



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

Drew Barbier - Sr. Product Marketing Manager
Amy Lindburg - Director Product Management
May, 2019



About This Presentation

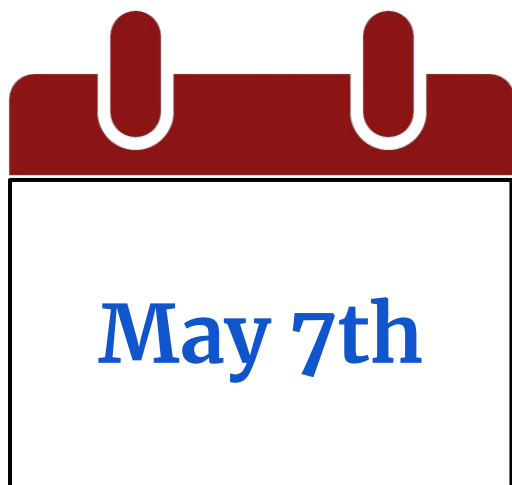
This presentation will introduce SiFive Core Designer, our RISC-V Core IP deliverables, and software development methodologies.

At the end of this presentation you will know how to create a custom SiFive 2 Series Core and write software targeting that core.

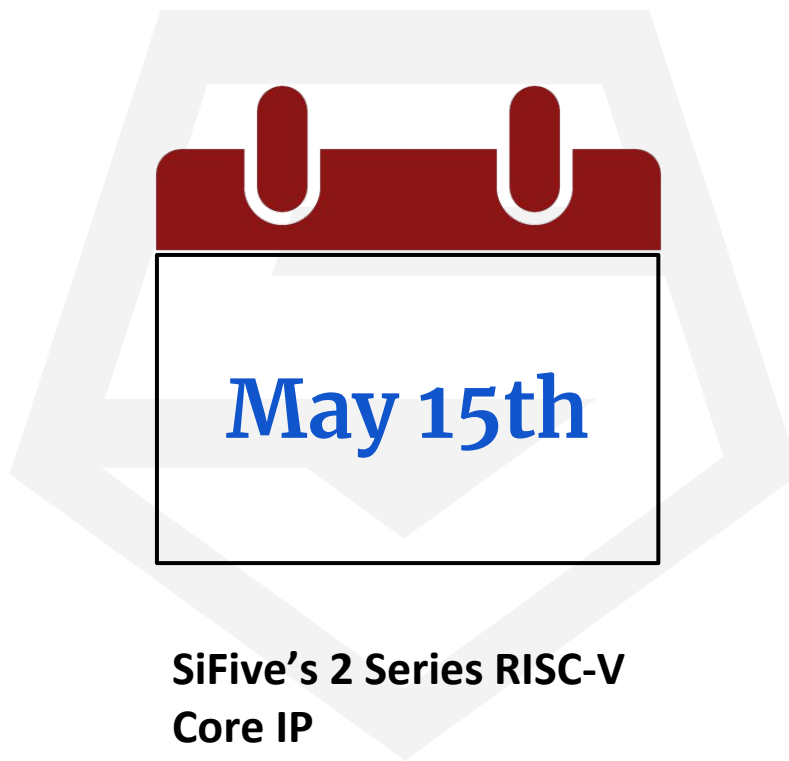




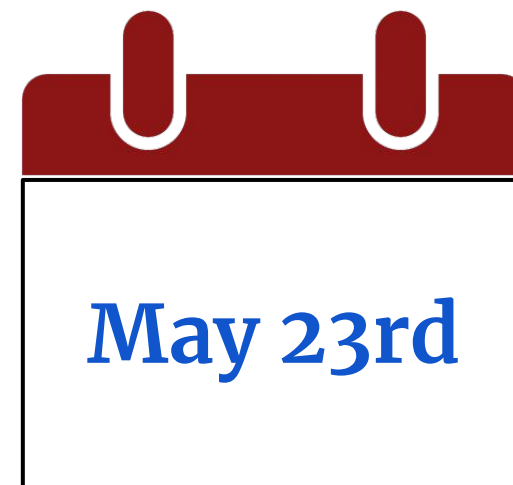
3-Part Webinar Series



**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>



SiFive Core Designer



2 Series IP Bundle



SCD-deliverable

- **arty-a7-100t-scd-deliverable**
 - FPGA bitstream
- **docs**
 - **Manual** - functional description of the deliverable
 - **User Guide** - detailed deliverable description including information on the integration, and testbench
 - **Arty FPGA User Guide** - describes how to use the FPGA deliverable
- **info**
 - Metadata about the design including device tree, retiming, and SiFive Insight description
- **tests**
 - Included testbench tests
- **verilog**
 - Design
 - the actual verilog design
 - Memories
 - SRAM behavioural models
 - sifive_insight
 - Sifive Insight modules
 - Testbench
 - simulation testbench
- **Makefile - runs the simulation testbench**



2 Series Clocking

2 Series has 2 clocks

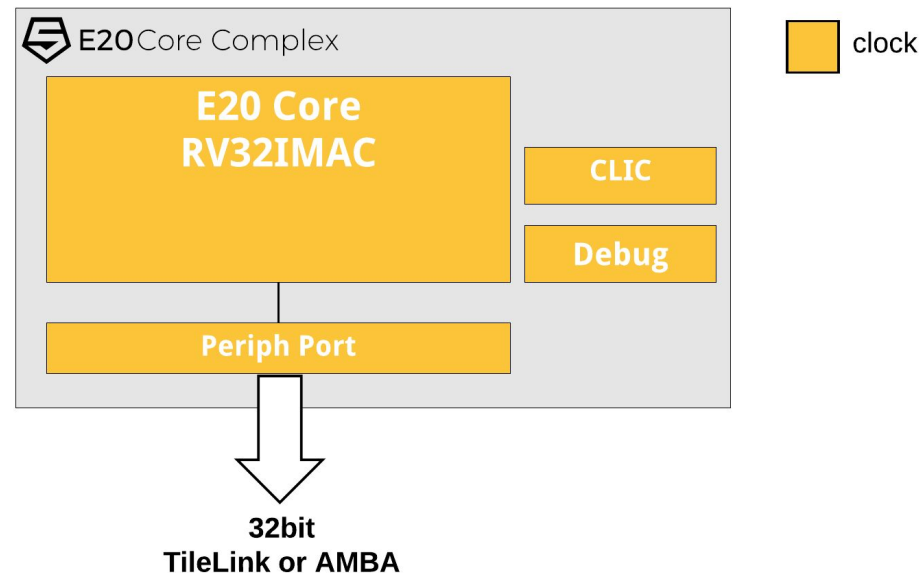
- **clock**
 - Main CPU and L1 memory clock
- **rtc_toggle**
 - Real Time Clock input as defined by RISC-V Architecture (mtime)
 - Must run at strictly less than half the rate of *clock*
- **Clocking relationship**
 - $\text{clock} > (2 \times \text{rtc_toggle})$

Constraints File

- Constraint files, if required, are documented in the User Guide

Retiming

- Module retiming can help improve frequency
- Modules which need to be retimed are documented in the info folder in the retiming_modules.txt file
- 2 Series FPU benefits from retiming





Software Development



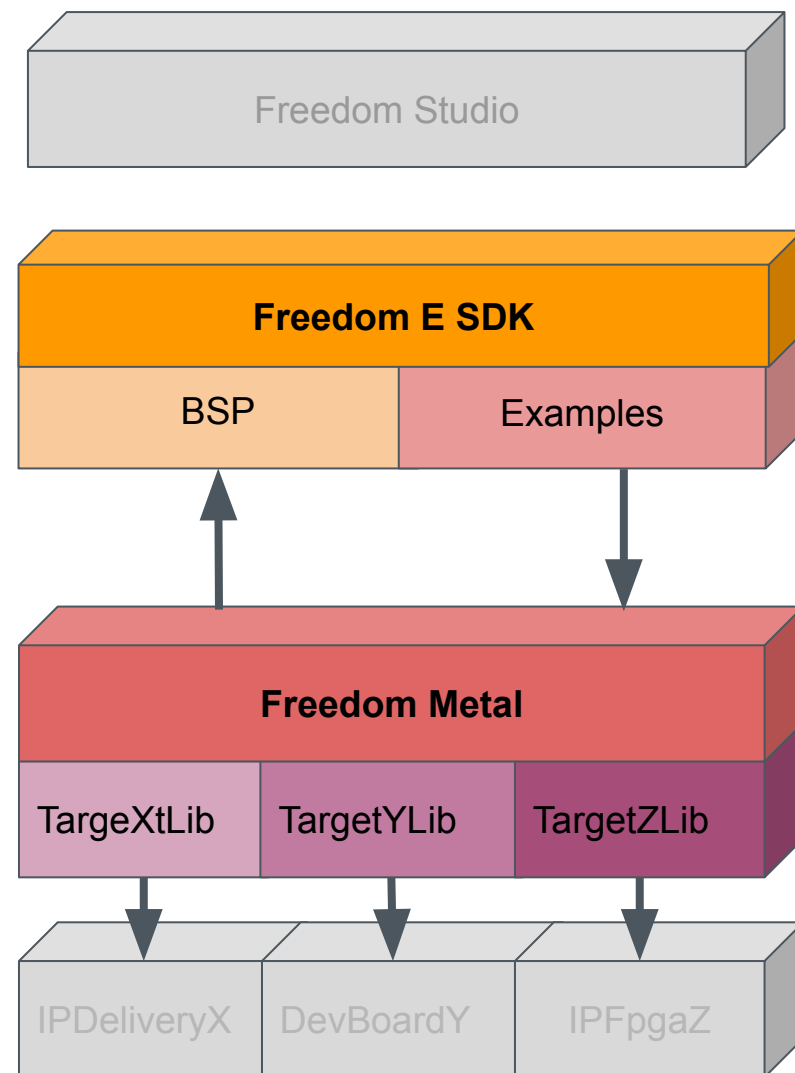
Freedom E SDK & Metal - A Bare Metal SW Stack for SiFive Devices

What is Freedom E SDK

- Embedded development kit providing a command line driven workflow with Examples and Utilities including BSP's for
 - Standard Core IP Deliverables
 - Standard Core FPGA Deliverables
 - SiFive Development Boards
- Examples use Freedom Metal to provide portability
- Open source repository
 - <https://github.com/sifive/freedom-e-sdk>

What is Freedom Metal

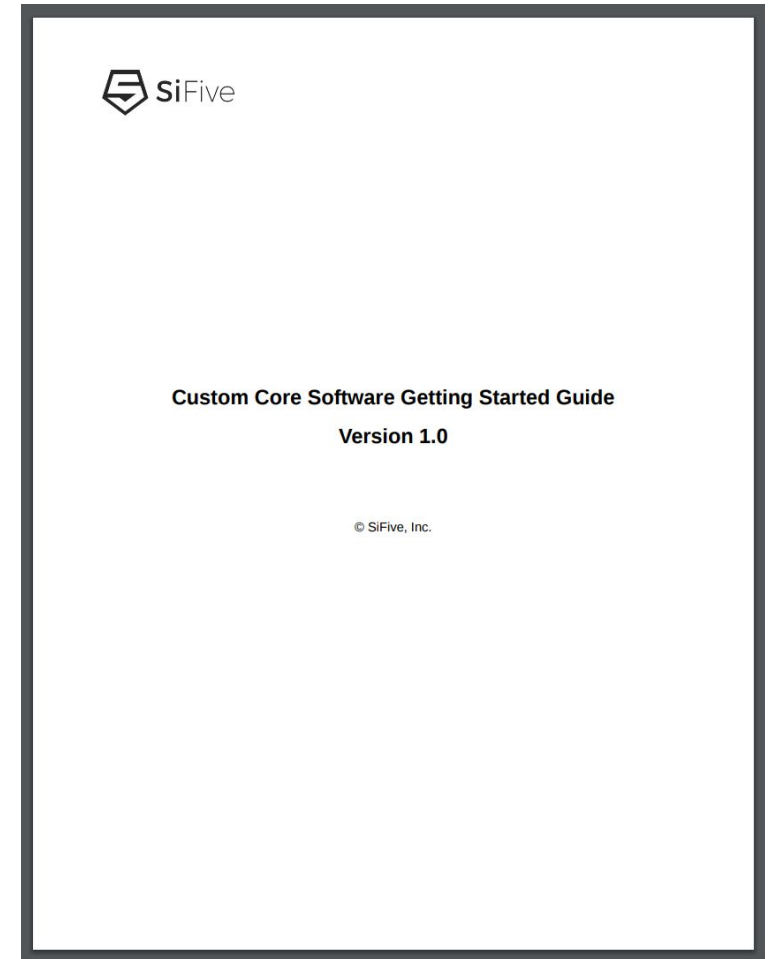
- Library for writing Portable, Bare Metal SW for all SiFive devices
 - A Bare Metal C application environment
 - An API for controlling CPU features and peripherals
 - The ability to retarget to any SiFive RISC-V product
 - A RISC-V hardware abstraction layer (HAL)
- Uses BSP's to provide target adaptation
- Open source repository
 - <https://github.com/sifive/freedom-metal>





Freedom Metal BSP for Custom Cores

- The customizability of our RISC-V Core IP also means that our software has to be as easily customizable
- SiFive have created tools to quickly and easily create Freedom Metal BSPs using the DTS delivered with our Core IP
 - <https://github.com/sifive/freedom-devicetree-tools>
 - Application Note describing how to create custom core BSPs
- Moving forward, Freedom Metal BSPs will be included in the IP delivery bundle



<https://www.sifive.com/documentation>

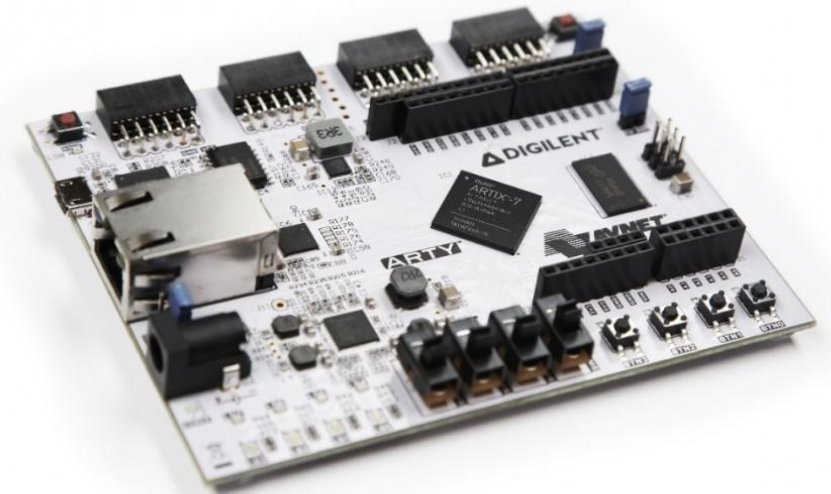


Using the FPGA Bitstream



Digilent Arty FPGA Platform

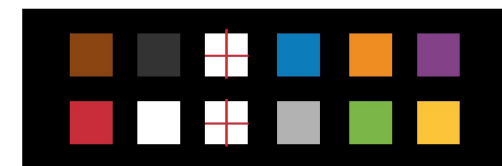
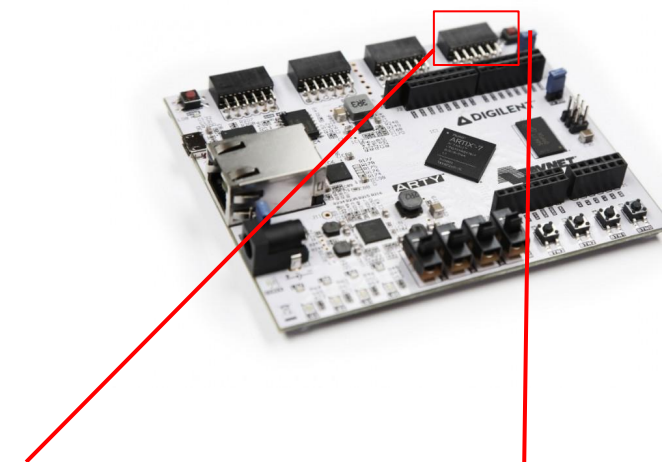
- **Popular, Low cost, FPGA development board**
 - Xilinx Artix-100T FPGA - 101,440 Xilinx logic cells
 - 16MB QSPI serial flash interface, 256MB DDR3
 - USB-UART, buttons, switches, LEDs, etc...
- **2 Series synthesises to 32 MHz for fast program execution**
 - Allows for evaluation of CPU performance running real software on real hardware at speed
- **Purchase directly from Digilent:**
 - <http://store.digilentinc.com/artix-a7-artix-7-fpga-development-board-for-makers-and-hobbyists/>



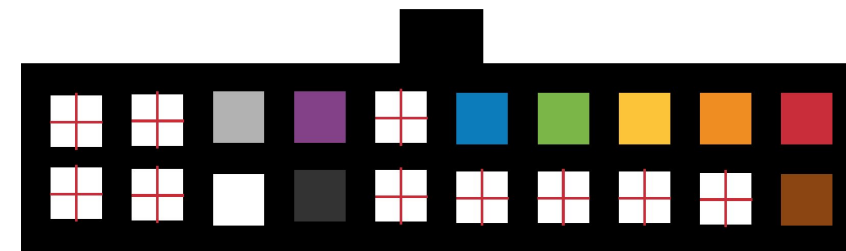


Connecting the FPGA to a JTAG Debugger

- Debug signals are connected to the Arty Board's JD PMOD header (closest to the reset button)
- Tested Probes:
 - Olimex ARM-USB-Tiny (OpenOCD)
 - SEGGER JLINK
 - IAR i-jet
 - Lauterbach TRACE32
- JTAG Signal Mapping
 - Purple – TDO Yellow - TDI
 - Orange- nTRST Green – TMS
 - Blue - TCK Grey - nRST
 - Black - GND White - GND
 - Brown – VREF RED - VREF



JD PMOD

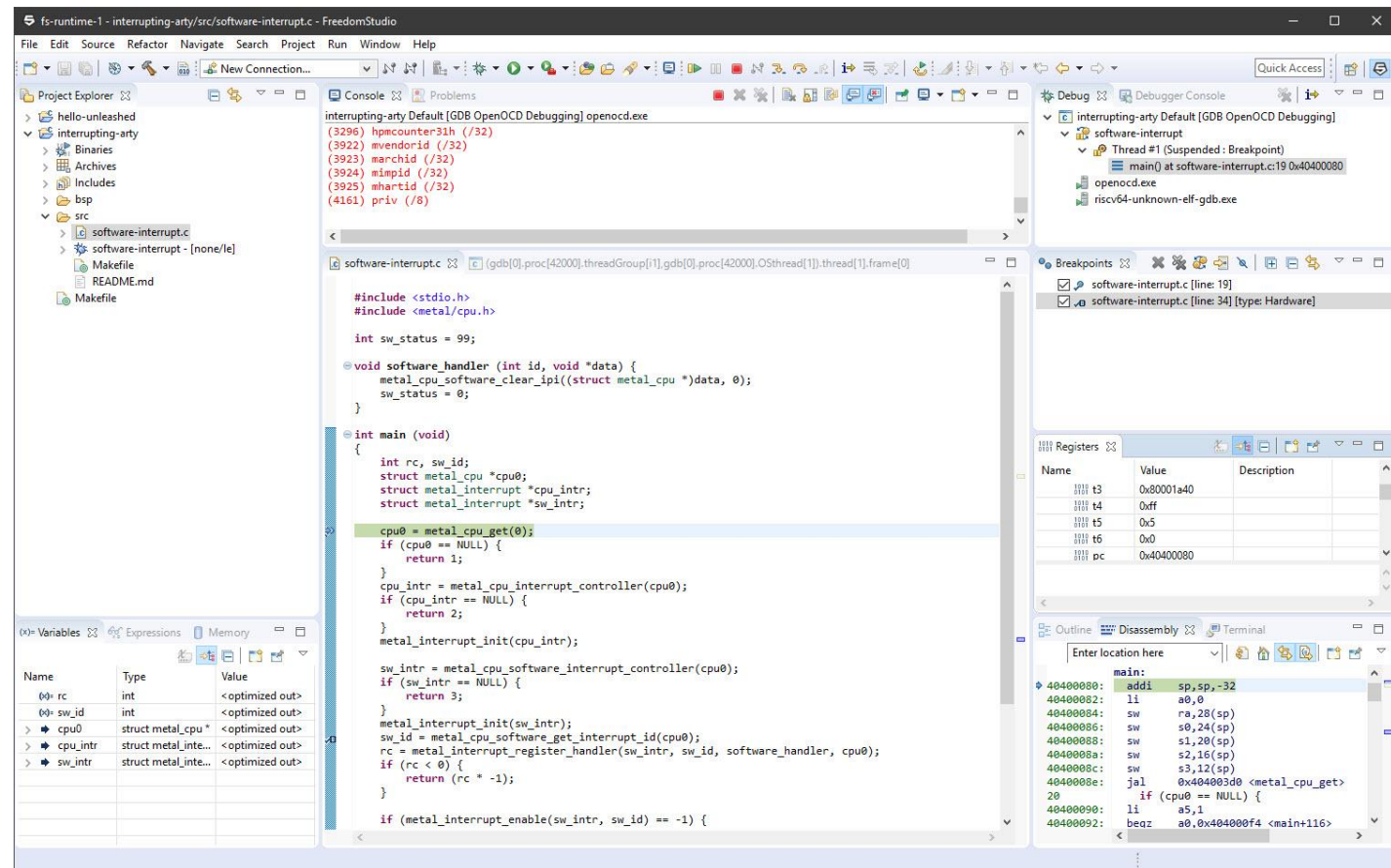


ARM 20 Pin Header



Demo

1. Use Freedom Studio to program the Arty 100T FPGA with the SiFive Core Designer generated bitfile
2. Create a new project targeting this specific FPGA platform
3. Program the application into the FPGA and run/debug it



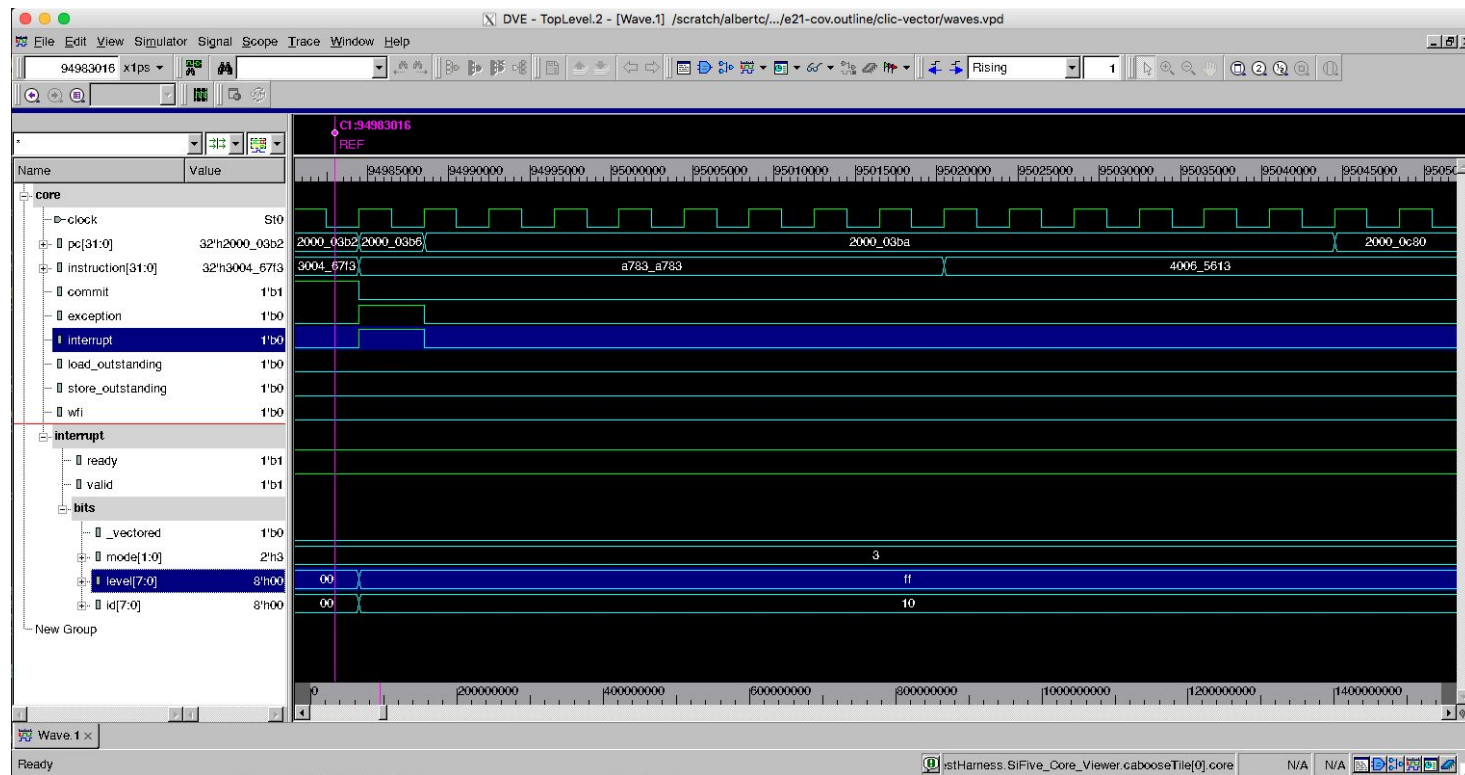


Using the RTL Testbench



Demo

1. Use Freedom Studio to create a new project targeting the RTL testbench
2. Run the test
3. Explore the resulting wave using SiFive Insight





Testbench Logs

hartid Cycle count Inst. Committed PC Address Register Values Opcodes

```
drew@gamma06: /scratch/drew/coreplex-release/sifive_coreip_E21_AHB_rtl_full_v19_02
```

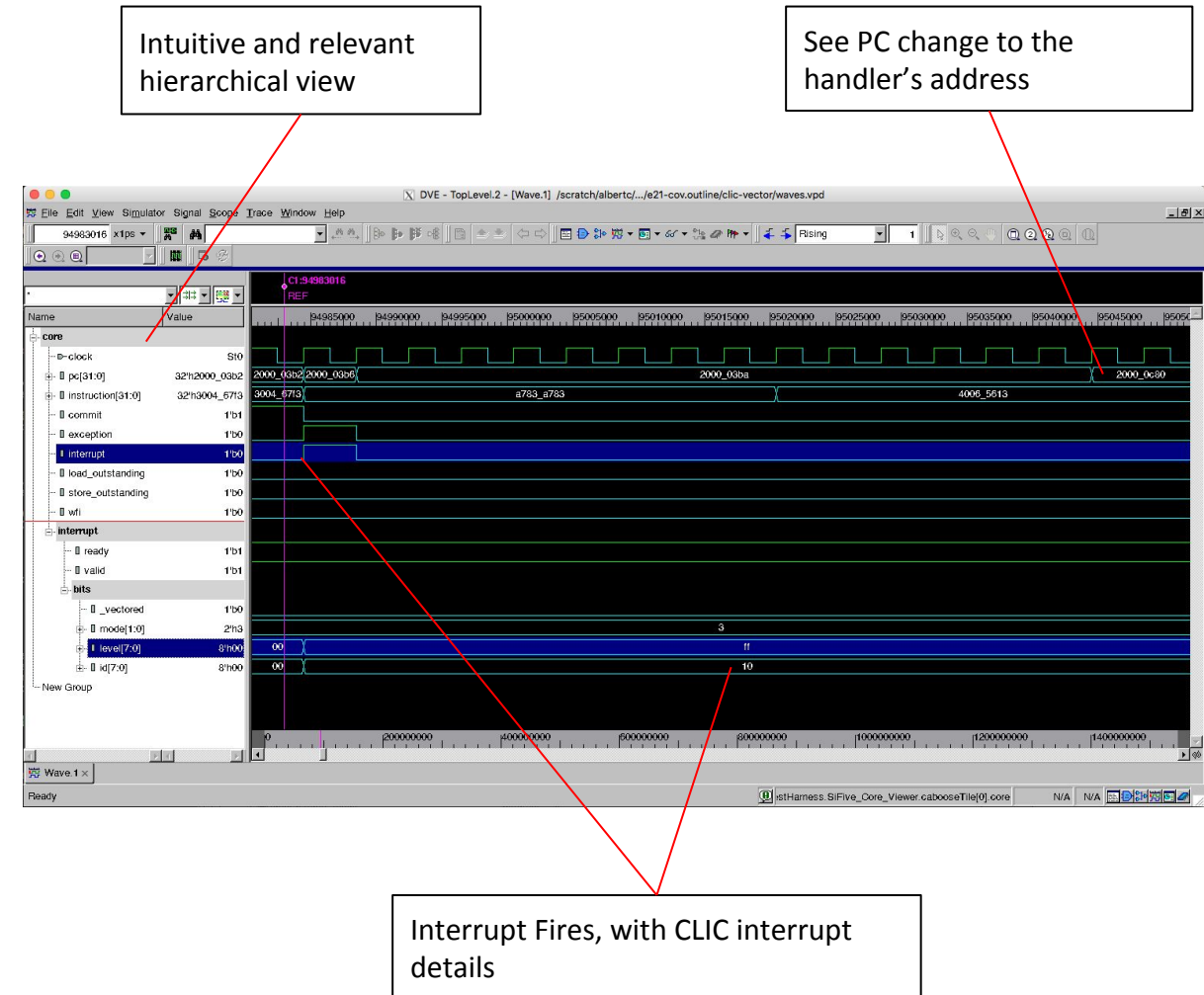
C0:	628865	[1]	pc=[8000d416]	W[r 1=8000d418]	R[r15=8000d32e]	R[r 0=00000000]	inst=[00009782]	DASM(00009782)
C0:	628866	[0]	pc=[8000d32e]	W[r 0=8000d418]	R[r 2=80005130]	R[r 8=80005150]	inst=[00001101]	DASM(00001101)
C0:	628867	[1]	pc=[8000d32e]	W[r 2=80005100]	R[r 2=80005130]	R[r14=00000000]	inst=[00007179]	DASM(00007179)
C0:	628868	[1]	pc=[8000d330]	W[r 0=80005100]	R[r 2=80005100]	R[r 8=80005150]	inst=[0000d622]	DASM(0000d622)
C0:	628869	[1]	pc=[8000d332]	W[r 8=80005130]	R[r 2=80005100]	R[r 8=80005150]	inst=[00001800]	DASM(00001800)
C0:	628870	[1]	pc=[8000d334]	W[r 0=80005130]	R[r 8=80005130]	R[r10=800020d8]	inst=[fca42e23]	DASM(fca42e23)
C0:	628871	[1]	pc=[8000d338]	W[r 0=80005130]	R[r 8=80005130]	R[r11=00000001]	inst=[fcb42c23]	DASM(fcb42c23)
C0:	628872	[1]	pc=[8000d33c]	W[r15=800020d8]	R[r 8=80005130]	R[r28=00000002]	inst=[fdc42783]	DASM(fdc42783)
C0:	628873	[0]	pc=[8000d340]	W[r 0=8000510c]	R[r 8=80005130]	R[r15=8000510c]	inst=[fef42623]	DASM(fef42623)
C0:	628874	[1]	pc=[8000d340]	W[r 0=8000510c]	R[r 8=80005130]	R[r15=800020d8]	inst=[fef42623]	DASM(fef42623)
C0:	628875	[1]	pc=[8000d344]	W[r15=00000001]	R[r 8=80005130]	R[r24=918a0623]	inst=[fd842783]	DASM(fd842783)
C0:	628876	[0]	pc=[8000d348]	W[r 0=80005108]	R[r15=80005108]	R[r16=00000000]	inst=[01079713]	DASM(01079713)
C0:	628877	[1]	pc=[8000d348]	W[r14=00010000]	R[r15=00000001]	R[r16=00000000]	inst=[01079713]	DASM(01079713)
C0:	628878	[1]	pc=[8000d34c]	W[r15=00000001]	R[r 8=80005130]	R[r24=918a0623]	inst=[fd842783]	DASM(fd842783)
C0:	628879	[0]	pc=[8000d350]	W[r 0=80005108]	R[r15=80005108]	R[r 0=00000000]	inst=[0000e789]	DASM(0000e789)
C0:	628880	[1]	pc=[8000d350]	W[r 0=80005108]	R[r15=00000001]	R[r 0=00000000]	inst=[0000e789]	DASM(0000e789)
C0:	628881	[0]	pc=[8000d35a]	W[r 0=80005108]	R[r15=00000001]	R[r13=00000001]	inst=[00006795]	DASM(00006795)
C0:	628882	[1]	pc=[8000d35a]	W[r15=00003000]	R[r15=00000001]	R[r11=00000001]	inst=[0000678d]	DASM(0000678d)
C0:	628883	[1]	pc=[8000d35c]	W[r15=00003333]	R[r15=00003000]	R[r19=918a0623]	inst=[33378793]	DASM(33378793)
C0:	628884	[1]	pc=[8000d360]	W[r15=00013333]	R[r15=00003333]	R[r14=00010000]	inst=[000097ba]	DASM(000097ba)
C0:	628885	[1]	pc=[8000d362]	W[r 0=00013333]	R[r 8=80005130]	R[r15=00013333]	inst=[fef42423]	DASM(fef42423)
C0:	628886	[1]	pc=[8000d366]	W[r15=800020d8]	R[r 8=80005130]	R[r12=00000001]	inst=[fec42783]	DASM(fec42783)
C0:	628887	[0]	pc=[8000d36a]	W[r 0=8000511c]	R[r15=8000511c]	R[r15=8000511c]	inst=[0000479c]	DASM(0000479c)
C0:	628888	[1]	pc=[8000d36a]	W[r15=00004000]	R[r15=800020d8]	R[r15=800020d8]	inst=[0000479c]	DASM(0000479c)
C0:	628889	[0]	pc=[8000d36c]	W[r 0=800020e0]	R[r 0=00000000]	R[r15=800020e0]	inst=[0000873e]	DASM(0000873e)
C0:	628890	[1]	pc=[8000d36c]	W[r14=00004000]	R[r 0=00000000]	R[r15=00004000]	inst=[0000873e]	DASM(0000873e)
C0:	628891	[1]	pc=[8000d36e]	W[r15=00013333]	R[r 8=80005130]	R[r 8=80005130]	inst=[fe842783]	DASM(fe842783)
C0:	628892	[0]	pc=[8000d372]	W[r 0=80005118]	R[r14=00004000]	R[r15=80005118]	inst=[0000c31c]	DASM(0000c31c)
C0:	628893	[1]	pc=[8000d372]	W[r 0=80005118]	R[r14=00004000]	R[r15=00013333]	inst=[0000c31c]	DASM(0000c31c)
C0:	628894	[1]	pc=[8000d374]	W[r 0=8000d376]	R[r15=00013333]	R[r11=00000001]	inst=[0000bfcd]	DASM(0000bfcd)
C0:	628895	[0]	pc=[8000d366]	W[r 0=8000d376]	R[r 2=80005100]	R[r 8=80005130]	inst=[00001101]	DASM(00001101)

SiFive testbench produces logs for each test providing cycle-by-cycle information



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
- **SiFive Insight is conditionally included in the design**
 - Use for simulation, and by default is removed for synthesis
 - Or... Include in synthesis and connect SCV signals to on chip debug facilities like chipscope or Debug IP



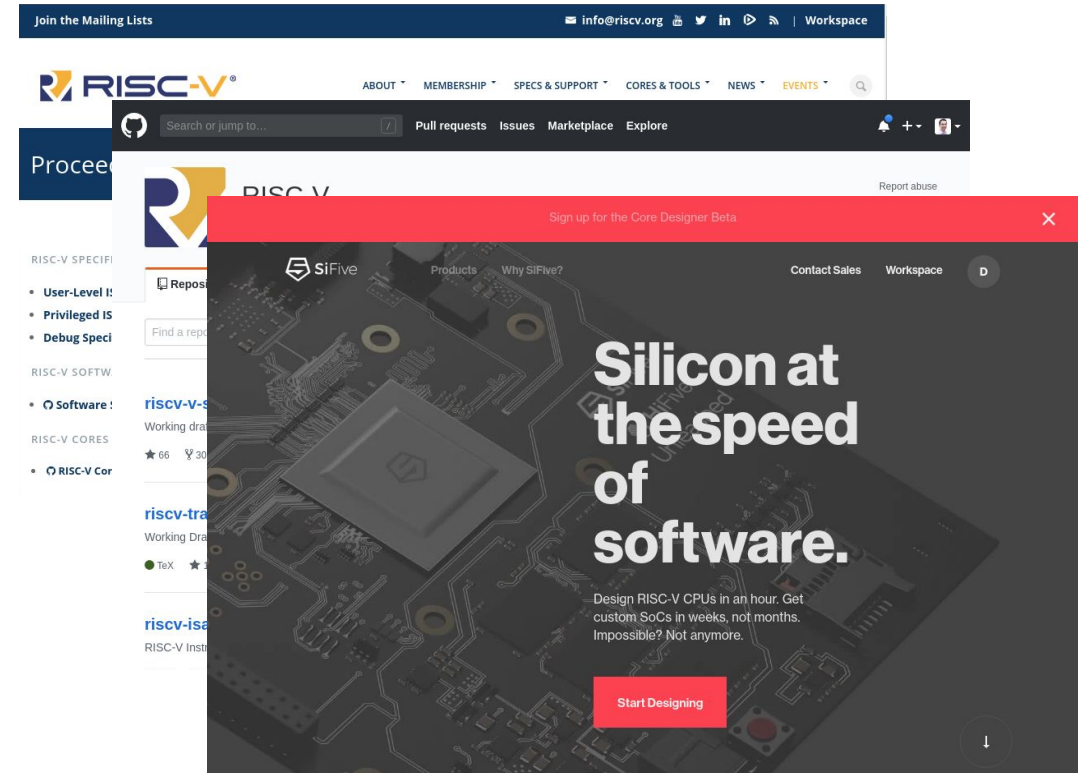


More Information



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





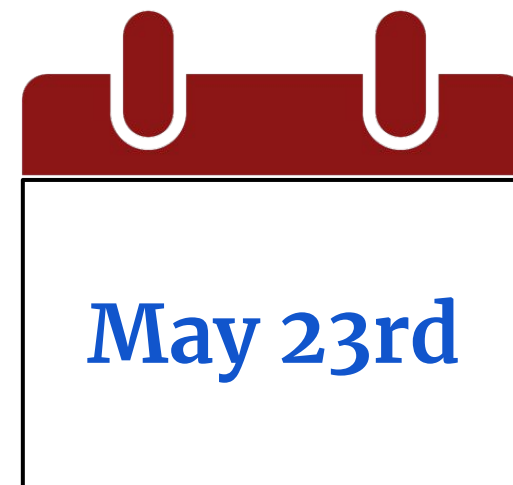
3-Part Webinar Series



**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!!!

Workspace CoreDesigner Contact Sales

01. Design 02. Review 03. Build

E2 Series
Untitled E2 Core Review

Modes & ISA

Modes & ISA
On-Chip Memory
Ports
Security
Debug
Interrupts
Power Management

Privilege Modes

☒ Machine Mode ?
☒ User Mode

Core Interfaces

Shared Instruction and Data
Separate Instruction and Data

ISA Extensions

☒ Multiply (M Extension) ?
Multiply Performance
8 Cycle 4 Cycle 1 Cycle (Pipelined)
☒ Atomics (A Extension) ?
☐ Single Precision FP (F Extension) ?

Untitled E2 Core Core Complex

E2 SERIES CORE RV32IMAC
Machine Mode • User Mode
Multiply (1 Cycle) • Atomics • No FP
2 Core Interfaces 1 Perf Counter
PMP 4 Regions

Front Port 32-bit AHB
System Port 0 32-bit AHB
System Port 1 None
Peripheral Port 32-bit AHB

TIM 0 16 KiB TIM 1 16 KiB

JTAG Debug
4 HW Breakpoints
JTAG – DMA

CLIC
4 Configuration Bits
127 Interrupts

<https://www.sifive.com/>



Thank You
