

FlightLinux Project

Target Architecture Technical Report

Oct. 6, 2000

Revised 12/15/2001

Patrick H. Stakem
QSS Group, Inc.

Revision History

The following dates chronicle the evolution of this Target Architecture Report.

- * Oct. 6, 2000 Initial Release
- * Oct. 26, 2000 Updated Flight Processor Usage and RSDO Spacecraft OBC Characteristics Tables: Added Performance Table.
- * Oct. 30, 2000 Updated HETE-2 Information.
- * Nov. 21, 2000 Updated and Expanded Table 1:
Expanded Discussion of 0386EX
- * Mar. 6, 2001 Corrected RAD750 and Thor Results
- * Oct. 2001 Updated RAD750 and RSDO information; report edited and reformatted to be put on NASA web site.
- * Dec. 2001 Edit

1. Introduction

The intent of this technical report is to define the trends and most likely candidates for spacecraft on-board processors in the near term. The report defines architectures that the FlightLinux software effort should target and provides extensive background information for candidate architectures.

2. Background

This work was conducted under Task NAS5-99124-297, with funding by the NASA Advanced Information Systems Technology (AIST) Program, NRA-99-OES-08. Personnel of QSS Group, Inc. conducted this task in partnership with NASA/GSFC Code 586 (Science Data Systems), Code 582 (Flight Software), and Code 588 (Advanced Architecture & Automation). Work continues on this effort under Task NAS5-99124-564.

The FlightLinux project has the stated goal of providing an on-orbit flight demonstration of the software within the contract period. Numerous other Linux efforts exist within the GSFC flight software community. For example, the Triana flight code is currently being ported to Linux, and most "legacy" flight code is being examined and modified for Posix compliance.

3. Management Summary

Various microprocessor architectures have been and are being adapted from commercial products for space-flight use. For all of the primary architectural candidates identified,

Linux is available in COTS form. The primary hardware for flight computers in the near term appear to be derived from the Motorola PowerPC family (RHPPC, RAD6000, RAD750); the SPARC family (EH32); the MIPS family (Mongoose, RH32); the Intel architecture (space flight versions of 80386, 80486, Pentium, Pentium-II, Pentium-III); and the Intel ARM architecture. Versions of FlightLinux for the PowerPC and MIPS family are important goals. Because almost all of the effort in developing onboard computers for spacecraft seems to involve adapting existing commercial designs, the next logical step is to adapt COTS software such as the Linux operating system.

4. Approach

For this report, we first examined and collected a taxonomy of flight processor usage in historic, on-orbit, and planned near-term missions from several sources, including the web pages of GSFC Code 582 (Flight Software) and the Rapid Spacecraft Development Office at Goddard. These data were collected into tabular form, and trends were extracted. The processors were grouped into families based on their commercial predecessors.

In certain areas of interest, detailed information on the processor family was included in this report, along with references. The following subsection explains the role of this report in the larger context of the FlightLinux Project.

a. Steps to FlightLinux Implementation

The purpose of this subsection is to define the reports (which were produced as milestones in the FlightLinux process) and their interrelationships. The goal of the FlightLinux Project is an on-orbit flight demonstration and validation of the operating system.

We have defined the steps to a space-flight demonstration of the Linux operating system. Regardless of the implementation architecture, certain pivotal issues must be defined. This was done in a series of reports. These reference reports and ongoing research information related to the topics have been collected and can be found on the FlightLinux Project web site. The key issues include the architecture of the target systems, the nature of application software, the architecture of an onboard LAN, and the requirements for support, the architecture of the onboard storage system, the requirements for support, and the nature and design of the software development test bed.

In this *Target Architecture Technical Report*, we examine the current, near term, and projected computer architectures that will be used onboard spacecraft. From this list, we examine the feasibility and availability of Linux. The choice of the actual architecture for implementation will be determined more by opportunity of a flight than by choice of the easiest or most optimum architecture.

The *POSIX Report* examines and documents the POSIX-compliant aspects of Linux and other Flight Operating systems as well as the POSIX-compliant nature of legacy flight

application software. This is an ongoing effort by GSFC Code 582, the Flight Software Branch.

The *Onboard LAN Architecture Report* discusses the physical-level interfaces on existing and emerging missions as well as the device drivers required to support Internet Protocol (IP) over these interfaces. Ongoing work in this area is being done by the CCSDS committee and the OMNI Project (GSFC, Code 588). The choice of a demonstration flight defines which interfaces will need to be implemented first. In addition, those interfaces with COTS drivers and those for which device drivers need to be defined are delineated.

The Bulk Memory Device Driver Report defines the approach to be taken to implement the Linux file system in the bulk memory ("tape recorder") of the spacecraft onboard computer. It defines which elements are COTS and which need to be developed.

The Embedded Test Bed Report defines the requirements and architecture for the facility to develop and validate the operating system code for the flight experiment. Guidance has been drawn from similar past facilities.

These reports serve as living documents, updated as required to document new developments. A major purpose of the reports is to collect the COTS aspects of the specific aspect of the FlightLinux implementation so that attention may be focused on the remaining "missing pieces."

Table 1 shows the usage of flight processors by various operational and development missions in the near term. This information was gleaned from the GSFC Flight Software group web site and other sources. It includes missions from GSFC, JPL, NRL, APL, and ESA.

Table 1. Flight Computer Usage from GSFC, JPL, and DoD Sources

Flight Processor Usage
update 11/21/00

| <u>Mission</u> | <u>Processor</u> |
|----------------|----------------------|
| Cassini | 1750A |
| Cluster (ESA) | 1750A |
| MSTI-1,2 | 1750A |
| Rosetta (ESA) | 1750A |
| EOS Terra | 1750A (2) |
| EOS Aqua | 1750A (4) & 8051 (2) |
| EOS Aura | 1750A (4) & 8051 (2) |
| Clementine | 1750A, 32 bit RISC |
| MSTI-3 | 1750A, R-3000 |
| Pluto Express | 32 bit RISC |
| Sampex | 80386, 80387 |

| | |
|-----------------|------------------------------|
| SMEX | 80386, 80387 |
| SWAS | 80386, 80387 |
| TRACE | 80386, 80387 |
| WIRE | 80386, 80387 |
| FUSE | 80386, 80387, 68000 |
| Surrey MicroSat | 80386EX (2) |
| UoSat-12 | 80386EX (3) |
| FAST | 8085 (2) |
| HealthSat-II | 80c186 (2), 80c188 |
| PoSat-1 | 80c186, TMS320C25, TMS320C30 |
| Sojourner | 80c85 |
| Galileo AACS | ATAC (bit slice) |
| SPOT-4 | F9450 |
| EO-1/WARP | Mongoose V |
| IceSat Glas | Mongoose V |
| MAP | Mongoose V, UTM 69R000 |
| CGRO | NSSC-1 |
| Topex/Poseidon | NSSC-1 |
| UARS | NSSC-1 |
| EUVE | NSSC-1, 1750A |
| HST | NSSC-1/386, DF-224->486 |
| Coriolis | RAD6000 |
| Deep Space-1 | RAD6000 |
| Gravity Probe B | RAD6000 |
| HESSI | RAD6000 |
| MARS 98 | RAD6000 |
| MARS Pathfinder | RAD6000 |
| SIRTF | RAD6000 |
| SMEX-Lite | RAD6000 |
| Swift | RAD6000 |
| Triana | RAD6000 |
| MightySat-II | TMS320C40 (4) |

Table 2 was constructed from the Rapid Spacecraft Development Office web site. These represent quick turn-around spacecraft offerings by commercial vendors.

Table 2. Spacecraft On-board Computers from the Rapid-I and Rapid-II Programs

| RSDO | Spacecraft | OBC Characteristics |
|-----------------------|----------------------|----------------------------|
| <u>Rapid-I</u> | | |
| <u>Manufact.</u> | <u>Platform</u> | <u>cpu</u> |
| Ball | RS2000 | 32-bit risc |
| SSTL | Minisat 400 | 80386EX (2) |
| Orbital | MidStar | 80386 |
| Orbital | Starbus | 1750A |
| Orbital | PicoStar | 80C186 |
| Orbital | Ministar | 80C86 |
| Orbital | LeoStar | 80C186 |
| Orbital | MicroStar | 68302 |
| Orbital | PegaStar | 68302 |
| Swales | RapidCore | Mongoose V |
| LockMart | LM-100 | tbd |
| LockMart | LM-900 | R3000 |
| Loral | LS-400 | 1750A |
| Spectrum | SA-200 | 32 bit risc |
| TRW | SSTI | RS-3000 |
| | Step-E | -?- |
| <u>Rapid-2</u> | | |
| | (duplicates removed) | |
| Ball | BCP 600 | tbd |
| Ball | BCP 2000 | tbd |
| LockMart | LM-900 | R3000 |
| Spectrum | SA-200 S | 32 bit risc |
| Spectrum | SA-200 HP | 32 bit risc |
| TRW | T100 | 80c86 |
| TRW | T200a | 80c86 |
| TRW | T200b | 80c86 |

Table 3 shows the performance of selected flight architectures in the near term.

Table 3. Performance of Selected Flight Architectures

| <u>Processor</u> | <u>Closest Commercial</u> | <u>Clock Speed</u> | <u>mips</u> | <u>mflops</u> | <u>SPECint95</u> | <u>SPECfp95</u> |
|------------------|---------------------------|--------------------|-------------|---------------|------------------|-----------------|
| Rad750 | PPC 750 | 166 | 300. | | 7 | 4.7 |
| Pentium | Pentium-I | 166 | 178.4 | | 4.5 | 3.73 |
| RHPPC | PPC 603e | 166 | 210 | | 3.94 | 2.71 |
| Rad6000 | PPC | 33 | 35 | | 1.68 | 4.9 |
| StrongArm | SA1100 | 88 | 2.1 | n/a | | n/a |
| Thor | none | 50 | 50 | 16 | | |
| Mongoose V | R3000 | 12 | 10.6 | | | |
| 1750A | n/a | 1 | 3 | n/a | | n/a |

Certain other informational sources were considered, and that information is presented in this report.

b. SNAP-1

The Surrey Satellite Technology Nanosat Applications Platform (SNAP-1) was launched on June 28, 2000. The onboard computer (OBC) is based on Intel's StrongArm SA-1100 with 4 Mbytes of 32-bit wide EDAC protected SRAM. The error correction logic can correct 2 bits in every 8 using a modified Hamming code implementation and the errors are "washed" from memory by software to prevent accumulation from multiple single-event upsets. There is also 2 Mbytes of Flash memory containing a simple bootloader. The bootloader is either used to load the application software into SRAM or, in the case of a Linux kernel with ramdisk, to load a second-stage loader. Communication to the other spacecraft modules takes place through a Controller Area Network (CAN) interface. Two serial ports and one synchronous communications channel provide the connections to the RF modules. One asynchronous serial port (SA-1100 on-chip peripheral) is configured for reception at 9600 bps while the second is configured for transmission at 38.4 kbps. The synchronous communications channel is full duplex (configured for 9.6 kbps to receive and 38.4 kbps to transmit) and is implemented in an FPGA. No other space application of the ARM processor was found. The architecture of the ARM is discussed in Appendix A.

c. HETE-2

The HETE-2 spacecraft, a follow-on to replace the lost HETE-1 spacecraft, was launched on Oct. 9, 2000. The spacecraft computer system consists of four identical processor boards: each board contains one T805 Transputer, two Motorola 56001 digital signal processors (DSP's), and 20 Mbytes of RAM. The processors are assigned to the

spacecraft and science needs. The "links" feature of the transputer allows for quick and efficient communications between processors. The Digital signal processors (DSP's) serve as the interface to the instruments.

The embedded instrument controller for the Coronal Diagnostic Spectrometer was prototyped at GSFC using a T222 Transputer. This instrument was a part of the SOHO/ISTP (Solar Heliospheric Observatory / International Solar-Terrestrial Physics) Program, jointly funded by the European Space Agency. This work was done as part of the Laboratory for Astronomy and Solar Physics. The flight unit, using a T800 processor, was delivered for a summer, 1995 launch. The transputer serves as an embedded controller, orchestrating the operation of the 1024 x 1024 element CCD (charge-coupled device) sensing element. The data system on the SOHO spacecraft also uses transputers.

The Inmos Transputer architecture, introduced in 1985, is a single-chip microcomputer architecture, optimized for parallel use in Multiple Instruction, Multiple Data (MIMD) configurations. It provided balanced interprocessor communications as well as computational ability. The University of Surrey, U.K., has extensive flight experience with transputers.

The transputer required a minimum of external support chips. The unit performed fast, on-chip, 64-bit floating point processing and had built-in support for parallelism. The transputer had 4 kbytes of fast SRAM on-chip so that practical systems can be built with no external memory. The transputer had four bi-directional synchronous serial links built into the chip, which operate at a DMA rate of 20 Mbps each. These links allowed the transputer to be connected as building blocks into arrays of arbitrary size and complexity. Unfortunately, the transputer chip is no longer in production. Curiously, the transputer is the one processor for which we could not locate a Linux port. Minux, the original basis of Linux, was ported to the transputer, but Linux was evidently never implemented.

5. Synopsis

Early flight computers were generally custom designs, but cost and performance issues have driven the development of variants of commercial chips. Aerospace applications are usually classic embedded applications. Military systems live in their own world, and a discussion is not within the scope of this effort. Space applications are rather limited in number, and, until recently, almost exclusively meant National Aeronautics and Space Administration (NASA), European Space Association (ESA), National Aero Space Development Agency (NASDA) - Japan, or some other government agency. Flight systems electronics usually require MIL-STD-883b, Class-S, radiation-hard (total dose), SEU-tolerant parts. MIL-STD-883 is the standard for testing and screening of parts. Specific issues of radiation tolerance are discussed in MIL-M-38510. Class-S parts are specifically for space-flight use. Issues of radiation-hardened and single-event upset hardening will be discussed briefly in Appendix B of this report.

Processors used in aerospace applications, as any semiconductor-based electronics, need to meet stringent selection, screening, packaging and testing requirements, and

characterizations because of the unique environment. Most aerospace electronics, and the whole understanding of radiation effects, were driven by the cold war defense buildup from the 1960's through the 1980's. This era was characterized by the function-at-any-cost, melt-before-fail design philosophy. In the 1990, the byword was COTS -- use of Commercial, Off-The-Shelf products. Thus, instead of custom, proprietary processor architecture's, we are now seeing the production of specialized products derived from commercial lines. In the era of decreasing markets, the cost of entry, and of maintaining presence in this tiny market niche, are prohibitively high for many companies.

Given the candidate processors identified in missions under development and planned in the short term, we then examined the feasibility of Linux ports for these architectures. In every case, a Linux port was not only feasible, but was available as a COTS item. Specific Board Support Packages (BSPs) for the specific board architecture had to be written.

Existing space processors in recent or planned use include the RAD6000, the RH32, and the MIPS-derived Mongoose-V. Generally, Linux requires a memory management unit (MMU) for page-level protection as well as dynamic memory allocation. However, ports of Linux (uClinux) exist for the Motorola ColdFire processor series, and similar architectures, without memory management. The Mongoose architecture does not include memory management hardware. A Mongoose port of Linux is feasible, and this will be examined in conjunction with GSFC's Code 582, Flight Software Branch. The future usage plans of these hardware architectures will determine the direction of our efforts on the FlightLinux software ports. The RAD6000 is reported to be "a direct transfer of the IBM RISC System/6000 single-chip CPU to the Lockheed Martin radiation-hardened process." The RAD/6000 is a PowerPC-like architecture; IBM implemented their later R/6000 systems with PowerPC chips. The PowerPC architecture is the result of a joint venture between IBM and Motorola; it incorporates the instruction set of the RAD/6000 line with the RISC features of the Motorola 88k line.

Emerging space processors include Honeywell's RHPPC, the Lockheed's RAD750, ESA's ERC32, and the Sandia radiation-hard Pentium. All are viable targets for FlightLinux. The RHPPC and the RAD750 are variations of the Motorola PowerPC architecture. GSFC's Code 586 already has Linux running on the PowerPC architecture in a laboratory environment. The Intel (Pentium) version of Linux is the most common and can be found in the Code 586 lab as well. ESA's ERC32 is a variation on the SPARC architecture, and Linux is available for the Sun Sparc architecture. Table 4 shows the assessment of the Linux port feasibility for existing and emerging flight computer architectures. The label "COTS" should be taken to mean that a commercial version for that processor architecture is available. A specific port for the Flight Computer embedded board would involve coding specific device drivers, reconfiguration, and recompilation of the kernel. Linux is a 32-bit operating system, matching the emerging 32-bit class of flight computers.

Table 4. Flight Computer Linux Port Feasibility

| <u>Target</u> | <u>Base Architecture</u> | <u>Assessment</u> |
|--------------------|--------------------------|--|
| WIRE - SMEX SCS | i80386/80387 | COTS, tune for limited memory resources |
| RAD6000 | R/6000 - Power PC | COTS |
| RH32 | MIPS R3000 | COTS |
| Mongoose-V | MIPS R3000 | COTS, need to modify for lack of memory management |
| RHPPC | Power PC | COTS |
| RAD750 | Power PC | COTS |
| ERC32 | SPARC | COTS |
| Sandia Pentium | Pentium | COTS |
| Ericsson Space | Thor | COTSRTEMS |

6. Processor Discussions

6 a. NSSC-1

No discussion of flight computer usage at GSFC is complete without a mention of the NSSC-1. The NASA Standard Spacecraft Computer-1 (NSSC-1) was developed at Goddard Space Flight Center as a standard component for Earth orbiting missions. The NSSC-1 is a fixed point, 18-bit machine constructed with discrete logic components, and using core or plated wire memory. Memory was organized in 4 kbyte banks, with up to 16 banks (64k bytes) available. "Typical" CPU speed was 1.25 Mhz.

Precursors of the NSSC-1 were 1) the On-Board Processor (OBP), used on the OAO-C mission (1972), and 2) the Advanced Onboard Processor (AOP). The AOP was used on the Landsat-B and -C, IUE, and OSS spacecraft. The NSSC-1 became part of the MMS (MultiMission Modular Spacecraft). It was used on the IUE, SMM, Hubble Space Telescope, CGRO, EUVE, UARS and Landsat D and D' missions. It is still supported but will not be used for any further missions.

6b. Intel 80c86, 80c186, 80386, 80386EX (Intel ISA)

Only Intel processors of the 80386 class and subsequent classes offer memory management. Models prior to those (80c86, 80c186) do not, and, thus, are not considered reasonable platforms for Linux. A special Linux kernel that does not use memory management is known as ELKS - - Embeddable Linux Kernel System. It is also a minimal footprint system, which we have implemented on an 80386sx system with 0.5

megabyte of memory and no secondary storage. However, system software without the memory management feature is not seen as applicable for future applications.

The Intel 80x86 Instruction Set Architecture (ISA) is probably the most pervasive in today's world, and various Linux variants are available for it.

The 80386EX model includes the memory management features of the baseline 80386, and adds an interrupt controller, a watchdog timer, sync/async serial I/O, Direct Memory Access (DMA) control, parallel I/O and dynamic memory refresh control. These devices are DOS-compatible in the sense that their I/O addresses, dma and interrupt assignments correspond with an IBM pc board-level architecture. The DMA controller is, however, an enhanced superset of the 8237A DMA controller. The 80386EX processor core is static.

The 80386EX includes: two dma channels, three channels of 8254 timer/counter, dual 8259A interrupt controller functionality, a full-duplex synchronous serial I/O channel, two channels of 8250A asynchronous serial I/O, a watchdog timer, 24 lines of parallel I/O, and support for dram refresh.

6c. MIL-STD-1750A

The MIL-STD-1750A architecture has extensive heritage in aerospace missions. Avionics systems have favored the 1750A architecture, a 16-bit, non-RISC design manufactured by a variety of companies. The 1750A is an instruction set architecture, specified in MIL-STD-1750A. Different companies chose to implement different features, but the basic instruction spec is adhered to. The United Technologies UT1750AR is a RISC microprocessor that implements the 1750A instruction set through an emulation ROM. United Technologies also makes the UT69R000 microcontroller, a 16-bit RISC processor with a large number of registers, and a Harvard bus structure.

6d. Motorola 68302/68xxx

The Motorola 68302 is a derivative of the 68000 family. These units are no longer in production. A version of uCLinux may have been ported to this processor, but this was not confirmed. Versions of Linux exist for the baseline M68000 processor, particularly the 68020 and subsequent models, which support memory management. The 68302 processor does not support memory management. The M68302 is termed an integrated multi-protocol processor. Built with a M68000 core, it includes support for a variety of communications protocols. It includes three full duplex communication controllers in hardware.

6e. Mongoose V and RH32

The Mongoose V processor is a space-rated derivative of the LR-3000 processor of MIPS heritage. It includes a 4-kbyte instruction cache and a 2-kbyte data cache, as well as floating-point capability. However, the omission of the memory management unit forces

the use of a flat memory model and precludes one of the more powerful features of Unix: memory mapping.

The Honeywell and TRW RH32 were developed from a MIPS R3000 model, under sponsorship of the USAF Phillips Lab at Kirkland Air Force Base in New Mexico. It features 16 kbytes of data cache and 16 kbytes of instruction cache. It includes four serial I/O channels, four timers, a built-in 1553 bus, 40 programmable I/O lines, and DMA capability. At a module level, the Sun M-bus is supported. The module is available in 100 K rad to one megarad hardness with no single-event latchup. It incorporates IEEE-754 floating-point capability, and memory management features. The RH32 processor is an integral part of the Advanced Spaceborne Computer Module (ASCM). The RH32 supports a VxWorks operating system and the gnu-c compiler.

The architecture of the Stanford-derived MIPS machines includes the models R-2000, R-3000, R-4000, and derivatives, available from multiple manufacturers. MIPS, meaning Microprocessor without Interlocking Pipeline Stages, is a multi-sourced architecture, built to an Instruction Set Architecture (ISA) specification maintained by MIPS Computer Corporation. Kane's book (Reference 16), describing the R2000 base architecture, is considered a hardware bible for this family. Originally addressing the workstation market, the family has expanded into special embedded versions.

The MIPS design originated from John Hennessy's work at Stanford. The original MIPS processor was the R2000. It had an associated floating-point processor, the R2010. Second-generation devices include the R3000 and 3010, along with a variety of specialized embedded controllers. The next generation included the 64-bit R4000.

The R2000 has two main functional units, the integer CPU and the MMU/TLB. This latter unit is referred to as System Control Coprocessor. The R2000 implementations have a five-stage pipeline, with thirty-two 32-bit registers and use 32-bit-wide byte addresses. The translation look-aside buffer (TLB) is a 64-entry, fully associative element. Page size is 4 kbytes, and pages are identified as to read/write access permission, cacheability, and process ID. All instructions are 32 bits in length, and there are only three different formats. The architecture relies on instruction synthesis by the compiler for complex instructions. This is essentially macro expansion at the assembler level.

The MIPS R3000 is a 32-bit architecture with a separate R3010 floating-point unit. Thirty-two 32-bit registers are included on-chip. The pipeline is five stages in length. Branch latency is one clock time, and load/store latency is also one clock. The device averages 1.25 instructions per clock. Several addressing modes are supported, including base register and 16-bit signed immediate offset. The Harvard architecture chip supports caching of 4-256k instruction or data, and instruction streaming. On the main memory side of the cache, multiprocessor support is provided by provisions to ensure cache coherency.

Originally a 32-bit architecture, the MIPS specification has been expanded to include 64 bits. The R4000 is the first in the new 64-bit MIPS machines. This means that the address bus, the registers, the ALU, and all external and internal data paths are 64 bits in width. Compatibility has been maintained with the previous 32-bit members of the family at the source-code level. Additional instructions are provided to optimize certain operations.

On the MIPS, each pipeline step takes one clock or less. After the fill latency, instructions operate at one per clock. Instruction interference is minimized. Delayed loads and delayed branches are utilized to eliminate the latency inherent in these operations.

The MIPS architecture is load/store, with all ALU operations occurring between registers. The hardware is configurable at reset time to operate in a little-endian or big-endian mode. There are 32 registers, of which R0 is defined as zero on a read, and R31 holds the return address for a subroutine call. The others are general purpose. There are two special registers to provide a 64-bit holding area for multiply and divide operations.

The MIPS machines are designed to operate with one to four tightly coupled coprocessors, of which one is the integral CP0, system control coprocessor. The system control coprocessor handles the tasks of memory management, exceptions, and diagnostics. Co-processor 1 is the floating-point unit. The other units are implementation unique. The functional units may be implemented in separate chips, or may share the same chip. In user mode, half of the virtual address space is available for up to 64 user processes. In kernel (or supervisor) mode, three distinct virtual address spaces are available.

The on-chip data and instructional caches can feed two instructions and two data words every pipeline cycle. The ALU supports single-cycle execution. No restrictions are placed on instruction issue so that any two arbitrary instructions can be issued each cycle.

Upon stall, all stages of the pipeline are frozen. If necessary, incorrect results from previous erroneous steps are corrected before resumption of the pipeline operation; this operation is called correction of pipeline overrun. Upon pipeline slip, certain stages must advance to resolve data dependencies.

Internally, the MIPS memory bus interface is Harvard-built and goes to a unified (Von Neumann) format off chip. Bus width is 32-bits in the R3000 series, 64-bits in the R4000. From the processor, multiplexed address/data lines are provided. In the R3000, a six-element-deep write buffer allows for up to six single-cycle writes in a row. All instructions are 32 bits in size. There are three formats: immediate, jump, and register.

The "load/store" instructions are the only ones that reference memory, and only a single addressing mode of base register; in addition, offset is supported. Load instructions have an inherent latency of one cycle before data is available to subsequent instructions. This can cause a load delay in the pipeline. However, the pipeline always executes the instruction following the load. It is up to the compiler to ensure that the instruction following the load does not depend on the data to be loaded. If the compiler cannot

rearrange the code properly, a "nop" is inserted as a last resort. A SYNC instruction is provided to force completion of pending loads and stores before any new loads or stores are initiated. Delayed branches are handled in the same fashion.

As with the R2000 architecture, four coprocessors are supported on the R3000 and R4000. These are closely coupled, synchronous units, sharing that data bus and bus transactions with the main processor. By convention, coprocessor 0 is the System control co-processor, and co-processor 1 is the floating-point unit. In the R3000, co-processor 0 (MMU) is implemented on the same chip as the main CPU. For a co-processor load or store operation involving memory, the main CPU handles the addressing and control, and ignores the data. Data transfers between units use the data bus. The co-processor can initiate a stall of the main processor, if a data dependency needs to be resolved.

Integer math instructions for the MIPS architecture include add, subtract, multiply, and divide. The floating-point instructions, implemented in Co-processor One, include add, subtract, multiply, and divide, conversion, comparison, square root, and conditional branches. Bit manipulation instructions include AND, OR, XOR, and NOR, and various shifts are provided. For flow control, jumps and branches are used. The difference is that jumps are to an absolute address, and branches are relative to the program counter, with a 16-bit offset. Processor state control is provided by systems calls, breakpoints, and traps.

The R2010 and R3010 floating-point chips implement IEEE 32- and 64-bit arithmetic, with a simple co-processor interface to the processor to allow concurrent operation. The floating-point unit, Co-processor 1, has a simple load/store architecture, with sixteen 64-bit on-chip registers. Three internal execution units allow the device to expedite the operation of traditionally time-intensive, floating-point operations. These include the add/subtract unit, the multiply unit, and the divide unit. The unit does the four basic math operations plus format conversion. The floating-point unit has a six-stage pipeline. Instructions require from one to 19 cycles to complete in this unit, so pipeline stalls are more frequent. Instructions that don't contend for the same resources can be overlapped, but only one multiply and one divide can be running at one time. However, an add/subtract can be started and complete during a multiply.

In the 2000 and 3000, cache is external. Starting with the 4000 series, some primary cache is provided internal to the processor chip. Separate data and instruction caches are provided in the R3000. A six-element-deep write buffer allows for up to six single-cycle writes in a row. The MMU divides the memory space into a 2-gigabyte user space, and a 2-gigabyte kernel/systems space. The MMU uses a 64-entry fully associative translation look-aside buffer (TLB). Externally implemented write buffers provide decoupling of write latency from CPU operations. Both big- and little-endian data formats are supported, with the mode selected during reset.

Direct mapped external cache is used, with a line size of four words for instruction, and one or four words for data. Physical tags are kept, so that a context switch does not require a cache flush. The data and instruction caches share the same implementation. A write-through policy is used for the data cache. A cache parity error is treated as a cache

miss. Since DMA as well as multiprocessing can invalidate cache contents in a system, coherent DMA is used. For cache flushing and diagnostics, there is a provision to swap the I- and D- caches. This is controlled by a bit in the status register. A load or a store cannot be executing at the time of the swap. In addition, execution must be proceeding from a non-cacheable region of memory.

The MIPS processors were designed with multiprocessing in mind, using such features as physical cache, and a mode for externally generated stall and the ability to invalidate cache status inputs. Memory management functions are provided by co-processor zero. Multiprocessing depends on interprocessor communication, synchronization between processes, and data coherence. A mix of hardware and software is used to address these issues.

The interrupt capability of the chip is provided by six hardware pins that are sampled during the Arithmetic Logic Unit (ALU) step only. There are two software interrupts. Interrupt response has a one-cycle latency. There are no hardware interrupt vectors, and only one general exception vector. Other vectors are used for reset, and Translation Look-aside Buffer (TLB) miss. All exception cases force the processor into kernel mode. The system control co-processor handles all exception conditions and processing.

Upon an exception case, the current instruction and all subsequent instructions in the pipeline are aborted. The exception program counter is loaded with the restart address, which may be an instruction address or the address of a branch following a delay slot. Extensions to the instruction set may be provided by the reserved instruction exception and associated handler.

Although the machine state is saved, certain conditions or fragments of incomplete instruction execution may remain. For example, a multiply or divide may be in progress. The contents of the Instruction- (I-) cache may have been updated, or the cache may have been updated in response to a bus error. Thus, certain conditions, called reorganization constraints, are imprecise after an interrupt. The alternative to this is not to allow exceptions or interrupts, or to save a massive amount of state information for each exception.

6f. PowerPC

This section covers IBM-derived RISC microprocessors, starting from the RS-6000 family of computers to the PowerPC architecture. For use in space, these are implemented as the RAD6000, the RAD750, and the RHPPC.

The RAD-6000 is a port of the RS-6000 processor to an inherently radiation-tolerant technology. The MC version, shown in early 1993, is a six-chip version of the processor from the IBM RISC system 6000, model 320H. In the fall of 1993, IBM showed the SC, a one-chip version of the model 200 CPU. The radiation-hard variant was developed at the IBM facility in Manassas, Virginia, and was sold subsequently: first, to Lockheed Martin and in November 2000, to BAE Systems.

The advantage of a rad-hard version of a commercial chipset is the ability to piggyback onto the existing set of software tools, applications, and development environments. For the RS-6000, this includes the languages ADA, c++, and FORTRAN, and the AIX operating system, as well as several real-time and embedded kernels. The RAD6000 was used in a variety of military missions, including Brilliant Pebbles, part of the Star Wars ballistic missile defense system. IBM integrates it into the Advanced Spaceborne Computer Module (ASCM). It was also used on the Mars Pathfinder mission.

The RAD750 from Lockheed Martin Space Electronics (now, BAE Systems) is considered a follow-on to the RAD6000. JPL is sponsoring the development of the RAD750 single-board computer under the X2000 program. The X2000 program is a series of exploratory missions to deep space. The RAD750 is a superscalar design, with a 32-kilobyte instruction and a 32-kilobyte data cache. It includes a Memory Management Unit (MMU), IEEE floating-point support, and support for external caches. The RAD750 is based on the PowerPC 750 architecture. The RAD750 is supported by the Air Force Research Laboratory, Space Vehicles Directorate (AFRL/VS).

The Honeywell Rad-Hard PowerPC (RHPPC) derives from the Motorola PowerPC 603e and retains complete software compatibility with that part.

The PowerPC family was a joint effort between IBM, Apple, and Motorola to design and manufacture a family of RISC processors scaleable from palmtop to mainframe. The first in the series, the Motorola produced MC98601, was a 32-bit implementation of the 64-bit family architecture. The chip first appeared in 1992. The PowerPC design is a blend of IBM's RIOS processor architecture and Motorola's 88100 internal bussing. IBM and Apple laid out the basic architecture of the device in 1991.

The PowerPC chip family currently includes the 601, 603, 604, and 620. The 620 is a 64-bit architecture. The Power-PC design was influenced by the previous IBM ROMPS architecture. The newest members of the architecture are the 740 and 750 models.

The PowerPC features superscalar architecture, which allows simultaneous dispatching of three instructions into the three independent execution units. *These execution units are the integer, the floating point, and the branch-processing units.* Simultaneous instructions can execute in parallel and complete out of order, while the hardware ensures program correctness. The on-chip branch-processing unit does hardware branch prediction with reversal. Thirty-two 32-bit registers are provided, and two modes of operation (supervisor and user) are implemented.

The instruction unit contains the instruction queue and a branch-prediction unit. Unconditional branches are folded, and conditional branches result in pre-fetches from a predicted target instruction stream. The instruction queue holds up to eight instructions. The instruction decoding is hardwired.

If the branch processor can resolve a branch early, a zero-cycle branch will result. The branch-processing unit has its own adder to compute target addresses and has its own registers. The instruction unit handles memory access as well as integer ALU operations. It includes a multiplier and divider. Loads and stores can be issued back to back. The floating-point unit contains a multiply/adder, and a divider. This unit is IEEE-754-compliant and pipelined.

Integrated functions are included in embedded versions of the PowerPC. A memory management unit is provided. The memory/bus interface is Harvard architecture, with a width of 32 address, 64 data, and 52 control lines on the 501. Between the cache and the memory is a two-entry read, and a three-entry write queue. An entry is eight words. Read-before-write is supported with data coherency maintained. There is dynamic optimization and run-time ordering of the load/store memory traffic on the bus. Except for dependencies, "reads" precede "stores." Secondary cache support is provided.

The instruction set operates entirely on registers, except for the load/store. Integer math instructions include the four basic math operations on bytes, half-words, and words. Compares are also provided. Floating-point operations support single- and double-precision format; they include the four basic math operations, rounding, conversion, and compare.

Load/store supports integer or floating operands. Addressing modes are register-indirect with the options of immediate index or indexed. There are load/store multiple instructions and move strings, which can function as a software DMA. These instructions are interruptible.

Logic operations include AND, OR, XOR, NAND, NOR, extract and insert (bits), and various rotates and shifts. Flow control includes a conditional branch. Processor state control instructions allow moving data to/from the registers. There is an instruction to enforce in-order execution, by delaying subsequent executes until all previous instructions have completed to the point where no exceptions are possible.

There are 32 general-purpose registers, and 32 x 64 floating point. There is a processor ID register with a 4-bit tag for use in multiprocessing. The tag is cleared upon reset. There are condition registers, floating-point status and control, a machine-state register, and sixteen 32-bit segment registers (in the 32-bit implementations; the 64-bit implementations use a segment table). There can be implementation-unique special purpose registers

The byte-ordering format is little- or big-endian with big-endian as the default. The mode is switchable by a bit in a register (in the 601 model). In the PowerPC, 2 bits are used for the endianness of the separate supervisor and user modes. Data types supported include bytes to 64-bit double words.

The integral floating-point unit of the PowerPC handles IEEE single- and double-precision operations. The unit is pipelined with two stages, although an executing

instruction may make multiple use of these stages. It includes a single-precision multiply/add unit, a divider, and its own registers. The floating-point unit has its own dual instruction queues. These buffer instructions from the main instruction issuer, allowing other integer instructions to proceed. The unit supports the IEEE data formats, exception models, and rounding modes. The integral Floating Point Status and Control Register is 32 bits wide and contains flag bits, and enable bits. The floating-point register set consists of thirty-two 64-bit-wide registers. All floating-point operations occur between floating-point registers. The load and store operations allow for format conversion and are performed by the integer unit. Operations supported include add, subtract, multiply, divide, multiply-add, multiply-subtract, and compare.

The virtual address space is 2^{52} bytes in extent, mapped to a 2^{32} -bit real address space. The memory bus width is 64 bits in the low-end units, going to 128 bits in the higher end units. Access control is provided at the block and page levels. An eight-way set associative tag array is used in the translation. Demand paged virtual memory is supported. A block is 128k to 8 megabytes selectable in software. Instructions can be issued out-of-order but are guaranteed to produce results sequentially.

Exceptions are particularly tricky on a machine with out-of-order instruction dispatch. On the PowerPC, exceptions may be recognized out-of-order but will be handled in order. Exceptions are precise or imprecise and synchronous or asynchronous. Upon recognition of an exception, instructions in the stream are allowed to complete. A vector address table is maintained.

The PowerPC 750 is the basis of the Apple Computer G3/G4 desktop and laptop family. It is superscalar, incorporating dual integer units and an IEEE 754 floating-point unit. It includes 32 kbyte caches for data and instructions separately with level 2 cache support. The data interface is 64 bits in width with a 32-bit address. Memory management features are included. The chip has several power-saving modes selectable from software.

The basic design of the 750 model includes six execution units on the chip, and the superscalar design can complete two instructions simultaneously. Most integer instructions take one clock, and the floating-point unit is pipelined.

6g.SPARC

The SPARC chip is the basis for ESA's ERC32 space-rated processor. The Scalable Processor Architecture (SPARC) is an open architecture, based on the Berkeley RISC. In contrast to the specification level of the MIPS processor, SPARC processors are instruction-set compatible and may be hardware compatible. Multiple vendors support SPARC in different technologies. In early 1991, over 36 SPARC implementations were in existence from at least eight vendors. The SPARC 64-bit Version Nine architecture was rolled out in 1994. The ESA implementation includes an integer unit, a floating-point unit, and a memory manager.

Initial work on the flight version of the MIPS R3000 architecture took place at Sandia Laboratories with a working prototype being produced in 1989. The R3000 is favored for

a number of flight projects due to its 32-bit architecture, throughput, and large number of support tools. It is, however, somewhat power-hungry; it ripples through the system design impacting, for example, the thermal design. The R3000 has the advantage of being an approved choice for avionics use.

A classic RISC design, the SPARC provides fixed-length instructions, a load/store model, hard-wired decoding of instructions, and single-clock execution achieved by pipelining. The integer, floating-point, and memory management units may be separate or integrated. A four-stage pipeline is used, with six to 32 overlapping register windows. Each window features eight global and 24 local 32-bit-wide registers, with six control registers. A single-length load is a two-cycle instruction, while a single-store is a three-cycle instruction. A branch is a one- or two-cycle instruction. An annul bit is used for conditional branches.

The SPARC architecture defines a pipelined integer unit. The four stages of the SPARC integer pipeline are: opcode fetch, decode/operand read, execute, and writeback.

Different implementations of the SPARC architecture implement the reference differently. Recent designs have used superscalar techniques within the envelope of the reference architecture to provide enhanced performance. The SPARC includes 84 integer and control instructions; in addition, 57 floating-point instructions are provided that are all 32 bits in size. Most instructions specify three registers.

The SPARC architecture relies on the register window architecture, which provides a hardware mechanism for procedure calling, reduces access to memory, and is configurable for fast-context switching. Each window contains a number of registers. The alternative to register windowing is a flat register file.

In a flat register file model, all of the registers are visible at once; each has a unique address or identifier. In the SPARC windowing scheme, the processor state register has a Current Window Pointer (CWP) entry. If the CWP is changed by "1," the register addressing changes by "16." The CWP partitions the register file into separate sets called windows. Only one window is visible at a time. For context switching, a new set of windows is available by changing the contents of the CWP, which occurs in one cycle.

Registers are 32 bits in size, and there are 32 of these in a window. Registers are classified as ins, outs, globals, and locals. The windows are overlapped in the sense that in a "call" operation to a subprogram, the caller's "outs" are the called routine's "ins." Thus, according to a basic tenet of software engineering, we do not move information, only pointers. The architecture supports up to 32 windows. Various implementations have different numbers of windows. In a window are eight unmapped, eight out, eight in, and eight local registers. In a new window, the old "out" becomes the new "in," and eight new locals are provided, along with eight new "out."

Register windowing has some advantages for procedural languages such as c++ or small talk where the number of registers used is not known at compile time but only at run time. Here, the low overhead of context-switching has performance advantages.

There are 32 floating-point registers, not in a window format. Byte ordering in the SPARC is big endian. Supported data types are the byte half word, word (32 bits), and double word. SPARC development and test tools are generally hosted on Unix (Solaris) and include assemblers and compilers for "c." A real time operating system VxWorks is available. Emulation/simulation tools are available as In-circuit emulation (ICE) support. A debugger is available for assemble or higher order language code.

The SPARC memory management scheme includes an address space identifier bit field, which identifies the memory access types as user or supervisor, instruction or data fetches (four types). The SPARC reference MMU specifies a 32-bit virtual to 36-bit physical address space translation with support for multiple contexts, page-level-protection mechanisms, and virtual cache. Each manufacturer implements the reference MMU and some additional special or custom functions.

6h. Thor

The Saab Ericsson Space THOR processor was developed as a real-time embedded processor qualified for space use. The THOR is not based on a previous COTS product. It is a stack-oriented machine, with features for non-intrusive real-time debugging. It includes a four-stage pipeline and is a 32-bit RISC design with both integer and IEEE floating-point data support. Hardware support is included for the ADA tasking model. The processor can address 2 Gigabytes of memory; the upper half is memory-mapped I/O.

According to Saab-Ericsson, "Instead of a more traditional register-based microprocessor, this device has a stack-oriented instruction set, supporting Ada and other high-level languages and giving efficient and compact code with very efficient subprogram handling. The instruction set includes instructions for both integer and IEEE-754 floating-point arithmetic."

Saab-Ericsson has supplied the processors for numerous European space missions and for the Ariane launch vehicles.

7. References

1. Stakem, Patrick H. *A Practitioner's Guide to RISC Microprocessor Architecture*, New York: Wiley, 1996, ISBN 0-471-13018-4.
2. Tomayko, James E. *Computers in Space*, Alpha Books, 1994.
3. Tomayko, James E. *Computers in Spaceflight*, NASA Contractor Report 182505, 1988.
4. Trevathan, Charles E., Thomas D. Taylor, Raymond G. Hartenstein, Ann C. Merwarth, William N. Stewart, "Development and Application of NASA's First Standard Spacecraft Computer," *CACM*, Vol. 27, Number 9, Sept 1984.
5. Rapid Spacecraft Development Office Web Site at: <http://rsdo.gsfc.nasa.gov>
6. NASA/GSFC Code 582 (Flight Software) Web Site at: <http://fsw.gsfc.nasa.gov/>
7. HETE-2 Project Web Site: <http://space.mit.edu/HETE/>
8. Surrey Satellite Technology Web Site, <http://www.sstl.co.uk/>
9. *A Processor-Transparent On-Board Computer Architecture Using a Radiation Hard Microprocessor*, Saab Ericsson, Space AB, NRA 98-OSS-10.
10. BAE Systems, *RAD750 3U CompactPCI Board Hardware User's Manual*, Document 234A533, August 9, 2000, <http://www.baesystems-iew.com/space/index.html>
11. <http://www.sandia.gov/media/rhp.htm>
12. Marshall, Joseph R., Stuart Schimkat, Matthew W. Deeds. "Computer Systems," *Aerospace America*, December 1999, pp. 30-31.
13. "New Advanced Computer," *Hubble Facts*, FS-1999-06-009-GSFC.
14. Stiles, G. S. *How the Transputer Stacks Up to Other Processors: A Comparison of Performance on Several Application Programs*.
15. *The Military and Space Transputer Databook*, 1990, Inmos.
16. Kane, Gerry. *MIPS Risc Architecture*, Prentice Hall, 1988, ISBN 0-13-584293-X.
17. *SPARC Architecture Manual (Version 7)*, LSI Logic, 1990.
18. Agrawal, A. and R. B. Garner. "SPARC: A Saleable Processor Architecture", *Future Generations Computer Systems: FGCS*, Apr 1992, Vol. 7, Number 2/3, pp. 303.

19. Paul, Richard F. *SPARC Architecture, Assembly Language Programming, & C*, 1994, Prentice Hall, ISBN 0-13-876889-7.
20. "The SPARC Architecture Manual," *Version 8, SPARC International*, 1992, Prentice-Hall, ISBN 0-13-825001-4.
21. "The SPARC Architecture Manual" *Version 9, SPARC International*, 1994, Prentice-Hall, ISBN 0-13-099227-5.
22. *UT1750AR RISC Microprocessor Data Sheet*, United Technologies, June 1990.
23. *UT1750AR RISC Assembly Language Manual*, United Technologies, December 1991.
24. Ciecior, F., W. Arens-Fischer, H. Iglseder, E. Backhus, H. J. Rath. "A Transputer Based On-Board Data Handling System for Small Satellites." *AIAA Paper 93-4467CP*, presented at Computers in Aerospace, 1993.
25. Malhorta, M. "Fault Models of the UoSat Prototype Parallel On-Board Computer." *AIAA paper 93-4472CP*, presented at Computers in Aerospace Conference, 1993.
26. *Mongoose-V 32-bit MIPS Microprocessor Architectural Description*, rev. 1.3, Synova, Inc. Jan. 30, 1997.
27. *MPC750 RISC Microprocessor User Manual*, Motorola, MPC750UM/AD, Aug. 1997.
28. <http://www.saabericsson.space.com/thor/> (also, www.space.se)
29. *Rad Hard Thor Microprocessor Description*, Saab Ericsson Space, P-THOR-NOT-0004-SE, Jan. 20, 1999.
30. *Introducing Intel's Family of Embedded Intel 386 Microprocessors*, Intel AP499, Feb. 1994.
31. *Next Generation Space Telescope (NGST) Requirements Document for Single Board Computer (SBC)*, NGST-RQMT-000752 (Draft), NASA/GSFC, Sept. 30, 2001.

Appendix A. Architectural Overview of the Advanced RISC Machine

This section discusses the Advanced RISC Machine or 'ARM' (nee, Acorn RISC Machine). ARM claims the distinction of having the first commercial RISC microprocessor, circa 1985. It is, in the US, not a well-known or popular design. ARM processors represent a non-traditional RISC design, optimized for low-power consumption. Their high-power efficiency gives them a serious edge in battery-powered portable equipment. ARM currently has one of the best MIPS/watt ratings in the industry. The Intel variation of the ARM processor, the DEC-designed StrongARM, is used on the SNAP-1 Spacecraft, from Surrey Satellite Technology. It is also found in personal digital assistant (PDA) hand-held devices, and the recently announced IBM Linux wristwatch.

The ARM design is the result of an effort by Advanced RISC Machine, Ltd., a joint venture of Acorn Computers (U.K.), Apple, and VLSI Technology. The actual fabrication of the devices is done by VLSI, Plessey, Intel, and others. The major applications for the device are in 32-bit embedded control, and the portable computing market, including Apple's 'Newton' palmtop product.

'ARM' describes a family of processors, 32-bits in word size. There are only 10 basic instruction types. On-chip will be found a 16-element register set, a barrel shifter, and a hardware multiplier. According to the designers, the ARM is easier to program in assembly language than most other RISC processors. The ARM is designed as a static device, meaning that the clock may be arbitrarily slowed or stopped, with no loss of internal state. This also affects power consumption, which is specified at 1.5 mA per Mhz. for the processor core and is currently one of the lowest in the industry. The ARM is also available as an ASIC macrocell, allowing integration into systems at the chip level. The ARM6 core has 37,000 transistors.

The architecture of the ARM600 processor was driven, in part, by Apple's requirements. It has a 32-bit address bus and supports virtual memory. The hardware is optimized for applications that are price- and power-sensitive. The processor supports both a user and a supervisor mode. Not a Harvard design, the ARM caches data and instructions together on-chip. However, it has a 64-way set-associative cache with 256 lines of four words each. The cache is virtual, and the memory management unit must be enabled for caching to become effective. A cacheable bit allows the I/O space to be marked as not cacheable.

The architecture is load/store with no memory reference instructions. The load/store operand is a register (32-bit) or immediate. These operations may specify operand increment or decrement, pre- or post-operation. There is a load/store multiple feature, essentially a block data transfer. But it effects interrupt response, because it is not interruptible. A three-operand format is used. The hardware includes a barrel shifter that one operand always goes through. A barrel shifter is a combinatorial circuit that takes no clock cycles for its operation. The instruction execution process is pipelined.

Multiply and multiply/accumulate operations take up to 16 cycles. Although the instruction encoding only allows 16 registers to be addressed, the instruction format

allows a complete orthogonal encoding of ALU operations. An integer multiply/accumulate instruction is included. The instruction encoding, which is very "microcode"-like, gives a very large number of possible instruction cases with the conditional execution feature. Floating-point operations are not supported. The processor provides a complete co-processor interface. Request lines for two interrupts are included.

The chip uses Von Neumann architecture with only one address space. Data width is 32 bits for data, with 32-bit address on the 620 model and subsequent. A write buffer is included with room for two pending writes of up to eight words. Used with the write-through cache, this feature allows the processor to avoid waiting for external memory writes to complete. Reset initialization provides for the execution of NOP's when activated and for going to the reset vector address when deactivated.

All instructions are 32 bits in size. An interesting feature is that all instructions are conditionally executable -- not just the branches! This means that the programmer does not need to conditionally branch over instructions; the instruction itself is conditional. This feature is under program control via setting of an "S-bit." Conditions for execution include the cases 'always' and 'never.' Instructions have three operand references, two sources, and one destination. Integer math operations include add, subtract, and multiply. No floating-point instructions are provided. Load/store operations provide memory access, and a block transfer is provided that can be aborted. There are AND and XOR operations as well as bit clears and compares. Flow control is accomplished by 'branch' and 'branch and link.' A software interrupt mechanism provides for supervisor service calls.

The ARM architecture provides visibility and use of twenty-seven 32-bit registers. The ARM620 has thirty-seven 32-bit registers in six overlapping sets, and six modes of operation. Although the registers are general purpose, Register 15 is the program counter. This has some advantages and disadvantages. If we use Register 15 as the destination of an ALU operation, we get a type of indirect branch free. However, this approach doesn't support delayed branches, which require two program counters for a short while.

Register 14 holds the return address for calls and is shadowed in all cases. The stack pointer is generally held in Register 13, with Register 12 being used for stack frame pointer. Interrupts and the supervisor mode have private stack pointers and link registers.

In terms of byte ordering, the 610 is considered little endian, but the 620 can operate in little-or big-endian mode. Supported data types include bytes and 32-bit words. The cache is a 4-kbyte unified organization in the ARM620. It is a 64-way set-associative unit with a line size of 16 bytes. Multiprocessor support is provided by semaphores. Write-through policy is used. Memory can be marked as not cacheable in control Register 3, allowing for the memory mapped I/O region.

The ARM architecture implements facilities for object-oriented programming, including memory protection strategies; it has features for real-time concurrent garbage collection. The memory management unit controls memory access permissions, with tables in main

memory, and a 32-element translation look-aside buffer on chip. Pages can be 1 megabyte and 4 or 64 kbytes in size. Access permissions are mapped separately and manipulated independently of addresses. Address faults are handled separately from permission faults. Semaphore operations for multiprocessing are implemented by means of indivisible read-lock-write bus cycles. There are two levels of access permission maintained. The first is straightforward, but the domain level is a differentiator for the ARM MMU. Domain access, maintained in a register, is defined as the four cases: no access, client, reserved, or manager. For the 'client' case, access permission checking is traditional and straightforward. For the 'manager' case, an override of the encoded permission allows unrestricted access to the section; this facilitates the task of a garbage collector. The domain concept allows the privilege bits in the access control register to override both levels of protection encoded in the descriptor fields. Of course, access to the control register is a restricted operation.

There are two interrupt pins: regular, fast, and two corresponding modes. In fast mode, there is a 2.5 clock interrupt latency best case. Exception vectors are used. A software interrupt instruction is included, and execution of an undefined or unsupported instruction causes an exception (abort mode) that may be used for software emulation. Exceptions can also occur due to internal events, such as undefined instruction. An abort pin provides the MMU with a signal to the processor to indicate a memory access problem. Worst case latency in the fast interrupt case is 25 cycles.

ARM Bibliography

- 1) "Acorn Risc Machine Data Manual," *VLSI Technology*, 1990, Prentice Hall, ISBN 0-13-781618-9.
- 2) Pountain, Dick. "ARM600: RISC Goes OOP," *Byte*, Dec 1991.
- 3) *ARM600 Data Booklet*, VLSI, June 1992.
- 4) *VL86C010 32 bit RISC PU and Peripherals User's Manual*, VLSI Technologies, Inc., 1989, Prentice-Hall, ISBN 0-13-944968-X.
- 5) Case, Brian. "Arm-600 Targets Low Power Applications," *Microprocessor Report*, Vol. 5, No. 23, Dec 18, 1991.
- 6) Case, Brian. "ARM Architecture Offers High Code Density," *Microprocessor Report*, Vol. 5, No. 23, Dec 18, 1991.
- 7) *VL86C010, 32-bit RISC Microprocessor*, VLSI Technology.
- 8) *VL86V010, An Affordable 32-bit RISC Microprocessor System*, VLSI Technology.
- 9) *Intel StrongARM SA-1100 Microprocessor DataSheet*, 1998.

Appendix B. Radiation Hardness Issues for Space Flight Applications

NASA Goddard has an ongoing program to test and characterize the radiation tolerance of electronic components, including processors. The Radiation Physics Group maintains a website on the topic.

A complete discussion of the physics of radiation damage to semiconductors is beyond the scope of this document. However, an overview of the subject is presented. The tolerance of semiconductor devices to radiation must be examined in the light of their damage susceptibility. The problems fall into two broad categories, those caused by cumulative dose, and those transient events caused by asynchronous very energetic particles, such as those experienced during a period of intense solar flare activity. The unit of absorbed dose of radiation is the rad, representing the absorption of 100 ergs of energy per gram of material. A k-rad is one thousand rads. At 10k rad, death in humans is almost instantaneous. One hundred k-rad is typical in the vicinity of Jupiter's radiation belts. Ten to twenty k-rad is typical for spacecraft in low Earth orbit, but the number depends on how much time the spacecraft spends outside the Van Allen belts, which act as a shield by trapping energetic particles.

Absorbed radiation can cause temporary or permanent changes in the material. Usually, neutrons, being uncharged, do minimal damage, but energetic protons and electrons cause lattice or ionization damage in the material, and resultant parametric changes. For example, the leakage current can increase, or bit states can change. Certain technologies and manufacturing processes are known to produce devices that are less susceptible to damage than others.

Radiation tolerance of 100 k-rad is usually more than adequate for low Earth orbit (LEO) missions that spend most of their life below the shielding of the Van Allen belts. For Polar missions, a higher total dose is expected, from 100k to 1 megarad per year. For synchronous, equatorial orbits, that are used by many communication satellites, and some weather satellites, the expected dose is several k-rad per year. Finally, for planetary missions to Venus, Mars, Jupiter, Saturn, and beyond, requirements that are even more stringent must be met. For one thing, the missions usually are unique, and the cost of failure is high. For missions towards the sun, the higher fluence of solar radiation must be taken into account. The larger outer planets, such as Jupiter and Saturn, have large radiation belts around them as well.

Cumulative radiation dose causes a charge trapping in the oxide layers, which manifests as a parametric change in the devices. Total dose effects may be a function of the dose rate, and annealing of the device may occur, especially at elevated temperatures. Annealing refers to the self-healing of radiation induced defects. This can take minutes to months, and is not applicable for lattice damage. The total dose susceptibility of the Transputer has been measured at 35-50 k-rad with no internal memory (ref. 14). The internal memory or registers are the most susceptible area of the chip, (3 k-rad - ref. 8) and is usually deactivated for operations in a radiation environment. The gross indication of radiation damage is the increased power consumption of the device, and one researcher

reported a doubling of the power consumption at failure. In addition, failed devices could operate at a lower clock rate, leading to speculation that a key timing parameter was being effected in this case.

Single event upsets (seu's) are the response of the device to direct high energy isotropic flux, such as cosmic rays, or the secondary effects of high energy particles colliding with other matter (such as shielding). Large transient currents may result, causing changes in logic state (bit flips), unforeseen operation, device latchup, or burnout. The transient currents can be monitored as an indicator of the onset of SEU problems. After SEU, the results on the operation of the processor are somewhat unpredictable (ref.). Mitigation of problems caused by SEU's involves self-test, memory scrubbing, and forced resets.

The LET (linear energy transfer) is a measure of the incoming particles' delivery of ionizing energy to the device. Latchup refers to the inadvertent operation of a parasitic SCR (silicon control rectifier), triggered by ionizing radiation. In the area of latch-up, the chip can be made inherently hard due to use of the Epitaxial process for fabrication of the base layer (ref. 12). Even the use of an Epi layer does not guarantee complete freedom from latchup, however. The next step generally involves a silicon on insulator (SOI) or Silicon on Sapphire (SOS) approach, where the substrate is totally insulated, and latchups are not possible.

In some chases, shielding is effective, because even a few millimeters of aluminum can stop electrons and protons. However, with highly energetic or massive particles (such as alpha particles, helium nuclei), shielding can be counter-productive. When the atoms in the shielding are hit by an energetic particle, a cascade of lower energy, lower mass particle's results. These can cause as much or more damage than the original source particle.

Bibliography

- 1) "An Upper Bound Estimate of Heavy Ion-Induced Soft Error Rates in CMOS Devices Based on a Experimental Cross Section versus LET Curve", April 1990, Edmonds, JPL, NPO-17566/7071.
- 2) "Rad-hard ICs", Paul D. Coe, Electronic Products, Oct. 1990.
- 3) "Rad-Hard Technology: Shaking Down, Shaping Up", Lisa Burgess, Military & Aerospace Electronics, Nov. 1990.
- 4) "Failure of CMOS Circuits Irradiated at Low Rates", C. Goben, W. Price, NASA/JPL, NASA Tech Brief Vol. 14, No. 8, JPL Invention Report NPO-17867/7361, Aug. 1990.
- 5) <http://radhome.gsfc.nasa.gov/top.htm>

- 6) "Detection of Upset Induced Execution Errors in Microprocessors", Khan and Tront, IEEE Computer Society, 1989, in Eighth Annual International Phoenix Convergence on Computers and Communications Proceedings.
- 7) Elder, J.H.; Osborn, J.; Kolasinski, W. A.; "A method for characterizing a microprocessor's vulnerability to SEU", IEEE Transactions on Nuclear Science, Dec 1988 v 35 n 6.
- 8) " Report on the suitability of the Inmos T222 and C011 for use in the Cluster Mission Radiation Environment", Feb. 7, 1990, Thompson & Hancock, University of Sheffield, UCL.
- 10) "Total Dose Radiation Tests on Inmos T800, Final Report", M. Lopez Cotarelo, B. Johlander, 4 Sept. 1989, ESTEC.
- 11) Hass, K.J.; Treece, R.K.; Giddings, A.E.; "A Radiation-hardened 16/32-bit Microprocessor", IEEE Transactions on Nuclear Science, Dec 1989 v36 n6.
- 12) "Trends in Parts Susceptibility to Single Event Upset from Heavy Ions", Donald K. Nichols, William E. Price, W. A. Kolasinski, R. Koga, James C. Pickel, James T. Blandford, Jr., A. E. Waskiewicz, 1986.
- 13) Estimating Rates of Single-Event Upsets, J. Zoutendyk, NASA Tech Brief, Vol. 12, No. 10, item #152, Nov. 1988.
- 14) "The SEU and Total Dose Response of the Inmos Transputer", J. Thomlinson, et al, IEEE Transactions on Nuclear Science, Vol. NS-34, No. 6, Dec. 1987.
- 15) Shaefer, D. L.; Kimbrough, J. R.; Denton, S. M. "Microprocessor and R3010 Floating Point Unit", IEEE Transactions on Nuclear Science, DEC 01 1991 v 38 n 6 Page: 1421.
- 16) "Estimating the On-orbit Single Event Upset Behavior of a MIPS R3000 Microprocessor," D. Hail, Harris Corp., 2/91, (GSFC study).
- 17) "Single Event Test Method and Test Results of an Intel 80386", IBM Corp, 1989.
- 18) Shaeffer, D.L.; Kimbrough, J. R.; Denton, S. M., "High Energy Proton SEU Test Results for the Commercially Available MIPS R3000 Microprocessor and R3010 Floating Point Unit," IEEE Transactions on Nuclear Science, Dec 1991 v 38 n 6 Page: 1421.
- 19) Sexton, F. W.; Treece, R. K.; Hass, K. J., "SEU Characterization and Design Dependence of the SA3300 Microprocessor," IEEE Transactions on Nuclear Science, Dec 1990 v 37 n 6.