Professional development tools for RISC-V

Ryan Sheng, IAR Systems (China)

Future-proof software tools and service for embedded development

- Leading embedded software development tools vendor
- Dedicated team of support, sales and service worldwide
- 32% of revenue invested in R&D

36 years in the industry
Listed on NASDAQ Stockholm

Uppsala	Shanghai
Munich	Dallas
Paris	Boston
Tokyo	Los Angeles
Seoul	San Francisco

Distributor representation in 40+ countries

2018
- Sales SEK 385.2M
- Operating profit SEK 115.6M
- Net profit SEK 87.6M

Large and loyal customer base worldwide

46,000+ customers

Industrial automation
Medical
Wearables
Consumer electronics
Automotive
Internet of Things

95% recurring customers

IAR Embedded Workbench

Complete C/C++ compiler and debugger toolchain

IAR SYSTEMS

Most widely used development tools for embedded applications

User-friendly IDE features and broad ecosystem integration

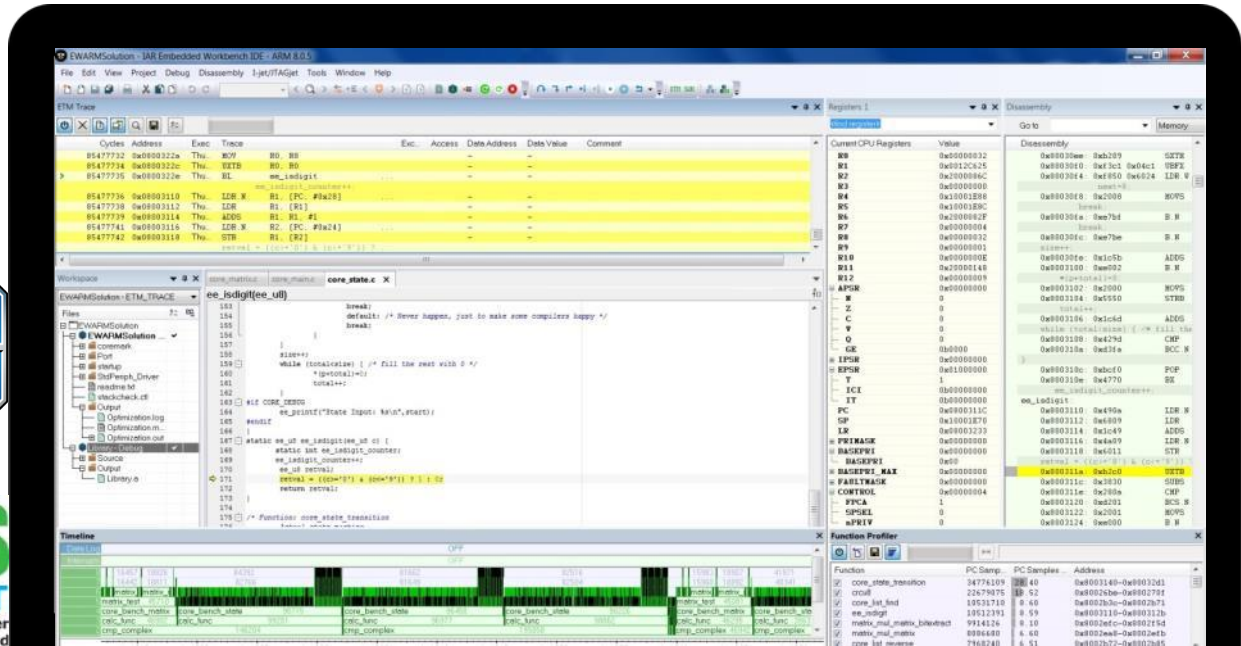
Industry leading code optimization technology

Comprehensive debugger

Integrated code analysis tools

ISO/ANSI C/C++ compliance with C18 and C++14

Global support and training service



Safety tested · Production monitored · TÜV SÜD · Functional Safety

CMSIS COMPLIANT
ARM® Cortex™ Microcontroller Software Interface Standard

Support for 12,000+ devices

40+ architectures

All available 8-,16- and 32-bit MCUs

Cortex-M0	Cortex-R8	AVR	H8
Cortex-M0+	Cortex-A5	AVR32	STM8
Cortex-M1	Cortex-A7	RX	ColdFire
Cortex-M3	Cortex-A8	RL78	HCS12
Cortex-M4	Cortex-A9	RH850	S08
Cortex-M7	Cortex-A15	78K	MAXQ
Cortex-M23	ARM11	SuperH	CR16C
Cortex-M33	ARM9	V850	SAM8
Cortex-R4	ARM7	R32C	RISC-V
Cortex-R5	SecurCore	M32C	
Cortex-R52	8051	M16C	
Cortex-R7	MSP430	R8C	

Partnership with SiFive

Company News

Date: December 3, 2018

IAR Systems and SiFive partner to meet customers' demands for professional solutions for RISC-V

Establish partnership for delivering increased possibilities for powerful RISC-V implementations

RISC-V Summit, Santa Clara, California—December 3, 2018—IAR Systems®, the future-proof supplier of software tools and services for embedded development, and SiFive, the leading provider of commercial RISC-V processor IP, announce that they have formed a partnership in order to deliver increased possibilities for powerful RISC-V implementations with compact code and high performance.

IAR Embedded Workbench for RISC-V

Milestones

IAR SYSTEMS

2016
Starts exploring RISC-V

Q3 2017
Small pre-dev on compiler for RISC-V

Q1 2018
Joins the RISC-V foundation

Q2 2018
Official start of the development project

Q1 2019
Beta edition for partners

Q2 2019
First release

TBD
Functional safety release

Device support

RV32I Base Integer Instruction Set

Supported Extensions:

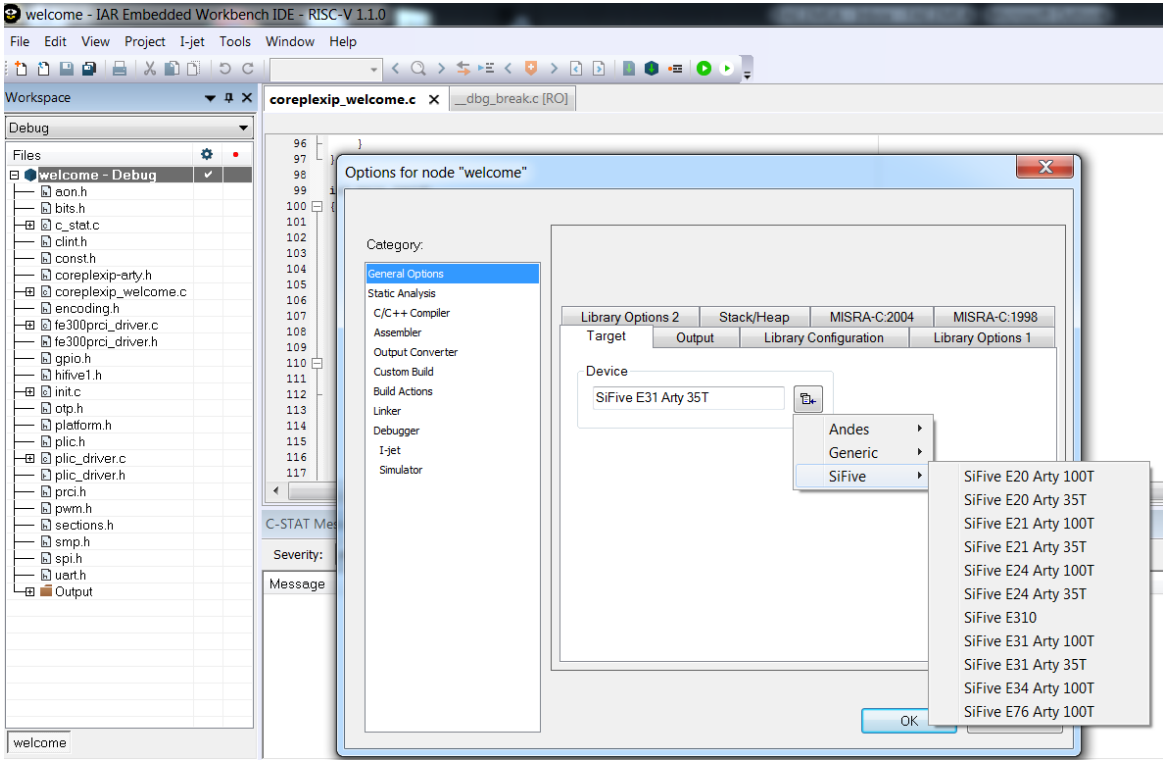
- M – integer mul & div
- F – single precision float
- D – double precision float
- C – compressed instruction

Support for SiFive 32-bit **E** Cores

E20 E21 E24
E31 E34 E76

Out-of-the-box experience on Xilinx Artix-7 Arty 35T/100T board

* More cores and devices will be added in later versions



Compiler

- Proprietary design based on over 36 years of experience
- Based on a platform that is common among different targets to handle global optimizations, language compliance, etc.
  - Different architecture, one solution
  - Easy source code migration among different architectures
- Target unique backend for specific adaptations and optimizations
- RISC-V specifics
  - Primary focus will be supporting standard extensions
  - Initial prioritization on optimizations is code size

Challenges on optimization

- **Size**
  - Compared to more complex instruction sets, RISC-V have some challenges especially when it comes to code size
  - Arithmetic with higher resolution than the natural data size yields larger code
  - Absence of carry flags and instructions to save and restore multiple registers are other examples
- **Speed**
  - When it comes to speed, RISC-V are competitive

Our initial target will be on reduced code size for small embedded systems. Our main focus have always been to supply the best code size and speed on the market.

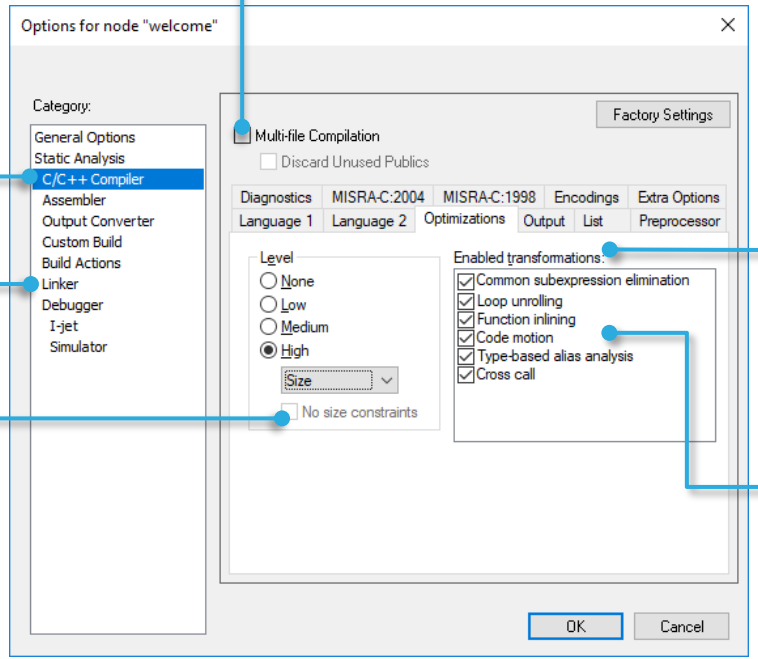
IAR C/C++ Compiler

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

Multiple optimization levels for code size and execution speed

The linker can remove unused code

Option to maximize speed with no size constraints



Language standards

- ISO/IEC 14882:2015 (C++14, C++17)
- ISO/IEC 9899:2018 (C18)
- ANSI X3.159-1989 (C89)

- IEEE 754 standard for floating-point arithmetic

Major features of the optimizer can be controlled individually

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

Well-tested

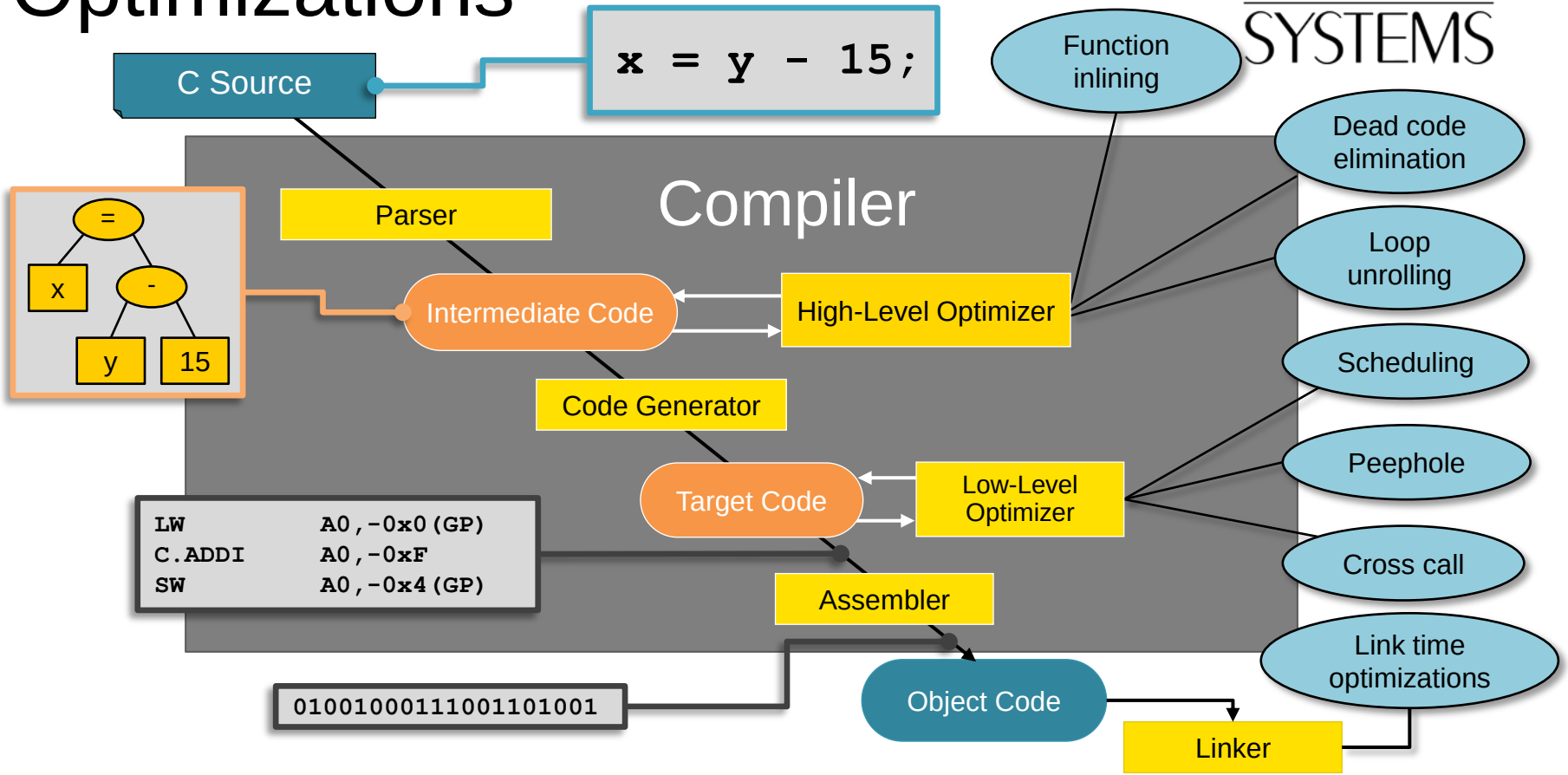
Commercial test suites

- 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

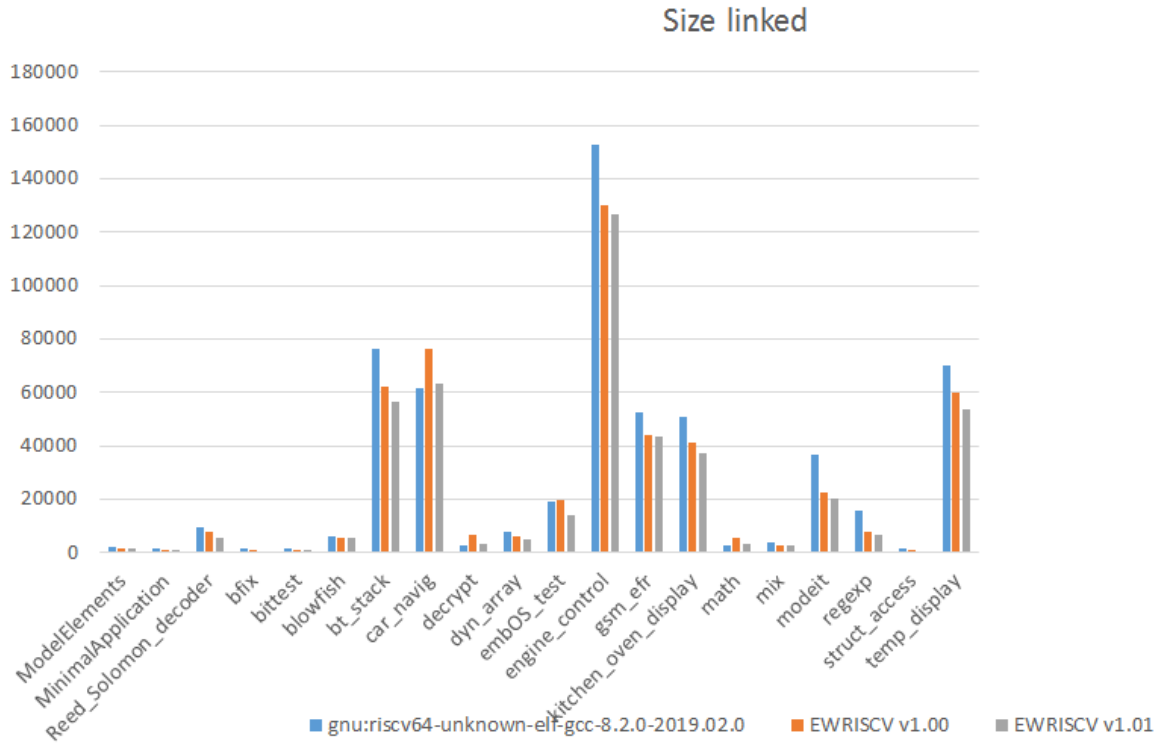
Optimizations



Speed, size or both?

Optimization	Effect	
Common sub-expressions	Speed ↑	Size ↓
Loop unrolling	Speed ↑	Size ↑
Function inlining	Speed ↑	Size ↑
Code motion	Speed ↑	Size →
Dead code elimination	Speed →	Size ↓
Static clustering	Speed ↑	Size ↓
Instruction scheduling	Speed ↑	Size →
Peephole	Speed ↑	Size ↓
Cross call	Speed ↓	Size ↓

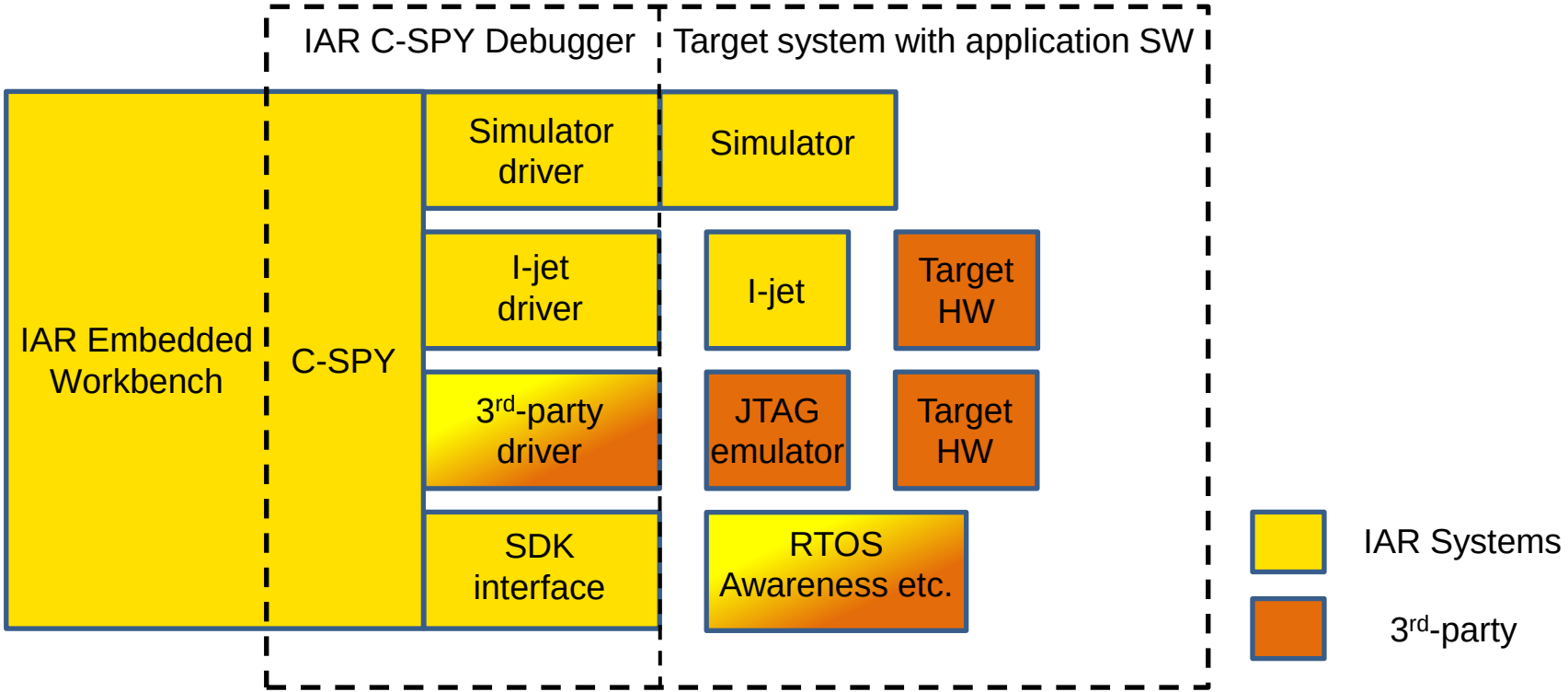
Current code size benchmark



icc options: --core=RV32IMAC -e –Ohz
--dlib_config=full
gcc options: -Os -D NDEBUG -march=rv32imac
-mabi=ilp32 -mcmodel=medlow
--specs=nano.specs

Internal benchmark shows ~20% smaller code in average !!!

C-SPY debugger overview



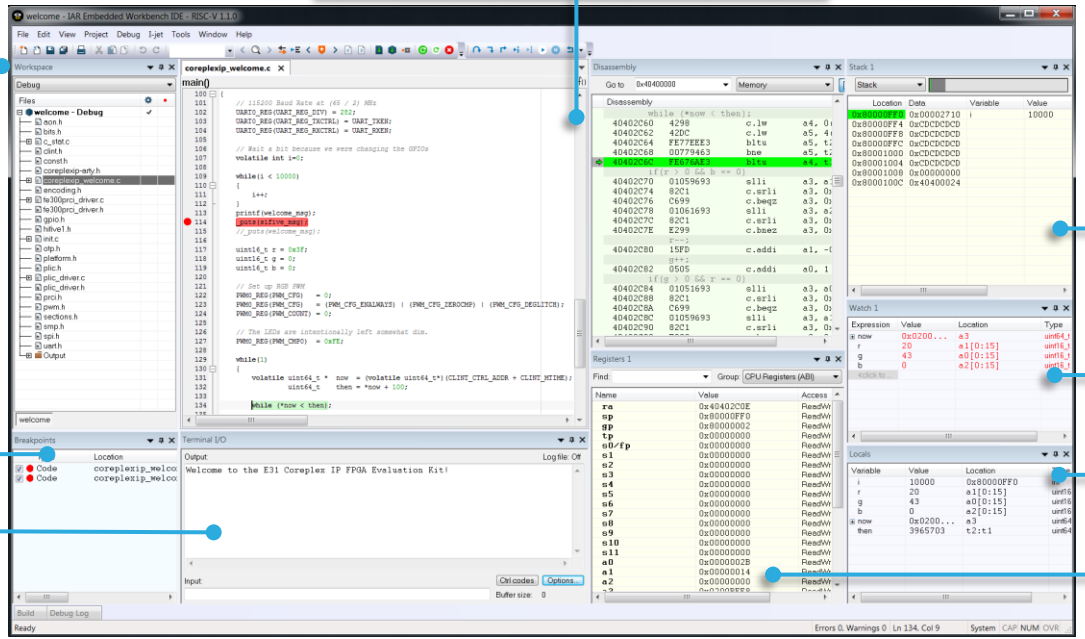
C-SPY debugger

Integrated debugger for source and disassembly debugging

Dockable windows and tab groups

Complex breakpoints

Semihosted terminal I/O



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

Stack usage

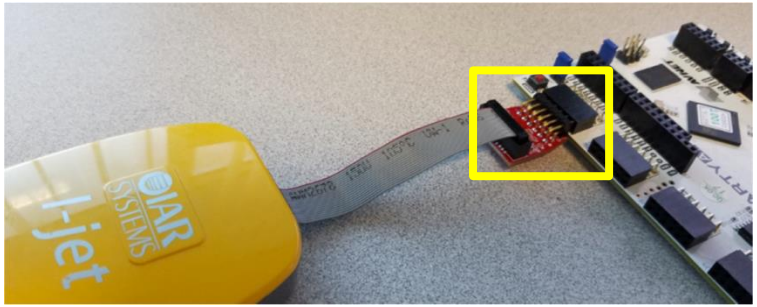
Watch

Locals

Registers

I-jet in-circuit debugging probe

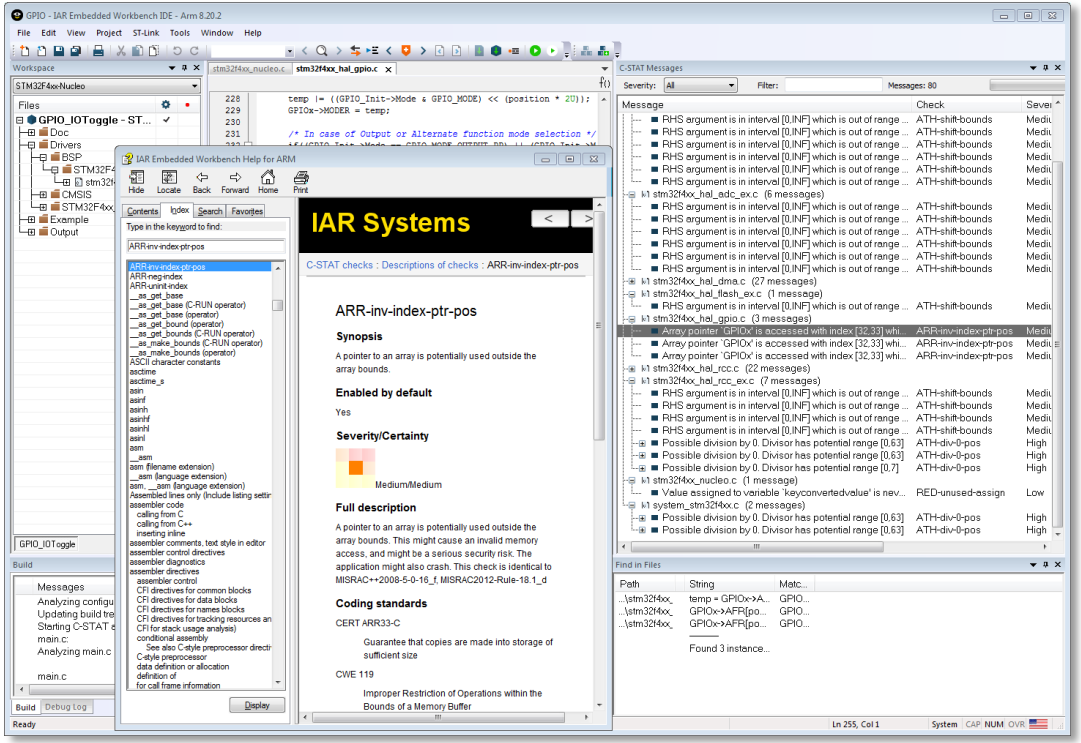
- Support RISC-V and Arm cores
- Hi-speed USB 2.0 interface (480Mbps)
- Target power of up to 400mA can be supplied from I-jet with overload protection
- Target power consumption can be measured with ~200µA resolution at 200kHz
- JTAG and Serial Wire Debug (SWD) clocks up to 32MHz
- No limitation on the CPU clock speed
- Debug adapter (ADA-MIPI20-RISCV12)



C-STAT: static code analysis

- Advanced analysis of C/C++ code
- Fully integrated within IAR Embedded Workbench
- Check compliance with **MISRA C:2004**, **MISRA C++:2008** and **MISRA C:2012**
- Include ~250 checks mapping to hundreds of issues covered by **CWE** and **CERT C/C++**
- Intuitive and easy-to-use settings with flexible rule selection
- Support for command line execution
- Extensive and detailed documentation

CWE (Common Weakness Enumeration): http://cwe.mitre.org
CERT (Computer Emergency Response Team): http://www.cert.org

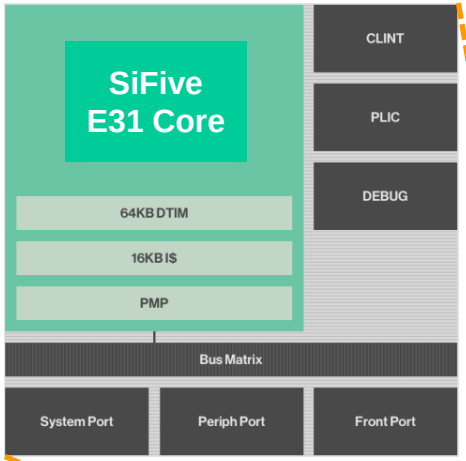


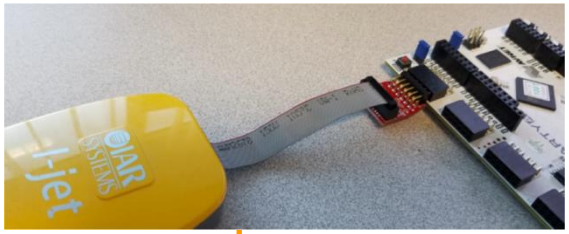
High level timeplan

Future releases

- Improved optimizations for code size and speed
- Atomics
- RV32E
- RV64I
- C RISC-V ABI compliance
- C-RUN (runtime code analysis)
- Trace (depending on spec maturity)
- Functional safety certification
- 3rd-party debug probes

Demonstration



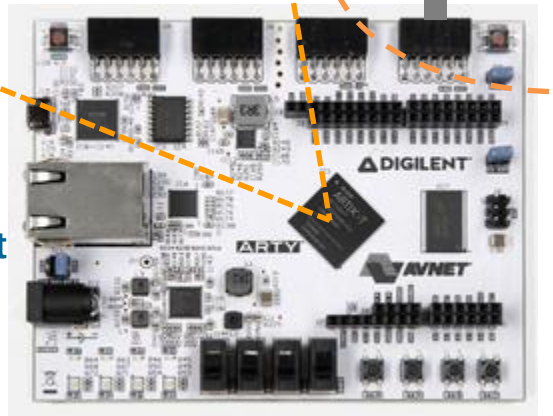


JTAG

I-jet

USB

编译/链接/下载/调试



Xilinx Artix-7
Arty FPGA Kit


IAR
SYSTEMS
Thanks for your attention!
www.iar.com/riscv