SiFive 2 Series RISC-V Core IP

Drew Barbier - Sr. Product Marketing Manager

Date May, 2019

About This Presentation

This presentation will introduce the SiFive 2 Series RISC-V Core IP including architecture, configurability, and feature set.

At the end of this presentation you should have a basic understanding of SiFive's 2 Series RISC-V Core IP and know where to go for more information.

3-Part Webinar Series

An Introduction to the RISC-V Architecture

SiFive's 2 Series RISC-V Core IP

From a Custom 2 Series Core to Hello World in 30 Minutes

https://info.sifive.com/risc-v-second-webinar-series

How To Ask Questions

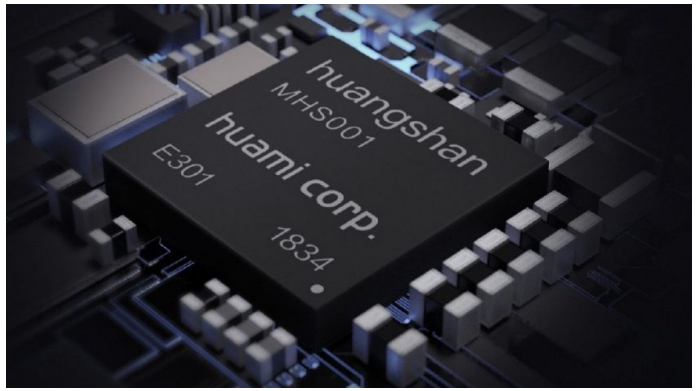
SiFive

SiFive Core IP Overview

RISC-V®

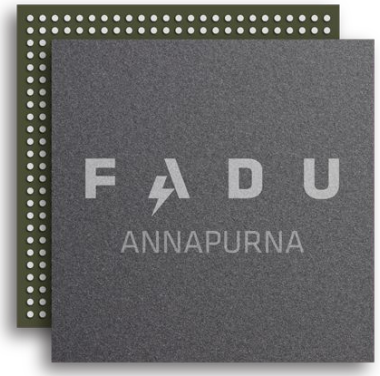
E Cores

32-bit Embedded Processors



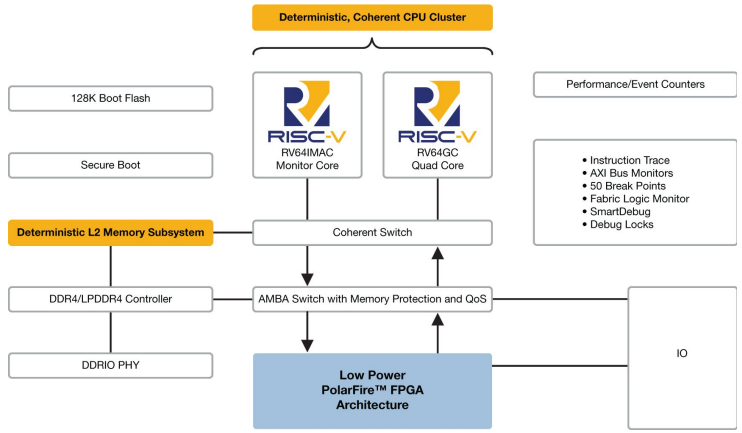
S Cores

64-bit Embedded Processors



U Cores

64-bit Application Processors



"SiFive's RISC-V Core IP was **1/3 the power** and **1/3 the area** of competing solutions, and gave FADU the flexibility we needed in optimizing our architecture to achieve these groundbreaking products."

-J. Lee, FADU CEO

"SiFive's **64-bit S Cores** bring their hallmark efficiency, configurability and **silicon-proven** Core IP **expertise** to 64-bit embedded architectures"

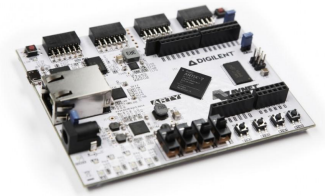
-Ted Speers, Head of Product Architecture and Planning, Microsemi, a Microchip Company

Product Map

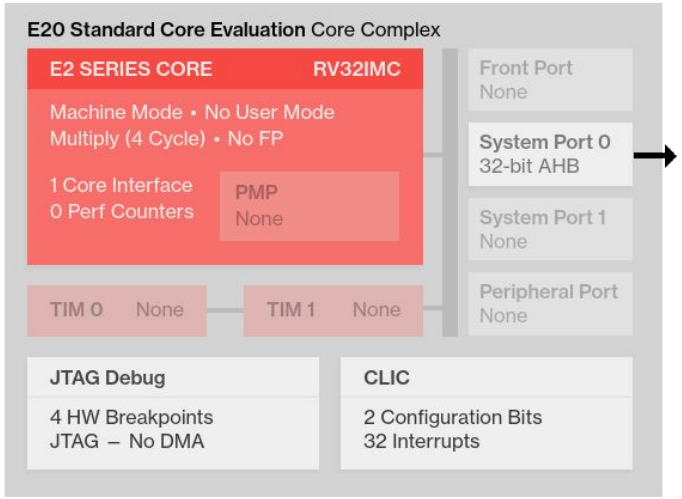
	E Cores	**S** Cores	**U** Cores
	32-bit embedded cores MCU, edge computing, AI, IoT	**64-bit embedded cores** Storage, AR/VR, machine learning	**64-bit application cores** Linux, datacenter, network baseband
7 Series **Highest performance:** 8-stage, dual-issue superscalar pipeline	**E7 Series**	**S7 Series**	**U7 Series**
	E76-MC Compare to Cortex-R8 Quad-core 32-bit embedded processor	**S76-MC** Compare to Coretex-R8 Quad-core 64-bit embedded processor	**U74-MC** Compare to Cortex-A55 Multicore: four U74 cores and one S76 core
	E76 Compare to Cortex-M7 High performance 32-bit embedded core	**S76** Compare to Cortex-R8 High-performance 64-bit embedded core	**U74** Compare to Cortex-A55 High performance LInux-capable processor
3/5 Series **Efficient performance:** 5–6-stage, single-issue pipeline	**E3 Series**	**S5 Series**	**U5 Series**
	E34 Compare to Cortex-R5F E31 features + single-precision floating point	**S54** Compare to Cortex-R5F S51 features + double-precision floating point	**U54-MC** Compare to Cortex-A53 Multicore application processor with four U54 cores and one S76 core
	E31 Compare to Cortex-R5 Balanced performance and efficiency	**S51** Compare to Cortex-R5 Low-power 64-bit MCU core	**U54** Compare to Cortex-A53 Linux-capable application processor
2 Series **Power & area optimized:** 2–3-stage, single-issue pipeline	**E2 Series**	**S2 Series**	
	E24 Compare to Cortex-M4F E21 + single-precision floating point	**S21** No 64-bit Cortex equivalent Area-efficient 64-bit MCU core	
	E21 Compare to Cortex-M4 E20 + User Mode, Atomics, Multiply, TIM		
	E20 Compare to Cortex-M0+ Our smallest, most efficient core		

Standard Cores - Pre-Configured Implementations of a Core Series

E2 Series



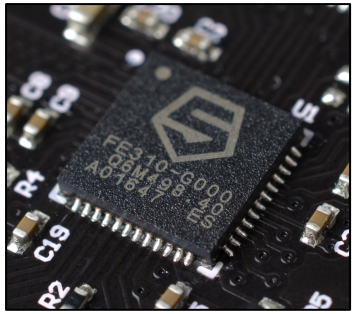
FPGA Evaluations



E20 Standard Core Definition

Benchmarks

RTL Evaluations



Silicon

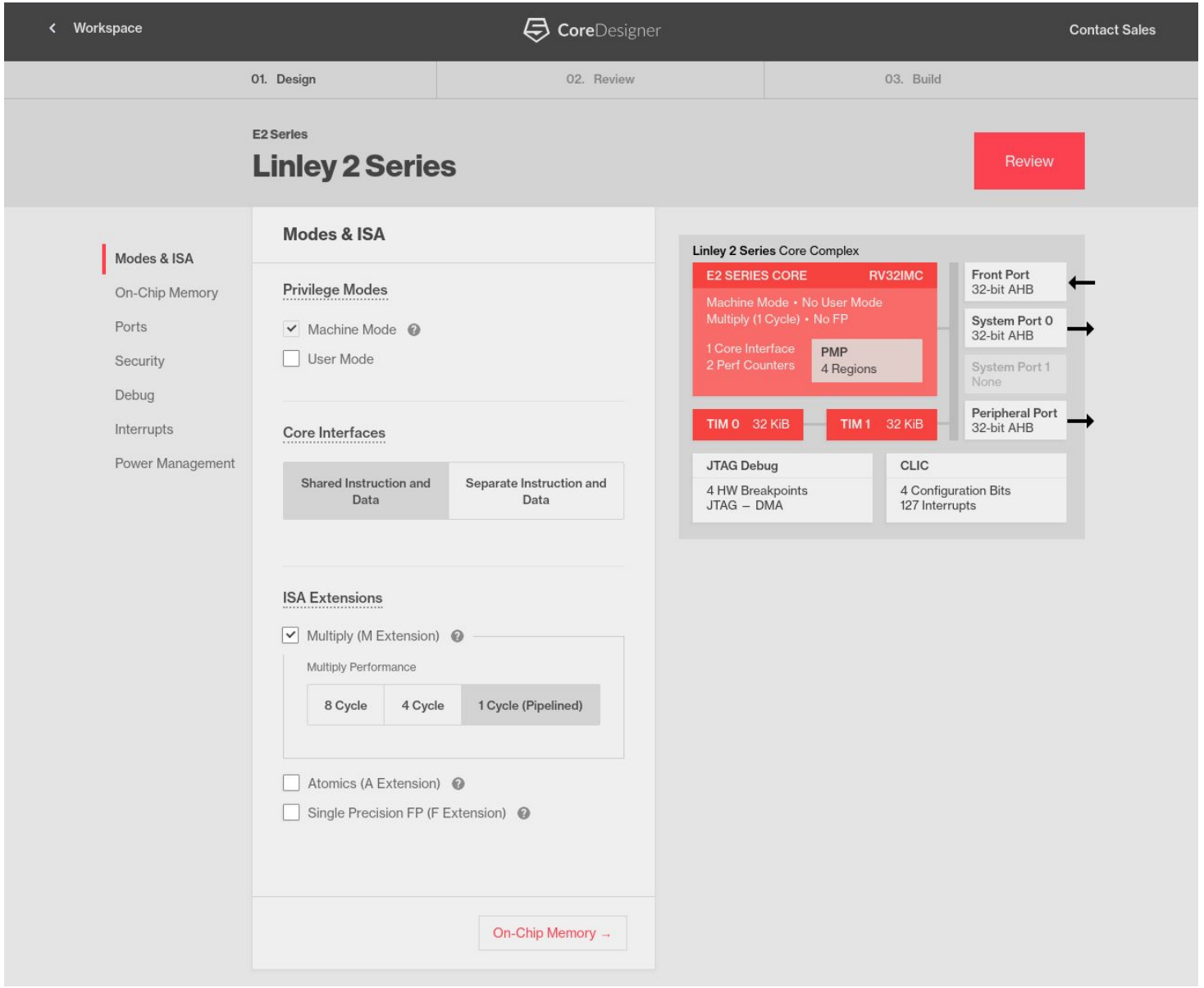
25+ Design Wins Since Launching in June 2018

And Everything In-Between

SiFive Core Designer - Your Interface to SiFive Core IP

- **Web-Based Core Configuration Tool**
  - No on-site tools to install
  - No licenses servers to manage
  - No complex scripting languages to learn

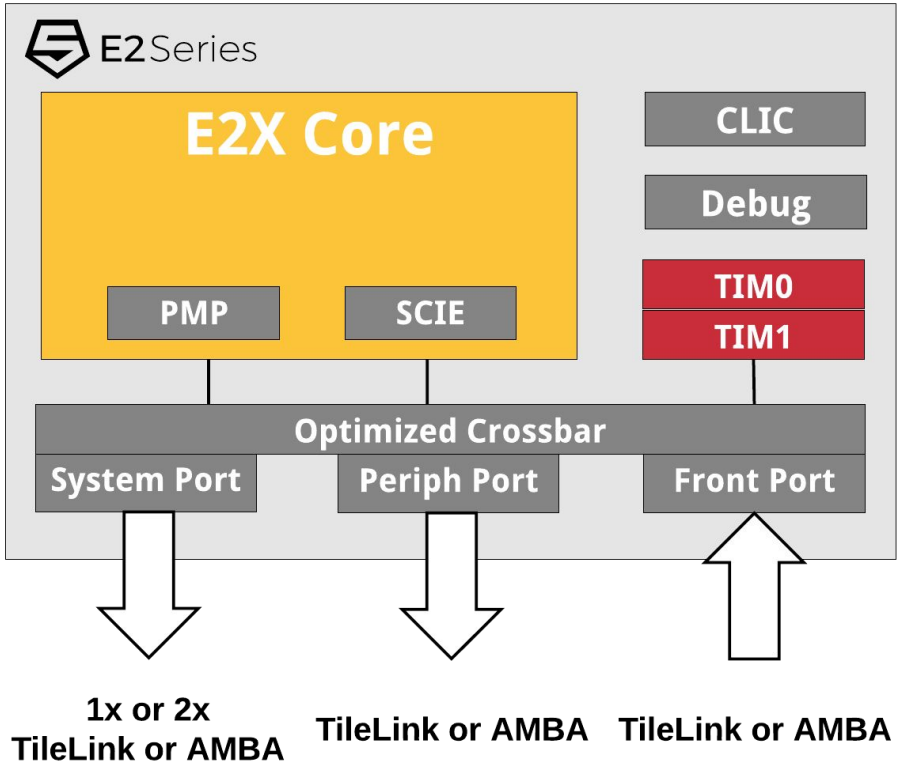
- **Configure, Download, Use**
  - Intuitive interface allows for easy configuration
  - Configured designs are placed into your workspace for download
  - Synthesize the RTL, run the testbench, execute your software on the FPGA bitstream



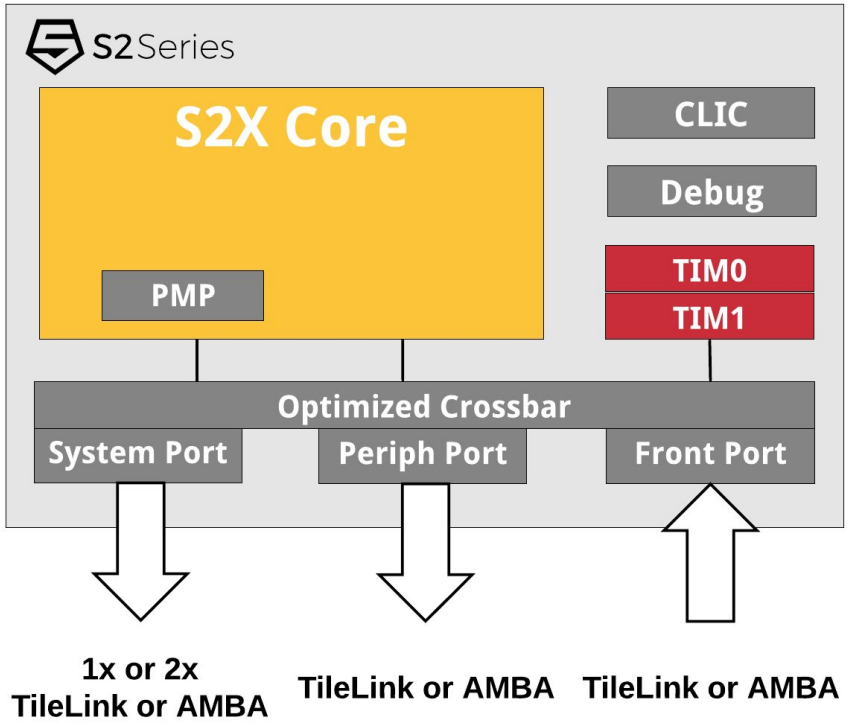
E2 Series Features

The Smallest, Most Efficient RISC-V MCU Family

- **E2 Series core architectural overview**
  - RV32(E)IMAFC capable core
  - 2-3 stage, optional, Harvard Pipeline
- **Efficient memory accesses**
  - Ability to add multiple outbound Ports
  - Optional Tightly Integrated Memory (TIM) and Optional Instruction Cache
- **First RISC-V core with support for the RISC-V Core Local Interrupt Controller (CLIC)**
  - Provides hardware interrupt prioritization and nesting
  - Only 6 cycles to execute the first instruction of IRQ
- **SiFive Custom Instruction Extension (SCIE)**
  - Easily add support for custom instructions



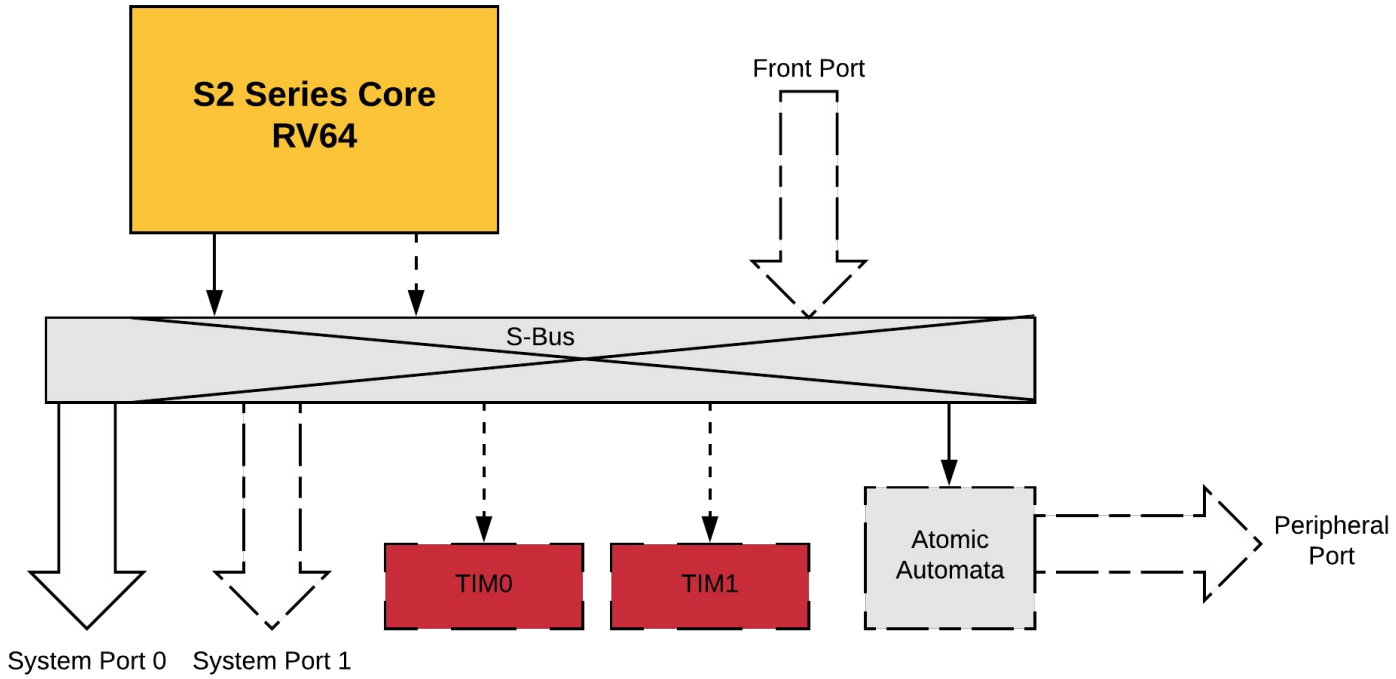
S2 Series - Extending the 2 Series to 64-bit



- **Very small 64-bit MCU**
- **Same familiar pipeline and feature set as the the E2 series but now 64-bit**
  - 2-3 stage, optional, Harvard Pipeline
  - 64-bit arithmetic operations
  - Double the Load/Store bandwidth
- **Easier integration into larger SoCs**
  - S2 Series can directly address >32-bit physical address space
- **No impact on code size thanks to RISC-V Compressed instructions**
  - The RISC-V RV64IC ISA uses the same 16-bit and 32-bit instructions as the RV32IC ISA

No Competitive Equivalents in the Market Today

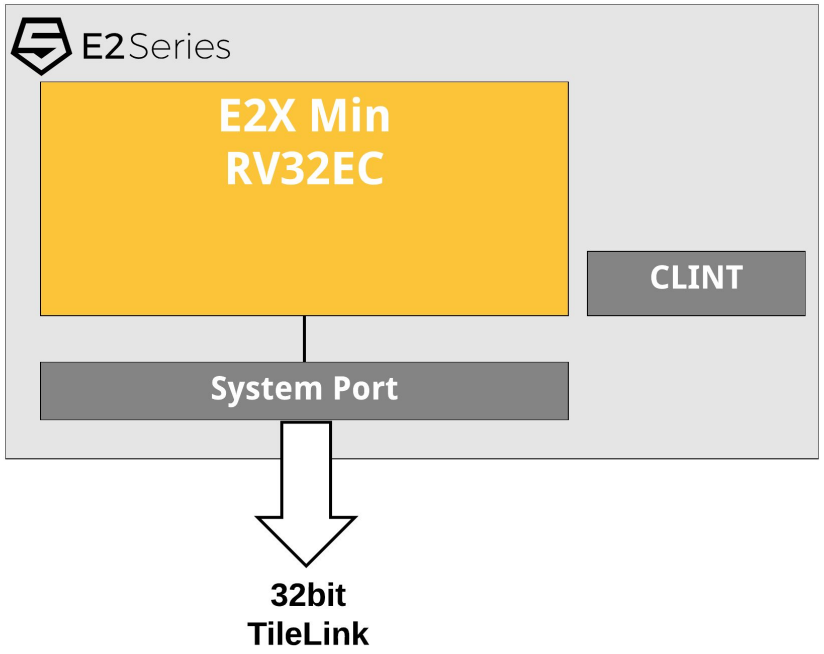
2 Series Memory Subsystem



- **2 Core can be configured with 1 or 2 native interfaces**
  - Choose between performance optimized vs area optimized configurations
  - S2 has native 64-bit interfaces for double the Load/Store bandwidth vs the E2
- **Configurable Ports**
  - Choose architecture, data width, and address map of all ports
  - >32-bit addressability with the S2
- **Optional 2x low-latency TIM banks**
  - Supports parallel access to both TIM banks for 2 cores with 2 bus interface configurations
- **Optional Peripheral Port**
  - Support for RISC-V Atomic instructions which can be used for single-cycle Read-Modify-Write operations

E2M - SiFive's Smallest E2 Core

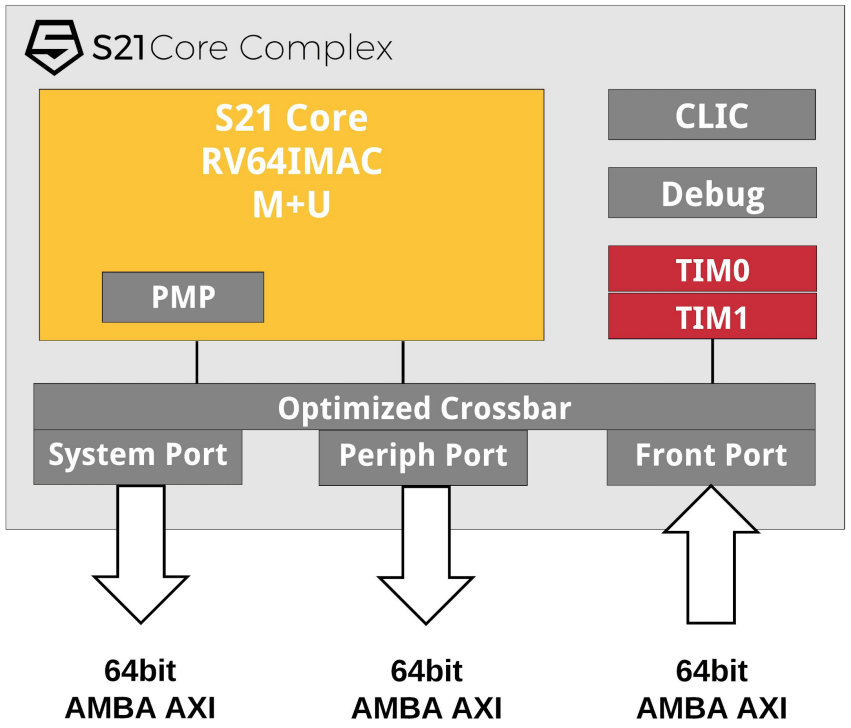
- **E2M Configuration**
  - **RV32EC** Base ISA
    - Reduced integer register file for area optimized implementations
    - Fully supported with SiFive's software development tools
  - Area optimized **interrupt controller** with 16 peripheral interrupts
    - Supports RISC-V Software, Timer, External interrupts in addition to the 16 local peripheral interrupts
  - Single 32-bit TileLink Port
- **Extremely Small Area**
  - 13.5k Gates Core Area; .005mm2 in 28nm
- **Without Sacrificing Performance**
  - 1.07 DMIPS/MHz



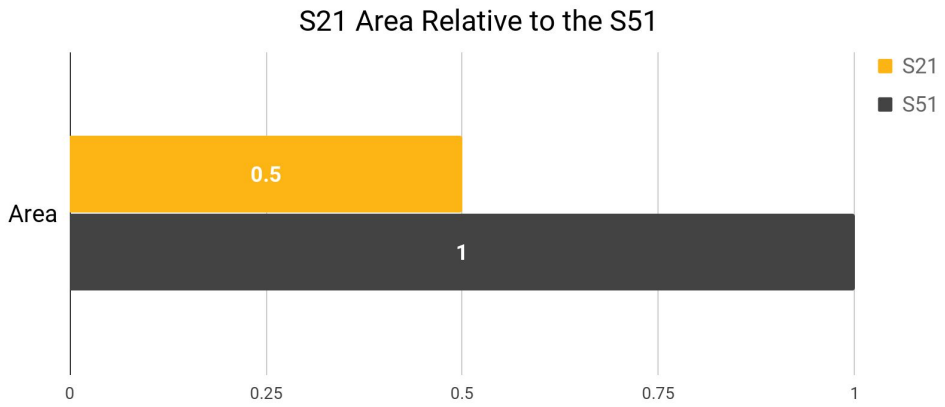
E2 Min Post-Route Physical Design	
	TSMC 28HPC
Frequency target	50MHz
Worst Setup Corner	ssg_0p81v_m40c_cworst
Implementation Details	9t; LVT, SVT, UHVT
Core Complex Area (mm^2)*	0.008
Core Only Area (mm^2)**	0.005

Note: All area and power numbers do not include RAMs
*Core Complex includes the Core plus CLINTinternal bus and ports
**Core only includes the core pipeline only

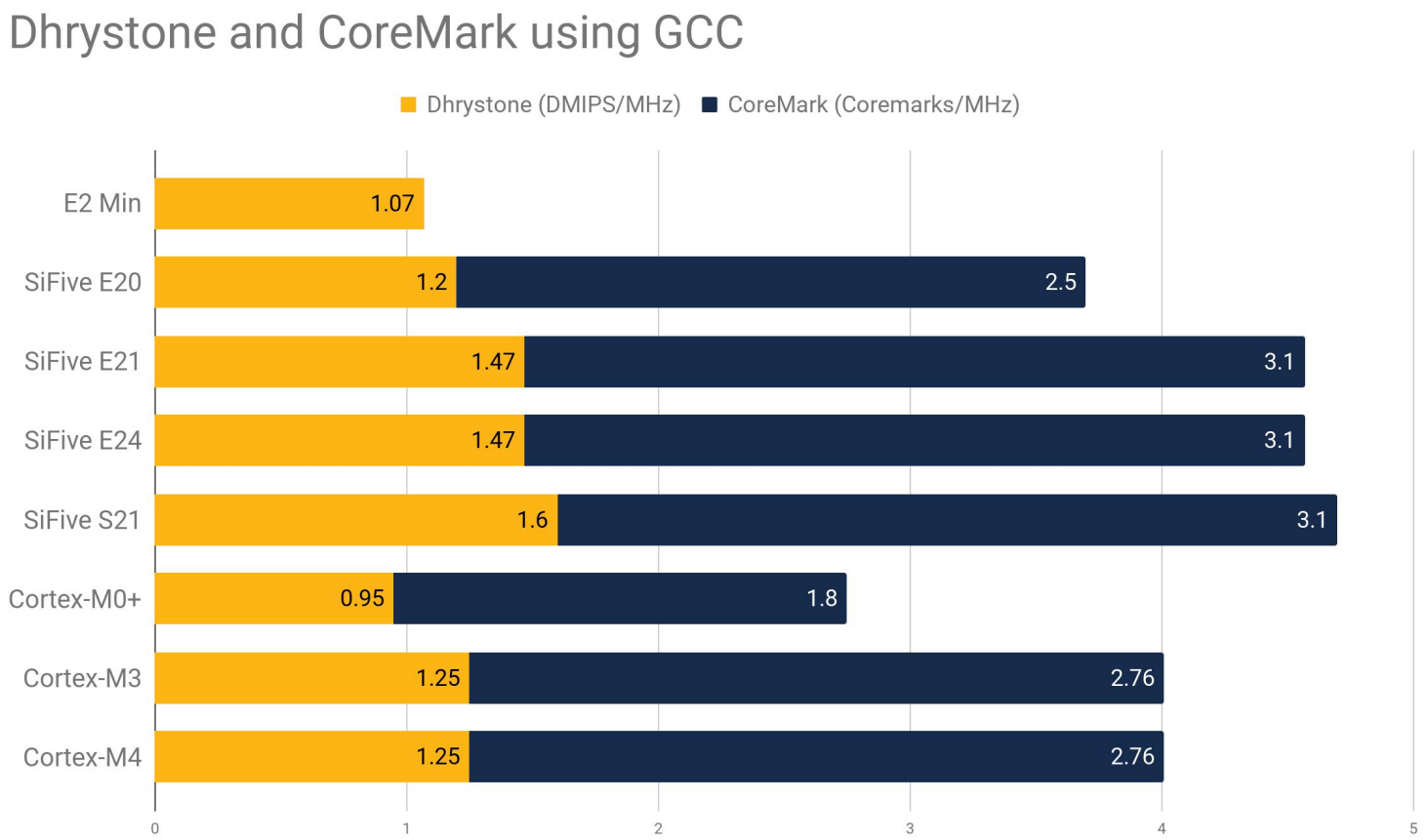
S21 Standard Core - Tiny, Full Featured, 64-bit MCU



- **S21 is a balanced implementation of the S2 Series**
  - RV64IMAC with Harvard Architecture
  - 2x TIM banks
  - 64-bit AXI Ports
- **Half the area of the S51**
  - .035mm2 in TSMC 28nm
- **S21 Performance**
  - 1.6 DMIPS/MHz
  - 3.2 Coremarks/MHz
  - 850MHz+ at TSMC 28nm at TT 85C corner



2 Series Performance Improvements



February release of the 2 Series has improved hardware Multiply and Divide

E2 Series VS ARM Cortex-M

The E2 Series can be configured to meet your application requirements

E2 Series VS ARM Cortex-M Comparison Table						
	E2 Series Options	E20 Standard Core	E21 Standard Core	Cortex-M0+	Cortex-M3	Cortex-M4
Dhrystone (using GCC)	From 1.07 to 1.47 DMIPS/MHz	1.2 DMIPS/MHz	1.47 DMIPS/MHz	0.95 DMIPS/MHz	1.25 DMIPS/MHz	1.25DMIPS/MHz
CoreMark (using GCC)	Up to 3.1	2.5 CoreMarks/MHz	3.1 CoreMarks/MHz	1.8 CoreMarks/MHz	2.76 Coremarks/MHz	2.76 CoreMarks/MHz
Integer Registers	31 Useable, 16 Useable Option	31 Useable	31 Useable	13 Useable	13 Useable	13 Useable
FPU	Optional FPU	None	None	None	None	Optional
Hardware Multiply and Divide	Optional, with configurable performance	Yes	Yes	Slow and Fast Option	Yes, always	Yes, always
Memory Map	Customizable	SiFive Freedom Platform	SiFive Freedom Platform	Fixed ARMv6-M	Fixed ARMv7-M	Fixed ARMv7-M
Atomics	Optional: RISC-V standard AMO support via Peripheral Port and TIMs	No Peripheral Port	RISC-V AMO standard support via Peripheral Port	None	Bit-band and Load/Store Exclusive	Bit-band and Load/Store Exclusive
Number of Interrupts	Up to 1008 peripheral interrupts	32	127	32	240	240
Interrupt Latency into C Handler	6 Cycles – CLIC Vectored Mode	6 Cycles	6 Cycles	15 Cycles	12 Cycles	12 Cycles
Memory Protection	Optional up to 16 Regions	N/A	4 Regions	Optional, ARMv6m	0 or 8 Region	0 or 8 Region
Tightly Integrated Memory	Optional 2 Banks	None	2 Banks	No	No	No
Bus Interfaces	Configurable: Up to 3 masters and 1 slave with support for TileLink, AXI, AHB-Lite, APB	1 Master	2 Master, 1 Slave	1 AHB-Lite	3 AHB-Lite	3 AHB-Lite

SiFive

2 Series Feature Details

RV32E - Reduced Integer Register File

- **2 Series now supports the RISC-V Embedded ISA; RV32E**
  - Reduces integer register file from 32 to 16 general purpose registers
  - Can be combined with existing supported extensions: M, A, F, C
- **Up to a 25% area reduction depending on the core configuration**
- **Minimal impact on benchmarks**
  - Only a 6% impact on Dhrystone for the E20
- **Fully supported in SiFive tools and software**
  - RV32E supported in SiFive's latest toolchain distribution
  - RV32E support added to Freedom E SDK and Freedom Metal
  - No changes necessary in C code

Register	ABI Name	Description	Saver
x0	zero	Hard-wired zero	-
x1	ra	Return address	Caller
x2	sp	Stack pointer	Callee
x3	gp	Global pointer	-
x4	tp	Thread pointer	-
x5-7	t0-2	Temporaries	Caller
x8	s0	Saved register	Callee
x9	s1	Saved register	Callee
x10-11	a0-1	Function Arguments/return values	Caller
x12-15	a2-5	Function arguments	Caller

RV32E ABI

Optional M and F Extensions

- **Optional "M" extension with configurable performance**
  - Pipelined multiplier for efficient performance
  - 4 and 8 cycle multiplier options for area sensitive applications
  - Option to remove the multiplier for extreme area constrained applications

- **Adding the "M" extension to the 2 Series includes an area efficient hardware Divide**
  - 1 bit per cycle divide with early out

- **Optional "F" extension adds fully pipelined single-precision FPU**
  - IEEE 754-2008 compliant FPU
  - Features: Fused-Multiply-Add, iterative divide and sqrt, magnitude comparators, float to integer converters, subnormal support

- **D extension is coming soon to the 2 Series**

Choose the Right Interrupt Controller for Your Application

Area Optimized (CLINT)

- Simple controller allowing for up to 16 or 48 peripheral interrupts depending on XLEN
- Fixed priority scheme
- Support for software handler and limited vectoring capability

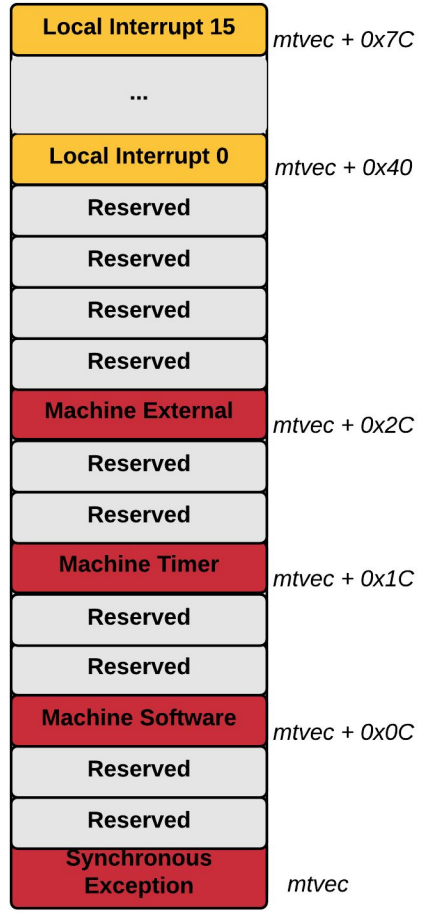
Feature Optimized (CLIC)

- Featurefull controller allowing for up to 1008 peripheral interrupts
- Hardware prioritization, nesting, and vectoring
- Configurable number of priority bits

CLINT Controller

Use the CLINT when optimizing for area over interrupt latency and features

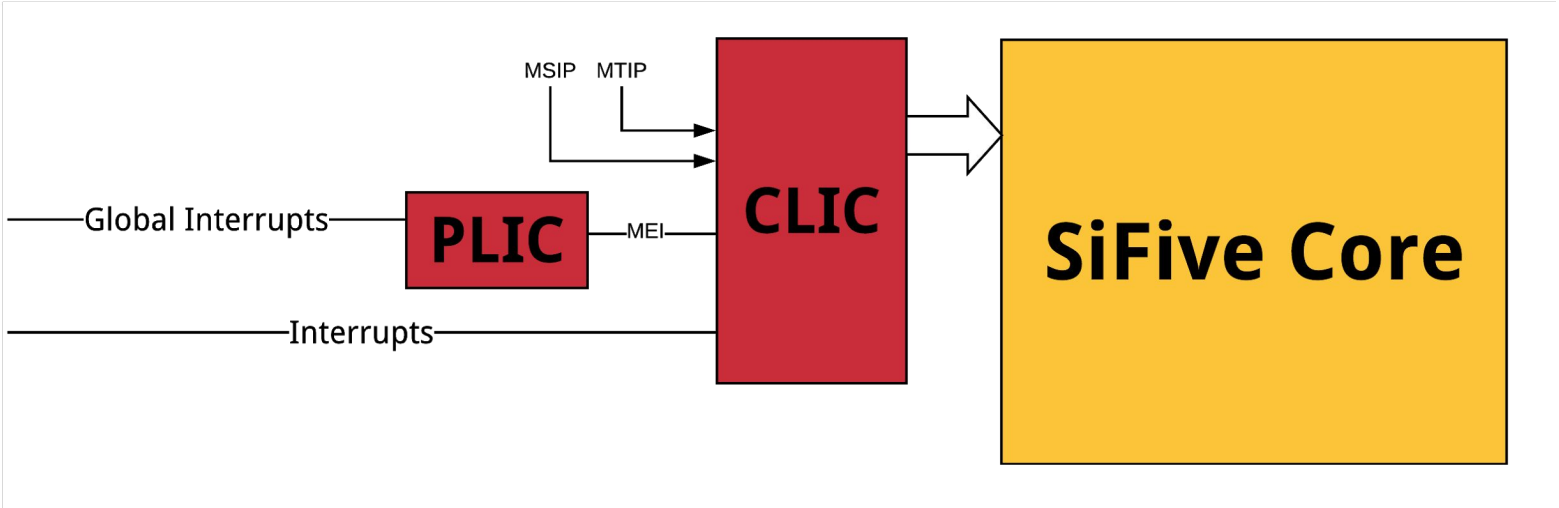
- **E2 supports 16 CLINT interrupts, the S2 supports 48 CLINT interrupts**
  - In addition to the architecturally defined Software, Timer, and External interrupts
- **CLINT Interrupts have a fixed priority scheme**
  - The higher the Interrupt ID, the higher the Priority
- **Optional Interrupt Vectoring**
  - Vector table contains jump instructions directly to C Code handlers
- **Highest number interrupt is highest priority and special due to its position in the vector table**
  - No additional jump necessary
  - Handler code can begin executing immediately and does not need to jump to a handler



Vector Table w/ 16 Local Interrupts
mtvec + (4*Exception Code)

Core Local Interrupt Controller (CLIC)

- **Simplified interrupt scheme which allows for low latency interrupt servicing, hardware prioritization, and pre-emption**
  - Support for up to **1008 peripheral interrupts**
  - Global programmable prioritization of all interrupts including software and timer
  - PLIC for global interrupts shared with other cores in the Core Complex
- **Extreme low latency**
  - **6 cycles** into first instruction of ISR, **18 cycles** total to complete a simple ISR in a 2 Series Pipeline
  - Vector table contains function pointers (addresses) to ISR
- **Interrupt pre-emption capabilities**
  - Up to **256 levels** of nesting, and programmable priorities within each level
- **Easy to use programmers model**
  - GCC interrupt function attribute, no assembly necessary
  - Multiple SW interrupts with programmable priority levels
  - CLIC driver and ISR included in SiFive software deliverables



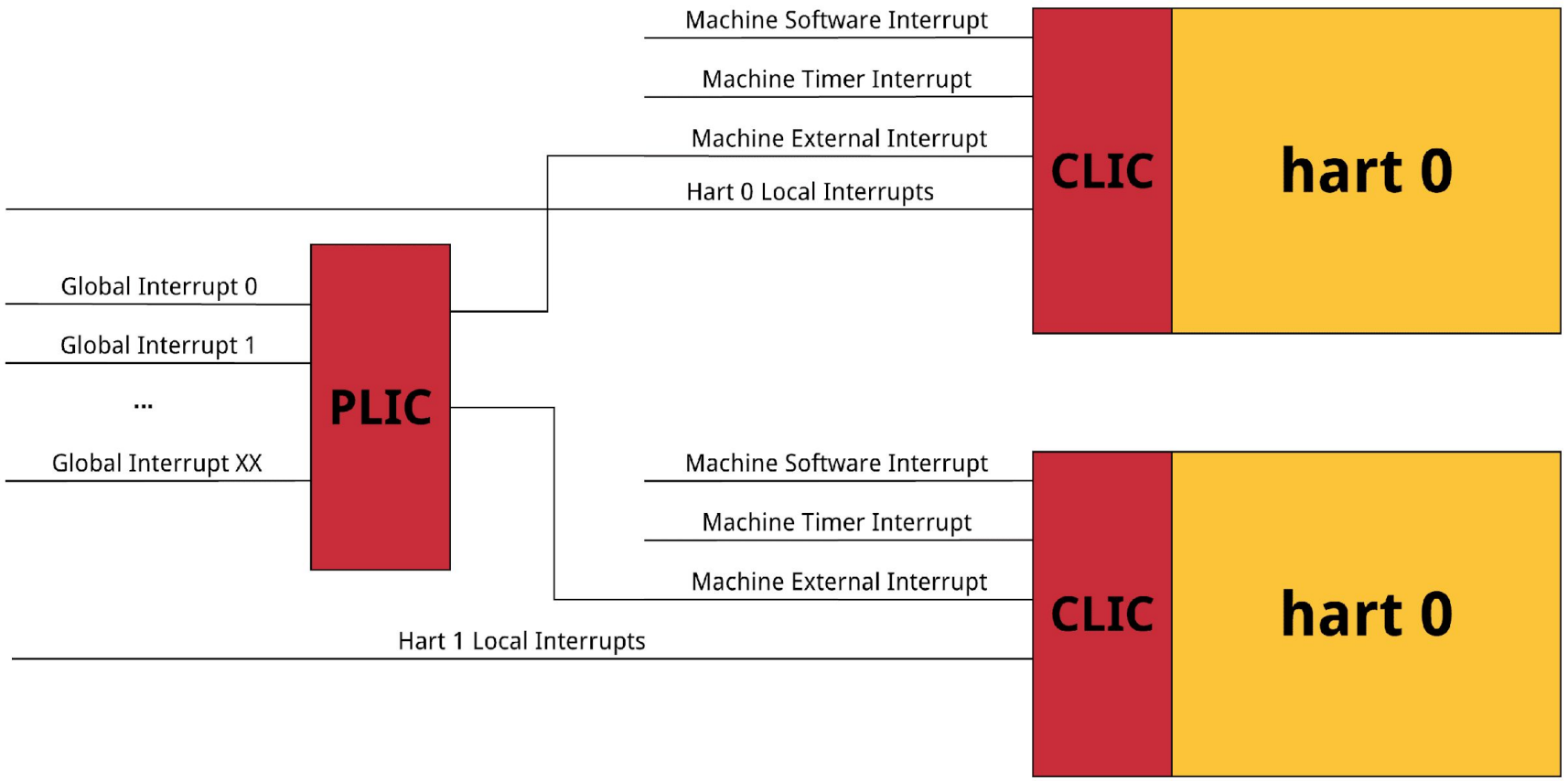
Preemptible Compiler Interrupt Attribute

- **Extends the *interrupt* function attribute to support CLIC Vectored Mode preemptible interrupts**
  - Saves *mcause* and *mepc* and re-enables interrupts
- **Handlers using this attribute will be preempted by interrupts of higher levels**

Interrupt handler with *interrupt("SiFive-CLIC-preemptible")* attribute

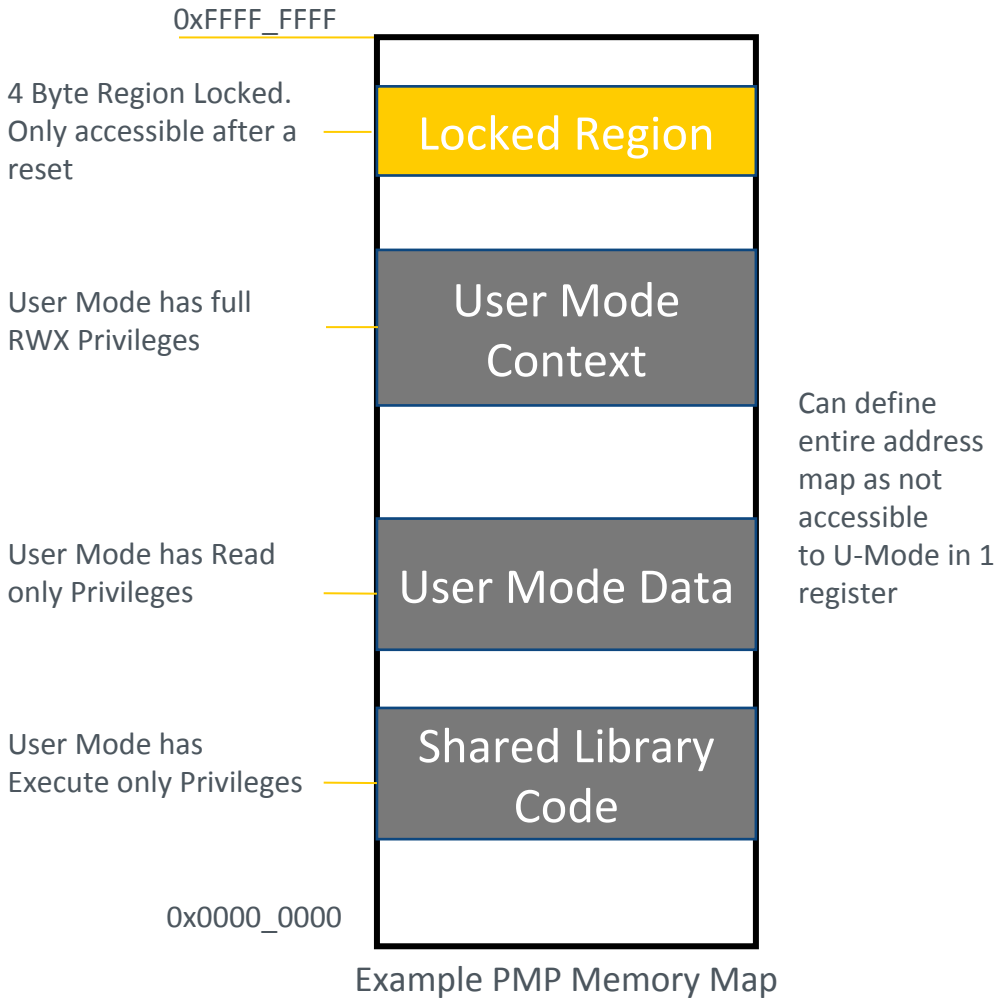
```
void mti_handler(void)
__attribute((interrupt("SiFive-CLIC-preemptible")));
void mti_handler()
{
    //C Code Handler
 }

//install machine timer handler into the vector table
vect_table[7] = mti_handler;

//write Base handler address to mtvec and set CLIC Vectored Mode
write_csr(mtvec, ((unsigned long)&handle_trap | CLIC_VECTORED));
//point mtvt to the vector table
write_csr(mtvt, vect_table));
```

RISC-V Interrupt System Architecture (M-mode only example)



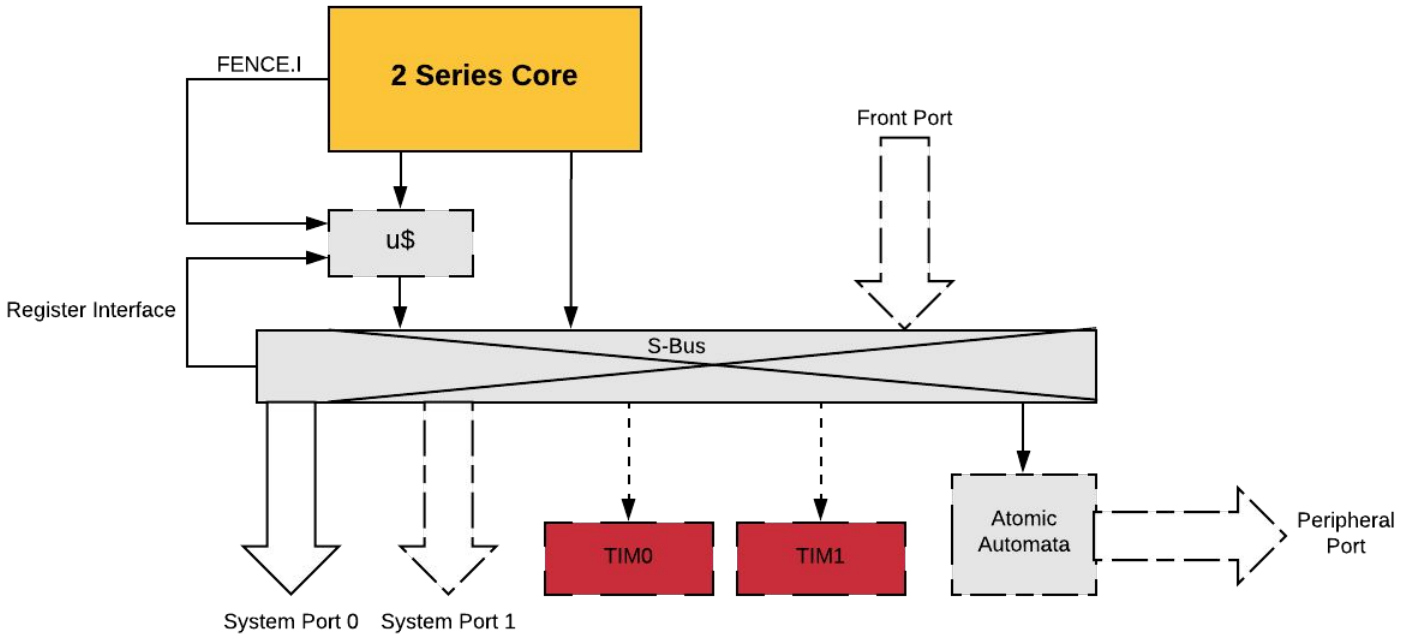
Physical Memory Protection (PMP)

- **Can be used to enforce access restrictions on less privileged modes**
  - Prevent Supervisor and User Mode software from accessing unwanted memory
- **Up to 16 regions with a minimum region size of 4 bytes**
- **Ability to Lock a region**
  - A locked region enforces permissions on all accesses, including M-Mode
  - Only way to unlock a region is a Reset



Example PMP Memory Map

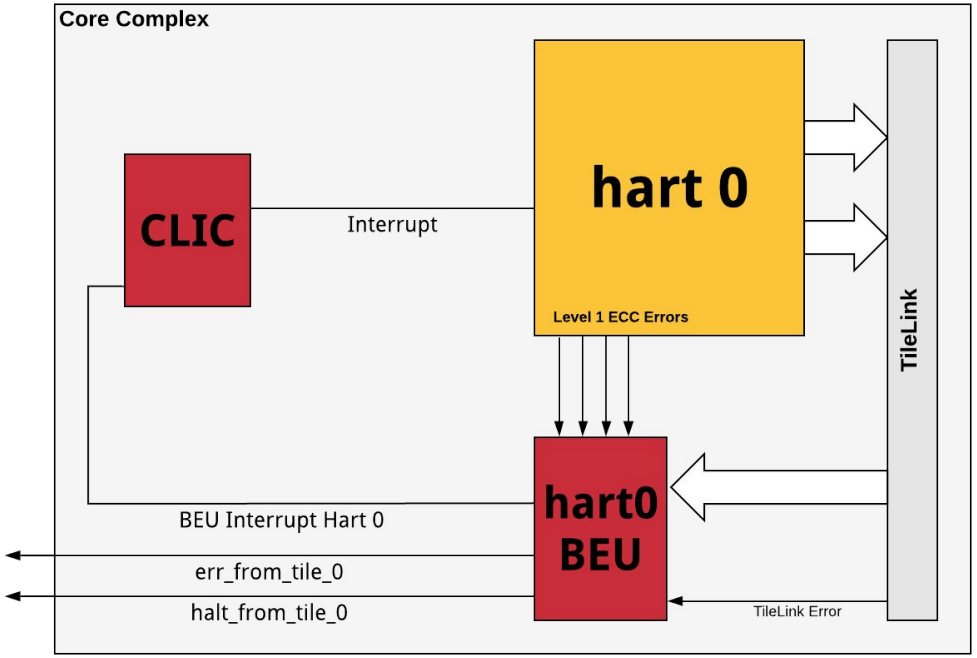
2 Series µInstruction Cache Option



- **The SiFive 2 Series µI-Cache enables Fast and Efficient accesses to slow memories**
  - Flushable with RISC-V FENCE.I instruction
- **Cache Details**
  - Cache size – 1kB to 16kB
  - Configurable cache line size
- **Configurable Cacheable Address Range**
  - Software defined “cacheable region”. Programmed using NAPOT base and sizes similar to PMP region registers.
  - Cache enable/disable register

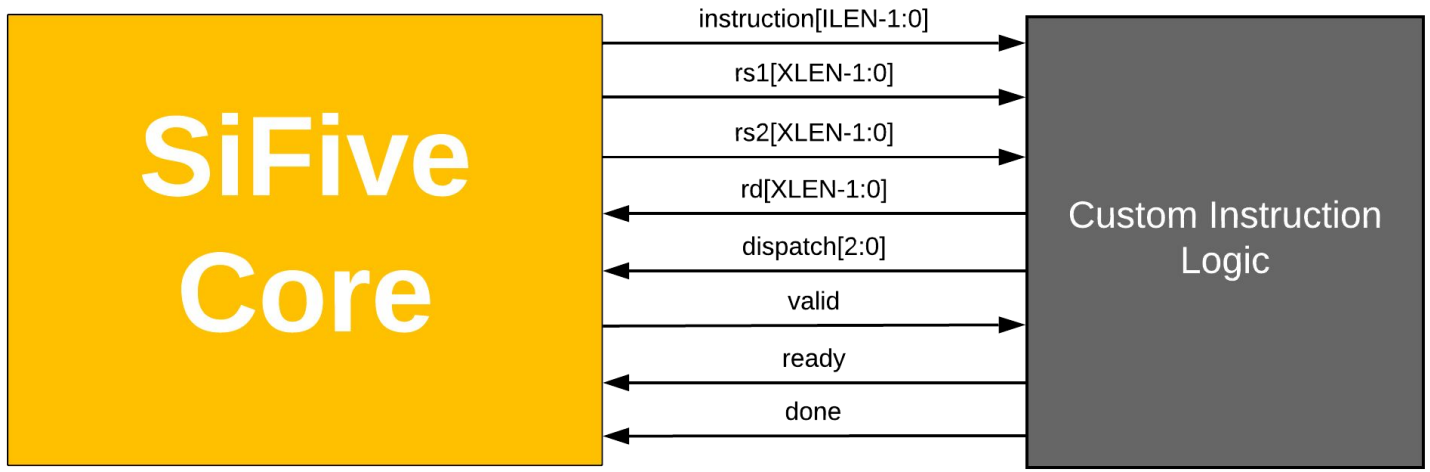
ECC and the Bus Error Unit

- **All SiFive cores have optional ECC capabilities**
  - Single Error Correct, Double Error Detect (SECDED)
  - Supported on 2 Series TIMs
- **Bus Error Unit can be used to record and report ECC errors**
  - Per-processor unit
  - Tracks and reports ECC events as well as bus errors
  - Can optionally generate interrupts on events
- **Single Bit Errors**
  - I$ - the error is corrected and the cache line is flushed
  - Single bit errors are written back to the RAM
  - Toggles the *err_from_tile_X* signal for system level tracking
- **Double bit errors**
  - Double bit errors are reported at the Core Complex boundary via the signal: *halt_from_tile_X*
  - *halt_from_tile_X* remains high until reset



SiFive Custom Instruction Extension (SCIE)

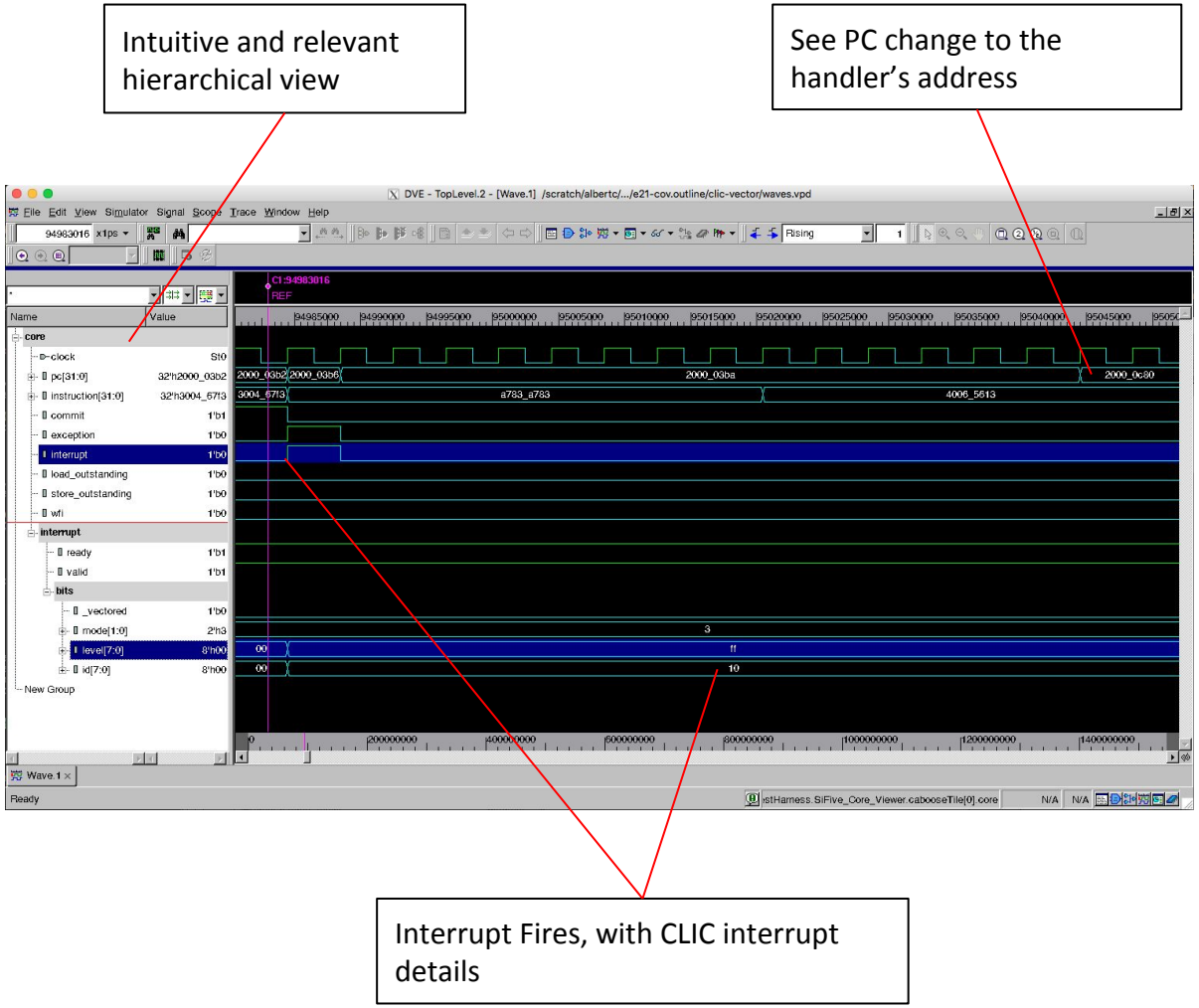
- **The SCIE creates a Verilog black box for your custom instruction**
  - A simple pre-defined interface and is the same across the entire SiFive product portfolio
  - Custom Instructions can be created in Verilog with an existing EDA flow
- **The SCIE allows for operations on the Integer register file**
  - *rs1* and *rs2* are decoded by the core and provided over the SCIE, no register copies necessary
  - *rd* is used to pass destination register data back to the core
- **Tightly coupled to the core**
  - Core pipeline handles all hazards
- **Flexible custom instruction support**
  - Support for 1, 2 cycle instructions



SiFive Insight - The Fast and Easy Way to See Inside SiFive Core IP

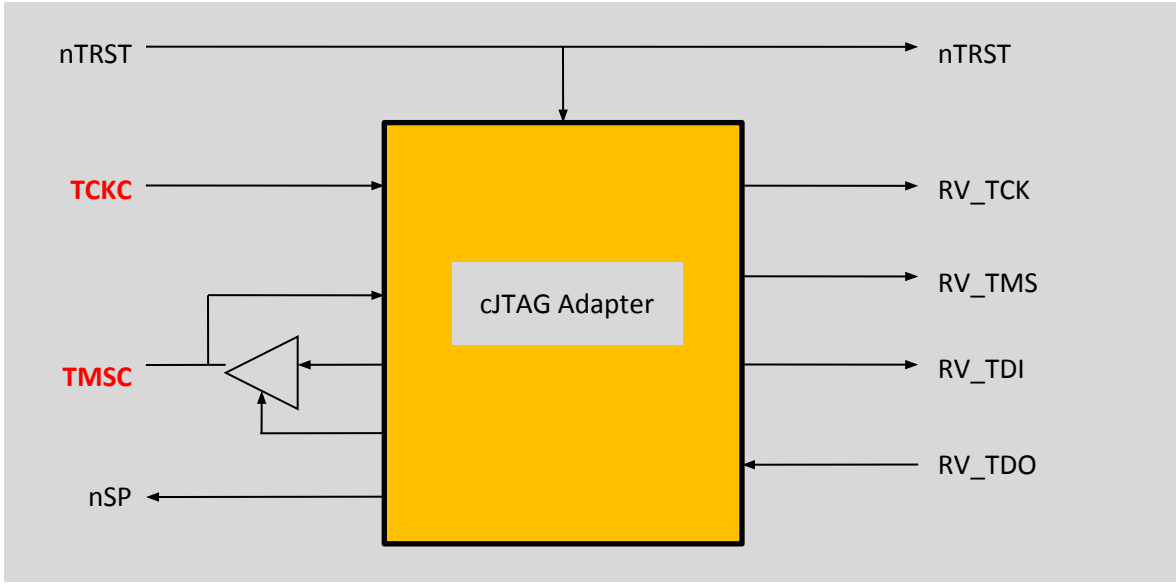
- **All important SiFive IP signals exposed in a single Verilog module**
  - Logical hierarchy allows for easily discovering signals of interest
  - Clear and intuitive signal names
  - Signals and hierarchy picked and used by the IP designers

- **All signals have an English language description**
  - Documented in each release
  - A yaml file is also included in the delivery allowing for easy integration with 3rd party tools



cJTAG Support: Available Now

- **Optional 2-wire IEEE 1149.7 (cJTAG) interface**
- **Classic JTAG run control debug with only 2-Wires**
  - Perfect for pin constrained designs
- **Fully supported by SEGGER JLINK and Olimex probes**



cJTAG System Block Diagram

SiFive

2 Series Software development

SiFive Embedded Software Ecosystem

- **SiFive Freedom Studio**
  - Eclipse CDT, GNU MCU Eclipse, pre-built GCC, and OpenOCD
  - Built on Open Source technology
- **SEGGER -** JLINK Probe and Embedded Studio RISC-V IDE
- **Lauterbach -** Lauterbach TRACE32 for silicon bring up and debug
- **IAR -** IAR Embedded Workbench with SiFive support in development
- **Ashling -** RiscFree C/C++ IDE for development and debug
- **Embedded Operating Systems**
  - Express Logic – Thread X
  - FreeRTOS
  - Micrium - µCOS
  - NuttX
  - RIOT
  - RTEMS
  - Zephyr OS
- **Imperas -** Simulation models and tools for early software development
- **UltraSoC -** IP and tooling supporting SiFive instruction trace

Open-Source Embedded Tools and Libraries by SiFive

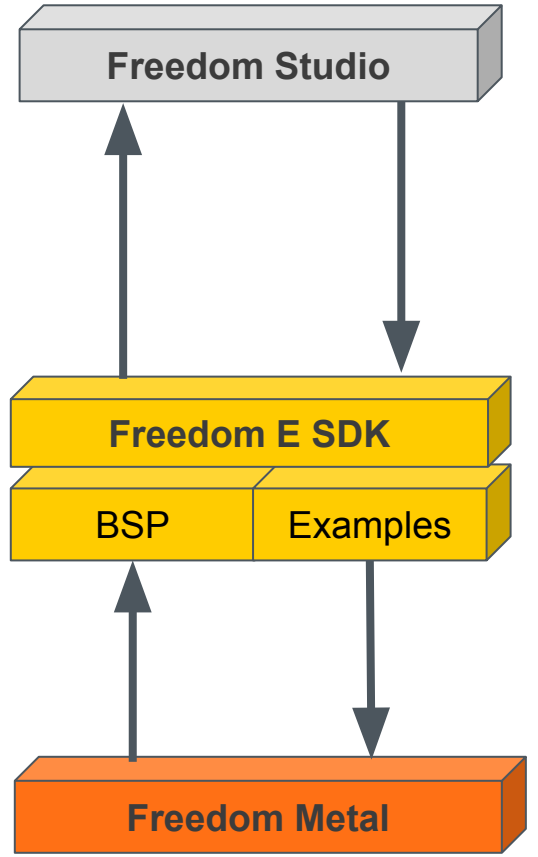
Freedom Studio provides an Eclipse based GUI for developing and debugging Freedom E SDK applications

Freedom E SDK provides a command line driven workflow with Examples and Utilities including targets for:

- Standard Core IP Deliverables
- Standard Core FPGA Deliverables
- SiFive Development Boards
- SiFive Core Designer Deliverables

Freedom Metal is a library for writing Portable, Bare Metal, software for all SiFive devices

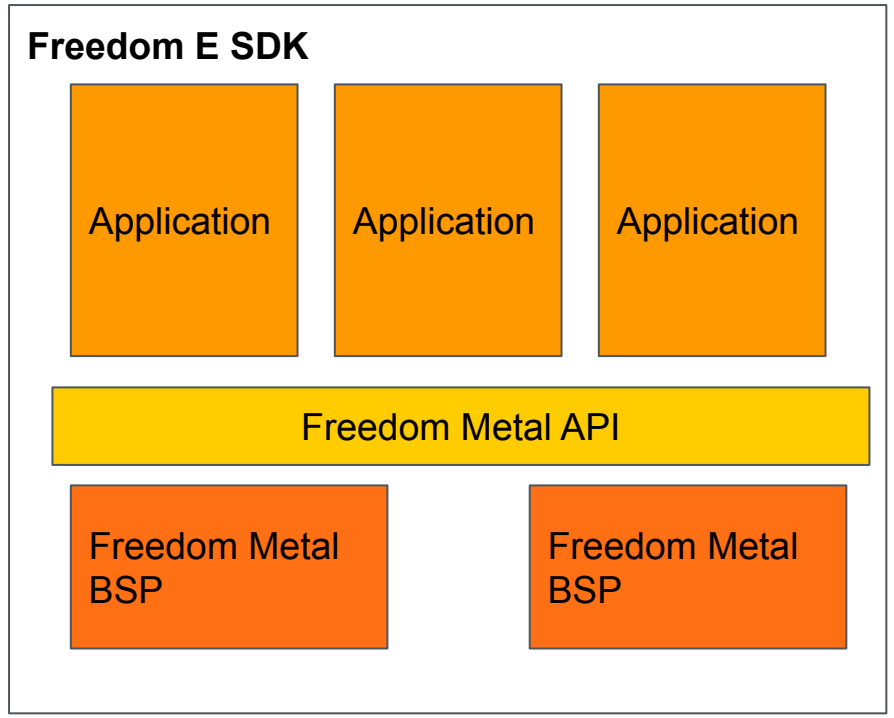
- Freedom Metal BSPs are automatically generated



https://github.com/sifive

Freedom Metal - A Bare Metal Framework for SiFive Devices

- **Freedom Metal provides:**
  - A bare-metal C application environment
  - An API for controlling CPU features and peripherals
  - The ability to retarget to any SiFive RISC-V product
- **Use Freedom Metal for:**
  - Writing portable hardware tests
  - Bootstrapping bare metal application development
  - A RISC-V hardware abstraction layer
  - And more!
- **Targeting Freedom Metal means your software will work on all SiFive devices**
  - API Documentation: https://sifive.github.io/freedom-metal-docs/
- **Freedom-E-SDK provides a complete Freedom Metal based CLI workflow**
  - https://github.com/sifive/freedom-e-sdk

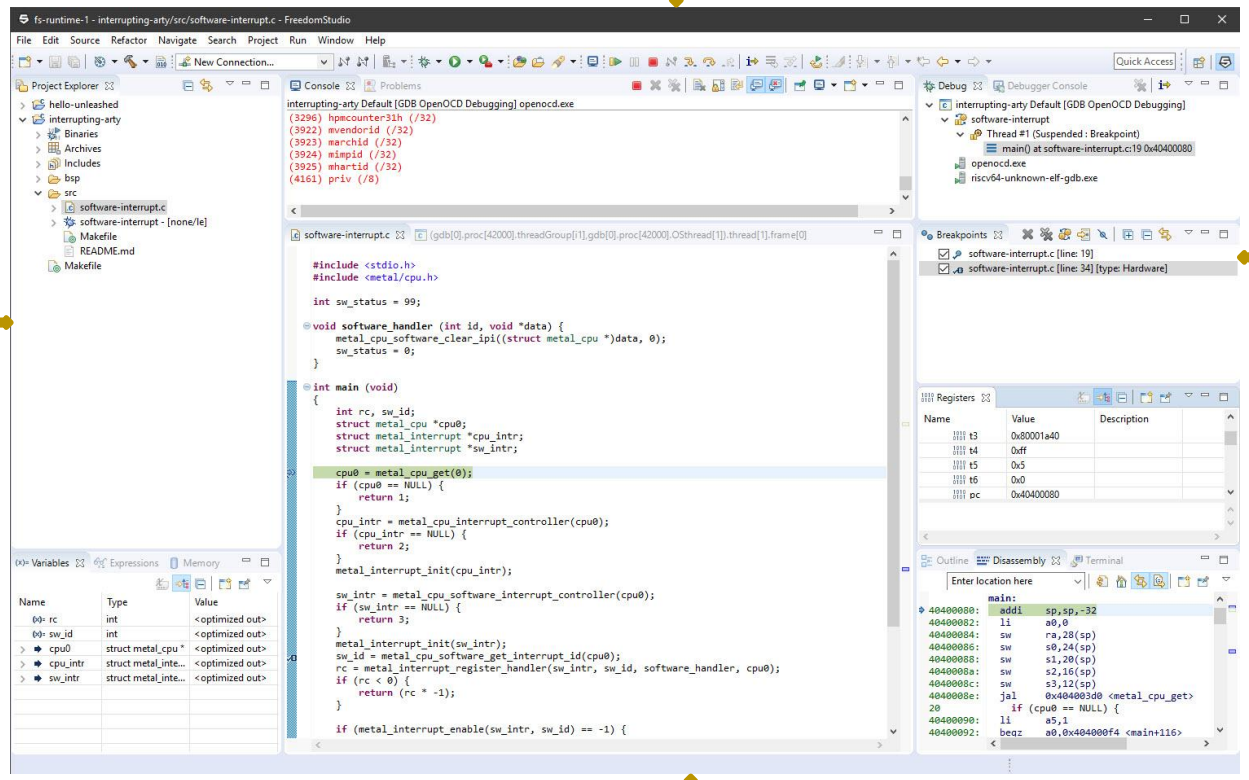


Freedom Metal BSPs are created for all SiFive RTL deliverables

SiFive Freedom Studio

SiFive Freedom-E-SDK

SiFive RISC-V GCC Toolchain



Eclipse Plug-Ins

Simulators

GNU GDB Debugger

OpenOCD

Segger J-Link OB

Olimex Probe

Target

Target

SiFive

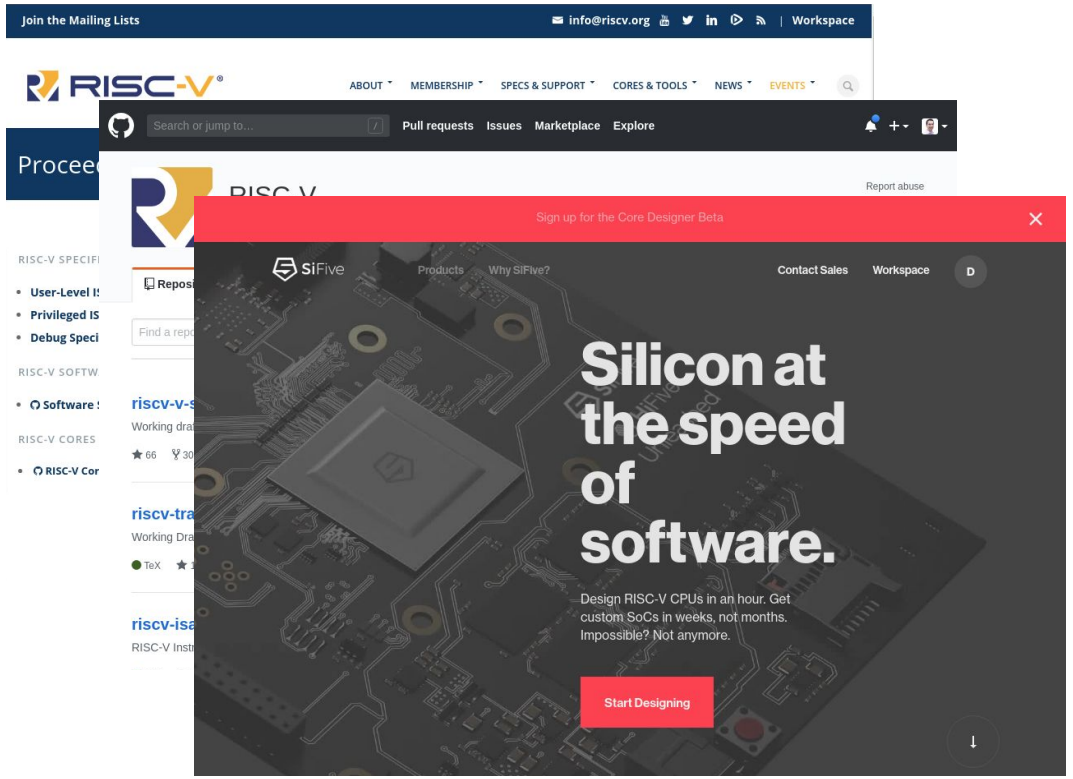
More Information

Resources

- **https://riscv.org/**
  - RISC-V Specifications
  - Links to the RISC-V mailing lists
  - Workshop proceedings

- **GitHub**
  - https://github.com/sifive/
  - https://github.com/riscv

- **https://www.sifive.com/**
  - RISC-V IP and Development Boards
  - RISC-V Tools
  - Forums



3-Part Webinar Series

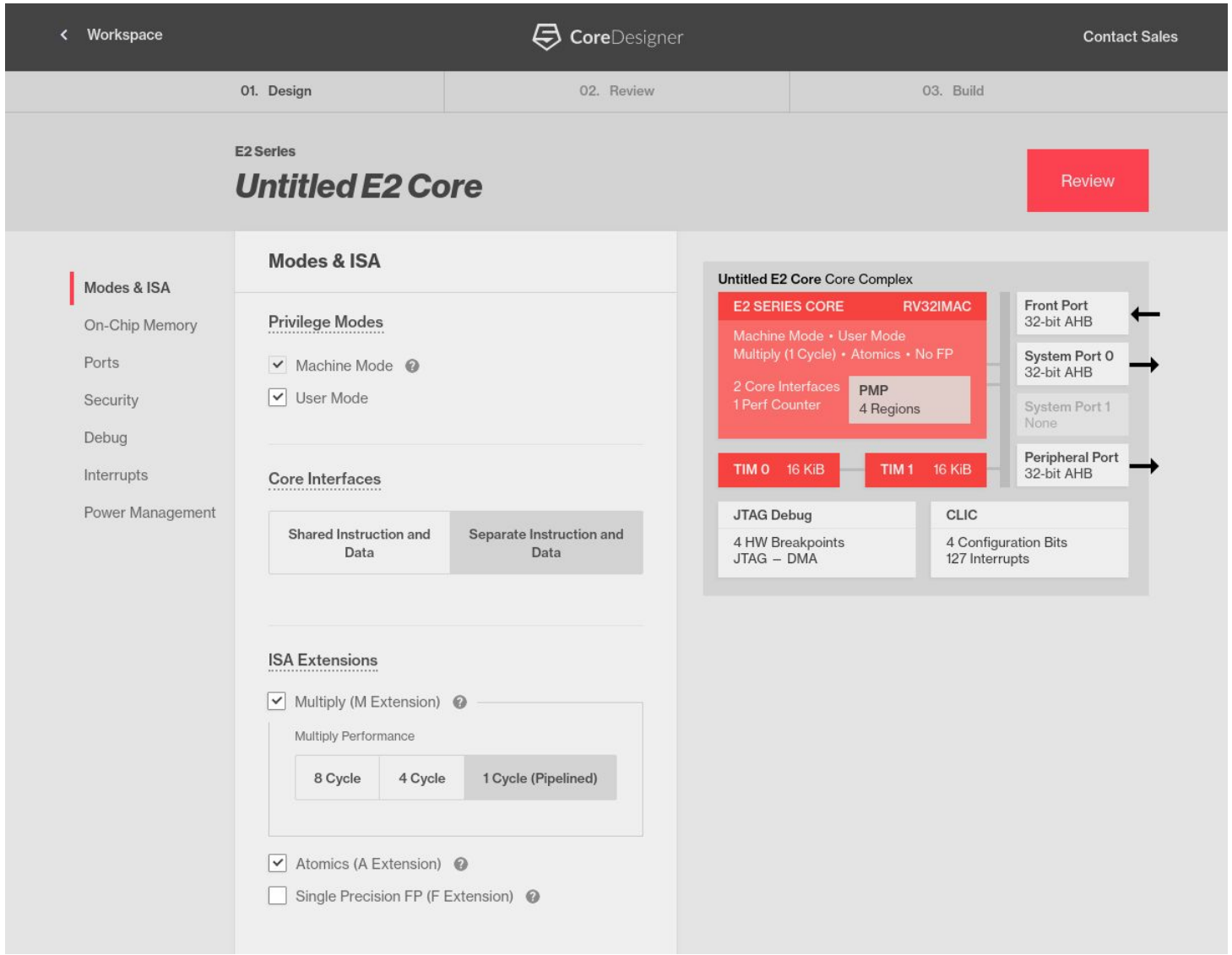
An Introduction to the RISC-V Architecture

SiFive's 2 Series Core IP

From a Custom 2 Series Core to Hello World in 30 Minutes

https://info.sifive.com/risc-v-second-webinar-series

Configure a 2 Series Core Now!!!



https://www.sifive.com/

Questions?