

LEON 3FT Processor Based Design for Spacecraft Applications - Frequency Based Performance Analysis

Shruthi N¹, C K Vinay²

¹(Electronics & Communication Engineering, APSCE/VTU, INDIA)

²(Application Services, Hewlett-Packard, INDIA)

ABSTRACT : It is evident that the LEON architecture is widely utilized for spacecraft applications which demand stringent real-time execution-time guarantees. The work carried out in this paper is basically an analysis of performance of the LEON 3FT processor at different operating frequencies. Previous works have by and far proved that the 32-bit LEON 3FT processor based designs gave better performance efficiencies in terms of execution speed when compared to the 16-bit MA 31750 processor. The LEON 3FT based hardware implementation has been set up and efficiently tested. A set of selected benchmark programs have been executed on an AOCS (Attitude and Orbit Control Subsystem) practical test bench to track the execution times of the processor at different operating frequencies. A possible list of real-time tasks occurring in a typical spacecraft application constitutes the benchmark suite. The end results of this paper show a remarkable execution time when compared to one of the processors among LEON's predecessors. The code execution time increases with additional complex code logic. Using a higher order clock frequency in the hardware definitely reduces execution time in spite of the additional complex logic. However, the results indicate the maximum possible frequency up to which the hardware responds effectively in a radiation-prone space environment. The future work of this paper mainly revolves around accelerating this limiting value of frequency of operation without affecting the performance of the system subject to modifications in power constraints of the system.

KEYWORDS: LEON 3FT, performance analysis, benchmark programs, computational capabilities, code execution times.

I. INTRODUCTION

The diverse improvement in the field of processor technology has accelerated the advancement towards miniaturization, performance and operation control of the processors being used in real-time systems. The latest spacecraft applications have started implementing the LEON 3FT processor which is much more superior when compared to its predecessors. An end-user can benefit from the widespread features of the 32-bit LEON processor like enhanced data handling capability, throughput, reduced chip size and hence weight, better power saving features and data handling capability. Results show a gigantic leap in execution time. The work in this paper also tries to reveal the optimum operation of the LEON 3FT processor at varying frequencies. The processor fails to get connected to the hardware after a particular value of frequency. Section II highlights the features of the 32-bit processor, which provide extensive support to the design being implemented later in Section III. Additional code logics bring down the execution speed of the processor. The user thought of an alternate solution to minimize the execution time and yet implement the additional complex code logics. This formed the foundation for the current paper. Section IV emphasizes on the behaviour of the hardware setup subject to varied frequencies. Even though the theoretical limiting frequency value promises optimum operation, it is not easy to witness the same practically. The work carried out under the surveillance of this paper demonstrates the variation of hardware results with changing frequencies. The future work of this paper can supposedly be an attempt to accelerate the limiting frequency value to the higher side and yet maintain the processor performance. Section V discloses the test environment followed by the test results in Section VI. There is a mention of the conclusion of this paper and a proposal to continue future work as an extension to the current implementation.

II. THE LEON 3FT PROCESSOR – AN OVERVIEW

LEON 3FT aims at providing support to work in harsh, highly variant space environments. The LEON 3FT is a monolithic, SPARC V8 processor which follows the RISC based Harvard architecture and projects a 7-stage instruction pipeline with a predominant fault-tolerant feature. The European Space Research and Technology Centre (ESTEC) originally designed the LEON family of processors. This research was funded by the European Space Agency (ESA), in the year 1997, but was later handed out to Gaisler Research (Aeroflex

Gaisler), which is suitable for system-on-chip (SoC) designs. LEON 3FT is the fault tolerant version of LEON facilitates work in harsh, highly variant space environments. The Debug Support Unit (DSU) provides complete access to all processor registers and cache memory, it acts as an interface between the processor core and the peripherals. User can set instruction breakpoints and perform single stepping through the DSU. The DSU acts as an Advanced High Speed Bus (AHB) slave and can be accessed by any one of the following AHB masters: JTAG port, PCI port or a Space Wire link using RMAP protocol. A block diagram of LEON3 architecture is shown in Fig. 1.

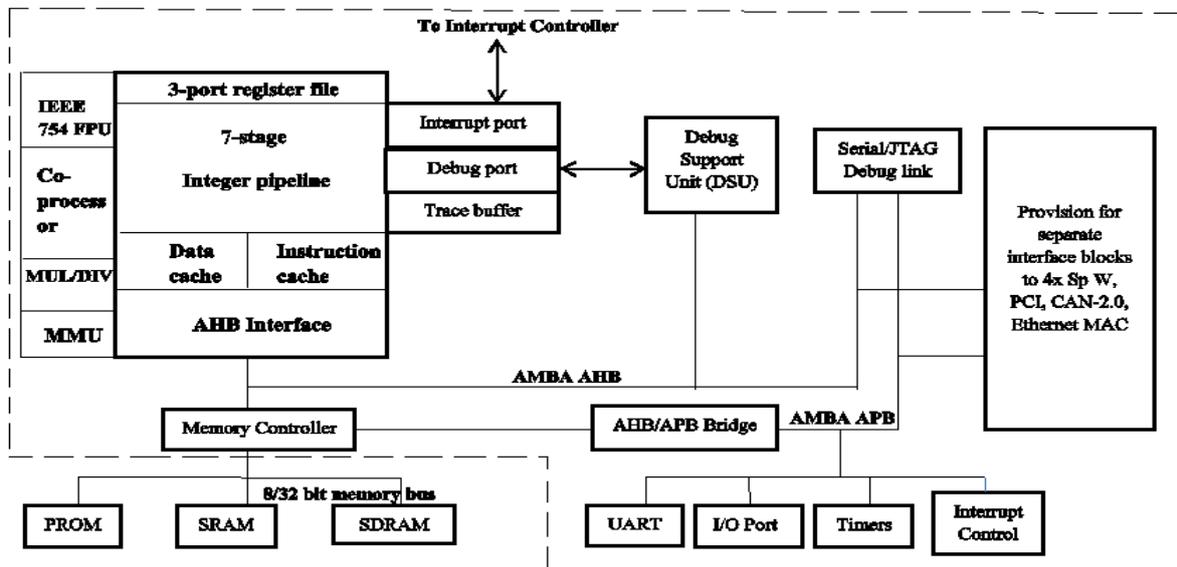


Fig. 1 – Functional Block Diagram of LEON 3FT processor

An Advanced Peripheral Bus (APB) caters to the slower devices like the UART, Timers, I/O port and the Interrupt Controller. Both integer and floating point calculations can be carried out simultaneously without affecting each other's operation and this, is made possible by the existence of two separate calculation units – the Integer unit (MUL/DIV) and the Floating Point Unit (FPU) [2]. The FPU contacts the processor core through GRFPC, a FPU controller.

III. HARDWARE IMPLEMENTATION

Prior to materializing the hardware setup as a flight model, which has been considered in our design, the design of the CPU system for spacecraft applications is implemented on a proto board, the four major modules of the proto board being the I/O & Power Supplies, Processor, 1553 communication and Memories. The 1553 communication unit has not been implemented in this paper. It can be carried out as future work which can be accomplished with additional arbitration logic. The features of the then proposed designs which have been implemented in our portion of the work are as follows: (1) Usage of high density EEPROMs for software storage and minimal PROM for Boot Code. (2) Boot code can perform EEPROM update by telecommands in addition to BOOT functions. (3) Program execution from SRAM for faster execution, EEPROM being slow and more susceptible to upsets. (4) 16-bit level translators to interface to external +5V I/O bus – to achieve +3.3 V for I/O supply and +2.5 V for the core supply. (5) Software tool set that supports both ADA & C (cross-compilation).

The design implementation diagram is shown below in Fig. 2 followed by a brief explanation.

- a) **Bus** - The external memory address bus is 28-bit wide whereas the data bus is 32-bit wide with 8 check bits. The control bus comprises of dedicated chip select, read and write signals for various memory blocks.
- b) **Memory** – Memory Controller acts as the interface between memory (PROM, EEPROM and SRAM) and the AHB bus. Requirement is 32K × 32 of boot PROM, 512K × 32 of EEPROM & 1.5M × 32 of SRAM and all three can be EDAC protected using (39, 7) BCH code, nevertheless, PROM does not require EDAC implementation. PROM is 3.3V operated. It is slower than SRAM and has the boot program residing on it. We observe that even though the OBC PROM requirement is only 32K × 32, the configuration is 32K × 40, since EEPROM requires additional 8 bits (512K × 40) for implementing the EDAC logic and both EEPROM & PROM share the same memory bank with respect to the processor; we extend a common configuration of 40 for

both PROM & EEPROM. EEPROM is slowest & more vulnerable to disturbances. Therefore, there arises a need to load the program from EEPROM to SRAM. One can interface up to 1GB SRAM externally through the Memory Controller. The SRAM area is divided into five RAM banks and the size of each bank can be set as varying from 8Kbyte to 256 Mbyte through the MCFG2 configuration register. A SRAM read constitutes two data cycles and 0-3 wait states. PROM is fabricated with QML-qualified radiation-hardened technology and is designed for use in systems operating in radiation environments. PROM operates over the full military temperature range, requires a single 3.3 V \pm 5% power supply, and is available with TTL-compatible I/O. Power consumption is typically 15mW/MHz in operation and is less than 10mW/MHz in the low power- enabled mode. The PROM operation is full asynchronous, with access time < 60 ns.

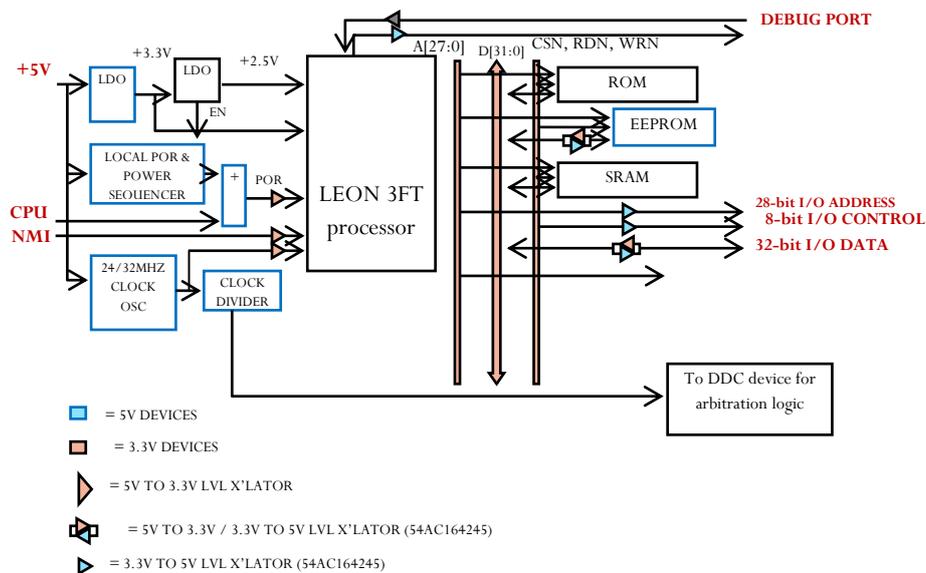


Fig. 2 – Design Implementation Diagram

c) **Level Translators** – 5V to 3.3V translators are implemented using 54LVTH162244. On the other hand, 5V to 3.3V and vice versa translations are implemented using 54AC164245.

d) **Power on Reset Generation & Power Sequencing** - The first start-up is the I/O (requiring 3.3V) and next is the Core (requiring 2.5V). Proper power sequencing of the processor is achieved by bringing up VDD to its recommended minimum operating voltage of 3.0V, and then delaying tVCD clock cycles before bringing up the VDDC supply. If power is applied to the VDDC supply pins while VDD is less than 3.0V, excessive current or damage to the device could occur. The design was proposed to implement suitable delays using RC based circuits. The delay values in the design for I/O, core signals and the RESET signals in this design are 10ms, 15ms and 25ms. A typical scenario is depicted in Fig. 3 which displays the actual delay values obtained from the hardware setup.



Fig. 3 – Power On Reset Generation & Power Sequencing

e) **Clock** – The 16-bit processor competed with a frequency of 12 MHz, the current 32-bit processor promising a theoretical capability of nearly 48 MHz. However, at higher frequencies, power consumption increases and also, board design is complex. So, a maximum frequency of operation of 32MHz is chosen.

f) **Low Dropout Regulators** - Low dropout regulators are used to provide regulated voltages to I/O and the core of the processor. LDOs improve transient response. The advantages of a low dropout voltage include a lower minimum operating voltage, higher efficiency operation and lower heat dissipation. LDO regulator is a DC linear voltage regulator which can operate with a very small input–output differential voltage and provides output voltages of 1.5V, 1.8V, 2.5V & 3.3V with 1.21V reference voltage. The only disadvantage of LDOs is their weight. Normal regulators cannot be used.

IV. HARDWARE TESTING

Some preliminary hardware testing procedures are inevitable. The software residing on the LINUX workstation establishes connectivity with the external hardware setup by means of the debug monitor GRMON through a simple serial link. The test environment for execution of spacecraft application oriented ADA codes is discussed in detail in Section V. Once the LINUX workstation hosting the ADA code is powered on, an increase in current values in the circuit can be witnessed. The card is powered on to display at least around 400 mA. After ensuring that the serial port cable is connected, GRMON is used to connect to the target. As discussed before, GRMON connectivity can be either through direct commands in the GRMON window, or the GDB protocol or the GRMON RCP Client. As soon as GRMON attaches to the proto board, current shoots up considerably. The execution of the ADA code will now be indicated by a relevant LED glow, with a further increase in current value. The above procedure was started with a 24 MHz clock frequency on board. Further, a 32 MHz clock and 36 MHz replaced the previously used 24 MHz clock, followed by a 40 MHz crystal. The purpose of increasing the crystal frequency is to increase the clock rate and hence reduce the code execution time. A clamp meter can be wound around to tap current values at desired circuit junctions. Current values increase with increase in clock frequency. It is evident that in a precise crystal resonator, the crystal electric current, otherwise termed as the drive level also influences the oscillator frequency [3]. It is evident that the CPU execution time is the product of CPU clock cycles for the program and the clock cycle time. As frequency increases, the molecules of the crystal used vibrate more number of times when compared to the lower frequency crystal and hence the number of clock cycles shoots up. Since clock cycle time is the reciprocal of the clock rate, we can conclude saying that clock rate reduces at higher frequencies to decelerate program execution time.

V. TEST ENVIRONMENT

This section intends to provide a brief description of the various tool sets and utilities as well as the development environment in which the tests related to this paper were carried out. All tests were carried out on the LEON 3FT core, SPARC V8 compatible processor. The compilation and simulation support is provided by GNAT Pro from AdaCore. GNAT Pro is a compiler and software development tool set for ADA programming language in a cross- compilation environment. By default, it assumes ADA 95 however it may be implemented in ADA 83/2005/2012 as well. All testing tools are installed on a LINUX workstation; GRMON is a general debug monitor for the LEON processor, and for SOC designs based on the GRLIB IP library. The monitor connects to a dedicated debug interface on the target hardware, through which it can perform read and write cycles on the on-chip bus (AHB).

The debug interface can be of various types: the LEON2 processor supports debugging over a serial UART and 32-bit PCI, while LEON3 also supports JTAG, Ethernet and SpaceWire debug interfaces. On the target system, all debug interfaces are realized as AHB masters with the debug protocol implemented in hardware. GRMON can operate in two modes: command-line mode and GDB mode. In command-line mode, GRMON commands are entered manually through a terminal window which is at the memory level and might not be very useful. In GDB mode, GRMON acts as a GDB gateway and translates the GDB extended-remote protocol to debug commands on the target system. The programmer can examine the program variables which prove to be much more convenient for usage and debugging. In our test setup, GDB gateway has been utilized. ADA codes get executed in the same environment. By default, serial debug link of baud 115200 is used. UART1 of the host (tty0 or COM1) is being used. The test set up is represented diagrammatically as in Fig. 4.

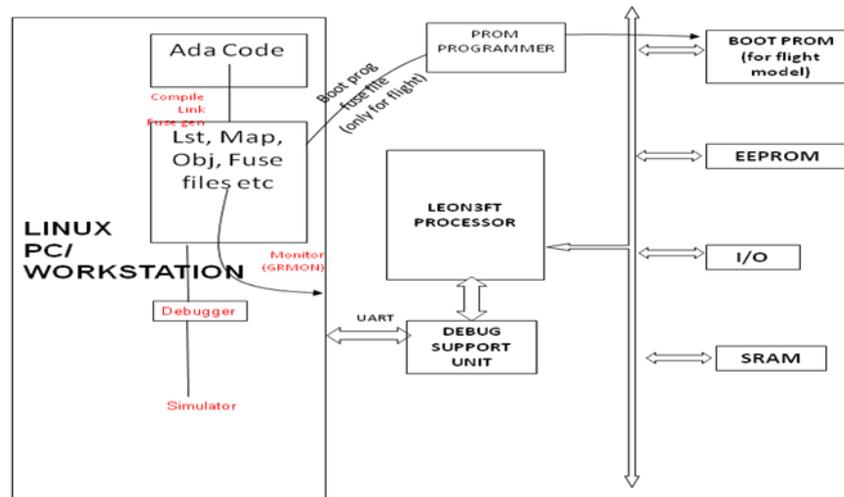


Fig. 4 – Test Setup

The compiler and debugger actions are carried out using the following set of commands.

grmon -gdb

The programmer is expected to obtain the dump file using the command:

leon3-elf-objdump -d filename | tee filename.dump

Attaching to GDB and debugging through the GDB protocol and code execution:

leon3-elf-gdb filename

target remote: portaddress,

load filename

The dump file reveals the generated assembly level code from which one can calculate the total execution time of a specific computer program. The compile, link and fuse files were loaded on to the proto board with the help of a debug monitor tool – GRMON. This tool controls the program execution on the proto board for LEON processors. GRMON communicates with the LEON processor through a gateway section, the non-intrusive debug support unit (DSU) of the LEON system. The GRMON architecture shows the command layer as an integration of both Basic commands and GDB protocol. This facilitates the programmer with two ways to provide inputs to GRMON. User can either provide a set of GRMON commands from the terminal or remotely connect to the GNU debugger.

VI. RESULT ANALYSIS

We understand that the results of the entire set-up can be obtained when a related software code runs and executes fine on the hardware support established. The current space application certifies the workability of the proto board after executing a predefined benchmark ADA code on it. The ADA code, in itself contains all vital space application logics. As discussed before, GRMON debug monitor supports the system by loading this ADA code on to the processor residing on the proto board. Prior to the implementation of the 32-bit LEON 3FT processor, the same ADA code was even executed on the previously used 16-bit MA 31750 processor. For the purpose of simplicity, logics covering quaternion multiplication and norm calculations were only executed instead of the entire program module and a main program calls these logics. The logics may also be called within multiple looping constructs for the purpose of testing complex looping times. This main program is a simple toggle generating code, henceforth one can expect to see the output in the form of a periodic square wave on the CRO as depicted in Fig. 5.



Fig. 5 – Output waveform in response to ADA code enclosing complex logics

Either the positive pulse width (T_{on}) or the negative pulse width (T_{off}) is the execution time. As the programmer inserts much more complex logic statements into the code, the enhanced code complexity is demonstrated by increased pulse width of this waveform. With an objective to tabulate a fine set of readings for many more computational logics that may possibly be used in the spacecraft application, the programmer can execute each of the logics as separate entities and note the execution time and the possible number of instruction cycles consumed by each of the functions when executed on the LEON 3FT processor. Typically, the execution time will just be a few tens of microseconds as against a few milliseconds taken by a non-LEON 3FT processor. This is a very significant performance elevation in terms of execution speed. On the other hand, increase in clock frequency of the operating hardware reduces the code execution time. We, as programmers tend to conclude that code complexities can still be handled effectively without enhancing the code execution time by using higher order clock frequencies in the hardware setup. However, there definitely exists a limit on the frequency that shall be used. High frequency operation definitely consumes lot of power and may cease circuit operation after a saturated frequency value is reached. The datasheets of the LEON 3FT processor promises a 53 MIPS and 66 MHz clock operation. But, the debug monitor fails to connect to the hardware at 40 MHz.

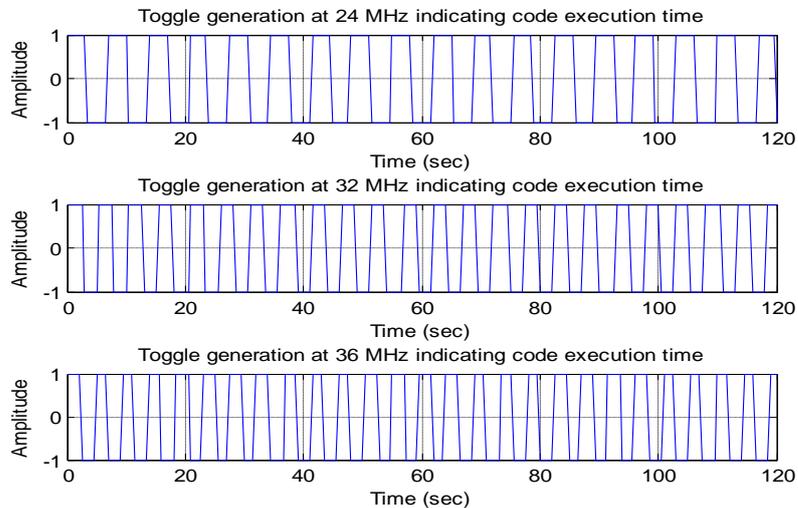


Fig. 6 - Output waveforms for varying crystal frequencies indicating code execution times

As a part of the result, the time period of the generated square wave for different frequencies of operation have been presented in the form of a graph, T_{on} and T_{off} being almost symmetrical values. All complex logics related to the relevant spacecraft application were enclosed within the main ADA code. The user noted a time period of 6.852 seconds at 24 MHz, 5.14 seconds at 32 MHz and 4.568 seconds at 36 MHz. GRMON failed to attach to the hardware at 40 MHz.

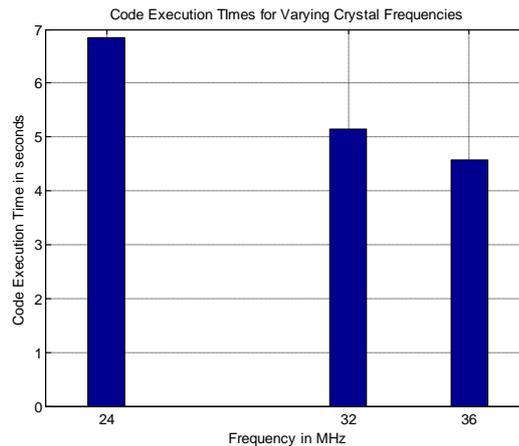


Fig. 7 – Graphical Comparison of code execution times for varying frequencies

The code execution times at varying crystal frequencies have been represented diagrammatically with the help of MATLAB simulations as in Fig. 6 and Fig. 7. These values confirm the related work to have been handling complex code logic much more quickly when a higher frequency crystal is used for our application.

VII. CONCLUSION & FUTURE WORK

In this paper, the design and implementation of CPU for spacecraft applications using a 32-bit processor is presented. The superseding processor indeed advertises an enhancement in terms of execution speed, operation in HiRel environments, fault tolerance, multiple options for external communication, power and timing constraints and many more. However, even the LEON processor lacks the performance counter feature. This missing feature of the LEON processor has been imbibed by presenting a performance-monitoring unit [4]. The scope of this paper mainly revolves around the execution speed of the processor along with testability of the hardware setup for different values of clock frequency. The successful workability of the proto board bearing our design is verified by the benchmark suite. A remarkable leap in execution speed has been achieved with the LEON 3FT compatible superior processor. LEON 3FT processors have already joined the league of super processors. Its reduction in chip size and a 53 MIPS throughput via 66 MHz base clock frequency as against a 1.5 MIPS throughput via a 12 MHz base clock frequency (as of MA 31750 – [5]) are also added advantages. A typical spacecraft application requires 24 MHz for all its CPU cards. However, the system can be made to work efficiently at 32 MHz as well with a few changes in the power and sequencing requirements. In fact, the use of a higher frequency crystal in the circuit to handle accelerating code complexity significantly reduces the code execution time, contributing effectively to any real-time space application system. However, one can try to achieve a higher frequency of operation to accelerate the quality and performance of spacecraft operations, not to forget that there shall arise a need for stringent power consumption constraints as well. The future work of this project shall probably be the use of much more higher frequencies in the hardware to handle complex code logics with easily configurable power constraint design.

REFERENCES

- [1] LEON3FT SPARC V8 Micro Processor, Functional manual, August 2010.
- [2] Edvin Catovic , Sweden, GRFPU - High Performance IEEE754 Floating Point Unit .
- [3] Hui Zhou, Charles Nicholls, Thomas Kunz and Howard Schwartz, Carleton University, Systems & Computer Engineering Schwartz, Carleton University, Systems and Computer Engineering, Frequency Accuracy & Stability Dependencies of Crystal Oscillators (Technical Report SCE-08-12, November 2008).
- [4] David Guzman, Improving the LEON Spacecraft Computer Processor for Real-Time Performance Analysis (Journal of Spacecrafts and Rockets, Vol 48, No. 4, July-August 2011).
- [5] MA 31750 High Performance MIL-STD-1750 Microprocessor – Dynex Semiconductors (DS3748-8.2 Feb 2006 (LN24435).