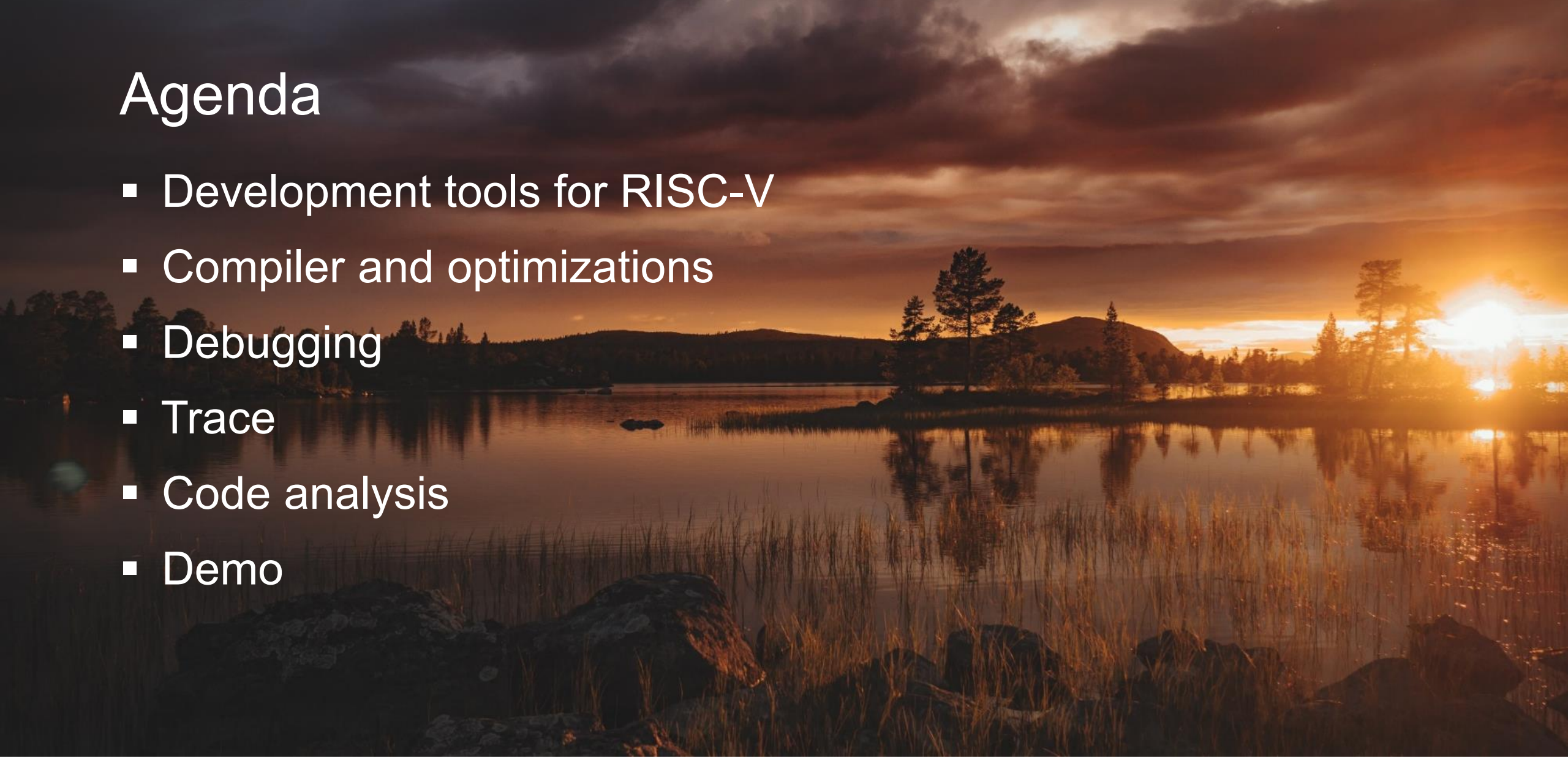
Stay ahead with the latest advances in RISC-V development tools

Shawn A. Prestridge
US FAE Team Leader, IAR Systems

IAR SYSTEMS



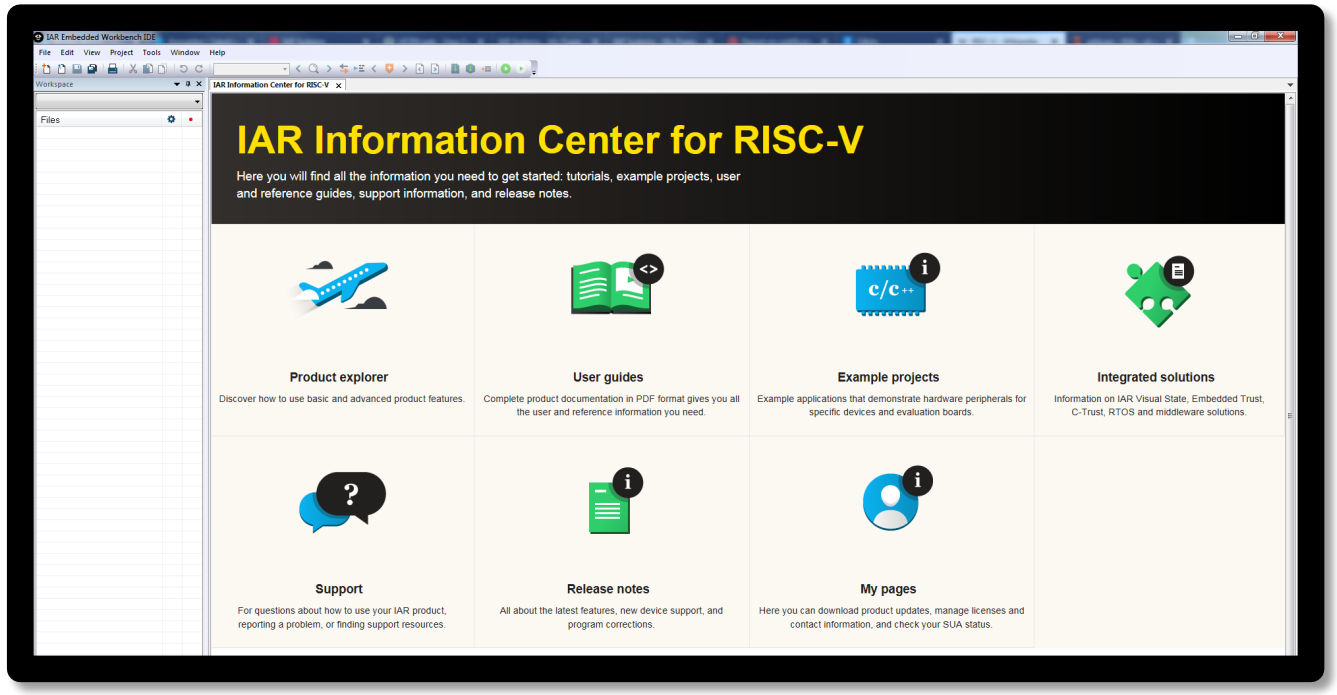
Agenda

- Development tools for RISC-V
- Compiler and optimizations
- Debugging
- Trace
- Code analysis
- Demo

Development tools for RISC-V

IAR Embedded Workbench for RISC-V

- Complete build and debug toolchain for RISC-V
- User-friendly IDE features and broad ecosystem integration
- Outstanding performance through sophisticated optimization technology
- Comprehensive debugger
- ISO/ANSI C/C++ compliance with support for C18 and C++17
- Integrated static analysis

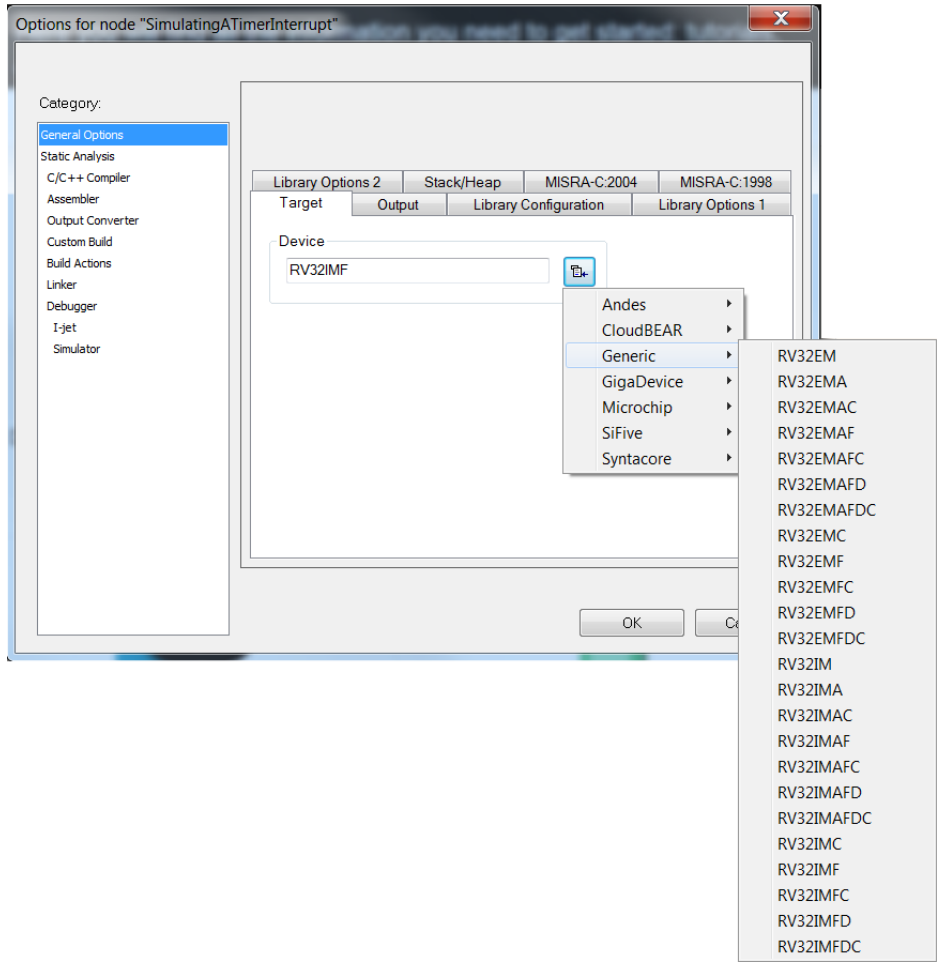


Device support

Supported base instruction sets:
- RV32I Base Int instruction set
- RV32E Base Int instruction set (embedded)

Supported extensions:
- M integer mul & div
- A atomic instructions
- F single precision float
- D double precision float
- C compressed instructions
- P DSP and Packed SIMD

Device support for RISC-V devices from:
- Andes
- CloudBEAR
- Microchip
- SiFive
- Syntacore
- GigaDevice



Compiler and optimizations

Compiler design

- Proprietary design based on over 37 years of experience
- Based on a platform that is common among different targets to handle global optimizations, etc.
- Target unique backend for specific adaptations and optimizations
- RISC-V specifics:
  - Primary focus will be on adding standard extensions
  - Initial prioritization is on code size
  - Canary stack protection

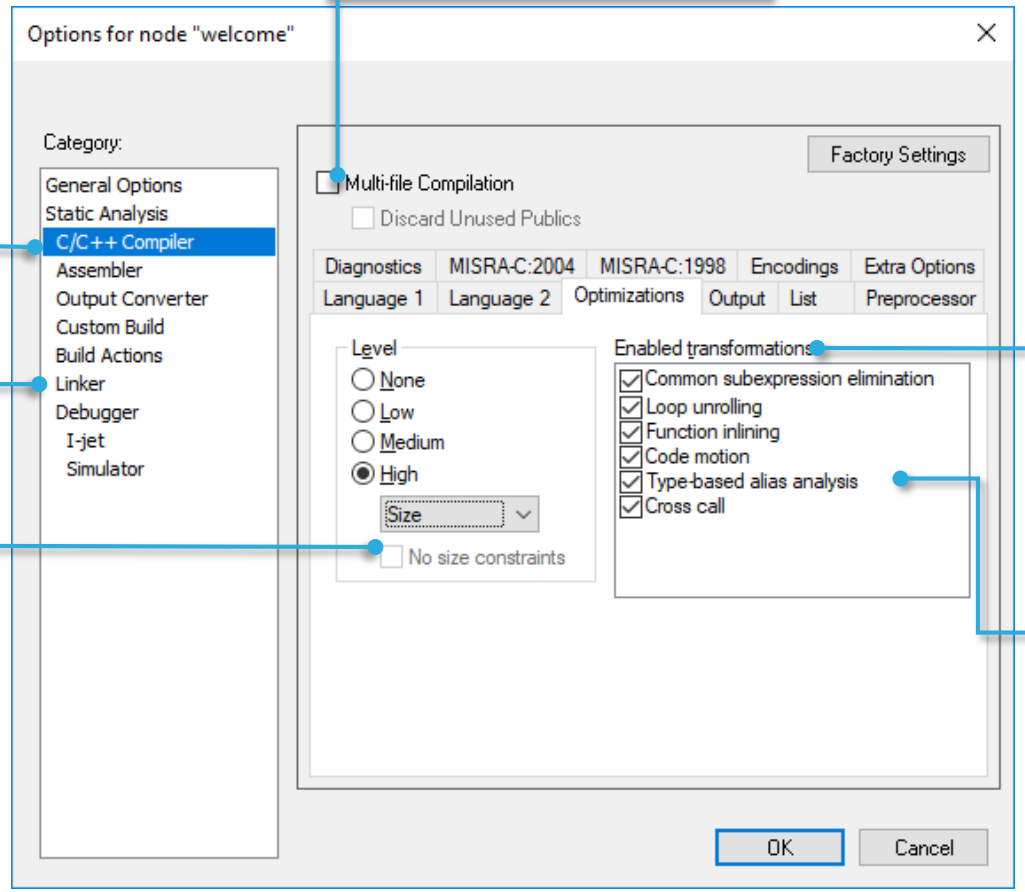
Compiler technology

Multi-file compilation allows the optimizer to operate on a larger set of code

Multiple optimizations levels for code size and execution speed

The linker can remove unused code

Option to maximize speed with no size constraints



Major features of the optimizer can be controlled individually

Balance between size and speed by setting different optimizations for different parts of the code

Language standards

- ISO/IEC 14882:2015 (C++14, C++17)
- ISO/IEC 9899:2012 (C11, C18)
- ANSI X3.159-1989 (C89)
- IEEE 754 standard for floating-point arithmetic

Well-tested

Commercial test suites

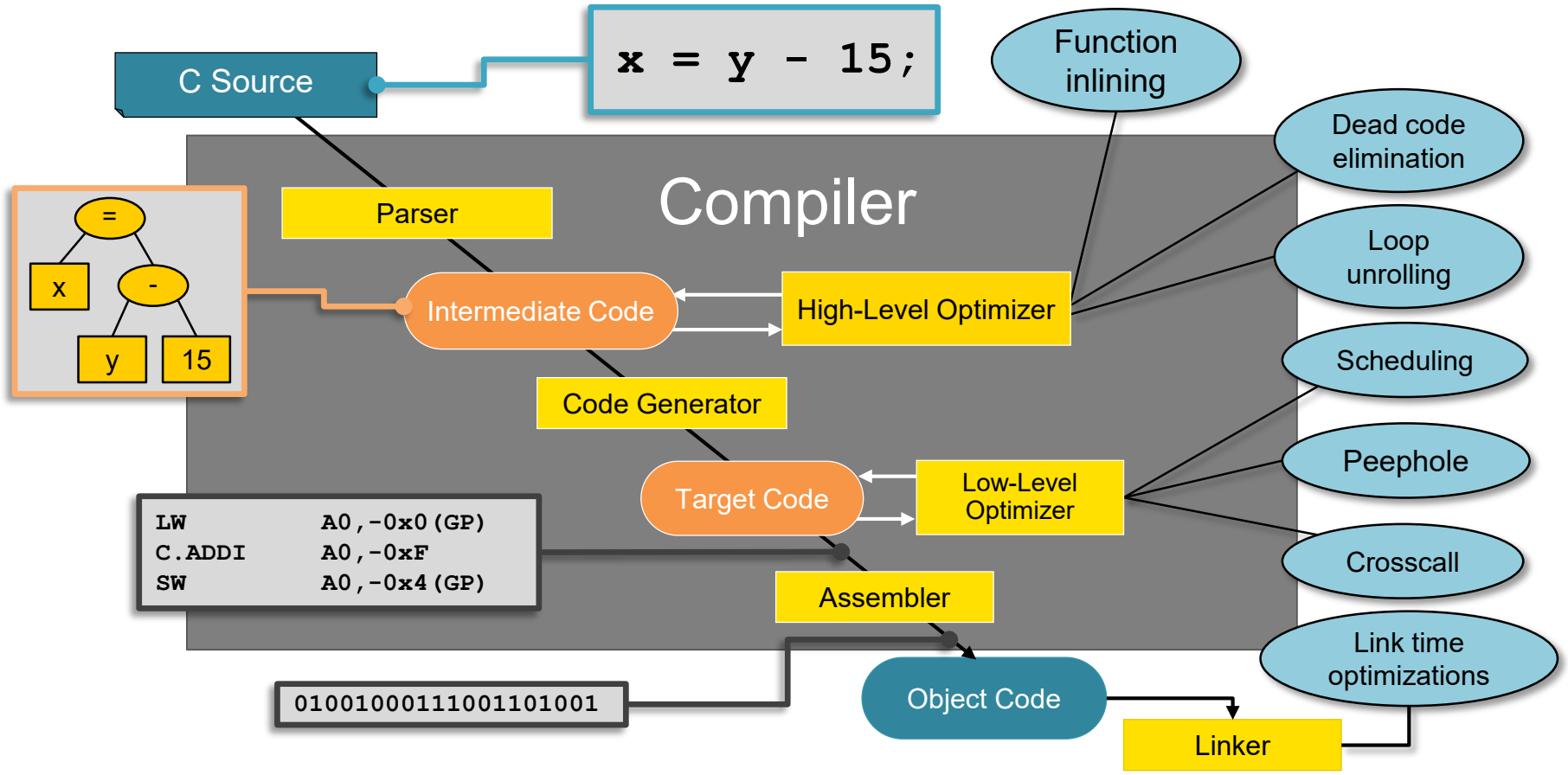
- Plum-Hall Validation test suite
- Perennial EC++VS
- Dinkum C++ Proofer

In-house developed test suite >500,000 lines of C/C++ test code run multiple times

- Processor modes
- Memory models
- Optimization levels

IAR SYSTEMS

Compiler optimizations

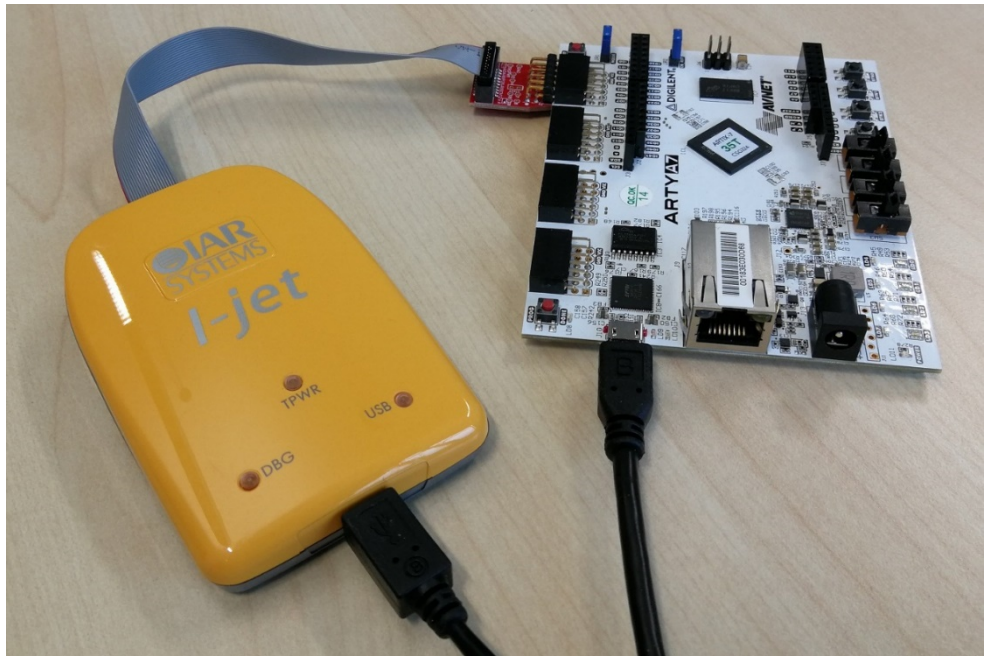


Debugging

Debugging

The IAR Embedded Workbench for RISC-V offers a fully integrated debug solution

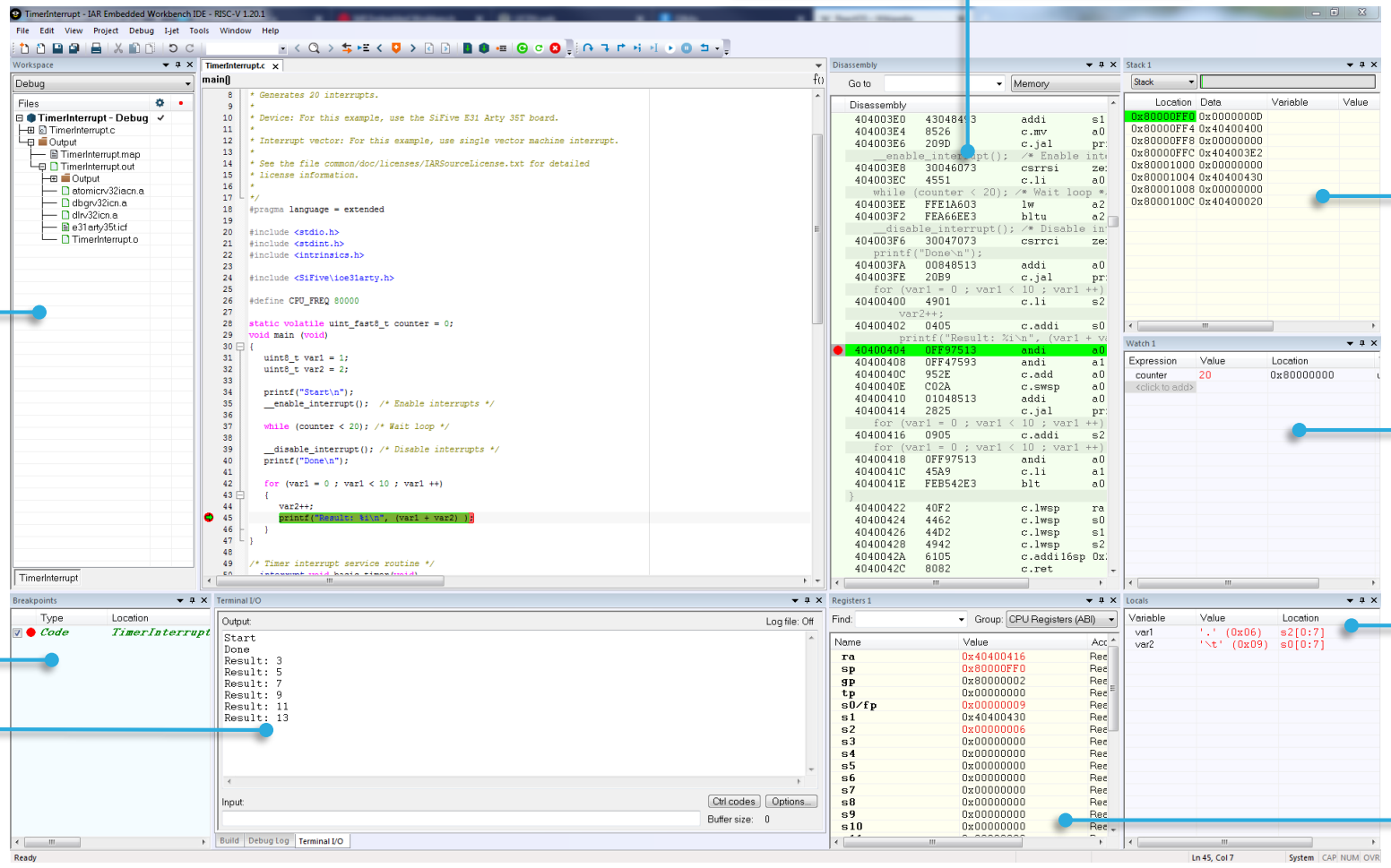
- Built on the same C-SPY debugger interface that is used in the IAR Embedded Workbench for different architectures
- Built in Simulator
- RTOS awareness plugins
- Support for C-SPY debug macros that can be used to automate debug sessions
- IAR Systems I-jet debug probe support (3rd party debug probe support will be added in future releases)
- Trace support



Debugging

Integrated debugger for source and disassembly debugging

- C like macro system
- Built-in simulator
- RTOS aware
- Trace



Dockable windows and tab groups

Stack usage

Watch

Complex breakpoints

Locals

Semihosted terminal I/O

Registers

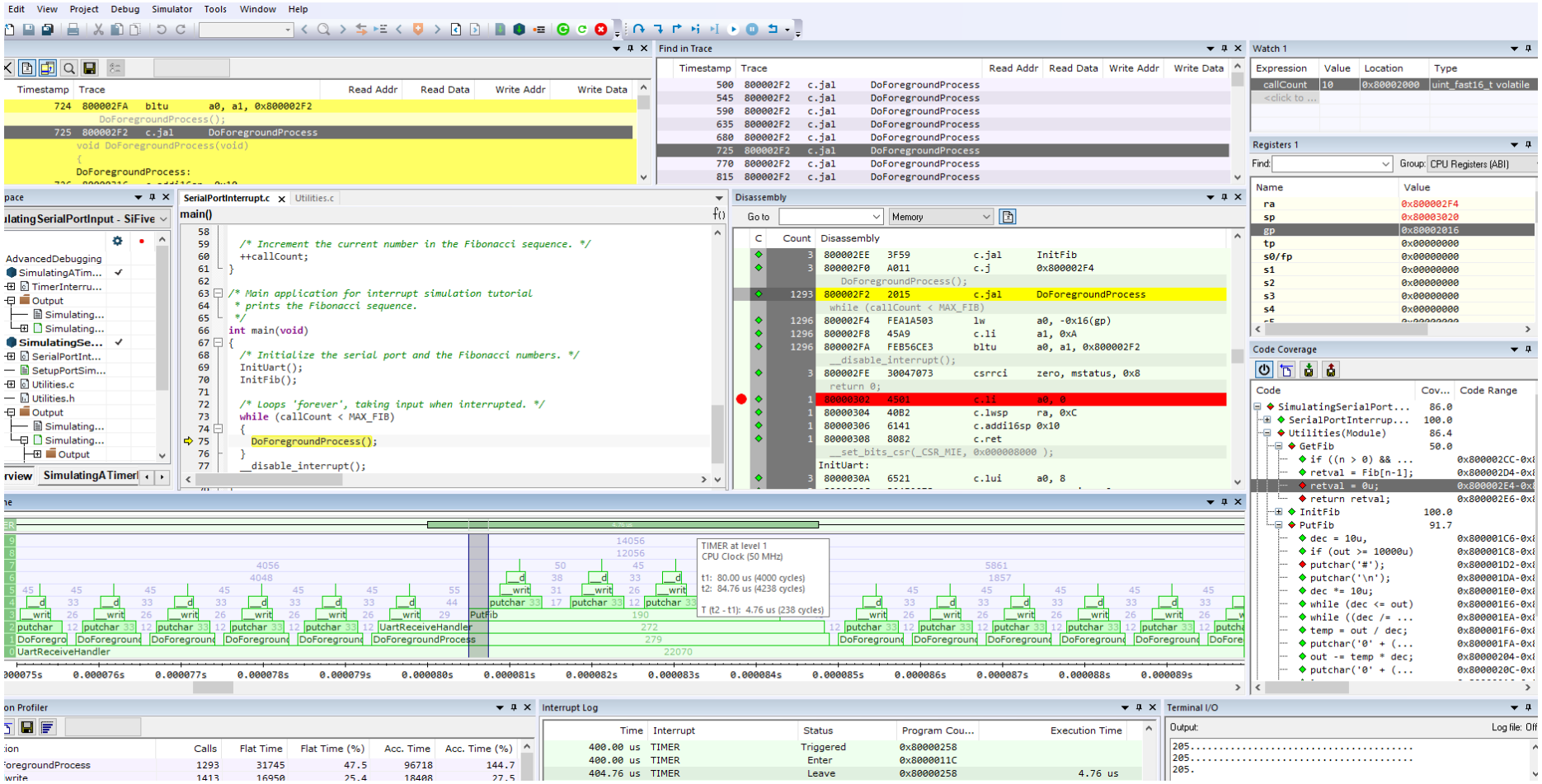
Debugging

IAR Systems I-jet debug probe

- Supports RISC-V and Arm cores
- Hi-speed USB 2.0 interface (480Mbps)
- Target power of up to 400mA can be supplied from I-jet with overload protection
- Target power consumption can be measured with ~200µA resolution at 200kHz
- JTAG and Serial Wire Debug (SWD) clocks up to 32MHz
  (no limit on the MCU clock speed)
- Support for SWO speeds of up to 60MHz
- Unlimited flash breakpoints (*to be added for RISC-V)

Trace

Advanced debugging and trace capabilities



*Nexus IEEE-ISTO 5001™ compatible trace

Trace

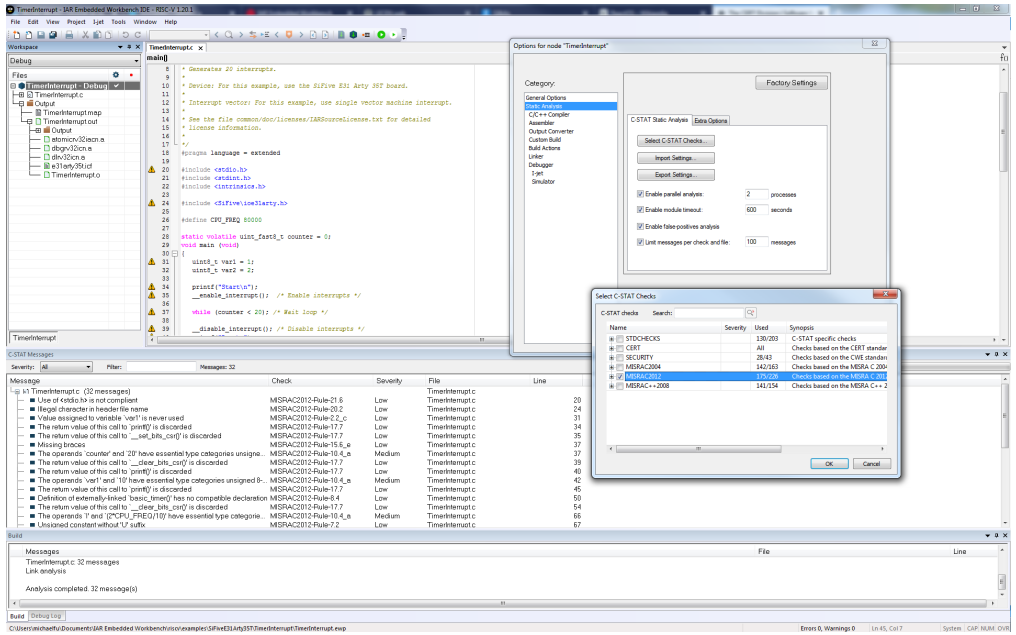
IAR Systems I-jet Trace debug probe

- Supports RISC-V and Arm cores
- SuperSpeed USB 3.0 interface (5 Gbps) Fully compatible with USB 2.0 (480 Mbps)
- JTAG and Serial Wire Debug (SWD) clocks up to 100MHz
  (no limit on the MCU clock speed)
- Up to 150MHz trace clock
- 64-bit time stamp with CPU cycle accuracy for timing analysis
- Support for SWO using Manchester and UART, at up to 200 Mbps

Code analysis

Code analysis

IAR Systems C-STAT static code analysis



- Complete static analysis tool fully integrated in IAR Embedded Workbench
- Intuitive and easy-to-use settings with flexible rule selection
- Support for export/import of selected checks
- Support for command line execution
- Extensive and detailed documentation
- Checks compliance with MISRA C:2004, MISRA C++:2008 and MISRA C:2012
- Includes ~250 checks mapping to hundreds of issues covered by CWE and CERT C/C++

http://cwe.mitre.org/ https://www.sei.cmu.edu/about/divisions/cert/index.cfm

Demo

Summary

- Advanced toolchain for RISC-V
- Sophisticated compiler technology
- Comprehensive C-SPY debugger including Simulator
- Built in static code analysis C-STAT
- More than an ordinary toolbox