POSITION PAPER

LEON and NOEL-V SoC Architectures

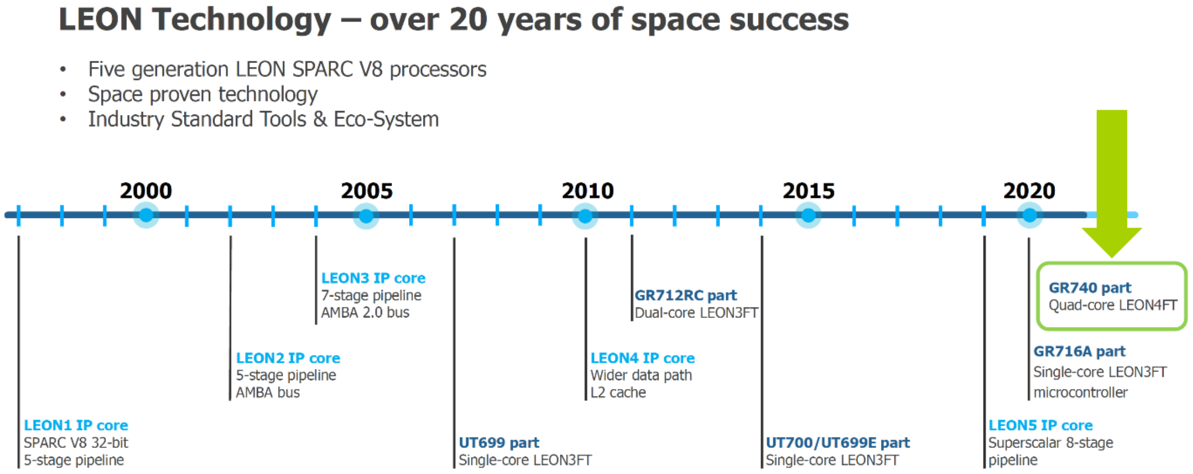
Summary/Abstract

This document describes CAES' past and ongoing component development and explains the rationale for some architectural design choices for future roadmap products. Included herein are trends in the Space industry that are driving key new features example application use-cases and tradeoffs from a software perspective of a legacy LEON design vs. a new RISC-V architecture. Various comparison tables are included for reference.

Background

As the largest provider of radiation hardened technology to the United States aerospace and defense industry, CAES has a long heritage of providing space-grade processors based on the LEON SPARC V8 architecture, along with it being dominant in Europe and used worldwide. A common characteristic of all the CAES LEON implementations is that they are radiation-hardened fault-tolerant (FT) microprocessors that allow software to continue operation uninterrupted in the presence of correctable errors. Detected uncorrectable errors are handled by stopping operation and a bad state is never allowed to propagate outside of the System-on-Chip (SoC).

Figure 1 below shows the timeline during which the components have been introduced. All of the components are based on 32-bit LEON processors implementing the SPARC V8 instruction set architecture (ISA). The GR740 quad-core LEON4FT is the latest SoC offering currently in production.



LEON Technology – over 20 years of space success

- Five generation LEON SPARC V8 processors
- Space proven technology
- Industry Standard Tools & Eco-System

Figure 1 LEON development timeline

The SoC architecture has evolved between the UT699 and GR740 with more processor cores being added to the same component and an increasing level of integration with more communication interfaces being added at the same time. Figure 2 shows the single-core UT699 (similar to the later products UT699E and UT700), Figure 3 shows the dual-core GR712RC, and Figure 4 shows the quad-core GR740.

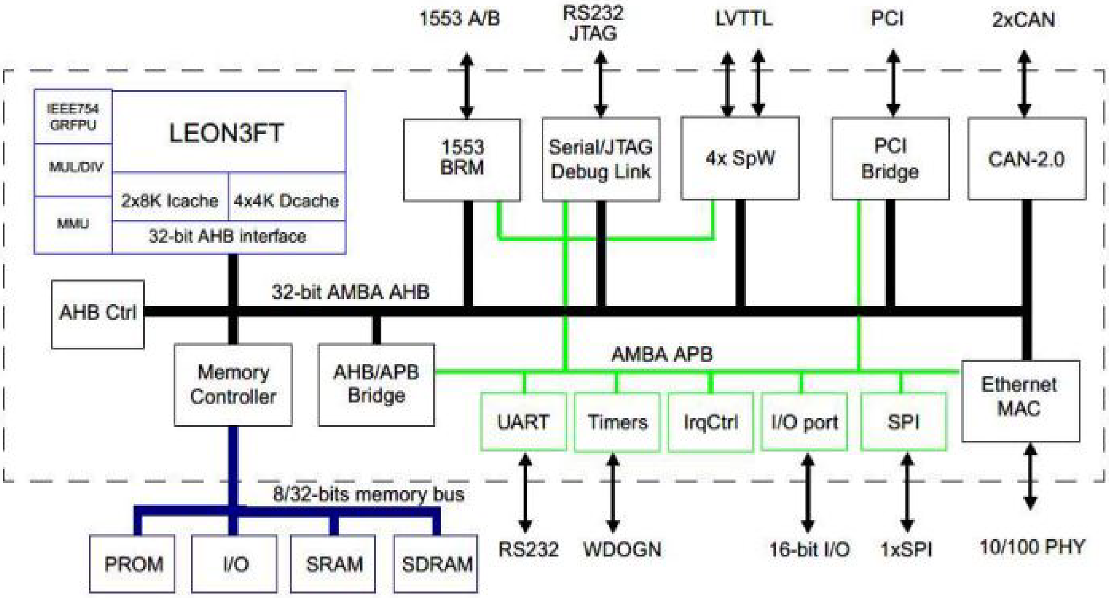


Figure 2 UT699 block diagram

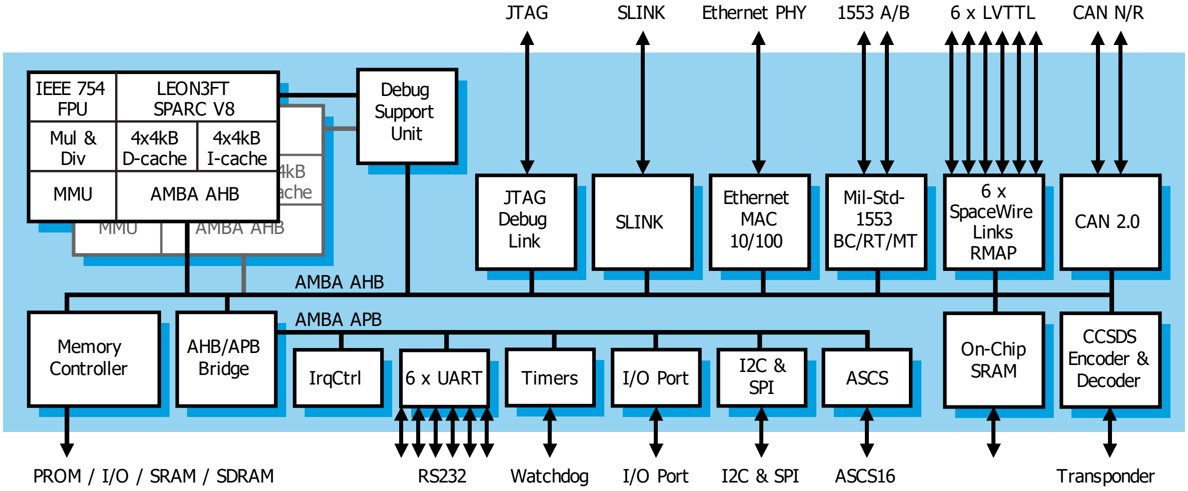


Figure 3 GR712RC block diagram

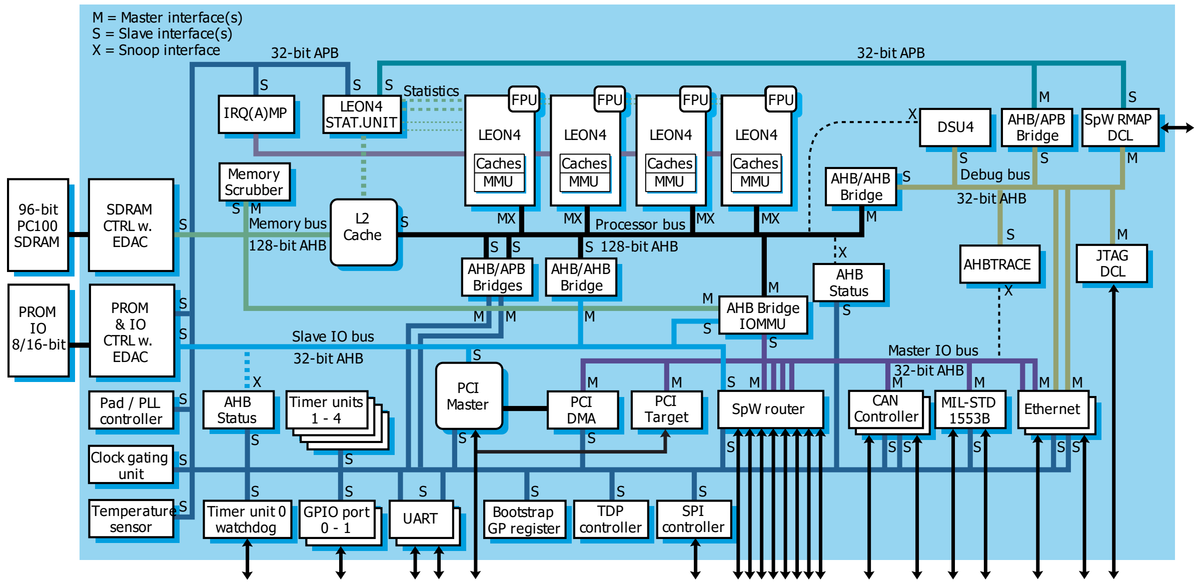


Figure 4 GR740 block diagram

The move from the single-core UT699 to the dual-core GR712RC introduced one more processor core connected to the same bus, and memory controller, while expanding the set of on-chip communication interfaces.

The GR712RC was the first widely available space-grade dual-core processor and its introduction required engineers in the space industry to tackle problems with multi-core timing interference. It has been shown, that an architecture like the GR712RC can still allow worst case timing analysis that does not become overly pessimistic by controlling the number of bus accesses that software images running on each of the processors are allowed to make. This allows the user to calculate the maximum amount of interference a software image can suffer. While having an architecture that is simple enough to be analyzable, the GR712RC was designed to be controlled by a SMP operating system. Running software images in asymmetric multiprocessing (ASMP or AMP) is not possible while keeping full separation due to resources such as the interrupt controller register interface being shared between all processors.

The GR740 introduced two additional processor cores, creating a quad-core system, a Level-2 (L2) cache, and additional communication interfaces, also with peripherals capable of direct memory access (DMA) connected to the system through an IOMMU (IO memory management unit). The GR740 was designed to allow both SMP and AMP operation by duplicating shared resources or extending peripheral register interfaces to allow the processor cores to map in only a subset of the registers within a MMU page. The GR740 also included features in the L2 cache to improve separation between software instances, such as functionality to prevent one processor from evicting cache lines allocated by another processor in the L2. These features together with the IOMMU allow use of the GR740 to build time and space partitioned systems.

Today, the LEON line of SoC has a large installed customer base and the development of the LEON line of products continues. The LEON5FT processor IP was introduced in December 2019 and is now being applied in a GR740 follow-on SoC development with the product name GR765; please refer to Figure 5 for a block diagram. The GR765 improves performance over previous generations and improves support for time and space partitioning. Due to shared resources (such as the bus connecting the processor to the L2 cache and memory controller) there is timing interference between processor cores in the GR712RC and GR740. In addition to this, the LEON processor cores implement the SPARC V8 ISA that does not support a hypervisor mode, requiring hypervisor software that run on the component to apply paravirtualization. This has a significant cost for complex hypervisor guests that may need to be modified to run in only one processor privilege mode, and practically prevents running complex hypervisor guests such as the Linux operating system. The GR765 architecture addresses these items, reducing the needed software verification effort for multicore systems.

The GR765, GR740, GR712RC and UT700/UT699 remain software compatible. As an example, an operating system image built for the LEON3 will be possible to boot on all the architectures. There are differences in terms of number of peripherals and version of the peripheral units that requires different software drivers to be loaded depending on the component used, the interfaces used in an application, and specific board that the component is mounted on. Since CAES provides in-production components of both LEON3FT and LEON4FT systems, and through the software compatibility, support for these components as part of the LEON software ecosystem will remain for the foreseeable future.

The LEON software ecosystem is complete for traditional users in the space industry. Since the space industry is the predominate industry that make use of the SPARC architecture, there are two main challenges; (1) reuse of software packages from other industries may be difficult (this applies to for example specialized optimized math libraries and operating systems such as Android), and (2) due to other industries moving away from SPARC, it is difficult to recruit new engineers that have experience with the architecture. With software development increasingly moving to higher layers (a larger number of users select to implement their application on top of an operating system instead of using bare-metal applications), CAES tries to reduce the effect of (2) by providing a larger set of board support packages (BSPs) and software drivers.

CAES and its Gaisler Products team intend to support the LEON technology for decades to come. At the same time, we are introducing additional architectures. On the microcontroller side, CAES has introduced a space-grade ARM Cortex-M0-based microcontroller, the UT32M0R500. CAES Gaisler Products introduced another LEON based microcontroller in parallel, the GR716A which is a single core LEON3FT, the LEON-REX instruction set extension for improved code density, running at 50MHz and providing 98 DMIPS. An enhanced version, the GR716B, is in development and will run up to twice as fast and have an extended set of digital and analog functions. For more advanced processing, CAES offers a RISC-V processor core named the NOEL-V and has on-going funded projects with the objective to introduce standard products based on this core. See more below on our next-next generation architecture: NOEL-V RISC-V GR7xV.

Trends

The sections below provide an overview of some of the trends we have observed in the industry and how these trends affect our future products.

COTS vs traditional space-grade electronics, and qualification

The below table provides a high level comparison of upscreened COTS (Commercial-Off-the-Shelf) components versus SoCs designed for space, with fault-tolerance (FT) built in.

Parameter	upscreened COTS processor	LEON SoC	NOEL-V RISC-V SoC
component price	lowest	moderate	moderate
screening cost	unknown, moderate	none	none
fault-tolerance	no, not built in	yes, built in	yes, built in
memory protection	soft, not up to space standards	built in	built in
recovery time on errors	long, may require software restart	none (errors handled in background)	none (errors handled in background)
detectable errors	no, not built in	yes, built in	yes, built in
flight heritage	some	extensive	expected in 2022 (for IP only)
rad effects	soft, unknown	good, known	expected to be similar to LEON SoC
3rd party software	extensive	some, well known	extensive
simulators	readily available	provided by CAES	to be provided by CAES & 3rd parties

Table 1 Upscreened COTS vs FT SoCs

Notes:

- Breakeven point for upscreening COTS vs. using FT SoCs depends heavily on quantity of components as well as mission requirements & available software and system engineering resources
- Upscreening can require special access to the manufacturer, such as for a single lot date for radiation testing and on a flight production population, and my not be available.

- Using COTS components typically increases the cost of software & system engineering to handle errors

Screening of COTS components has traditionally been a way to secure access to components for use in space applications in cases where there is no high-reliability counterpart available. This applies to low-complexity ICs and to high-performance microprocessors, where the space-grade processors have not been able to provide the same performance improvements as seen in other industries. With the event of "New Space", selection of COTS components with lower reliability has also been a strategy in cases where reliability can instead be achieved at a system level, either by adding boards, or by adding additional spacecraft.

Selecting COTS to save cost is a strategy that has been applied successfully for applications with large volumes. Using COTS as a cost saver for a single mission can bring a large cost in screening efforts and engineering resources to achieve system-level reliability. In the specific case of a microprocessor, handling of single event effects may lead to an increased effort in software development, verification and validation due to the need of increased error handling and software redundancy.

The strategy from CAES is to keep developing our high-reliability, fault-tolerant and radiation hardened microprocessor implementations that have traditional flight qualification levels (QMLQ & QMLV). A focus is kept on supporting traditional space missions, with additional components following the GR740 to get QML class V quality level.

Communication interfaces

During the past decade there has been increased interest in high-speed serial links (HSSL) using protocols such as Serial RapidIO (SRIO), PCIe and SpaceFibre. Traditionally the trend in the US has been to prefer SRIO and for European entities PCIe. At the same time, both sides of the Atlantic have expressed some interest in PCIe.

To our knowledge, wide adoption of SpaceFibre and Serial RapidIO in space applications has not yet happened. HSSLs are being applied, but instead using simpler protocols. There is also a case for PCIe in that it is a protocol well supported by COTS devices such as microprocessors and FPGAs.

CAES also supports SpaceFibre with a controller IP core. This controller has recently also been extended with a WizardLink mode, allowing for the HSSL protocol to be defined in software. We are currently also assessing inclusion of PCIe support and SRIO, listed in order of priority.

Communication interfaces – data aggregation

A common use case for CAES microprocessor and microcontroller products is for the processors to collect information from one or several data sources and either do computations on the data or deliver it to other parts of the system. In many cases this data is obtained through relatively low speed interfaces such as UART, SPI, and I^{2}C. These interfaces have traditionally been accessed by software through memory-mapped register and streaming data that way leads to a significant amount of effort on the general-purpose processor compared to the amount of data moved. To offload the processors from these tasks, CAES now implements programmable DMA controllers, starting with the GR716 LEON3FT microcontroller and GR765 LEON5FT microprocessor, that can continuously poll the controllers for the previously mentioned communication interfaces and allow a more efficient use of the SoC, reducing the need to use software to collect data and freeing up processor resources to do computations.

Processor Instruction Set Architecture

Space-grade (fault-tolerant, radiation-hardened component capable of withstanding vibration and shock from launches, and the temperature and vacuum in space) processors have in recent years been predominantly based on implementations of the SPARC and PowerPC instruction set architectures (ISA).

An ISA describes the functionality provided by hardware to software, and how software makes use of this functionality. For most end users of a SoC, the primary difference due to an ISA is the software ecosystem available. For component vendors, ISAs matter due to licensing conditions, availability of processor models supporting the ISA, and also from an export control perspective. There have been some indications that ISAs are critical for performance. ISAs introduced today describe Reduced Instruction Set Computers (RISC). The specification of an ISA does matter for how efficient the underlying architecture can be made. However, a RISC is a RISC and ISAs are not magic. Performance and power metrics are more closely tied to microprocessor model implementations, target technology choices, and quality of software ecosystem.

A trend over the past decade has been the introduction of more ARM-based space-grade processors. With ARM being the dominant vendor in several other industries, the adoption within space is a natural step. In some cases, the software ecosystem has still been forgotten. Traditional users in the industry have expected support for operating systems such as RTEMS, and components have been proposed for space applications using the ARMv8-R architecture, which is targeted towards automotive, but lacks support for a memory management unit (MMU), a feature that many users within the space industry have come to expect and rely upon.

At CAES we recognize the importance of the space industry to be able to leverage software developed in other, larger industries and the need to support the same level of debug support that developers in other industries have come to expect. We also see the need to continue to support the LEON line of processors and that improvements in that ecosystem will benefit the whole line of standard products, past and future. As a result, we are increasing focus on software ecosystem developments in parallel with our next-generation GR765 LEON5FT processor and we have also initiated our RISC-V line of processor products, complemented by software development efforts and partnerships.

At the time of writing, our assessment is that SPARC is the architecture that provides most flexibility when applying it in space industry. Qualified software environments (a recent example is the European Space Agency (ESA) funded effort to create a qualification data package for RTEMS) exist for SPARC and the same efforts have not yet been put into ARM or RISC-V based systems. At the same time, operating systems such as Linux, Zephyr and VxWorks that are widely used in other domains are available for SPARC. With the availability of more traditional qualified software for RISC-V, the benefits of being able to leverage a larger amount of software from other industries is likely to give an advantage to RISC-V.

eFPGA

Following the success of the Xilinx Zynq and the ongoing French NG-Ultra/Dahlia development that combine a hard processor subsystem together with FPGA fabric, inclusion of eFPGA fabrics has been met with positive feedback from most customers.

One difference between the products mentioned above and future CAES SoCs such as the next-generation GR765 LEON5FT is that the GR765 provides a complete SoC with communication controllers as part of the hard system. The GR765, being based on the GR740 SoC, has been defined as a complete system on its own and should, subject to CAES' ability to listen to user needs, be a viable product without the eFPGA. If an eFPGA will be utilized in future roadmap products it would likely be to implement IO glue logic for bespoke interfaces, such as glue logic between a microprocessor and sensors, and for accelerator functions. Having this integrated eFPGA could eliminate the need for an external FPGA, for some applications, without paying high price of having all the communication controllers implemented in an eFPGA fabric.

An interesting application of eFPGA is to connect it with an on-chip SerDes. Obtaining high-frequency input data from a sensor may require filtering before it is feasible to process the sensor data in software. Creating a generic filter to meet various application requirements is very challenging. However, by connecting a SerDes to an eFPGA fabric, where logic can be implemented to do data reduction, could be more efficient (in size weight and power (SWAP) and price) than utilizing an external FPGA.

Currently, the baseline specification for a FPGA fabric as we intend to implement it in next generation products is to have approximately 32k LUTs, allowing them to replace small and mid-size FPGAs that are today being applied in space applications. This FPGA fabric will allow to introduce glue logic using on-chip resources and accelerator functions such as support for Ultra-long FFTs that are typically efficiently implemented in a FPGA fabric while requiring a substantial amount of general-purpose processing power.

Cybersecurity

The space industry has not been without cybersecurity requirements and some applications have had very stringent requirements. An observed trend is that flow down of cybersecurity requirements is now affecting a broader range of missions and requests for functionality such as secure boot (checking the authenticity of software before running it) are now also coming from customers who traditionally have not considered cybersecurity aspects for the avionics at all.

CAES is currently assessing how to best support secure boot. In connection with adding secure boot support we are also assessing inclusion of secure elements in the design that would provide authentication and cryptography functions, including key storage that is secured from software access. Current projects are running to evaluate a Common Criteria security evaluation of platforms that are designed using our IP, which could then be combined with software that has been evaluated per the same flow, bridging the gap that exists in cybersecurity between hardware and software.

In connection with adding secure boot support, we are also assessing inclusion of secure elements in the design that would provide authentication and cryptography functions, including key storage that is secured from software access. AES-256 encryption/decryption is among the approved security functions that will be targeted.

System partitioning – containers/enclaves

In line with the current use of hypervisors to provide separation in single- and multi-core systems there is an expressed need for hardware platforms to provide partitioning between software instances. This is driven both by software verification and validation costs, where modification of one software image in a multi-core system should not lead to the need to rerun verification and validation for all software images running on the same SoC, and by cybersecurity requirements.

CAES Gaisler is addressing this by introduction of architectural features to provide timing isolation and introduction of secure enclaves to support functional partitioning, covering as follows:

- CPU core extensions: The identifiers allocated by the separation kernel to guests or processes running on the processor require several extensions to the processor cores. A hardware/software interface for assigning and controlling the identifiers needs to be defined. When software running on the processor performs an access to the on-chip bus then the identifier needs to be propagated as sideband information to the access.

- IO subsystem extensions: The system must support associating each IO unit to at least one ID. This can be accomplished by introducing wrappers to IO units and does not require an extension of each peripheral core. For some peripheral cores it is expected that further future extensions will be made, so that for example different virtual channels on a communication link can be associated to different IDs.

- IOMMU extensions: The IO memory management unit (MMU) will be extended to support a page table format suitable for RISC-V systems and to associate page tables to enclave IDs. In addition to this, performance counters will be added to support the separation kernel software in measuring access budgets.

- Peripheral register areas and memory areas: Access restriction based on allocation of peripheral and memory areas to different IDs needs to be supported by the SoC.

Hardware support for software partitioning has been standardized for single and multi-core, in the form of MMU and Physical Memory Protection (PMP) functionality. This functionality partitions the system from the perspective of software running on one processor but does not extend to peripheral units and specifically not to peripheral units that are able to perform direct memory accesses (DMA). This means that the specified functionality can protect software instances from directly accessing memory that belong to another software instance, but it does not prevent a software instance from controlling a DMA capable peripheral to access otherwise restricted memory. To be able to create a secure enclave, covering both software elements and peripherals in the system, additional functionality is required. Traditionally, an IOMMU is employed to restrict which memory areas that a peripheral can access.

Next-generation component: LEON5FT GR765

CAES Gaisler's next-generation LEON component is developed under the product name GR765. The GR765 builds in the successful GR740 architecture and the baseline architecture for the SoC architecture is as follows (enhancements from the prior generation are in **bolded text**):

- **Octa-core** rad-tolerant SoC device
  - **LEON5FT** with dedicated FPU and MMU
- 2 MiB L2 cache, 256-bit cache line, 4-ways
- DMA controllers
- DDR2/3 SDRAM memory I/F (+32 checkbits)
- SpaceFibre x4+ lanes 6.25 Gbit/s, simpler protocols
- **12-port** SpaceWire router with +4 internal ports
- 32-bit 33 MHz PCI interface
- 2x 10/100/1000 Mbit Ethernet
- 3x 10/100/1000 Mbit Ethernet with TSN and TTE support
- Debug links: Ethernet, JTAG, SpaceWire
- **2x** MIL-STD-1553B, 2x **CAN-FD**, **12** x UART
- **2x** SPI master/slave, GPIO, Timers & Watchdog
- 2x I2C interface
- NAND Flash controller interface
- FPGA Supervisor interface
- High-Speed Serial Link support: SpaceFibre x4+ lanes 6.25+ Gbit/s
- SoCBridge (interface for controlling FPGA companion devices)
- eFPGA (TBD)
- High-pin count – LGA1752
- Reduction of pin sharing

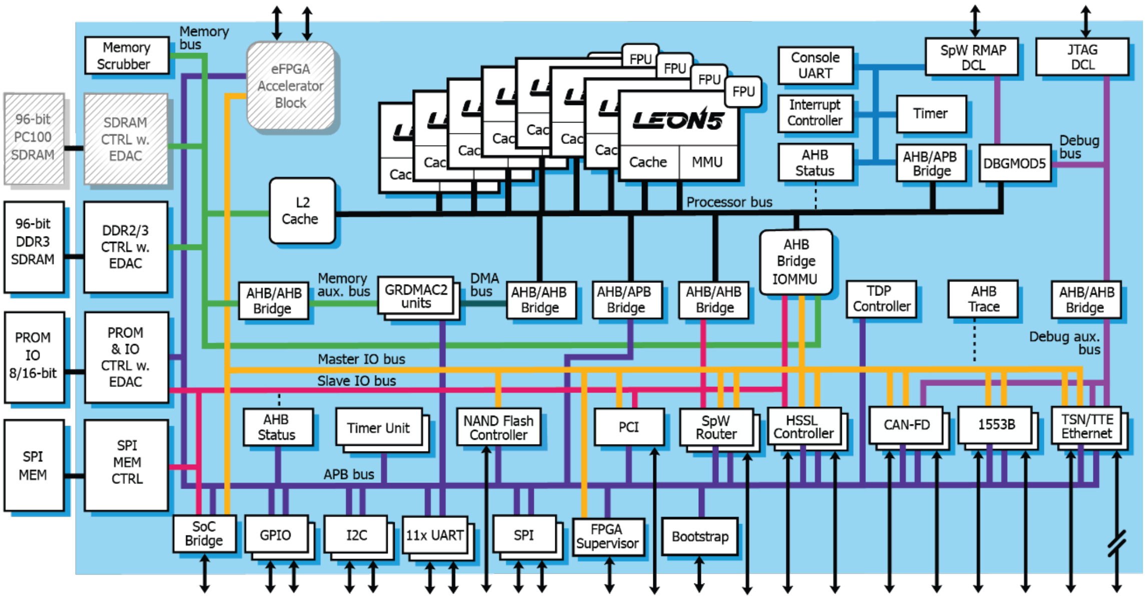


Figure 5 GR765 block diagram

LEON5FT processor core

To meet customer requests for increased computational performance, the GR765 includes the LEON5FT processor instead of the GR740's LEON4FT processors. LEON5 provides backward compatibility for most software implementations that have targeted LEON3 and LEON4 processors.

The processor core pipeline design of the LEON5 is significantly enhanced compared to earlier LEON3 and LEON4 processors, and evaluations show that LEON5 can provide up to 85% faster execution for single-threaded integer benchmarks compared to LEON4. The main new feature of the LEON5 pipeline is the **dual-issue functionality**, allowing up to two instructions per cycle to be executed in parallel in the processor. To support the increased issue rate of the pipeline, the LEON5 has **advanced branch prediction capabilities**. The cache controller of the LEON5 supports a store buffer FIFO with one cycle per store sustained throughput, wide AHB slave support to **enable fast stores and fast cache refill**, as well as several other enhancements.

The processor supports the MUL and DIV instructions, an IEEE-754 floating-point unit (FPU) and Memory Management Unit (MMU). The cache system consists of separate I/D multi-set Level-1 (L1) caches with up to 4 ways per cache, and an optional Level-2 (L2) cache for increased performance in data intensive applications.

The LEON5 processor has the following features:

- SPARC V8 instruction set with V8e extensions and compare-and-swap
- Advanced 8-stage dual-issue pipeline
- Complex dynamic branch predictor and small branch target buffer
- Addition of Late ALU to decrease pipeline stalls
- 64-bit single-clock load/store operation
- 64-bit 4-port register file
- Hardware multiply and divide units
- Hardware floating-point support
- Separate instruction and data L1 cache (Harvard architecture) with snooping
- SPARC Reference MMU (SRMMU) with TLB
- Advanced on-chip debug support with instruction and data trace buffer, and performance counter
- Symmetric Multi-processor support (SMP)
- Power-down mode and clock gating
- High performance:
  - Dhrystone*: 3.23 DMIPS/MHz (-O3, inlining allowed)
  - Coremark* : 4.52 CoreMark/MHz (-O3,-funroll-all-loops -finline-functions -finline-limit=1000)

    * All the results generated using BCC 2.0.7 toolchain

Example of architectural improvement: data aggregation

A common use case for CAES microprocessor and microcontroller products is for the processors to collect information from one or several data sources and either do computations on the data or deliver it to other parts of the system. In many cases this data is obtained through relatively low speed interfaces such as UART, SPI, and I^{2}C. There interfaces have traditionally been accessed by software through memory-mapped register and streaming data through this leads to a significant amount of effort on the general-purpose processor compared to the amount of data moved. To offload the processors from these tasks, the GR765 includes **programmable DMA controllers**, which can continuously poll the controllers for the previously mentioned communication interfaces and allow a more efficient use of the SoC, reducing the need to use software to collect data and freeing up processor resources to do computations.

Example of architectural improvement: main interconnect

The GR740 makes use of a single-layer bus to connect the processor cores and communication controllers, via an IOMMU, to the L2 cache. Due to this shared bus resource, there is timing interference between the processor cores and between DMA traffic and the processor core bus accesses. The GR740 reduces this interference by applying techniques where bus accesses that case a miss in the L2 cache are given a reply that leads to the requesting IOMMU or processor core to refrain from using the bus until the data is available. This allows the L2 cache to continue to serve accesses that hit in the cache while doing miss processing (fetching data from main memory). While the cache allows to serve cache hits while processing cache misses, it still causes some interference between, since the L2 cache only has one port that can serve one incoming request at a time.

The GR765 introduces additional processor cores connected to the same cache while **reducing the amount of interference** on the L2 cache port. The interconnect between the L2 cache, processor cores and IOMMU has been extended and makes use of **multiple parallel ports**. This improves available bandwidth and also allows for special modes where a connection is dedicated to one processor, isolating it from interference from DMA traffic and accesses from other processor cores and the L2 cache level.

Example of architectural improvement: interfacing companion FPGAs

Companion FPGAs are commonly used together with LEON microprocessors to provide IO expansion and accelerator functionality. In applications with the GR740, companion FPGAs have typically been interfaced using the memory-mapped IO interface for low-speed control, PCI for control and data transfers, or SpaceWire for control and data transfers. These options remain available also in the GR765 architecture, however the GR765 also **provides additional options for connecting companion FPGAs**. The high-speed serial link interfaces (**HSSIs**) can be used to obtain a high-bandwidth connection and a new interface named **SoC Bridge** has been introduced to simplify board design, FPGA design, and software development. SoC Bridge provides a bridge between the GR765 and a companion FPGA (with a SoCBridge IP core included in the FPGA). The interface provides a memory-mapped window between the GR765 and FPGA system, using fewer pins and less FPGA resources required compared to PCI.

Comparison table

A high-level comparison table of the GR740 versus the GR765 is shown below.

Parameter	**GR740**	**GR765**
Technology	ST C65SPACE (65nm bulk CMOS)	STM 28nm FDSOI
Availability	CCGA now, PBGA in development	Prototypes: 2023, flight models: 2024*
Temp range	MIL	MIL
Quality	QML	QML equivalent (plastic), QML target (ceramic)
Rad tolerance (SEL)	125 MeV-cm²/mg	TBD (comparable to GR740)
Rad tolerance (TID)	300 krad	50 krad (TBC)
Rad tolerance (SEE)	1E-5 events/device/day	TBD (comparable to GR740)
Operating frequency	250 MHz (WC corner)	1 GHz
Power	< 2 W @ room, < 3 W	Target < 3 W
Processor core	LEON4FT 32-bit SPARC V8	LEON5FT 32-bit SPARC V8
Number of cores	4	8
Performance, peak, MIPS	1000	16000
Performance, peak, DMIPS	1900	25840
Performance, estimated, in CM	1970	28k
Primary memory interface	SDRAM	DDR2/3 SDRAM, DDR4 SDRAM TBD
L2 cache size	2 MiB	2 MiB
eFPGA	No	TBD - Target 32k LUT eFPGA
High-Speed Serial Link	No	6.25 - 12 Gbps, 4 lanes SpFi, Wizlink, PCIe (TBD), SerDes
SpaceWire router	8 external ports	12 external ports
Parallel PCI	Yes	TBD
Ethernet	2x 10/100/1000	2x 10/100/1000; 3x TSN/TTE (TBD)
MIL-STD-1553B	1	2
CAN	2.0B	FD
UART	2	11 + 1
SPI	1	2
I2C	No	2
NAND Flash interface	No	Source sync, NV-DDR2/DDR3 TBD
Package	CCGA625 / PBGA625	LGA1752 plastic first, ceramic later

Table 2 GR740 vs. GR765

*Note: schedules are subject to change

The case for RISC-V in space applications

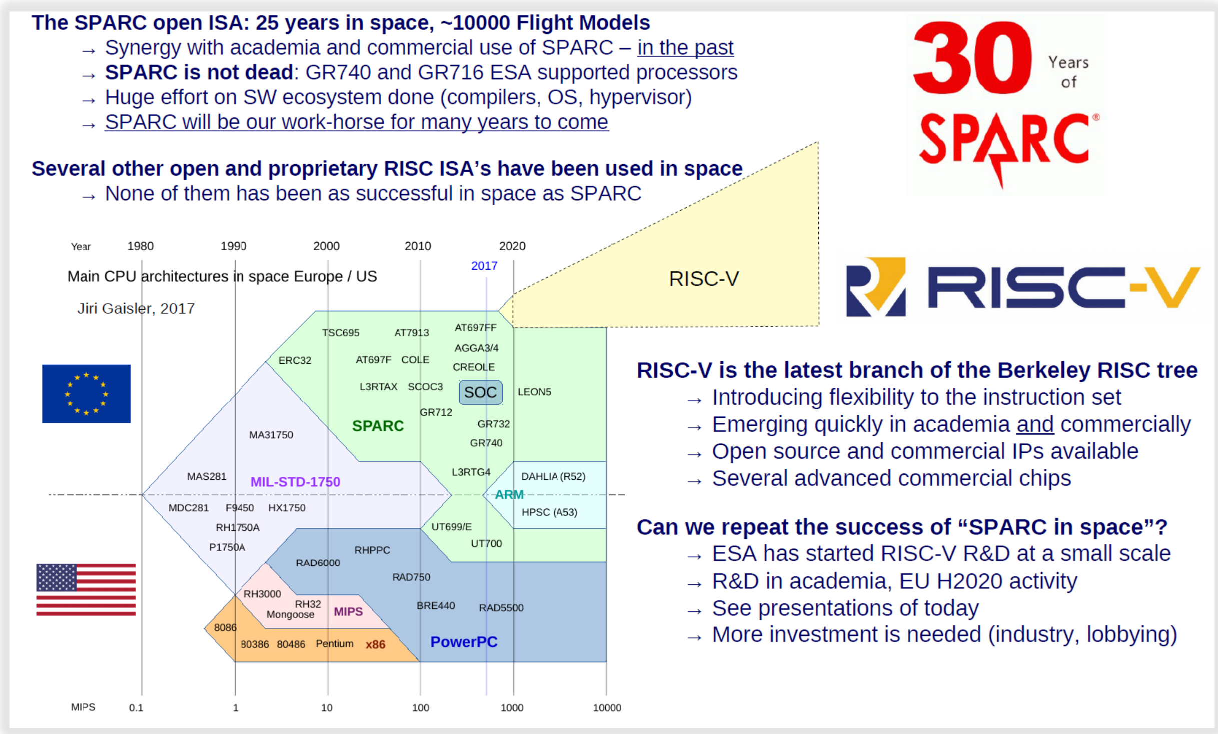
CAES Gaisler Products has benefited from the success of the SPARC architecture within space applications. SPARC V8, and specifically LEON processors, have been adopted and continue to be selected for missions worldwide. The architecture is dominant in Europe, has a significant footprint in the Americas and Asia, and is also being adopted in emerging space industries.

The success of SPARC can be attributed to several reasons, including the following:

- Spin-in of commercial technology, the space industry could take advantage of software development taking place in other bigger industries
- Non-proprietary architecture, allowing implementations from several sources
- Flight silicon available from several vendors: Microchip (Atmel), CAES and CAES Gaisler Products, TTTech, Thales Alenia Space, Airbus Defense and Space
- Global acceptance: ESA, NASA, ROSCOSMOS, JAXA, ISRO, KARI, DLR, CNES, INVAP, ASI, UAE, BMTI

The effects of wide adoption of one architecture have led to tremendous savings in hardware and software development costs, increasing impact of technology developments and leaving more funds to do science.

Today there are two main alternatives to SPARC as follows: (1) ARM and (2) RISC-V. RISC-V has gained a significant amount of momentum in other industries and represents a modern alternative with the same openness as was the case for SPARC in the 1990's. Figure 6 below shows an example of the view of SPARC and RISC-V from staff at the European Space Agency.



The SPARC open ISA: 25 years in space, ~10000 Flight Models

→ Synergy with academia and commercial use of SPARC – in the past
→ **SPARC is not dead**: GR740 and GR716 ESA supported processors
→ Huge effort on SW ecosystem done (compilers, OS, hypervisor)
→ SPARC will be our work-horse for many years to come

Several other open and proprietary RISC ISA's have been used in space

→ None of them has been as successful in space as SPARC

RISC-V is the latest branch of the Berkeley RISC tree

→ Introducing flexibility to the instruction set
→ Emerging quickly in academia and commercially
→ Open source and commercial IPs available
→ Several advanced commercial chips

Can we repeat the success of "SPARC in space"?

→ ESA has started RISC-V R&D at a small scale
→ R&D in academia, EU H2020 activity
→ See presentations of today
→ More investment is needed (industry, lobbying)

Figure 6 Slide from R. Weigand, ESA ESTEC Microelectronics section

The prospect of aligning on one open architecture in both Europe and the US is interesting due to the synergy effects that can be attained where funding for hardware development being applied in both EU and US will accelerate one another and software development of niche software for the space industry will have a market in both Europe and the US.

The ESA have funded R&D activities to explore RISC-V. The activity "Introduction of Fault-Tolerant Concepts for RISC-V in Space Applications" included an end user evaluation from the company Qinetiq Space (NL). The conclusion was that an implementation of the RISC-V ISA with the IMAFD extensions (support for Integer, Multiply, Atomics, and Floating point operations) matches contemporary LEON implementations in terms of functionality and that the RISC-V extensions H, B and V (Hypervisor, Bit manipulation, and Vector extensions).

Porting of hypervisors is included as an activity within the funded projects.

CAES believes the success and lessons learned for the SPARC-V8/LEON can be repeated to with the RISC-V/NOEL-V/GR7xV to meet the space industries future needs for more computing and more seamless ecosystem while addressing SWAP.

Next-next-generation architecture: NOEL-V RISC-V GR7xV

Figure 7 shows an overview of the next-next-generation multi-processor SoC (MPSoC) that is in development at CAES Gaisler Products, with support from the Swedish National Space Agency (SNSA) within ESA's ARTES program. The planned end product is a multi-processor SoC component that contains a multi-processor system, computing accelerator, embedded FPGA fabric and communication controllers. The product is complemented by a software ecosystem consisting of toolchains, debug tools and simulators.

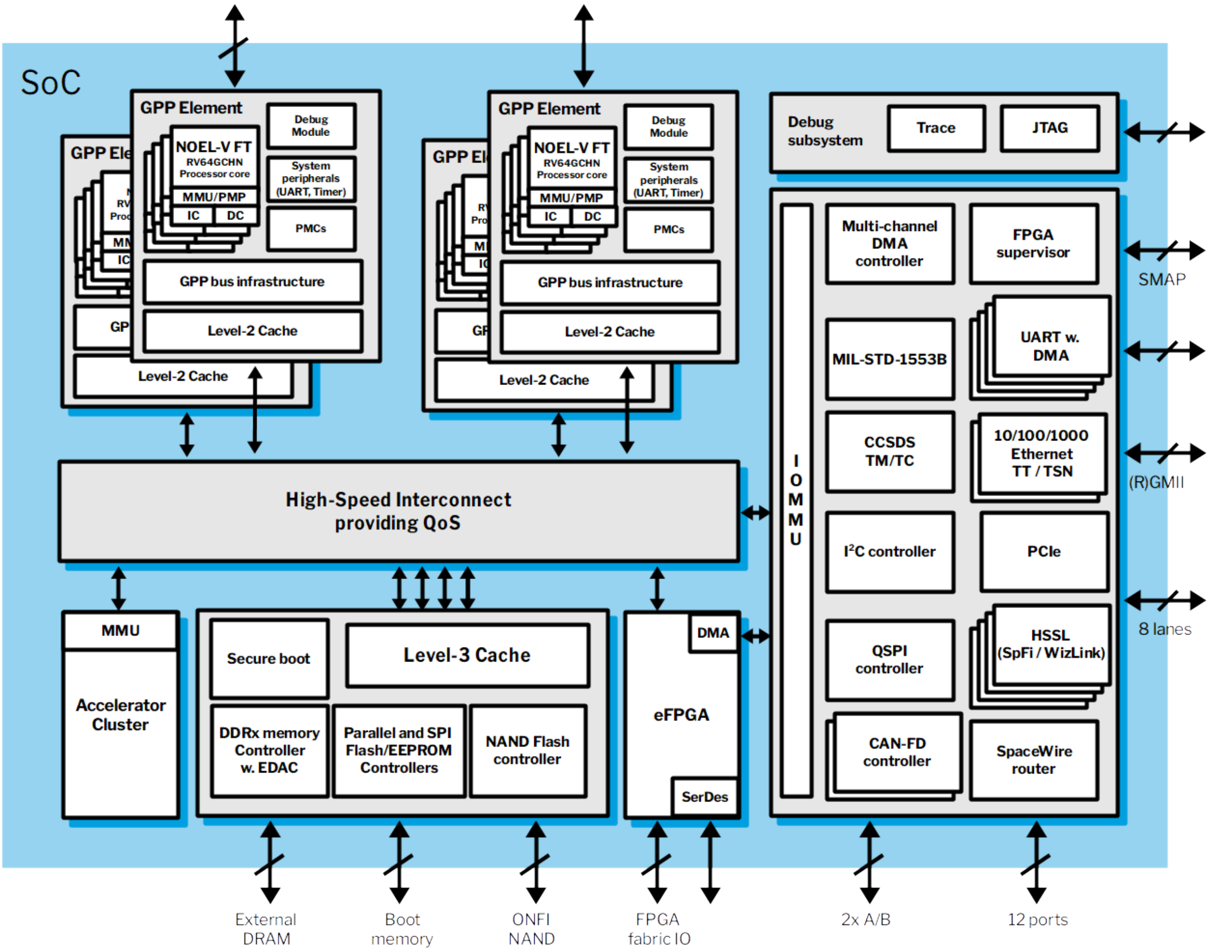


Figure 7 GR7xV block diagram

The architecture has been defined to allow management of the MPSoC using off-the-shelf operating system controlling all processor clusters in the design. The architecture also allows partitioning of software instances onto individual clusters and separation of instances within one cluster. The intent is to provide a solution for general purpose payload processing applications and for centralized architectures where users may apply the GR7xV with mixed-criticality workloads such as platform functions, combined with a software GNSS receiver, combined with sensor data processing.

Operation

In one extreme, one operating system is controlling all the processor cores leaving it up to the operating system how to schedule processes and relying on processor management units and the IO management unit to enforce functional separation between instances. In the other extreme, each processor has a dedicated software instance and both functional and timing separation is required for each instance.

The foreseen use of the GR7xV is to apply a separation kernel that runs in hypervisor mode on the processors. The separation kernel will host guest operating systems that run on one or several processor cores and may also have a schedule where different partitions are active on the system over time.

The GR7xV has been specified to allow safety critical tasks to run on the general-purpose processors as opposed to dedicating a real time subsystem of processors. To accomplish this, several hardware features must be provided, including:

- A definition of a SoC level configuration to prevent timing interference between processor cores.
- Management of caches and buffers (such as branch prediction) buffers to prevent casual effects between software partitions (state changes from one partition may unintentionally not affect state of the next partition).
- Timing interference on shared resources, such as a shared bus, must be avoided or upper bounded.
- Processor core local activities may not be affected by activities from other processors (through for example cache evictions due to coherency traffic)

Timing isolation is achieved within the processor elements by being able to reconfigure the interconnect between the processors and L2 cache to use dedicated paths for each processor and the cache. This isolation feature does impact timing performance but does also guarantee that timing interference is removed within a GPP element. In the next level, the high-speed interconnect can also be configured to support timing isolation, again at a maximum performance impact for the overall system. It is expected that the majority of safety critical applications can be supported by employing isolation within a general-purpose processing element and that the application of full isolation over the system will be used by a limited set of applications.

Processor elements

The GPP elements in the MPSoC architecture each constitutes a multiprocessor system. Each GPP element has several microprocessors, system peripherals such as UART interrupt controller and performance monitoring counters. Each processor core in a GPP element has the following configuration:

- RISC-V 64-bit core implementing RV64IMAFDGCHV extensions:
  - RV64I 64-bit Base integer instructions
  - MUL/DIV (M)
  - Atomics (A)
  - Single/Double Float (FD)
  - Compressed instructions (C)
  - Supported modes: M/S/U
  - MMU (Sv39/Sv32)
  - Hardware hypervisor support (H)
  - Vector operations (V):
- Physical Memory Protection (PMP)
- 16 KiB I-cache, 16 KiB D-cache
- TBD KiB Tightly Coupled Memory

The GR7xV is currently in the architectural design phase and additional extensions are being defined based on voice of the customer efforts where end users are interviewed. The primary benefit of the new architecture, next to being able to leverage RISC-V software from other domains/markets, is the improved support for partitioning and the availability of the on-chip accelerator cluster. The target technology selection for GR7xV will determine the delta in terms of processing and power consumption compared to the GR765 and additional information will be made available about this as both product developments progress.

Hardware and software-based fault-tolerance theory of operation

The fault-tolerance features of the NOEL-V RISC-V processor follows the same philosophy that has previously been applied by CAES for our LEON microprocessors (used in products like the UT699, UT700, GR712RC, GR740) where error correction is done transparently to software and software execution is allowed to continue uninterrupted in the presence of correctable errors. Detected uncorrectable errors lead to a software trap and corrupted state is never allowed to propagate outside of the microprocessor component. One difference between CAES designs and microprocessor designs developed for other domains/markets is reaction time on errors. For datacenter applications, it may be sufficient to detect that an error has occurred, discard the affected batch of work and restart the computations. Similar concepts with fault-tolerance solutions implemented in software, or lockstep operation with restart have a long recovery time. In contrast, the LEON and NOEL-V solutions provided by CAES do not impose long recovery times. The LEON and NOEL-V approach to fault-tolerance extends to peripherals and software libraries. This allows end users and applications to have a coherent handling of errors which can occur within the microprocessor core, peripherals, and external memory.

All on-chip memories in our designs are protected against Single Event Upsets (SEUs) and large memories such as the Level-1, Level-2 and Level-3 caches have built in scrubbers in the cache controllers to prevent accumulation of errors. All external memory interfaces are also protected by ECC and scrubbers.

Standard products from CAES protect against radiation effects using hardened implementation technologies and design solutions as described above. The approach is to provide redundancy and protection using hardware functions so that no special software design approaches need to be taken except for software routines that are optionally called on correctable errors and routines for handling detected uncorrectable errors.

CAES Gaisler is also introducing new memory controller IP cores as new memory devices are being utilized in the space market. Two recent examples are the FTADDR and the NANDFCTRL2. The first is a DDR2/DDR3 controller with an EDAC providing correction of up to two fully corrupted 8-bit lanes. FTADDR also provides a SEFI detection and recovery mechanism and is available now. (A variant to support DDR4 will be investigated in the future). The latter is an advanced NAND Flash controller IP core with EDAC and SEFI detection and handling, which is currently in development and already available in beta version.

Comparison table

Please see the table below for a high level comparison of the GR765 LEON5FT SoC and GR7xV NOEL-V RISC-V SoC, mainly for legacy LEON users, from the perspective of some key considerations (performance and software aspects).

Key consideration	GR765 LEON5FT	GR7xV NOEL-V RISC-V
reuse of code from prior LEON designs	yes	no
3rd party OS support	some	extensive
qualified flight software	yes, prior LEONs	no
3rd party software	some	extensive
ecosystem comparable to other industries	no	yes
computational performance	good	high
target DMIPS	26k DMIPS	>32k DMIPS
on chip accelerator	no	yes; 1.23 TOPS of INT8.32 for MDL, 19 GFLOPS FP16, 8.5 GFLOPS FP32
CPU core architecture	32-bit SPARC V8	64-bit RISC-V; RV64GCH configuration
number of cores	octa core	16-core
hypervisor support	limited	yes
support for partitioning	not applicable	yes
support of isolation	not applicable	yes
cyber security features (secure boot)	limited	yes
cache	L1/L2 (2+ MiB)	L1/L2/L3
external memory	DDR2/3	DDR2/3/4
eFPGA	yes (TBD)	yes
support of single core versions (& IP)	yes	yes
quality levels	QML targeted	TBD
fault-tolerance (internal memory protected against SEUs, EDAC for external memory)	yes	yes
radiation performance	comparable to GR740 *	comparable to GR740 *
core rail power	<3W	TBD (comparable to GR765 or lower)

Table 3 GR765 vs GR7xV

Notes:

- GR765 & GR7xV have different system architectures (including interconnect topology)
- GR765 & GR7xV have different processor core ISAs
- GR765 is an extension/improvement of GR740
- GR7xV adds additional processor cores, an accelerator cluster & extends fundamental building blocks (interconnect, cache coherency protocols, support for atomic operations, time & spatial separation, …)
- * radiation performance are expected to be similar (depends more on target technology)

Conclusion

Gaisler/CAES remains committed to providing space-grade microprocessors to target traditional space applications that require high-reliability components at established quality levels. New product generations continue the LEON line of microprocessors with increased levels of integration, improved performance, and improved power efficiency.

The continuation of the LEON line of processors is complemented by supporting activities to keep the software ecosystem up to date and to extend the software ecosystem. Through software compatibility, these improvements will also benefit earlier generations of LEON processors.

The LEON product line has been complemented by the NOEL-V RISC-V microprocessor, to enable the space industry to leverage software developed in other industries in the future. Examples include operating systems and software libraries developed by companies that are not present in the space market and focus on other industries where ARM and RISC-V are more prevalent. The NOEL-V inherits the LEON line’s fault-tolerance approach of allowing software to transparently continue execution in the presence of correctable errors.

At the time of writing, our assessment is that SPARC is the architecture that provides most flexibility when applying it in space systems. Qualified software environments (a recent example is the ESA funded effort to create a qualification data package for RTEMS) exist for SPARC and the same efforts have not yet been put into ARM or RISC-V based systems. At the same time, operating systems such as Linux, Zephyr and VxWorks that are widely used in other domains are available for SPARC. With the availability of more traditional qualified software for RISC-V, the benefits of being able to leverage a larger amount of software from other industries is likely to give an advantage to RISC-V.

The development of the company’s next-next generation architecture is being made with an increased focus on cybersecurity features, including functional and timing isolation features to partition systems.

Please note this document describes CAES Gaisler Products ongoing component development and explains the rationale for some architectural design choices for future roadmap products.