SiFive

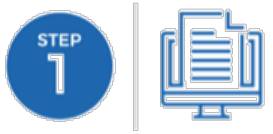
Evaluating SiFive RISC-V Core IP

Drew Barbier – January 2018
drew@sifive.com

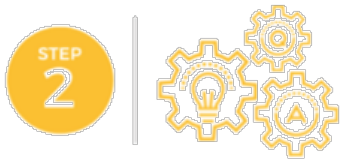
3 Part Webinar Series

Webinar Recordings and Slides: https://info.sifive.com/risc-v-webinar

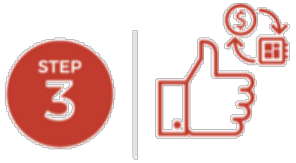
- RISC-V 101
  - The Fundamentals of RISC-V architecture
- Introduction to SiFive RISC-V Core IP
  - Introduction to the SiFive E31 and E51 Core Complexes
- Getting Started with SiFive RISC-V Core IP



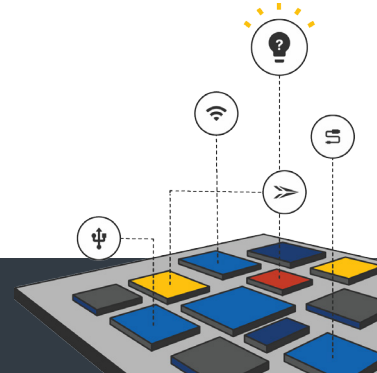
Study



Evaluate



Buy

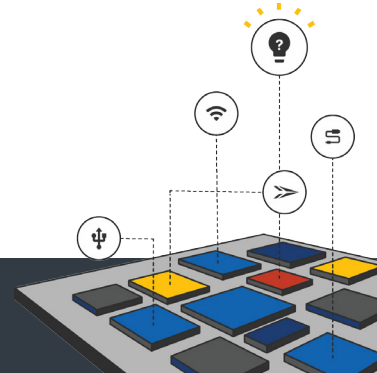


SiFive

SiFive RISC-V Core IP Products

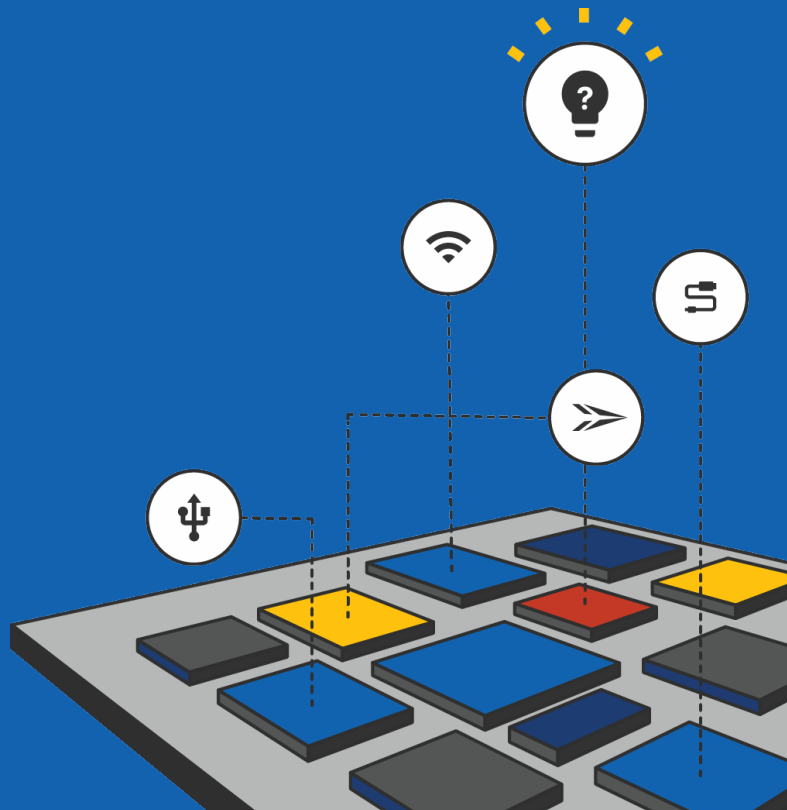
- This presentation is targeted at hardware and software embedded designers who want to get started evaluating SiFive RISC-V Core IP

- This presentation will walk through the RTL Evaluation deliverable of the E31 Core Complex as well programing, writing software, and Debugging the E31 Arty FPGA image



SiFive

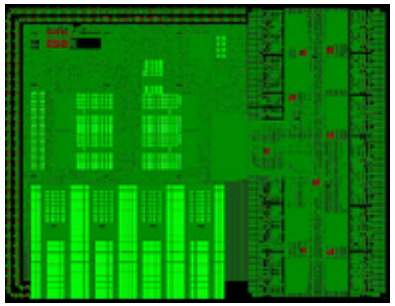
Obtaining the Evaluations



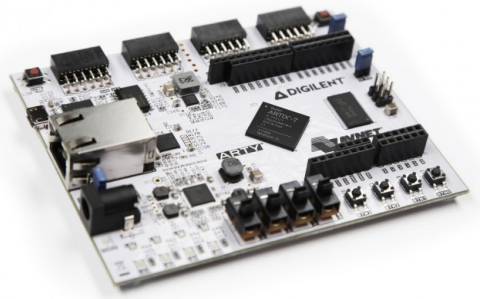
SiFive RISC-V Core IP Evaluations

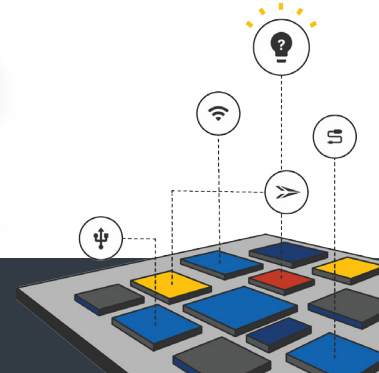
Free, Instant Access to Evaluation RTL and FPGA bitstreams

- **Verilog** RTL Evaluation
  - Fully functional, synthesizable, **Verilog RTL**
    - Synthesize into your process/eda flow to obtain accurate PPA for your process node
    - Simulate in your environment
  - Restrictions on Evaluation RTL
    - Obfuscated Verilog
    - Reduced DTIM and System/Peripheral Port address space

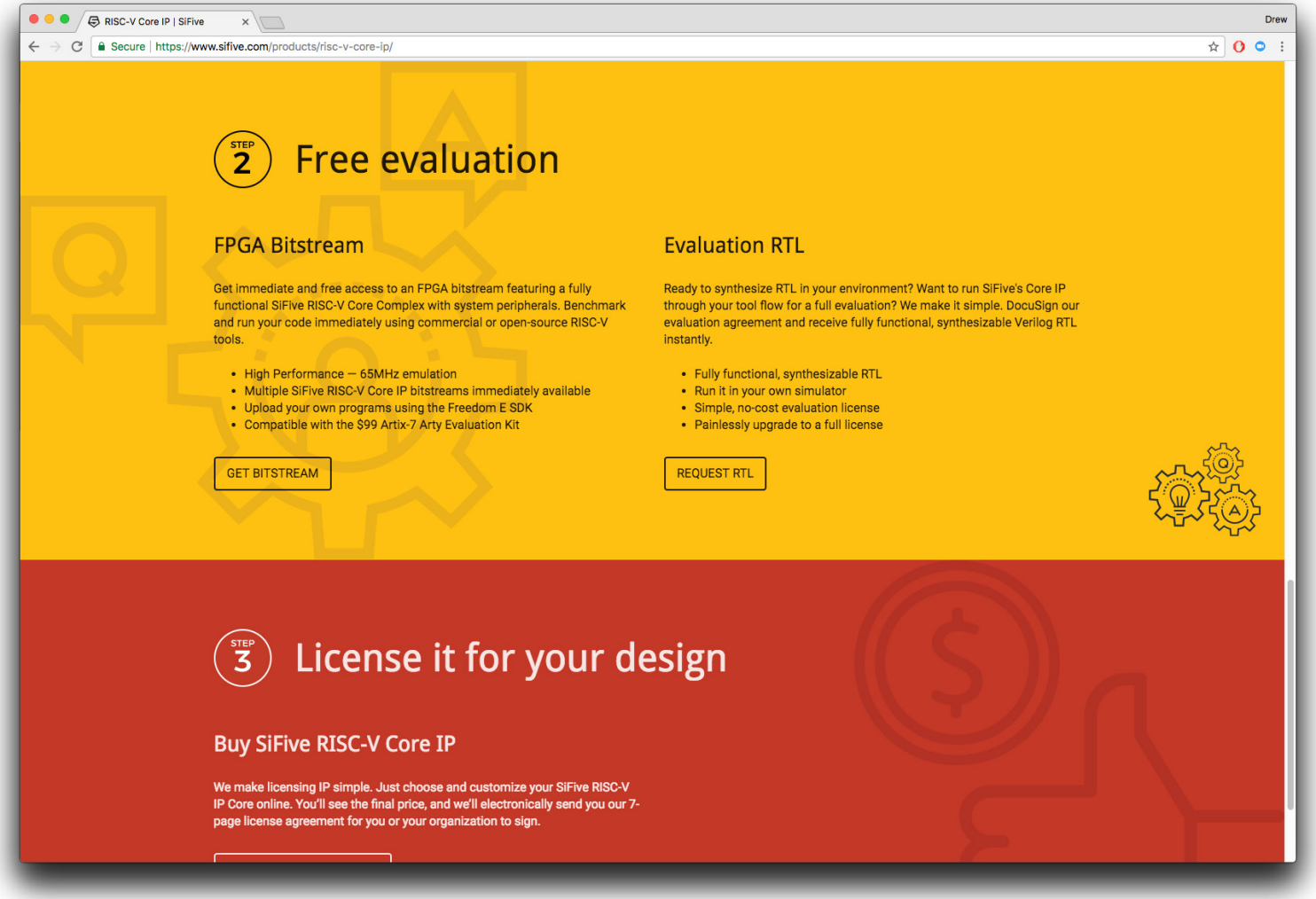


- FPGA Evaluation
  - Pre-built FPGA bitstreams for low cost FPGA Platforms
  - Useful for software development

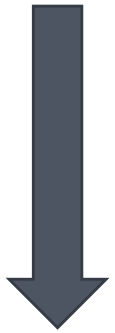


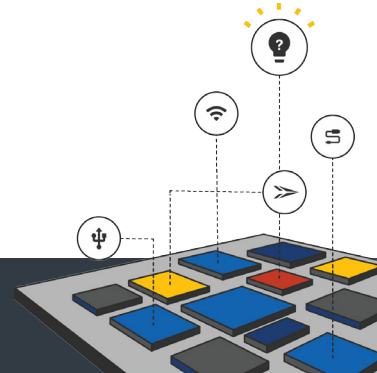


SiFive Product Evaluations are 2 Clicks Away



Scroll Down

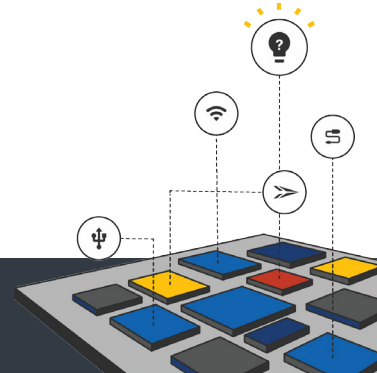




Evaluation License Agreement

DocuSign
The Global Standard for eSignature®

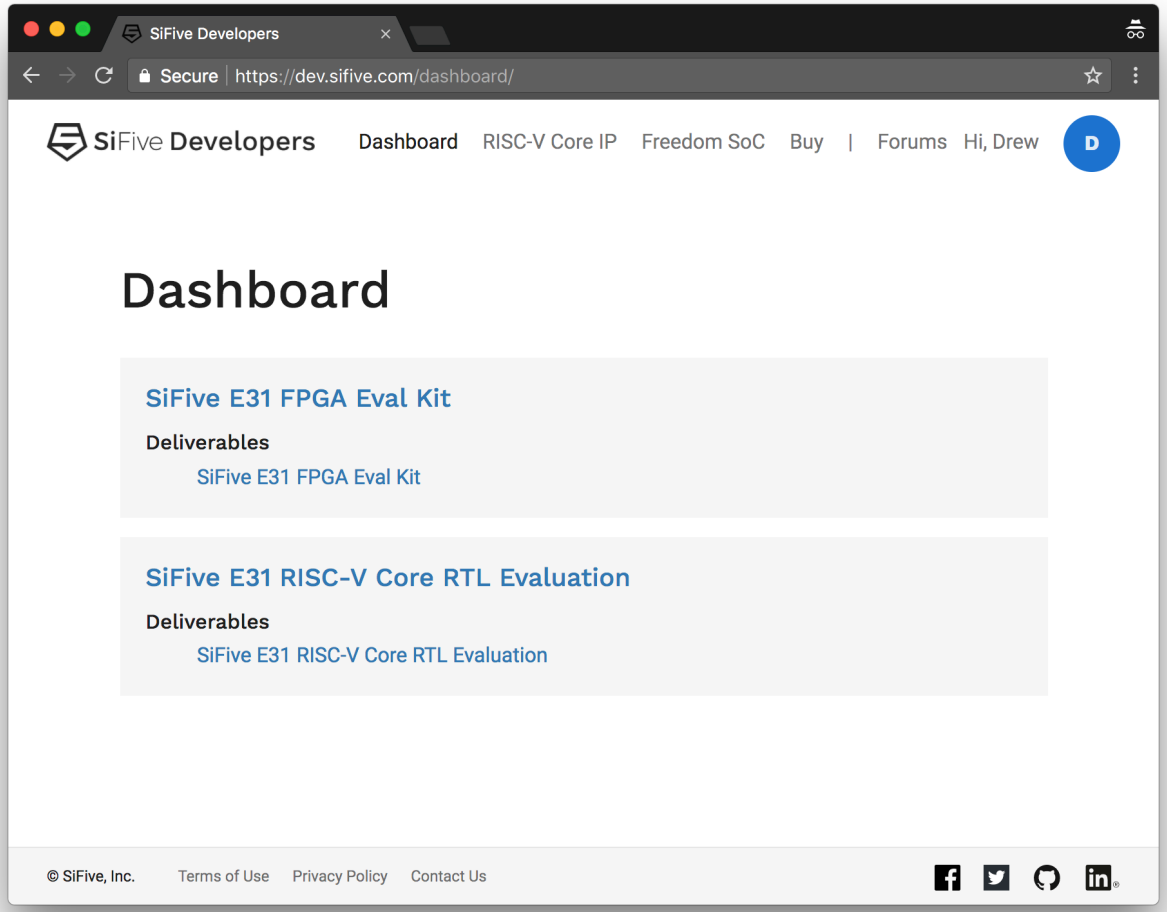
- Fill in your information on the web form
- An Evaluation agreement is then sent to the email address provided via DocuSign
- Sign the document via DocuSign being sure to click Finish at the bottom of the document
- Help:
  - sales@sifive.com
  - order-support@sifive.com

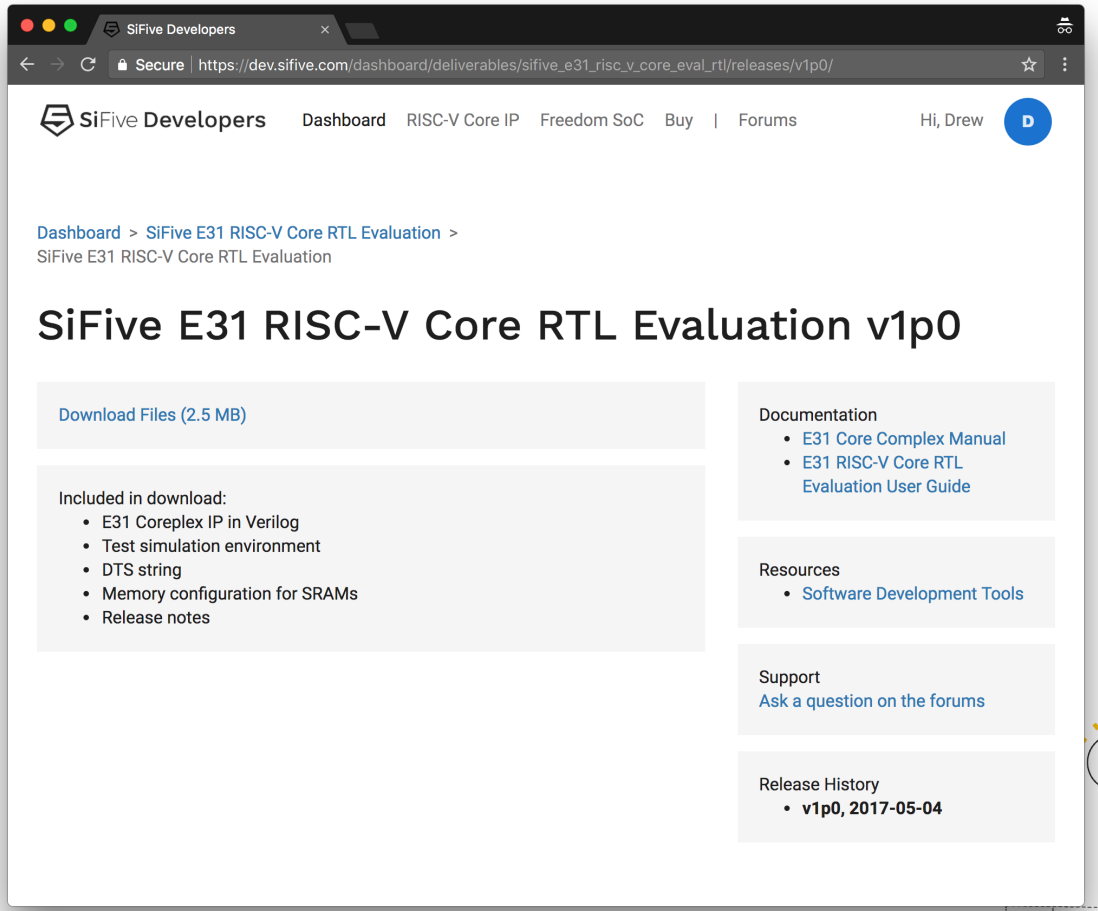


SiFive

Download via SiFive Developer Dashboard

https://dev.sifive.com

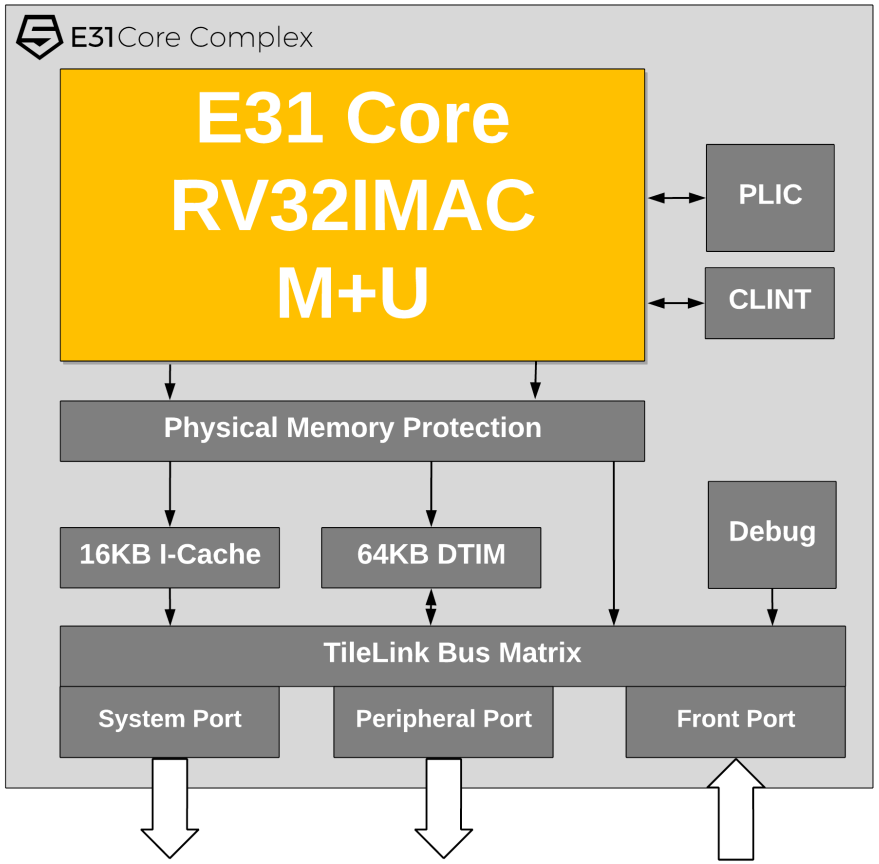


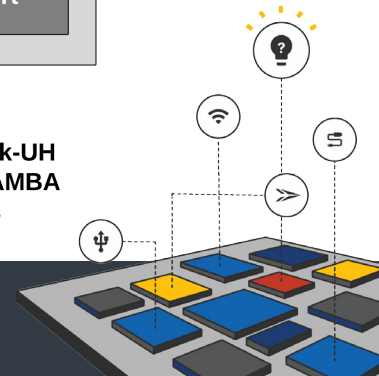


SiFive

Evaluation Product Refresh – January 2018

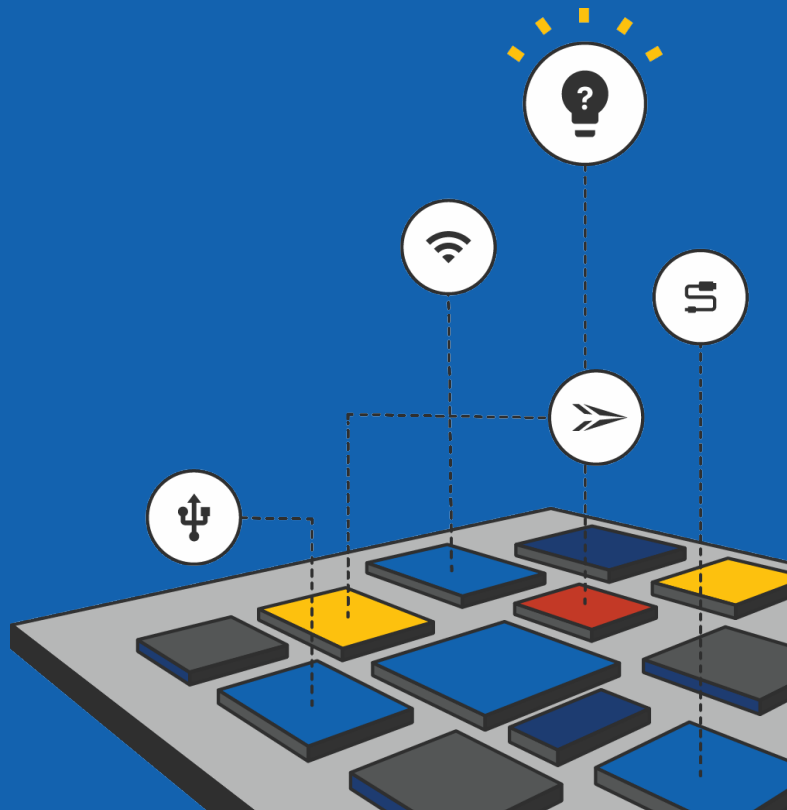
- RTL Evaluation Changes
  - ITIM
  - 4 Hardware Breakpoints
  - Performance Counters
  - Port Interfaces
    - E31 – AHB-Lite
    - E51 – AXI
  - Number of Global Interrupts
    - E31 – 127
    - E51 – 255
- FPGA Evaluation Changes
  - ITIM
  - 64kB of DTIM
  - 8 Hardware Breakpoints





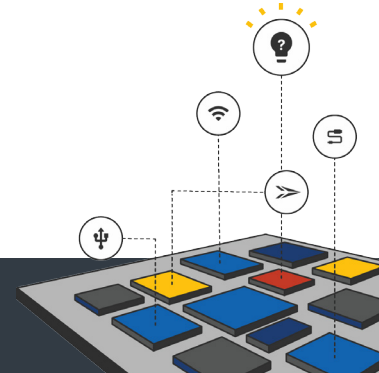
SiFive

E31 RTL Evaluation



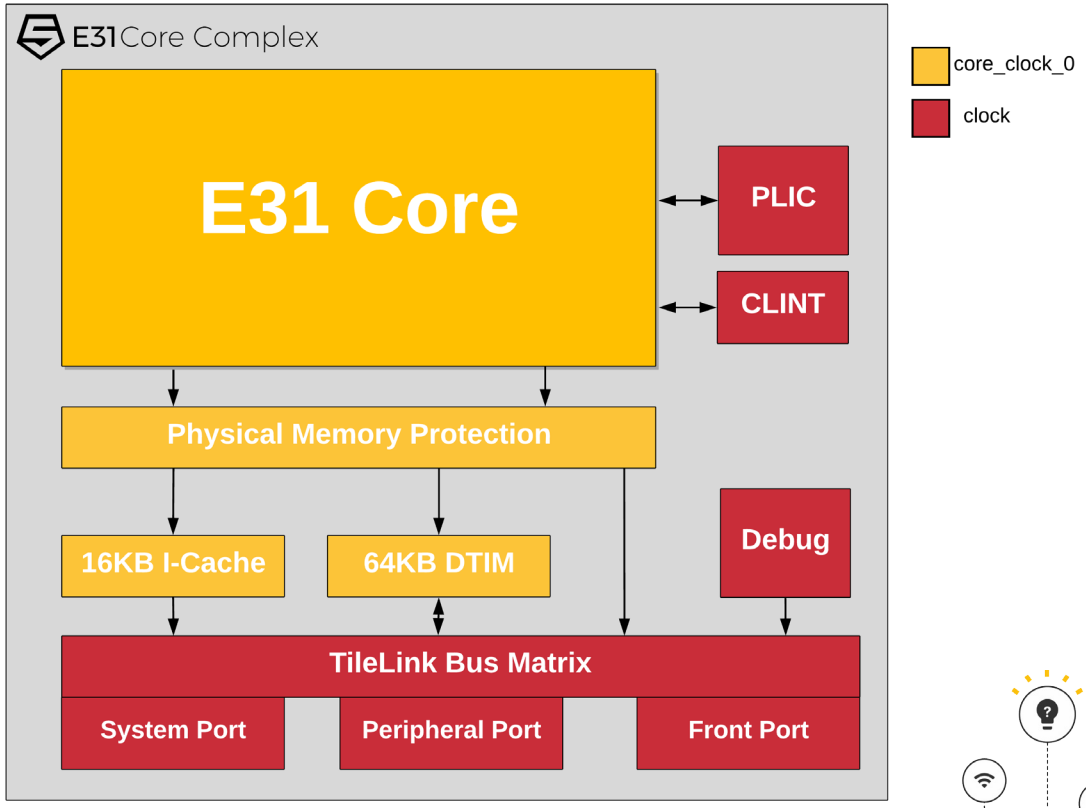
The Verilog RTL Bundle

- *Sifive_coreip_E31_AHB_rtl_eval_v2p0/*
  - *info/*
    - metadata about the design (DTS and memory dimensions)
  - *tests/*
    - simple tests to ensure a functional delivery
    - Also includes device header files
  - ***verilog**/*
    - *design/*
      - DUT (device under test)
    - *memories/*
      - behavioral models of SRAMs
    - *testbench/*
      - simulation testbench
  - *Makefile*
    - Runs VCS simulations



E-Series Core Complex Clocking

- *core_clock_0*
  - Main CPU and L1 memory clock
- *clock*
  - Secondary peripheral clock
  - Frequency must be an integer multiple of *core_clock_0*
  - 1:1 ratio is acceptable
- *rtc_toggle*
  - Real Time Clock input as defined by RISC-V Architecture (mtime)
  - Must run at strictly less than half the rate of *clock*
- *core_clock_0 ≥ clock > (2x rtc_toggle)*

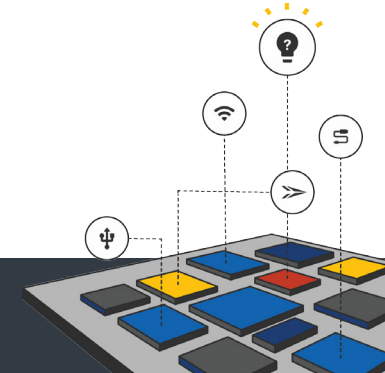


E31 Core Complex Clocking

Synthesis Constraints and Clocking Recommendations

- For Maximum frequency, it is recommended for:
  - clock = ½(core_clock_0)
- If targeting lower frequencies, *clock = core_clock_0* is OK
- Typical *rtc_toggle* frequencies
  - 32.768kHz
  - 1MHz

```
#####################################################
#
# Local Synopsys Design Constraints (SDC)
#
#####################################################

#####################################################
# Timing constraints
#####################################################

#-----------------------------------------------------
# Clocks
#-----------------------------------------------------

# set period to create desired core frequency
set CLK_PERIOD 5
set SYS_CLK_PERIOD [expr $CLK_PERIOD * 2]

# create core clock
create_clock -name ``core_clk'' -period $CLK_PERIOD [get_ports core_clock_0]

# create system clock
create_clock -name ``sys_clk'' -period $SYS_CLK_PERIOD [get_ports clock]
```



Core Complex Memory Instances

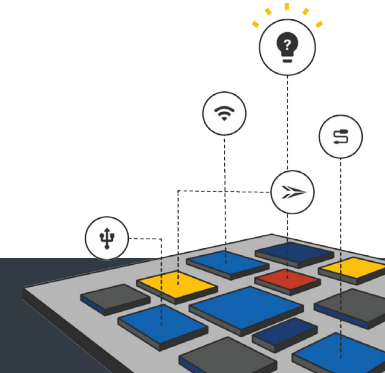
- Behavioral models of the RAMS are provided as part of the deliverable in the file:
  - sifive_coreip_E31_AHB_rtl_eval_v2p0/verilog/memories/ECoreIPSubsystemAllPortRAMTestHarness.*.rams.v
- For implementation it is necessary to generate memory instances which are specific to your foundry/process
  - Typically done using Memory Compilers from the foundry, or 3rd party IP providers

Name	Depth	Address Width (N_{addr})	Data Width (N_{data})	Write Mask Granularity (N_{part})	Description
data_arrays_0_ext	4096	12	32	8	DTIM data array
data_arrays_0_0_ext	2048	11	64	32	I-cache data array
tag_array_ext	256	8	38	19	I-cache tag array

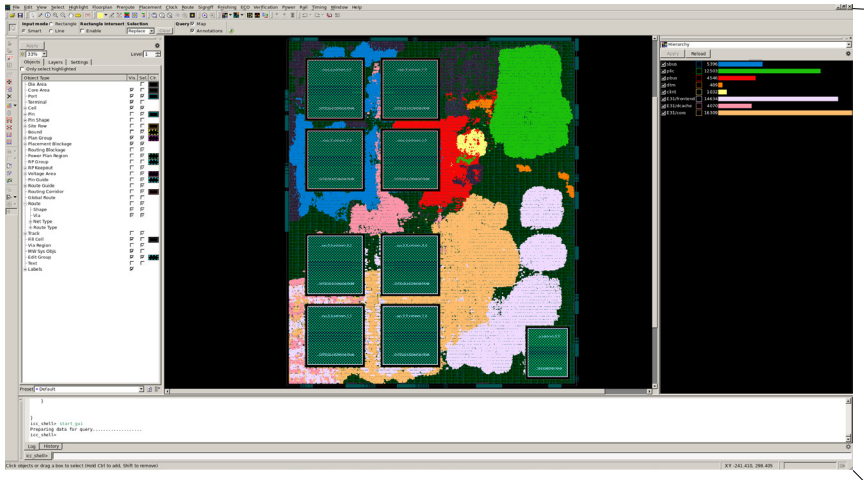
Table 3.1: SRAM Modules and Configuration

Name	Direction	Width	Description
RW0_clk	Input	1	Memory clock.
RW0_en	Input	1	Active-high signal indicating that the memory is being accessed. This may be used for clock gating.
RW0_addr	Input	N_{addr}	Address of access.
RW0_rdata	Output	N_{data}	Read data.
RW0_wmode	Input	1	Active-high signal indicating that the access is a write operation.
RW0_wdata	Input	N_{data}	Write data.
RW0_wmask	Input	N_{data}/N_{part}	Active-high write mask. Each bit controls whether or not the corresponding N_{part}-bit subword is written. This is present only in memories that require masked write functionality.

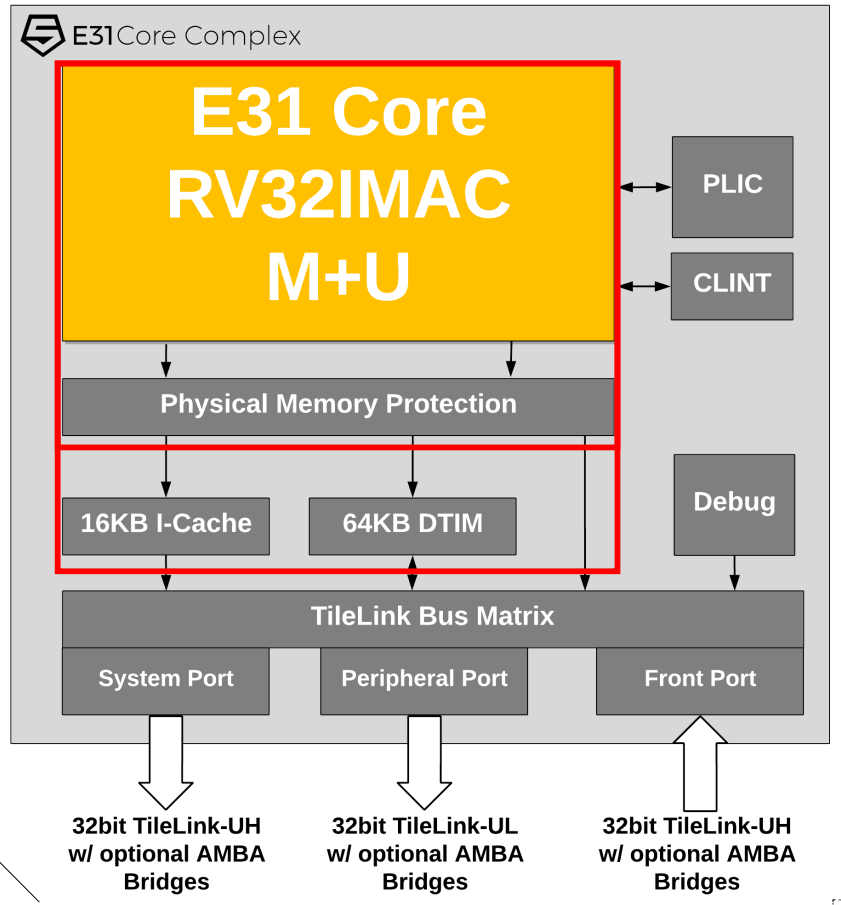
Table 3.2: SRAM Signals



E31 Synthesized Area Hierarchy

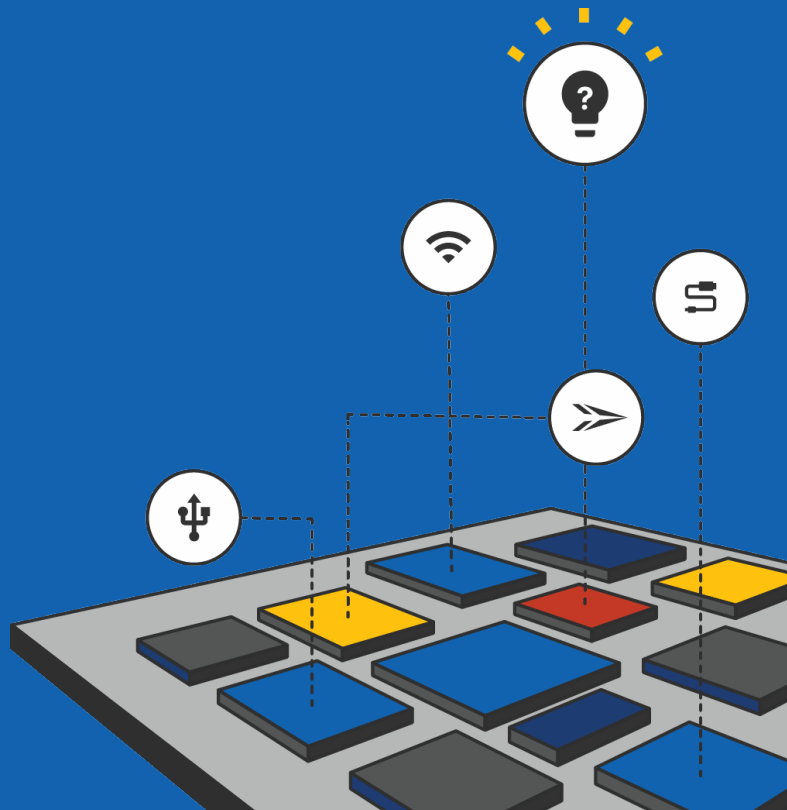


- *system/E31/core*
  - Core only (pipeline)
- *system/E31*
  - Core + L1 Instruction Cache + DTIM
- *system*
  - top-level

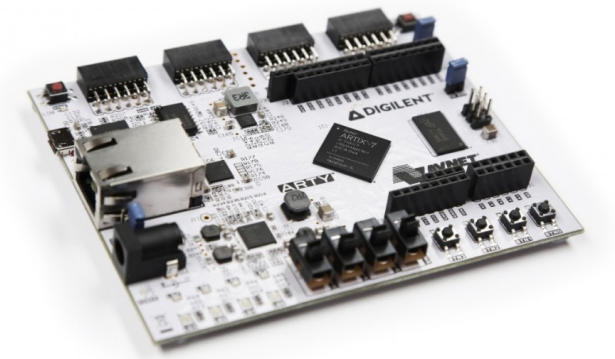


SiFive

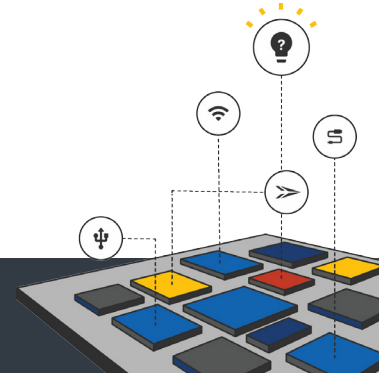
E31 FPGA Evaluation



Digilent Arty FPGA Platform

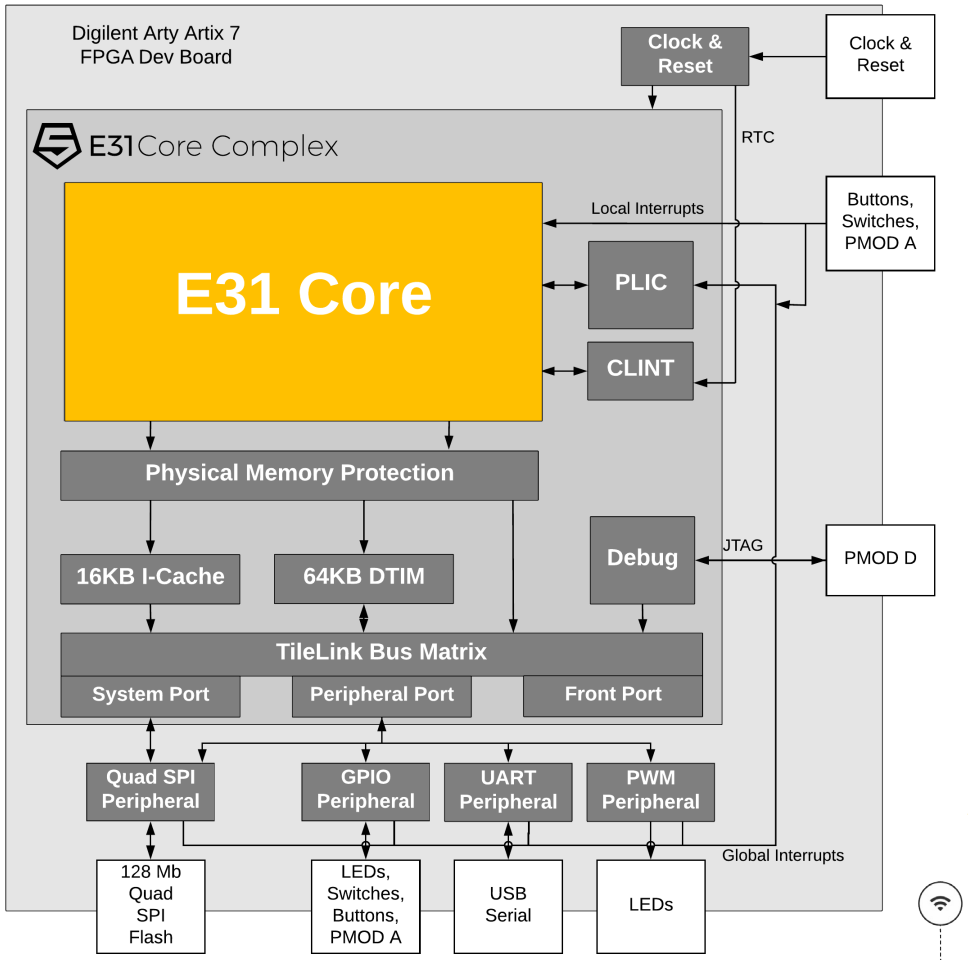


- Popular, Low cost, FPGA development board
  - Xilinx Artix-35T FPGA - 33,280 Xilinx logic cells
  - 16MB QSPI serial flash interface
  - USB-UART, buttons, switches, LEDs, etc…
- Our FPGA evaluations execute at 65MHz
  - Allows for fast Software execution
- Purchase directly from Digilent:
  - http://store.digilentinc.com/arty-a7-artix-7-fpga-development-board-for-makers-and-hobbyists/



E31 FPGA Evaluation Configuration

- SiFive Peripherals and Integration to Digilent Arty Platform
  - **QSPI** serial flash
  - **GPIO** to LEDs, Buttons, PMOD
  - **PWM** to LEDs
  - JTAG to PMOD-D
  - Buttons and Switches to Local and PLIC interrupt inputs
- Core Complex features
  - User Mode
  - 8 Hardware Breakpoints
  - 8 Region PMP
  - ITIM
  - Vectored Interrupts
  - Performance Counters



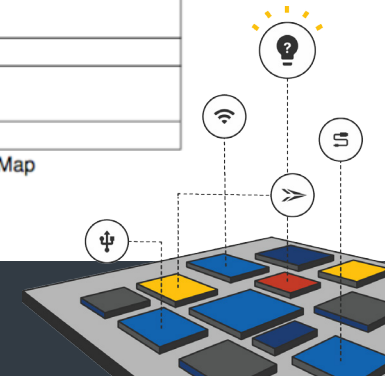
SiFive

FPGA Memory Map, Interrupts, and Pins

- Manual is available from developer dashboard
  - Memory Map
  - Pinout of peripherals to PMOD connector, LEDs, Buttons, switches
  - Interrupt numbers for connected devices

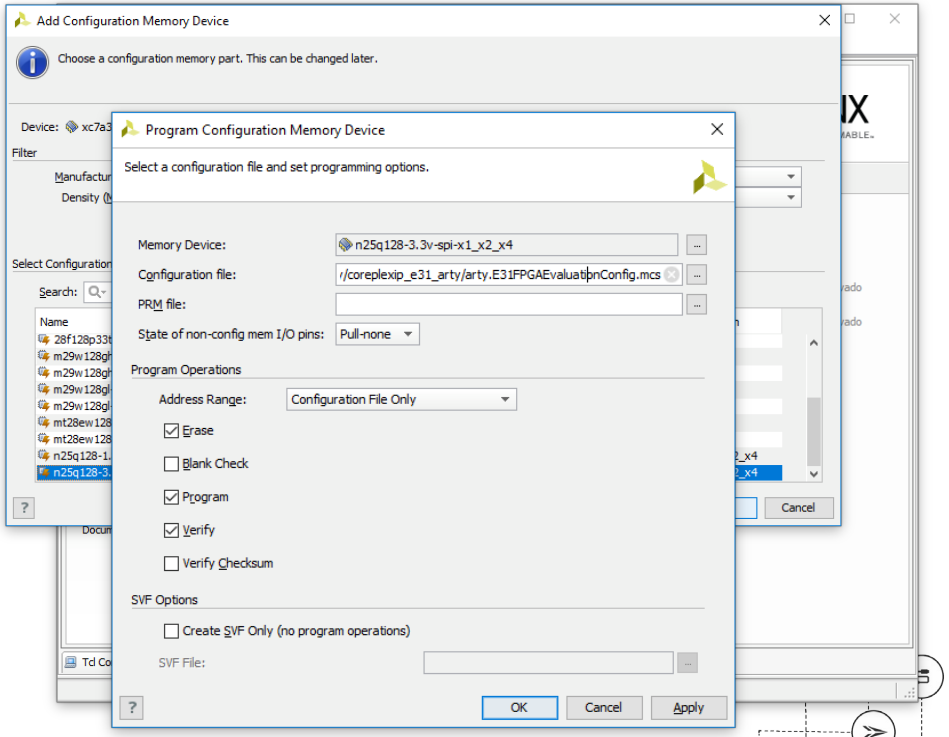
Base	Top	Description	Notes
0x0000_0000	0x0000_00FF	*Reserved*	Debug (4 KiB)
0x0000_0100		Halt Notification	
0x0000_0104		Start Notification	
0x0000_0108		Resume Notification	
0x0000_010C		Exception Notification	
0x0000_0110	0x0000_02FF	*Reserved*	
0x0000_0300	0x0000_03FF	Debug RAM (≤ .25 KiB)	
0x0000_0400	0x0000_07FF	Debug Flags (≤ 1 KiB)	
0x0000_0800	0x0000_0FFF	Debug ROM (≤ 2 KiB)	
0x0000_1000	0x01FF_FFFF	*Reserved*	
0x0200_0000	0x0200_FFFF	Core Complex-Local Interrupts (CLINT) (≤64 KiB)	on core complex Devices (224 MiB)
0x0201_0000	0x0BFF_FFFF	*Reserved*	
0x0C00_0000	0x0FFF_FFFF	Platform-Level Interrupt Control (PLIC) (64 MiB)	
0x1001_0000	0x1FFF_FFFF	*Reserved*	
0x2000_0000	0x2000_0FFF	UART Peripheral	Off Core Complex address space accessed via System and Peripheral busses
0x2000_1000	0x2000_1FFF	*Reserved*	
0x2000_2000	0x2000_2FFF	GPIO Peripheral (16 pins)	
0x2000_3000	0x2000_3FFF	*Reserved*	
0x2000_4000	0x2000_4FFF	Quad SPI Flash Control	
0x2000_5000	0x2000_5FFF	8-bit, 4-comparator PWM	
0x2000_3000	0x3FFF_FFFF	*Reserved*	
0x4000_0000	0x5FFF_FFFF	Memory Mapped Quad SPI Interface	
0x6000_0000	0x7FFF_FFFF	*Reserved*	
0x8000_0000	0x8000_FFFF	Data Tightly Integrated Memory (DTIM) (64kB)	
0x8000_4000	0xFFFF_FFFF	*Reserved*	

Table 7.1: E31/E51 Core Complex FPGA Eval Kit Physical Memory Map



Program FPGA with Vivado

- Download Vivado HLx Edition from Xilinx (free):
  - https://www.xilinx.com/products/design-tools/vivado.html
- From Vivado with the Arty board connected over USB
  - Open Hardware Manager
  - Tools – Auto Connect
  - Tools - Add Configuration Memory Device
  - Select Micron n25q128-3.3V
  - Select the Configuration File (mcs file downloaded from SiFive)
- Hit the PROG button on the Arty to reboot and load the new image

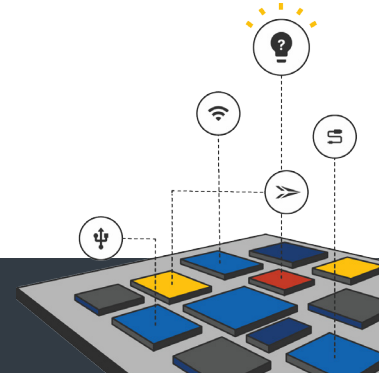


Connecting the FPGA to the Debugger

- 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
  - 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

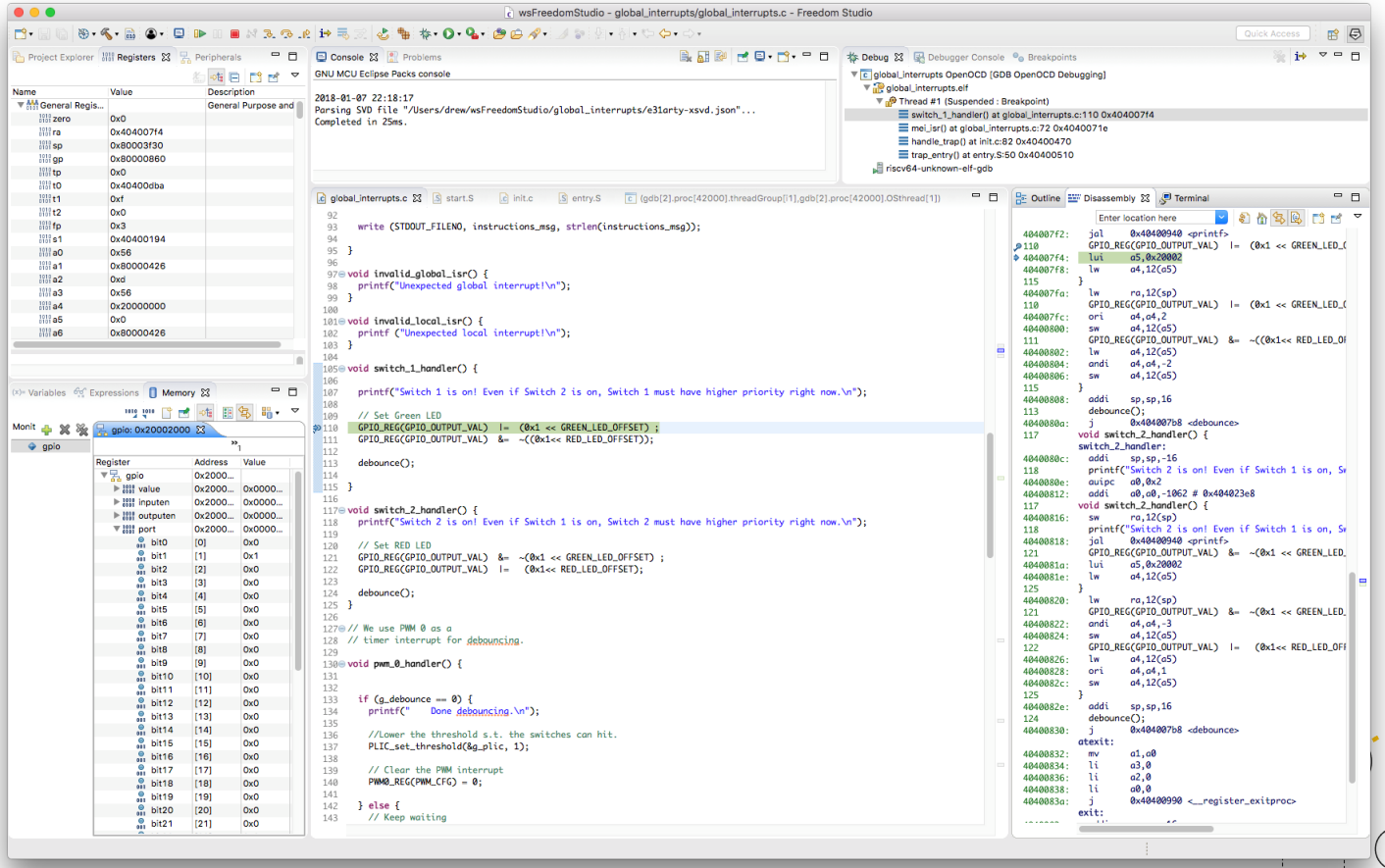
ARM 20 Pin Header



SiFive

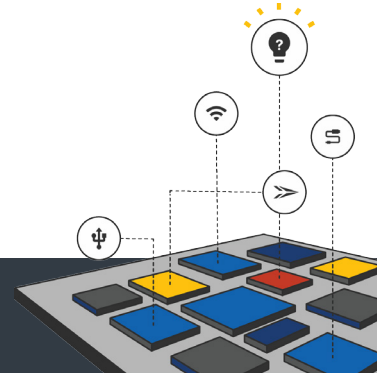
Software Development with Freedom Studio

- Freedom Studio
  - Eclipse + CDT + GNU MCU Eclipse
  - Bundled Toolchain and OpenOCD Binaries
  - Examples for SiFive platforms



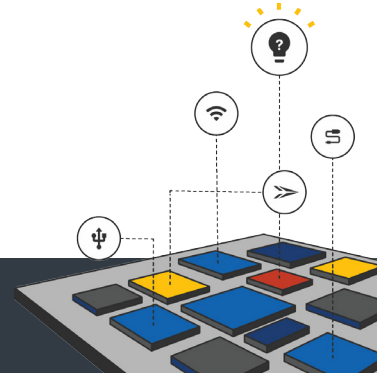
Freedom Studio – Set Up

- Download and Extract to desired location
  - https://www.sifive.com/products/tools/
- If on Windows, install platform drivers located in:
  - *FreedomStudio/SiFive/Drivers*
- That's It!



Freedom Studio - Demo

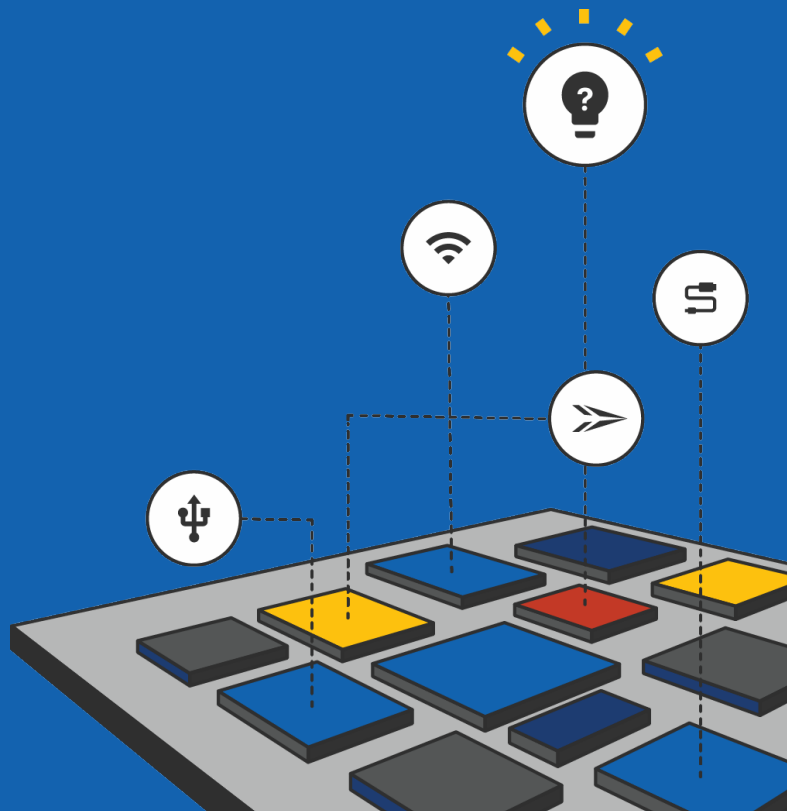
- Importing Examples
- Freedom Studio SiFive Perspective
- Programming and Debugging the E31 Arty FPGA Platform
  - Dhrystone Demo – use the debugger to change the number of iterations



SiFive

Questions

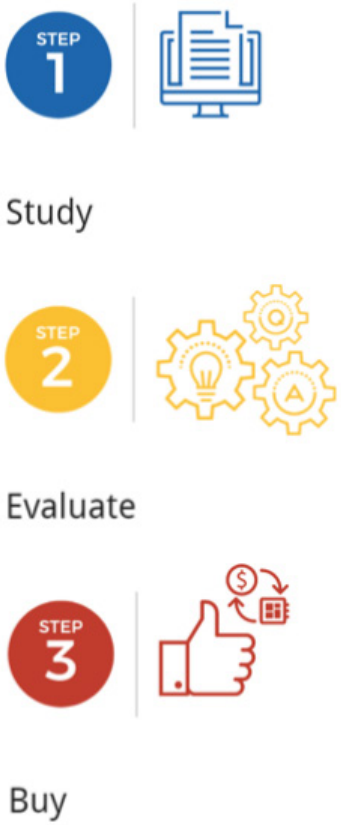
info@sifive.com



3 Part Webinar Series

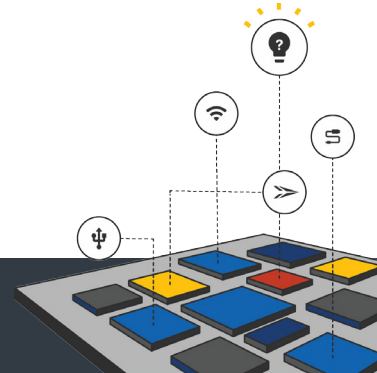
Webinar Recordings: https://info.sifive.com/risc-v-webinar

- RISC-V 101
  - The Fundamentals of RISC-V architecture
- Introduction to SiFive RISC-V Core IP
  - Introduction to the SiFive E31 and E51 Core Complexes
- Getting Started with SiFive RISC-V Core IP



Resources

- https://www.sifive.com/
  - RISC-V Core IP and Development Boards
  - RISC-V Tools
  - SiFive Youtube Channel:
    - https://www.youtube.com/channel/UCqpdhncf4nxTfy0QZh1YWLQ
- https://riscv.org/
  - RISC-V Specifications
  - Links to the RISC-V mailing lists
  - Workshop proceedings
- GitHub
  - https://github.com/sifive/
  - https://github.com/riscv/




SiFive