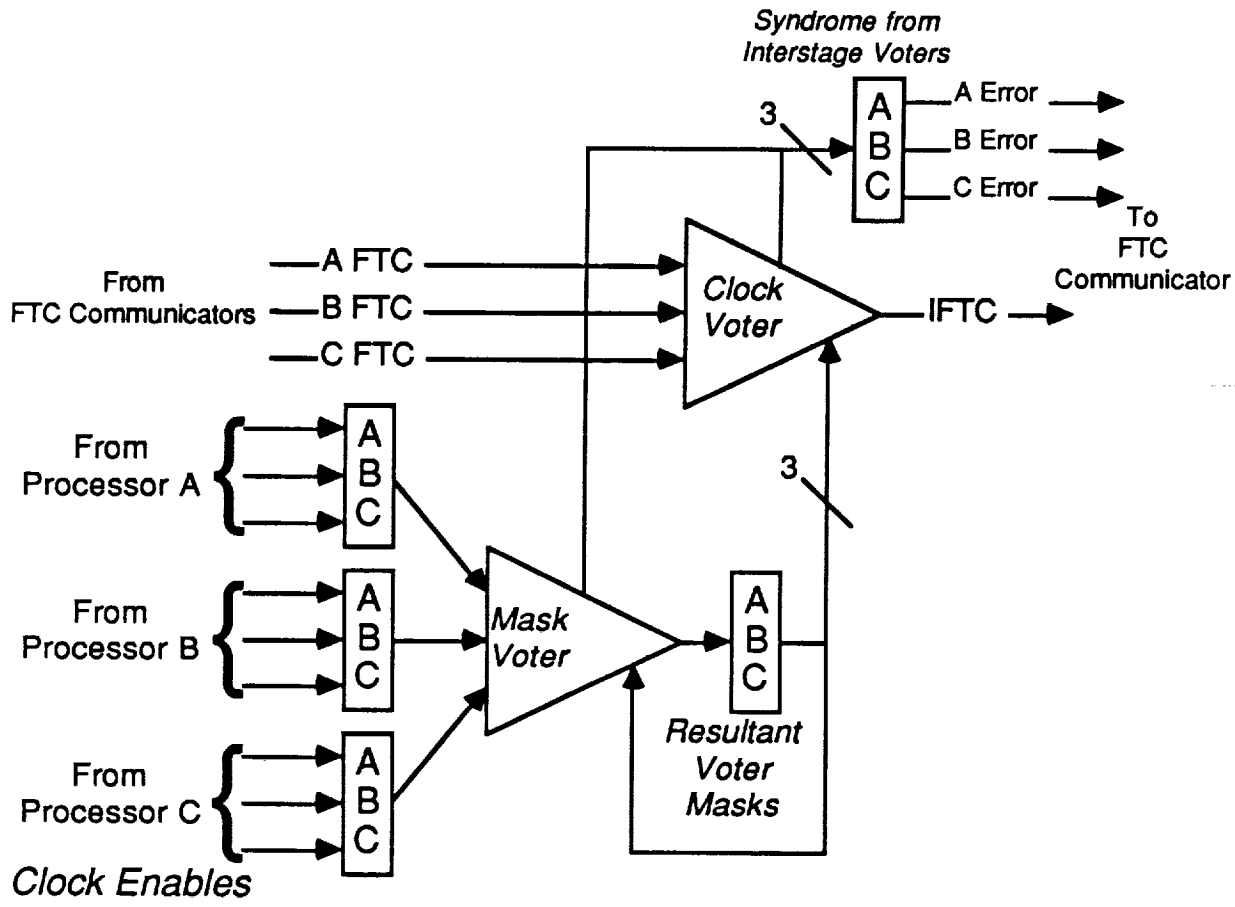

Figure 3-7. FTC Masks and Syndromes at Interstage

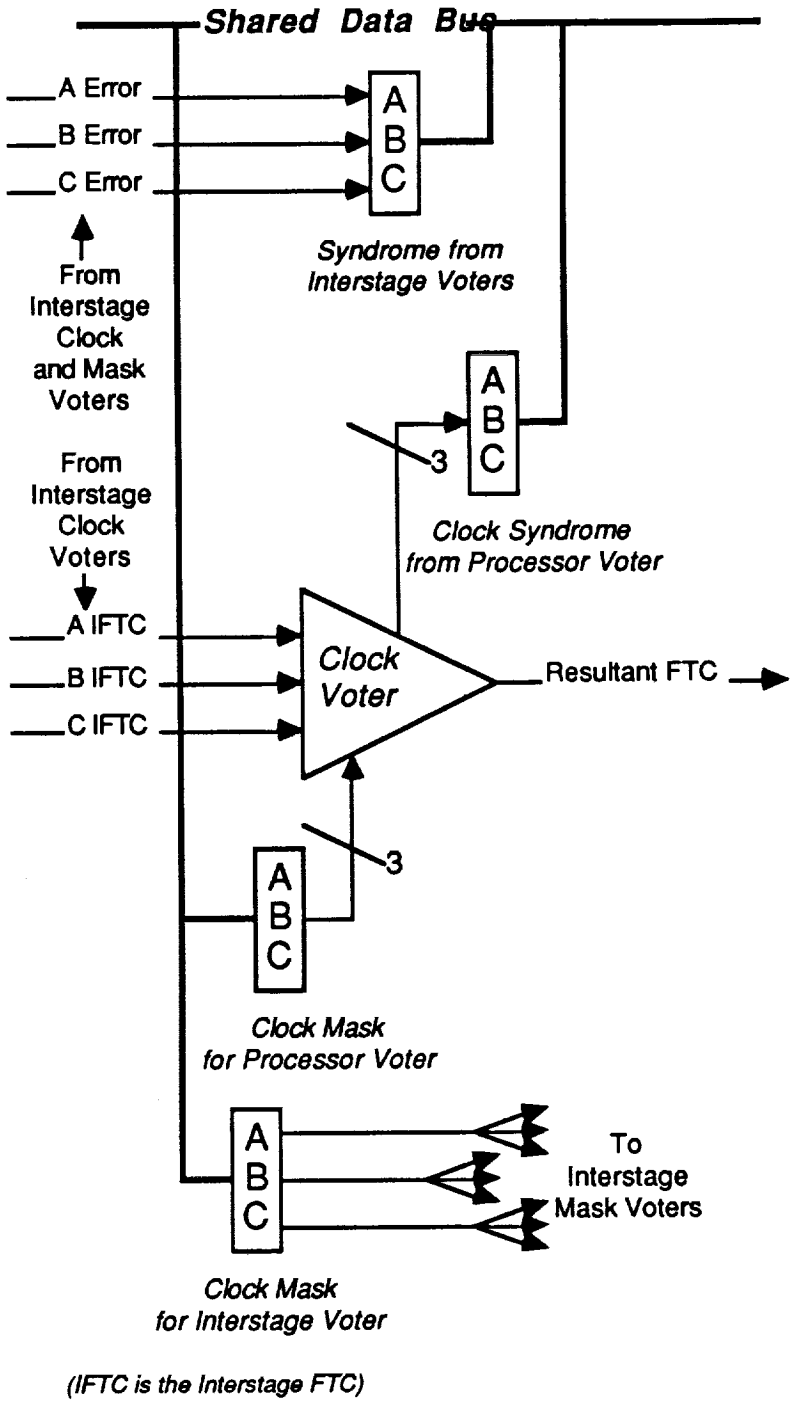

Figure 3-8. FTC Masks and Syndromes at FTC Communicator

3.3.2 DATA EXCHANGE MECHANISM

The ERV FTP data exchange hardware must assure that a congruent copy of all data

is received by all non-faulty processors. Typical purposes of data exchanges include proliferation of single source input data (data available only to one channel), check of critical data (especially output data), fault detection, isolation, and reconfiguration (FDIR) software routines, and software synchronization of processors.

While the ERV processing core may operate on many floating-point numbers, there does not seem to be a great need to exchange such large values. This is because the major type of data traffic through the exchanger will be I/O data, which will primarily be no greater than 16-bit values. Although some complex navaids, including the LINS and MLS, may require 32-bit I/O fields, the frequency of these data rates are small in comparison to those with 16-bits in length. Based upon this rationalization and in order to keep the data exchanger as simplistic as possible, the ERV FTP data exchange mechanism will pass only 16-bit words.

A major constraint in the ERV design is the packaging of the flight computer. As mentioned in the introduction to this section, one important criteria to be evaluated during this design phase is the amount of interconnections between hardware modules. Specifically, the concern here is the physical size of the ICB. Different versions of the CSDL FTP data exchanger which vary the ICB data bus width have been implemented. Papadopoulos integrated a serial bit stream exchange mechanism for the Iron Bird simulation of the F-8 Digital Fly-By-Wire program at NASA's Dryden Research Center [Hopkins, Lala, Smith]. As a research and development effort, this implementation also allowed a variety of experimental tasks to be performed at the interstages. For the NASA Langley Research Center's AIRLAB FTP, McKenna and Hughes redesigned the data exchanger to implement byte parallel data widths [Gauthier]. In addition, this latest method simplified the interchannel communication by reducing the functionality at the interstages, thereby reducing the unit's complexity. A description of the differences between the two configurations has been presented elsewhere [Lala 85], [Lala 86].

The ERV FTP data communication scheme will employ the simplistic elements of both designs mentioned above: a serial data stream with reduced interstage complexity. The topology is shown in Figure 3-9. In this illustration, all data paths are serial and accompanied by a high-speed clock. Since the rate of word exchanges will be governed by the FTC speed, the bit rate must be *at least* 16 times the frequency of the FTC. This will insure that the word to be exchanged can be transferred through the communication network within one FTC period. With a 1 MHz FTC rate, then, a serial data stream must be driven at minimally 16 MHz. To allow for hardware overhead delay times, this rate will

be 25 MHz. That is, the data exchange mechanism for the ERV FTP will drive a serial bit stream at 25 MHz.

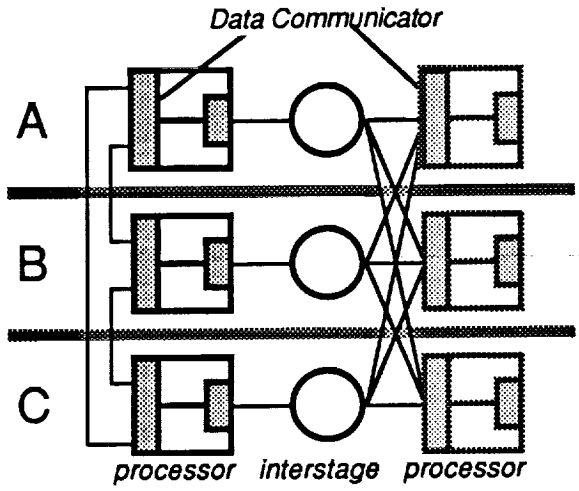

Figure 3-9. Data Exchange Topology

3.3.2.1 HARDWARE IMPLEMENTATION AND VOTING

The data exchange hardware can be divided into 3 sections which follow the flow of data from source to final destination: communicator-to-communicator, communicator-to-interstage, interstage-to-communicator. These sections are described below.

3.3.2.1.1 Communicator-to-Communicator Hardware

For single source exchanges (the most frequent type of data exchange), the source processor transmits its data to the other processors. This precedes the broadcast of data to the interstages. A block diagram of the hardware is seen in Figure 3-10.

As seen from the figure, data is sourced from one communicator to the others through serial bit streams. Both communicators are able to execute the same type of exchange as long as the Source Data Register has a unique memory location, since the controllers can decode the Channel Bus address and recognize the operation. The FTC is used to synchronize the execution.

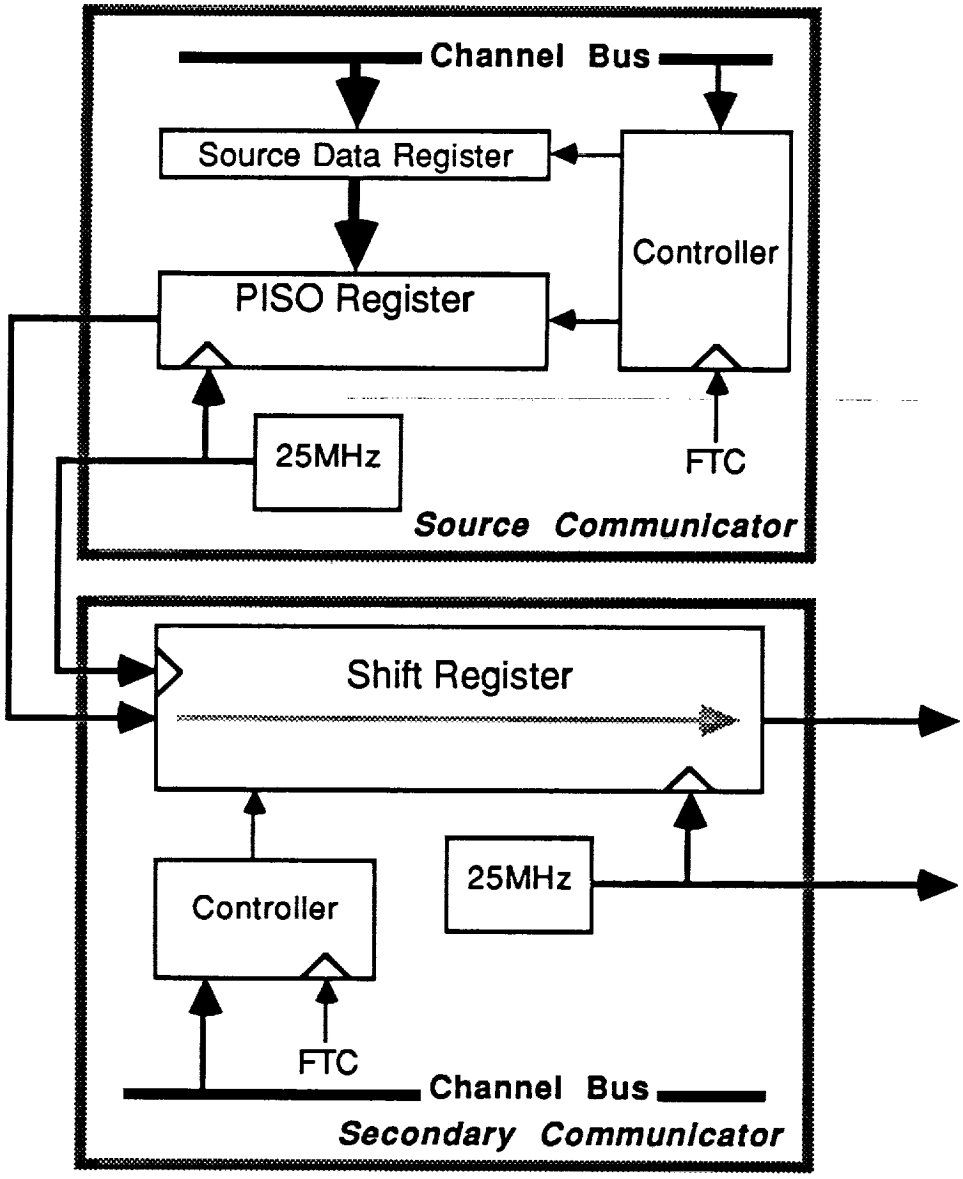

Figure 3-10. Communicator-to-Communicator Hardware

It is important to note that each communicator uses independent clocks for data broadcast. This insures they follow the requirements of a fault-containment region.

3.3.2.1.2 Communicator-to-Interstage Hardware

Once the non-source communicators (simultaneously) receive the data to be exchanged, all three communicators transmit the data to their own interstages. This is a simple task and is outlined in Figure 3-11.

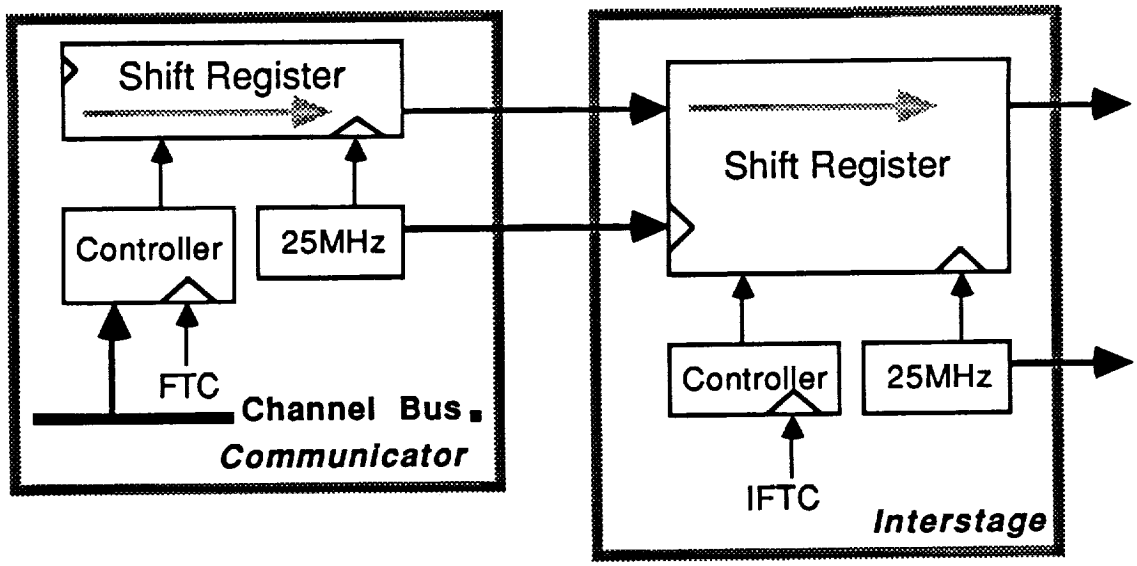

Figure 3-11. Communicator-to-Interstage Hardware

Notice that, as in the Hughes implementation of the data exchange hardware for the AIRLAB FTP, there is no voting of data at the interstage. This is because the interstage need only be a repeater of the data. While voting at the interstage does provide further information for the FDIR regiment, it does not provide a guaranteed increase in system reliability (it does not allow for 100% coverage of the second failure, for instance). It also serves to make the interstage hardware more complex (and thus more likely to fail) and is therefore not included in the ERV FTP.

The Fault-Tolerant Clocks are voted (Section 3.3.1) at the interstage, however, because the clocking at each FCR must be independent from the clocking of the others. The FTC is used as the clocking mechanism at the interstage and at the same time synchronizes each interstage to the remainder of the system. Since clocking at each FCR must be independent, the FTC cannot originate from a single source. Voting must be employed to make the signal independent from the other FCRs.

3.3.2.1.3 Interstage-to-Communicator Hardware

The final leg of all data exchanges is the interstage-to-communicator transmission. Each interstage broadcasts its data and 25 MHz clock to all communicators. Once the data is stable (all 16-bits) at the communicator, a vote is taken. The resultant copy is then placed in a register accessible from the Channel Bus. As with the clock voters, masks and syndrome bits are provided for the FDIR routines. The hardware for the data

communicator receiver with voter is diagrammed in Figure 3-12.

At 40 ns per bit (25 MHz), hardware overhead per data exchange cycle is 360 ns. This includes transmission time and all hardware (including the voter) delay times. This seems to be plentiful for the compact implementation of the FTP in the ERV.

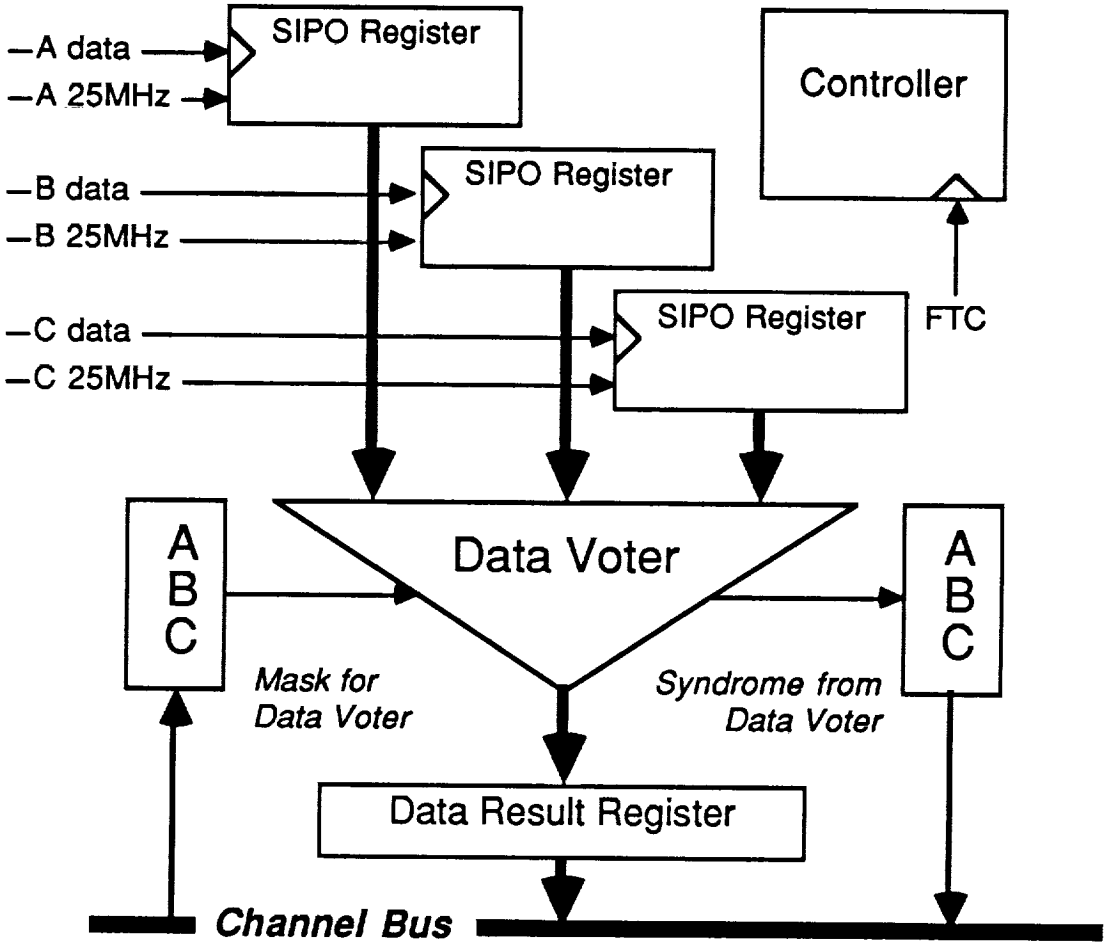

Figure 3-12. Hardware for Communicator Receiver with Voter

3.3.3 ICB BUS INTERFACE UNITS AND TRANSMISSION LINE CHARACTERISTICS

The BIUs from the ICB include all hardware described in the previous sections. The BIUs must also include the line drivers and receivers for the data transmission and reception. The 40 ns per bit rate is not unrealistic using standard transistor-transistor logic (TTL) technology gates, but other sources for data communication should be examined in the final design.

All interconnection will be unidirectional, except for the data communicator-to-communicator links, which will be bidirectional. A final analysis during the detailed design phase should consider whether or not the links need to be differentially driven. This would be necessary if the FCRs are located a large distance (≥ 2 feet) from one another or if the environment of the ICB will be particularly noisy (including noise induced by the FTP itself).

3.3.4 SUMMARY OF THE ICB IMPLEMENTATION

The ICB will consist of two separate topologies: a clock exchange mechanism and a data exchange mechanism. The interconnection, and implementation of the hardware for both can be functionally separated, although the data exchanger relies heavily on the fault-tolerant clock exchanger for synchronizing the inter-channel communication. The figures of the ICB topology (Figures 3-5, 3-6, and 3-9) describe the ICB interconnection fully.

3.4 INPUT/OUTPUT NETWORK

To provide the centralized ERV flight computer with access to the variety of input and output (I/O) located throughout the vehicle, there will be a distributed I/O network consisting of three MIL-STD-1553 buses, each doubly cross-strapped to three FTP I/O buses (IOBs). This section describes the simple functionality and protocol of the IOB, its BIU to the FTP channel, and the IOB-1553 BIU.

3.4.1 THE IOB

The IOB will provide a link between each channel's shared hardware to several custom I/O devices and the three MIL-STD-1553 buses that are distributed throughout the spacecraft. The configuration is depicted in Figure 3-13, a block diagram of the ERV flight computer.

3.4.1.1 I/O INTERLOCK

An IOB BIU will interface the IOB to the channel's shared hardware. This BIU will have the capability to shut down the IOB to prevent a failed processor from babbling on it. This type of operation is called *Output Interlock*. This section briefly describes this function and a possible implementation of it.

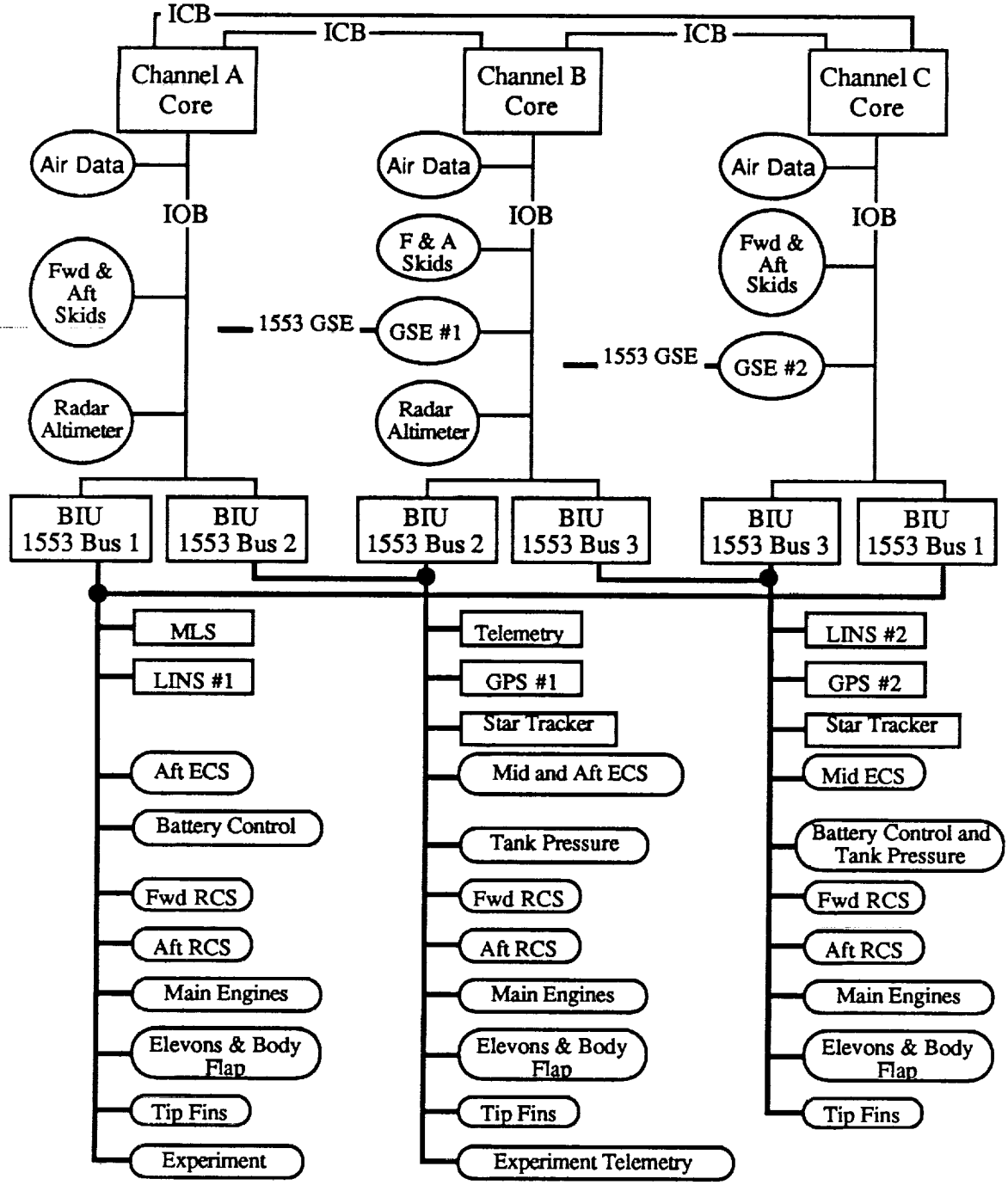

Figure 3-13. *I/O and MIL-STD-1553 Buses in ERV Flight Computer*

Should a channel fail in such a way as to babble on its IOB, severe consequences could result. Output could be actuated in a manner detrimental to vehicle health. The FTP must be able to shut down this failed channel's output to the IOB. This is accomplished using the Output Interlock. The Interlock receives an OUTPUT-ENABLE signal from each of the channels. A two-of-three voter at the BIU determines whether or not to allow a

channel to drive the IOB signals. This means, however, that additional interconnections must link the processors.

A total of four unidirectional links will connect all processors, two originating from and two incoming to each channel. Note that these need not undergo a source congruency exchange, since all non-faulty processors will transmit a congruent ENABLE signal for the specified channel. Figure 3-14 illustrates the interconnections of the Interlock.

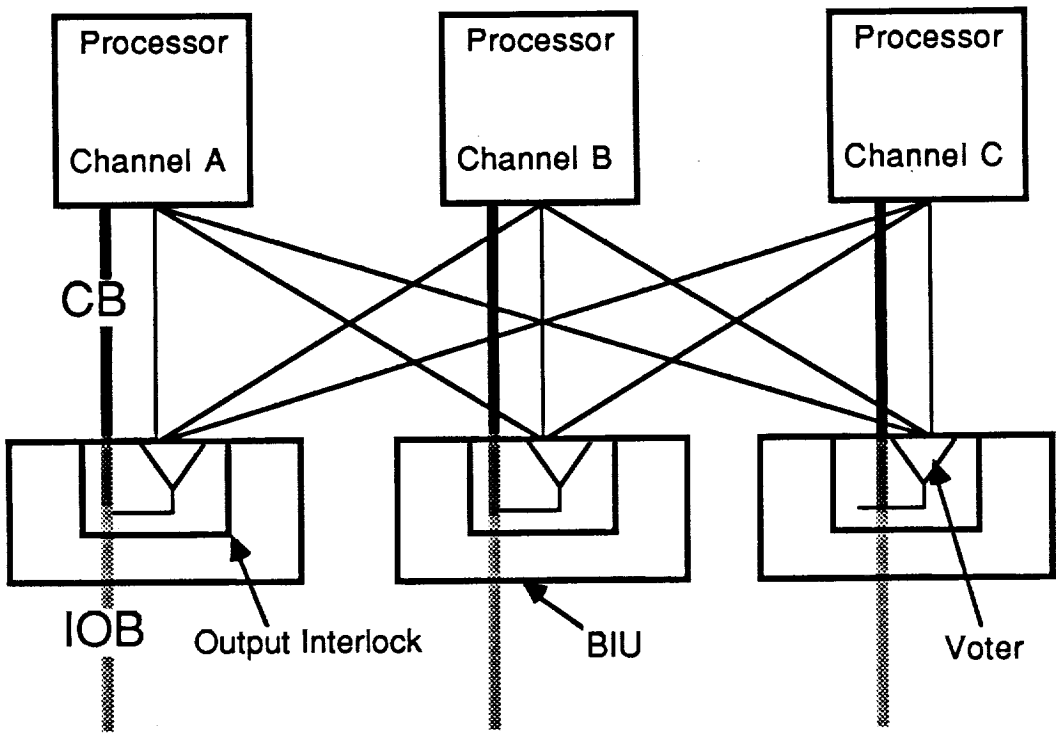

Figure 3-14. FTP Output Interlock

3.4.1.2 IMPLEMENTATION OF IOB

The following characterize the IOB:

- All devices - including the 1553 BIUs - will be memory-mapped. Intelligent control from memory-mapped registers will be utilized when necessary.
- There will be 1 Kwords of address space (10 address lines).
- There will be a 16-bit wide (word) data bus.
- All transactions will be word-oriented.

- There will be Address Stable and Read-Write signals, taking on their conventional meanings.
- There will be no Data Transfer Complete (DTC) signal.
- The IOB BIU will be the sole bus master.
- There will be no clock on the bus.

From the above items, one can see that the IOB will be extremely similar to the Channel Bus with reduced functionality.

3.4.2 1553 BIUs

Each of the three MIL-STD-1553 buses will each be attached to two IOBs through a 1553-BIU, as seen Figure 3-13. These BIUs serve as the 1553 masters for all transactions. Since only one of the BIUs can be a 1553 master at a time, the *dynamic mastership* capabilities of the 1553 will be utilized.

Under normal operation, each channel will control exactly one 1553-BIU and thus indirectly serve as that bus' master. Upon failure of a channel, the mastership of its 1553 will be dynamically reconfigured to another channel. For instance, should Channel A fail, Channel C could become master of 1553 #1. This assures that all I/O devices can be accessed from a duplex FTP configuration.

The devices on the 1553 buses will be of two types: "off-the-shelf" standard 1553 slaves found commonly in most aerospace applications and ERV-custom slaves. The latter type will conform fully to MIL-STD-1553 specification, but will be designed expressly for the ERV application. These devices will employ a standard interface unit described in Section 5.2.2. They consist of:

- 1553 line driver and receiver
- 1553 "address" decoder
- ADC or DAC module with data latches
- Interface to the sensor/actuator

The 1553 bus is inherently dual-redundant. This feature will be included in the ERV implementation.

3.4.3 SUMMARY OF THE IOB IMPLEMENTATION

To summarize, the IOB will be fashioned much like the CB with a reduced address range. Also excluded from the IOB definition will be bus arbitration and clocking. An Output Interlock will assure the shutdown of the output from all failed processors. This Interlock will be part of the IOB BIU. The three MIL-STD-1553 buses will be cross-strapped to the IOB through specialized BIUs as well.

4.0 MICROPROCESSOR SELECTION

This section describes the microprocessor selection process, examines the qualities of some of the most up-to-date microprocessors, and selects one for the ERV flight computer CPU.

Microprocessor selection for a specific application is based on a number of factors. These factors include characteristics such as performance, memory and I/O capability, reliability, power dissipation, size, and software support. Comparing these microprocessor features with the application's requirements determines whether the microprocessor will meet the application's needs. The main characteristics considered in the microprocessor selection for the ERV flight computer were performance, memory capability, and reliability.

4.1 MICROPROCESSOR PERFORMANCE

Microprocessor performance is measured in *instructions per second* and is calculated by measuring the time it takes the microprocessor to process a particular program. The ideal way to measure a microprocessor's performance for a specific application would be to run the application program on the microprocessor. This, obviously, is impossible to do in the preliminary design stages of a computer system. As an alternative, vendors of microprocessors provide performance benchmarks for their products. Benchmark programs resemble typical scientific programs that are run and timed on the microprocessors to give the performance benchmarks.

At first glance, performance benchmarks can be very deceiving. Many times vendors claim to have a microprocessor with a very high performance level but don't specify the benchmark that was used to obtain the performance value. Comparing processors' performance levels without knowing from which benchmark program they were obtained can be misleading. For instance, some benchmarks use only floating-point calculations while others use only integer calculations. Comparing a floating-point benchmark to an integer benchmark is comparing apples to oranges. Also, because most of the standard benchmarking programs are written in high level languages they measure compiler and operating system efficiency as well as pure microprocessor performance. When a vendor compares its processor's performance directly with another processor, it must be made clear what type of systems the processors are running on and what types of

compilers are implemented, otherwise the comparison is meaningless. It is assumed that when a vendor quotes a standard performance benchmark for its own microprocessor, a compiler of maximum efficiency was used, thus enhancing the processor's performance level to obtain the benchmark.

The Whetstone benchmark is used to measure floating-point performance. It was originally written in ALGOL but is now almost always run in FORTRAN. The program includes procedure calls, array indexing, and transcendental operations, as well as floating-point operations. One pass through the program executes one million Whetstone instructions. Because it is written in a high level language, the Whetstone benchmark measures compiler efficiency as well as processor performance. The benchmark can, however, be compared quite easily over different systems because of the single language (FORTRAN) implementation.

The Dhrystone benchmark, like the Whetstone, represents a typical scientific program, but it does not implement any floating-point calculations. It makes no calls to the operating system so it can be accurately compared over different systems. The Dhrystone benchmark is written in Ada® with C and Pascal translations, and is a good measure of compiler maturity and processor performance.

The Linpack benchmark program has a high percentage of floating-point operations. It is written in FORTRAN and comes in two versions, single and double precision floating-point operations. The Linpack benchmark measures floating-point processor performance and compiler efficiency.

The Digital Avionics Instruction Set (DAIS) mix has been adopted by the Air Force to measure the performance of 1750A microprocessors. The DAIS mix is different than the other benchmarks described above in that it is not a program that can be run on a processor. It is a list of instructions that are found in typical scientific programs. The performance value for a microprocessor using the DAIS mix is obtained by summing the process times for the instructions in the mix and is given in instructions per second.

Because of the floating-point requirements of the navigational and guidance algorithms to be processed by the ERV flight computer, the Whetstone benchmark is best suited for performance measurements of CPU candidates. The DAIS mix, because of its

---

®Ada is a registered trademark of the U.S. Government (Ada Joint Program Office).

floating-point considerations, is used in comparing the 1750A microprocessors.

4.2 THE 1750A INSTRUCTION SET ARCHITECTURE

The Air Force has adopted a standard hardware Instruction Set Architecture (ISA), MIL-STD-1750A, for microprocessors used in spacecraft flight control. This was done to standardize the software support for microprocessors, thus creating competition among vendors to reduce costs of development systems. Without a standard ISA, each microprocessor has its own software support tools making the cost of microprocessor-based systems high. Another advantage to having a standard ISA is that mature software assemblers and compilers are available off the shelf. Compilers for microprocessors without standard ISAs are usually less efficient because of their immaturity.

A big disadvantage in setting a standard ISA is that soon after it is set, it tends to be out of date from rapidly moving technology. The main disadvantage of the 1750A ISA is that it is only a 16-bit architecture. Most of today's 32-bit microprocessors can out perform any 1750A microprocessor. Without a Memory Management Unit, 1750A microprocessors can only access 16K bytes of RAM or I/O space where a 32-bit address space covers 4G bytes.

4.3 MICROPROCESSOR STATISTICS

Figure 4-1, on the following page, lists the microprocessors considered for the ERV flight computer along with some of their pertinent characteristics:

4.4 MICROPROCESSOR SELECTION

A comparison of all the microprocessors resulted in selecting the Motorola 68020/68881 combination for the ERV flight computer. The following paragraphs describe the most qualified microprocessors in greater detail and then a rationale for the selection of the 68020/68881 pair is given.

Microprocessor	Architecture	Technology	Throughput	Memory Capability	Power, Size
Motorola 68020/68881	32-bit	CMOS	1.24 MWhets Double-precision	4 Gbytes	1.75 W/chip 2 chips, PGA
Fairchild CLIPPER™	32-bit	CMOS	2.0 MWhets Double-precision	4 Gbytes	6 W, 3 chips 3 x 4.5" PC board
AT&T WE®32200	32-bit	CMOS	2.5 MWhets Single-precision	4 Gbytes	2 W/chip 3 chips, PGA
Intel 80386/80387	32-bit	CHMOS III	1.8 MWhets Single-precision	4 Gbytes	2.25 W/chip 2 chips
Nat. Semi. NS32332	32-bit	NMOS	.130 MFlops Linpack Benchmark	4 Gbytes	2.0 W/chip 2 chips
Performance Semiconductor PACE1750A	1750A	CMOS	1.7 MWhets Single-precision 2.6 MIPS DAIS	1 MWords w/MMU	2.5 W 2 chips
McDonnell Douglas MDC281	1750A	CMOS	0.4 MWhets Single-precision	1 MWords w/MMU	2 W/chip 3 chips
Fairchild 9450	1750A	Bipolar	0.7 MIPS DAIS	1 MWords w/MMU	3 Watts
Texas Instruments VHSIC DPU	1750A	VHSIC Bipolar	2 MIPS DAIS	1 MWords	2.3 Watts One chip
Delco Magic 5	1750A	CMOS	1 MIPS DAIS	1 MWords w/MMU	2.2 watts

Figure 4-1. Comparison of Microprocessor Characteristics

4.4.1 1750A ISA

The overall statistics shows that the 1750A microprocessors cannot perform on the same level as the 32-bit architectures. Only one, the PACE1750A, is in the range of the 32-bit microprocessors with 1.7 MWhets (single precision) at 40 MHz. The PACE1750A cannot, however, perform double-precision floating-point calculations which will be required in the processing of ERV navigational algorithms.

4.4.2 FAIRCHILD CLIPPER

The CLIPPER™ is a three chip set mounted on 3.0 x 4.5 inch printed circuited board with a 96-pin connector attaching it to the rest of the system. The chip set includes a CPU/FPU chip, an instruction cache-MMU chip, and a data cache-MMU chip. The inexpedience of physically separating the CPU from the rest of the system in an avionics flight computer is the primary reason for not selecting the CLIPPER for the ERV computer.

It is also a relatively new product and Fairchild has no intention of producing a military specification version of the CLIPPER in the immediate future.

4.4.3 AT&T WE 32200

The AT&T 32-bit three chip family consists of the CPU (WE® 32200), a Memory Management Unit (WE 32201), and a Math Accelerator Unit (WE 32206). This group can perform 2.5 single-precision MWhets, but no double-precision benchmarks were available.

On a rough scale, the WE 32200 performs 2.5 single-precision MWhets. With no overwhelming performance benchmarks, three chips to its set compared to Motorola's two, and no military specification version in the near future, the WE 32200 microprocessor was not chosen as the CPU for the ERV flight computer.

4.4.4 INTEL 80386

Intel claims a Whetstone benchmark for its 80386/80387 combination by C version of the Whetstone benchmark program which has never been verified to be comparable to the standard FORTRAN or ALGOL versions. The C program requires single-precision rather than double-precision floating point calculations. With this in mind, the 80386/80387's performance does not compare with the Motorola 68020/68881.

4.4.5 NATIONAL SEMICONDUCTOR NS32332

The NS32332/32081 combination performs as well as the Motorola 68020/68881 under the Linpack benchmark, according to National Semiconductor. However, the military version of their 32-bit processor, the NS32C032 performs only one third as well as the NS32332. No information on the Whetstone benchmark was available for the NS32332.

4.4.6 MOTOROLA 68020

The 68000 series is a proven product by Motorola. The 68020 combines a pipe-lined architecture with on-chip cache memory to enhance its performance. The 68881 floating-point coprocessor conforms to the IEEE P754 standard. It contains eight general purpose floating-point data registers, each supporting a full 80-bit extended precision data

format. With software support in many high level languages including C and Ada, and a military specification version in the planning, the 68020/68881 combination is the favorable choice for the ERV flight computer CPU.

On strictly a performance scale, the top microprocessors are quite close. Other factors, however, separate the MC 68020/68881 from the rest. These factors include development risks with new products such as the CLIPPER and the WE 32200 series, reliability measures (military specification parts have a much higher reliability than commercial parts), and lack of existing software support. The MC 68020/68881 is the best candidate for the ERV flight computer microprocessor.

4.4.7 TRADEMARKS

- CLIPPER™ is a trademark of the Fairchild Camera and Instrument Corporation.
- WE® is a registered trademark of AT&T.
- PACE Technology is a trademark of the Performance Semiconductor Corporation.

Any use of trademarks in this document which have not been specifically recognized is purely unintentional.

5.0 FTP INPUT/OUTPUT

The FTP Input/Output (I/O) network will consist of three 1553 buses and three custom I/O buses. The custom I/O buses will be connected one to each channel of the FTP and will interface the FTP to the I/O devices that are not 1553 compatible. Each custom I/O bus will be connected to two of the 1553 buses to provide a redundant cross-strapped network. The dual cross-strapped architecture provides tolerance of avionics failures (external to the FTP) without full triple replication of the hardware. A block diagram of the FTP and its I/O buses to the rest of the avionics is shown in Figure 5-1.

5.1 CUSTOM I/O BUS

Each channel of the FTP will have a custom I/O Bus (IOB) which will interface the FTP to the avionics that are not 1553 compatible. These avionics include the skids, the air data, the Ground Support Electronics (GSE), and the Radar Altimeter. There will be some electronics to interface each I/O device it to the IOB.

5.1.1 AIR DATA

There will be fourteen orifices in the nose cap of the ERV each containing three pressure transducers to collect air data. This provides a triply redundant interface for the air data. Each transducer will send three wires back to the FTP (126 total) where the signals will be converted to digital levels before being transmitted on the IOB.

5.1.2 RADAR ALTIMETER

The Radar Altimeter will be located in the avionics bay and will have an antenna attached to the aft skid. It will have a dual redundant link to the FTP through a serial digital interface. The Radar Altimeter will give a 32-bit input to the FTP at each reading.

5.1.3 GROUND SUPPORT ELECTRONICS (GSE)

There will be two GSEs each with a single 1553 interface to the FTP. GSE #1 will be connected to Channel B and GSE #2 will be connected to Channel C as shown in Figure 5-1.

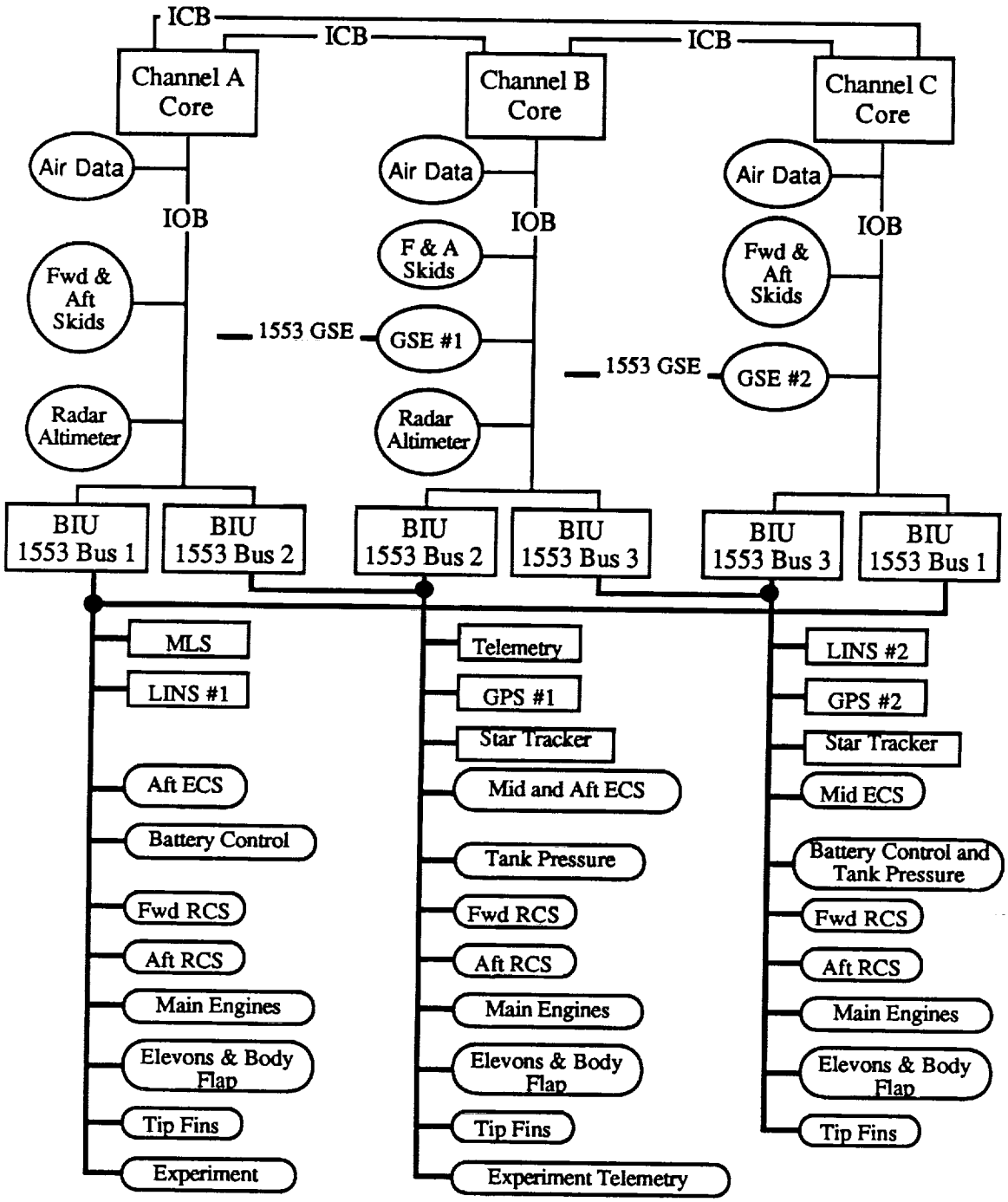

Figure 5-1. ERV Avionics Block Diagram

5.1.4 SKIDS

Both the forward and aft skids will be linked to all channels of the FTP through the IOB. The criticality of the skids requires this triply redundant interface. The interface itself

will be a simple A/D converter for the sensors and a D/A converter with forced voting for the actuators.

5.2 MIL-STD-1553 BUS

The MIL-STD-1553 bus (Military Standard Digital Time Division Command-Response Multiplex Data Bus) was designed to efficiently integrate avionics subsystems. The current version, the MIL-STD-1553B, is implemented widely in military applications and allows flexible, lightweight, and easily tested and modified integration of electronic subsystems.

The 1553B is a serial data bus with dual redundancy to eliminate single point failures. The physical bus is a shielded twisted pair cable. Traffic on the bus travels in one direction at a time over one of the redundant cables. All subsystems have access to all transmissions and are connected to the data bus via terminals. The terminals are transformer-coupled to the bus for signal common mode rejection and are isolated by resistors for fault protection. As many as 31 terminals can be attached to a 1553B data bus, and each terminal can service up to 30 subsystems.

There are three types of interfaces (terminals) on each bus: a bus controller, a bus monitor, and a remote terminal unit. The bus controller initiates bus message transfers on the data bus. Bus monitors receive bus traffic and extract selected information to be used at a later time. A terminal not operating as a bus controller or bus monitor is a remote terminal unit. Typically, bus control is centralized. There is, however, a mode called dynamic bus control which allows distributed control of the bus by passing it from one terminal to another.

The 1553B network functions in a command/response sequence. Bus transmissions are serial time division multiplex messages in pulse code modulation form. They are coded in Manchester II biphase level form which is self-clocking with a zerocrossing during each bit time. Because it is not level sensitive, Manchester coding is ideal for the transformer coupling requirement of 1553B.

1553B bus transmissions are word-oriented. Each word consists of a 3-bit "sync" (which determines the format of the word), 16 bits of data, and a parity bit to detect errors. There are three types of word formats: command words, data words, and status words. Command words come before a message and describe the format of the subsequent

message. They can only be sent by the bus controller. Data words are then sent with the most significant bit first (immediately following the 3-bit "sync"). Status words are transmitted during each message except during a broadcast message. They indicate the remote terminal status and any errors detected in the message.

Each type of terminal provides an interface between a subsystem and the serial data bus. It performs four basic functions: 1) It transmits and receives the analog signals to and from the data bus, 2) it encodes/decodes the Manchester coded signal, 3) it has protocol circuits to manage the data bus transfers, and 4) it contains the subsystem interface.

Because the MIL-STD-1553B is a mature standard that has been utilized on various military programs, numerous suppliers have introduced components that can be used as terminals. Power requirements and circuit complexities have created a need for multiple chip 1553B terminals. However, these designs are often packaged in a single hybrid circuit to meet the small size requirements of airborne military equipment.

Figure 5-1 shows three types of interfaces to the 1553 buses: The Bus Interface Units, "off the shelf" 1553 interfaces (drawn in boxes), and custom 1553 interfaces (drawn in ovals). The "off the shelf" interfaces refer to avionics in which the 1553 bus connects directly without any outside interfacing. These avionics include the Microwave Landing System (MLS), the Laser Inertial Navigational System (LINS), the Global Positional System (GPS), the star tracker, and the telemetry. The other types of interfaces are described in the following sections.

5.2.1 BUS INTERFACE UNIT (BIU)

The BIUs will contain the 1553 bus controller chips and will be the only masters of the 1553 buses. Each channel of the FTP is connected to two of the 1553 buses and, under normal operation, each channel will have control over one them. If, however, one of the channels fails, the control of its 1553 bus will be passed to the other channel connected to it.

5.2.2 CUSTOM 1553 INTERFACES

Although the interfaces differ in their sizes and speeds, the basic custom 1553 interface to be used in the ERV is shown in Figure 5-2.

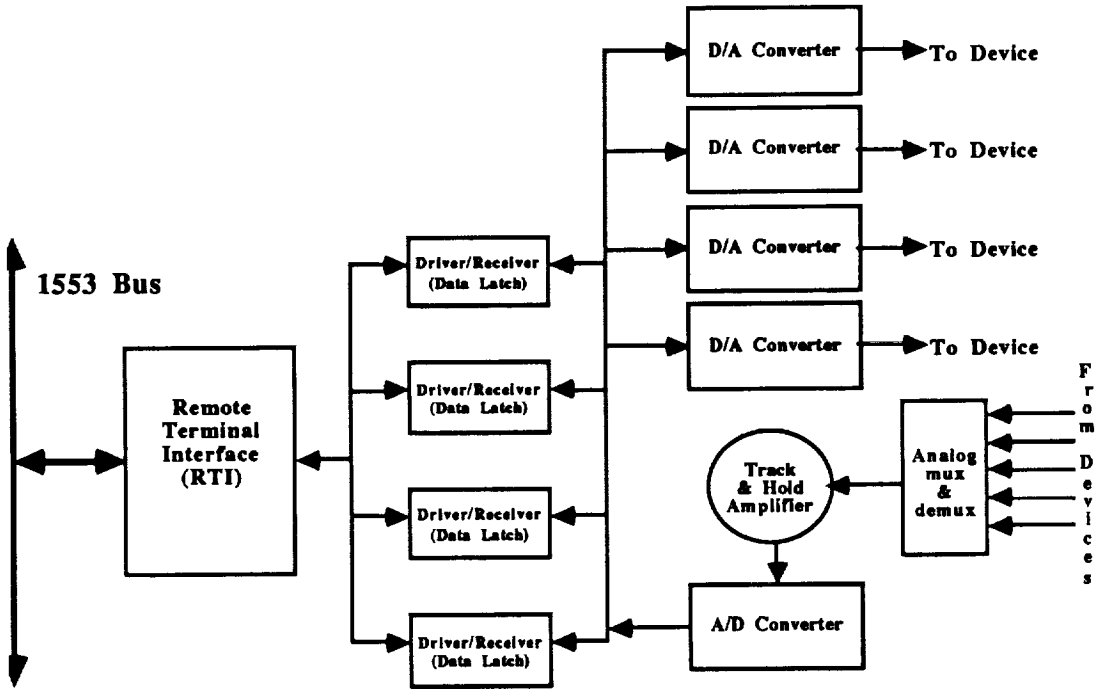

Figure 5-2. 1553 Interface

5.2.2.1 AERODYNAMIC CONTROL SURFACE

The aerodynamic control surfaces on the ERV consist of two tip fins, two elevons, and a body flap. There are seven controllers/sensors connected to these five surfaces (one additional for each tip fin). The interface to each surface will be triply redundant with three sensors and one or more controllers. Signals from all three channels will be force voted and then sent to the controller while each sensor will be read by one channel of the FTP.

5.2.2.2 MAIN ENGINES

There are three main engines and each will have a triplex interface to the FTP. Each engine will have eight actuators: two safety regulators, two throttle valves, two solenoids, a yaw actuator, and a pitch actuator. The sensors in the main engines are not yet defined.

5.2.2.3 REACTION CONTROL SYSTEM (RCS) JETS

The RCS jets are divided into forward and aft modules. The forward jets will have a duplex interface to the FTP, and the aft jets will have a triplex interface. Each jet has two transducers and a thermocouple.

5.3 I/O SUMMARY

Other I/O devices not mentioned in the previous sections, such as the experiment and the experiment telemetry, have not yet been defined to the point where an I/O interface can be designed.

The criticality of each I/O device is discussed in Section 7.0 (Reliability Analysis). Also presented there is the rationale for the redundancy of their interfaces to the FTP.

To summarize the I/O architecture, a schematic of the central computer with its attached distributed I/O is depicted in Figure 5-3. The schematic illustrates the layout of the electronics in the vehicle. For context, Figure 5-4 shows the various bays of the vehicle

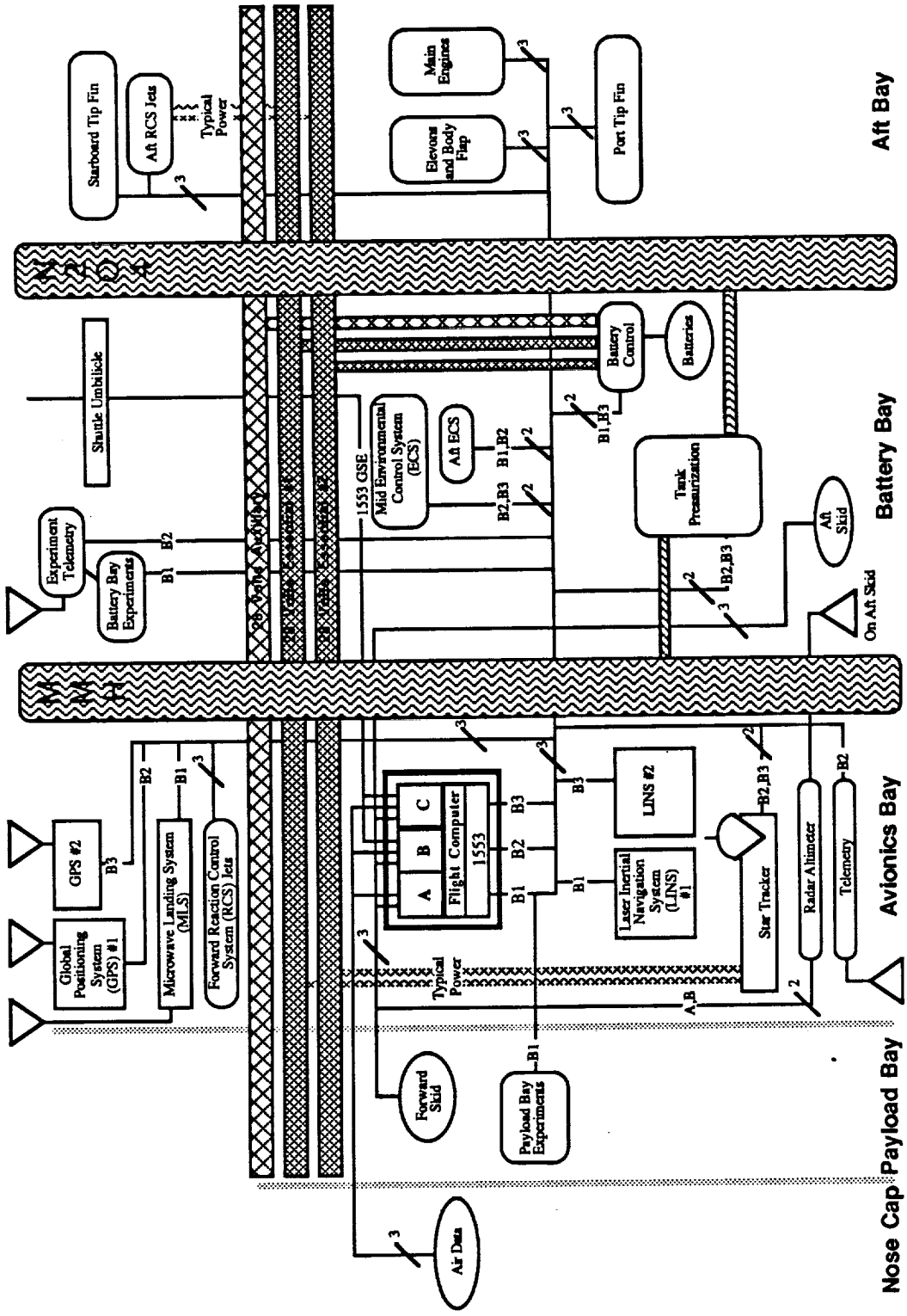

Figure 5-3. Avionics Layout in the ERV

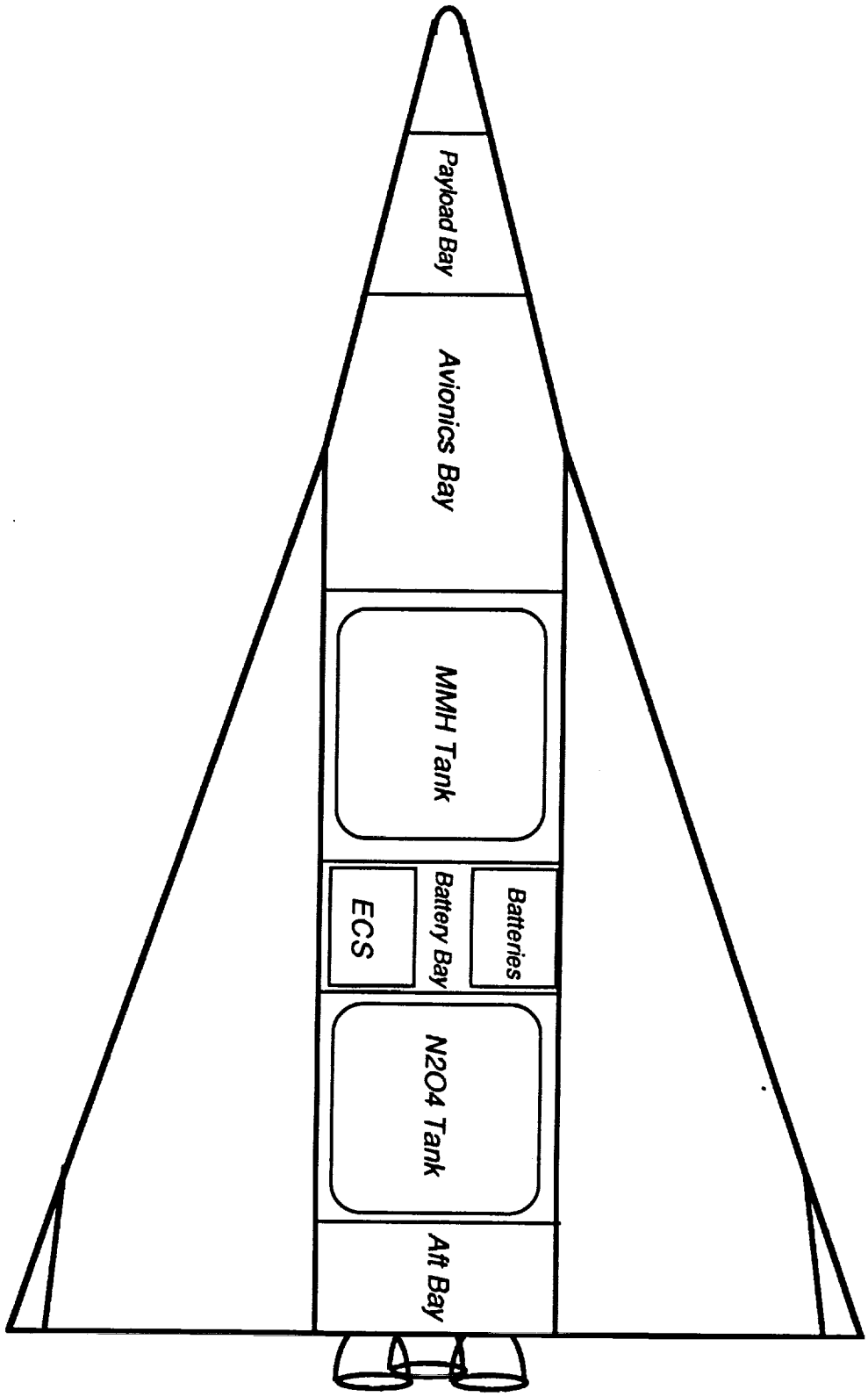

Figure 5-4. The Entry Research Vehicle

6.0 FTP PACKAGING CONCEPT

The conceptual packaging design and sizing for the ERV FTP is based on available circuit design information, plus several estimates and assumptions regarding environments and physical constraints. This information is included here so that the assumptions are obvious for critical review. Two cases were examined: one, for electronics satisfying the basic requirements; and two, for electronics including a 50% reserve.

Much of the physical environment assumed would be transmitted from the Space Shuttle payload bay through the ERV to its avionics bay. The maximum ambient temperature in the ERV avionics bay has been estimated at 60° C. Although no levels have been estimated, it is assumed that there would be significant transmitted vibration input, as well as mechanical shock during flight boost phases. It is assumed that the avionics bay will be reasonably dry and unpressurized.

The required physical characteristics of the package (form factor, mounting interface, I/O connector location, etc.) are largely unspecified at this time. Volume, weight, and power dissipation limits given were 2.5 cubic feet, 150 pounds, and 650 watts, respectively. This study indicates that these numbers can be significantly reduced.

It is assumed that cooling for the FTP will be available in the form of a water-cooled coldplate. Coldplate temperature is unspecified at this time, but 60° F has been assumed because lower temperatures might cause sweating. Consideration will be given to a resilient "coldplate" design to minimize structure and interface problems associated with thermal stresses. This cooling approach is described in Section 6.1 and Appendix A.

A standard military packaging approach has been taken with respect to the electronics. Assumptions in this area include:

1. Modules and backplane wiring will be multi-layer printed circuit.
2. Leaded array and DIP packages will be used for semiconductor components.
3. Modules will require both heat sinking and stiffening.
4. Module guides will employ a wedge lock or cam lock device to provide good heat transfer and positive locking in position.
5. The three electronics channels will be physically separated except for interconnection of the shared hardware modules through backplane

connectors.

6. Each channel will have its own modular power conversion unit and I/O connectors to both the primary and auxiliary ERV power supplies.
7. Separate I/O connectors will be provided for each channel for each 1553 bus (including test) and for Air Data, Skid, and Radar Altimeter links.
8. The I/O connectors will be located on the front of the box.

A summary of the available circuit design information pertinent to the packaging design is included in Section 6.2. The following electronics and power module sizing is based on this information and design experience on similar packages.

For packaging on 6" x 9" printed circuit cards with approximately 45 square inches of usable component area (excluding connectors), the board requirements per channel are as follows:

Function	**Case 1**	**Case 2 (50% Reserve)**
CPUs	4 boards (2 per CPU)	6 (2 per CPU)
Shared Hardware	2	3
I/O	2	3
Total Boards/Channel	8	12
Total Power/Channel	65W Max	98W Max

The power conversion modules are estimated to be about 60% efficient and require a volume equivalent to 1 cubic inch per watt supplied. The resulting volume and power dissipation estimates are:

	Case 1	**Case 2 (50% Reserve)**
Volume	65 in^3	98 in^3
Power Dissipation	43W	65W

An FTP packaging concept based on electronics sizing and partitioning, I/O, and possible thermal/structural requirements is illustrated in Figures 6-1 and 6-2. The power modules are shown bolted to the rear wall of the box to maximize the heat sinking surface contacted. Heavy walls are shown around all the channel partitions to both maximize heat flow from all the electronics modules and provide a sturdy enclosure for all three channels. The heavy walls can also be easily adapted to the resilient coldplate design. Material can be removed selectively to minimize weight while retaining optimal thermal/structural

characteristics.

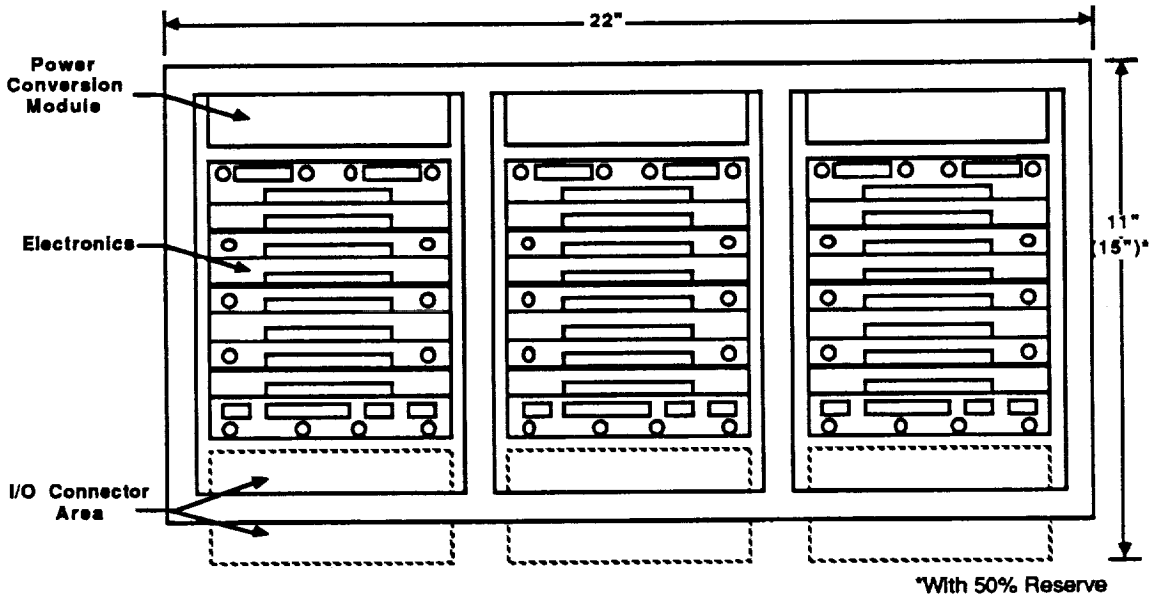

Figure 6-1. ERV FTP Packaging Concept (Top View)

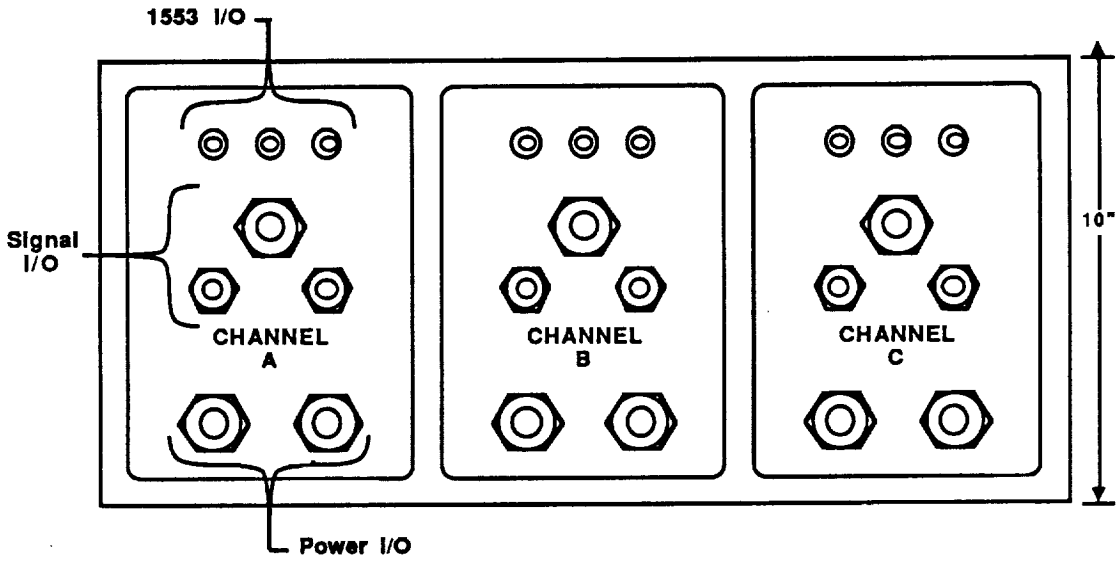

Figure 6-2. ERV FTP Packaging Concept (Front View)

Each channel backplane is shown with connectors to provide interconnection with the channel power module and I/O connectors, and also with the other channel backplanes. An alternate design possibility is an integral multi-layer PC board and flexible circuit for each backplane. The flexible circuit portion could be designed to interface directly with the intrachannel power module and I/O connectors.

The FTP package can be EMI, moisture, and/or pressure sealed as required.

The estimated volume, weight, and power consumption of the conceptual designs for each case is as follows:

	Case 1	Case 2	Limit
Volume, ft^3	1.4	1.9	2.5
Weight, lb	90	120	150
Power, w	325	490	650

6.1 RESILIENT COLDPLATE DESCRIPTION

A lightweight, compact, efficient, and reliable thermal interface scheme provides improved heat transfer from electronics equipment to a heat sink. Conductive heat transfer between chassis and heat sink has usually been across simple planar interfaces. Temperature differences across these interfaces induce thermal stresses which cause warpage and changes in the interface contact, particularly when the interface is over an extended area. A scheme using tapered pins or wedges in mating holes or slots as a thermal interface alleviates these problems.

For example, electronic equipments have been bolted to planar cold rails for cooling. Thermally induced warpage has opened equipment cover seals and changed the thermal interface resistance. Improvements have usually been brought about by the addition of material and fasteners. A system using several tapered pins, mounted on an appropriately bent coolant tube, which mate with tapered holes in the electronic equipment chassis provides a solution to this problem. Stress relief bends in the coolant tube easily remove the thermal stress problem. The tapered pin and hole provide greater interface pressure than would be achieved using the same mating force on the same area in a planar mating situation. This greater contact pressure, in turn, increases the coefficient of heat transfer across the interface. This cooling scheme is illustrated in Appendix A, NASA Tech Brief 71-10088, April, 1971.

6.2 ERV FTP ELECTRONICS INFORMATION

The ERV FTP will consist of three channels. Each channel contains four functions: two CPUs, one Shared Hardware and one I/O Interface. Each channel will also have a

power module converter. The following figures list an approximate device count for the FTP by functional modules.

Device	Quantity	Package	Size inches	Power Dissipation (per device)
MC68020	1	PGA	1.3 x 1.3 (113 pin)	1 Watt, 5 V
MC68881	1	PGA	1.0 x 1.0 (68 pin)	1 Watt, 5 V
SRAM 64K x 4	4	SIP	3.0 x 0.2 (x 0.6 high)	0.5 W, 5 V
ROM 128 K x 8	16	DIP	1.6 x 0.6	0.15 W, 5 V
Gate Array	1	PGA	1.7 x 1.7 (181 pin)	1 W, 5 V
7400 TTL	25	DIP	0.3 x 1.0 (20 pin)	0.1 W, 5 V
PAL	5	DIP	0.3 x 1.2 (24 pin)	0.2 W, 5V
Drivers & Receivers	6	DIP	0.3 x 1.0 (20 pin)	0.1 W, 5V

Figure 6-3. Device Count for CPUs

Device	Quantity	Package	Size inches	Power Dissipation (per device)
Gate Array	1	PGA	1.7 x 1.7 (181 pin)	1 W, 5 V
7400 TTL	65	DIP	0.3 x 1.0 (20 pin)	0.1 W, 5 V
PAL	13	DIP	0.3 x 1.2 (24 pin)	0.2 W, 5V
Drivers & Receivers	38	DIP	0.3 x 1.0 (20 pin)	0.1 W, 5V

Figure 6-4. Device Count for Shared Hardware

Device	Quantity	Package	Size inches	Power Dissipation (per device)
Drivers & Receiver	28	DIP	0.3 x 1.0 (20 pin)	0.1 W @ 5 V
A/D Convertor	3	DIP, Hybrid	1.2 x 2.0 (32 pin)	0.75 W @ +15 V 1.2 W @ -15 V 1.1 W @ +5 V
D/A Convertor	1	DIP, Hybrid	1.2 x 1.8 (32 pin)	0.75 W @ +15 V 0.53 W @ -15 V 0.2 W @ + 5 V
Track &Hold Amplifier	3	DIP, Hybrid	0.8 x 1.3 (24 pin)	0.45 W @ +15 V 0.45 W @ -15 V 0.05 W @ +5 V
PAL	20	DIP	0.3 x 1.3 (24 pin)	0.2 W @ 5 V
Total			**50 in^2**	**22.9 W**

Figure 6-5. Device Count for Channel A I/O

Device	Quantity	Package	Size inches	Power Dissipation (per device)
Drivers & Receiver	42	DIP	0.3 x 1.0 (20 pin)	0.1 W @ 5 V
A/D Convertor	3	DIP, Hybrid	1.2 x 2.0 (32 pin)	0.75 W @ +15 V 1.2 W @ -15 V 1.1 W @ +5 V
D/A Convertor	1	DIP, Hybrid	1.2 x 1.8 (32 pin)	0.75 W @ +15 V 0.53 W @ -15 V 0.2 W @ + 5 V
Track &Hold Amplifier	3	DIP, Hybrid	0.8 x 1.3 (24 pin)	0.45 W @ +15 V 0.45 W @ -15 V 0.05 W @ +5 V
RAM Dual Ported	2	DIP	0.6 x 2.5 (48 pin)	0.4 W @ 5 V
PAL	25	DIP	0.3 x 1.3 (24 pin)	0.2 W @ 5 V
7400 TTL	2	DIP	0.3 x 1.0 (20 pin)	0.1 W, 5 V
Total			**67 in^2**	**26.3 W**

Figure 6-6. Device Count for Channel B I/O

Device	Quantity	Package	Size inches	Power Dissipation (per device)
Drivers & Receiver	42	DIP	0.3 x 1.0 (20 pin)	0.1 W @ 5 V
A/D Convertor	3	DIP, Hybrid	1.2 x 2.0 (32 pin)	0.75 W @ +15 V 1.2 W @ -15 V 1.1 W @ +5 V
D/A Convertor	2	DIP, Hybrid	1.2 x 1.8 (32 pin)	0.75 W @ +15 V 0.53 W @ -15 V 0.2 W @ + 5 V
Track &Hold Amplifier	3	DIP, Hybrid	0.8 x 1.3 (24 pin)	0.45 W @ +15 V 0.45 W @ -15 V 0.05 W @ +5 V
RAM Dual Ported	2	DIP	0.6 x 2.5 (48 pin)	0.4 W @ 5 V
PAL	25	DIP	0.3 x 1.3 (24 pin)	0.2 W @ 5 V
7400 TTL	2	DIP	0.3 x 1.0 (20 pin)	0.1 W, 5 V
Total			**71 in^2**	**27.8 W**

Figure 6-7. *Device Count for Channel C I/O*

7.0 AVIONICS RELIABILITY ANALYSIS

This section presents a reliability study of the ERV avionics. This document was originally released (as ERV-87-05) at the Architecture Design Midterm Review in January 1987 and again (as ERV-87-12) at the Final Review in July 1987. It has been updated to reflect feedback from those meetings and additional analysis that has been suggested.

Since the ERV avionics is still in the definition phase, an approach which contributes to the evolution of the vehicle design was undertaken to study the flight electronics' reliability. Markov models were employed as the means of obtaining the analytical failure probabilities. These figures were complemented with qualitative listings of electronic requirements and availability. The method proved successful. It resulted in the redesign of some areas of the hardware redundancy. The approach is described in detail in Section 7.1.

The criticalness of each of the ERV input/output (I/O) devices is examined in Section 7.2. Information for this section was obtained from the Task 1 final report [Kriegsman et al] and from consultations with B. Kriegsman of CSDL. That section examines the I/O devices in the context of vehicle survival and mission success.

Several iterations of an ERV strawman architecture have been developed. The latest of these was used to determine the availability of the various I/O devices in the presence of flight computer failures. Those listings are presented in Section 7.3.

The flight computer (including its attached I/O) for ERV was assumed to have failure characteristics that resemble Markov processes. Several Markov models were developed to exploit this trait. They are described in Section 7.5. The failure rates that were used in those models are based upon the latest ERV schematic. Section 7.4 presents the figures and explains how they were derived. Finally, Section 7.6 combines the models and the failure rates and discusses the modelling results.

7.1 METHODOLOGY FOR THE RELIABILITY STUDY

The following is a description of the methodology that was employed for the reliability analyses of the ERV architecture design. Preliminary analysis on the subject and discussions with NASA/Langley led to the decision to use this approach for the baseline study to be completed under this contract.

The goal of this method was threefold: to conservatively limit the reliability studies that were performed under Task 7; to study the reliability of the Fault-Tolerant Processor (FTP) and its I/O cross-strapping, not necessarily the reliability of the I/O itself; and to provide valuable feedback to the design process about hardware redundancy.

The method is a four point iterative analysis based upon reliability of the core FTP and its electronic interfaces. The four points (FTP model, I/O criticalness, I/O availability, assuring success) are discussed below.

7.1.1 FTP MARKOV MODEL

The core FTP with its interfacing electronics will be modelled as a Markov process with the eight possible final states or modes:

	Channels On-Line			
Mode	A	B	C	Configuration
7	1	1	1	Triplex
6	1	1	0	A & B duplex
5	1	0	1	A & C duplex
4	1	0	0	A simplex
3	0	1	1	B & C duplex
2	0	1	0	B simplex
1	0	0	1	C simplex
0	0	0	0	None operational

Triplex operation (mode 7) is a fully operational FTP. Mission/vehicle[3] success is assured (with respect to the flight computer).

Duplex operation (modes 6, 5, and 3) is a degraded but operational FTP. Vehicle success is nearly assured. (This will be determined from the qualitative analysis described in Section 7.1.3 and Section 7.1.4.)

Simplex operation (modes 4, 2, and 1) is a doubly-degraded FTP. It is operational

[3]Since the vehicle is the mission, the terms are interchangeable. Vehicle success will be used predominantly in this report.

only because of good coverage by the FDIR (Fault Detection, Isolation, and Recovery) routines. Graceful degradation from duplex is not guaranteed. Also, mission/vehicle success is not necessarily assured if the FTP is operating in simplex , since interfaces to some critical I/O may not be triplicated. (Again, this will be seen in Section 6.1.3 and Section 7.1.4.)

Mode 0 would be a catastrophic failure (two failures before recovery, or three failures, etc.). It would cause both mission and vehicle loss since the flight computer would not be operational.

These 8 modes refine previous Markov models which gave death states of "Fail-Catastrophic" and "Fail-Safe". Mode 0 is now the catastrophic failure state while the other modes increase the granularity of the paths to that state. Note that there can be no Fail-Safe operation (graceful shutdown of the computer) since there is no secondary pilot (automatic or human) to take control of flight.

A model has been developed (Section 7.5) to implement the scenario described above. State transition probabilities were obtained (Section 7.4) by roughly sizing the electronics in each channel and determining λ_{pa}, λ_{pb}, λ_{pc}, and λ_i (each processor and interstage). λ_{pa} may or may not equal λ_{pb}, etc., because of the differences in electronics in each channel. That is, the interface electronics to the I/O modules would be included in λ_p.

We make the tacit assumption that $\lambda \approx 0$ for the I/O devices themselves. An order of magnitude estimation of the required failure rate for each I/O device has been provided, however. (For example, the star tracker must have $\lambda \leq 10^{-7}$ failures/hour.) This is termed P_{CDL} and is described in Section 7.5.3.

7.1.2 I/O CRITICALNESS

Based upon expertise in the field of flight guidance, navigation and control, lists of the minimum complement of I/O required for mission and vehicle success have been compiled. There may be several of these lists, since, for example, a GPS and radar altimeter combination could cover for the MLS.

Assume that navaids V, W, X, Y, & Z are available. Lists of those navaids which would be minimally required for vehicle success may be:

List 1		List 2		List 3		... etc.
V		V		V		
W	*or*	X	*or*	W	*or*	
Y		Y		Z		
		Z				

The above would mean that the ERV vehicle could not survive without navaid V, but navaid W could be lost if navaids X and Z were present, etc. These lists are accompanied by qualitative description of each I/O device's function in the system.

7.1.3 AVAILABLE I/O

For each of the 8 possible operational modes of the FTP, a list of accessible I/O can be determined from the latest avionics schematic. For example:

Mode	Configuration	Available I/O
7	Triplex, ABC	3V, 3W, 2X, 2Y, Z
6	Duplex, AB	3V, 2W, X, 2Y, Z
5	Duplex, AC	etc.
4	Simplex, A	V, W, Y
3	Duplex, BC	etc.
2	Simplex, B	V, W, Z
1	Simplex, C	V, W
0	failed	none

From this list and the ones compiled in Section 7.1.2, we can determine that only mode 1 is a vehicle loss mode (only V and W available). Then we can say,

$$P\{\text{vehicle loss}\} = P\{0\} + P\{1\} + P\{\text{other critical event}\}$$

7.1.4 ASSURING SUCCESS

If from the calculations in Section 7.1.3 it is determined that P{vehicle loss} > 10^{-7}, then we need to make mode 1 a non-vehicle loss mode. This is because we cannot (easily) decrease P{1} or P{0}. The only means, then, to decrease P{vehicle loss} is to eliminate P{1} from affecting it.

Qualitatively scanning the results, we see that we can make mode 1 a non-vehicle loss mode by assuring that navaid Y or Z is available to channel C (triplicate navaid Y?, duplicate Z?).

Thus qualitative iterations of Section 7.1.3 and Section 7.1.4 will assure that the FTP and its I/O architecture will meet the desired reliability numbers (10^{-7} vehicle loss, 10^{-6} mission loss).

7.1.5 SUMMARY OF METHODOLOGY

This approach is a qualitative one based upon the analytical numbers obtained from the FTP model. Using the basic lists described here, it can be assured that the avionics meet the required reliability. Also from this procedure, it can be determined if too much I/O redundancy has been designed into the system (if none of the modes in Section 7.1.3 provide a minimum complement, i.e., they all have more than enough I/O).

7.2 Criticalness of ERV I/O

The ERV I/O architecture implements a mixed redundancy of the attached avionics. The replication of the various devices will be a function of their individual reliability and criticalness - that is, how crucial their operation is for mission and vehicle success.

This section discusses the criticalness of all the ERV I/O for a typical mission. The 8 hour flight is described by these attributes:

1. deployment from Shuttle to in-atmosphere plane change
2. return to low-Earth orbit for two revolutions
3. entry (with blackout)
4. landing in desert area

The mission is defined in more detail in the ERV Task 1 Report [Kreigsman et al], pages. 4-99 through 4-100.

The I/O items are divided into three categories: Flight Control, Navaids, and non-GN&C. This is a convenient categorization of the devices and does not imply mutual exclusion. That is, the navaids are certainly an integral part of the flight control system and their separate grouping does not connote independence from it.

The function of each I/O device is briefly described. Following this are the consequences of device loss. Also discussed with these consequences is the coverage that any other I/O device could provide in the event of failure. For example, a radar altimeter and GPS combination can cover for a failed MLS.

7.2.1 FLIGHT CONTROL I/O

The I/O items discussed below perform flight control on the ERV. The majority of these devices will have triply replicated hardware interfaces. The initial assumption (for this study) is that no other controller can be substituted for any of the devices. In addition, because the devices are so critical in flight control, loss of any of them will result in vehicle loss. *That is, the failure of any of the following devices will result in system loss.*

These assumptions may not necessarily be valid, but are used because of a lack of detailed information (at this time) about the ERV flight controlling devices. A more rigorous analysis would prepare true minimum complement lists for the control items. (The above assumption sets the minimum list to all devices.)

Although the Laser Inertial Navigation System (LINS) plays a crucial role in flight control, it is not discussed in this section. Rather, it is presented with navaids I/O since the required accuracy for flight control is far less than that for navigation.

The flight control I/O devices are as follows:

1. SKIDS, FORWARD & AFT

   The landing skids are located at the forward and aft sections of the plane. They are highly critical at the time of landing only. Triplex electronics will control

their action.[4]

2. REACTION CONTROL SYSTEM (RCS) JETS, FORWARD & AFT

Nineteen RCS thrusters control the attitude maneuvers of the vehicle. While some RCS jet failures can be tolerated by the system without vehicle or mission loss, all interface electronics to them will be triply replicated to assure aircraft survival.

3. MAIN ENGINES

Three engines will provide the main thrust for the vehicle. Although an engine failure can be tolerated, it was assumed for this study that the engines are of utmost importance for vehicle success. Electronic interfaces will be triply redundant.

4. TIP FINS, STARBOARD AND PORT

The tip fins provide rudder and braking control of the vehicle. Interface to each of them will be triply redundant.

5. ELEVONS, STARBOARD AND PORT

The elevons control the pitch and roll of the vehicle. Triplex electronic interface to them will be provided.

6. BODY FLAP

The body flap modifies the flight characteristics at various speeds. Its primary function is to control the longitudinal trim. Interface to it will be triply redundant.

7. AIR DATA (SEADS).

Air data measurements will be taken through 14 triply redundant pressure transducers located in orifices in the vehicle nose cap. Airspeed, dynamic pressure, and wind relative attitudes are provided to the flight control system in the FTP. Some data will also be used for experimentation input.

4Dual-redundancy was the initial skid design. This study has resulted in the decision to triplicate skid electronics.

As with the RCS jets, some transducer failures can be tolerated by the system without vehicle or mission loss. For this study, however, it will be assumed that the loss of the devices is vehicle critical and no coverage is available from other I/O.

7.2.2 NAVAID I/O

Five navaids in the I/O network provide vital information to the GN&C algorithms. A thorough discussion of these aids was prepared by B. Kriegsman (as ERV-87-01). The following is a summary of that paper.

1. LASER INERTIAL NAVIGATION SYSTEM (LINS)

Since it is not affected by atmospheric blackout, the LINS is the sole navaid which can provide position and velocity information throughout the entire ERV flight. It is a critical component of the navigation system and will be dual redundant: two LINS units will be provided in the vehicle.

The LINS alignment, position, and velocity estimates must continually be updated, however, in order to compensate for various navigation errors that occur throughout the mission. Such updates must be implemented immediately before deployment and before major aerodynamic and propulsive maneuvers. Other navaids must provide the LINS alignment data, either directly (star tracker) or indirectly using error correlations (GPS). The GPS must provide position and velocity updates.

Consequences of and Coverage for Failure: No other navaid can cover for a failed LINS. Loss of (both) LINS would result in vehicle loss.

2. STELLAR-UPDATE DEVICE; STAR TRACKER OR SCANNER

A star tracker is an excellent device for LINS realignment. Before deployment from the Shuttle, an on-board tracker will align the LINS. The tracker will also be used to update the alignment before major aerodynamic and propulsive burns. Because of high expense and other navaid coverage, only one tracker will be provided aboard the vehicle, although two interfaces to it will be present.

Consequences of and Coverage for Failure: The sole purpose of the star tracker

is LINS Inertial Measuring Unit (IMU) alignment. While not as accurate, the Shuttle tracker can be used to provide pre-deployment data to the LINS, should the ERV tracker fail before flight. (If this occurs, however, aborting the mission should be considered.) Once in flight, the GPS can serve to align the IMU but only by correlation between alignment error and position and velocity errors. This error cross-correlation method requires an IMU-sensed change in velocity, i.e., an aerodynamic or propulsive maneuver.

3. GLOBAL POSITIONING SYSTEM (GPS) RECEIVER

The GPS provides the IMU with position and velocity data before major aerodynamic and propulsive maneuvers. It suffers from radio blackout during entry, but its blackout period is less than other radio-navaid systems (e.g., TACAN, VOR/DME) since it operates at higher frequencies. Once through blackout, it provides the means of updating the IMU-derived position and velocity estimates for landing. As such, the GPS has the sole responsibility of providing navigation data to the flight control system to enable the vehicle to be within range of the MLS. Thus, it is an extremely valuable navaid for ERV. Two GPS receiver units will be provided on-board.

Consequences of and Coverage for Failure: Failure of the GPS will mean system loss, since after entry it is the sole means of updating the horizontal plane position and velocity estimates.

4. MICROWAVE LANDING SYSTEM (MLS)

The MLS considered for ERV will be the same as provided in the Shuttle: a microwave scanning beam landing system. The MLS provides excellent azimuth, elevation, and range data. Additionally, the MLS information is provided with respect to the landing site, not the local terrain, and position-fix accuracy is increased as the site is approached.

Consequences of and Coverage for Failure: Although an excellent facility for landing, the MLS can probably be replaced by the GPS, LINS, and radar altimeter. The GPS/LINS will provide accurate landing information in the horizontal plane. Altitude data assistance from the radar altimeter is required for the vertical channel.

5. RADAR/RADIO ALTIMETER

The radar altimeter is used to measure z position over the local terrain.

Consequences of and Coverage for Failure: Loss of the altimeter is not critical. In conjunction with the GPS/LINS, the altimeter will provide coverage for the MLS. Thus, in effect, loss of the altimeter can be covered by the MLS.

7.2.3 NON-GN&C I/O

Several I/O devices on the vehicle do not provide GN&C data. They are listed in this section.

1. GROUND SUPPORT ELECTRONICS (GSE)

The GSE is used only for pre- and post-flight checkout. Two GSE units will be provided on-board.

Consequences of and Coverage for Failure: Should the GSE fail before flight, aborting the mission should be considered, since knowledge about the condition of the other avionics will be dubious. Once in flight, however, GSE failure will not harm the mission or vehicle.

2. TELEMETRY

The function of telemetry on the vehicle is still in question. "Wake up" and "self-destruct" are two messages under consideration. Definition beyond that is subject to further examination.

Consequences of and Coverage for Failure: Loss of telemetry to the vehicle could result in both vehicle and mission loss, since a "wake-up" message may never be received. However since this is only during the first seconds of the mission and the telemetry will be subject to GSE checkout in the STS bay, only one telemetry unit will be provided. Loss of "self-destruct" information may be a more serious problem, however, and should be studied further. At this point, loss of vehicle telemetry will not be accounted for in the reliability studies since it is a very unlikely occurrence and it is still in a definition phase.

3. EXPERIMENT AND EXPERIMENT TELEMETRY

The experiment aboard the vehicle will vary with each mission. It will have no replicated hardware interface to the flight computer.

Consequences of and Coverage for Failure: Loss of experimentation and its telemetry will not affect the vehicle. Mission failure may result, depending upon the state of the experiment at the time of failure.

4. ENVIRONMENTAL CONTROL SYSTEM (ECS)

The Environmental Control System includes the following subsystems: battery control, tank pressurization, and cooling. All interfaces will be duplex since the devices should be fairly reliable.

Consequences of and Coverage for Failure: Loss of any ECS system will result in the loss of power to the flight computer and/or to the actuators. Engines may also fail due to lack of fuel. In any case, vehicle loss will be the result. ECS subsystems can not be covered by any other device.

7.2.4 MINIMUM COMPLEMENT OF I/O

Based upon the above descriptions, lists of the minimum complement of I/O required for vehicle success can be compiled. The function of these lists is discussed in Section 7.1.2.

7.2.4.1 MINIMUM COMPLEMENT OF I/O FOR VEHICLE SUCCESS

There are four lists of I/O required for vehicle success. The lists all contain the items described in Section 7.2.1 above. The items in Section 7.2.2 have some interchangeability and account for the list differences. Section 7.2.3 items are not all vehicle critical.

Lists C and D assume that upon loss of the star tracker the mission is immediately aborted and the vehicle is returned to Earth. When the tracker is lost, IMU realignment can only be based upon error correlation between position and velocity readings (from the GPS) and alignment readings. This method of error cross-correlation cannot be performed during gravitation-only flight paths: propulsive or aerodynamic maneuvering is required.

Nevertheless, the accuracy of this method is not welldetermined and, it should not be used to complete the mission.

List A	List B	List C	List D
Skids, F & A	Skids, F & A	Skids, F & A	Skids, F & A
RCS Jets, F & A	RCS Jets, F & A	RCS Jets, F & A	RCS Jets, F & A
Main Engines	Main Engines	Main Engines	Main Engines
Tip Fins, S & P	Tip Fins, S & P	Tip Fins, S & P	Tip Fins, S & P
Elevons, S & P	Elevons, S & P	Elevons, S & P	Elevons, S & P
Body Flap	Body Flap	Body Flap	Body Flap
Air Data	Air Data	Air Data	Air Data
LINS	LINS	LINS	LINS
GPS	GPS	GPS	GPS
Star Tracker	Star Tracker	Radar Altimeter	MLS
MLS	Radar Altimeter		
ECS	ECS	ECS	ECS

Figure 7-1. Minimum Complement of I/O for Vehicle Success

7.3 AVAILABILITY OF ERV I/O

Because the ERV architecture employs three MIL-STD-1553B buses cross-strapped to two channels each, the association of an I/O device with a channel is not straightforward. However, the ERV reliability study methodology (as outlined in Section 7.1) requires some identification of an I/O device with a channel. More precisely, the study requires a correlation between channel failure and loss of I/O device.

It is no accident that the I/O architecture avoids a correlation between channel faults and device loss. Indeed, if a rigid link existed, many costly (in weight, power, size, and dollars) I/O devices would need triplex replication. The architecture designed for the ERV allows for less than triplex replication of I/O.

In view of this, the following "availability" lists cannot be construed as final, rigid results. Modelling will need to account for this lack of total correlation between channel fault and device loss. The modelling process may be somewhat iterative.

7.3.1 I/O AVAILABILITY (CHANNEL)

Using latest schematics of the ERV avionics as a guide, the availability lists for the ERV I/O (per channel) are compiled in Figure 7-2. The lists assume that a faulty channel cannot cause a failure in an attached 1553 bus. This assumption is not without foundation:

monitor/interlocks - which shut down a failed channel's outputs - can be developed which will prevent channel failures from propagating to a 1553 bus.

Channel A List	Channel B List	Channel C List
Air Data	Air Data	Air Data
Forward Skids	Forward Skids	Forward Skids
Aft Skids	Aft Skids	Aft Skids
Radar Altimeter	Radar Altimeter	
Microwave Landing System		Microwave Landing System
	Ground Support #1	Ground Support #2
Telemetry	Telemetry	
LINS #1		LINS #1
	LINS #2	LINS #2
GPS #1	GPS #1	
	GPS #2	GPS #2
Star Tracker	Star Tracker	Star Tracker
Mid ECS	Mid ECS	Mid ECS
Aft ECS	Aft ECS	Aft ECS
Battery Control	Battery Control	Battery Control
Tank Pressurization	Tank Pressurization	Tank Pressurization
Forward RCS Jets	Forward RCS Jets	Forward RCS
Aft RCS Jets	Aft RCS Jets	Aft RCS Jets
Main Engines	Main Engines	Main Engines
Elevons	Elevons	Elevons
Tip Fins	Tip Fins	Tip Fins
Experiment		Experiment
Experiment Telemetry	Experiment Telemetry	

Figure 7-2. I/O Availability per Channel

7.3.2 I/O AVAILABILITY (OPERATIONAL MODE)

From Section 7.3.1, availability lists based upon mode of operation can be compiled. (Modes of operation are discussed in Section 7.1.1.) The following lists are not exactly of the type discussed in Section 7.1.3. Here, an item is listed as available if there is at least one connection to it.

Mode	Configuration	Available I/O
7	Triplex, ABC	Air Data; Forward Skids; Aft Skids; Radar Altimeter; MLS; GSE #1; GSE #2; Telemetry; LINS #1; LINS #2; GPS #1; GPS #2; Star Tracker; Mid ECS; Aft ECS; Battery Control; Tank Pressurization; Forward RCS Jets; Aft RCS Jets; Main Engines; Elevons; Tip Fins; Experiment; Experiment Telemetry.

6	Duplex, AB	Air Data; Forward Skids; Aft Skids; Radar Altimeter; MLS; GSE #1; Telemetry; LINS #1; LINS #2; GPS #1; GPS #2; Star Tracker; Mid ECS; Aft ECS; Battery Control; Tank Pressurization; Forward RCS Jets; Aft RCS Jets; Main Engines; Elevons; Tip Fins; Experiment; Experiment Telemetry.
5	Duplex, AC	Air Data; Forward Skids; Aft Skids; Radar Altimeter; MLS; GSE #2; Telemetry; LINS #1; LINS #2; GPS #1; GPS #2; Star Tracker; Mid ECS; Aft ECS; Battery Control; Tank Pressurization; Forward RCS Jets; Aft RCS Jets; Main Engines; Elevons; Tip Fins; Experiment; Experiment Telemetry.
4	Simplex, A	Air Data; Forward Skids; Aft Skids; Radar Altimeter; MLS; Telemetry; LINS #1; GPS #1; Star Tracker; Mid ECS; Aft ECS; Battery Control; Tank Pressurization; Forward RCS Jets; Aft RCS Jets; Main Engines; Elevons; Tip Fins; Experiment; Experiment Telemetry.
3	Duplex, BC	Air Data; Forward Skids; Aft Skids; Radar Altimeter; MLS; GSE #1; GSE #2; Telemetry; LINS #1; LINS #2; GPS #1; GPS #2; Star Tracker; Mid ECS; Aft ECS; Battery Control; Tank Pressurization; Forward RCS Jets; Aft RCS Jets; Main Engines; Elevons; Tip Fins; Experiment; Experiment Telemetry.
2	Simplex, B	Air Data; Forward Skids; Aft Skids; Radar Altimeter; GSE #1; Telemetry; LINS #2; GPS #1; GPS #2; Star Tracker; Mid ECS; Aft ECS; Battery Control; Tank Pressurization; Forward RCS Jets; Aft RCS Jets; Main Engines; Elevons; Tip Fins; Experiment Telemetry.
1	Simplex, C	Air Data; Forward Skids; Aft Skids; MLS; GSE #2; LINS #1; LINS #2; GPS #2; Star Tracker; Mid ECS; Aft ECS; Battery Control; Tank Pressurization; Forward RCS Jets; Aft RCS Jets; Main Engines; Elevons; Tip Fins; Experiment.
0	failed	none

Figure 7-3. Vehicle I/O Availability

7.3.3 SUMMARY OF AVAILABILITY LISTS

The lists given in this report are for use with the reliability studies to be performed on the ERV under Task 7. This study is a preliminary one which stresses the qualitative analyses due to the imprecise definition of the vehicle, its avionics, and its missions. The

lists, then, should not be construed as formal and final studies in themselves. The interdependence of the avionics design and its own reliability forces these lists to be used in an iterative process which will point to an ideal flight electronics system for the vehicle.

7.4 AVIONICS FAILURE RATES

Equations and formulas for calculating failure rates are quite complex. Much must be known about a device before its failure rate can be accurately calculated. Because we are in the early stages of the ERV flight computer design, specific devices within the system are not known. Failure rate estimations were, therefore, based upon MIL-HDBK-217D guidelines, Reliability Prediction of Electronic Equipment, January 1982.

The following three sections assume familiarity with the latest ERV avionics schematics. They have been discussed in the previous sections of this volume.

7.4.1 FAILURE RATE FACTORS

Many factors are involved in determining the failure of a particular device including its complexity, its packaging, and the environment it is used in. The general equation from MIL-HDBK-217D for calculating the failure rate per 10^6 hours, λ_P, of a microelectronic device is:

$$\lambda_P = \pi_Q * [(C_1 * \pi_T * \pi_V) + (C_2 + C_3) * \pi_E] * \pi_L$$

where

π_Q is the quality factor and is determined by the quality of the device.

C_1 and C_2 are device complexity failure rates based upon the transistor count for linear devices, the gate count for logic devices, or the bit count for memory devices.

π_T is the temperature acceleration factor based upon the technology of the part.

π_V is the voltage derating stress factor which is dependent upon the supply voltage of the device.

C_3 is the package complexity failure rate based upon the number of functional pins on the device.

π_E is the application environment factor.

π_L is the device learning factor based upon the age of the technology of the device.

7.4.1.1 DEVICE QUALITY

The quality of a part, as one can see from the equation, has a linear effect on the part's failure rate. Many parts are covered by specifications that have several quality levels. There are three basic military-standard quality levels, S, B, and C, and a commercial quality level D. The MIL-HDBK-217D assigns values to π_Q for each quality level:

Quality Level	π_Q	Description
S	0.5	Class S qualification is the highest a device can have. It means the device has been procured in full accordance with MIL-M-38510, Class S requirements. Class S is most often associated with the term "Space Qualified."
B	1.0	To qualify as a Class B part, a device must be procured in full accordance with MIL-M-38510, Class B requirements.
C	8.0	To qualify as a Class C part, a device must be procured in full accordance with MIL-M-38510, Class C requirements.
D	17.5	A part qualifies as Class D if it is hermetically sealed with no screening beyond the manufacturer's regular quality assurance practices.

In calculating the failure rate of the ERV flight computer it was assumed that Space

Qualified (Class S) parts would be used.

7.4.1.2 DEVICE COMPLEXITY FAILURE RATES

C_1 and C_2 are polynomial functions of the transistor count for linear devices, the gate count for logic devices, and the bit count for memory devices. The functions vary with component size and device technology. The handbook includes look-up tables for these values.

7.4.1.3 TEMPERATURE ACCELERATION FACTOR

The temperature acceleration factor, π_T, is based upon the device's capability to perform under extreme temperatures. It takes into account a number of the device characteristics and the temperature of its operating environment. The equation for π_T is:

$$\pi_T = 0.1 * e^X$$

where:

$$X = -A*[1/(T_J+273) - 1/298]$$

A is a value taken from the MIL-HDBK-217D based on the device's seal and technology.

T_J is the devices worst case junction temperature which is based upon the device's thermal resistance, its worst case power dissipation and the temperature of its operating environment.

The most dominant contributors to the temperature acceleration factor are the device's thermal resistance and its operating environment temperature. Their effect on the overall failure rate of a device can be seen in Figures 7-4 and 7-5. The data in the figures were taken from the failure rate calculations of the gate array used in the VLSI FTP[5]. As can be seen from the graphs, both the thermal resistance of a device and its environment's temperature are significant factors to the devices failure rate. If these two factors are controlled, the reliability of a device can be increased. To lower the thermal resistance of a

[5]As part of an internal research and development effort, CSDL has designed and fabricated a miniaturized version of the Langley AIRLAB FTP.

device, eutectic die attaches should be used.

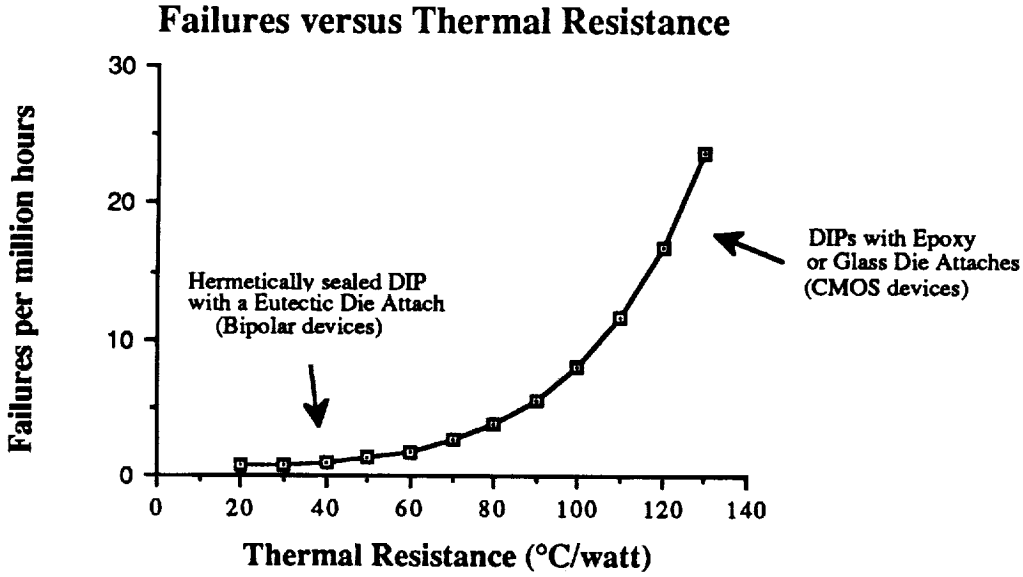

Figure 7-4. Device Failure Rate versus Thermal Resistance

Typical temperatures of a few environments are noted in Figure 7-5. The ERV flight computer will operate in an environment that will not exceed 60° C. For conservative measures, failure rates were calculated using a temperature factor that corresponds with with an ambient temperature of 60° C.

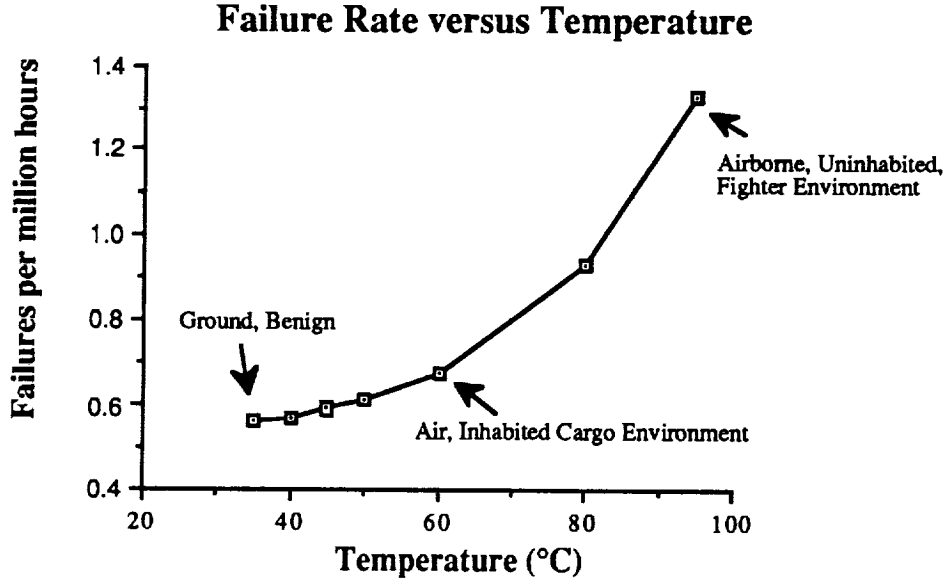

Figure 7-5. Gate Array Failure Rate versus Temperature

7.4.1.4 VOLTAGE DERATING STRESS FACTOR

The voltage derating stress factor is dependent upon the supply voltage of the device. For a supply voltage of 5 volts, $\pi_V = 1.0$ and it has no effect on the failure rate calculation for a particular device. It was assumed in this study that all devices would have supply voltages of 5 volts and π_V was ignored.

7.4.1.5 PACKAGE COMPLEXITY FAILURE RATE

C_3 is a polynomial function of the number of functional pins on a particular device. The functions vary with component size and device technology. The handbook contains look-up tables where values for C_3 can be found.

7.4.1.6 APPLICATION ENVIRONMENT FACTOR

The application environment factor, π_E, includes the effects of environmental stress on a device in the reliability analysis. The MIL-HDBK-217D gives a list of descriptions of typical operating environments for electronic equipment along with their associated values for π_E. These environments range from a immobile, laboratory environment where π_E = 0.38 to a cannon, launch environment where π_E = 220. The first estimate of π_E for the ERV flight computer in ERV-87-05 suggested that the computer would operate in an environment similar to a missile, launch environment. The missile, launch environment includes severe conditions related to missile launch and space vehicle boost into orbit and to vehicle re-entry and landing by parachute. After review, it was determined that this environment description is more severe than the environment in which the ERV flight computer will operate. It is assumed that the avionics in the ERV will be powered down during the shuttle boost into orbit and the ERV will not free fall through the atmosphere and land by parachute as described in the missile, launch environment. The flight computer operating environment will be more like the airborne, uninhabited cargo (A_{UC}) environment as described in MIL-HDBK-217D which includes extreme pressure, vibration, and temperature cycling similar to that endured in equipment bays of long mission transport aircraft. The handbook states that the temperature in the A_{UC} environment ranges up to 90° C. It was assumed, that the environmental control system in the ERV will keep the flight computers ambient temperature below 60° C.

Although when the ERV flight computer is in orbit, the environment will be much

less severe than an A_{UC} environment. Conservative measures were taken once again and π_E (3.0) for the A_{UC} environment was used in all calculations of device failure rates. Figure 7-6 shows a device's failure rate versus some typical environments.

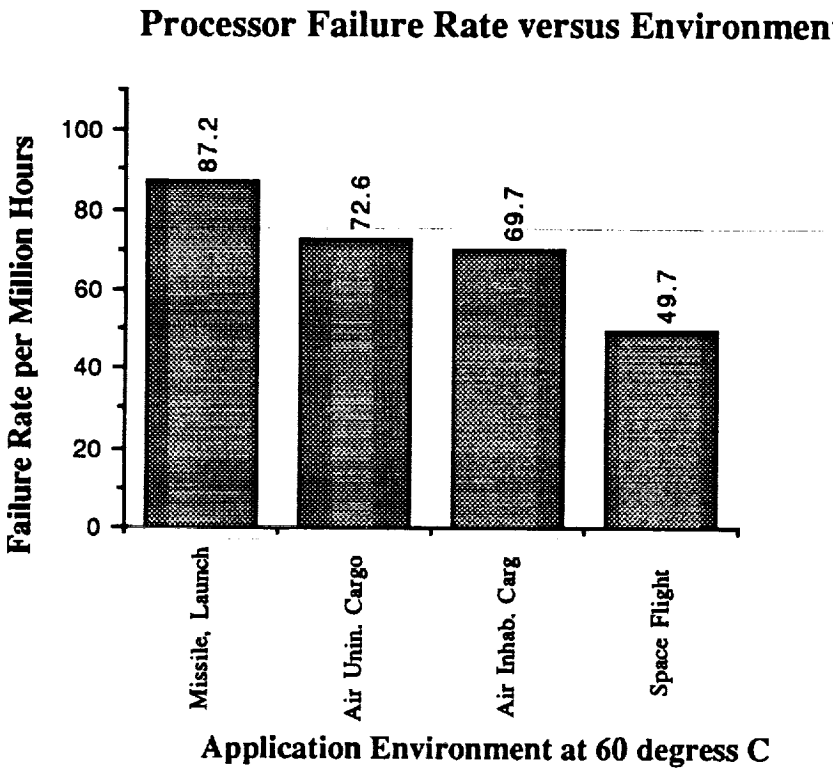

Figure 7-6. Microprocessor Failure Rate versus Environment

7.4.1.7 DEVICE LEARNING FACTOR

The device learning factor π_L depends on how long the device has been in production. For a device that has been in production for at least four months, $\pi_L = 1.0$ and it does not affect the reliability of a device. This was the case assumed in the ERV flight computer failure rate determination.

7.4.2 ERV FTP FAILURE RATE

The hardware failure rate of the ERV flight computer was derived using the Very Large Scale Integration/Very High Speed Integrated Circuit (VLSI/VHSIC) FTP as a model. The VLSI FTP is being developed at CSDL with the objective of continuing the development of the FTP and producing a practical sized, VLSI/VHSIC FTP with sufficient

throughput to handle a wide range of DoD and NASA applications. This makes the VLSI FTP a suitable hardware model of the ERV flight computer.

The VLSI FTP processor and interstage failure rates were calculated by adding the failure rates of the individual devices constituting the processor and interstage which, in turn, were determined from MIL-HDBK-217D standards as discussed in the previous section. Failure rates of the processor and interstage were determined based solely on failures of solid state devices and interconnections. All devices were assumed to be Space Quality and an airborne, uninhabited cargo environment was assumed as suggested by the handbook. Failures of power supplies and backplanes were not considered. Failure rates of the processor and interstage were based upon failure rates of the following devices:

Device	Quantity in Processor	Quantity in Interstage	Failure Rate (per 10^6 hrs)
Microprocessor	2	-	0.32
Floating-point CP	2	-	0.32
64K x 4 static RAM	4	-	6.0
128K x 8 Bipolar ROM	16	-	2.6
VLSI Gate Array	3	-	0.31
7400 TTL (worst case)	80	35	0.02
PALs	10	4	0.15
Drivers/Receivers	35	-	0.03
Connectors	4	1	0.16

Total Failure Rate: **Processor Failure Rate: 72.6 failures/10^6 hours.**
Interstage Failure Rate: 1.5 failures/10^6 hours.

Figure 7-7 is a graphical look at the failure rates of the individual components comprising the processor. In ERV-87-05 it was assumed that the processor would contain 1 Mbyte of RAM and 1 Mbyte ROM. The failure rate of the RAM clearly dominated the overall failure rate of the processor. Based upon CSDL previous experiences, 128 Kbytes is typically sufficient for flight computers with execution requirements similar to the ERV. However, more ROM may be needed. With an increase in ROM storage, the failure rates of RAM and ROM are similar and the overall processor failure rate is much less.

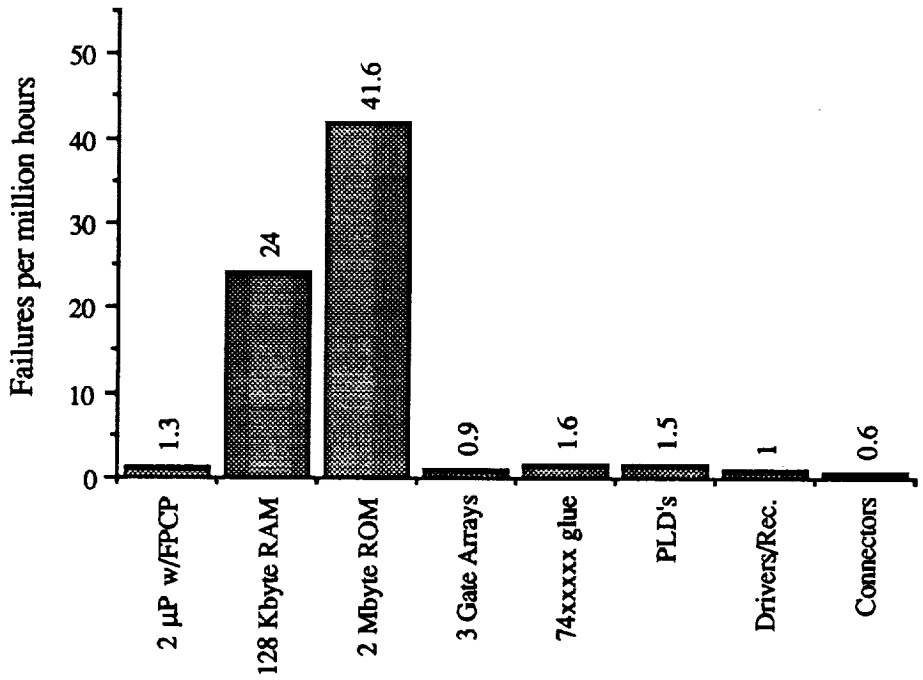

Fig ure 7-7. *Processor Failures by Device*

Failure rates of the interfaces to the I/O devices were also calculated. The I/O interfaces were assumed to be composed of the following parts with the following failure rates:

Device	Quantity		Failures per million hours		
Channel Core to IOB Interface - 1.2 failures/10^6 hours.					
Bus Controller	1	x	0.09	=	0.09
Bus Arbiter	1	x	0.05	=	0.05
Driver/Receiver	8	x	0.033	=	0.26
PAL	4	x	0.15	=	0.6
Connector	1	x	0.16	=	0.16
IOB to Air Data Interface - 4.0 failures/10^6 hours.					
Driver/Receiver	4	x	0.033	=	0.13
Analog/Digital Converter	1	x	1.8	=	1.8
Analog Multiplexer	1	x	0.4	=	0.4
Track/Hold Amplifier	1	x	0.9	=	0.9
Decoder/PAL	4	x	0.15	=	0.6
Connector	1	x	0.16	=	0.16

IOB to Skids Interface - 5.4 failures/10^6 hours.

Driver/Receiver	4	x	0.033	=	0.13
Digital/Analog Converter	1	x	1.4	=	1.4
Analog/Digital Converter	1	x	1.8	=	1.8
Analog Multiplexer	1	x	0.4	=	0.4
Track/Hold Amplifier	1	x	0.9	=	0.9
Decoder/PAL	4	x	0.15	=	0.6
Connector	1	x	0.16	=	0.16

IOB to Radar Altimeter Interface - 4.0 failures/10^6 hours.

Driver/Receiver	4	x	0.033	=	0.13
Analog/Digital Converter	1	x	1.8	=	1.8
Analog Multiplexer	1	x	0.4	=	0.4
Track/Hold Amplifier	1	x	0.9	=	0.9
Decoder/PAL	4	x	0.15	=	0.6
Connector	1	x	0.16	=	0.16

IOB to GSE Interface - 2.3 failures/10^6 hours.

Bus Controller R.T. Interface	1	x	1.3	=	1.3
PAL	4	x	0.15	=	0.6
Driver/Receiver	8	x	0.033	=	0.26
Connector	1	x	0.16	=	0.16

IOB to 1553 Interface - 2.3 failures/10^6 hours.

Bus Controller R.T. Interface	1	x	1.3	=	1.3
PAL	4	x	0.15	=	0.6
Driver/Receiver	8	x	0.033	=	0.26
Connector	1	x	0.16	=	0.16

1553 to I/O Device Interface - 6.1 failures/106 hours.

Remote Terminal Interface	1	x	1.3	=	1.3
Driver/Receiver	4	x	0.033	=	0.13
Digital/Analog Converter	1	x	1.4	=	1.4
Analog/Digital Converter	1	x	1.8	=	1.8
Analog Multiplexer	1	x	0.4	=	0.4
Track/Hold Amplifier	1	x	0.9	=	0.9
Connector	1	x	0.16	=	0.16

The failure rates of the D/A Converter, A/D Converter, Track/Hold Amplifier, and the Remote Terminal Interfaces were supplied by a vendor. The rest of the devices' failure rates were calculated using the MIL-HDBK-217D guidelines.

All of the I/O devices connected to the 1553 bus were assumed to be connected to the bus via identical "generic" interfaces consisting of the parts listed above. A block diagram of the interface is shown in Figure 7-8.

Although many factors are involved in estimating failure rates of avionics, we feel

that we have considered the most dominant ones. Implementing space qualified parts and controlling the operating environment of the avionics are the two most significant methods of lowering device failure rates. Throughout our calculations we assumed that space qualified parts would be used but we incorporated the ERV worst case environment.

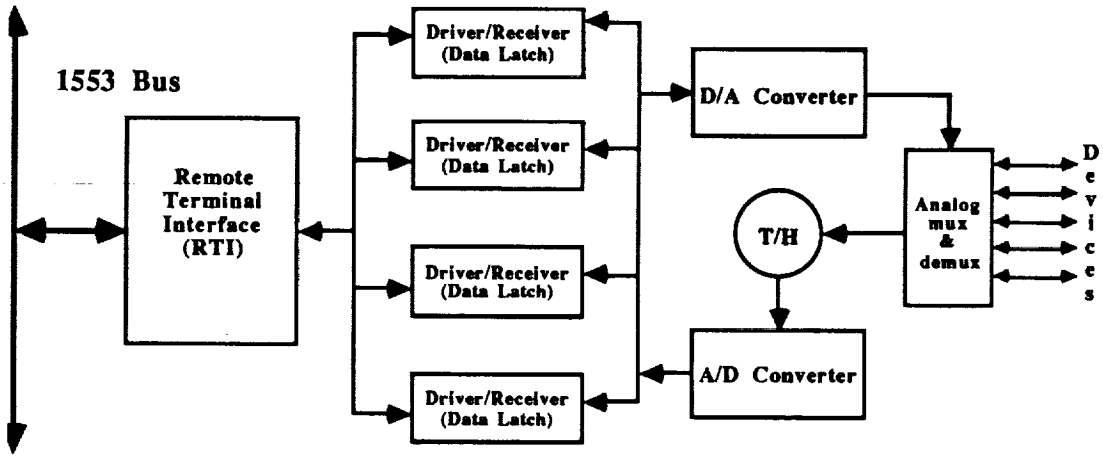

Figure 7-8. 1553 Interface Block Diagram

7.5 MARKOV MODELS

In order to quantify the reliability analysis of the ERV avionics suite, the failure modes of the system have been modelled as a series of Markov processes. In a Markov model of any system, each possible state of the system is identified. The associated state-transition rates are also determined. States in Markov models are characterized by their lack of history - the next state is solely dependent upon the current state and, thus, wholly independent from the previous states. Solutions to the probability of being in any state n as a function of time t, $P_n(t)$, can be found using differential equations or, more conveniently, standard software programs. Markov models have proven to be an effective means of analyzing the reliability of systems which undergo failure and repair [Babcock], [Gai et al], [Schabowsky et al].

One drawback of Markov modelling is that the number of states grows exponentially with the number of identified components. For systems whose failure modes can become quite complex, such as the ERV avionics suite - where failures of the Fault-Tolerant Processor (FTP), 1553 bus system, and ERV-specific critical devices can all cause system loss - the models easily become unwieldily large.

The Markov modelling of the ERV avionics suite must account for failures not only of the triplex core FTP, but for failure modes which involve the attached I/O buses (notably the triple redundant MIL-STD-1553 buses), and vehicle critical devices such as aerosurfaces, engines, guidance and navigation sensors, including the interfaces to these devices. As one soon realizes, the ERV Markov model quickly becomes very large.

To combat this problem, the analysis described in this section has made some conservative assumptions which reduce the avionics suite model into three simpler models. These assumptions also allow the analysis to consider the simpler models as independent, thus making them easier to handle.

The three models considered in the analysis are:

1. A model for the triplex core flight computer (the FTP). This results in a probability of triplex core loss (P_{TCL}).
2. A model for the 1553 bus I/O system, resulting in a probability of I/O system loss (P_{IOL}).
3. A model for the devices critical to ERV vehicle success, resulting in a probability of critical device loss (P_{CDL}).

The probability of vehicle loss (P_{VL}) can be determined as the probability of any of the above three events occurring, that is,

$$P_{VL} = P_{TCL} \cup P_{IOL} \cup P_{CDL}. \qquad (7\text{-}1)$$

This section describes in detail the models which were employed to determine P_{TCL}, P_{IOL}, and P_{CDL}. The results of those models, based upon failure rates determined in the previous section are discussed in Section 7.6. The actual simulations were run using Mark 1 and SURE, both Markov model solvers. The input source files for these programs are found in Appendices C and D, respectively.

7.5.1 Probability of Triplex Core Loss (P_{TCL})

The Markov model which describes the failure characteristics of the ERV triplex FTP core bases its state transition rates on the following:

- The (hardware) failure rate of each channel's *processor* (λ_p).
- The (hardware) failure rate of each channel's *interstage* (λ_i).

- The (software) rate for fault identification and reconfiguration (μ).
- The second failure coverage (C_2)[6].

The model is depicted in Figure 7-9.

The states in the model can be categorized into six classifications, starting with the healthiest:

- **NO FAILURES** (S1 IN THE MODEL)

  A fully operative FTP is identified by this state.

- **SINGLE UNIDENTIFIED FAILURE** (S2 - S7)

  Upon the first failure of any channel's processor or interstage, the model transitions from S1 to any of S2 - S7, depending upon which hardware fault-containment region failed. Although the FTP hardware has correctly detected and masked the failure and thus prevented it from corrupting the system, the FTP software must isolate the fault and reconfigure the system around it. These states are the point in time before the FTP Fault Detection, Isolation, and Reconfiguration (FDIR) software has properly identified the failure. Should a second failure occur when the system is in any of these states, a catastrophic failure would occur (loss of system).

- **SINGLE FAILURE IDENTIFIED - DUPLEX CONFIGURATION** (S8 - S10)

  Once FDIR has identified the first failure and reconfigured the system around it - by effectively taking the entire faulty channel off-line - the system gracefully degrades into a duplex mode, with only two channels operating correctly. Duplex operation presents no harmful consequences to the proper control of the spacecraft.

---

[6] $C_1 = 100\%$.

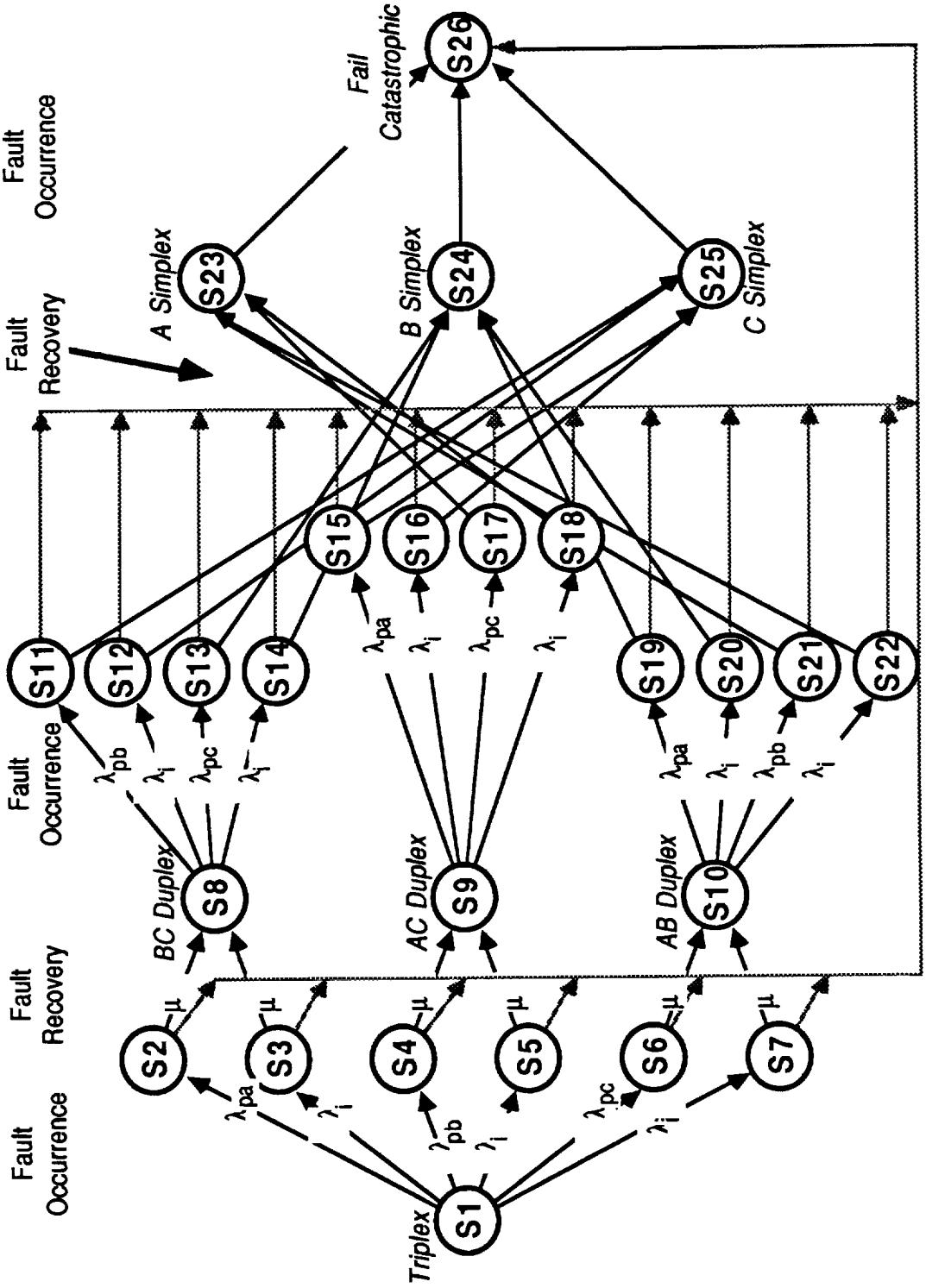

Figure 7-9. ERV FTP Triplex Core Markov Model

- **SECOND UNIDENTIFIED FAILURE** (S11 - S22)

  A second failure could occur in the system. As in the Single Unidentified Failure state, the system is in a limbo state until the FDIR properly identifies the failure or a third failure occurs, causing system loss. While the triplex FTP configuration provides 100% coverage for the first failure, once the system degrades into a duplex system, the coverage is lowered. CSDL estimates coverage for the second failure to be 95%: 90% coverage by the FDIR and a 50/50 chance of correctly guessing for the remaining 10%. Thus, once in S11 - S22, the system has a 5% chance of loss of integrity, assuming no other failures arise.

- **SECOND FAILURE IDENTIFIED - SIMPLEX CONFIGURATION** (S23 - S25)

  If the system properly recovered from the second failure, it degrades into a simplex mode with a single channel operative. With no other failures in the system (bus loss, critical device failure, etc.), a simplex configuration can control the vehicle. (Should a 1553 bus also fail, vehicle loss is assumed. This is discussed further in this section.)

- **CATASTROPHIC FAILURE - SYSTEM LOSS** (S26)

  Loss of the FTP system integrity can occur in any of three ways: if a second failure occurs when the first has not yet been identified (S2- S7); inability of the system to identify the second failure; the occurrence of three failures (in separate fault-containment regions).

Figure 7-10 shows the devices that constitute each channel of the FTP. The hardware failure rates (λ_p and λ_i) were obtained from the work described in Section 7.4. These rates are tabulated in Figure 7-11. The processor failure rate, λ_p, includes the processor in the FTP channel along with all of the interfaces on that channel's IOB. As a basis for the software identification and reconfiguration rate (μ), twice the period of the expected FDIR rate was used. This is a worst-case estimate since a failure could occur just as the FDIR passes through its fault identification scheme; reconfiguration of that failure could not occur until the end of the next FDIR execution. In the ERV, FDIR is expected to be run with the fastest real-time rate group (estimated at 50 Hz, for now). Thus $1/\mu = 2$ (1/50) = 40 ms. As previously mentioned (¶S11 - S22), the second failure coverage, C_2, was set to 95% for these models.

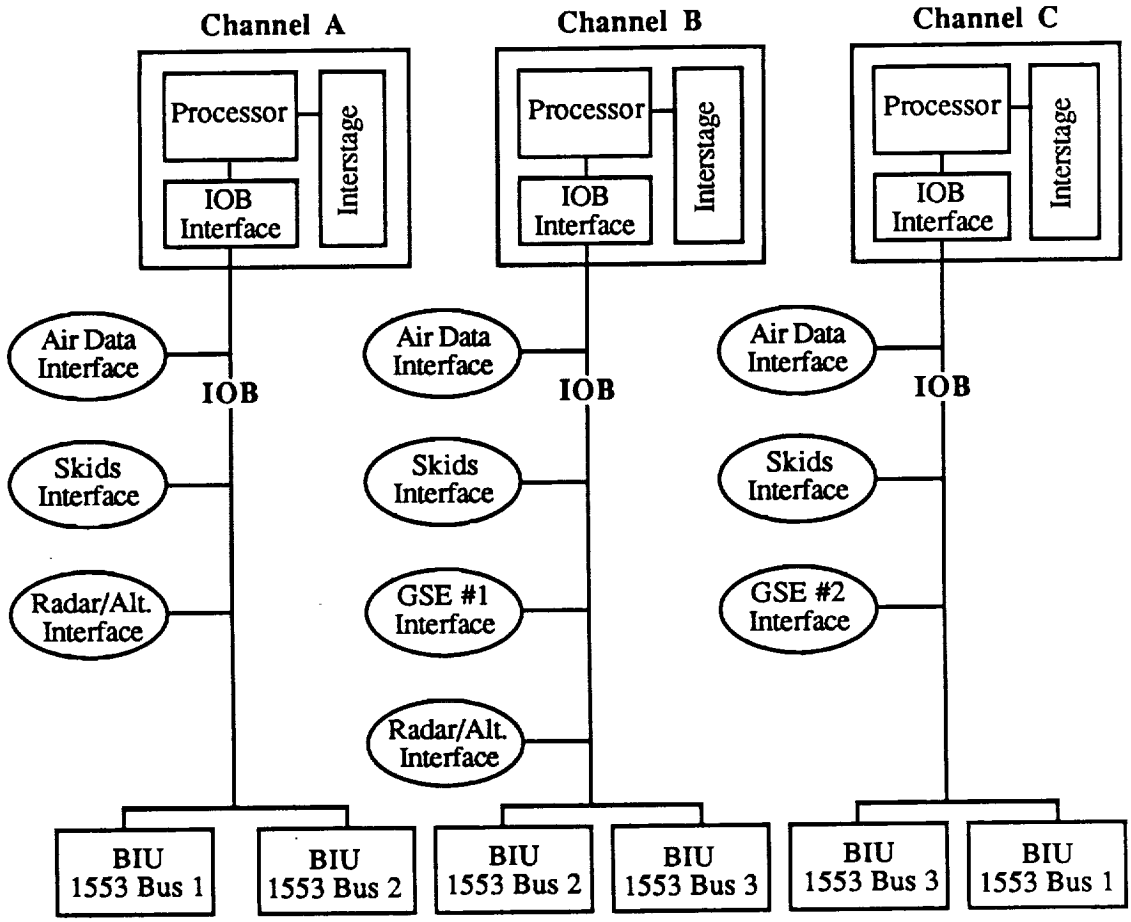

Figure 7-10. FTP Core

λ_p for Channel A	λ_p for Channel B	λ_p for Channel C
Processor: 72.6 f./10^6 hrs	Processor: 72.6 f./10^6 hrs	Processor: 72.6 f./10^6 hrs
IOB Int.: 1.2 f./10^6 hrs	IOB Int.: 1.2 f./10^6 hrs	IOB Int.: 1.2 f./10^6 hrs
Air Data Int.:4.0 f./10^6 hrs	Air Data Int.: 4.0 f./10^6 hrs	Air Data Int.: 4.0 f./10^6 hrs
Skids Int.: 5.4 f./10^6 hrs	Skids Int.: 5.4 f./10^6 hrs	Skids Int.: 5.4 f./10^6 hrs
Radar Alt.: 4.0 f./10^6 hrs	GSE Int.: 2.3 f./10^6 hrs	GSE Int.: 2.3 f./10^6 hrs
1553 #1 Int.:4.6 f./10^6 hrs	Radar Alt.: 4.0 f./10^6 hrs	1553 #3 Int.: 4.6 f./10^6 hrs
1553 #2 Int.: 4.6 f./10^6 hrs	1553 #2 Int.: 4.6 f./10^6 hrs	1553 #1 Int.: 4.6 f./10^6 hrs
	1553 #3 Int.: 4.6 f./10^6 hrs	
λ_p = 91.8 failures/10^6 hrs.	**λ_p = 94.1 failures/10^6 hrs.**	**λ_p = 90.8 failures/10^6 hrs.**
λ_i = 1.5 failures/10^6 hrs. (for all channels)		

Figure 7-11. Tabulation of λp and λi

The Markov model for the FTP failure modes notably excludes repairs. Being an

unmanned, short-mission vehicle, it would be impossible to repair hard failures in-flight. This assumption, however, implies that even transient failures - those that cause a channel to be faulty for a short time and then disappear - are irreparable. Once a failure occurs in this model, the system degrades to a less redundant process. A complete list of the FTP model states and transition rates is given in Appendix B.

Vehicle loss due to triplex core loss is not solely based upon the probability of reaching the catastrophic failure state (S26) in the FTP model. Should the FTP degrade to simplex mode and lose one of that channel's attached 1553 buses, vehicle loss is also assumed. The inclusion of the probability of this combination, $P_{S/B}$ (simplex mode/bus failure), is because the sole controlling channel needs access to both of its 1553 buses for successful vehicle operation.

If A, B, and C are the probabilities of simplex operation by the respective channel and a', b', and c' are the failure probabilities of the 1553 buses not attached to the respective channel, then one sees that,

$$P_{S/B} = A \cdot (b' + c') + B \cdot (a' + c') + C \cdot (a' + b') \qquad (7\text{-}2)$$

since A, B, and C are mutually exclusive events. $P_{S/B}$ has been conservatively simplified by assuming that any 1553 bus failure while the FTP is in simplex mode results in vehicle loss. Thus, $P_{S/B}$ is assumed to be,

$$P_{S/B} = (A + B + C) \cdot (a' + b' + c') \qquad (7\text{-}3)$$

which overestimates the actual value by $(A \cdot a') + (B \cdot b') + (C \cdot c')$, but makes the calculations simpler.

The aggregate model for P_{TCL} is shown below in Figure 7-12. From this model, one can see that the assumed P_{TCL} is calculated to be

$$P_{TCL} = P_{S26} + P_{S/B}, \qquad (7\text{-}4)$$

where

$$P_{S/B} = (P_{S23} + P_{S24} + P_{S25}) \cdot \text{Prob[single 1553 failure]}, \qquad (7\text{-}5)$$

Prob[single 1553 failure] is discussed thoroughly Section 7.5.2.

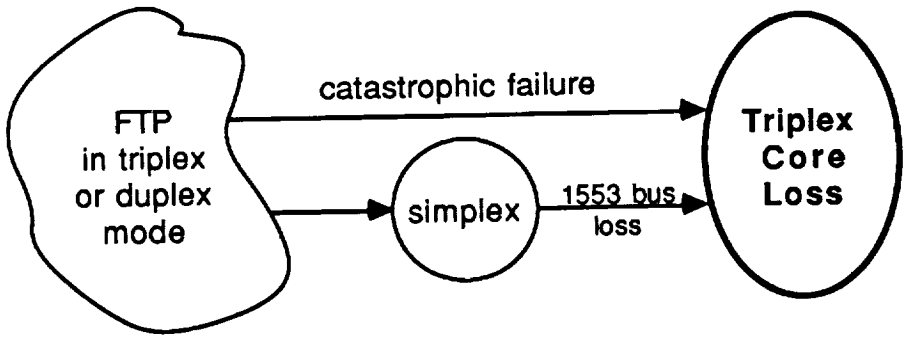

Figure 7-12. Aggregate Model for P_{TCL}

7.5.2 PROBABILITY OF SYSTEM BUS FAILURE (P_{IOL}).

The second major area of concern in the reliability model is the 1553 subsystem which provides the triplex core with its I/O interface. If access to this system is lost, regardless of the condition of the core, vehicle loss will result. If the failure of the I/O system is considered independent from triplex core failures, the probability of vehicle loss can be determined by ORing the two probabilities.

I/O system loss has been assumed to include these events:

- loss of two 1553 buses
- loss of one 1553 bus and one interface in a duplex interface to an I/O device
- loss of one 1553 bus and two interfaces in a triplex interface to an I/O device
- loss of all interfaces to an I/O device (two/duplex, three/triplex)

In other words,

$$P_{IOL} = P_{DBF} \cup (P_{SBF} \cap (P_{DISF} \cup P_{TIDF})) \cup P_{IF} \qquad (7\text{-}6)$$

where P_{IOL} is the probability of I/O subsystem (1553) loss,
P_{DBF} is the probability of a double (1553) bus failure,
P_{SBF} is the probability of a single (1553) bus failure,
P_{DISF} is the probability of duplex interface single failure,
P_{TIDF} is the probability of triplex interface double failure, and
P_{IF} is the probability of an entire interface failure.

Each of the three OR terms [P_{DBF}, $P_{SBF} \bullet (P_{DISF} \cup P_{TIDF})$, and P_{IF}] are described in the following sections.

7.5.2.1 PROBABILITY OF A DOUBLE BUS FAILURE (P_{DBF})

The probability of losing two 1553 buses can be modelled by the process shown in Figure 7-13.

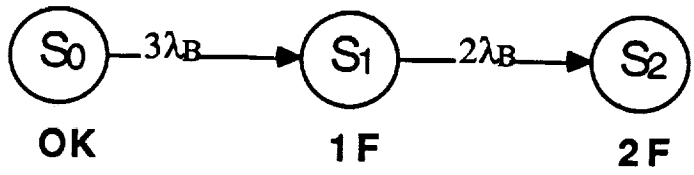

Figure 7-13. *Probability of Double (1553) Bus Failure (P_{DBF})*

The process starts in State 0 (S_0) with three operational 1553 buses. It transitions to S_1 (a single bus failure) at a rate of $3\lambda_B$, where λ_B is the failure rate for one 1553 bus. After one bus has been lost, the model transitions to S_2 (double bus failure), now at a rate of $2\lambda_B$, since one bus has already failed.

The calculation of λ_B was determined by making the following assumptions:

- Only hardware directly interfacing to a bus can fail the bus. This includes the channel Bus Interface Units (BIUs) and the device interfaces' Remote Terminal Interfaces (RTIs) and driver/receiver (D/R) pairs. See Figure 7-9.
- Only 10% of the failures of that hardware are active (not passive) failures and result in the loss of the bus.

The failure rate, λ_B, is thus calculated as,

$$\lambda_B = 0.10 \bullet ((\lambda_{RTI} + \lambda_{D/R}) \bullet n + 2\lambda_{BIU}) \qquad (7\text{-}7)$$

where λ_{RTI} is the failure rate of the Remote Terminal Interface,
$\lambda_{D/R}$ is the failure rate of the driver/receiver pairs,
n is the number of 1553 devices on the bus, and
λ_{BIU} is the failure rate of the channel's 1553 Bus Interface Unit.

From Section 7.4, λ_B is calculated as

$$\lambda_B = 0.10 \bullet ((1.3 + 0.13) \bullet 7 + 2(1.2)) \text{ failures}/10^6 \text{ hours}$$
$$= 1.241 \times 10^{-6} \text{ failures/hour,}$$

and using the simple model of Figure 7-13, both P_{SBF} and P_{DBF} are easily determined.

7.5.2.2 PROBABILITY OF SINGLE BUS AND PARTIAL I/O DEVICE INTERFACE FAILURES

The 1553 I/O system can also be considered failed if a single bus is lost and the interfaces to the same device on the non-failed buses are lost. For example, if 1553 B_1 fails and the RCS interfaces on B_2 and B_3 also fail, the system would be considered failed since no interface to the RCS jets would be available. This is given by the quantity

$$P_{SBF} \cap (P_{DISF} \cup P_{TIDF}) \qquad (7\text{-}8)$$

in (7-6).

This condition is simplified if one disregards which bus has failed. Similar to the discussion in Section 7.5.1 for determining $P_{S/B}$, this failure mode is assumed if a single bus fails and any majority of an I/O device interface (one/duplex, two/triplex) fails. If there are *n* duplex interfaces and *m* triplex interfaces, the above quantity becomes equal to

P_{SBF}
$\cap$
(Prob[at least one of n duplex interfaces incurred a single failure]
$\cup$
Prob[at least one of m triplex interfaces incurred two failures]).

Since P_{SBF} has been determined (Section 7.2.1), the latter two terms in (7-8), now described by the above quantities, need only be determined. Before arriving at the formulas for these probabilities, some discussion is necessary to further clarify the model.

Suppose the ERV has n I/O devices which have duplex interfaces. The failure model for one of these interfaces is described by Figure 7-14. In the model, λ is the failure rate of the device interface (λ = 6.1 failures/10^6 hours from Section 7.4).

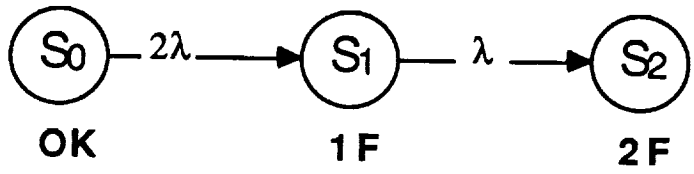

Figure 7-14. Simple Markov Model for Failure of a Duplex Device Interface

Let P_0 = Prob[S_0 at time t], P_1 = Prob[S_1 at time t], and P_2 = Prob[S_2 at time t], where t is some specified value. For n devices at time t, the probability that all are OK - that is, in S_0 - is $(P_0)^n$, since these are all independent events. Likewise, the probability that at least one of n has failed is $1 - (P_0)^n$. Also note that the probability that all have failed (none in S_0) is $(1 - P_0)^n$. These probabilities are more clearly seen if the event space is drawn out for the process of n events (Figure 7-15).

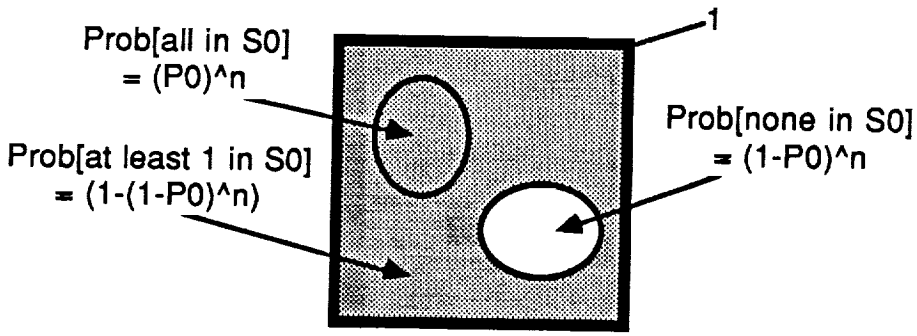

Figure 7-15. Event Space for n Duplex Device Interfaces

For this section, we are concerned with P_1 over n events. Specifically, the probability that at least 1 of n events resulted in S_1 (and not S_2) needs to be determined. Using similar reasoning as the above paragraph and figures, the probability that none of n events have resulted in S_1 is $(1 - P_1)^n$. The probability that at least one of the n events is in S_1 is merely the complement of this quantity, i.e., $1 - (1 - P_1)^n$.

For ERV I/O devices with triplex interfaces (assume m of these) the model becomes slightly more complex, although the analysis is similar. These interfaces are modelled by the illustration in Figure 7-16. The process starts in S_3 with no failures. At a rate of 3λ (λ = 6.1 failures/10^6 hours, again from Section 7.4), the interface decays to a duplex configuration. A second failure occurs at a rate of 2λ and the device is then operated by a simplex interface (S_5). At a rate of λ, a third failure occurs and the device is lost.

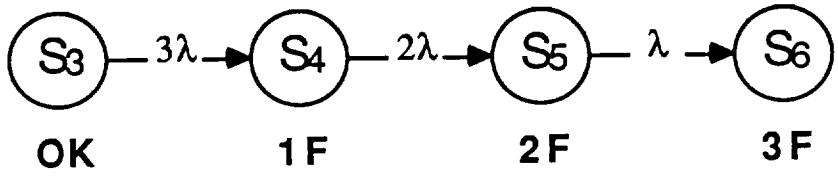

Figure 7-16. Simple Markov Model for Failure of a Triplex Device Interface

Again letting P_x = Prob[S_x at time t], we are interested in P_5 over m events, or more specifically, the probability that at least 1 of m devices resulted in S_5. Following the same reasoning which was used for duplex interfaces, this quantity is $1 - (1 - P_5)^m$.

The union of these probabilities for failures with duplex and triplex interfaces is easily determined using $A \cup B = A + B - A \bullet B$, since these events are not mutually exclusive.

In summary then, the probability of a single bus failure and a failure of an I/O device can be assumed to be

$$P_{SBF} \cap (P_{DISF} \cup P_{TIDF}) = PSBF \cap ((1 - (1- P_1)^n) \cup (1 - (1- P_5)^m)), \quad (7\text{-}9)$$

where P_1 and P_5 are given by the models in Figures 7-14 and 7-16, respectively, and n and m are the number of I/O devices with duplex and triplex interfaces, respectively.

7.5.2.3 PROBABILITY OF INTERFACE FAILURE (P_{IF})

An entire interface failure (two/duplex, three/triplex) is also a possible failure mode for P_{IOL}. Conveniently, this probability has been determined in the previous section. In the models of Figures 7-11 and 7-13, entire interface failures are represented by S_2 and S_6. The probability either of these events occurring is merely the union of P_2 and P6 (This must be done using $A \cup B = A + B - A \bullet B$, since these events are also not mutually exclusive.)

7.5.2.4 COMBINING THE RESULTS TO OBTAIN THE PROBABILITY OF I/O LOSS (P_{IOL})

The previous sections have discussed each of the three terms found in (7-6) which contribute to P_{IOL}, namely,

$$P_{IOL} = P_{DBF} \cup (P_{SBF} \cap (P_{DISF} \cup P_{TIDF})) \cup P_{IF} \qquad (7\text{-}6)$$

To determine the exact formula for P_{IOL}, careful attention must be paid to the event space in question. Figure 7-17 diagrams the event space for P_{IOL} loss. From the figure and the above equation (7-6), we can determine the probabilities in terms of the proper AND and OR functions. Alternatively, we can use the fact that

$$A \cup B \cup C = A + B + C - A \cdot B - A \cdot C - B \cdot C + A \cdot B \cdot C, \qquad (7\text{-}10)$$

letting $P_{DBF} = A$, $P_{IF} = C$, and $P_{SBF} \cap (P_{DISF} \cup P_{TIDF})$ equal the single term B. The event space diagram can be used to simplify the resulting P_{IOL} by removing zero terms (the fact that SBF and DBF are mutually exclusive events, etc).

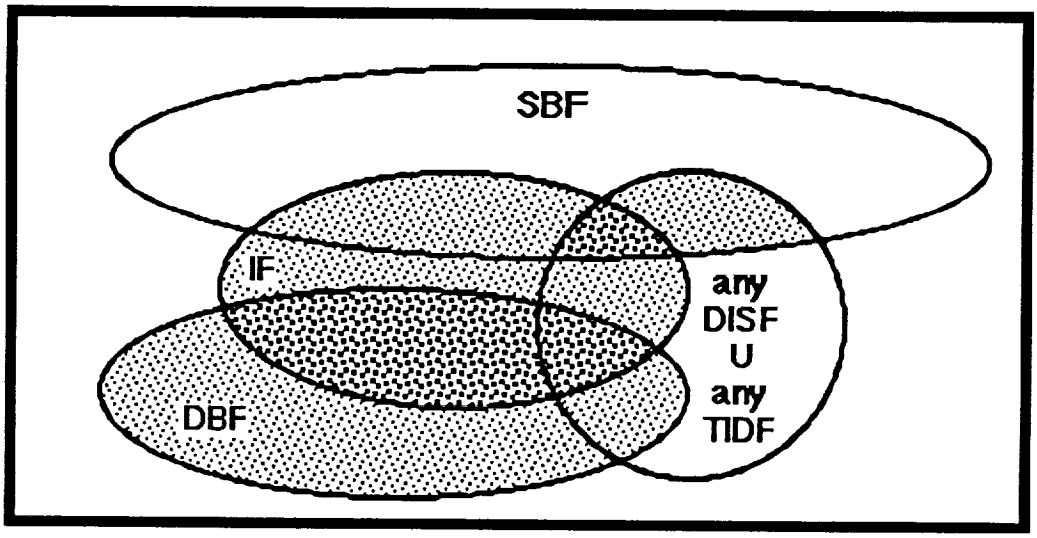

Figure 7-17. Event Space for P_{IOL}

The gray areas of the figure are those which add to P_{IOL}. By inspection of the figure, one can see the overlapping events (darker gray) and deduce the following:

$$P_{IOL} = P_{DBF} + P_{SBF} \cdot (P_{DISF} \cup P_{TIDF}) + P_{IF} - P_{DBF} \cdot P_{IF} - P_{SBF} \cdot (P_{DISF} \cup P_{TIDF}) \cdot P_{IF} \qquad (7\text{-}11)$$

Equation (7-11) can also be derived by using (7-10) with $A = P_{DBF}$, $B = P_{SBF} \cdot (P_{DISF} \cup P_{TIDF})$, and $C = P_{IF}$. We see that

$$
\begin{aligned}
P_{IOL} = {} & P_{DBF} \\
& + P_{SBF} \bullet (P_{DISF} \cup P_{TIDF}) \\
& + PIF \\
& - P_{DBF} \bullet P_{SBF} \bullet (P_{DISF} \cup P_{TIDF}) \\
& - P_{DBF} \bullet P_{IF} \\
& - P_{SBF} \bullet (P_{DISF} \cup P_{TIDF}) \bullet P_{IF} \\
& + P_{DBF} \bullet P_{SBF} \bullet (P_{DISF} \cup P_{TIDF}) \bullet PIF. \qquad (7\text{-}12)
\end{aligned}
$$

Since all $P_{DBF} \bullet P_{SBF}$ terms are zero, (7-12) reduces to

$$
\begin{aligned}
P_{IOL} = {} & P_{DBF} \\
& + P_{SBF} \bullet (P_{DISF} \cup P_{TIDF}) \\
& + PIF \\
& - \sout{P_{DBF} \bullet P_{SBF} \bullet (P_{DISF} \cup P_{TIDF})} \\
& - P_{DBF} \bullet P_{IF} \\
& - P_{SBF} \bullet (P_{DISF} \cup P_{TIDF}) \bullet P_{IF} \\
& + \sout{P_{DBF} \bullet P_{SBF} \bullet (P_{DISF} \cup P_{TIDF}) \bullet PIF}
\end{aligned}
$$

$$
\begin{aligned}
P_{IOL} = {} & P_{DBF} \\
& + P_{SBF} \bullet (P_{DISF} \cup P_{TIDF}) \\
& + PIF \\
& - P_{DBF} \bullet P_{IF} \\
& - P_{SBF} \bullet (P_{DISF} \cup P_{TIDF}) \bullet P_{IF}
\end{aligned}
$$

which is the same as (7-11).

In summary, the probability that the I/O system fails (P_{IOL}) is given by

$$
\begin{aligned}
P_{IOL} = {} & P_{DBF} + P_{SBF} \bullet (P_{DISF} \cup P_{TIDF}) + PIF \\
& - P_{DBF} \bullet P_{IF} - P_{SBF} \bullet (P_{DI'SF} \cup P_{TIDF}) \bullet P_{IF}
\end{aligned}
$$

where

$$
\begin{aligned}
& P_{DISF} \cup P_{TIDF} = \\
& ((1 - (1 - P_1)^n) + (1 - (1 - P_5)^m)) - ((1 - (1 - P_1)^n) \bullet (1 - (1 - P_5)^m))
\end{aligned}
$$

and where P_1 and P_5 are described by the models in Section 7.5.2.2.

7.5.3 PROBABILITY OF CRITICAL DEVICE LOSS (P_{CDL})

This analysis has also accounted for failures of I/O devices critical to ERV vehicle success. These include aerosurfaces, propulsion jets, and G/N devices. Due to the

preliminary stage of this analysis, however, each of these components could not be addressed directly. Instead, it has been assumed that the devices have a combined probability of failure of 10^{-7} after 10 hours. (This number has been determined by assuming each device has a failure rate of 10^{-9} failures/hour. There are approximately 10 such devices.) For future analysis, this number is easily altered.

In the model, P_{CDL} is easily unioned with P_{TCL} and P_{IOL} to obtain P_{VL}.

7.6 MARKOV RESULTS

Both SURE and Mark 1 model solvers were used to determine the state probabilities described in Section 7.5. The results for an 8-hour mission are tallied below.

7.6.1 PROBABILITY OF VEHICLE LOSS

Prob{Triplex Core Loss}	$P_{TCL} = 8.42 \times 10^{-8}$
Prob{I/O System Loss}	$P_{IOL} = 1.23 \times 10^{-8}$
Prob{Interface Failure}	$P_{IF} = 2.33 \times 10^{-9}$
Prob{Double Interface Single Failure or Triplex Interface Double Failure}	$P_{DISF} \cup P_{TIDF} = 6.83 \times 10^{-4}$
Prob{Single Bus Failure}	$P_{SBF} = 1.44 \times 10^{-5}$
Prob{Double Bus Failure}	$P_{DBF} = 6.91 \times 10^{-11}$
Prob{Simplex Processor Only}	$P_{SIMP} = 2.09 \times 10^{-5}$
Prob{Critical Device Loss}	$P_{CDL} = 8.0 \times 10^{-8}$
Prob{Vehicle Loss}	**$P_{VL} = 1.77 \times 10^{-7}$**

Figures 7-18, 7-19, and 7-20 illustrate the probability characteristics over time. Set to the same scale, the figures show P_{TCL}, P_{IOL}, and P_{VL}, respectively, from t = 0.1 hours to 10,000 hours.

While Figures 7-18 and 7-19 resemble expected probability curves, Figure 7-20 has an interesting slope character. From Section 7.5 one can recall that

$$P_{VL} = P_{TCL} \cup P_{IOL} \cup P_{CDL}$$

where P_{CDL} is the probability of a critical device loss. In Section 7.5.3 it was noted that $\lambda_{CDL} = 10^{-9}$ failures/hour. Thus, P_{CDL} should be a linear function of time. At the start of the mission, where P_{TCL} and P_{IOL} are significantly less than 10^{-9}, P_{CDL} is the dominant term in P_{VL}. As the mission progresses, P_{CDL} becomes less dominant and P_{TCL} and P_{IOL} rule P_{VL}. Thus, as one can see in the figure, P_{VL} has an interesting twist between early mission times and later ones.

C-2

Figure 7-18. Probability of Triplex Core Loss

Figure 7-19. Probability of I/O System Loss

Figure 7-20. Probability of Vehicle Loss

7.6.2 SENSITIVITY OF VEHICLE LOSS TO VARIATION OF TRANSITION RATES

In addition to the nominal numbers given above, sensitivity analyses were performed. Sensitivities of the models to processor failure rate (λ_p), second failure coverage (C_2), 1553 Bus failure rate (λ_B), and device interface failure rate were run. The results are given below.

Processor Failure Rate (λ_p)		
	2x Nominal	1/2 Nominal
P_{TCL}=	3.33×10^{-7}	2.17×10^{-8}
P_{IOL}=	1.23×10^{-8}	1.23×10^{-8}
P_{VL}=	4.25×10^{-7}	1.14×10^{-7}

Second Failure Coverage (C_2)		
	90%	85%
P_{TCL}=	1.68×10^{-7}	3.32×10^{-6}
P_{IOL}=	1.23×10^{-8}	1.23×10^{-8}
P_{VL}=	2.60×10^{-7}	3.44×10^{-7}

Bus Failure Rate (λ_B)		
	2x Nominal	1/2 Nominal
P_{TCL}=	8.42×10^{-8}	8.42×10^{-8}
P_{IOL}=	2.23×10^{-8}	7.32×10^{-9}
P_{VL}=	1.87×10^{-7}	1.72×10^{-7}

Device Interface Failure Rate		
	2x Nominal	1/2 Nominal
P_{TCL}=	8.42×10^{-8}	8.42×10^{-8}
P_{IOL}=	2.93×10^{-8}	5.48×10^{-9}
P_{VL}=	1.93×10^{-7}	1.70×10^{-7}

The sensitivity analyses show that the effect of processor failure rate and the second failure coverage are the dominant factors in determining P_{VL}. Changing the bus failure rate and the device interface failure rate by a factor of two resulted in only a 10 percent change in P_{VL}. However, similar changes in processor failure rate and second failure coverage yielded a 50 to 240 percent change in P_{VL}. The processor failure rate and second failure coverage contribute solely to the probability of triplex core loss (P_{TCL}) indicating that P_{TCL} dominates over P_{IOL} and P_{CDL} in determining P_{VL}.

7.6.3 IMPROVING ERV RELIABILITY

There are many ways of improving system reliability (i.e., decreasing PVL). Since ERV avionics are still the design phase, three major suggestions for reaching this goal are given below.

1. Utilize a quadruplex core processor. A quad-redundant processor would improve second failure coverage to 100%, and give 95% coverage of third failures. This should increase the reliability of the processing core by 1-2 orders of magnitude.

2. Decrease the failure rate of a processor channel. This can easily be accomplished by decreasing on-board memory or by using SECDED (Single Error Correction, Double Error Detection) memory since memory failures comprise 85% of a processor channel's failure rate.

3. Decrease the failure rate of I/O device interfaces. This could be accomplished by simplifying the interfaces, or by building redundancy into the device interfaces themselves.

8.0 CONCLUSIONS

This report has presented some of the important research contributions which, over the past twelve months, have resulted in the preliminary design and analysis of the avionic suite for an Entry Research Vehicle. The vehicle electronics could now enter the implementation stages of detailed design, prototyping, and flight-qualified fabrication and integration.

The strawman architecture which grew out of the initial task assignment was further refined as more detailed specifications of the flight hardware and software were examined. While it was apparent from the onset of this program that the criticality of the mission would require a digital flight computer which could tolerate internal failures, the degree and implementation of redundancy in that computer was not entirely obvious. The architecture chosen was the Fault-Tolerant Processor because of its resiliency to malicious failures with 100% coverage. (Failure coverage below 100% was shown to be inadequate for this vehicle.) A doubly-redundant architecture could not meet the reliability requirements. However, quadruplex redundancy in the design provided an overabundance of hardware for the stated goals. Triplex hardware was found to be sufficient.

Coupled to the core fault-tolerant flight computer are a set of replicated MIL-STD-1553 buses, each with mixed redundancy I/O devices. The interface between the central processors and the distributed network of I/O was carefully designed to avoid failure correlation between the two subsystems. Furthermore, the devices attached to the buses are not all fully replicated: analytical redundancy among dissimilar sensors and actuators was exploited to reduce the amount of hardware onboard the vehicle without compromising reliability.

An outline of the requirements of the bus structures for all of the avionics was also drawn. Included in this analysis was the design of typical interface units at the three major levels of the system: intrachannel, interchannel, and input/ouput.

In addition to these architectural issues, design details of the flight electronics were examined. A study of microprocessors illustrated problems with entrusting design decisions to performance benchmark statistics alone. For the central processing unit, characteristics such as power consumption, chip size, memory and I/O management capabilities, device reliability, software support, and short-term risks were compared to make a credible selection. To establish hardware viability, a mechanical analysis was

performed which suggested packaging, cooling, environmental, and other physical requirements. A scheme presented in NASA Technical Brief 71-10088 was proposed as the principle cooling method for the electronics.

A reliability analysis was perfomed for the candidate architecture using Markov modelling techniques. To simplify the analysis, conservative assumptions were made which divided the avionics suite for evaluation as three separate entities. Rigorous probabalistic methods were employed to combine the independent analyses. Assumptions which were made in the process proved to be significant only for evaluations performed for flight durations greater than 10^4 hours. Since ERV missions are expected to be on the order of 10 hours, the results - showing that the architecture meets the desired probability of mission failure of 10^{-6} - can be credibly accepted.

A viable candidate architecture for the Entry Research Vehicle has been presented. The underlying principles of this architecture represent a foundation that may be applicable to other advanced space transportation systems. Shuttle II, the National Aerospace Plane, Heavy Lift Vehicles, and other advanced space vehicles which require ultra-reliable high-performance control in real time can make use of the developments of this task assignment.

BIBLIOGRAPHY

Alger, L.S. and J.H. Lala, "A Real-Time Operating System for a Nuclear Power Plant Computer", C. S. Draper Laboratory technical report #CSDL-P-2718. Presented at *IEEE Computer Society Real-time Systems Symposium*, December 1986.

Babcock, IV, P. S., *An Introduction to Reliability Modeling of Fault-Tolerant Systems*, C. S. Draper Laboratory technical report #CSDL-R-1899, September 1986.

Byington, L. and D. Theis, "Air Force Standard 1750A ISA Is the New Trend," *Computer*, Vol. 19, No. 10 (November 1986), pp. 50-59.

Cooper, T., W. Bell, F. Lin., N. Rasmussen, "A Benchmark Comparison of 32-bit Microprocessors," *Micro*, Vol. 6, No. 4 (August 1986), pp. 53-58.

Dolev, D., "The Byzantine Generals Strike Again", *Journal of Algorithms 3*, 1982, pp. 14-30.

Gai, E., J. V. Harrison, and R. H. Luppold, "Reliability Analysis of a Dual Redundant Engine Controller," *IEEE Transactions on Reliability*, Volume R-32, April 1983.

Gauthier, R.J., *The AIRLAB Fault-Tolerant Processor: Physical Implementation*, C. S. Draper Laboratory technical report #CSDL-R-1928, December 12, 1986.

Hopkins, Jr., A.L., J.H. Lala, T.B. Smith, "Evolution of Fault-Tolerant Computing at CSDL", C. S. Draper Laboratory technical report #CSDL-P-2701, July 1986. Reprinted in *Proceedings of the Symposium on the Evolution of Fault Tolerant Computing*, July 1986.

Kriegsman, B., R.T. Richards, T.J. Brand, R.J. Gauthier, *Guidance and Navigation System Studies for Entry Research Vehicle*, C. S. Draper Laboratory technical report #CSDL-P-2864, April 1986.

Lala, J, H., "Advanced Information Processing System", *Proceedings of the AIAA/IEEE Sixth Digital Avionics Systems Conference*, December 1984, pp. 199 - 210.

Lala, J.H., "A Fault-Tolerant Computer for Nuclear Power Plant Applications", C. S. Draper Laboratory technical report #CSDL-P-2022, April, 1985. Presented at *EPRI Seminar: Power Plant Digital Control and Fault-Tolerant Microcomputers*, April 1985.

Lala, J.H, "A Byzantine Resilient Fault-Tolerant Computer for Nuclear Power Plant Applications", C. S. Draper Laboratory technical report #CSDL-P-2700, July 1986.

Reprinted in *Digest of Papers: The Sixteenth International Symposium on Fault-Tolerant Computing*, July 1986.

Lala, J. H., L.S. Alger, R.J. Gauthier, and M.J. Dzwonczyk, "A Fault-Tolerant Processor Architecture to Meet Rigorous Failure Requirements", C. S. Draper Laboratory technical report #CSDL-P-2705, July 1986. Reprinted in *Proceedings of the AIAA/IEEE Seventh Digital Avionics Systems Conference* , October, 1986, pp. 555 - 562.

Lamport, L., R. Shostak, P. Marshall, "The Byzantine Generals Problem", *ACM Transactions on Programming Languages and Systems*, Vol. 4, No. 3 (July 1982), pp. 382-401.

Marrin, K., "Microprocessor experts tackle benchmarking, support and architectural issues in open debate," *Computer Design*, Vol. 26, No. 1 (January 1, 1987), pp. 21-29.

Schabowsky, Jr., R. S., and W.W. Weinstein, "On the Evaluation and Validation of Fault-Tolerant Digital Control Systems", *Proceedings of the EPRI Seminar: Power Plant Digital Control and Fault-Tolerant Microcomputers*, April 1985.

Smith, T.B, "Fault-Tolerant Clocking System", *Digest of Papers: The Eleventh International Symposium on Fault-Tolerant Computing*, June 1981.

Smith, T.B., "Fault Tolerant Processor Concepts and Operation", C. S. Draper Laboratory technical report #CSDL-P-1727, May 1983. Reprinted in *Digest of Papers: The Fourteenth International Symposium on Fault-Tolerant Computing*, June 1984.

Suydam, W., "Developers Debate Merits of Military Microprocessors," *Computer Design*, Vol. 26, No. 3 (February 1, 1987), pp. 38-44.

APPENDIX A: NASA TECH BRIEF 71-10088

APPENDIX A: NASA TECH BRIEF 71-10088

APPENDIX B: FTP MARKOV MODEL DETAILS

The following are the states defined in the Markov model of the ERV Fault-Tolerant Processor. The model is depicted in Figure 7-9.

S1	No Failures
S2	Processor A Failure
S3	Interstage A Failure
S4	Processor B Failure
S5	Interstage B Failure
S6	Processor C Failure
S7	Interstage C Failure
S8	BC Duplex, A Off-line
S9	AC Duplex, B Off-line
S10	AB Duplex, C Off-line
S11	BC Duplex, Processor B Failure
S12	BC Duplex, Interstage B Failure
S13	BC Duplex, Processor C Failure
S14	BC Duplex, Interstage C Failure
S15	AC Duplex, Processor A Failure
S16	AC Duplex, Interstage A Failure
S17	AC Duplex, Processor C Failure
S18	AC Duplex, Interstage C Failure
S19	AB Duplex, Processor A Failure
S20	AB Duplex, Interstage A Failure
S21	AB Duplex, Processor B Failure
S22	AB Duplex, Interstage B Failure
S23	A Simplex, BC Offline
S24	B Simplex, AC Offline
S25	C Simplex, AB Offline
S26	Fail Catastrophic

The basis for the state transition rates are given by the following 5 terms:

L1=91.8 x 10^{-6} failures/hour	Failure Rate of Channel A Processor (λ_{pa})
L2=94.1 x 10^{-6} failures/hour	Failure Rate of Channel B Processor (λ_{pb})

L3=90.1 x 10^{-6} failures/hour	Failure Rate of Channel C Processor (λ_{pc})
L4=1.5 x 10^{-6} failures/hour	Failure Rate of Any Interstage (λ_i)
L5=9.0 x 10^{+4} IDs/hour	Software Fault Identification and Reconfiguration Rate (μ)

The following are the state transition rates that were used in the Markov model for the ERV FTP triplex Core.

TRANSITION	RATE	DESCRIPTION
1→2	L1	Processor A Fails
1→3	L4	Interstage A Fails
1→4	L2	Processor B Fails
1→5	L4	Interstage B Fails
1→6	L3	Processor C Fails
1→7	L4	Interstage C Fails
2→8	L5	System Reconfigures to BC Duplex
3→8	L5	System Reconfigures to BC Duplex
4→9	L5	System Reconfigures to AC Duplex
5→9	L5	System Reconfigures to AC Duplex
6→10	L5	System Reconfigures to AB Duplex
7→10	L5	System Reconfigures to AB Duplex
2→26	L2+L3+3*L4	Second Failure While Reconfiguring - System Loss
3→26	L1+L2+L3+2*L4	Second Failure While Reconfiguring - System Loss
4→26	L1+L3+3*L4	Second Failure While Reconfiguring - System Loss
5→26	L1+L2+L3+2*L4	Second Failure While Reconfiguring - System Loss
6→26	L1+L2+3*L4	Second Failure While Reconfiguring - System Loss
7→26	L1+L2+L3+2*L4	Second Failure While Reconfiguring -

		System Loss
8→11	L2	Duplex BC, Processor B Fails
8→12	L4	Duplex BC, Interstage B Fails
8→13	L3	Duplex BC, Processor C Fails
8→14	L4	Duplex BC, Interstage C Fails
9→15	L1	Duplex AC, Processor A Fails
9→16	L4	Duplex AC, Interstage A Fails
9→17	L3	Duplex AC, Processor C Fails
9→18	L4	Duplex AC, Interstage C Fails
10→19	L1	Duplex AB, Processor A Fails
10→20	L4	Duplex AB, Interstage A Fails
10→21	L2	Duplex AB, Processor B Fails
10→22	L4	Duplex AB, Interstage B Fails
11→25	0.95*L5	System Reconfigures to Simplex C
12→25	0.95*L5	System Reconfigures to Simplex C
13→24	0.95*L5	System Reconfigures to Simplex B
14→24	0.95*L5	System Reconfigures to Simplex B
15→25	0.95*L5	System Reconfigures to Simplex C
16→25	0.95*L5	System Reconfigures to Simplex C
17→23	0.95*L5	System Reconfigures to Simplex A
18→23	0.95*L5	System Reconfigures to Simplex A
19→24	0.95*L5	System Reconfigures to Simplex B
20→24	0.95*L5	System Reconfigures to Simplex B
21→23	0.95*L5	System Reconfigures to Simplex A
22→23	0.95*L5	System Reconfigures to Simplex A
11→26	L3+2*L4+.05*L5	3rd Failure or Unable to Downgrade
12→26	L2+L3+L4+.05*L5	3rd Failure or Unable to Downgrade
13→26	L2+2*L4+.05*L5	3rd Failure or Unable to Downgrade

14→26	L2+L3+L4+.05*L5	3rd Failure or Unable to Downgrade
15→26	L3+2*L4+.05*L5	3rd Failure or Unable to Downgrade
16→26	L1+L3+L4+.05*L5	3rd Failure or Unable to Downgrade
17→26	L1+2*L4+.05*L5	3rd Failure or Unable to Downgrade
18→26	L1+L3+L4+.05*L5	3rd Failure or Unable to Downgrade
19→26	L2+2*L4+.05*L5	3rd Failure or Unable to Downgrade
20→26	L1+L2+L4+.05*L5	3rd Failure or Unable to Downgrade
21→26	L1+2*L4+.05*L5	3rd Failure or Unable to Downgrade
22→26	L1+L2+L4+.05*L5	3rd Failure or Unable to Downgrade
23→26	L1	Simplex A, Processor A Fails
24→26	L2	Simplex B, Processor B Fails
25→26	L3	Simplex C, Processor C Fails

APPENDIX C: MARK 1 SOURCE CODE

The following is the source code for the reliability evaluation of the ERV flight computer using the Mark 1 Markov modeling package:

```
TITLE--ERV RMA MODEL - BASELINE EXACT (ERV21D) 5/18/87
M1=26--TRIPLEX WITH INTER-STAGES
S1=1--NOTAILURES
S2=0--PROCESSOR A FAILURE
S3=0--INTERSTAGE A FAILURE
S4=0--PROCESSOR B FAILURE
S5=0--INTERSTAGE B FAILURE
S6=0--PROCESSOR C FAILURE
S7=0--INTERSTAGE C FAILURE
S8=0--BC DUPLEX, A OFFLINE
S9=0--AC DUPLEX, B OFFLINE
S10=0--AB DUPLEX, C OFFLINE
S11=0--BC DUPLEX, PROCESSOR B FAILURE
S12=0--BC DUPLEX, INTERSTAGE B FAILURE
S13=0--BC DUPLEX, PROCESSOR C FAILURE
S14=0--BC DUPLEX, INTERSTAGE C FAILURE
S15=0--AC DUPLEX, PROCESSOR A FAILURE
S16=0--AC DUPLEX, INTERSTAGE A FAILURE
S17=0--AC DUPLEX, PROCESSOR C FAILURE
S18=0--AC DUPLEX, INTERSTAGE C FAILURE
S19=0--AB DUPLEX, PROCESSOR A FAILURE
S20=0--AB DUPLEX, INTERSTAGE A FAILURE
S21=0--AB DUPLEX, PROCESSOR B FAILURE
S22=0--AB DUPLEX, INTERSTAGE B FAILURE
S23=0--A SIMPLEX, BC OFFLINE
S24=0--B SIMPLEX, AC OFFLINE
S25=0--C SIMPLEX, AB OFFLINE
S26=0--FAIL CATASTROPHIC
L1=91.8E-6--FAILURE RATE OF PROC A
L2=94.1E-6--FAILURE RATE OF PROC B
L3=90.1E-6--FAILURE RATE OF PROC C
L4=1.5E-6--FAILURE RATE OF INTER-STAGE
L5=9.0E+4--RECOVERY RATE
L6=.95--PERCENTAGE OF TIME 2ND FAILURE IS RECOVERABLE
T1>2=L1--PROCESSOR A FAILS
T1>3=L4--INTERSTAGE A FAILS
T1>4=L2--PROCESSOR B FAILS
T1>5=L4--INTERSTAGE B FAILS
T1>6=L3--PROCESSOR C FAILS
T1>7=L4--INTERSTAGE C FAILS
T2>8=L5--SYSTEM RECONFIGURES TO BC DUPLEX
T3>8=L5--SYSTEM RECONFIGURES TO BC DUPLEX
T4>9=L5--SYSTEM RECONFIGURES TO AC DUPLEX
T5>9=L5--SYSTEM RECONFIGURES TO AC DUPLEX
T6>10=L5--SYSTEM RECONFIGURES TO AB DUPLEX
T7>10=L5--SYSTEM RECONFIGURES TO AB DUPLEX
```

```
T2>26=L2+L3+3*L4--SECOND FAILURE WHILE RECONFIGURING - SYSTEM LOSS
T3>26=L1+L2+L3+2*L4--SECOND FAILURE WHILE RECONFIGURING - SYSTEM LOSS
T4>26=L1+L3+3*L4--SECOND FAILURE WHILE RECONFIGURING - SYSTEM LOSS
T5>26=L1+L2+L3+2*L4--SECOND FAILURE WHILE RECONFIGURING - SYSTEM LOSS
T6>26=L1+L2+3*L4--SECOND FAILURE WHILE RECONFIGURING - SYSTEM LOSS
T7>26=L1+L2+L3+2*L4--SECOND FAILURE WHILE RECONFIGURING - SYSTEM LOSS
T8>11=L2--DUPLEX BC, PROCESSOR B FAILS
T8>12=L4--DUPLEX BC, INTERSTAGE B FAILS
T8>13=L3--DUPLEX BC, PROCESSOR C FAILS
T8>14=L4--DUPLEX BC, INTERSTAGE C FAILS
T9>15=L1--DUPLEX AC, PROCESSOR A FAILS
T9>16=L4--DUPLEX AC, INTERSTAGE A FAILS
T9>17=L3--DUPLEX AC, PROCESSOR C FAILS
T9>18=L4--DUPLEX AC, INTERSTAGE C FAILS
T10>19=L1--DUPLEX AB, PROCESSOR A FAILS
T10>20=L4--DUPLEX AB, INTERSTAGE A FAILS
T10>21=L2--DUPLEX AB, PROCESSOR B FAILS
T10>22=L4--DUPLEX AB, INTERSTAGE B FAILS
T11>25=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX C
T12>25=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX C
T13>24=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX B
T14>24=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX B
T15>25=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX C
T16>25=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX C
T17>23=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX A
T18>23=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX A
T19>24=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX B
T20>24=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX B
T21>23=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX A
T22>23=L6*L5--SYSTEM RECONFIGURES TO SIMPLEX A
T11>26=L3+2*L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T12>26=L2+L3+L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T13>26=L2+2*L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T14>26=L2+L3+L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T15>26=L3+2*L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T16>26=L1+L3+L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T17>26=L1+2*L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T18>26=L1+L3+L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T19>26=L2+2*L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T20>26=L1+L2+L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T21>26=L1+2*L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T22>26=L1+L2+L4+(1-L6)*L5--3RD FAILURE OR UNABLE TO DOWNGRADE
T23>26=L1--SIMPLEX A, PROCESSOR A FAILS
T24>26=L2--SIMPLEX B, PROCESSOR B FAILS
T25>26=L3--SIMPLEX C, PROCESSOR C FAILS
M2=3--THIS MODEL IS FOR WHOLE BUS FAILURES
S1=1--XX
S2=0--XX
S3=0--XX
L1=0.6E-6--FAILURE RATE OF BUS CONTROLLER
T1>2=3*L1--ONE BUS FAILS
T2>3=2*L1--SECOND BUS FAILS - BUSS SYSTEM LOSS
M3=3--THIS MODEL IS FOR DUPLEX DEVICE INTERFACES
S1=1--XX
```

```
S2=0--XX
S3=0--XX
L1=6.1E-6--FAILURE RATE OF DEVICE INTERFACE
T1>2=2*L1--ONE DEVICE FAILS
T2>3=L1--SECOND DEVICE FAILS - DEVICE LOSS
M4=4--THIS MODEL IS FOR TRIPLEX DEVICE INTERFACES
S1=1--XX
S2=0--XX
S3=0--XX
S4=0--XX
L1=6.1E-6--FAILURE RATE OF DEVICE INTERFACE
T1>2=3*L1--ONE DEVICE FAILS
T2>3=2*L1--SECOND DEVICE FAILS
T3>4=L1--TRIPLEX DEVICE LOSS
M5=2--THIS MODEL IS FOR TRIPLEX DEVICE INTERFACES
S1=1--XX
S2=0--XX
L1=1E-9--FAILURE RATE OF ANOTHER CRITICAL DEVICE
T1>2=10*L1--ONE CRITICAL DEVICE FAILS
RUN 0 1E5
YMIN=-14
F1=S1.26--PROB(S26) - TRIPLEX ATTRITION LOSS
F2=S3.3+S4.4-S3.3*S4.4--PIF
F3=1-S3.2--THIS IS (1-P1)
F4=1-S4.3--THIS IS (1-P5)
F5=1-(F3*F3*F3*F3*F3*F3*F3)--THIS IS 1-(1-P1)^N
F6=1-(F4*F4*F4*F4*F4*F4)--THIS IS 1-(1-P5)^M
F7=F5+F6-F5*F6--THIS IS PDISF UNION PTIDF
F8=S2.2--THIS IS PSBF
F9=S2.3--THIS IS PDBF
F10=F9+F8*F7+F2-F9*F2-F8*F7*F2--THIS IS PIOL
F11=(S1.23+S1.24+S1.25)*F8--PS/B
F12=F11+F1--PTCL
F13=F10+F12-F10*F12--PVCL=PTCL UNION PIOL
F14=S5.2--THIS IS PCDL
F15=F13+F14-F13*F14--PVL=PVCL UNION PCDL
PLOT F12--PROBABILITY OF TRIPLEX CORE LOSS (PTCL)
PLOT F10--PROBABILITY OF I/O LOSS (PIOL)
PLOT F15--PROBABILITY OF VEHICLE LOSS
PLOT F2--PIF
PLOT F7--PDISF U PTIDF
PLOT F8--PSBF
PLOT F9--PDBF
PLOT F11--PROB(LOSS IN SIMPLEX MODE)
PLOT F13--PROB(VEHICLE CONTROL LOSS)
PLOT F14--PCDL
END
```

APPENDIX D: SURE RESULTS FOR MARKOV MODELS

So that the reliability of the ERV could be determined as accurately as possible, the Markov models for this system were evaluated by both the MARK 1 and SURE Reliability Analysis programs. The results of both programs were virtually identical; the few differences in output can be attributed to roundoff errors.

The following is the input file for the SURE program and the results obtained using the it.

D.1 SURE INPUT FILE

We utilized one input file to run SURE: it contains the model describing the probability of an FTP core loss due strictly to an FTP failure (the probability of being in State 26 of the model described in Section 7.). Because of the complexity of the true FTP core loss model, SURE could not be used to calculate the real probability of the FTP core loss which includes the probability of the FTP operating in simplex with a double bus failure.

```
(*****************************************************************)
(* File: FTPCORE.MOD                                             *)
(*  Author:John Esielionis                                       *)
(*  Facility: The Charles Stark Draper Laboratory                *)
(*  Created:  01/20/87                                           *)
(*  Last Mod: 04/04/87                                           *)
(*  Purpose:  This file contains the model describing the        *)
(*probability of an FTP core loss.                               *)
(*****************************************************************)
 L1 = 91.8E-6;                                  (* Failure rate of Processor A *)
 L2 = 94.1E-6;                                  (* Failure rate of Processor B *)
 L3 = 90.8E-6;                                  (* Failure rate of Processor C *)
 L4 = 1.5E-6;                                   (* Failure rate of Interstage *)
 L5 = 9.0E+4;                                   (* Recovery rate of Interstage *)
 L6 = .95;                                      (* Percent of time 2nd fail is recoverable *)
 1, 2 = L1;                                     (* Processor A fails *)
 1, 3 = L4;                                     (* Interstage A fails *)
 1, 4 = L2;                                     (* Processor B fails *)
 1, 5 = L4;                                     (* Interstage B fails *)
 1, 6 = L3;                                     (* Processor C fails *)
 1, 7 = L4;                                     (* Interstage C fails *)
 2, 8 = L5;                                     (* System reconfigures to BC duplex *)
 3, 8 = L5;                                     (* System reconfigures to BC duplex *)
 4, 9 = L5;                                     (* System reconfigures to AC duplex *)
 5, 9 = L5;                                     (* System reconfigures to AC duplex *)
 6, 10 = L5;                                    (* System reconfigures to AB duplex *)
```

```
7, 10 = L5;                                          (* System reconfigures to AB duplex *)
2, 26 = L2 + L3 + 3*L4;(                           (* 2nd fail while reconfiguring, system loss *)
3, 26 = L1 + L2 + L3 + 2*L4;                       (* 2nd fail while reconfiguring, system loss *)
4, 26 = L1 + L3 + 3*L4;                            (* 2nd fail while reconfiguring, system loss *)
5, 26 = L1 + L2 + L3 + 2*L4;                       (* 2nd fail while reconfiguring, system loss *)
6, 26 = L1 + L2 + 3*L4;                            (* 2nd fail while reconfiguring, system loss *)
7, 26 = L1 + L2 + L3 + 2*L4;                       (* 2nd fail while reconfiguring, system loss *)
8, 11 = L2;                                                  (* Duplex BC, Processor B fails *)
8, 12 = L4;                                                  (* Duplex BC, Interstage B fails *)
8, 13 = L3;                                                  (* Duplex BC, Processor C fails *)
8, 14 = L4;                                                  (* Duplex BC, Interstage C fails *)
9, 15 = L1;                                                  (* Duplex AC, Processor A fails *)
9, 16 = L4;                                                  (* Duplex AC, Interstage A fails *)
9, 17 = L3;                                                  (* Duplex AC, Processor C fails *)
9, 18 = L4;                                                  (* Duplex AC, Interstage C fails *)
10, 19 = L1;                                                 (* Duplex AB, Processor A fails *)
10, 20 = L4;                                                 (* Duplex AB, Interstage A fails *)
10, 21 = L2;                                                 (* Duplex AB, Processor B fails *)
10, 22 = L4;                                                 (* Duplex AB, Interstage B fails *)
11, 25 = L6 * L5;                                    (* System reconfigures to simplex C *)
12, 25 = L6 * L5;                                    (* System reconfigures to simplex C *)
13, 24 = L6 * L5;                                    (* System reconfigures to simplex B *)
14, 24 = L6 * L5;                                    (* System reconfigures to simplex B *)
15, 25 = L6 * L5;                                    (* System reconfigures to simplex C *)
16, 25 = L6 * L5;                                    (* System reconfigures to simplex C *)
17, 23 = L6 * L5;                                    (* System reconfigures to simplex A *)
18, 23 = L6 * L5;                                    (* System reconfigures to simplex A *)
19, 24 = L6 * L5;                                    (* System reconfigures to simplex B *)
20, 24 = L6 * L5;                                    (* System reconfigures to simplex B *)
21, 23 = L6 * L5;                                    (* System reconfigures to simplex A *)
22, 23 = L6 * L5;                                    (* System reconfigures to simplex A *)
11, 26 = L3 + 2*L4 + (1-L6)*L5;                     (* 3rd failure or unable to downgrade *)
12, 26 = L2 + L3 + L4 + (1-L6)*L5;                  (* 3rd failure or unable to downgrade *)
13, 26 = L2 + 2*L4 + (1-L6)*L5;                     (* 3rd failure or unable to downgrade *)
14, 26 = L2 + L3 + L4 + (1-L6)*L5;                  (* 3rd failure or unable to downgrade *)
15, 26 = L3 + 2*L4 + (1-L6)*L5;                     (* 3rd failure or unable to downgrade *)
16, 26 = L1 + L3 + L4 + (1-L6)*L5;                  (* 3rd failure or unable to downgrade *)
17, 26 = L1 + 2*L4 + (1-L6)*L5;                     (* 3rd failure or unable to downgrade *)
18, 26 = L1 + L3 + L4 + (1-L6)*L5;                  (* 3rd failure or unable to downgrade *)
19, 26 = L2 + 2*L4 + (1-L6)*L5;                     (* 3rd failure or unable to downgrade *)
20, 26 = L1 + L2 + L4 + (1-L6)*L5;                  (* 3rd failure or unable to downgrade *)
21, 26 = L1 + 2*L4 + (1-L6)*L5;                     (* 3rd failure or unable to downgrade *)
22, 26 = L1 + L2 + L4 + (1-L6)*L5;                  (* 3rd failure or unable to downgrade *)
23, 26 = L1;                                         (* Processor A fails or unable to downgrade *)
24, 26 = L2;                                         (* Processor B fails or unable to downgrade *)
25, 26 = L3;                                         (* Processor C fails or unable to downgrade *)
etcalc = 1;
list = 2;
time = 0.01 to* 100000 by 10;
```

D.2 SURE RESULTS

The following graph depicts the results obtained through the SURE Reliability

Analysis Program. It represents the output for the FTP failure model.

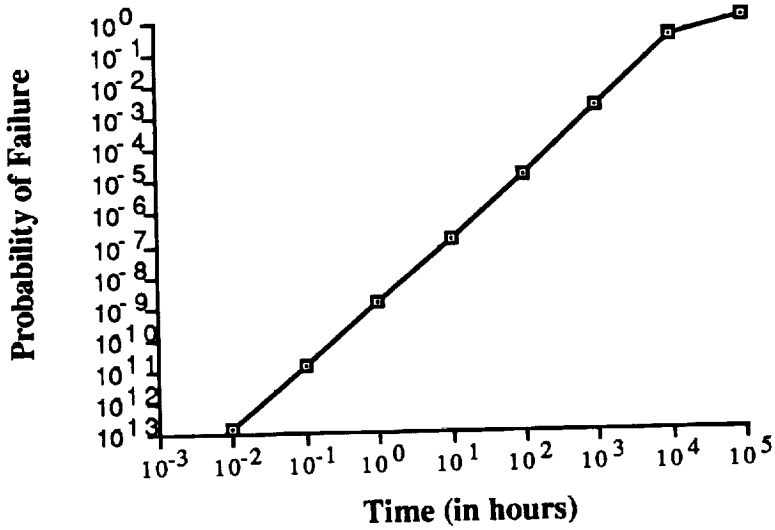

Figure D-1. Probability of an FTP Failure

D.3 COMPARISON OF SURE AND MARK 1 RESULTS

Figure D-2 shows a comparison of the SURE and MARK 1 results, which for all practical purposes are identical. In fact, the results graph over each other.

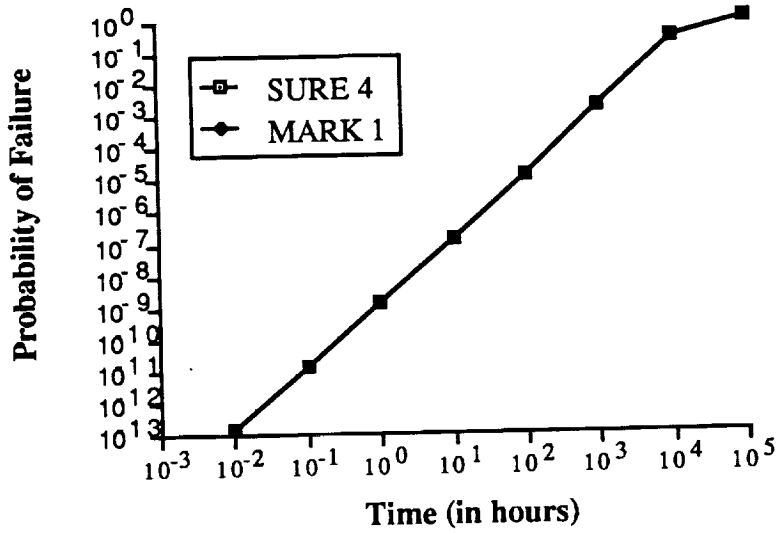

Figure D-2. SURE vs. MARK 1 FTP Failure Probability

APPENDIX D: SURE RESULTS FOR MARKOV MODELS

APPENDIX E: LIST OF ABBREVIATIONS

1553	MIL-STD-1553B Digital Time Division Data Bus (Rev B)
1750	MIL-STD-1750A Instruction Set Architecture (Rev A)
A/D	Analog/Digital
ADC	Analog-to-Digital Converter
AIPS	Advanced Information Processing System
AIRLAB	NASA Langley's Avionics Integration Research Laboratory
AS	Address Strobe
Axx	Address line xx
BIU	Bus Interface Unit
CB	Channel Bus
CDL	Critical Device Loss
CHMOS	Complementary Hybrid Metal Oxide Silicon
CMOS	Complementary Metal Oxide Silicon
CP	Computational Processor
CPU	Central Processing Unit
CPUCLK	Clock signal driving a CPU
CSDL	Charles Stark Draper Laboratory
D/A	Digital/Analog
D/R	Driver/Receiver
DAC	Digital-to-Analog Converter
DAIS	Digital Avionics Instruction Set
DBF	Double Bus Failure
DIP	Dual In-line Pin
DISF	Double Interface Single Failure
DoD	Department of Defense
DPU	Data Processing Unit
DTC	Data Transfer Complete
Dxx	Data line xx
ECS	Environmental Control System
ERV	Entry Research Vehicle
FCR	Fault Containment Region
FDIR	Fault Detection, Isolation, and Reconfiguration
FLOPS	Floating Point Operations Per Second
fph	failures per hour
FPU (aka FPCP)	Floating Point Unit
FTC	Fault-Tolerant Clock
FTP	Fault-Tolerant Processor
G	10^9
GN&C	Guidance, Navigation, and Control
GPS	Global Positioning System
GSE	Ground Support Electronics
I/O	Input/Output
ICB	InterChannel Bus
IEEE	Institute of Electrical and Electronics Engineers
IF	Interface Failure
IFTC	Interstage Fault-Tolerant Clock
IMU	Inertial Measuring Unit
IOB	Input/Output Bus
IOL	Input/Output Loss
IOP	Input/Output Processor
IPS	Instructions Per Second
ISA	Instruction Set Architecture
K	10^3
LaRC	Langley Research Center
LINS	Laser Inertial Navigation System

APPENDIX E: LIST OF ABBREVIATIONS

λ_X	Failure rate for device X
MLS	Microwave Landing System
MMU	Memory Management Unit
MPU	Microprocessing Unit
ms	millisecond
NASA	National Aeronautics and Space Administration
navaid	navigation aid
NMOS	n-channel Metal Oxide Silicon
ns	nanosecond
PAL®	Programmable Array Logic
PC	Printed Circuit
PGA	Pin Grid Array
PISO	Parallel In Serial Out
PLD	Programmable Logic Device
PROM	Programmable Read Only Memory
P_X	Probability of event X occuring or probability of being in state X
R-W	Read-Write
RAM	Random Access Memory
RCS	Reaction Control System
ROM	Read Only Memory
RTI	Remote Terminal Interface
S/B	Simplex Mode Bus Failure
SBF	Single Bus Failure
SEADS	Shuttle Entry Air Data System
SECDED	Single Error Correction, Double Error Detection
SIP	Single In-line Pin
SIPO	Serial In Parallel Out
SRAM	Static Random Acess Memory
STS	Space Transportation System
S_X	State X
TCL	Triplex Core Loss
TIDF	Triple Interface Double Failure
TTL	Transistor-Transistor Logic
VHSIC	Very High Speed Integrated Circuit
VL	Vehicle Loss
VLSI	Very Large Scale Integration
WSG	Wait-State Generator
μ	Fault Recovery Rate
μs	microsecond

NASA National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
181828		

4. Title and Subtitle
Avionics Architecture Studies for the Entry Research Vehicle

5. Report Date
May 1989

6. Performing Organization Code

7. Author(s)
M.J. Dzwonczyk, M.F. McKinney, S.J. Adams, R.J. Gauthier

8. Performing Organization Report No.

10. Work Unit No.
506-49-11-01

9. Performing Organization Name and Address
The Charles Stark Draper Laboratory, Inc.
555 Technology Square
Cambridge, MA 02139

11. Contract or Grant No.
NAS1-18061

13. Type of Report and Period Covered
8/86-7/87

12. Sponsoring Agency Name and Address
NASA
Langley Research Center
Hampton, VA 23665

14. Sponsoring Agency Code

15. Supplementary Notes
NASA Technical Monitor: Howard Stone, Space Systems Division, Langley Research Center

Summary paper to appear in Proceedings of the AIAA-IEEE Eighth Digital Avionics System Conference, October 1988.

16. Abstract

This report is the culmination of a year-long investigation of the avionics architecture for NASA's Entry Research Vehicle. The Entry Research Vehicle is conceived to be an unmanned, autonomous spacecraft to be deployed from the Shuttle. It will perform various aerodynamic and propulsive maneuvers in orbit and land at Edwards AFB after a 5-10 hour mission. Preliminary study of the vehicle's avionic architecture was performed by CSDL under Task Assignment Number 1 of the same contract in the spring of 1986. This follow-on study detailed the design and furthered the analysis of the architecture. This report describes in detail the avionics architecture proposed by CSDL and its subsequent reliability analysis. The architecture consists of a central triply redundant ultra-reliable fault tolerant processor attached to three replicated and distributed MIL-STD-1553 buses for I/O. It was found to be sufficiently reliable for the ERV mission plan.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
digital avionics architectures fault-tolerant computers highly reliable systems Entry Research Vehicle	Unclassified-Unlimited Subject Category 62

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	136 pages	

NASA FORM 1626 OCT 86