

Development of an Open Architecture  
Flight-Qualified Computer  
(CDDF Project No. 92-R07)

---

*B. Beabout*





# Development of an Open Architecture Flight-Qualified Computer (CDDF Project No. 92-R07)

---

*B. Beabout  
Marshall Space Flight Center • MSFC, Alabama*



## ACKNOWLEDGMENTS

The author would like to thank the following computer hardware designers.

Thomas Asquith	High Rate Multiplexor Interface Module, <i>Spacelab</i> Interface Agent, and Serial Input/Output Agent
David Hyde	Multibus II™ System Architecture Firmware
Donna Kaukler	Central Services Module
Fred Kroeger	CPU Agent and Test Software
Russ Lakin	Interconnect Microcontroller Firmware
Leslie McClendon	Ground Support Equipment
Kosta Varnavas	Remote Acquisition Interface Module, <i>Spacelab</i> Interface Agent



## TABLE OF CONTENTS

	Page
I. INTRODUCTION .....	1
II. PRELIMINARY STUDIES .....	1
III. DEVELOPMENT .....	4
A. Design Guidelines .....	4
B. Requirements Definition .....	5
IV. HARDWARE ARCHITECTURE .....	5
A. Central Services Module .....	7
B. Interconnect Module .....	8
C. CPU Agent .....	9
D. 1553 Agent .....	11
E. <i>Spacelab</i> Interface Agent .....	12
V. FABRICATION AND TESTING .....	14
VI. FOLLOW-ON EFFORTS .....	14
VII. CONCLUSIONS .....	15
REFERENCES .....	16

## LIST OF ILLUSTRATIONS

Figure	Title	Page
1.	System overview .....	6
2.	CSM .....	7
3.	Interconnect module.....	9
4.	CPU agent .....	10
5.	1553 agent .....	12
6.	SIA .....	13

## STANDARD ABBREVIATIONS

ANSI	American National Standards Institute
ASIC	application specific integrated circuit
BIST	built-in self test
BTL	bus transceiver logic
CISC	complex instruction set computer
CPU	central processing unit
EDAC	error detection and correction
EEPROM	electrically erasable programmable read only memory
FIFO	first in first out
GMT	Greenwich Mean Time
IEEE	Institute of Electrical and Electronics Engineers
I/O	input/output
IC	integrated circuit
OS	operating system
PLD	programmable logic device
RISC	reduced instruction set computer
SEU	single event upset
SRAM	static random access memory
TTL	transistor-transistor logic

## NONSTANDARD ABBREVIATIONS

AR&C	automatic rendezvous and capture
HRM	high rate multiplexor
HRMI	high rate multiplexor interface
MPC	message passing coprocessor
OBC	onboard computer
PSB	parallel system bus
RAU	remote acquisition unit
RAUI	remote acquisition unit interface
SIA	<i>Spacelab</i> interface agent
SIO	serial I/O
TMI	timer module interface

## UNUSUAL TERMS

The following terms are used to describe entities and operation of the architecture of the system design in this report.

AGENT	An intelligent board in a Multibus* II™ system.
I/O SPACE	The address space used for I/O references.
INTERCONNECT SPACE	The functional entity and address space used for configuration control and initialization in a Multibus II system.
MEMORY SPACE	The address space on the PSB used for traditional memory references.
MESSAGE SPACE	The address space used for solicited and unsolicited messages.
MODULE	A subsystem on an agent.

---

\* Multibus is a registered trade mark of Intel Corporation



## TECHNICAL MEMORANDUM

### DEVELOPMENT OF AN OPEN ARCHITECTURE FLIGHT QUALIFIED COMPUTER

(Center Director's Discretionary Fund Final Report, Project Number 92-R07)

#### I. INTRODUCTION

Experiment control and data collection are often accomplished using a dedicated computer system. Conventionally, purpose-built, custom-designed, computer systems have been developed for each experiment. While sometimes unique requirements may dictate such an approach, time and cost constraints prohibit such approaches. To reduce the hardware development cycle, a set of core modules providing generic capabilities could be developed. Once these modules are developed, they could be used in differing configurations as a basis of an experiment data system. Unique elements for specific experiments would be developed as appropriate. To reduce development time, it is desirable to leverage off commercially available standardized open architecture hardware and software. This report will cover the selection of a backplane architecture, selection of a central processing unit (CPU) module micro-processor, development of the core processing capability, and development of two general-purpose spacecraft interfaces (to *Spacelab* and the space station) and lessons learned.

The development of an experiment computer hardware cannot be done without proper consideration for application software development that must follow. To reduce the software development cost, consideration must be given for the development environment, tools, availability of appropriate run time environment, and the possibility of a functionally equivalent commercial hardware that is similar to the experiment computer.

Note, the experiment computer under development is prototyped using commercial grade parts on wire wrap boards and hence is not actually "flight qualified"; however, all parts used have equivalent MIL-STD-883 versions and can be directly implemented on printed circuit boards to build a fully flight-qualified system.

Using these guidelines, a set of decision criteria was developed.

#### II. PRELIMINARY STUDIES

Two major studies were done in preparation for design of this experiment controller. The first was to select an open system standard backplane architecture. The second selected the primary computational processor family. The decision algorithm used in these studies was the analytic hierarchy process.

The candidates considered for the backplane architecture were:

1. Versa Module Eurocard bus (VMEbus™)† (IEEE Std 1014) a 32/64 bit asynchronous backplane.
2. Futurebus+ (IEEE Std 896) a 32/64/128 bit backplane architecture using BTL transceiver technology and having multiple profile definitions.

---

† VMEbus is a registered trademark of Motorola Incorporated.

3. Multibus II (IEEE Std 1296) a 32-bit synchronous backplane with four distinct address spaces.

Other backplane architectures, such as Nubus, Standard 32 (Std 32), and Extended Industry Standard Architecture (EISA) were reviewed for consideration in this study, but were not included for various reasons. These reasons include, unsuitable parts for flight, limited development tools, and unsuitability for real-time applications.

The VMEbus (IEEE Std 1014) is a 32-bit asynchronous bus, which has a 32-bit address bus and a 16- or 32-bit data bus. The address and data buses are not multiplexed in either the 16- or the 32-bit data bus configurations. It has a theoretical maximum transfer rate of 40 Mbytes per second (for the 32-bit bus). The actual data transfer rate depends on the response times of the interacting boards, which can vary depending of the individual board implementations. Hence, the bus is referred to as an asynchronous bus. The VMEbus specification has only one bus defined. Using a one connector "P1," the data bus width is limited to 16 bits; using an additional 96 pin "P2" connector, the data bus is expanded to 32 bits. There is a 64-bit implementation of the VME bus referred to as VME64. The 64-bit implementation multiplexes the additional 32 bits of data onto the 32 address bits after the address is latched. At the time of the backplane study, the 64-bit data bus VME64 standard was not released and, hence, its additional capabilities was not included in the scoring of the VMEbus standard. On the "P2" connector, there are 64 undefined signals. Note, however, that if the two-connector configuration is used, signals to both connectors need to span the entire backplane. There is no parity for either the data or control signals on the backplane. The central arbiter for the backplane must reside in slot 0. There are seven interrupt lines that can be daisy chained to provide prioritized interrupt services to various cards in the system. There is an additional IEEE standard for mechanical core specification for conduction-cooled Eurocards (IEEE Std 1101.2) for the VMEbus architecture. The primary microprocessor family used in commercial VMEbus systems is the Motorola\*\* 680x0 series. VMEbus has many advantages: a large commercial and military user base, a standardized conduction-cooled card definition, and a wide selection of development tools; however, the lack of parity on its data and control lines causes concern in an environment subject to single event upsets (SEU).

The Futurebus+ (IEEE Std 896) is a newer open standard bus architecture. It is an asynchronous 32-, 64-, 128-, or 256-bit wide bus with data transfer rates from 80 Mbytes per second to 3.2 Gbytes per second. The electrical specifications on the backplane are specifically designed for the bus transceiver logic (BTL) levels. This allows high-current, low-voltage transitions between logic levels. The low-voltage level transitions reduce the effects of stray capacitance, and the high-current drive capability of BTL allows for incident wave switching of logic states. In conventional TTL level backplane signal transmission, the relatively high-voltage transitions and lower drive current necessitates longer settling time, hence, TTL level backplanes have a lower theoretical bandwidth. At the time of the study, these devices were just being released as commercial products, and no release dates for MIL-STD-883 parts had been announced. Futurebus+ has multiple profile definitions to tailor the architecture for differing uses, such as an I/O bus, a multiprocessor computer system, and others. At the time of the study, the IEEE Std 896 had not been finalized and profiles were not fully developed. Hence, no chip sets had been developed to implement the backplane interfaces and protocols. Therefore, Futurebus+ scored low because of lack of commercially available products and maturity, not because of deficiencies in innate capabilities.

The Multibus II parallel system bus (PSB) is specified by IEEE Std 1296. This standard defines electrical, mechanical, and message passing protocol specifications. There are other bus specifications encompassed by Multibus II, such as the local bus extension (iLBX™)‡, that are not included in the IEEE 1296 specification. Also, there is an optional transaction-oriented information exchange protocol definition, the Multibus II transport protocol. The mechanical specifications for Multibus II include a two-96-pin connector backplane configuration. One of these connectors (P1) is used to implement the PSB, the second connector (P2) is free for user definition or can be used to implement the iLBX bus

---

\*\* Motorola is a registered trademark of Motorola Incorporated.

‡ iLBX is a registered trademark of Intel Corporation.

specification. The PSB defined in IEEE Std 1296 is a 32-bit synchronous bus with 32 bits of data multiplexed with 32 bits of address. The timing of control and data access to the bus is specified in respect to a system-wide clock, hence, the bus is a synchronous bus. The clock rate is 10 MHz. This sets the bus bandwidth at 40 Mbyte per second. A 20-MHz interface chip is being developed commercially and will be available in MIL-STD-883 form. This would double the bandwidth of the bus. The arbitration for access to the bus is distributed, hence, there is no designated bus master. There is parity on the data/address signals and parity on bus control signals. There are four address spaces defined for the PSB, memory, I/O, message, and interconnect. The memory and I/O spaces are conventional extensions to a processor's memory and I/O address spaces. Message space is similar to a local area network in function. In message space, information is exchanged by passing message packets between sockets. A socket is a unique identifier consisting of a port ID and a slot ID. A slot ID defines, to the hardware, the location of the board on the backplane and the port ID. The port ID is a software entity defining which task receives or transmits the data. Multibus II has a relatively large user base. The primary processor family used in commercial boards is the Intel 80x86 series, though the architecture is processor independent. The synchronous nature of the backplane prevents a slow module in the system from occupying large time slots for data transfers that could occur in an asynchronous implementation. Instead, the data are buffered on a module and burst across the backplane at the maximum bandwidth of the bus. The protocol chip that interfaces to the backplane also supports automatic retransmission of messages in the event of an error.

The final scores (out of a possible score of 1) were:

- |                |      |
|----------------|------|
| 1. Multibus II | 0.41 |
| 2. VMEbus      | 0.32 |
| 3. Futurebus+  | 0.27 |

These results indicate that Multibus II best meets the selected criteria. These results reflect data and hardware available at the time of the study, also differing requirements would change criteria's weightings that could change the results. The study was conducted June through August 1991. Note that a more stringent radiation environment would exclude all of the candidate architectures without the development of custom radiation tolerant ASIC's, which is beyond the scope of this development.

The processor study compared the following candidates:

1. Intel™§ 80386DX a 32-bit CISC processor.
2. Intel 80486DX a 32-bit CISC processor with an integrated floating point unit and 8 kbytes of memory and data cache.
3. IDT™§§ R3051 a 32-bit embedded RISC integer processor with a 4-kbytes instruction cache.
4. Intel 80960MX a 32-bit RISC processor with integrated floating point unit and 2-kbyte instruction cache and 2-kbyte data cache.

The final scores were:

- |                  |      |
|------------------|------|
| 1. 80486DX       | 0.32 |
| 2. 80386DX/80387 | 0.20 |

---

§ Intel is a registered trademark of Intel Corporation.

§§ IDT is a registered trademark of Integrated Device Technology Incorporated.

- |          |      |
|----------|------|
| 3. 80960 | 0.25 |
| 4. R3051 | 0.23 |

The results indicated that the Intel 80486DX best met the selected criteria at the given weightings. Note, the results of this study are somewhat biased by the selection of the backplane architecture. While the backplane architecture itself is not processor specific, Intel was the developer of the Multibus II specifications and interface chip set. Hence, a larger base of commercial products and software are designed around the Intel processors for Multibus II systems. This affects scoring in relation to the software development tools, run time environment, and commercial Multibus II hardware for the development environment. Also, an advantage for the Intel CISC processors is the large installed base of personal computers that use them and relatively low-cost development tools for a PC. Another advantage is the availability of Intel's iRMX III™ real-time operating system that runs on both Multibus II platforms and PC's and is written exclusively for execution on a 80386 or 80486 processor. The primary advantage of the 80486 over the 80386 is its integrated floating point coprocessor, which saves board space. The IDT R3051 and 80960 scored lower because of costlier software tools and lack of a real-time operating system for the Multibus II environment. The performance benefits of the IDT R3051 RISC processor are somewhat mitigated by its incompatibility with the IDT R3010 floating point unit. Note, the IDT R3051 was selected over the IDT R3000 for trade since it is better suited for embedded applications.

### III. DEVELOPMENT

The major thrust of the development for this open-architecture, flight-qualified computer was to design and build wire wrap prototypes of core elements for an experiment computer. These core elements consist of a CPU agent, a *Spacelab* interface agent, and a MIL-STD-1553 bus interface agent. In addition, a central services module (CSM) was designed and built to provide the common bus clocks, bus time-out monitoring, and perform slot identification and priority initialization for the system as required by IEEE Std 1296. While the actual wire wrap prototypes are not flight qualified, all components used in their design are available as MIL-STD-883 parts. The use of fully screened parts for a prototyping is cost prohibitive.

#### A. Design Guidelines

Several design guidelines were established to promote robust designs and easier integration of the system components, and to simplify transition to a flight program. These guidelines are:

1. All memory will be EDAC protected with one-bit error correction and two-bit error detection per word.
2. All parts will be available as MIL-STD-883, or equivalent, parts.
3. All devices will have decoupling capacitors between power and ground located in such a way as to minimize inductance.
4. All nonvolatile memory will be programmable on board, since socketing of the devices is not desirable on flight boards.
5. The interconnect and message spaces of the IEEE Std 1296 parallel systems bus will be fully supported. The memory and I/O spaces are optional.

---

† iRMX III is a registered trademark of Intel Corporation.

6. The Multibus II system architecture (MSA) will be supported.
7. The Multibus II transport layer protocol will be supported.
8. Self test and diagnostic capabilities will be emphasized.
9. Power consumption will be minimized where possible.
10. Parts subject to latchup will not be used.

## B. Requirements Definition

Since this development was done outside the auspices of a particular program, no requirements were predefined by mission goals, orbital environment, or host spacecraft interfaces. Some assumptions had to be made in regard to these future requirements. *Spacelab* and space station were assumed as the most likely host platforms for this experiment computer. This allows assumptions to be made regarding environment and host interfaces. The primary data interface to the space station data management system is via a MIL-STD-1553 bus. The primary experiment interfaces to the *Spacelab* data management system are the remote acquisition unit (RAU) and high rate multiplexor (HRM). No throughput requirements or memory sizing requirements could be defined for the system without software sizing and timing estimates, and these estimates are driven by particular mission requirements. The usual orbits for *Spacelab* missions and the proposed orbit for the space station presents a relatively benign radiation environment. The associated radiation requirements for these platforms were used. The power and thermal control systems of both of these hosts alleviate tight power and thermal budgeting requirements; however, power management, thermal management, and weight are always concerns on any platform. Also, this computer is targeted for class C or class D experiments; hence, no requirements for redundancy internal to the computer are mandated.

Using these assumptions, design guidelines, past experience, and requirements of the Multibus II specification, four functionally distinct elements were defined and requirements were developed for each. The elements are the CSM, CPU agent, 1553 agent, and *Spacelab* interface agent (SIA).

## IV. HARDWARE ARCHITECTURE

In order to best meet the requirements defined and to provide the most flexibility, a loosely coupled multiprocessor approach was adopted. Each functional element has its own processor, hence freeing the primary processor, the CPU agent, for faster execution of application software. The SIA is responsible for formatting any outgoing data, routing any incoming commands and data to the appropriate subsystem, and processing low level I/O specific to its interfaces. The 1553 agent provides all necessary formatting and protocol-specific processing to interface with two MIL-STD-1553 buses. This simplifies the application software interfaces to either of these two subsystems that reside on the CPU agent. The protocol for communications between these intelligent subsystems, or agents, is defined by IEEE Std 1296 and the transport protocol specification. Since the information exchange protocol is standardized, differing run time environments or operating systems meeting these standards can communicate across the PSB backplane, much like multiple computers on a local area network.

The core elements, or agents, are linked together by the PSB on the backplane (fig. 1). The architecture, by nature, is modular; only the CSM is required in any given configuration. All of the agents are intelligent, i.e., the 1553 agent and *Spacelab* interface agent each have their own processor; hence, they can operate autonomously of the CPU agent. Therefore, the CPU agent is not required unless there is a need for high-speed computational resources. Additional intelligent agents can be developed as needed to meet individual experiment requirements. While the figures for these discussions will show a CPU

agent, a *Spacelab* interface agent, a 1553 agent, and a CSM in the same chassis, it is not a definitive configuration. As a matter of point, the 1553 agent and the SIA will probably not both reside in the same experiment controller since they are interfaces to two different platforms. Also, any number of a given agent can be used in a single chassis to meet requirements. An example is, two or more CPU agents could be used to meet the requirements of a computationally intensive experiment, or multiple 1553 agents could be used to link 1553 busses together.

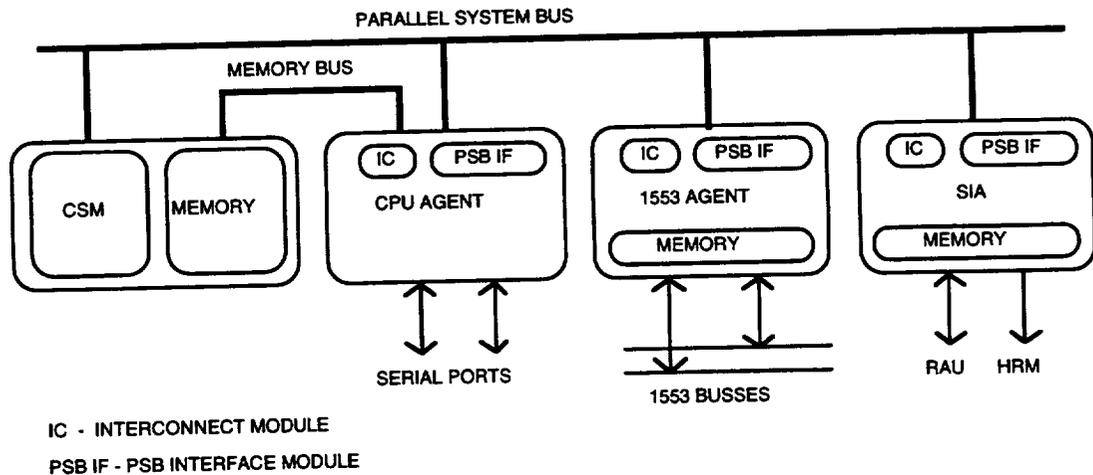


Figure 1. System overview.

The PSB as previously discussed, is a synchronous bus with four distinct address spaces. Two of which must be fully supported by any intelligent agent in the system. These address spaces are interconnect space and message space. Interconnect space is primarily used for agent and system initialization and auto configuration. Interconnect space is used to control execution of local and system-wide built-in self tests (BISTS), agent identification, and provides for an agent's configuration control via interconnect records. These functions are fully described in the "MPC User's Manual," "Multibus II Interconnect Interface Specification," and "Multibus II System Architecture Bootstrap Specification."

Message space is the primary information exchange mechanism between agents on the backplane. Unlike most other backplanes, data are not exchanged using shared memory mapped to multiple CPU's but is accomplished by sending messages to particular sockets on differing agents, much like a local area network (LAN). This approach has several advantages. Memory for communication need not be fixed to particular addresses and ranges for each agent. Memory on an agent is entirely under control of that agent, simplifying memory management and data coherency between agents. An agent's memory bus width can be 8-, 16-, or 32-bits wide to best suit its unique processor and system architecture, yet information is exchanged on the backplane using 32-bit wide data transfers. Memory can be allocated on the fly for messages and, if memory available is inadequate for the reception of a message, the message can be fragmented into smaller parts. All of these message exchange protocols, including message fragmentation, are defined in the IEEE Std 1296.

There are two types of messages, unsolicited and solicited. Unsolicited messages are short, one packet long messages that do not require any prearranged resources. A packet is a block of data exchanged between agents with 4 bytes of hardware-defined data and up to 28 bytes of user data for unsolicited messages or 32 bytes of user data for solicited message packets. Unsolicited messages are used to provide virtual interrupt services between agents, to exchange small amounts of information quickly, and to prepare agents for solicited messages. An unsolicited message can also be broadcast to all agents in a system simultaneously. Solicited messages are used for large data transfers. Exchanges using solicited messages must be set up using a set of standard unsolicited messages between the two agents involved. These unsolicited messages are buffer request, buffer grant, buffer reject, and fragmentation request. First, an agent needing to send data issues a buffer request corresponding to the amount of

data it needs to send to a particular agent. Next, the receiving agent checks its available free memory resources and issues a buffer grant or a buffer reject, depending on free memory available. If adequate free memory exists, a buffer grant is sent and the requesting agent begins transmission of data. If a buffer reject is sent due to insufficient memory, the rejecting agent can follow immediately with a request for fragmentation, and subsequently, the data are broken into a series of messages that fit the available memory. Solicited messages cannot be broadcast. For more information on these messages and the optional transport layer specification refer to "Multibus II Transport Protocol Specification and Designer's Guide."

The implementation of this message exchange protocol necessitates that each board addressable via message space have its own microprocessor. However, excepting the CPU agent that uses the 80486 microprocessor, the selection of a microprocessor for other agents is left to the designer of the agent.

The implementation of interconnect space is usually done using a dedicated 8-bit microcontroller (an 80C51FB) for each agent. Nonintelligent boards can be accessed through the more conventional memory and I/O spaces, though no development work on these type boards was done.

### A. Central Services Module

The CSM provides common system-wide functions for the PSB. It provides two system clocks, generates reset signals, initializes board slot and priority identities, monitors bus activities for time-outs, and relays low voltage indication from the power supply to the agents. Only one CSM is active in the system and it resides in slot 0.

The CSM is not an agent of itself, but it is a module that can be incorporated with other modules to comprise a board. For the wire wrap prototypes, the CSM is a separate board.

The CSM is designed as a state machine; it does not have a processor (fig. 2). The two system clocks, a 10-MHz "B" clock and a 20-MHz "C" clock required by the IEEE Std 1296, are generated by the CSM. These clock must be coherent, with a maximum skew of 10 ns. The B clock is used to synchronize the PSB interface logic on all the agents. The C clock is not currently used by any agent. The 10-MHz clock is also used locally on the CSM to synchronize the state logic in the CSM. The CSM

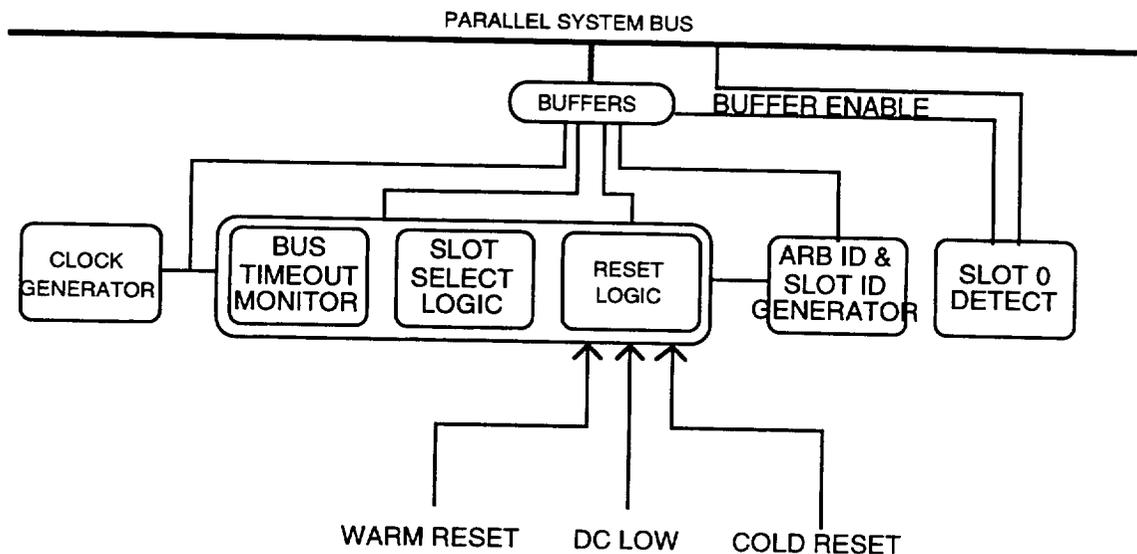


Figure 2. CSM.

implements the power-up reset logic for the system. There are two distinct types of resets that are generated by the CSM, cold reset and warm reset. On the prototype CSM, push buttons are used to initiate either a cold or warm reset; on power-up the CSM initiates a cold reset automatically. In a flight system, the push buttons could be replaced by discrete inputs or removed entirely. During the cold reset sequence, the CSM downloads the slot and priority ID's. The slot and priority ID's are programmable. Two time-out conditions are monitored by counters embedded in the CSM, data transfer cycle time-out and bus ownership time-out. In either case, the CSM will flag a time-out using the "TIMOUT\*" signal defined on the PSB. The CSM conditions the power failure indicator from the power supply and notifies all agents of imminent power failure. The CSM also detects whether it is located in slot 0; if not, it disables all of its outputs to the backplane.

## B. Interconnect Module

An interconnect module is required on all agents interfacing to the PSB and, hence, is discussed as a separate item.

The interconnect module is used to implement the interconnect space, to provide a standard diagnostic interface, to control reset of the local processor on an agent, and to provide a time tick to the PSB interface logic for solicited input and output message fail-safe time-outs. It also monitors the program input discrete and initiates the downloading of code to any agent in the system.

Initially, for the 1553 interface agent and the *Spacelab* interface agent (SIA), the designs of interconnect modules were left to the designer of each agent. Only the commonality of the 87C51FC microcontroller was mandated. However, with the subsequent discovery of latch-up failures of the 87C51FC, concerns about lack of EDAC for its on chip memory, and the desire for a common methodology for downloading code to an agent, a standardized design was adopted and implemented on the CPU agent. At a later date, the interconnect hardware on the SIA and 1553 interface agent will be updated to this standard.

The standardized interconnect space design (fig. 3) is based around the 80C51 family of 8-bit microcontrollers operating at 11.0592 MHz. All program instructions and data are stored externally to the microcontroller. The 80C51FB has separate 64-kbyte data and program spaces. There are 32 kbytes of EEPROM and 32 kbytes of SRAM; both are EDAC protected. The EEPROM can be shadowed to SRAM. No execution speed gains are made by shadowing; however, shadowing allows for single-bit error scrubbing of program space. Writes to EEPROM cause invalid data to appear in subsequent reads from EEPROM until its internal write sequencing is complete. This would cause invalid instruction fetches if the 80C51 is executing out of EEPROM. An Altera 5128 PLD is used to implement the 8-bit flow through EDAC to memory, and decode EEPROM and SRAM accesses. The microcontroller interfaces to the PSB via the message passing coprocessor (MPC). There are four other signals generated by the interconnect microcontroller to control the local processor; a reset signal, a nonmaskable interrupt, a maskable low-priority interrupt, and a cache enable signal; and one additional signal for fatal faults. The reset signal is used to place the local processor, not the interconnect micro, into reset. Several conditions can place the local processor in reset, a cold reset, warm reset, or a software controlled reset via interconnect space. The cache enable line is specific to the CPU agent and is used to enable or disable the 80486's on chip cache. The nonmaskable interrupt is used for requesting high-priority services of the local processor such as reprogramming the local processor memory. The low-priority interrupt is used for heart beat keep alive and diagnostic services between the local processor and its interconnect space. The interconnect module also monitors for fatal conditions in either the interconnect module or in the local processor module and sets the fatal fault flag when appropriate. The hardware set fatal conditions are double bit errors in interconnect memory, double bit errors in the local processor memory, double fault reset of the 80486, and dc low signal from the power supply. The 80C51 can also set the fatal fault flag via software control. The fatal fault flag is a "wired or" signal that can be set by any agent in the system, and each agent can have multiple modules capable of setting this signal. Note, only a cold reset and successful boot of the system will clear this flag. The interconnect module also implements the

required diagnostic port using the microcontroller's embedded serial port and an RS232 interface chip. The microcontroller also senses the presence of a terminal at the diagnostic port and sets a corresponding flag in interconnect space.

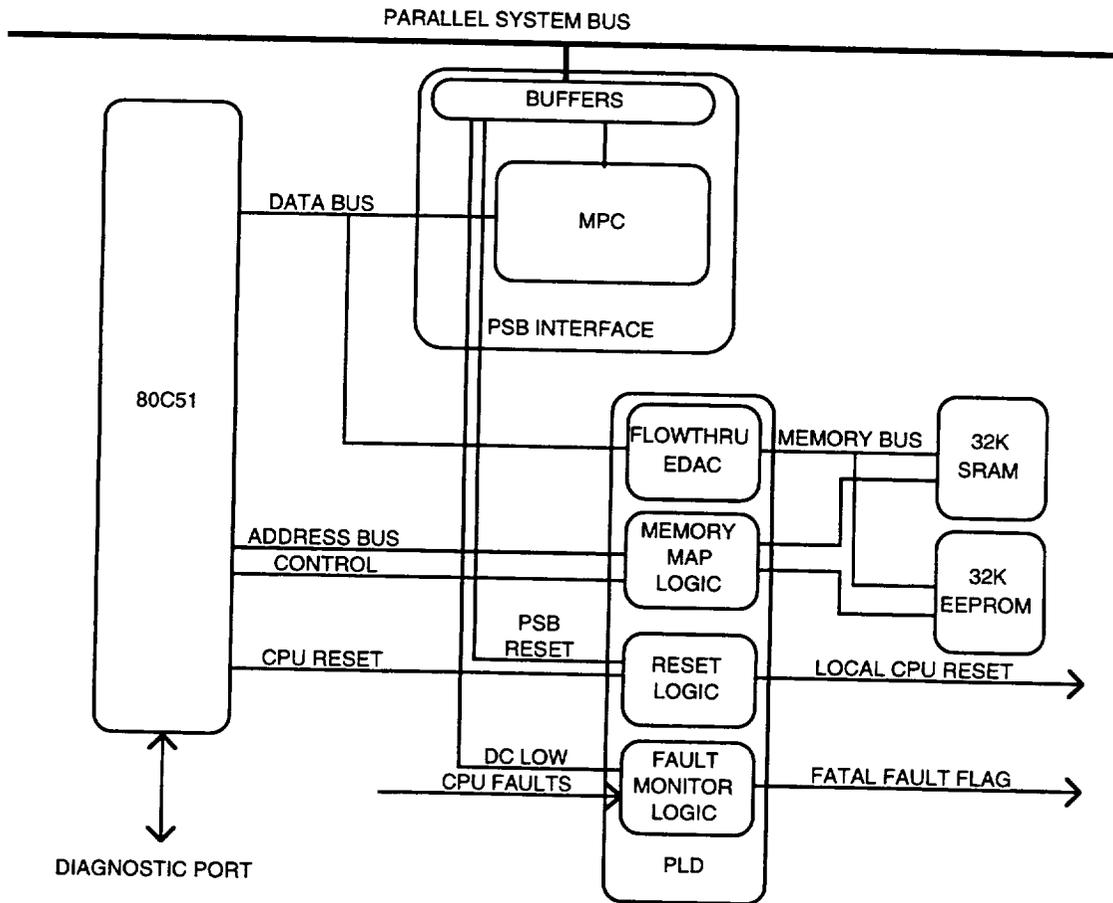


Figure 3. Interconnect module.

### C. CPU Agent

The primary function of the CPU agent is to provide computational resources for the system. It is based around an Intel 80486 microprocessor operating at 20 MHz. The CPU agent consists of six modules: an interconnect module, a memory interface module, a memory module, a PSB interface module, an I/O module, and a CPU module (fig. 4). All of these modules except the memory module reside on the same board. The memory module for CPU agent is on a separate board.

The CPU module is composed of a 80486 microprocessor, chip select logic, clock generation logic, reset synchronization logic, ready control logic, and an 82380 DMA controller with integrated peripherals. The 82380 provides four general purpose timers, eight 32-bit DMA channels, a wait state generator, a DRAM refresh controller, and three interrupt controllers with a total of 15 external interrupts and 5 internal interrupts. Six of the eight DMA channels are used, two for solicited message transfers and two for each serial channel. The DRAM refresh generator is used to implement automatic scrubbing of SRAM for single bit errors. The 20-MHz CPU clock frequency is used to allow 0 wait state accesses to memory. This optimizes throughput of the CPU when internal cache is disabled. The 80486's internal cache is not error protected, and, if it is susceptible to single event upsets, it can be

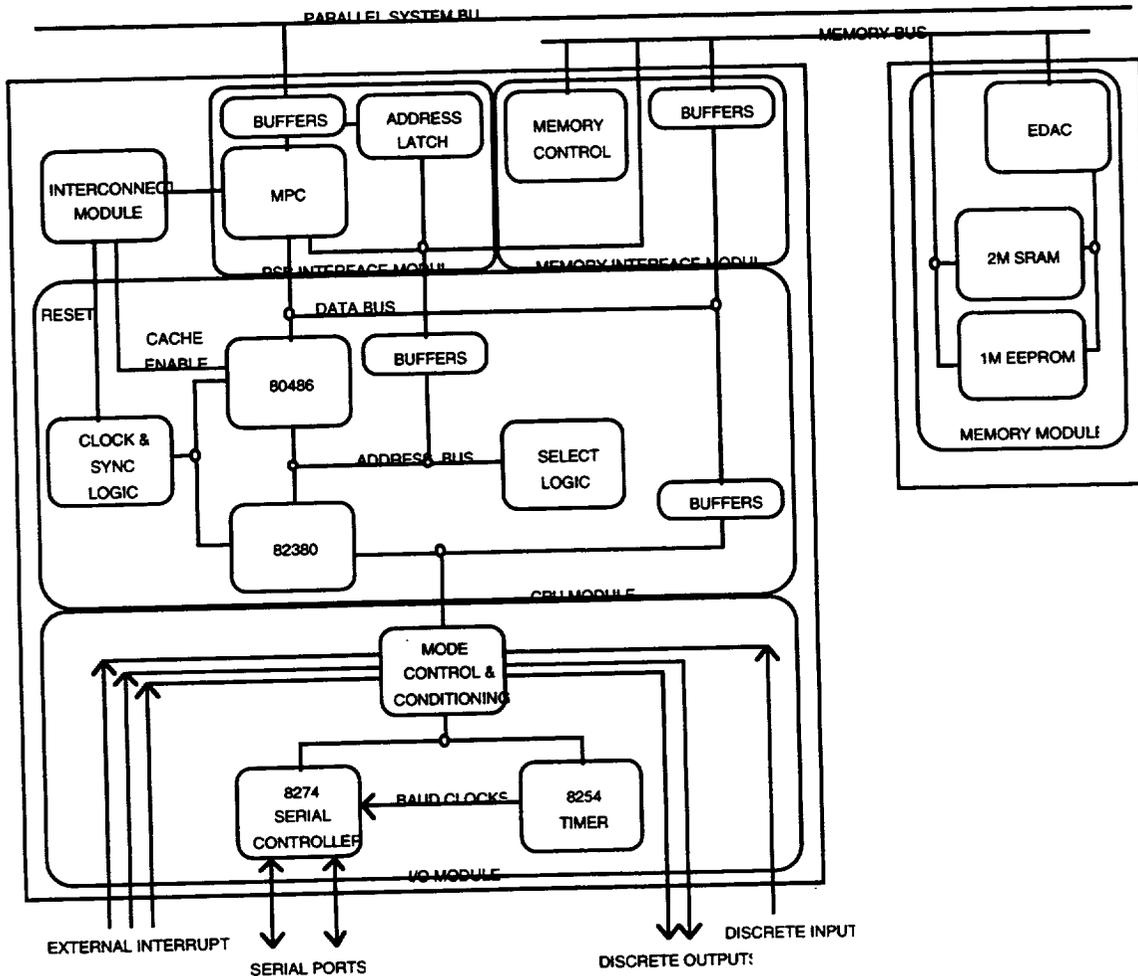


Figure 4. CPU agent.

disabled by the interconnect module. However, the CPU module is designed to fully support the use of the internal cache, if it is needed. Also, one upgrade path is to replace the 80486 DX with a frequency doubled 80486 DX2. This upgrade requires no modifications to the CPU agent.

The I/O module has two multiprotocol serial I/O channels, two output discretives, one input discrete, and three external interrupt inputs. The two serial I/O channels are implemented with a 8274 multiprotocol serial controller. Each serial channel can be programmed independently, supporting either DMA or interrupt driven operation, can have programmable data formats, and can support up to a 880-kbits per second data transfer rate. The baud rate clocks are generated using a 8254 programmable interval timer with an input frequency of 6.144 MHz. Channel A serial I/O supports hardware hand shaking. Channel B can be configured for transmitting or receiving an external data clock for synchronous operation. Both serial ports use RS-422 transmitters and receivers. The two output discrete use open collector outputs with high impedance pull up resistors. The discrete input is buffered through a high hysteresis inverter for noise immunity. The three external interrupts are buffered by high hysteresis inverters and conditioned by separate hardware enables and glitch suppression logic. The three interrupts are distributed in priority to provide high-, medium-, and low-priority interrupts to the CPU module.

The PSB interface module links the CPU agent to the backplane. The PSB module is implemented using a 82389 message passing coprocessor (MPC) that is buffered to the backplane by high current bus transceivers. The MPC and optional address latches implement all of the hardware protocol necessary to access all four address spaces on the PSB backplane. The MPC has four 32-byte unsolicited

message buffers arranged in a circular queue, and has two solicited message ping pong buffers. The MPC has separate interfaces to the interconnect module and the CPU module. It controls accesses to the memory and I/O spaces on the PSB. The PSB module is configured for single-cycle DMA transfers to and from memory using two separate DMA channels for solicited message transfers. All other access by the local CPU to the MPC is via conventional I/O references. The MPC also arbitrates for access to the PSB and to the interconnect module.

The memory module is linked to the CPU agent via the "P2" connector on the backplane and the memory interface module. The memory is located on a separate board primarily to reduce heat dissipation of the CPU board. This has an additional benefit of allowing easier memory upgrades without modification to the CPU board. Note, the memory bus on the backplane is totally independent of the PSB, and only spans two slots in the backplane. Hence, multiple CPU agents can reside in the system, each with its own independent memory bus and memory cards. The memory interface module also latches any error conditions for the memory module. The possible error conditions are single-bit error, double-bit error, and parity error. The first address of the error occurrence is also latched in the memory module for error recovery purposes. The memory module contains 2 Mbytes of SRAM and 1 Mbyte of EEPROM. Both SRAM and EEPROM are protected by an IDT49C465 flow through EDAC unit. Reads to SRAM require no wait states. Writes require one-wait state, since a write actually involves a read modify write cycle on the memory bus. This is necessary to support partial word writes to memory. EEPROM is accessed using two wait states.

#### **D. 1553 Agent**

The 1553 agent interfaces the other agents in the system to two dual-redundant MIL-STD-1553 buses. Each of these 1553 bus interfaces operates independently. The bus interfaces can operate as a bus controller (BC), a remote terminal (RT), or bus monitor (M). The 1553 agent is based on an Intel 80C186 microprocessor and two UTMC Summit 1553 interface chips (fig. 5). The 1553 agent is composed of six modules: a PSB interface module, a CPU module, two 1553 modules, an interconnect module, and a memory module. All of the modules reside on the same board.

The CPU module is composed of an 80C186, an optional 80C187 floating point coprocessor, interrupt control logic, a real-time clock, and local bus arbitration logic. The 80C186 is a 16-bit embedded processor, operating at 12 MHz. It has a built-in interrupt controller, three interval timers, memory and I/O select logic, and two 16-bit DMA channels. The local bus arbitration logic controls access to the memory among the two 1553 modules and the 80C186 microprocessor. The real-time clock is a 32-bit counter clocked at 1 kHz. The clock can be read without disturbing its count.

The 1553 modules implement the hardware aspects of MIL-STD-1553 protocol using UT69151 "Summit" intelligent protocol chip, UT63M127 transceivers, and 1.66:1 transformers. The UT69151 protocol chip executes semiautonomously of the local processor using control tables and a simple 4-bit instruction set. The 1553 protocol chips interrupt the processor only when necessary. The two protocol chips and the 80C186 share memory resources. The modes of operation of the protocol chips are programmed by the 80C186. The interface to the 1553 bus is via transformer coupling. Direct coupling to a bus is not supported. The remote terminal address of each channel is set by a connector on the front panel of the agent.

The memory module has 1 Mbyte of SRAM and 256 kbytes of EEPROM. Both EEPROM and SRAM are EDAC protected. SRAM accesses occur with no wait states. Both reads and writes to SRAM result in read modify write cycles to help insure data integrity. Memory scrubbing must be implemented by using software. EEPROM is mapped over SRAM at reset to allow simplified shadowing of EEPROM to SRAM.

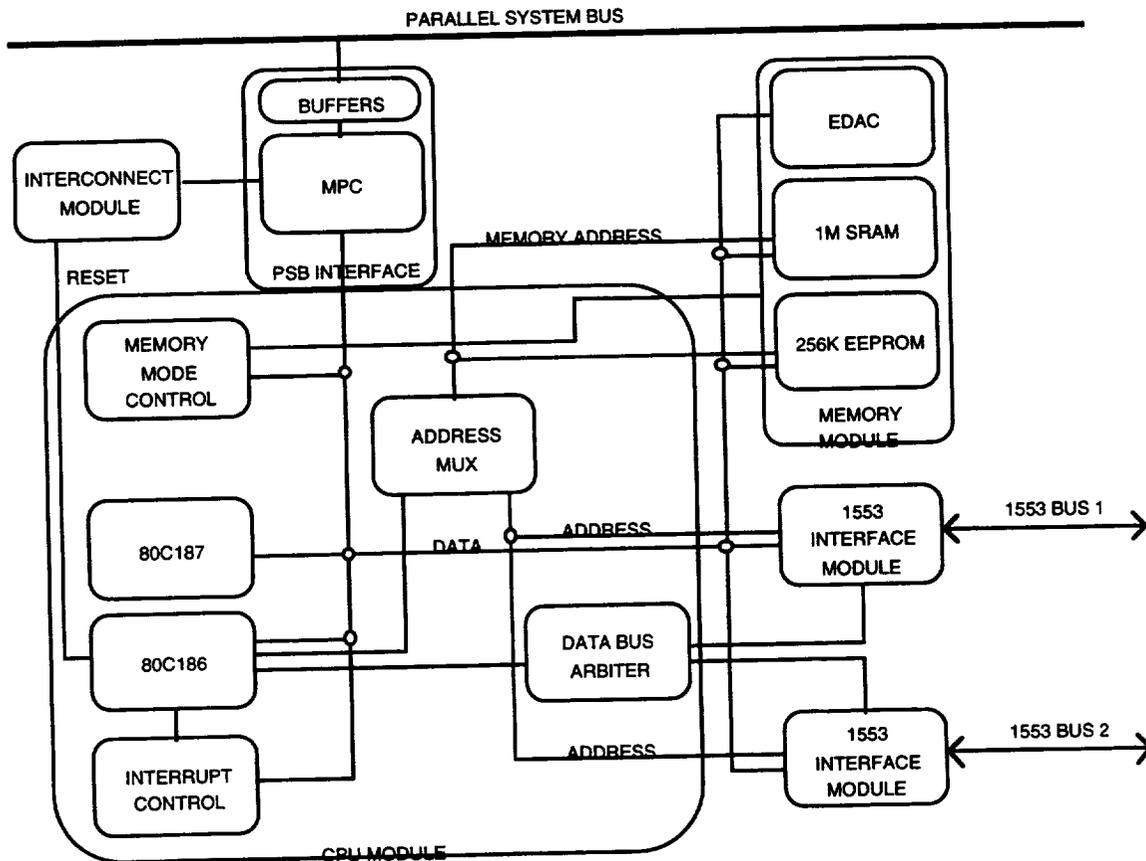


Figure 5. 1553 agent.

The PSB interface module is similar to the CPU agent's PSB interface module except that the 1553 agent only supports message and interconnect space references and the local bus is 16-bits wide. Also, the DMA accesses to the solicited message buffers use two-cycle transfers instead of fly-by transfers since the 80C186's embedded DMA controller only supports two-cycle transfers.

The interconnect module of the 1553 agent differs considerably from that of the CPU agents. The main differences are that the programming of the agents local memory is implemented using hardware, and an 87C51FC microcontroller with 32 kbytes of on chip UVPRAM and no external memory is used. This module will be redesigned to use the standardized module, used on the CPU agent, at a later time.

### E. Spacelab Interface Agent

The SIA provides all basic interfaces needed to interface with *Spacelab*. The SIA is composed of six modules: a PSB interface module, an interconnect module, a local CPU module, a memory module, a high rate multiplexor interface (HRMI) module, a remote acquisition unit interface (RAUI) module, and a timer module interface (TMI) module (fig. 6). All of these modules reside on a single card.

The CPU module of the SIA is responsible for controlling the individual *Spacelab* interfaces, formatting data, time tagging data, and monitoring the interface's health and status. The CPU module uses a 80C186 microprocessor. It has a 8259 programmable interrupt controller cascaded into the embedded interrupt controller in the 80C186. This is used by the RAUI, TMI, and HRMI to interrupt the processor to request services. The SIA CPU module has a total of nine 16-bit programmable interval timers: three embedded in the 80C186 and three in each of two 8254 devices. One 8254 is dedicated to the TMI module.

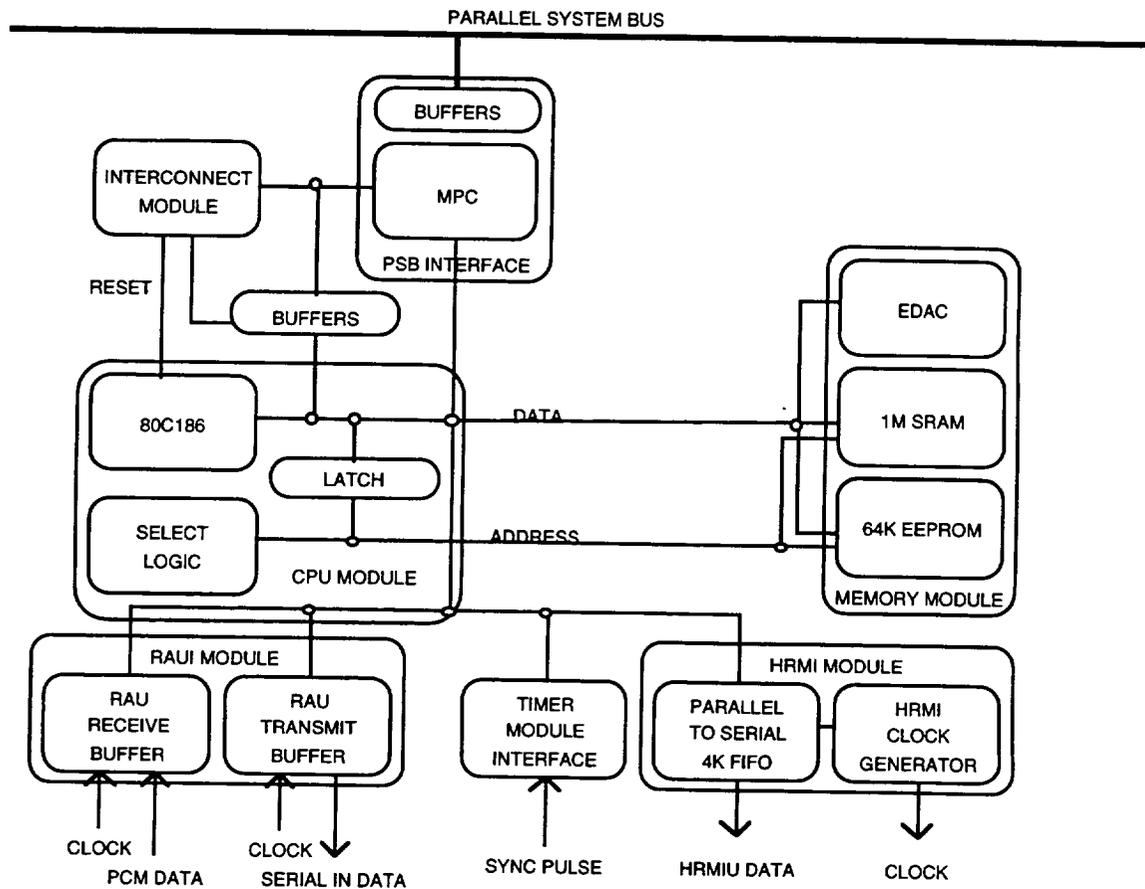


Figure 6. SIA.

The HRMI module is used to transmit experiment data to the *Spacelab* HRM via a synchronous serial output port. It has two output signals: HRM data out and HRM clock out. The HRMI module contains a 4-kbyte parallel-to-serial FIFO buffer. The FIFO generates status signals based on its start and end pointers to insure that valid data are always kept in the buffer. If the buffer is nearly empty, it will request more data from the CPU via an interrupt. The HRMI module also generates a programmable rate data clock. This clock can be programmed for data rates from 1 kbit to 4 Mbit per second and is output with the serial data from the FIFO's to the HRM.

The TMI module is used to maintain the experiment time synchronized to the *Spacelab* GMT. The GMT time value is down loaded via the RAUI and synchronized by a 4 pulse per second (PPS) update pulse from the RAU. The timer module interface uses one of the 8254 interval timer sets to maintain this time to the nearest fraction of a millisecond. Nominally this interval timer is clocked by a 1,024-kHz clock from the RAU. However, an onboard 1,024-kHz oscillator can be used if a failure is detected in the input clock from the RAU.

The RAUI module is used to interface to the *Spacelab* RAU, which is the primary command and status interface to the *Spacelab* experiment data system. It consists of two synchronous channels, a serial output (SO) data input and associated clock, and serial input (SI) data output port to the RAU. The "serial output" and "serial input" are named with respect to the RAU. The SO input link consists of two signals, synchronous serial data input and clock input. The SI channel consists of three signals, an output synchronous serial channel, an input data clock, and a SI request discrete. The SO and SI interfaces are described in the "*Spacelab* Payload Accommodation Handbook."

The memory module for the 80C186 consists of 64 kbytes of EEPROM and 1 Mbyte SRAM. SRAM is accessed with 0 wait states. All of the program memory can be shadowed to SRAM to improve execution speed. Both EEPROM and SRAM are EDAC protected. The EEPROM is programmed directly by the interconnect microcontroller.

The PSB interface module of the SIA is similar to the PSB module on the 1553 agent.

The interconnect module design for the SIA is unique to the SIA. This design will be replaced with the standardized interconnect module design at a later time.

## V. FABRICATION AND TESTING

The SIA, 1553 agent, CPU agent, and the CSM are built on wire wrap prototype boards. Wire wrap boards allow for easier implementation of design changes with minimal turnaround time. The PSB backplane and chassis are commercially purchased units. The CPU agent memory bus backplane is implemented using a three slot commercial iLBX backplane. The SIA, 1553 agent, and CSM were designed and fabricated first. Using lessons learned from the designs of the two agents, a simplified standard interconnect module was designed and incorporated into the design of the CPU agent.

Check out of the functionality of the boards will be accomplished using specifically designed ground support equipment (GSE). The GSE provides all necessary stimulus to exercise every external interface and to control execution of all test software in the system. The GSE also monitors and logs any error conditions that occur. The 1553 agent also may have an additional set of test requirements imposed by the MIL-STD-1553 RT validation test plan.

Currently, the *Spacelab* interface agent and the 1553 breadboards have not been tested pending redesign to incorporate the standard interconnect module and lessons learned. A separate wire wrap prototype of the standard interconnect module was built and used to verify its design and to develop and test interconnect firmware.

The wire wrap versions of the CSM and CPU agent are being tested as a part of the development effort of an onboard computer (OBC) for the automatic rendezvous and capture (AR&C) program.

## VI. FOLLOW-ON EFFORTS

During the last year of development, this design for an open architecture flight-qualified computer was proposed and accepted for use as the OBC for AR&C. The original proposed configuration consisted of a CSM, a CPU agent, and two new serial I/O (SIO) agents. The two SIO agents are identical in design, but uniquely configured for their associated external interfaces. Each SIO agent has four configurable RS-422 serial interfaces and uses a 80C186 microprocessor. The SIO's serial ports are accessed by the CPU agent via message space. The OBC will be fabricated using printed circuit boards and housed in a conduction-cooled chassis that the flight qualified version of this computer would use; however, commercial grade IC's will be used since AR&C is a ground demonstration program.

The iRMX III operating system (OS), a commercial, off-the-shelf real-time OS, was selected as the software development and execution environment for the OBC software for AR&C. The use of iRMX III and the use of a commercial hardware is allowing software development to be done in parallel with the hardware development and checkout. In previous efforts, a custom-built engineering unit had to be provided before application software development could begin. In this case, a commercial Multibus II system running iRMX III is being used for initial software development. Once the OBC hardware checkout is complete, the iRMX III nucleus will be ported to the OBC and, theoretically, the hardware

platform being used would be transparent to the application code. This parallel development should help reduce the overall development time of the OBC and its associated software.

One change was made in the OBC configuration from the original proposal for AR&C. The relative GPS filter function was moved inside the OBC from an external processor. This algorithm and proposed application code exceeded the projected throughput capabilities of a single CPU agent. Hence, the OBC configuration was changed to consist of a CSM, one SIO agent and two CPU agents, one for the GPS filter and one for the original application code. Each CPU agent would have its own copy of the iRMX III nucleus. The only redesign required for this configuration change was the addition of a slot to the backplane. The development environment for the GPS filter algorithm is a PC running iRMX III.

Checkout of the CPU agent, serial I/O agent, and CSM will be an on-going effort in support of AR&C. Redesign and checkout of the SIA and 1553 agent will be done as time permits or when these agents are needed.

## VII. CONCLUSIONS

The use of an open architecture standard that is commercially available as a basis for a flight computer has many benefits. The electrical, timing, and protocol characteristics of the backplane are predefined. Initial application code development can be accomplished using commercial hardware. Commercial systems can be used for initial hardware design checkout. A large knowledge base exists in the public sector that can be used to assist in hardware and software development. The use of a synchronous message passing architecture of Multibus II reduces hardware interdependencies between modules across the backplane, which simplifies integration of a variety of boards into the system. It also simplifies implementation of multiprocessor systems and adds flexibility. However, the presence of multiple processors and their associated peripherals and memory increases power consumption compared to a single processor architecture with nonintelligent I/O modules. Also, commercially based systems are by nature more generic than custom systems, hence, power, size, and weight cannot be optimized as well as in a custom system. Generally, the interface chip sets are not available in radiation hardened versions. Hence, in applications where radiation, size, weight, or power requirements are tight, a commercially based architecture may not be viable.

The development of an open architecture "flight qualifiable" computer based on a Multibus II architecture was more involved than first anticipated. The message passing architecture, interconnect space, and associated protocols are complex and require long learning curves. Also the desire to meet the optional Multibus II system architecture, and transport protocol specifications added to the complexity. Standard modules used in the designs of these agents can be used as building blocks for the designs of additional agents for the system. The use of this computer by the AR&C project will provide a real world verification of the concepts presented in this development.

## REFERENCES

1. "ANSI/IEEE Draft Standard P896.1: Futurebus+ Logical Layer Specifications," The Institute of Electrical and Electronics Engineers, Inc., February 1990.
2. "ANSI/IEEE Draft Standard P896.2: Futurebus+ Physical Layer and Profile Specifications," The Institute of Electrical And Electronics Engineers, Inc., January 1991.
3. "ANSI/IEEE Draft Standard P896.3: Futurebus+ Recommended Practices," The Institute of Electrical and Electronics Engineers, Inc., April 1991.
4. "ANSI/IEEE Std 1014-1987: IEEE Standard for a Versatile Backplane Bus: VMEbus," The Institute of Electrical and Electronics Engineers, Inc., 1988.
5. "ANSI/IEEE Std 1296-1987: IEEE Standard for a High-Performance Synchronous 32-Bit Bus: Multibus II," The Institute of Electrical and Electronics Engineers, Inc., 1988.
6. "Firmware User's Guide for Multibus II System Architecture (MSA) Firmware," Order Number 506090-002, Intel Corporation.
7. "MIL-STD-1553B Military Standard: Aircraft Internal Time Division Command/Response Multiplex Data Bus," Department of Defense, September 1978.
8. "MPC User's Manual," Order Number: 176526-002, Intel Corporation, 1990.
9. "Multibus II Interconnect Interface Specification," Order Number: 149299-003 Intel Corporation, 1990.
10. "Multibus II System Architecture Bootstrap Specification," Order Number: 455975-001, Intel Corporation, 1988.
11. "Multibus II Transport Protocol Specification and Designer's Guide," Order Number: 453508-001 Intel Corporation, 1988.
12. Saaty, T.L.: "Priority Setting in Complex Problems." IEEE Transactions on Engineering Management, vol. EM-30, No. 3, The Institute of Electrical and Electronics Engineers, Inc., August 1983.
13. "Spacelab Payload Accommodation Handbook, SLP/2104-1, Appendix A Avionics Interface Definition, Issue no: 4, Revision: 0," NASA MFSC Spacelab Program Office, June 23, 1993.

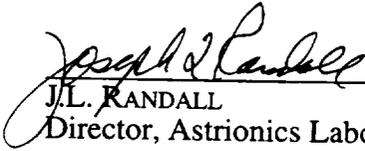
**APPROVAL**

**DEVELOPMENT OF AN OPEN ARCHITECTURE FLIGHT QUALIFIED COMPUTER**

**(Center Director's Discretionary Fund Final Report, Project Number 92-R07)**

By B. Beabout

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

  
\_\_\_\_\_  
J.L. RANDALL  
Director, Astrionics Laboratory

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE January 1996	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE Development of an Open Architecture Flight Qualified Computer (CDDF Final Report, Project Number 92-R07)			5. FUNDING NUMBERS	
6. AUTHOR(S) B. Beabout			8. PERFORMING ORGANIZATION REPORT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812			10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA TM-108504	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			11. SUPPLEMENTARY NOTES Prepared by Astrionics Laboratory, Science and Engineering Directorate.	
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified-Unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  A space flight qualified controller for experiments that is modular and based on an open architecture commercially available standard can reduce system development time by leveraging off commercial hardware and software. While the unique requirements of flight may mandate custom hardware designs, a modular design approach in which a core set of modules is designed and built would provide a basis for future experiment controllers. Any unique requirements could then be met by adding modules as necessary. A central processing unit module, a MIL-STD-1553 interface module, and a Spacelab interface module were developed. These modules are linked using the IEEE standard 1296 Multibus II™ bus architecture. This report describes the work done to develop this core set of processing and interface modules that meet the IEEE 1296 Multibus II™ standards.				
14. SUBJECT TERMS experiment controller, data system, Multibus II™, single board computer, MIL-STD-1553			15. NUMBER OF PAGES 26	
			16. PRICE CODE NTIS	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	



National Aeronautics and  
Space Administration  
Code JTT  
Washington, DC  
20546-0001

*Official Business*  
*Penalty for Private Use, \$300*

*Postmaster: If Undeliverable (Section 158 Postal Manual), Do Not Return*

---