HighIntegritySystems

WITTENSTEIN

SAFERTOS® & IAR Embedded Workbench For RISC-V

Issue 1.2 - March 28, 2023

Written in partnership with IAR Systems.

WITTENSTEIN high integrity systems
Americas: +1 408 625 4712
ROTW: +44 1275 395 600
Email: sales@high**integrity**systems.com
Web: www.high**integrity**systems.com

Contents

List of Figures

List of Notation

BSP	Board Support Package
DAP	Design Assurance Pack
DHF	Design History File
FPU	Floating Point Unit
ISA	Instruction Set Architecture
MCU	Microcontroller Unit
MPU	Memory Protection Unit
MMU	Memory Management Unit
RTOS	Real Time Operating System
SIL	Safety Integrity Level
SOUP	Software of Unknown Provenance

CHAPTER 1 Introduction

1.1 What is RISC-V?

RISC-V is an ambitious programme to develop a universal Instruction Set Architecture (ISA) that is suitable for all processors, from small embedded microcontrollers to fast high performance computers. Furthermore, it is designed to be usable with a wide variety of programming languages and software stacks and accommodate all implementation technologies (FPGA, ASIC and dedicated silicon implementations).

Where other proprietary ISA's have suffered from changes made to support the requirements of a single party, the RISC-V foundation have stated that absolute stability of the base architecture is a prime requirement for RISC-V. Conversely, it is very difficult to absolutely define an architecture that is applicable to a wide range of target platforms, and can never be updated; therefore RISC-V is a modular ISA, the base ISA which is known as RV32I is frozen and will never change. However extensions and advanced features can be added as modules and need only be adopted if desired. This makes supporting a wide range of processors much easier, however it also presents challenges to tool suppliers and software suppliers as the target of the supplied software must be precisely defined.

The RISC-V ISA belongs to the RISC-V foundation which is an open non-profit foundation whose mission is to maintain the stability of the RISC-V ISA, evolve it and champion its adoption with hardware vendors.

1.2 What is SAFE**RTOS**®?

SAFE**RTOS**® is a safety critical, highly deterministic, embedded RTOS with an accompanying Design Assurance Pack that provides an easy route to achieving certification of SAFE**RTOS** once integrated into a Safety Product.

With an imperceptible boot time, SAFE**RTOS** provides a priority-based, pre-emptive, real time operating system. The number of priority levels is user-defined. The highly deterministic scheduling algorithm ensures that the highest priority Task that is able to run, is the Task which is selected to run.

Inter-Task communication and synchronisation is achieved using features including Semaphores, Mutexes, Queues, Event Groups and Task Notification. SAFE**RTOS** also supports a timer feature.

In addition to the Task Separation and Isolation feature, SAFE**RTOS** contains a range of other safety features.

SAFE**RTOS** performs thorough API input validity checking (as far as practically possible) in order to mitigate the risk of misuse by the host application. If the value of an API parameter is found to be invalid, the API function will not perform any action other than returning an error code indicative of the error encountered.

SAFE**RTOS** also performs the following run time integrity checking with the intention of facilitating the detection of data corruption:

- The execution context of a Task that is not in the Running state is stored on the stack allocated to the Task. The context of a Task will only be saved onto the stack of the Task if there is sufficient stack space remaining to hold the entire stack;
- A check is performed to ensure that the Task Control Block associated with the Task selected to enter Running state is valid. This is achieved by checking key data parameters against their inverted mirror copies;
- Prior to implementing the tick count value, a check is performed to test that the current tick count value remains at the last written and therefore expected value;
- Verification that the MPU modes and privileges are restored correctly.

A failure in any of these integrity checks will result in a call to the application Error Hook function.

Some features that are present in other operating systems have been removed from SAFE**RTOS** due to safety concerns. For example, SAFE**RTOS** does not perform any dynamic memory allocation operations, but instead requires the application to allocate a block of memory for SAFE**RTOS** during the initialisation sequence. Reference to this memory block is passed to SAFE**RTOS** via the API during the initialisation phase. Application designers are still able to use dynamic memory allocation within their designs.

SAFE**RTOS** is supplied with a Design Assurance Pack (DAP) which contains every design artefact produced during the full development life cycle, from development and safety life cycle plans, requirements specifications and design documents, to HAZOPS, the source code, all verification and validation documents and relating evidence. The full test harness, with user and safety manuals, is also supplied. The DAP has been certified to IEC 61508 SIL 3 and ISO 26262. For medical applications SAFE**RTOS** is supplied with a Design History File (DHF). The DHF has been independently validated for compliance with FDA510(k) Class III medical device standards and EN62304.

1.3 SAFE**RTOS** and RISC-V

SAFE**RTOS** is a software component that is provided as C code, therefore from a functional perspective, it should be readily usable within processors based on the RISC-V architecture. Where the target is a safety certified system, other factors must be considered, such as maturity of tool chains, and availability of processor parts targeted at safety applications. Safety processors usually provide features such as safety manuals, safety usage guidelines, lockstep operational modes and diagnostic safety libraries. RISC-V is an ISA and therefore neither provides or prohibits any of these safety features; the availability of suitable silicon implementations is key to the usage in the safety market.

This white paper will address some issues relating to the porting of SAFE**RTOS** as well as discussing the maturity of the ISA and its use in safety certified products.

CHAPTER 2 Porting SAFE**RTOS** to RISC-V

2.1 Porting SAFE**RTOS** to RISC-V

SAFE**RTOS** is an embedded RTOS that provides a common API irrespective of the actual underlying hardware. The software component is structured as a common kernel which provides the algorithms and software necessary to support scheduler state management, task creation and management as well as providing inter-task communication mechanisms such as Queues, Event Groups, Semaphores, Mutexes and Timers. The kernel is supported by a processor specific portable layer that provides the necessary support routines to interface with actual hardware to perform context switching, interrupt interactions and support for hardware specific features such as FPU's, MPU's or MMU's.

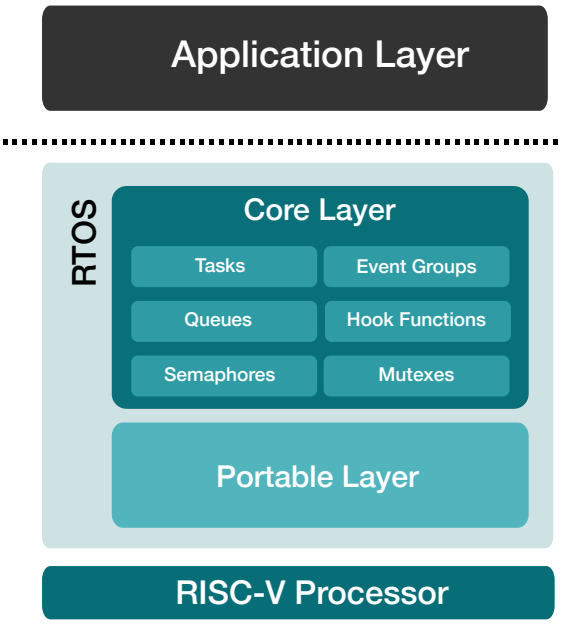

Figure 1. *RTOS Design*

SAFE**RTOS** has already been ported to many architectures and microcontrollers including devices based on ARM, PPC, ARC and MIPS, and the porting model is well established. Most vendors tune their devices to support specific industry needs and therefore the features included when porting SAFE**RTOS** to individual devices may vary even though the underlying architecture is the same.

When using an architecture like RISC-V or the ARMv7-M, the basic architecture or ISA is guaranteed to be common across a range of devices whatever the actual supplier of the microprocessor. This gives operating system suppliers like WHIS an opportunity to structure the port to reflect the common architecture and realise real technical benefits from applying common test strategies across a wide range of different physical processors.

The modular nature of RISC-V means that there are a number of "standard" extensions to the basic RV32I ISA standard. These include floating point operations, privileged architecture and support for 64 bit architectures. Even the extensions are governed by fixed specifications and there are opportunities to realise real savings through this as well.

CHAPTER 3 SAFE**RTOS** Certification with RISC-V

3.1 SAFE**RTOS** Certification with RISC-V

When producing a product comprised of hardware and software that will be used in a safety environment, many forms of testing and verification are typically required. Some (such as environmental testing) fall outside the scope of this paper, however software must be tested, hardware must be tested and complete product testing must also be performed. In addition to testing the product, it is also necessary to consider tooling and the effect of tool deficiencies on the final product

With respect to the SAFE**RTOS** product, it is a software component written in C and assembler. The Design Assurance Pack includes a complete test suite and full MCDC testing to ensure its correct operation in the selected operation environment irrespective of the compilation tools selected. Therefore, using SAFE**RTOS** in a RISC-V safety system presents no issues.

The SAFE**RTOS** code base is comprised of C and Assembler, the DAP however is developed using the tools selected by the host application developer and ultimately responsibility for compiler qualification lies with the host application developer. Compiler qualification is an issue in safety systems and therefore the effect of the available tools provided by IAR Systems or the RISC-V foundation themselves must be considered. Depending on the nature of the product being developed, the availability of certified tools can be a significant issue. At the time of writing, no certified tools are available for RISC-V.
.

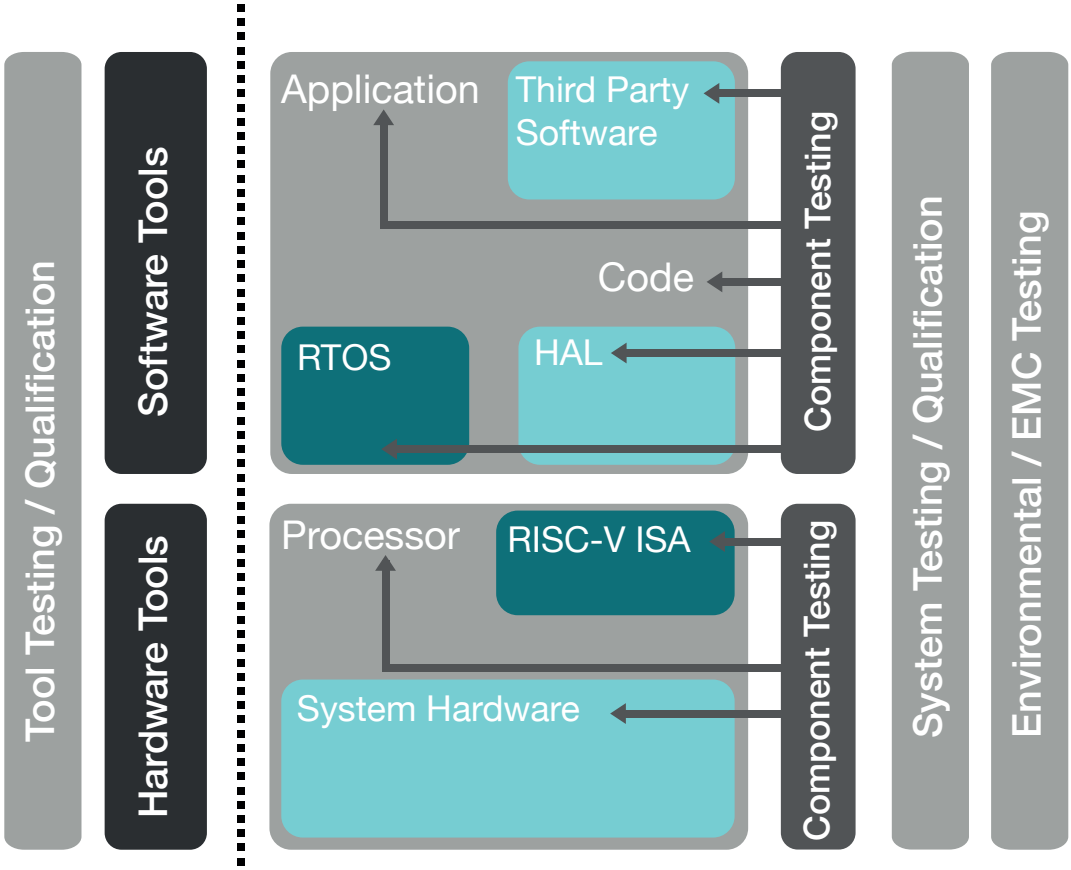

Figure 2. *RISC-V Ecosystem*

CHAPTER 4 IAR Embedded Workbench

IAR Embedded Workbench is a robust, flexible and complete platform for all aspects of embedded software development, with powerful functionality fully integrated, to support the entire development process.

IAR Embedded Workbench for RISC-V is built around the only available commercial compiler for RISC-V. Using a highly optimized compiler and a comprehensive debugger along with fully-integrated analysis tools will ensure the quality of the written code. IAR Embedded Workbench offers a complete set of build tools, code analysis (C-STAT) and debugging tools (C-SPY) from one feature-rich IDE, enabling shorter time-to-market and more time to spend on the innovative parts that will differentiate the end product.

IAR Embedded Workbench supports ISO/IEC 14882:2015 (C++14, C++17), ISO/IEC 9899:2012 (C11), ANSI X3.159-1989 (C89) and IEEE 754 standard for floating-point arithmetic.

C-STAT includes checks, complying with rules as defined by MISRA C:2012, MISRA C++:2008 and MISRA C:2004 and more than 250 additional checks mapping to issues covered by CWE and CERT C/C++.

Most of the key functionality of the IAR toolchain are held in a common cross-architecture platform. So when adopting the compiler to a new target many of the common code transformations and optimizations can be reused.
.

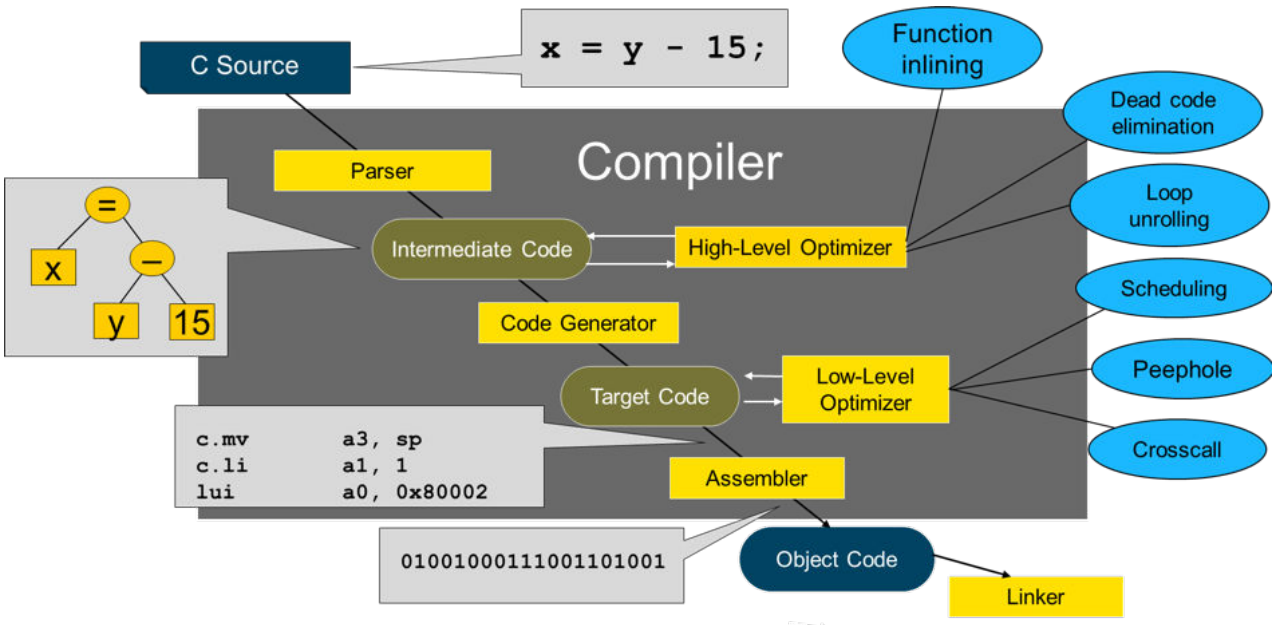

Figure 3. *IAR Toolchain*

4.1 Code Quality

IAR Systems is using many of industry-standard test suites to ensure compliance with relevant ANSI/ISO C/C++ standards. Additional test suites have been developed in-house to complement the industry-standards.

Some of the major test steps are;

- Plum-Hall (ANSI C)
- Dinkumware (Plauger) library tests “Dinkum C++ Proofer”
- Floating point tests; IAR Systems’ tests of float and double precision floating point numbers
- IAR Systems general compiler tests; test of volatile, and IAR Systems’ extensions to the C language
- General compiler regression tests of reported and fixed issues.
- Target specific compiler tests; test of bitfield, function/memory pointers, memory attributes and target dependent extensions to the C language
- Memory allocation order dependency tests: to verify that the compiler produces the same code output independent of the execution environment
- Assembler tests
- MISRA-C:1998 and MISRA-C:2004 tests

The test suites combined, add up to more than 75000 test cases. Each test is run multiple times for permutations of processor modes, memory models and optimization levels.

IAR Embedded Workbench for RISC-V, together with SAFE**RTOS** are a perfect combination for any safety critical application.

The RTOS awareness plugin for SAFE**RTOS** can be used to display a snapshot of tasks, queues, semaphores and mutexes each time the debugger is paused or single stepped. Two tables can be displayed, one to show task information and the other queue, semaphore and mutex information.

All the testing performed by product developers to verify system functionality and correct software operation relies on the correct operation of the underlying hardware. Virtually all ISA’s and devices have a published set of errata that in some cases is extensive. Verification of software in these environments requires that published errata are considered and appropriate workarounds used. RISC-V is an unambiguous formal ISA specification and as such allows for processor implementations that are amenable to verification. Much work is also being done within the RISC-V community to establish formal processor verification platforms for the architecture.

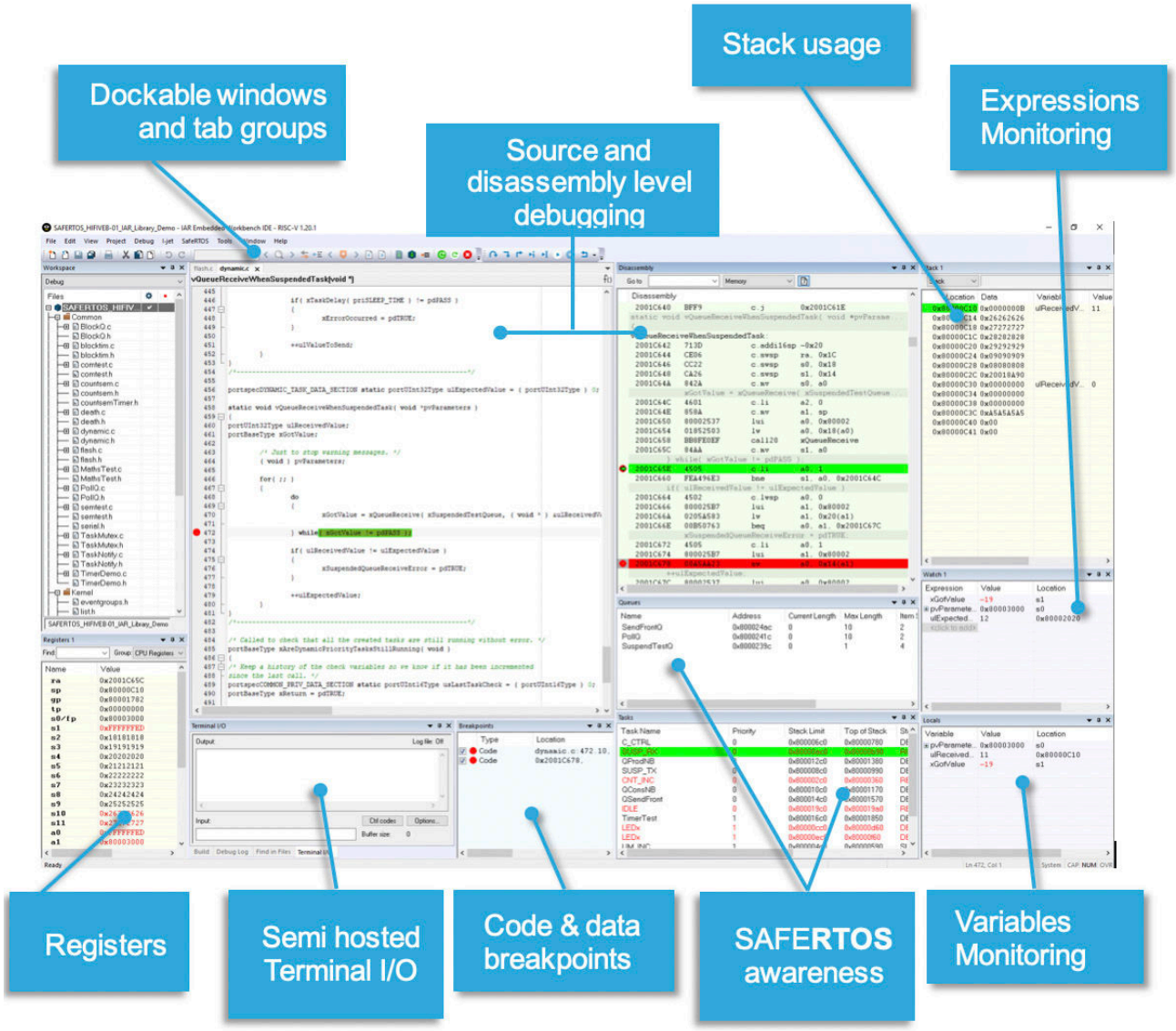

Figure 4. *IAR Embedded Workbench for RISC-V*

CHAPTER 5 Conclusion

5.1 Conclusion

SAFE**RTOS** is a software component that has been certified to IEC 61508 SIL-3 in many architectures and therefore presents no additional obstacles to producing a safety system that is based on a RISC-V processor. The extensive test suite and encompassing Design Assurance Pack verify SAFE**RTOS** in the environment under test. Using a highly optimized compiler and a comprehensive debugger, along with fully-integrated analysis tools, will ensure the quality of the written code.

IAR Embedded Workbench for RISC-V, together with SAFE**RTOS** are a perfect combination for any safety critical application using the RISC-V ISA.

Contact Information

Your Feedback

We would be pleased to receive your comments and any suggestions for improvement to our software and documentation. Please email us at: support@highintegritysystems.com

Contact WITTENSTEIN high integrity systems

Address: WITTENSTEIN high integrity systems
Brown's Court, Long Ashton Business Park
Yanley Lane, Long Ashton
Bristol, BS41 9LB
England

Phone:
Americas: +1 408 625 4712
ROTW: +44 1275 395600

Email: support@HighIntegritySystems.com

Website: www.HighIntegritySystems.com

Twitter: www.twitter.com/WITTENSTEIN_HIS

Contact IAR Systems

Address: IAR Systems AB
Box 23051, Strandbodgatan 1
SE-750 23 Uppsala,
SWEDEN

Phone: +46 18 16 78 00

Email: info.se@iar.com

Website: www.iar.com

Twitter: www.twitter.com/iarsystems