

Atmel AT697 validation report

ESA Contract 18533/04/NL/JD, call-off order 2

GR-AT697-002
Version 1.2
June 2005

Table of contents

1	INTRODUCTION	3
1.1	Scope	3
1.2	Background	3
1.3	Summary of results	3
1.4	Reference documents	3
1.5	Acronyms and abbreviations.....	4
2	VALIDATION OVERVIEW	5
2.1	Objective	5
2.2	Validation environment.....	5
2.3	Number of specimens.....	6
3	VALIDATION RESULTS	7
3.1	General	7
3.2	Functional tests	7
3.3	Performance	8
3.4	Power consumption.....	9
3.5	Hardware interfaces	9
3.6	Other issues and anomalies	10
3.6.1	Wrong PC stored during FPU exception trap	10
3.6.2	Single-stepping over SWAP and LDSTUB instruction locks AHB bus	10
3.6.3	Divide overflow will not clear zero flag	10
3.6.4	Register file fault-injection incorrectly implemented	10
4	SUMMARY AND RECOMMENDATIONS.....	12
4.1	General	12
4.2	Recommendations for improvements	12

1 INTRODUCTION

1.1 Scope

This document describes the validation results of the LEON2-FT processor manufactured by ATMEL (F). The validation has been performed on the first available prototypes samples of the ATMEL AT697 device.

1.2 Background

The LEON2-FT processor has been manufactured by ATMEL (F) as a product named AT697. The first samples of the AT697 prototypes have been made available for validation. The objective is to perform an independent validation of the AT697 by manufacturing a dedicated board to host the device and to execute appropriate test programs in order to validate that first ATMEL implementation and to consolidate the design for the future flight implementation of the AT697 by ATMEL.

To support the independent validation of the AT697 device, a development board in Compact PCI format has been developed, manufactured and tested. Both software and hardware validation of the AT697 processor has been performed using this board.

1.3 Summary of results

The software validation showed that the AT697 is fully functional and executed all validation tests correctly. In addition to the three previously known deficiencies, four new ones were found. These are associated with FPU exception handling, single-stepping in debug mode, condition code generation in the divider, and fault-injection in the regsiter file. They do not affect operation under normal conditions.

The hardware validation showed that all interfaces except the SDRAM interface worked correctly at nominal (100 MHz) frequency. These include the SRAM, PCI and serial interfaces. The SDRAM interface operated correctly up to frequency of 90 MHz. At higher frequencies, correct operation could only be achieved with SDRAMs from certain manufacturers. The reason for the incorrect SDRAM operation at high frequencies has not been determined.

1.4 Reference documents

- RD1 The SPARC Architecture Manual, Version 8, Revision SAV080SI9308, SPARC International Inc.
- RD2 IEEE Standard for Binary Floating-Point Arithmetic, IEEE Std 754-1985,
- RD3 Rad-Hard 32-bit SPARC V8 Processor - AT697 - Errata Sheet, 4409A–AERO–01/05, 2004, Atmel Corporation
- RD4 Rad-Hard 32-bit SPARC V8 Processor - AT697, Rev. 4226B–AERO–01/05, 2002, Atmel Corporation
- RD5 RTEMS On-Line Documentation, <http://www.rtems.com/onlinedocs/releases/rtemsdocs-4.6.1/share/rtems/html>
- RD6 uClinux, <http://www.uclinux.org>
- RD7 eCos Reference Manual, version 2.0, <http://ecos.sourceware.org/docs-2.0/pdf/ecos-2.0-ref-a4.pdf>
- RD8 Paranoia: A floating-point benchmark, Karpinski, R. 1985 BYTE 10

- RD9 IEEE 754 Compliance Checker, <http://www.win.ua.ac.be/~cant/ieeccc754.html>
- RD10 UCBTEST suite, <http://www.netlib.org/fp/ucbtest.tgz>
- RD11 TestFloat, <http://www.jhauser.us/arithmetic/TestFloat.html>
- RD12 GR-CPCI-AT697 Development Board, Board Specification, Rev 1.0, 20 Dec. 2004
- RD13 GR-CPCI-AT697 Development Board, Board Level Testing, Rev 0.1, 14 April 2005
- RD14 GR-AT697-001, "AT697 Validation test plan", issue 2, May 2005
- RD15 "Board Level Test Procedure and Report", revision 1.1, 25 May 2005.

1.5 Acronyms and abbreviations

API	Application Programming Interface
DSU	Debug Support Unit
ECOS	Embedded Configurable Operating System
EDAC	Error Correction And Detection
ESA	European Space Agency
FPU	Floating Point Unit
GCC	GNU Compiler Collection
GDB	GNU Debugger
GNC	Guidance, Navigation and Control
GNU	GNU's Not Unix
IEEE	Institute of Electrical and Electronics Engineers
JTAG	Joint Test Action Group
PCI	Peripheral Component Interconnect
PROM	Programmable Read Only Memory
RTEMS	Real-Time Executive for Multiprocessor Systems
SDRAM	Synchronous Dynamic Random Access Memory
SPARC	Scalable Processor ARChitecture
SRAM	Static Random Access Memory

2 VALIDATION OVERVIEW

2.1 Objective

The main objective of the validation of the AT697 device is to insure that the device operates correctly according to the SPARC V8 standard, and that the various interfaces operate as intended. Other issues such as power consumption and performance has also been measured, but were not the main focus of this activity.

The validation tests are divided into four categories:

- validation of instruction execution
- performance measurements
- power measurement
- validation of external interfaces

The test categories are described in further detail in the AT697 validation plan document (GR-AT697-001).

2.2 Validation environment

The validation has been performed on the GR-CPCI-AT697 board (figure 1) developed specifically for this activity. The board has been designed for LEON2 software development, and incorporates all the necessary features and interfaces. The aim has been to provide a platform which enables the validation of correct functioning of AT697 device, by exercising its features and interfaces in its different configurations.

The features of the GR-CPCI-AT697 development board are as follows:

- 3U format Compact PCI card
- ATMEL AT697 device in MCGA349 package (socketted)
- 1.8V and 3.3V power regulators
- On Board memory
 - PROM 4 Mbyte FLASH (organized x8 bit)
 - SRAM 1 Mword SRAM (organized 40 bit wide supporting EDAC)
 - SODIMM socket for 64-bit SDRAM (organized 40 bit wide supporting EDAC)
- Memory expansion
- On Board oscillators
- 16-bit I/O port
- Debug Support Unit serial interface (RS232)
- 32-bit PCI interface, including arbiter (configurable as PCI System Controller or Peripheral)
- LAN91C111 10/100Mbit/s Ethernet interface
- JTAG

The GR-CPCI-AT697 development board has been used in stand-alone mode for the instruction execution validation and in a Compact PCI rack for the interface validation.

All diagnostic communication with the board has been made via the Debug Support Unit interface (RS232) from a personal computer.

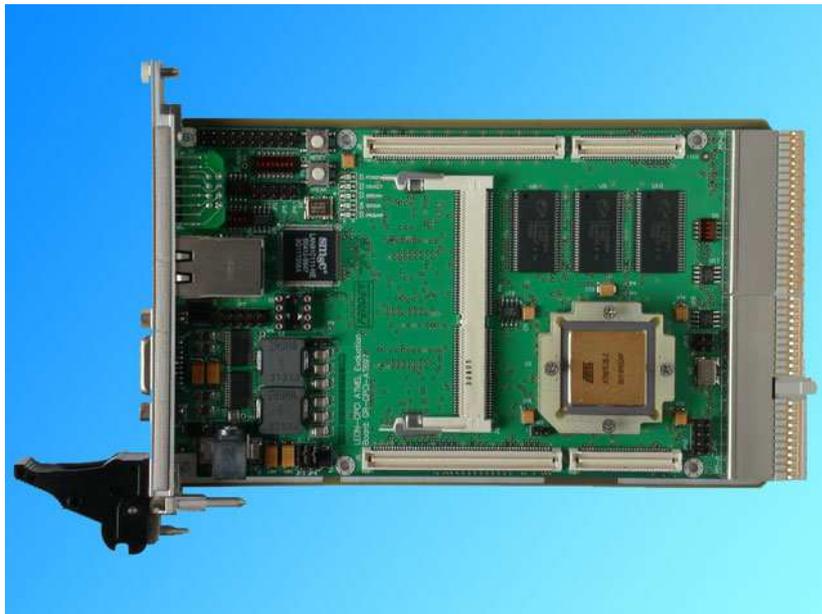


Figure 1: *GR-CPCI-AT697 board*

2.3 Number of specimens

The number of available specimen for the validation is two AT697 prototype devices. Power consumption and hardware interface testing has been performed on both devices, while only one device was used during functional and performance testing.

3 VALIDATION RESULTS

3.1 General

The software validation tests were run on the target hardware as defined in the ‘AT697 Validation plan’ document (RD14). The validation tests consisted of four main categories:

- Functional tests
- Performance measurement
- Power consumption
- Hardware interfaces

While the results of all four tests categories will be reported in this document, the details of the power consumption and hardware interface testing can be found in a separate document: “Board Level Test Procedure and Report”, revision 1.1, 25 May 2005 (RD15).

3.2 Functional tests

The functional tests are divided into five categories:

- SPARC International SPARC V8 validation test suite
- IEEE-Std-754 validation
- RTEMS test suite
- eCos basic test suite
- uClinux operating system

The first four were run through an automatic test script that loaded each test on the target board and verified its correct execution. The last category (uClinux) was executed manually by programming the boot-prom of the target board with the uClinux kernel, resetting the board and perform the uClinux test manually via the console.

All five test categories executed correctly at 100 MHz, using the on-board SRAM with one waitstate. To avoid the faulty divider in AT697, all applications were compiled without SPARC V8 multiply and divide instructions. Running at 80 MHz, or using a selected SDRAM module at 100 MHz, the functional test also passed when executed from SDRAM.

Below is the text log from the main test script:

```
$ ./runValidation.sh
#-----#
| Starting Leon2ft validation |
#-----#
```

Logfiles can be found in /home/jiri/src/validation/logs

```
Initialising memory
done
```

```
Starting validation
SPARCV8 (approx. 2 minutes)
```

```

    SPARCV8: Test Passed!
IEEE-Std-754 (approx. 23 minutes)
-> Running paranoia
    paranoia: Test Passed!
-> Running testfloat
    testfloat: Test Passed!
-> Running gnc_noirq
    gnc_noirq: Test Passed!
-> Running gnc_irq
    gnc_irq: Test Passed!
-> Running CompCheck
    CompCheck: Test Passed!
-> Running ucb
    ucb: Test Passed!
RTEMS (approx. 5.3 minutes)
    RTEMS: Test Passed!
eCos (approx. 1 minute)
    eCos: Test Passed!

```

3.3 Performance

To measure the computational performance of the AT697, a set of standard benchmarks were executed. As a comparison, the same benchmarks were executed on TSC695 (simulator) and on LEON2-FT-UMC. The UMC version of LEON2FT is similar to the AT697, but has smaller cache (2 * 8 Kbyte) and one cycle lower ICC branch delay. The table below summarizes the benchmark results. The figures in parenthesis are relative performance compared to TSC695 at 20 MHz. Note that the SPARC V8 multiply and divide instructions were enabled for LEON during these tests, as the faulty divider did not affect the results.

Benchmark	TSC695 (TSIM)	LEON2FT-UMC	AT697
	20 MHz, 0 ws	100 MHz, 0 ws	100 MHz, 0 ws
Dhrystone (MIPS)	14.6	88.8 (6.1x)	83.9 (5.7x)
Stanford (ms)	547	107 (5.1x)	121 (4.5x)
GNC (ms)	9648	2293 (4.2x)	1928 (5.0x)
Linpack (KFLOPS)	1328	5818 (4.4x)	5949 (4.5x)

Table 1: *Benchmark results*

The LEON2FT processor has roughly the same CPI (clocks-per-instruction) as the TSC695. This means that an AT697 device runs 5 times faster at 100 MHz than TSC695 at 20 MHz. The performance advantage for LEON2 is larger on integer applications, due to the SPARC V8 hardware multiply and divide instructions and the Harvard dual-bus architecture.

An interesting observation is the performance difference between LEON2-FT-UMC and AT697. On integer applications (Dhrystone and Stanford), the UMC device is roughly 5% faster than AT697. On floating-point intensive applications (GNC and Linpack), AT697 is a few percent faster. This can be explained with the difference in cache and ICC branch delay cycles. On integer code with many branches, the extra ICC branch delay cycle used in the AT697 pipeline results in a small performance penalty regardless of the larger cache. On floating-point applications with relatively few branch instructions, the larger cache size provides a small benefit to the AT697.

3.4 Power consumption

Power consumption has been measured under a number of frequencies and conditions. The full results are available in RD15. Below is a table summarizing the most significant results:

AT697 Operation	25 MHz	90 MHz	100 MHz	120 MHz
Idle (power -down)	0.15 W	0.45 W	0.48 W	0.56 W
GNC, 100% cpu load	0.21 W	0.60 W	0.64W	0.89 W
SRAM test, 100% cpu load	0.22 W	0.64 W	0.69 W	0.95 W

Table 2: *AT697 power consumption (1.8V + 3.3 V)*

The power consumption in the table includes both I/O (3.3V) and core (1.8V) supplies.

It can be noted that the power-down mode reduces the power consumption by approximately 25% compared to the power consumption at full load. At 100 MHz, MIPS/Watt figure is approximately 130. For TSC695, this figure is roughly 10 (15 MIPS, 1.5 Watts)

3.5 Hardware interfaces

The operation of the AT697 interfaces has been tested under various conditions. The full results are available in RD15. The outcome of the tests is summarized below:

- PROM and SRAM interfaces works correctly, including the EDAC function
- I/O port works correctly
- The PCI interface is fully functional in all modes (host/satellite/target), and all transfer types (direct access, DMA, configuration cycles).
- Serial ports and DSU interface works correctly.
- SDRAM works correctly up to 96 MHz

The SDRAM interface worked correctly with all tested SDRAM modules up to a frequency of 96 MHz. At 100 MHz, certain SDRAM modules (Kingston 256MB and PMI 128 MB) would give random errors during burst read operations. Other modules such as Kingston 512 MB and Apacer 128 MB would work correctly also at 100 MHz.

It was observed that SDRAM operation was very sensitive to the quality of the AT697 input clock. It was not possible to achieve correct SDRAM operation at any frequency when tunable frequency generator was used instead of a crystal oscillator. The maximum operating frequency for the SDRAM was also lowered to ~ 80 MHz when a zero-delay SDRAM clock buffer was used (see RD15). It has not been possible to identify the cause of the SDRAM problems since no detailed timing characterization of the AT697 SDRAM interface was available during the tests. Further analysis will be carried out when the timing characterization will become available.

3.6 Other issues and anomalies

In addition to the three previously known design deficiencies of the AT697, four new design anomalies were found during the setup of the validation tests. The anomalies are described in the following paragraphs.

3.6.1 Wrong PC stored during FPU exception trap

When a trap is taken by the processor, the program counter (PC) is stored into %11 of the trap window and the next program counter (nPC) is stored into %12. This operation works correctly for all traps except FPU exception (trap type 0x08). During FPU exception, the nPC is erroneously stored into both %11 and %12. This means that the exception handler can not return and re-execute the trapped FPU instruction. During normal operation, this is not a problem since re-executing the trapped instruction would just cause the instruction to trap again. FPU trap handlers in all examined operating systems (RTEMS, eCos, Linux, WxWorks) do not attempt to re-execute a trapped FPU instruction, but create a system error and aborts the task.

A modification to the LEON2-FT VHDL model will be necessary to correct this issue.

3.6.2 Single-stepping over SWAP and LDSTUB instruction locks AHB bus

During a debug session using the debug support unit (DSU), it is possible to perform single-stepping. If an attempt to single-step a SWAP or LDSTUB instruction is made, the AHB bus will be locked and further debugging will be impossible. The reason for this behaviour is that SWAP and LDSTUB instruction perform a read-modify-write cycle which locks the AHB bus to insure atomicity. When such an instruction is single-stepped, the lock signal will be kept active even after the processor enters debug mode, thereby preventing further bus arbitration. Since the communications with the DSU is done over the AHB bus, further debugging is impossible and the device is in principle dead-locked. This state can only be exited by de-asserting the DSUEN signal (resuming execution), or asserting the RESET signal.

The solution to this problem is to de-assert the AHB lock signal when the processor enters debug mode, requiring a modification of the LEON2-FT model.

3.6.3 Divide overflow will not clear zero flag

The divide instructions SDIVCC and UDIVCC set the integer condition codes (negative, overflow and zero) with respect to the final result. When a divide overflow occurs, a pre-defined non-zero value is returned, the overflow bit is set and the zero bit is cleared. However, under certain overflow conditions, the zero bit is wrongly set even though the result is always non-zero.

To correct this issue, a modification of the LEON2-FT VHDL model is necessary.

3.6.4 Register file fault-injection incorrectly implemented

The LEON2-FT processor has a fault-injection function which allows the insertion of errors into the 7-bit EDAC checksum that protects each word of the register file. When fault-injection is enabled, the EDAC checksums are supposed to be XORed with the value of the TCB field in the

%asr16 register. Due to an incorrect configuration of the VHDL model, the fault injection is instead implemented as follows:

- check bits[6:4] of dual-port ram 1 (corresponding to %rs1 operand) are XORed with TCB[2:0]
- check bits[6:4] of dual-port ram 2 (corresponding to %rs2 operand) are XORed with TCB[5:3]

Fault-injection is still possible as intended, but the injection software must be aware of the difference bit location of the injected errors.

4 SUMMARY AND RECOMMENDATIONS

4.1 General

The validation of the AT697 has been carried out according to the validation plans described in RD14 and RD15. It was found that the device was fully functional, although four new design deficiencies were found. The power consumption was measured to ~ 0.7 Watts @ 100 MHz, and in line with simulations. All hardware interface were found fully functional, with the exception of the SDRAM which only worked up to 96 MHz. The reason for this limit is not known and will be further analysed.

4.2 Recommendations for improvements

To assure optimal operation of the flight version of AT697, the following actions should be taken:

- All identified design deficiencies should be correct in the LEON2-FT VHDL model.
- The SDRAM operation should be analysed further to fully understand, and if possible circumvent, the limit in operational frequency.
- If compatible with the device timing, the LEON2-FT VHDL configuration should be changed to remove the extra ICC branch delay cycle in order to improve performance.