Publication Year	2016
Acceptance in OA	2020-07-09T12:29:27Z
Title	Performance analysis of the GR712RC dual-core LEON3FT SPARC V8 processor in an asymmetric multi-processing environment
Authors	GIUSI, Giovanni, LIU, Scige' John, GALLI, Emanuele, DI GIORGIO, Anna Maria, FARINA, Maria, VERTOLLI, Nello, Di Lellis, Andrea M.
Publisher's version (DOI)	10.1117/12.2232589
Handle	http://hdl.handle.net/20.500.12386/26407
Serie	PROCEEDINGS OF SPIE
Volume	9904

PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Performance analysis of the GR712RC dual-core LEON3FT SPARC V8 processor in an asymmetric multi-processing

Giovanni Giusi
Scige J. Liu
Emanuele Galli
Anna M. Di Giorgio
Maria Farina
Nello Vertolli
Andrea M. Di Lellis

SPIE.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 9/20/2017 Terms of Use: https://spiedigitallibrary.spie.org/ss/TermsOfUse.aspx

Performance analysis of the GR712RC dual-core LEON3FT SPARC V8 processor in an asymmetric multi-processing environment

Giovanni Giusi*[a], Scige J. Liu[a], Emanuele Galli[a], Anna M. Di Giorgio[a], Maria Farina[a], Nello Vertolli [a], Andrea M. Di Lellis[b]
[a] Istituto di Astrofisica e Planetologia Spaziali, Via Fosso del Cavaliere 100, 00133 Rome Italy;
[b]AMDL - Via di Grotte Portella, 6, 00044 Frascati (Rome) Italy

ABSTRACT

In this paper we present the results of a series of performance tests carried out on a prototype board mounting the Cobham Gaisler GR712RC Dual Core LEON3FT processor. The aim was the characterization of the performances of the dual core processor when used for executing a highly demanding lossless compression task, acting on data segments continuously copied from the static memory to the processor RAM. The selection of the compression activity to evaluate the performances was driven by the possibility of a comparison with previously executed tests on the Cobham/Aeroflex Gaisler UT699 LEON3FT SPARC™ V8. The results of the test activity have shown a factor 1.6 of improvement with respect to the previous tests, which can easily be improved by adopting a faster onboard board clock, and provided indications on the best size of the data chunks to be used in the compression activity.

Keywords: Leon3, GR712RC, AMP, dual core, RTEMS

*giovanni.giusi@iaps.inaf.it; +39 06 4993 4755

1. INTRODUCTION

In recent years, data rates and volumes produced by scientific space missions have increased exponentially with consequent new highly demanding requirements on onboard data processing and storage. The high costs in terms of memory and bandwidth increased the need of onboard analysis and compression of the collected data before sending them to the ground.

One possible solution is to have HW systems, e.g. ASICs, dedicated to these onboard activities, but the effort made in the past for producing powerful general purpose radiation tolerant processors like the Cobham/Aeroflex Gaisler UT699 LEON3FT SPARC™ V8[1], provided an indication of the possibility to implement an alternative and more flexible pure software solution. In particular, past feasibility studies[2][3][4] for instrument control units of space instrumentation showed the potentialities of the single core LEON3FT, which resulted to be one of the eligible onboard processors for implementing both instrument control and data acquisition and pre-processing functionalities in some of the future medium size missions proposed within the ESA Cosmic Vision program, e.g. Plato, EChO, Ariel, SPICA.

On the other hand, a detailed comparative analysis [5][6] of the performances of this processor when used to implement lossless compression on the astronomical images obtained with the large focal plane array of the VIS instrument[8] onboard the ESA Euclid mission, have demonstrated the impossibility to use it for this kind of activity, having provided compression times larger than the maximum allowed. Given the need of a software implementation of the compression and of all other data preprocessing activities onboard, the tests results caused the selection of an alternative HW architecture using a non European board hosting a more powerful processor (Maxwell SCS750 single board computer based on PowerPC 750FX™ processor).

Therefore, with the aim of finding a solution based on the use of a European processor to similar needs for scientific payloads in future space missions, the possibility to adopt a multi-core approach has been investigated. More specifically, we decided to carry out a series of performance tests using a prototype board mounting the Cobham Gaisler GR712RC[7] Dual Core LEON3FT processor.

Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave, edited by Howard A. MacEwen, Giovanni G. Fazio, Makenzie Lystrup, Proc. of SPIE Vol. 9904, 99045Q · © 2016 SPIE · CCC code: 0277-786X/16/$18 · doi: 10.1117/12.2232589

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 9/20/2017 Terms of Use: https://spiedigitallibrary.spie.org/ss/TermsOfUse.aspx

The used board is based on a homogeneous multi-core system architecture with a single bus and shared memory. The software environment is based on the use of RTEMS (Real-Time Executive for Multiprocessor Systems) version 4.10 (https://www.rtems.org/) as operating system (OS), used in Asymmetric Multi Programming mode (AMP). The application benchmark is based on the SZIP[9], the same lossless compression algorithms used in the past for characterizing the performances of the LEON3FT single core. The algorithm is based on a University of New Mexico open source implementation of the CCSDS 121.0-B-1[10] lossless compressor.

The idea is not to develop a parallel implementation of the algorithm, but to split the overall compression activity into autonomous sub-tasks, acting on independent data sets covering separate portions of the overall focal plane array. The tasks run asynchronously on the individual cores, configured to have separate memory addressable areas and hardware resources. The tests have included a performance analysis of the concurrent usage of the common bus in case of memory transfers from the static memory to the cache. This because this transfer is the one relevant to the optimization of the algorithm performances and allows us a direct comparison with the previous compression studies.

In section 2 the test environment is described, in section 3 the results of the memory throughput analysis are provided, in section 4 the data compression test execution is analyzed and finally results and considerations are summarized in section 5.

2. ADOPTED MULTIPROCESSING ENVIRONMENT

The tests have been conducted on a customized development board manufactured by AMDL Srl (http://www.amdl.biz/) in cooperation with Gaisler Cobahm/Aeroflex. The board is based on the use of GR712RC, a dual core Leon3FT SPARC V8 processor. GR712RC is a component designed for space applications and couples the rad tolerant technology of the chip with the fault tolerant design of the adopted Leon3 processor. A picture of the prototype board is provided on the left panel of Figure 1, the PCB is Eurocard size (160x100 mm2).

The two LEON3FT cores of GR712RC are inter-connected through the high bandwidth AMBA Advanced High-Speed Bus (AHB). Each core hosts a SPARC Reference Memory Management Unit (SRMMU) and an IEEE-754 compliant double-precision FPU and has its own Instruction and Data cache. The two cores are connected to the high-speed peripherals and to the system external memories through the AHB. Low-bandwidth peripherals, like the UART, are clustered under the Advanced Peripheral Bus (APB). Their access is guaranteed by a AHB to APB bridge.

The board is equipped with a 20Mb (540Kb x 40) Rad Hard Low Power CMOS SRAM, a 3Gb (128Mb x 24) SDRAM based scratch memory and an additional 2Gb (128Mb x 16) SDRAM based scratch memory. The used GR712RC part is a 240-pin, 0.5 mm pitch high-reliability ceramic quad flat package (CQFP). In Figure 1, left panel, a block diagram with the adopted GR712RC configuration is provided. In the diagram only the peripherals relevant to the present work are reported.

The adopted operating system is RTEMS (4.10 version) and the overall software development environment includes i. the RTEMS cross-compiler system RCC (1.2.17 version), which embeds the real time kernel of the operating system and the Leon3FT full support for AMP, ii. the bare cross-compiler system BCC (1.0.43 version), and iii. the GRMON2 debugger0. All the tools are provided and supported by Cobham/Gaisler directly.

RTEMS is a Real-Time Operating System (RTOS) designed for embedded systems. It is a free open source software used in many application domains including space applications. When using RTEMS on a multicore hardware configuration we distinguish two main operating modes: i. Symmetric Multi-Process (SMP) and ii. Asymmetric Multi process (AMP). In the first case there is only a single OS instance responsible for both the correct execution of threads flow and the access to peripherals, whereas in the second case it is possible to have one OS instance per core running independently with the footprint in RAM located in separated areas.

A SMP operating system, such as LINUX, automatically provides multi-core support, i.e. all processes are automatically distributed over the different cores. In the other case (AMP operating mode) a multi-core design requires extra-work to manage the possibility of concurrent access to all shared resources (interrupts, timers, peripherals, memory). In order to use a multi-core processor the software should be split up into items that can run in parallel on the different cores.

In this work we used RTEMS in AMP mode to execute our lossless compression benchmarks. This because in space applications a high level of reliability and testability is needed, with a deterministic behavior of the SW, and this can be

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 9/20/2017 Terms of Use: https://spiedigitallibrary.spie.org/ss/TermsOfUse.aspx

betters accomplished by AMP mode, where any hypothetical misbehavior or overload of the running tasks in the additional core do not affect the effective reactivity of the other one, granting a physical insulation of the running spaces.

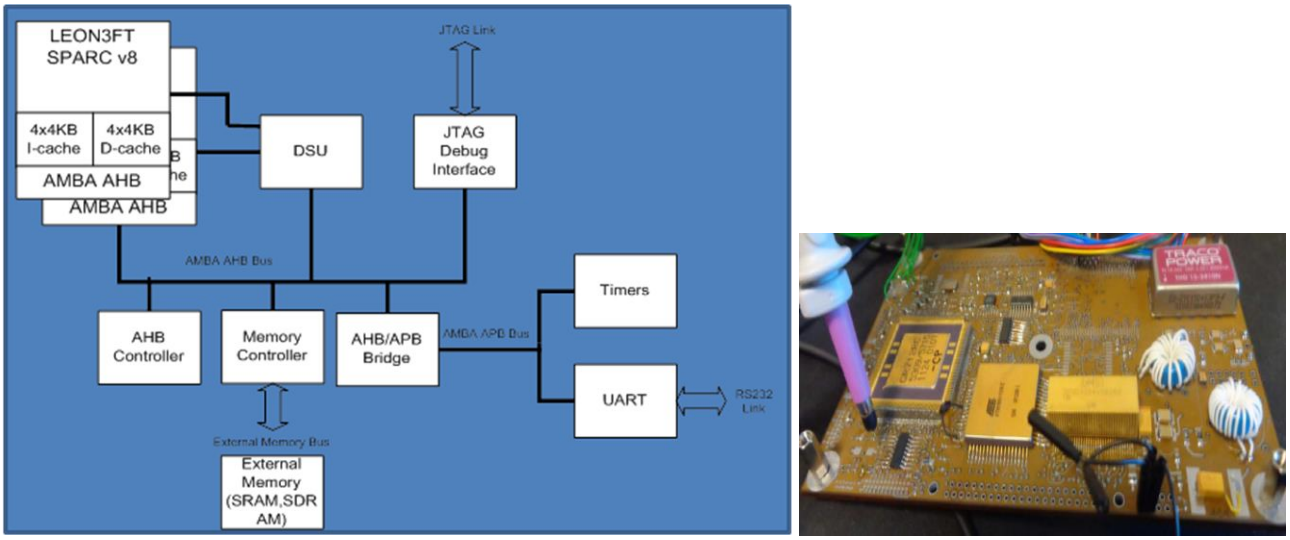

Figure 1 Left: GR712RC block diagram, including only the peripherals relevant to the present work. Right: picture of the used prototype board.

The RTEMS running on the first GR712RC core, the boot processor, has the control over the primary resources and initializes the overall environment, while the RTEMS running on the second core has not access to the main resources but keep full and independent control over its own thread scheduling, being the management of the other resources left to the developers choices.

The presented test scenarios refer to the performance evaluation of a data compression task; the algorithm under test, see section 2.1, is based on the compression of self-consisting contiguous data chunks and does not need any exchange of data between the various computing threads. Therefore, the instances of the compression tasks on the two cores can run independently on different segments of the input images.

2.1 Tested algorithms

To better understand the system behavior in terms of memory access and handling a preliminary set of tests have been done to characterize the system memory transfer performances. The second step has then been focused on the study of the performances of the SZIP algorithm, i.e. the CCSDS 121.0-B-1[10][9] standard lossless data compression algorithm developed at University of New Mexico presently under evaluation for the implementation onboard the Euclid VIS instrument. SZIP is a one dimensional stream-oriented data compressor. It relies on the correlations between linearly adjacent data. The sequential data are subdivided into scanlines, each one being composed of a set of blocks, on their turn composed of a set of samples (i.e. in our case a sample is 1 pixel). SZIP is an implementation of the extended-Rice lossless compression algorithm. It is reported to provide fast and effective compression, specifically for the EOS data generated by the NASA Earth Observatory System (EOS), refer to Pen-Shu Yeh et al.[9].

2.2 Input data

The set of input data used in the test is composed by images produced with the VIS Image Simulator[12], developed at Mullard Space Science Laboratory. The simulator allowed to generate a set of images of the extragalactic sky, as it will be observed by Euclid, including all the defect and noise sources expected for the Euclid VIS focal plane CCD array: continuous pixel sampling of the cosmic ray tracks, different and realistic bias levels for each quadrant, more reliable radiation dose for CTI effects (estimated at the end of mission), improved CTI effect for cosmic ray tracks, cosmetic defects, pixel-to-pixel variation, optical ghosts. The test have been conducted on single quadrant of a simulated CCD image (2048x2066pixels with 16bit/pixels). A portion of the used image is reported in Figure 2.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 9/20/2017 Terms of Use: https://spiedigitallibrary.spie.org/ss/TermsOfUse.aspx

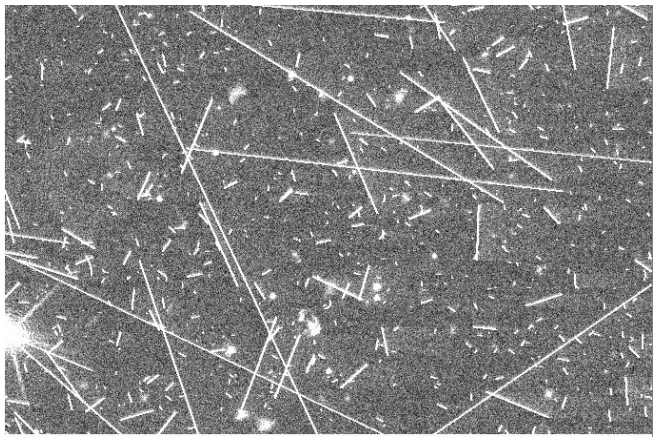

Figure 2 Section of the simulated VIS image used for the tests. The linear tracks are due to the simulated the cosmic ray hits effects.

3. MEMORY THROUGHPUT ANALYSIS

To evaluate and optimize the GR712RC internal resources utilization, a series of memory access performance tests have been carried out, based on the cyclic execution of array copy operations with variable array size and cache configuration. The aim was to analyze how the cache affects the data transport in the path memory-processor-memory and to evaluate the corresponding impact on the size of the array, finding the value that ensures the best use of the available memory bandwidth. This value will then be used as a reference size for the selection of the scanline dimension in the SZIP lossless compression algorithm and will allow to minimize the compression time.

The behavior of the memory copy activity was analyzed in both the single and dual core configurations. Since the effective interaction of each processor cache with the AHB depends on the code used to implement this functionality, we decided to develop a custom memcopy code using an unrolled loop of data load and store, able to transfer four 4 Byte words in 13 instructions (on average), using only 4 registers. This implementation has then to be coupled with the effective bandwidth made available onboard to the memory read/write operations by the processor memory controller.

The GR712RC Leon3FT cores cache controllers implement a 4-way set associativity. The set size is 4KiB divided into cache lines with 16 Bytes data. As a consequence, with this configuration each cache miss forces a read of four 4 Byte words from the static memory. To estimate the memory bandwidth effectively available we need to consider the number of bus cycles necessary for executing each 16 bytes read/write operation. The possibility to transfer 4 consecutive 4 Bytes words is accomplished is the so called memory access burst mode. The timing of the read operation in a burst cycle allows the transfer of 4 data words adding only an extra lead-out cycle after the last transfer. The write cycle is implemented by using the basic "lead-in, data, lead-out" cycle also in the burst mode. Therefore, considering that in the development board used for the tests a 50 MHz oscillator is used, when arrays contiguously mapped in the static memory have to be copied, the optimal bandwidth available for the memcopy operation is:

$$MemCopy|_{burst} = \frac{50MHz}{(4_{data}+1_{lead\_out})_{read} + (4_{lead-in}+4_{data}+4_{lead\_out})_{write}} * 16 \Rightarrow 47MByte/s_{memcopy}$$

On the other hand, in case of random access to the memory, induced either by cache miss or by a segmented mapping in memory of the data to be copied, the possibility to transfer 4 Bytes consecutive words is not applicable any more, and we have to consider each word transfer as a separate memory access. In this case the Leon3FT timing is such that the read bus cycle is of the type "data, data, lead-out" while the write cycle is the basic "lead-in, data, lead-out" cycle, where the unit size of each transferred data is 4 Bytes. The effective bandwidth related to this worst case becomes:

$$MemCopy|_{random} = \frac{Clock}{\sum Cycles} * UnitSize = \frac{50MHz}{3_{read}+3_{write}} * 4 \Rightarrow 33MByte/s_{memcopy}$$

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 9/20/2017 Terms of Use: https://spiedigitallibrary.spie.org/ss/TermsOfUse.aspx

In a real system based on the use of GR712RC the actual memcopy throughput will never be greater than the optimal bandwidth but can assume values well below the worst available bandwidth depending on the implementation of the fucntion.

3.1 Cache use effects on Memory Copy performance

To measure the efficiency of the developed memcopy function when applied to the GR712RC processor, the throughput profile at software level has been studied using two tasks in AMP configuration, running separately in each processor core (see section 2). The activity of the two tasks was to perform continuous array copy operations on different sections of the static memory. The array size has been initially set to 64 Bytes, a value below the processor D-cache single page size (4096 Bytes). The throughputs have been measured by counting the number of Bytes transferred to the two different sections of the static memory in a fixed time interval.

The first step of the test has been to measure the best obtainable throughput with one only core operating at a time. The result of these tests is reported in Figure 3, where the measured throughput is reported as a function of the array size. In this case, the average measured value has been between 24MBytes/s and 34 MBytes/s for array sizes smaller than the D-cache single page size. Increasing the array size there is a performance degradation, as expected, due to the increasing of cash miss effects on the overall data transfer budget. The best throughput is obtained for array sizes contained within the D-cache page size. When the array size exceeds the overall D-cache size of 16KiB, the system operates always in a cache miss regime and the throughput becomes the same as the one obtained by disabling the D-cache.

In the second step the two memcopy tasks were running in parallel on the two processor cores, accessing concurrently to the memory through the shared AHB. In this case the average measured throughput was between 33MBytes/s and 42MBytes/s. The overall behavior of the dual core test is similar to the one obtained for the test with the single core, but in the dual core case the overall cumulated throughput can reach values not too far from the optimal usage of the available bandwidth.

The array size which optimizes the efficiency of the memcopy operation is therefore between 2048 and 4096 Bytes.

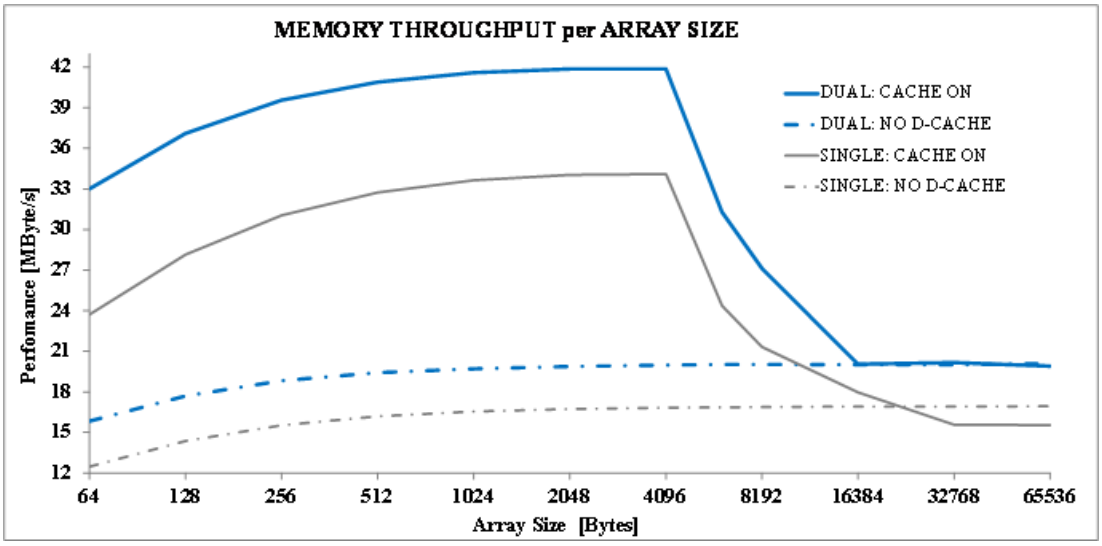

Figure 3: Memory throughput per array size considering D-Cache usage, and in single and dual core mode.

4. DATA COMPRESSION TESTS

To compare the compression performances between single and dual core configurations, we measured the compression task execution time in different operational scenarios.

With reference to Figure 4, the first scenario is the one in which the full computational power of both cores is dedicated to the compression activity. This scenario has allowed us to make a direct comparison to the single core Leon3FT processor compression performance analysis made in the past[5][2][6]. In the second scenario we assigned to the compression task a maximum overall CPU occupation of 50%, with one core fully dedicated to the compression activity

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 9/20/2017 Terms of Use: https://spiedigitallibrary.spie.org/ss/TermsOfUse.aspx

and the second one executing 10 different concurrent tasks. These tasks have been assigned all equal priority, higher than the one assigned to the compression activity, resulting in an equal CPU time occupation. In the third scenario we assigned to the compression task a 50% CPU occupation as before, but based on a compression activity running in parallel on the two cores. In each one of them, the task shares the CPU with 5 co-runner tasks.

The data flow model common to the three scenarios is shown in Figure 5. In all cases the compression task input and output data sets are stored in buffers resident in the static memory; TA-0(1) are the tasks, scheduled by the two cores, that implement the compression engine.

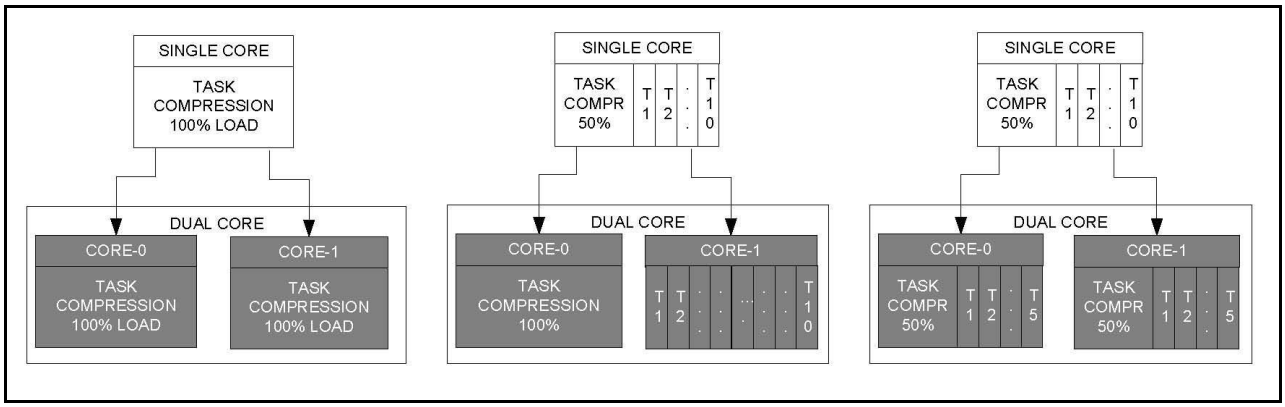

Figure 4: The three test configuration used: (left) scenario-1 (center) scenario-2 (right) scenario-3. See text for a detailed description of the three scenarios

The RTEMS Multiprocessor Communication Interface handler is responsible of the message exchange between the cores. Since the two cores work in an asynchronous an independent way, no messages are exchanged during the execution process except for the execution completion message. Each test terminates when the complete input data set (analyzed in chunks of 512KBytes) has been compressed. The compression execution time is measured and a "speedup" ratio is calculated comparing the compression time in single core configuration with the correspondent time measured in dual core configuration.

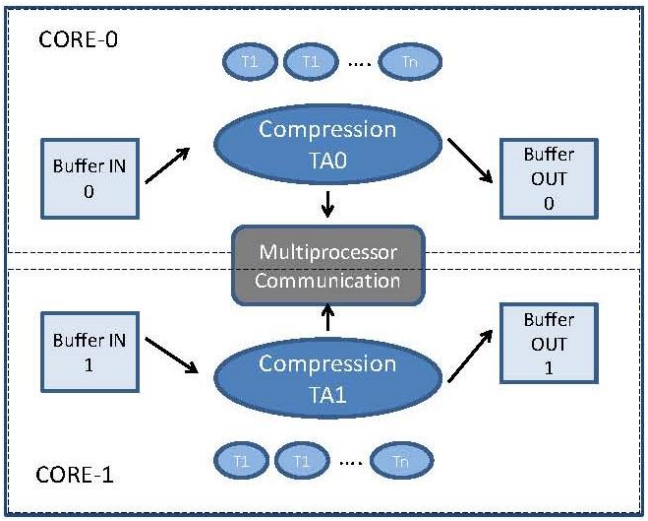

Figure 5: (left panel) data flow model (right panel) benchmark memory map

The first test scenario has been used to evaluate the SZIP compression algorithm performance as a function of the input dataset (scanline [9]) sizes. In Figure 6 the results of the tests executed by varying the scanline size from 32 to 4096 pixels are reported. It is possible to see that the measured compression efficiency decreases with shorter scanline sizes: this is

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 9/20/2017 Terms of Use: https://spiedigitallibrary.spie.org/ss/TermsOfUse.aspx

mainly due to the increasing overhead for the production of an increasing number of output packets. The obtained curve has a peak for a scanline size of 2048 pixels. For larger sizes the performance decreases.

If we couple these results with the memory throughput results obtained in the previous section, we see that the best optimization of the compression activity is obtained for a scanline size of 2048 pixels (4096 bytes). This size has been used in the SZIP performance assessment tests executed in the three scenarios described above. In Table 1 we report the compression efficiency, expressed in compressed Kpixels/s, obtained in the three cases with different cache conditions.

The single core result for Scenario 1 is comparable with the analogous performances evaluated in a previous work[13] on the GR-CPCI-UT699 LEON3-FT.

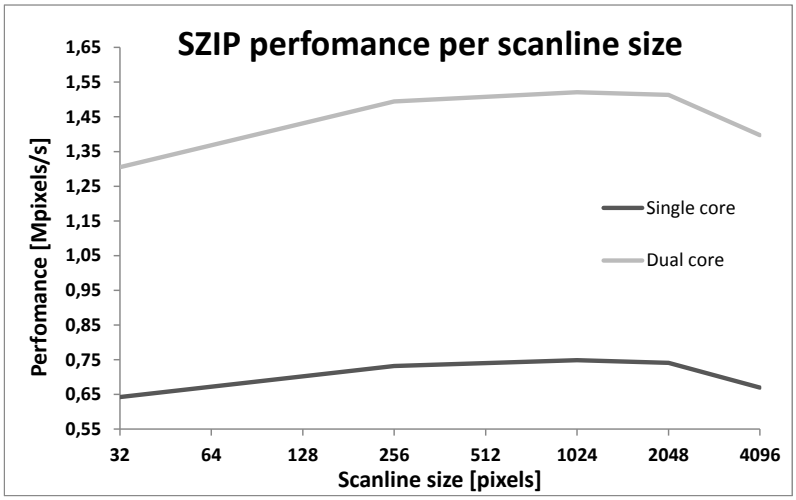

Figure 6: SZIP performance measured in number of compressed Mpixels per second as a function of the scanline size in single and dual core mode

In past studies[5][6] the SZIP compression algorithm was evaluated on the GR-CPCI-UT699 LEON3-FT single core development board. In that case, with Clock@66MHz and SRAM CY7C1069AV33@10ns, a maximum performance of 0.9 Megapixels/s was obtained. The single core results of the present tests are therefore suffering for the slower onboard clock (the used prototype board was based on Clock@50MHz and the SRAM ATMEL AT68166H@18-20ns), and provide approximately 0.8Mpixels/s. On the other hand, the dual core results obtained here and reported in Table 1show an improvement of a factor 1.6 (1.5 Mpixels/s).

Table 1. SZIP performance (in compressed Kpixels/s), for single and dual core configuration, with D-Cache On and Off

Core configuration	Test ID	D-Cache On	D-Cache Off
SINGLE CORE	Scenario 1	777	525
	Scenario 2,3	391	262
DUAL CORE	Scenario 1	1513	884
	Scenario 2	769	490
	Scenario 3	765	463

In multiprocessing systems the speedup of an architecture with multiple core depends both on the topology of the system and on the level of parallelization of the code. Such improvement can be modeled with the Amdahl's law[11]:

$$S = \frac{1}{(1-F) + \frac{F}{N}} \quad (1)$$

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 9/20/2017 Terms of Use: https://spiedigitallibrary.spie.org/ss/TermsOfUse.aspx

Where S (Speedup) is the relative speedup in execution time, N (Multiplicity) is the number of processors or resources driving the optimization, F (Optimization_Fraction) is the portion of the system that can be optimised, then (1-F) is the portion that cannot. The speedup upper limit is bound by (1-F) regardless of the multiplicity of resources involved.

Since the aim of our tests was to assess the performance of an algorithm to be eventually used for space applications and given that the space qualified RTEMS OS used for our tests doesn't support the Symmetric Multiprocessing, we didn't implement any kind of parallelization of the compression algorithm. Therefore only the effect of N on the overall speedup has been taken into account. The Speedup S can be obtained from the ratio between the performances mesured with single and dual core configurations.

Table 2. Obtained optimization fraction, a measure of the efficiency of the use of two cores wrt to one.

	Optimization Fraction F	
	D-Cache On	D-Cache OFF
Scenario-1	0,97	0,81
Scenario-2	0,98	0,93
Scenario-3	0,97	0,87

In Table 2, the values of F calculated using the speedup factors in the three test scenarios are reported for the two different cache enabling states. When D-cache is enabled F, i.e. the efficiency of the cores doubling, is high because the cache is so large to contain more scanlines of the image and the computation is local to the processor cache: no bottleneck is produced on the bus. The D-cache disabled figures give us a measure of the worst case in which the advantages of using two cores is partially lost due to both the need of continuously access the static memory and the balance between the OS latencies and the co-runner tasks traffic.

5. CONCLUSION

In this study we reported the results of a set of tests conducted on a customized development board based on the use of GR712RC, a dual core Leon3FT SPARC V8 processor.

In the first group of the tests the behavior of the memory copy activity was analyzed in both the single and dual core configurations. For this purpose a custom memcopy code has been designed to optimize the interaction with the processor cache. The tests provided us an evaluation of the memory throughput as a function of the size of the array to be copied. In case of the dual core configuration the average measured throughput was between 33MBytes/s and 42MBytes/s. The overall cumulated throughput can reach values not too far from the optimal usage of the available bandwidth, which has been estimated to be around 47MBytes/s. The best throughput is obtained for array sizes contained within the D-cache page size. Increasing the array size there is a performance degradation, as expected, due to the increasing of cash miss effects on the overall data transfer budget. When the array size exceeds the overall D-cache size of 16KiB, the system operates always in a cache miss regime and the throughput becomes the same as the one obtained by disabling the D-cache.

These preliminary tests showed that the array size which optimizes the efficiency of the memcopy operation is between 2048 and 4096 Bytes.

In the successive steps, the comparison of CCSDS121 lossless compression algorithm execution, on single and dual core configurations, have shown that processing enhancement may reach a speedup of 1.94 wrt the use of a single core. This speedup is dominated more by the preparation to the compression phase than by the algorithm engine itself: the performance with small scanline sizes are in fact dominated by the code latency.

Finally it has been explored how using the dual core with different tasks allocations, can affect the overall activity efficiency. Comparing the results of the three test scenarios used for this purpose, it came up evident that the best option is to dedicate a core fully to the compression task and the other to all other concurrent tasks (5 co-runner tasks have een used to simulate concurrent activities, e.g. instrument control activities running onboard in parallel to the acquired data compression.

We have now some a hardware characterization to be used for the code optimization and for the re-modulation of the compression engine, with the aim to localize the variable usage and modify it to best fit the hardware caches effects. A profiling of the execution flow under intensive I/O solicitation from SpaceWire data interfaces and MIL-STD-1553

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 9/20/2017 Terms of Use: https://spiedigitallibrary.spie.org/ss/TermsOfUse.aspx

control interface is in addition needed to better characterize the dual core processor behavior in case of more realistic environments.

REFERENCES

[1] Aeroflex Colorado Springs, Inc., "UT699 32-bit Fault-Tolerant SPARC^{TM} V8 LEON3FT Processor", Data Sheet, July (2013)

[2] M. Focardi, M. Farina, M. Pancrazzi, A.M. Di Giorgio, T. L. Lim, R. Ottensamer, S. Pezzuto, E. Pace, G. Micela, "The instrument control unit of the EChO space mission: electrical architecture and processing requirements," SPIE Proc. Vol. 9143, 91434H, (2014)

[3] M. Focardi, M. Pancrazzi, A. M. Di Giorgio, S. Pezzuto, G. Micela, E. Pace, "The Exoplanet Characterization Observatory (EChO) Payload Electronics," SPIE Proc. Vol. 8442, 84422T, (2012)

[4] Anna M. Di Giorgio, Francesco Nuzzolo, David Biondi, Massimiliano De Luca, Pasquale Cerulli Irelli, Renato Orfei, Luigi Spinoglio, "The Digital Processing Unit of the SPICA SAFARI instrument: an FPGA based architecture using the LEON2-FT," Proc. SPIE, Vol. 7731, 77314B, (2010)

[5] Di Giorgio, Anna Maria, Liu, Scige J., Giusi, Giovanni, Palamara, Giovanni, "Euclid Visible Imager on-board lossless data compression: performance assessment trade-off activities," Proc. of ESA On-Board Payload Data Compression Workshop OBPDC Barcelona-Spain, (2012)

[6] Anna Maria Di Giorgio, Paolo Bastia, Scige J. Liu, Giovanni Giusi, Roberto Scaramella, Mark Cropper, Richard Cole, Ady James, Jérôme Amiaux, Yannick Mellier, "The Command and Data processing Unit of the Euclid Visible Imager: impact of the data compression needs on the unit design," Proceedings of the SPIE, Volume 8442, article id. 844233, 9 pp. (2012)

[7] Cobham Gaisler AB, "GR712RC Dual-Core LEON3FT SPARC V8 Processor", Data Sheet, Jan (2016)

[8] Mark Cropper, S. Pottinger, S.-M. Niemi, J. Denniston, R. Cole, M. Szafraniec, Y. Mellier, M. Berthe, J. Martignac, C. Cara, A.M. di Giorgio, A. Sciortino, S. Paltani, L. Genolet, J.-J. Fourmond, M. Charra, P. Guttridge, B. Winter, J. Endicott, A. Holland, J. Gow, N. Murray, D. Hall, J. Amiaux, R. Laureijs, G. Racca, J.-C. Salvignol, A. Short, J. Lorenzo Alvarez, T. Kitching, H. Hoekstra, R. Massey, and the Euclid collaboration "VIS: the visible imager for Euclid," Proceedings of the SPIE, Volume 9143, article id. 84420J, 13 pp. (2014)

[9] Pen-Shu Yeh et al., "Implementation of CCSDS Lossless Data Compression in HDF," Earth Science Technology Conference–2002, Pasadena, California, 11–13 June (2002)

[10] CCSDS 121.0-B-2, Recommended standard, Blue Book, May (2012.)

[11] Amdahl, Gene M., "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities," AFIPS Conference Proceeding, 18-20 April (1967), Atlantic City, N.J.

[12] Niemi, S.M., "VIS Image Simulator", <https://www.mssl.ucl.ac.uk/euclid_vis/>.

[13] Giusi, Giovanni, Liu, Scige J., Di Giorgio, Anna M., Galli, Emanuele, Pezzuto, Stefano, Farina, Maria, Spinoglio, Luigi, "SAFARI digital processing unit: performance analysis of the SpaceWire links in case of a LEON3-FT based CPU," Proc. SPIE 9143, Space Telescopes and Instrumentation 2014: Optical, Infrared, and Millimeter Wave, 91434P August 2, (2014)

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 9/20/2017 Terms of Use: https://spiedigitallibrary.spie.org/ss/TermsOfUse.aspx