An Innovative On-Board Processor for Lightsats

R. M. Henshaw, B. W. Ballard, J. R. Hayes, and D. A. Lohr
The Johns Hopkins University Applied Physics Laboratory

The Applied Physics Laboratory has developed a flightworthy custom microprocessor that increases capability and reduces development costs of lightsat science instruments. This device, which APL calls the FRISC (Forth Reduced Instruction Set Computer), directly executes the high level language called Forth, which is ideally suited to the multitasking control and data processing environment of a spaceborne instrument processor. The FRISC (which is available commercially as the SC32) will be flown as the on-board processor in the Magnetic Field Experiment on the Swedish Space Corporations's Freja satellite. APL has achieved a significant increase in on-board processing capability with no increase in cost when compared to the magnetometer instrument on Freja's predecessor, the Viking satellite. These advantages are attributable to the high instruction execution rate, reduced software development effort, and shortened system integration time made possible by the nature of the microprocessor and the Forth language.

BACKGROUND

The Johns Hopkins University Applied Physics Laboratory (APL) has built a variety of small satellites and space instrumentation since the launch of its first TRANSIT navigational satellites in the early 1960's. Although these satellites and instruments have served a wide variety of functions, they share most of the following characteristics of small satellite applications:

- limited power
- limited weight
- limited telemetry data rate
- limited funds for development
- tight development schedules
- ever-increasing data processing requirements

APL has developed an innovative microprocessor called FRISC (Forth Reduced Instruction Set Computer) which reduces the impact of these constraints on small satellite applications. Its high speed, low power, space reliability, and programming ease suit it for the multitasking real time control and computation environment of modern space

instrumentation. Because of these advantages, APL is using the FRISC in a magnetometer instrument it is building at the invitation of the Swedish Space Corporation (SSC) for the Freja satellite. This paper presents FRISC's technical features and shows how they apply to small satellite applications, using the Freja Magnetic Field Experiment (MFE) as an example.

THE FREJA SPACECRAFT

Freja will orbit the Earth at high inclination to explore the physics of the auroral zones. Its eight instruments will obtain high resolution measurements of the upper ionosphere and lower magnetosphere to determine the fine structure of the particle and field environment in the circumterrestrial plasma. The best data currently available was acquired by the Viking satellite, Freja's predecessor from SSC. Freja will provide finer spatial resolution and higher dynamic range in measurements of the distribution of charged particles, waves, and electromagnetic fields.

Freja is very much an international venture. Instruments are being provided by the U.S., Canada, West Germany, and Sweden. SSC is purchasing several host spacecraft subsystems from the U.S., West Germany, and the People's Republic of China. The 2.2 meter diameter spacecraft is scheduled for a 'piggyback' launch on a Chinese Long March rocket in the summer of 1992 for a minimum one year mission.

Shown below are the key resources available for the Freja satellite, and the resource allocation for the MFE in particular. All three of these resources restricted the design of the MFE, creating a multitude of 'opportunities' for innovation.

FREJA SATELLITE RESOURCES

Resource	Spacecraft Total	MFE Allocation
Weight	230 kg	3.7 kg
Power	86.7 W	4.5 W
Telemetry Rate	256 kb/sec 512 kb/sec	14.3 kb/sec 28.7 kb/sec

FREJA MFE SCIENCE REQUIREMENTS

The MFE's primary objective is to collect and downlink 16-bit magnetic field vector samples at 128 Hz or 256 Hz, depending on which of the two spacecraft telemetry rates is selected. Since there is no mass storage device on-board, the ground station only receives these real time measurements during station passes.

Supplementary objectives are to make magnetic field measurements outside ground station passes using burst and full orbit data collection modes. These modes complement the real time function by storing data in the MFE's memory for downlink during a later station pass. The burst function collects data at 128 Hz for 40 seconds upon command

from the spacecraft processor, while the full orbit function collects data continuously at a low rate programmable from 0.0625 Hz to 16 Hz. The spacecraft processor will command all the Freja instruments to execute their burst data collection functions simultaneously to allow correlation between the instruments.

Another science objective is to provide spectral data of magnetic fields up to frequencies near the bandwidth cutoff of the magnetic detection circuit. This spectral data cannot be derived by processing the real time data samples because telemetry bandwidth limitations do not permit adequate sample rates. Therefore, high rate samples must be processed on-board to generate spectra to downlink selectively at a low rate.

ON-BOARD PROCESSING REQUIREMENTS

The following processing requirements are necessary for the Freja MFE to meet the science objectives discussed above:

1. Provide anti-alias low pass filters for DC and AC channels
   - 64 Hz cutoff during normal telemetry rate operations
   - 128 Hz cutoff during high telemetry rate operations

2. Digitize X, Y, Z AC and DC magnetic field measurements to 16 bits
   - 128 samples/sec during normal telemetry rate operations
   - 256 samples/sec during high telemetry rate operations

3. Oversample and average X, Y and Z DC measurements

4. Anti-alias filter one DC channel with 256 Hz cutoff and sample at 512 samples/sec

5. Provide amplitude spectrum 0 to 256 Hz for the above DC channel
   - send raw 512 sps samples for ground processing, or
   - perform FFT on-board and send amplitude information

6. Collect and digitize housekeeping and status data

7. Format and output telemetry

8. Interpret and execute commands

A conventional design approach for fulfilling these requirements might include a switchable hardware anti-aliasing filter (for the two different sampling rates), a 16-bit A/D converter, and a general purpose processor, with spectral analysis performed by post-processing on the ground. The processor would be programmed in assembly language or a high level language, cross-assembled or cross-compiled on a separate machine. The object code would be downloaded to the target hardware for debugging using in-circuit emulators or other support equipment.

Unfortunately, this configuration does not fit within the spacecraft resources allocated for the Freja MFE. First, there is neither the power, the circuit board space, nor the noise floor margin for switchable hardware anti-aliasing filters. Second, telemetry data

rate limitations preclude sending the one 512 sps channel to the ground for spectral processing. Although a separate on-board digital signal processing device could perform this processing, it too would exceed the available power and board space, and would add significantly to the hardware and software design time required. Finally, the conventional embedded system software development approach, with cross-development tools and in-circuit emulators, is extremely inefficient due to its long edit, compile, download, and emulate cycles.

The Freja MFE solves these resource limitation problems using simple, fixed hardware anti-aliasing filters, a 16-bit A/D converter, and the single-chip FRISC microprocessor. The FRISC performs data acquisition and averaging, digital anti-alias filtering, FFT computation, telemetry formatting, command interpretation and execution, and other instrument control functions. Furthermore, software development and debugging are performed interactively on the actual target hardware in high level language via a standard computer terminal.

MFE DESIGN OVERVIEW

Fig. 1 is a block diagram of the Magnetic Field Experiment, which consists of a probe and a magnetometer signal processor containing five electronics boards. The probe is mounted on a 2-meter boom to avoid spacecraft-generated magnetic fields, and measures magnetic fields with its three mutually perpendicular coils. The sensor electronics board processes and filters analog signals from the probe and then sends them to the Filter - A/D board. This board filters them further, converts them to digital form and buffers them in a FIFO under control of the on-board sampling sequencer unit (SSU). The CPU board reads the data from the FIFO, performs the FFT and digital filtering tasks, formats the resulting data and sends it to the telemetry interface board, which buffers it and sends it using a serial protocol to the Freja spacecraft electronics for transmission to the ground. Concurrent with the data handling tasks, the CPU controls the sampling sequencer, collects and formats the housekeeping data, and executes uplinked commands. The telemetry interface board also receives serial commands and selected telemetry words from the spacecraft. It converts them to parallel and passes them to the CPU board for interpretation and/or execution.

The DC/DC converter board receives 28 volt DC power from the FSU and generates ±5 volt and ±12 volt analog and +5 volt digital power for the other boards. It also contains current monitoring and power interruption circuitry to provide latchup detection and recovery.

Fig. 2 illustrates the hardware functions on the CPU board. A fusible link boot PROM program loads itself into SRAM after any reset command, turns off the PROM to save power, and waits for either a telemetry system command or a debug terminal command. If neither of these occurs within 10 seconds, the boot program automatically loads the application software stored in the first EEPROM module. We included the capability to uplink new application software into the EEPROM via the command system for programming upgrades. One memory module slot on the CPU board can be chosen to be used for more EEPROM or additional SRAM at the time the CPU is fabricated.

Other hardware functions on the CPU board include a prioritized interrupt controller, a real time clock, telemetry timers, a housekeeping A/D converter, and a

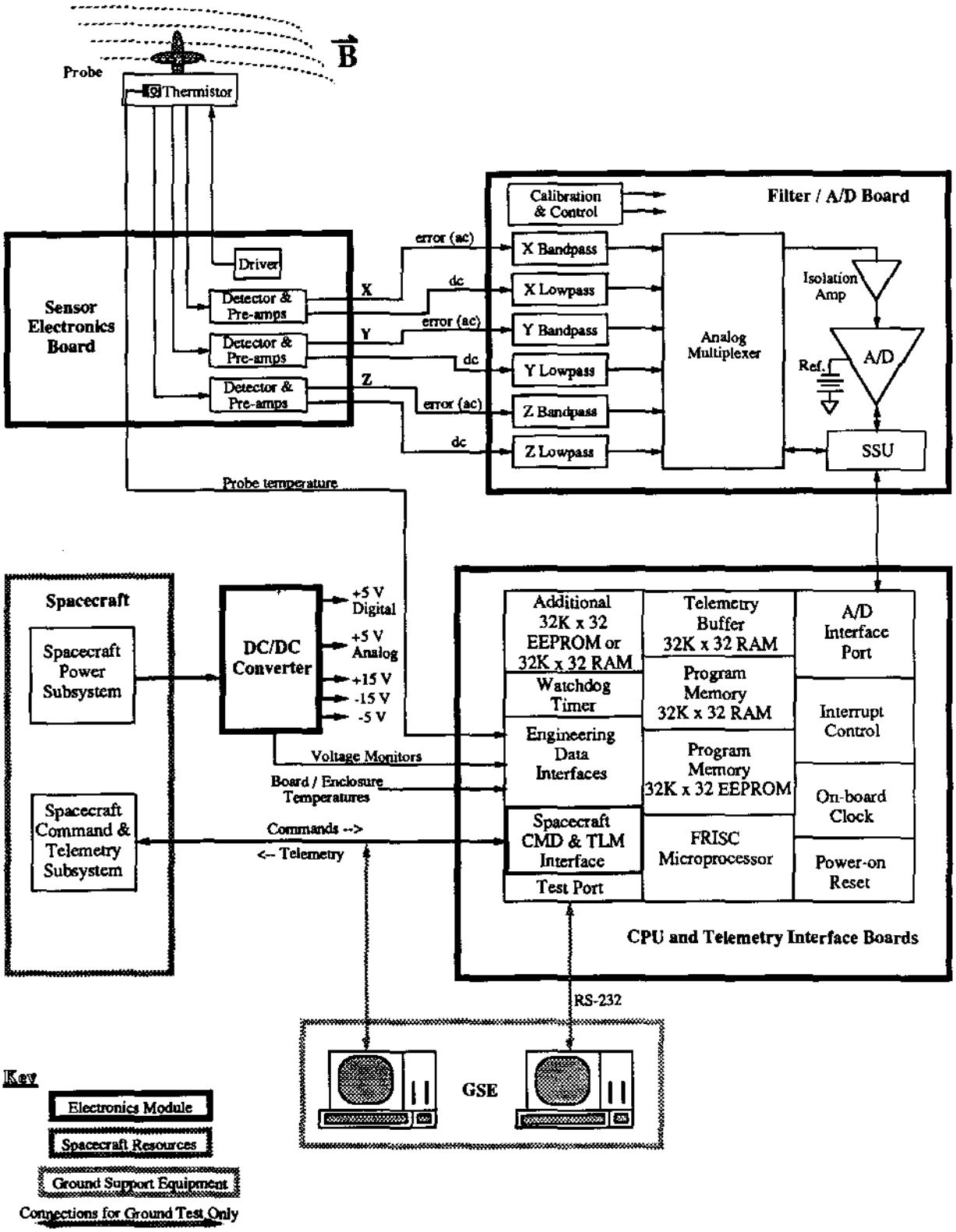

Fig. 1 Freja MFE Block Diagram

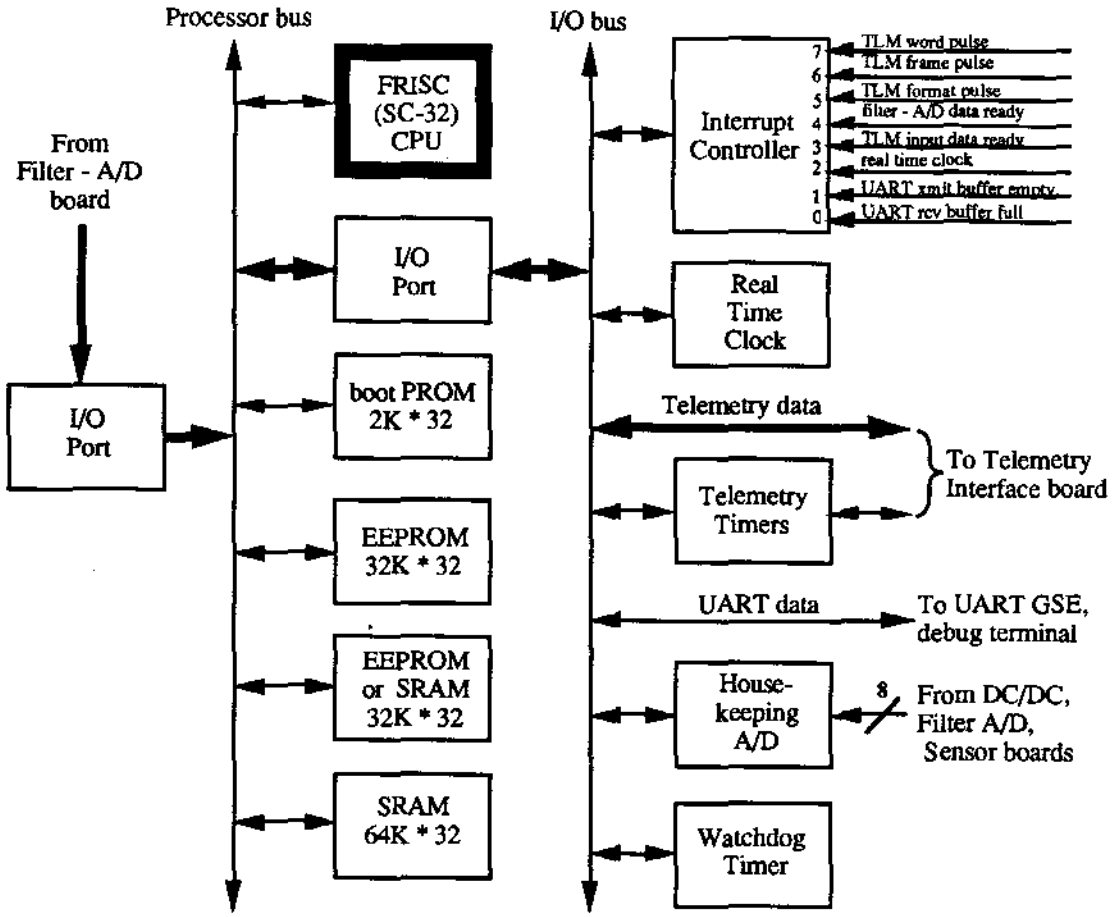

Fig. 2 CPU Board Block Diagram

watchdog timer (used to confirm that the application software is running properly). A CPU test port connects to a terminal to provide communication and control functions via the interactive Forth language interpreter during system development and testing.

THE FRISC

FRISC's power stems mainly from its direct execution of the Forth language. Contributing to its capability are many interrelated aspects of its design, including device architecture, execution speed, software structure, system test attributes, and other hardware characteristics. APL engineers designed and developed it using a silicon compiler CAD workstation, which allows specification of a custom integrated circuit design at a functional level. The final chip design resulted from our past experience in designing on-board processors for other space instruments with constraints similar to those described above. The following discussion provides an overview of the Forth language and then describes the main features of the FRISC design, illustrating why we are using it on the MFE.

WHY FORTH?

Forth is an interactive, stack-based hierarchical language which is ideally suited for embedded hardware control and processing applications. Astronomers and engineers at the National Radio Astronomy Observatory developed it in the early 1970's to control radio telescope dishes, and it has since spread throughout the embedded systems world from its original niche in the astronomical community. APL has used Forth successfully on several space missions, for tasks ranging from relatively simple data acquisition functions to control of the complex, space shuttle based Hopkins Ultraviolet Telescope (HUT). HUT is part of the Astro shuttle payload, in which three of the four major telescopes use Forth as their instrument control language. The American National Standards Institute (ANSI) is currently developing a Forth language standard (X3J14), and an APL engineer is serving on the standards committee. Forth development systems are commercially available for all commonly used processors.

The Forth language uses a small number of primitive instructions (Forth words) to form a kernel from which all other higher level Forth words are created. APL's FRISC processor implements this **reduced instruction set** of **Forth** primitives directly in hardware as its machine instruction set, hence the name - **F**orth **R**educed **I**nstruction **S**et Computer.

Higher level (i.e., non-primitive) Forth instructions are defined as sequences of lower level instructions, which can include both primitives and previously defined words. Thus, programming in Forth consists of extending the language by adding definitions specific to the application. This process thus creates a hierarchy consisting of both the operating/development system and the application software. Since definitions of higher level words consist only of sequence lists of previously defined words, the final software code is very compact. This attribute has allowed us to develop a combined operating system and software development system that resides on the CPU board and incorporates the constructs necessary for the real-time multitasking environment encountered on spacecraft. The fact that high performance and a self contained development system can be contained on a compact CPU board makes the FRISC system ideal for lightsat applications.

DEVICE ARCHITECTURE

Fig. 3 is a block diagram of the FRISC architecture. Four features of the architecture particularly enhance its capabilities as an on-board processor:

1. Single clock cycle execution of most Forth primitives

2. 32 bit data and address paths

3. On chip data and return address stack caches

4. Concurrency primitives and interrupt scheme to support multitasking

Almost all Forth instructions are executed in only one clock cycle, enabling high performance with relatively slow clock rates. During this single clock cycle, a pre-fetched

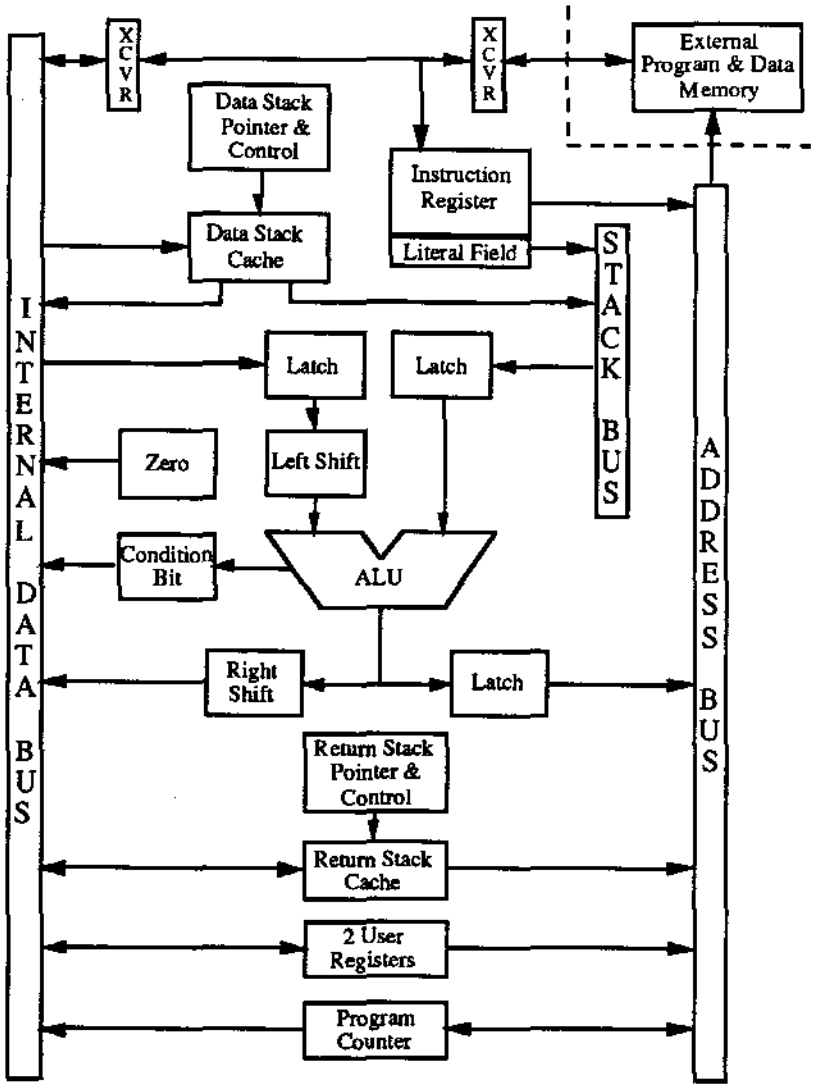

Fig. 3 FRISC Architecture. All data and address buses are 32 bits wide.

instruction is decoded and Forth primitive instructions are executed directly in hardware; simultaneously, the next instruction is being fetched. This architecture thus eliminates low level assembly language, allowing high execution rates of programs written in a high level language without compiling to machine code or the use of 'tricky coding' techniques. For example, in the MFE case we found that running the FRISC at 4 MHz was adequate to meet our real time processing requirements, even though the part is capable of running at 10 MHz. Using the lower clock rate saved power by allowing us to half-cycle the main memories, and gained additional timing margin against radiation-induced parameter shifts.

FRISC's 32 bit address space allows a directly addressed maximum memory size of 4.3 GWords, without external memory management devices. In real applications, however, the addressing scheme is optimized to accommodate memory mapped input/output decoding, bootstrap memory, program memory, and data buffer memories.

The data buffers can be very large, which is useful when several image arrays need to be manipulated, or when weight limitations preclude an on-board tape recorder. Many scientific instruments are proposing large CCD detectors (1000 x 1000, or larger) for data acquisition, which will require large data buffers for storage and processing. When the full complement of address bits are not required, the unused bits can aid the address decoding process. On the MFE, for example, we set appropriate address bits to specify several different bus wait state delays for various slow I/O devices.

The 32 bit data path is convenient for on-board data processing functions, since roundoff error is reduced when performing accumulation or digital signal processing functions. The 32-bit fixed point format contains enough dynamic range so that floating point operations (which usually imply a separate floating point device) can be avoided. There is no hardware multiplier on the FRISC, but the software multiply operation can be optimized for the number of bits required in the calculations.

The Forth language uses data and return address stacks to simplify the number of addressing modes needed to implement functions. Operands are popped off the stacks and results are pushed back onto the stacks. Normally, Forth software uses external memory to store these parameter stacks. Since APL's FRISC includes 16 word stack caches on chip, the number of memory read and write operations is greatly reduced and the instruction execution rate remains high. The stacks are automatically extensible to external data memory when the caches become too full or too empty; this cache management is performed in hardware and is invisible to the programmer.

SOFTWARE DEVELOPMENT AND SYSTEM TEST

The Forth language and the FRISC together provide an environment ideally suited for hardware debug, software development, and system integration and test. Since the small Forth operating / development system (under 7K memory words) is part of the flight software, the programmer can create new Forth words 'on the fly,' using only the flight CPU and a terminal. This capability is invaluable for debugging new subsystem interfaces during initial integration, and for 'glitch busting' the elusive, intermittent problems which remain when system integration is '99% complete.' For simple tests the engineers can type and execute new definitions directly, while they edit and save longer test routines and formal system test code in files. In either case they short circuit the traditional edit-compile-link-download-execute cycle and its usually agonizing slowness.

Development of flight software is similar to the development of formal test software, since both consist of a fairly large number of source code routines stored in files under of configuration control. The primary difference is that flight software usually contains many processes which must execute simultaneously at varying rates in response to hardware and software interrupts. Development of concurrent software such as this requires an environment which allows the programmer to describe asynchronous processes independently, and which also provides mechanisms for communication and synchronization between them.

FRISC provides a simple interrupt structure with only one interrupt level, but which is easy to expand with minimal external hardware. The Freja processor uses an eight level priority encoder and eight flip-flops to provide eight prioritized levels for hardware interrupt. The Forth operating system supports concurrent software, containing words to define software processes, to set their priorities and to activate and deactivate them. WAIT and SIGNAL are primitives which synchronize these processes, either to hardware

interrupts or to software events generated by other processes. With these underlying features it is easy for programmers to write independent processes and to implement mutual exclusion and other constructs necessary for orderly interprocess communication.

With the help of these features, one programmer integrated and debugged the prototype experiment and developed the flight code for the Freja MFE processor in about two months. The code contains 8 concurrent processes, 2500 lines of application Forth source code, and occupies 16 Kwords of memory (including the operating / development system). When the engineering model was delivered to Sweden for an interface test with the satellite processor prototype, the flight software worked perfectly without change.

HARDWARE CHARACTERISTICS

The FRISC is housed in an 84 pin grid array package, which represents a tradeoff between desire for small package size and enough I/O pins for the 32 bit buses. The chip is fabricated in 2 μm feature size CMOS technology, and dissipates 600 mW of power while running at its maximum 10 MHz rate. It should be noted that the hardware characteristics described here apply only to the version of the FRISC that is currently being fabricated. One of the attributes of our silicon compiled design is that it can be 'retargeted' without modification to several different foundries that use different fabrication technologies. Thus, the end product can be optimized for radiation hardness, power dissipation, speed, yield, military specification compliance, and cost. We chose European Silicon Structures (ESS) as the fabrication house because of their low cost per device when ordering small quantities to Mil-Std-883C.

RADIATION TOLERANCE

As of this writing, we have accomplished some, but not all, of the radiation tests we wish to perform on the ESS version of the FRISC. The total dose specifications for the expected Freja orbit are 7 krad/yr. Although the baseline mission duration is 1 year, we have a 2 year lifetime design goal, and have set minimum total dose tolerances in the 15-20 krad range. APL has an in-house facility to perform total dose tests with a Cobalt 60 radiation source. Unfortunately, our results to date show a wide range of total dose tolerance numbers. On two different fabrication lots that we used for breadboarding, we obtained total dose numbers between 15 krad and 22 krad, while irradiating at high dose rates (1 Krad/min). All of these parts recovered within a few days due to an inherent stored charge dissipation process (annealing). Our first flight lot showed a total dose resistance of 4 krad, which is unacceptable for the Freja mission without shielding. We are currently working with ESS to try to identify the source of the difference, and are simultaneously starting to upgrade some of our breadboard parts to flight quality.

Also at APL, we have a Californium facility that can be used for a limited latchup test. Heavy ions with a mean LET (linear energy transfer) of 36 Mev-cm^2/mg are emitted at a high flux rate. The FRISC did not latch during a 30 minute exposure.

A third radiation test was performed at Brookhaven National Laboratory to detect latchup susceptibility over a broad range of ion energies. Tests at Brookhaven indicated no large increase in power supply current that would typify a latched device, but a failure of the test circuitry created uncertainty as to whether the chip was running correctly. We intend to retest at Brookhaven with a more rugged set of test support electronics in August, 1990. If time allows, we will also quantify single event upset (SEU) tolerance.

STATUS AND AVAILABILITY

Currently, there is an APL patent pending on the FRISC design. APL has licensed the FRISC to Silicon Composers, Inc., Palo Alto, CA, who offers the chip as a commercial grade device designated the SC32. They also market a single board computer that plugs into IBM PC compatible computers, with their own operating/development system. The IC foundry (ESS), recently established a Mil-Std-883 line which was used for our Freja FRISC fabrication. Our reliability group performed a pre-cap visual inspection of the Freja parts at the foundry and confirmed that ESS has a high quality fabrication process. Additionally, we have an internal testing and screening program at APL that upgrades the parts to a reliability above Mil-Std-883 parts.

CONCLUSION

We have found that using the FRISC microprocessor streamlined the hardware and software development of the Freja Magnetic Field Experiment, and helped achieve conformance to the overall MFE electronics design envelope. Further, the design methodology of using a silicon compiler to produce a flightworthy custom integrated circuit has been validated. This technology allows the hardware designers to optimize the conflicting factors of cost, reliability, performance, and power dissipation for their project's needs.

ACKNOWLEDGEMENTS

We wish to thank NASA Headquarters and the Office of Naval Research for sponsoring the Freja MFE project. We also thank Larry Zanetti of APL, the MFE Principal Investigator, for his support and encouragement.

BIBLIOGRAPHY

1. B. W. Ballard, R. M. Henshaw, and T. Zaremba, "Development of Powerful Space-Qualified Computers," *Applied Physics Laboratory Developments in Science and Technology*, 1983

2. B. W. Ballard and R. M. Henshaw, "Forth Direct Execution Processors in the Hopkins Ultraviolet Telescope," *Proceedings of the 1984 University of Rochester Forth Applications Conference*

3. Philip J. Koopman, Jr., Stack Computers - The New Wave , pp.87-97, Ellis Horwood Limited, 1989

4. John R. Hayes, Martin E. Fraeman, Robert L. Williams, and Thomas Zaremba, "A 32 Bit Forth Microprocessor," *Proceedings of the 1987 University of Rochester Forth Conference*

5. John R. Hayes, Martin E. Fraeman, Robert L. Williams, and Thomas Zaremba, "An Architecture for the Direct Execution of the Forth Programming Language," *IEEE Proceedings of the Second International Conference on Architectural Support for Programming Languages and Operating Systems*, October, 1987

6. J. R. Hayes and S. C. Lee, "The Architecture of the FRISC3: A Summary," *Proceedings of the1988 University of Rochester Forth Conference*

7. John R. Hayes and Susan C. Lee, "Stack Caching in the SC32 Forth Processor," *1988 FORML Conference Proceedings*

8. John R. Hayes, "Multitasking: The Right Way," *1988 FORML Conference Proceedings*

9. John R. Hayes, "ANS Forth: Hardware Independence," *Forth Dimensions,* Vol. XI, No. 4, 1990