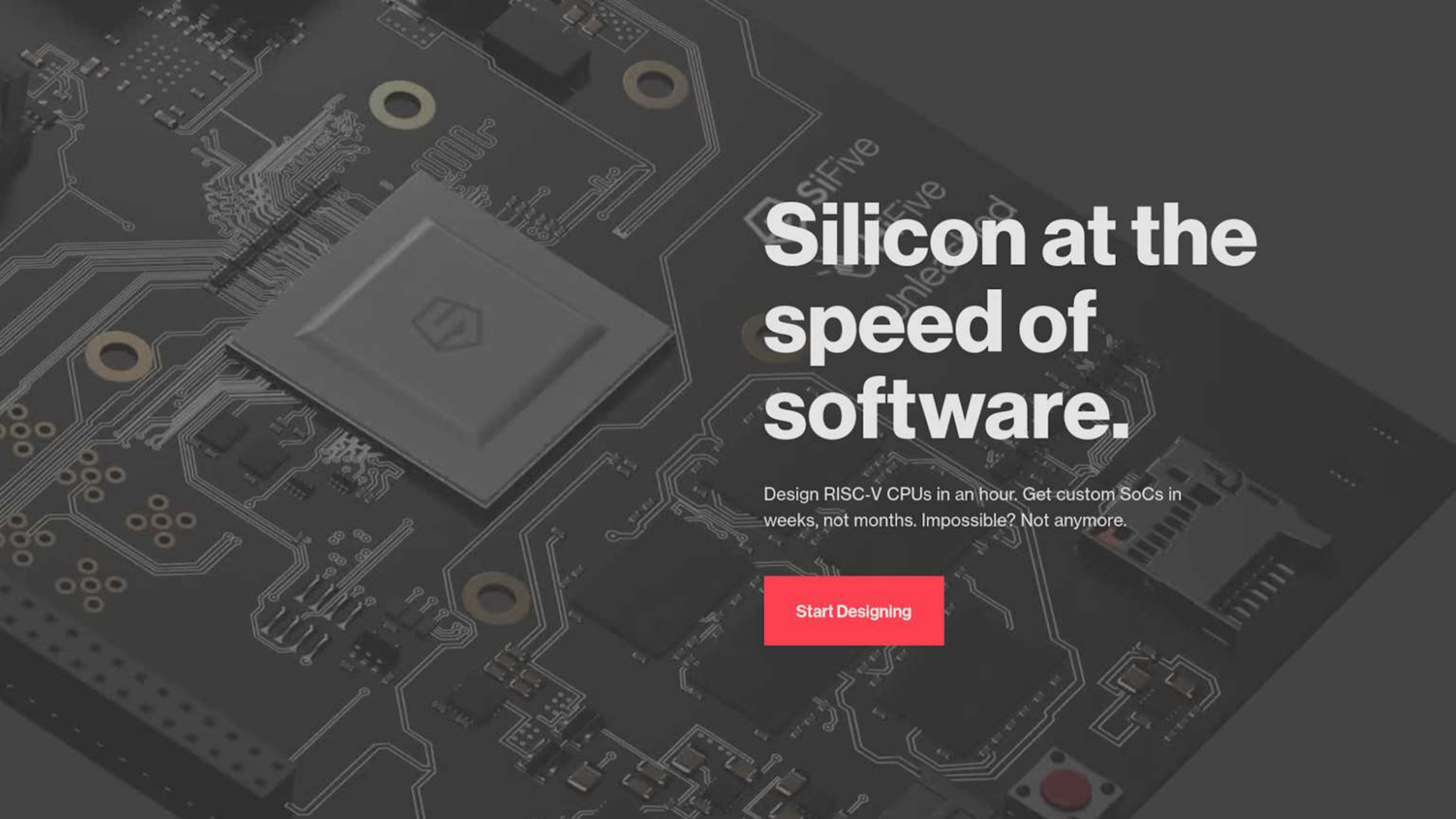
SiFive

SiFive Core Designer

From Custom CPU to Hello World in 30 Minutes



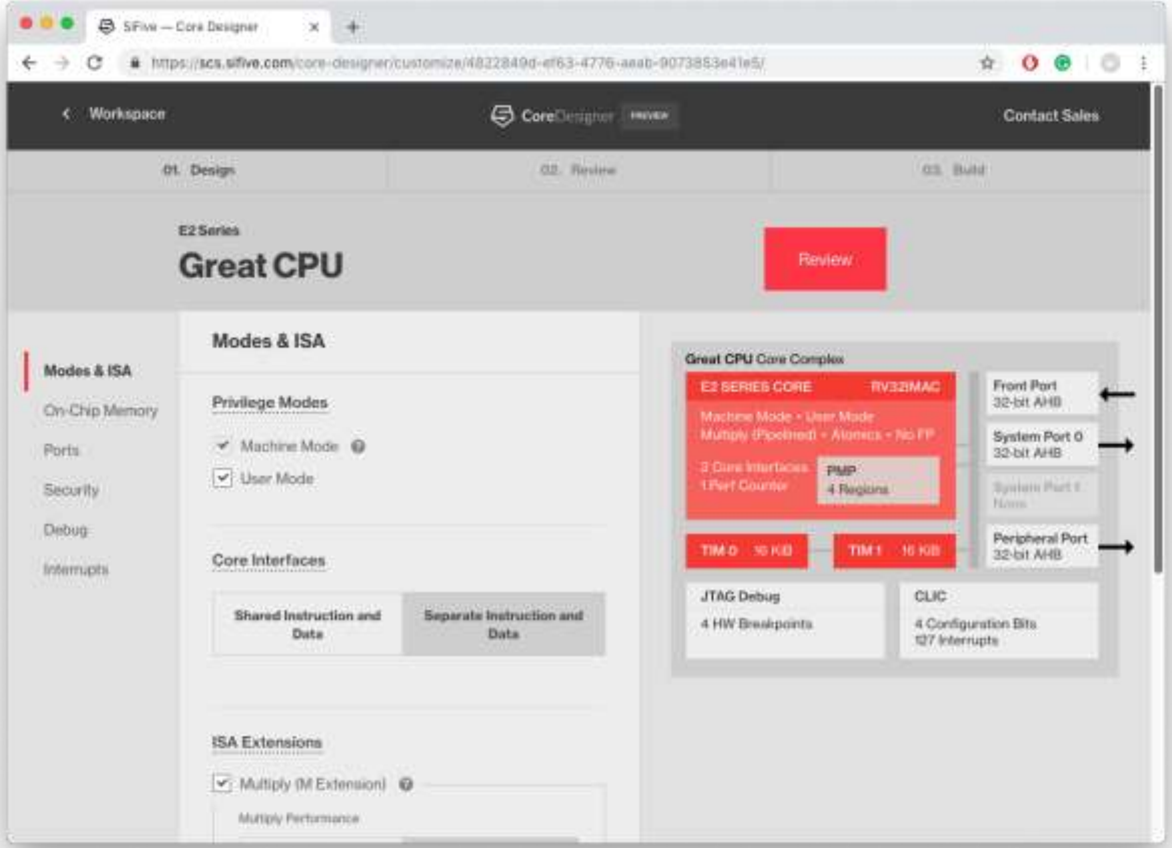
Silicon at the speed of software.

Design RISC-V CPUs in an hour. Get custom SoCs in weeks, not months. Impossible? Not anymore.

Start Designing

SiFive Core Designer

Your interface to SiFive RISC-V Core IP



Modes & ISA

Privilege Modes

✓ Machine Mode

☑ User Mode

Core Interfaces

Shared Instruction and Data | Separate Instruction and Data

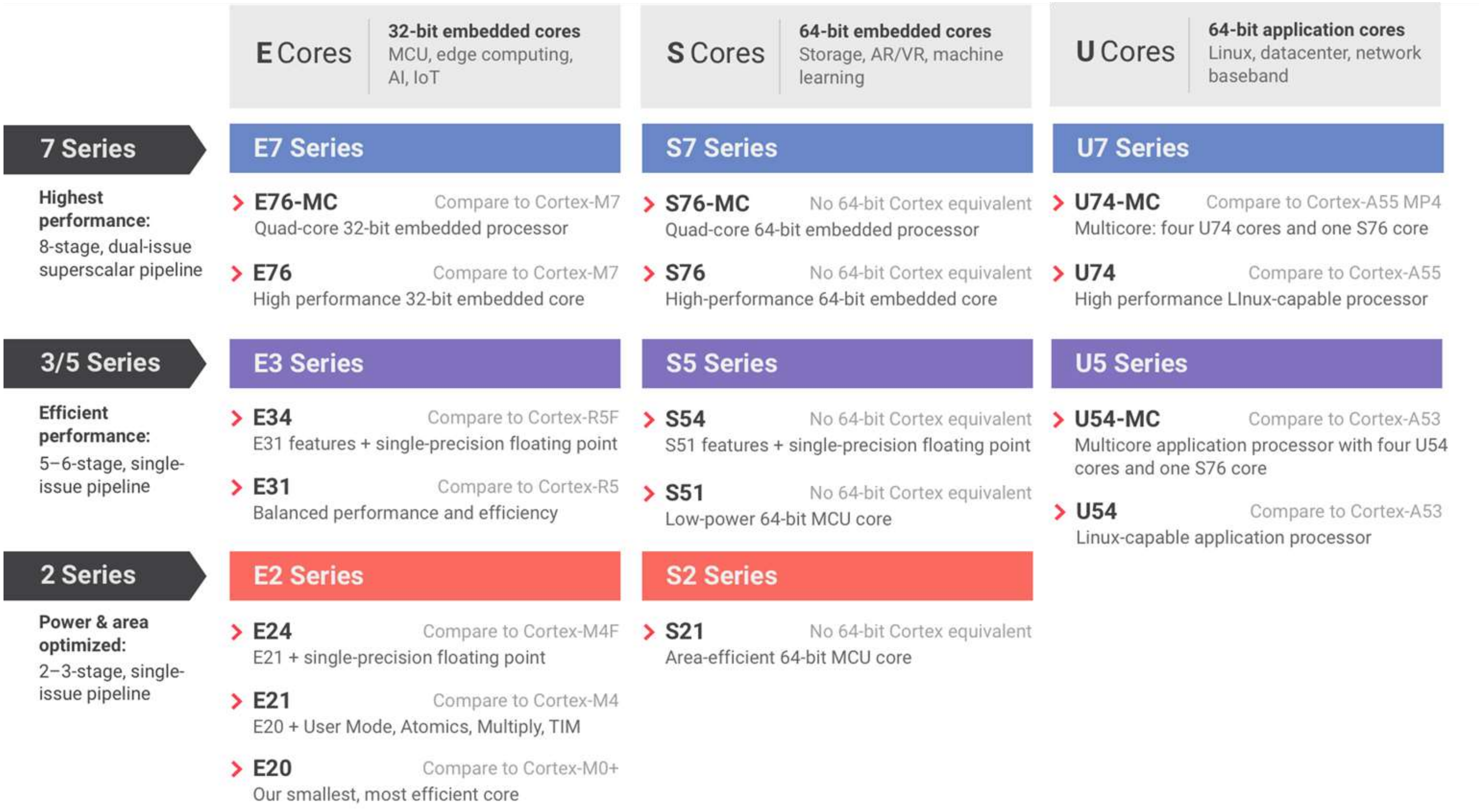
ISA Extensions

☑ Multiply (M Extension)

Multiply Performance

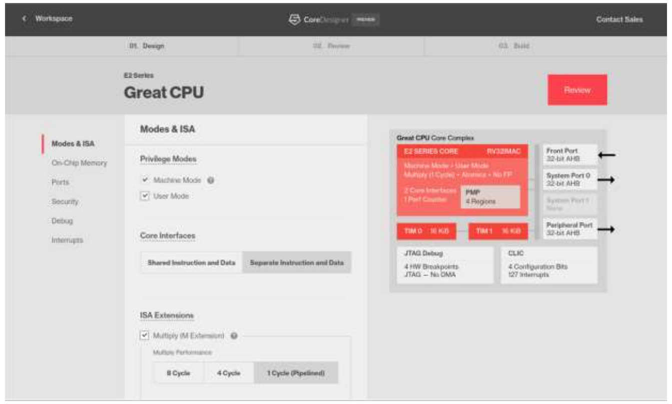
- All SiFive Core IP is configured and delivered via the SiFive Core Designer Web Portal
  - Simple, Easy to Use, Web Interface
- **Release Candidates** are generated with click of a button and available from the Workspace
- **Release Candidates** contain
  - RTL matching the configuration, including a testbench, and other collateral needed to realize the design
  - Documentation specific to the design
  - Customized bare-metal BSP for easy integration into SiFive's SDKs
  - FPGA bitstreams for common FPGA development boards for easy software benchmarking of the RC

In-house IP: SiFive RISC-V Core IP Product Offering



	E Cores **32-bit embedded cores** MCU, edge computing, AI, IoT	**S Cores** **64-bit embedded cores** Storage, AR/VR, machine learning	**U Cores** **64-bit application cores** Linux, datacenter, network baseband
7 Series	**E7 Series**	**S7 Series**	**U7 Series**
Highest performance: 8-stage, dual-issue superscalar pipeline	**E76-MC** Compare to Cortex-M7 Quad-core 32-bit embedded processor **E76** Compare to Cortex-M7 High performance 32-bit embedded core	**S76-MC** No 64-bit Cortex equivalent Quad-core 64-bit embedded processor **S76** No 64-bit Cortex equivalent High-performance 64-bit embedded core	**U74-MC** Compare to Cortex-A55 MP4 Multicore: four U74 cores and one S76 core **U74** Compare to Cortex-A55 High performance LInux-capable processor
3/5 Series	**E3 Series**	**S5 Series**	**U5 Series**
Efficient performance: 5–6-stage, single-issue pipeline	**E34** Compare to Cortex-R5F E31 features + single-precision floating point **E31** Compare to Cortex-R5 Balanced performance and efficiency	**S54** No 64-bit Cortex equivalent S51 features + single-precision floating point **S51** No 64-bit Cortex equivalent Low-power 64-bit MCU core	**U54-MC** Compare to Cortex-A53 Multicore application processor with four U54 cores and one S76 core **U54** Compare to Cortex-A53 Linux-capable application processor
2 Series	**E2 Series**	**S2 Series**	
Power & area optimized: 2–3-stage, single-issue pipeline	**E24** Compare to Cortex-M4F E21 + single-precision floating point **E21** Compare to Cortex-M4 E20 + User Mode, Atomics, Multiply, TIM **E20** Compare to Cortex-M0+ Our smallest, most efficient core	**S21** No 64-bit Cortex equivalent Area-efficient 64-bit MCU core	

From Custom CPU to Hello World in 30 minutes



E2 Series

Great CPU

Modes & ISA

Privilege Modes

- ☑ Machine Mode
- ☑ User Mode

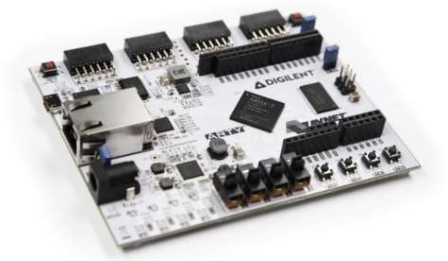
Core Interfaces: Shared Instruction and Data | Separate Instruction and Data

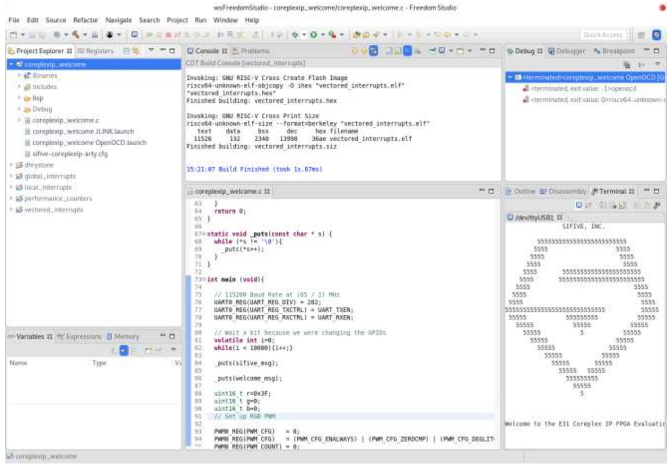
ISA Extensions

- ☑ Multiply (M Extension)

Multiply Performance: 8 Cycle | 4 Cycle | 1 Cycle (Pipelined)

E2 SERIES CORE RV32IMAC
Machine Mode + User Mode
Multiply (1 Cycle) + Atomics + No FP
2 Core Interfaces
1 Perf Counter
PMP
4 Regions
TIM 0 16 KiB
TIM 1 16 KiB
JTAG Debug
4 HW Breakpoints
JTAG — No DMA
CLIC
4 Configuration Bits
127 Interrupts





CDT Build Console [vectored_interrupts]

```
Invoking: GNU RISC-V Cross Create Flash Image
riscv64-unknown-elf-objcopy -O ihex "vectored_interrupts.elf"
"vectored_interrupts.hex"
Finished building: vectored_interrupts.hex

Invoking: GNU RISC-V Cross Print Size
riscv64-unknown-elf-size --format=berkeley "vectored_interrupts.elf"
   text    data     bss     dec     hex filename
  11526     132    2340   13998    36ae vectored_interrupts.elf
Finished building: vectored_interrupts.siz

15:21:07 Build Finished (took 1s.67ms)
```

```
  }
  return 0;
}

static void _puts(const char * s) {
  while (*s != '\0'){
    _putc(*s++);
  }
}

int main (void){

  // 115200 Baud Rate at (65 / 2) MHz
  UART0_REG(UART_REG_DIV) = 282;
  UART0_REG(UART_REG_TXCTRL) = UART_TXEN;
  UART0_REG(UART_REG_RXCTRL) = UART_RXEN;

  // Wait a bit because we were changing the GPIOs
  volatile int i=0;
  while(i < 10000){i++;}

  _puts(sifive_msg);

  _puts(welcome_msg);

  uint16_t r=0x3F;
  uint16_t g=0;
  uint16_t b=0;
  // Set up RGB PWM

  PWM0_REG(PWM_CFG)   = 0;
  PWM0_REG(PWM_CFG)   = (PWM_CFG_ENALWAYS) | (PWM_CFG_ZEROCMP) | (PWM_CFG_DEGLIT
  PWM0_REG(PWM_COUNT) = 0;
```

```
SIFIVE, INC.
Welcome to the E31 Coreplex IP FPGA Evaluati
```

Step 1

Configure a custom SiFive RISC-V Core using SiFive Core Designer

Step 2

Use the FPGA bitstream from Step 1 to program a Digilent Arty FPGA board with the configured CPU

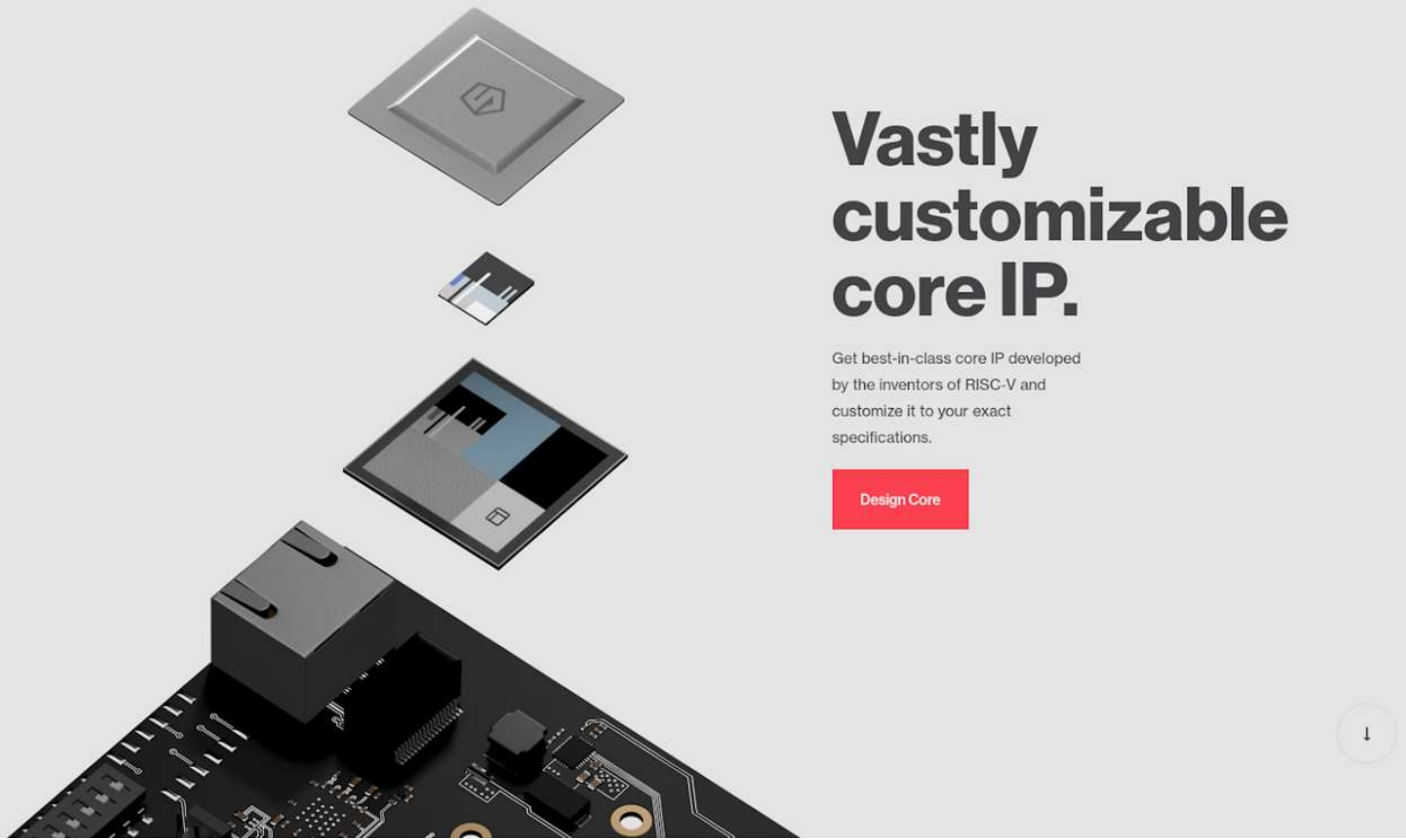
Step 3

Use Freedom Studio and the SiFive SDK to program and run Hello World

SiFive

Step 1 - Configure the Core using SiFive Core Designer

https://www.sifive.com/core-designer



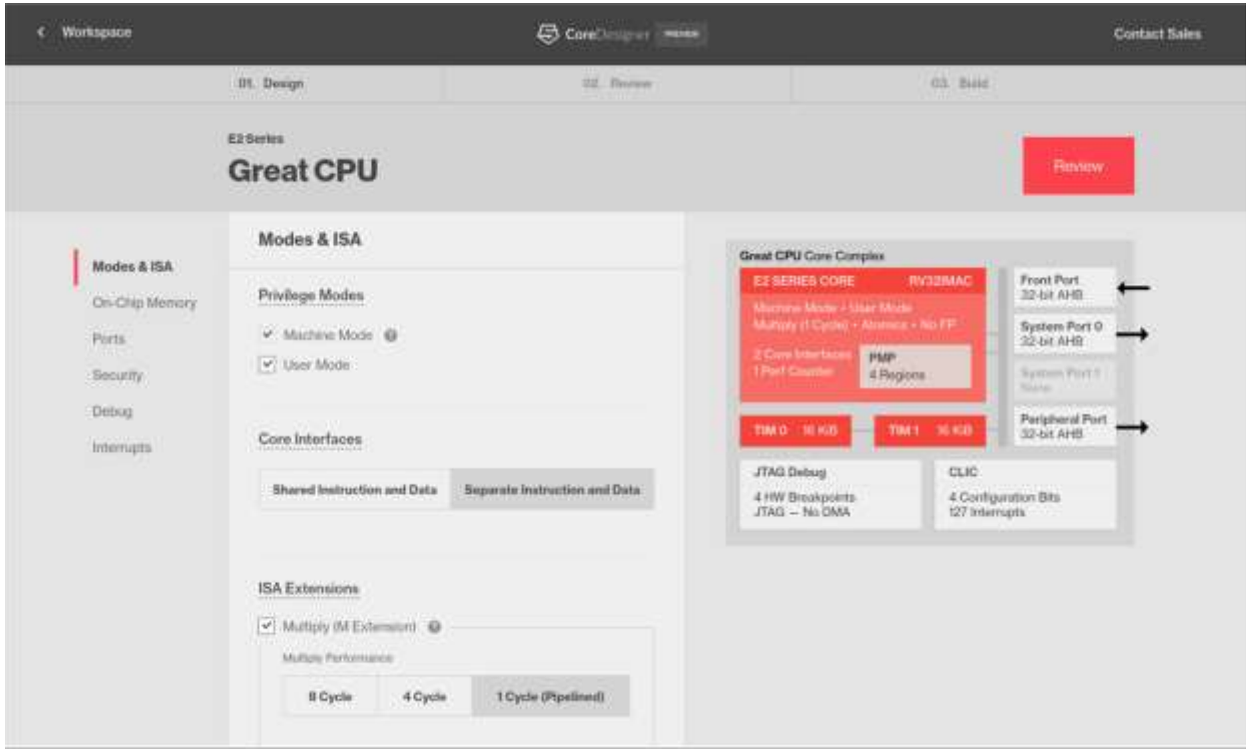
Vastly customizable core IP.

Get best-in-class core IP developed by the inventors of RISC-V and customize it to your exact specifications.

Configure a SiFive RISC-V CPU

SiFive Core Designer

- **Web Interface to Configure SiFive Core IP**
  - No Complex EDA tools or scripting languages to learn
- **What is configurable**
  - ISA, Performance levels, Modes, Ports, Interrupts, Security, Debug, and much more!
- **What is the output**
  - **Verilog** RTL and supporting collateral, an **FPGA bitstream**, **software**, and **documentation**



E2 Series

Great CPU

Modes & ISA

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)]

Core Designer UI Walkthrough

Go to the SiFive website and click “Start Designing”

- https://www.sifive.com/

Choose a Core Series to start from

- Start from a pre-configured Standard Core
- Or start from scratch

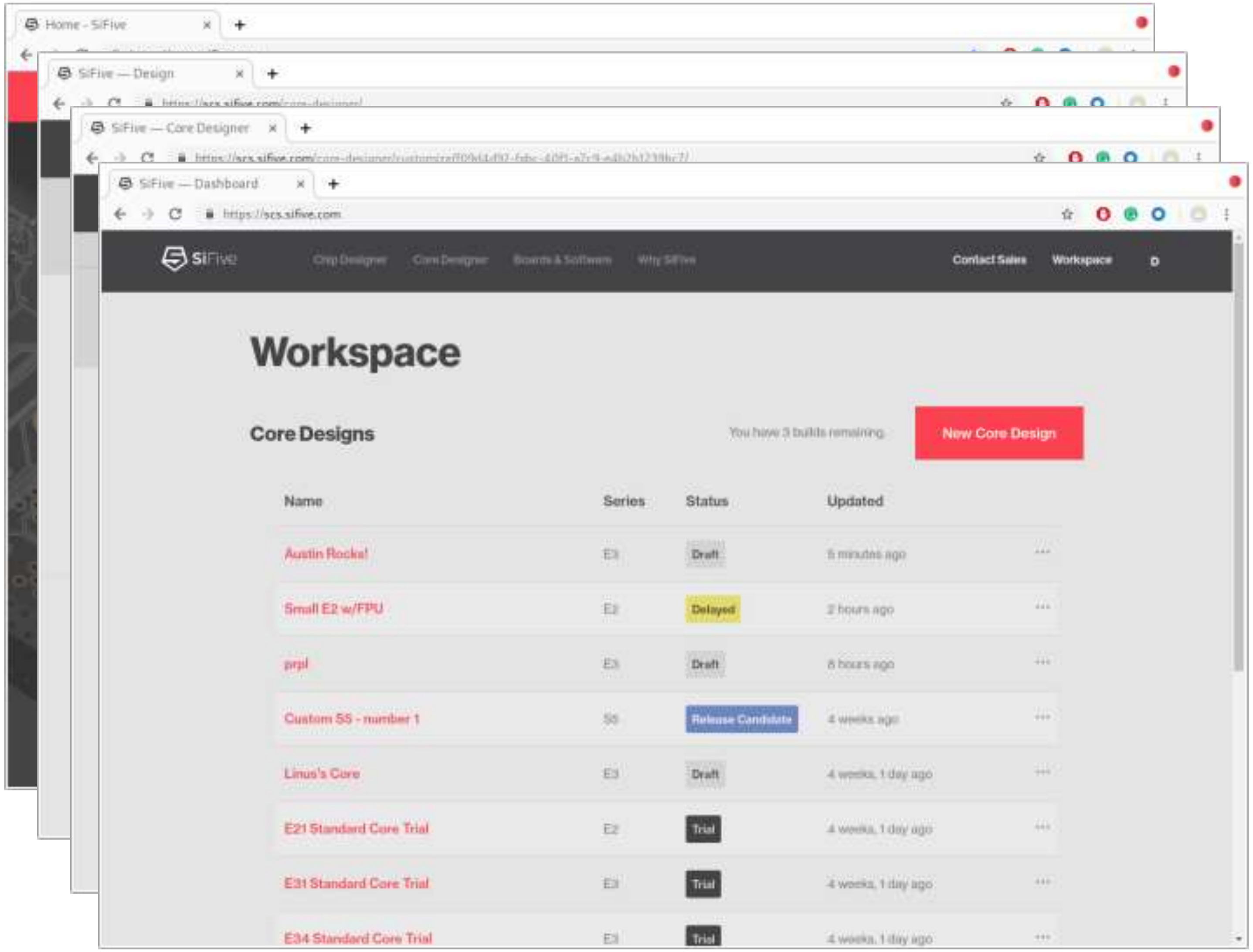
Name the Design and Start Clicking!

- Change performance levels, memory maps, Privilege modes, Instructions Sets, Security, Debug, etc…

Click Review and then Build

- Launches SiFive’s cloud based infrastructure to render and verify the design

Download from your SiFive Workspace



Workspace

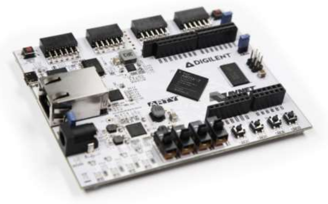
Core Designs

You have 3 builds remaining

New Core Design

Name	Series	Status	Updated
Austin Rocks!	E3	Draft	5 minutes ago
Small E2 w/FPU	E2	Delayed	2 hours ago
prpl	E3	Draft	8 hours ago
Custom S5 - number 1	S5	Release Candidate	4 weeks ago
Linus's Core	E3	Draft	4 weeks, 1 day ago
E21 Standard Core Trial	E2	Trial	4 weeks, 1 day ago
E31 Standard Core Trial	E3	Trial	4 weeks, 1 day ago
E34 Standard Core Trial	E3	Trial	4 weeks, 1 day ago

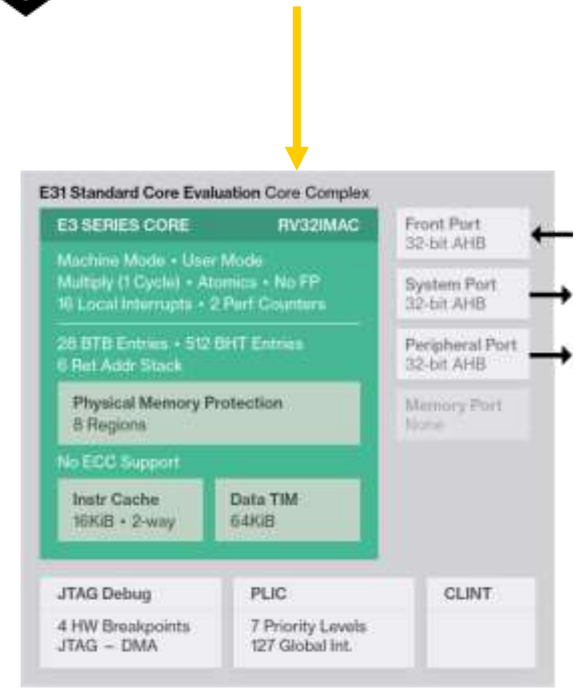
Too Many Choices? Start with a Standard Core



FPGA Evaluations

RTL Evaluations

E3 Series



E31 Standard Core Evaluation Core Complex

E3 SERIES CORE RV32IMAC

Machine Mode • User Mode
Multiply (1 Cycle) • Atomics • No FP
16 Local Interrupts • 2 Perf Counters

28 BTB Entries • 512 BHT Entries
6 Ret Addr Stack

Physical Memory Protection
8 Regions

No ECC Support

Instr Cache
16KiB • 2-way

Data TIM
64KiB

Front Port
32-bit AHB

System Port
32-bit AHB

Peripheral Port
32-bit AHB

Memory Port
None

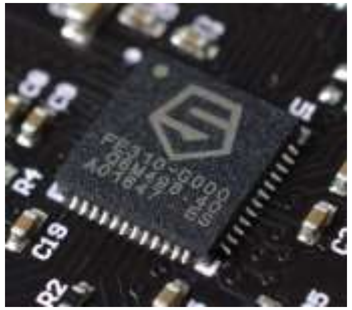
JTAG Debug
4 HW Breakpoints
JTAG – DMA

PLIC
7 Priority Levels
127 Global Int.

CLINT

E31 Standard Core Definition

Benchmarks



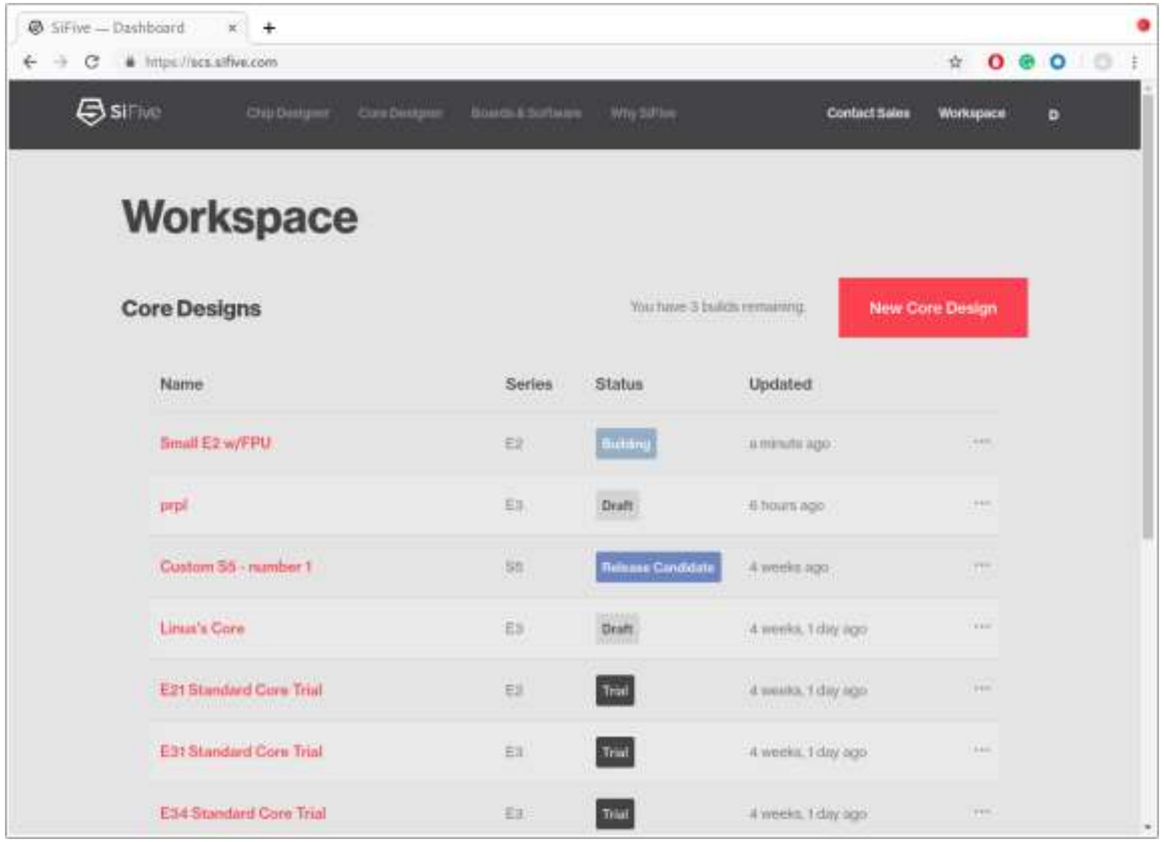
FE310 Silicon

Standard Core RTL and FPGA Evaluations are Available with a click-through License

SiFive

Step 2 - Download the Deliverables and Program the FPGA

Download the Deliverables from your SCD Workspace



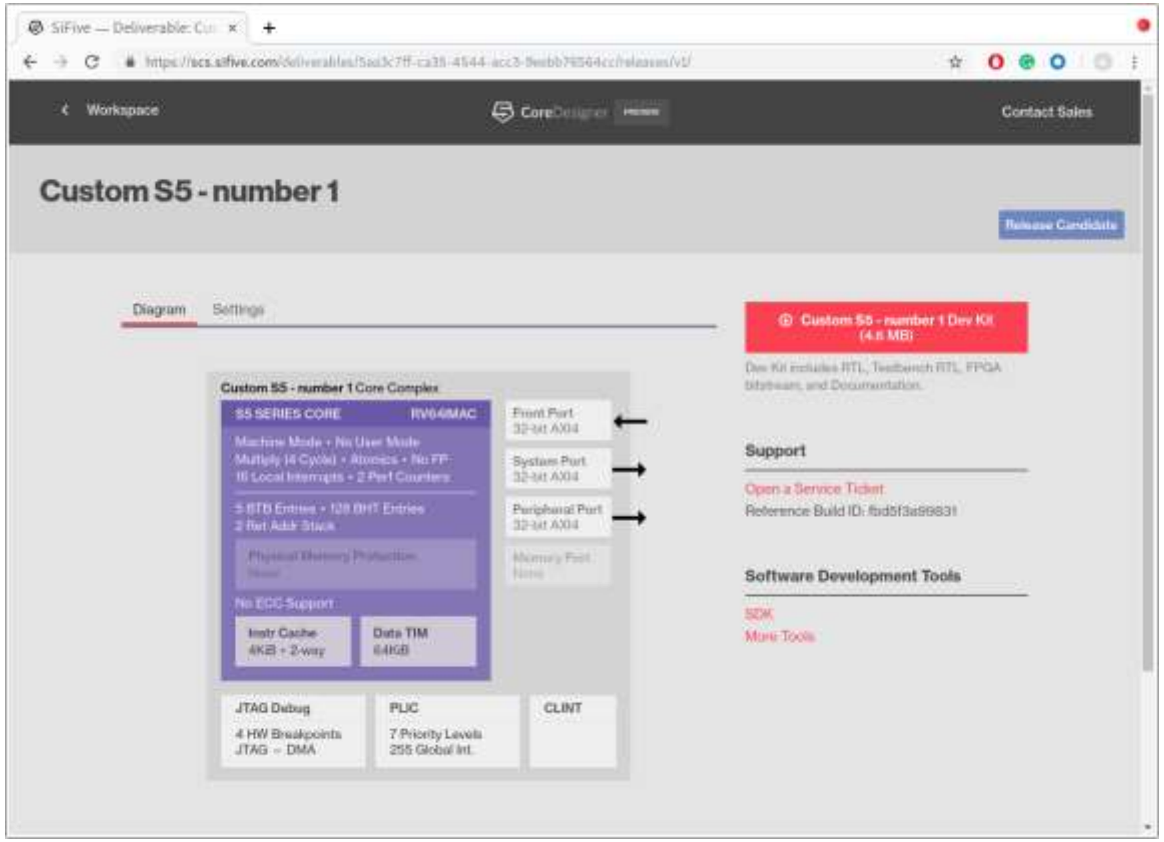
Workspace

Core Designs

You have 3 builds remaining.

New Core Design

Name	Series	Status	Updated
Small E2 w/FPU	E2	Building	a minute ago
prpl	E3	Draft	6 hours ago
Custom S5 - number 1	S5	Release Candidate	4 weeks ago
Linus's Core	E3	Draft	4 weeks, 1 day ago
E21 Standard Core Trial	E2	Trial	4 weeks, 1 day ago
E31 Standard Core Trial	E3	Trial	4 weeks, 1 day ago
E34 Standard Core Trial	E3	Trial	4 weeks, 1 day ago


SiFive — Deliverable: Cu
https://scs.sifive.com/deliverables/5ad3c7ff-ca35-4544-acc3-9eebb76564cc/releases/v1/
Workspace
CoreDesigner
Contact Sales
Diagram
Settings

Custom S5 - number 1

Release Candidate

Custom S5 - number 1 Core Complex

S5 SERIES CORE RV64IMAC

Machine Mode • No User Mode
Multiply (4 Cycle) • Atomics • No FP
16 Local Interrupts • 2 Perf Counters

5 BTB Entries • 128 BHT Entries
2 Ret Addr Stack

Physical Memory Protection
None

No ECC Support

Instr Cache
4KB • 2-way

Data TIM
64KB

Front Port
32-bit AXI4

System Port
32-bit AXI4

Peripheral Port
32-bit AXI4

Memory Port
None

JTAG Debug
4 HW Breakpoints
JTAG → DMA

PLIC
7 Priority Levels
255 Global Int.

CLINT

Custom S5 - number 1 Dev Kit (4.8 MB)

Dev Kit includes RTL, Testbench RTL, FPGA bitstream, and Documentation.

Support

Open a Service Ticket

Reference Build ID: fbd5f3a99831

Software Development Tools

SDK

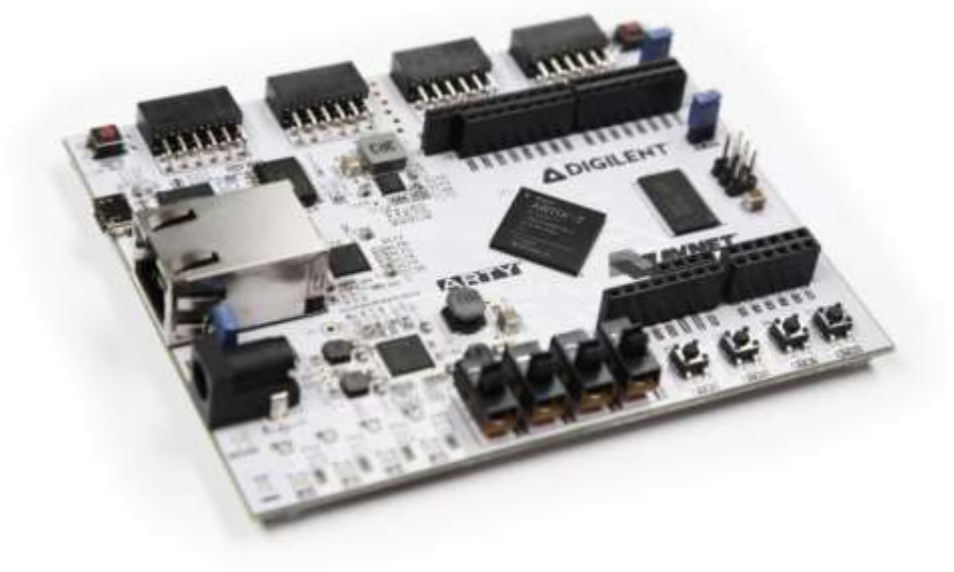
More Tools

Deploy the bitstream to the FPGA

1. **Purchase a Digilent Arty**
   https://store.digilentinc.com/arty-a7-artix-7-fpga-development-board-for-makers-and-hobbyists/

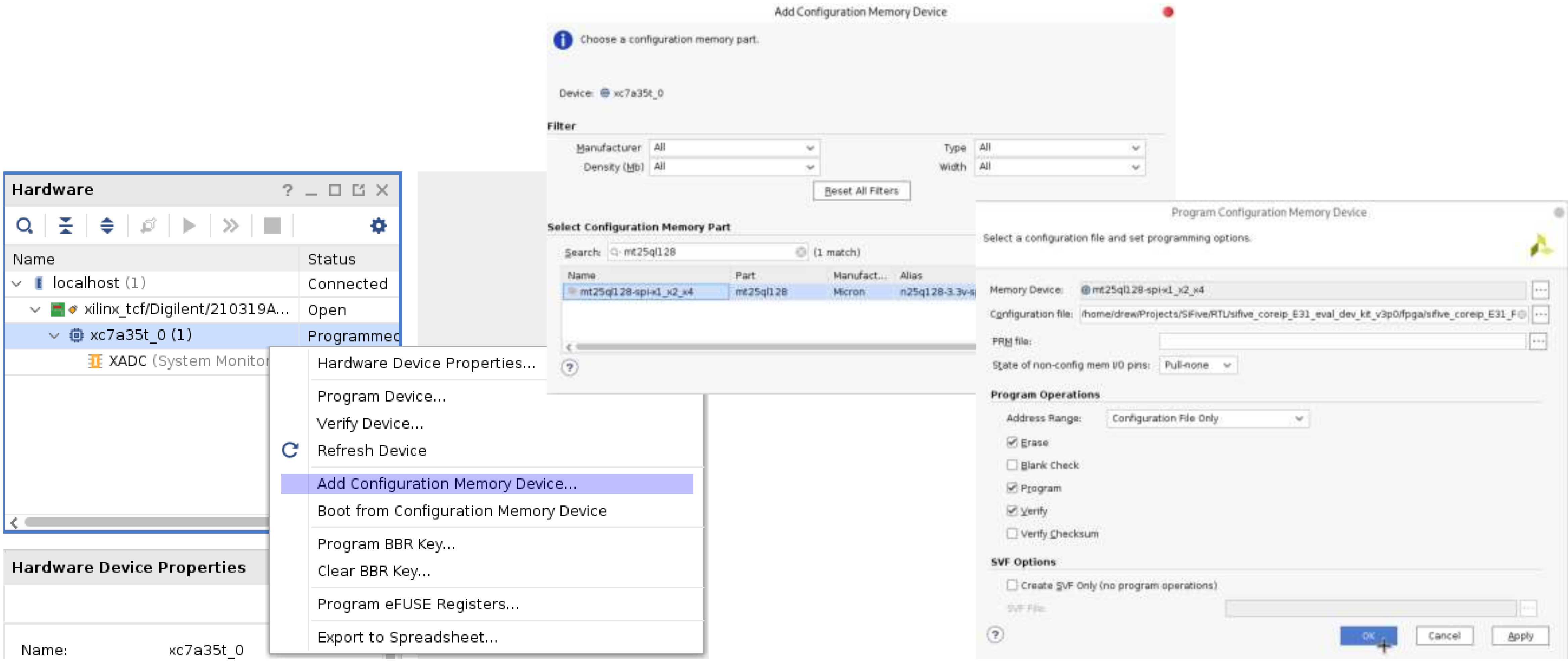
1. **Download Xilinx Vivado 2018.3 (Warning, HUGE 19GB)**
   https://www.xilinx.com/products/design-tools/vivado.html

1. **Open Vivado’s Hardware Manager Tool**



SiFive

Deploy the bitstream to the FPGA In Pictures



Hardware

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210319A...	Open
xc7a35t_0 (1)	Programmed
XADC (System Monitor	

Hardware Device Properties

Name: xc7a35t_0

Hardware Device Properties...
Program Device...
Verify Device...
Refresh Device
Add Configuration Memory Device...
Boot from Configuration Memory Device
Program BBR Key...
Clear BBR Key...
Program eFUSE Registers...
Export to Spreadsheet...

Add Configuration Memory Device

Choose a configuration memory part.

Device: xc7a35t_0

Filter

Manufacturer: All

Type: All

Density (Mb): All

Width: All

Select Configuration Memory Part

Search: mt25ql128 (1 match)

Name	Part	Manufact...	Alias
mt25ql128-spi-x1_x2_x4	mt25ql128	Micron	n25q128-3.3v-s

Program Configuration Memory Device

Select a configuration file and set programming options.

Memory Device: mt25ql128-spi-x1_x2_x4

Configuration file: /home/drew/Projects/SiFive/RTL/sifive_coreip_E31_eval_dev_kit_v3p0/fpga/sifive_coreip_E31_F

PRM file:

State of non-config mem I/O pins: Pull-none

Program Operations

Address Range: Configuration File Only

☑ Erase

☐ Blank Check

☑ Program

☑ Verify

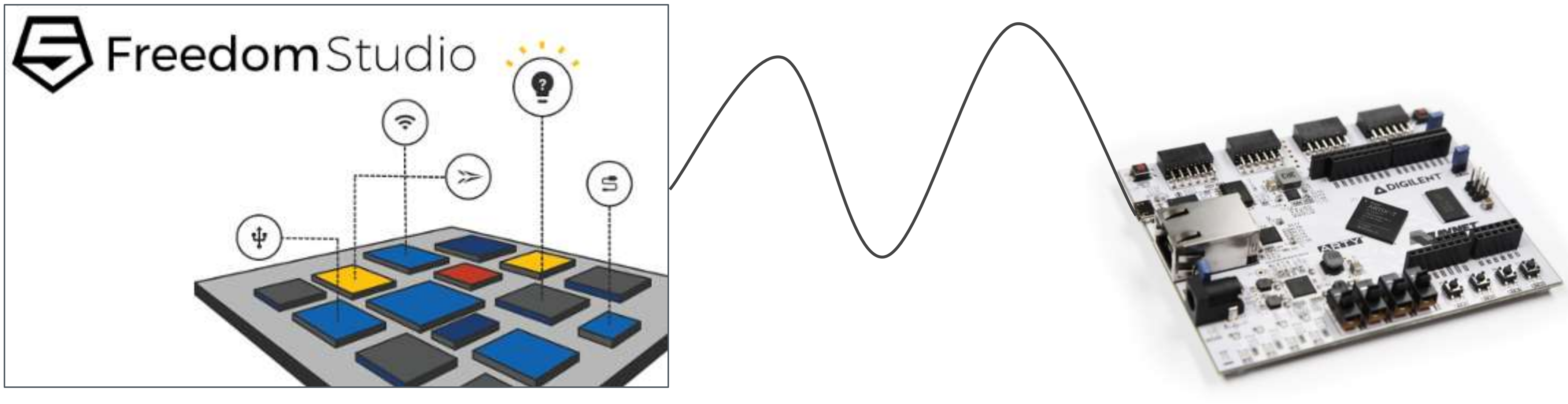
☐ Verify Checksum

SVF Options

☐ Create SVF Only (no program operations)

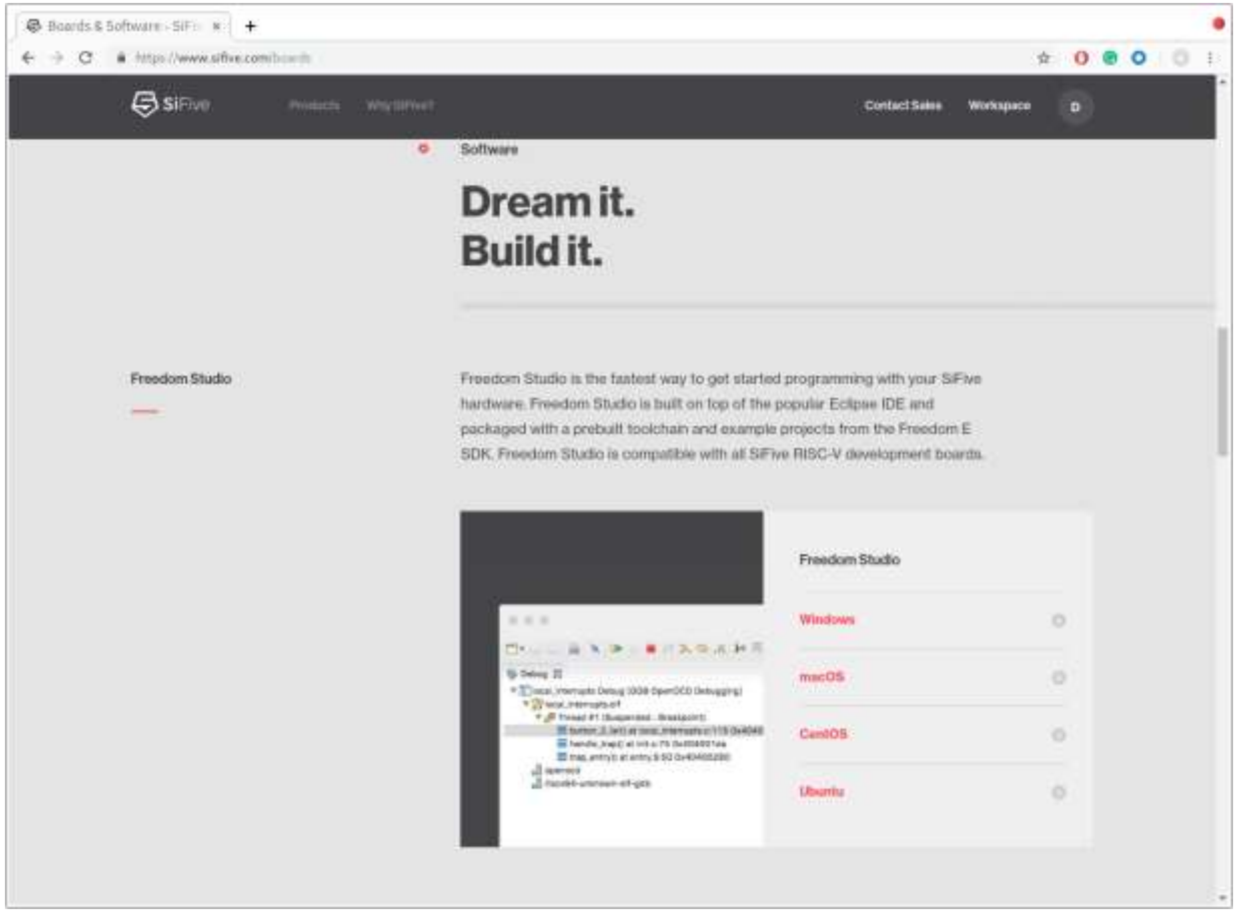
SVF File:

Coming Soon
The ability to flash Arty boards directly from Freedom Studio



Step 3 - Hello World!

Download Freedom Studio


Boards & Software - SiFive
https://www.sifive.com/boards
SiFive
Products
Why SiFive?
Contact Sales
Workspace
D
Software
Debug
local_interrupts Debug (GDB OpenOCD Debugging)
local_interrupts.elf
Thread #1 (Suspended : Breakpoint)
openocd
riscv64-unknown-elf-gdb
Windows
macOS
CentOS
Ubuntu

Software

Dream it.
Build it.

Freedom Studio

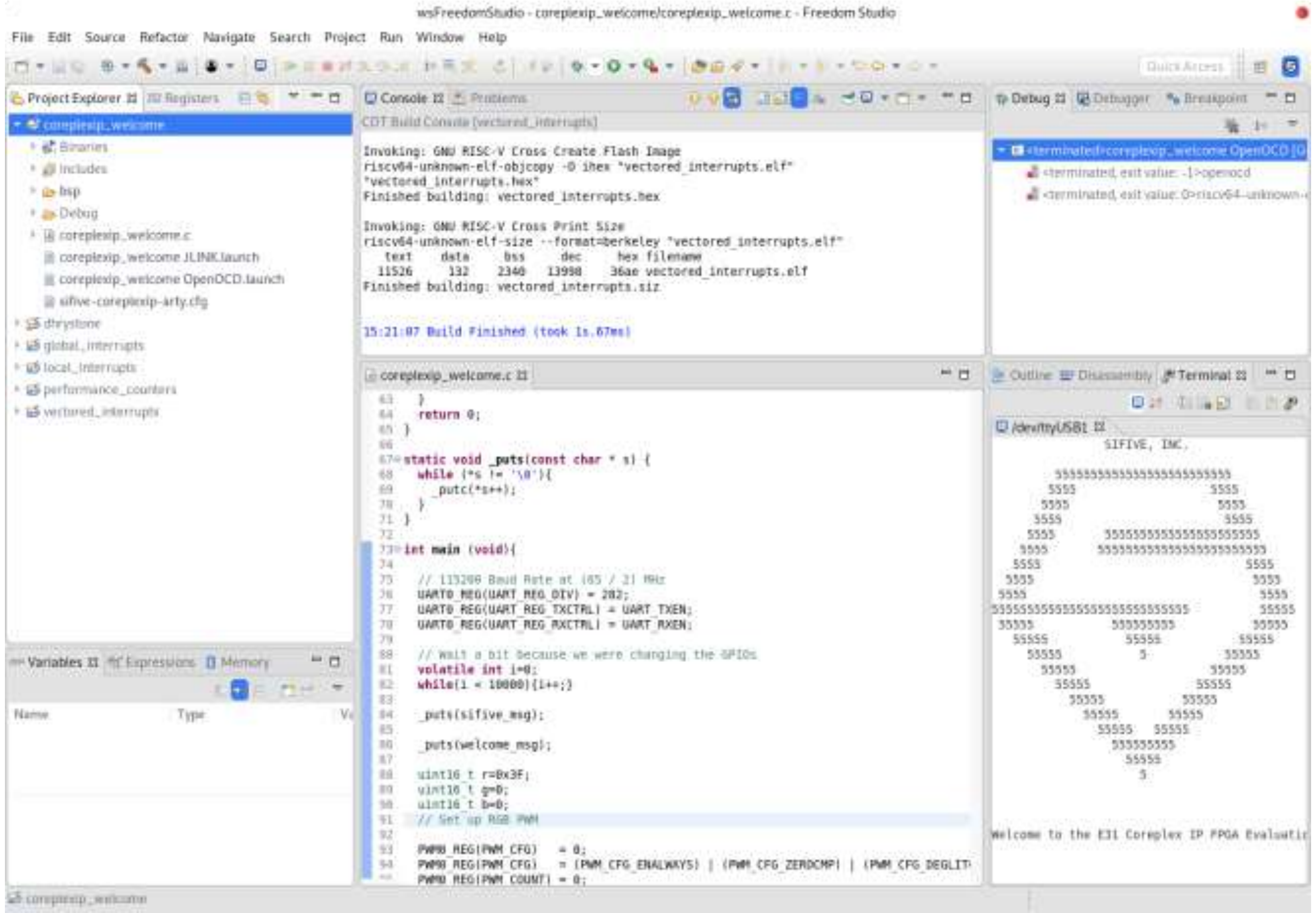
Freedom Studio is the fastest way to get started programming with your SiFive hardware. Freedom Studio is built on top of the popular Eclipse IDE and packaged with a prebuilt toolchain and example projects from the Freedom E SDK. Freedom Studio is compatible with all SiFive RISC-V development boards.

Freedom Studio

- **Freedom Studio is an Eclipse based IDE with**
  - pre-built GCC and OpenOCD
  - Bundled examples for SiFive targets
- **Download Freedom Studio**
  https://www.sifive.com/boards
  - Unzip to the desired installation directory
- **Or… Skip the IDE**
  - Download pre-built binaries of GCC and OpenOCD from the same webpage
  - Use Freedom-E-SDK to build and debug your software using a makefile CLI based flow https://github.com/sifive/freedom-e-sdk

Build and Run the Software

- File - Import - DevKit Examples - Browse
- Select the zip that matches your core
- Select the desired examples and click Finish
- Control-B will build the entire workspace
- **Run - Debug - OpenOCD** starts a JTAG Debug Session and Loads the program


wsFreedomStudio - coreplexip_welcome/coreplexip_welcome.c - Freedom Studio
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Project Explorer
Registers
coreplexip_welcome
Binaries
Includes
bsp
Debug
coreplexip_welcome.c
coreplexip_welcome JLINK.launch
coreplexip_welcome OpenOCD.launch
sifive-coreplexip-arty.cfg
dhrystone
global_interrupts
local_interrupts
performance_counters
vectored_interrupts
Variables
Expressions
Memory
Name
Type
Console
Problems
CDT Build Console [vectored_interrupts]
Debug
Debugger
Breakpoint
<terminated>coreplexip_welcome OpenOCD [G
<terminated, exit value: -1>openocd
<terminated, exit value: 0>riscv64-unknown-
coreplexip_welcome.c
Outline
Disassembly
Terminal
/dev/ttyUSB1
coreplexip_welcome

```
Invoking: GNU RISC-V Cross Create Flash Image
riscv64-unknown-elf-objcopy -O ihex "vectored_interrupts.elf"
"vectored_interrupts.hex"
Finished building: vectored_interrupts.hex

Invoking: GNU RISC-V Cross Print Size
riscv64-unknown-elf-size --format=berkeley "vectored_interrupts.elf"
   text    data     bss     dec     hex filename
  11526     132    2340   13998    36ae vectored_interrupts.elf
Finished building: vectored_interrupts.siz


15:21:07 Build Finished (took 1s.67ms)
```

```c
  }
  return 0;
}

static void _puts(const char * s) {
  while (*s != '\0'){
    _putc(*s++);
  }
}

int main (void){

  // 115200 Baud Rate at (65 / 2) MHz
  UART0_REG(UART_REG_DIV) = 282;
  UART0_REG(UART_REG_TXCTRL) = UART_TXEN;
  UART0_REG(UART_REG_RXCTRL) = UART_RXEN;

  // Wait a bit because we were changing the GPIOs
  volatile int i=0;
  while(i < 10000){i++;}

  _puts(sifive_msg);

  _puts(welcome_msg);

  uint16_t r=0x3F;
  uint16_t g=0;
  uint16_t b=0;
  // Set up RGB PWM

  PWM0_REG(PWM_CFG)   = 0;
  PWM0_REG(PWM_CFG)   = (PWM_CFG_ENALWAYS) | (PWM_CFG_ZEROCMP) | (PWM_CFG_DEGLIT
  PWM0_REG(PWM_COUNT) = 0;
```

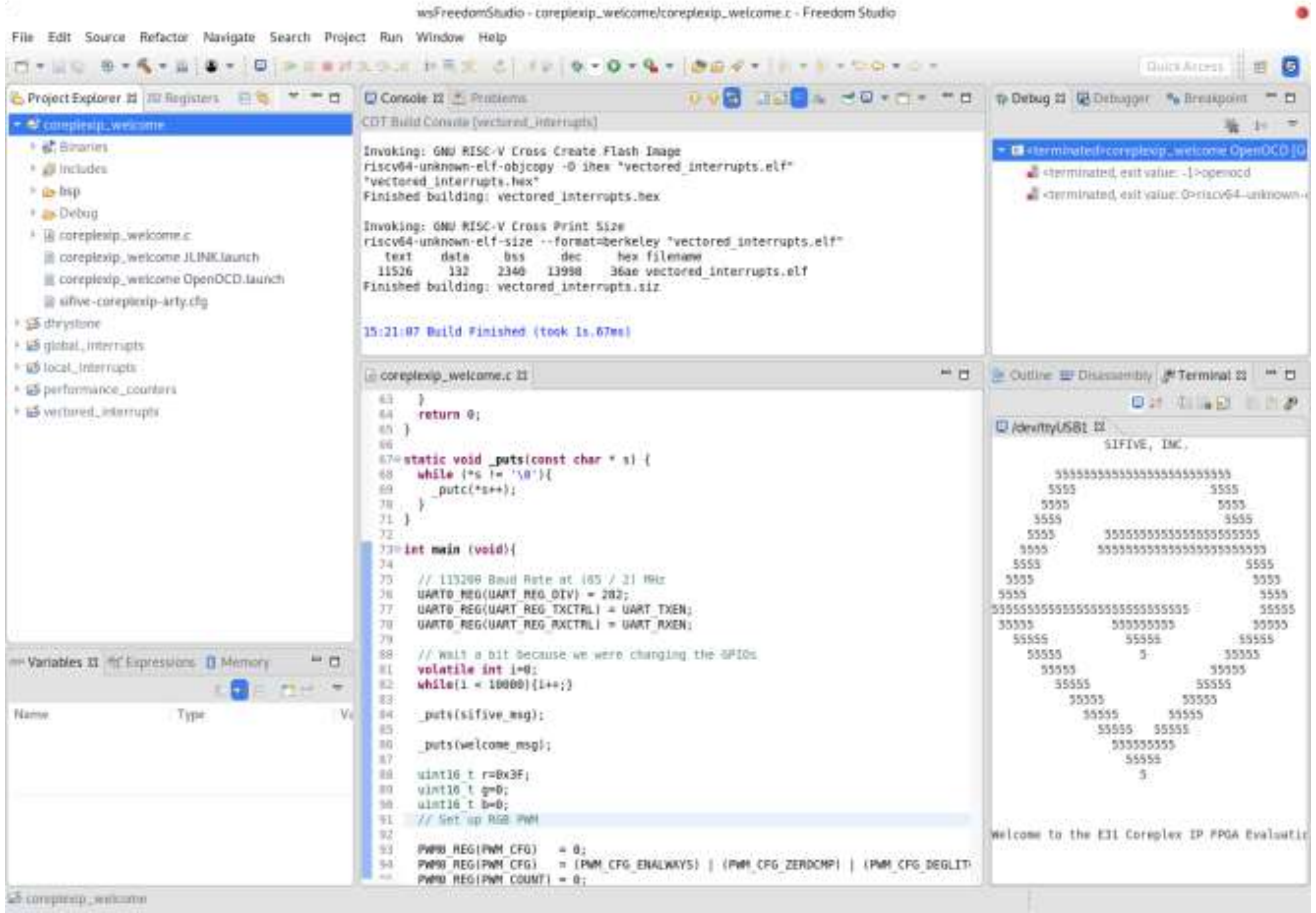
```
                SIFIVE, INC.

         5555555555555555555555555
        5555                   5555
       5555                     5555
      5555                       5555
     5555       5555555555555555555555
    5555       555555555555555555555555
   5555                             5555
  5555                               5555
 5555                                 5555
5555555555555555555555555555          55555
 55555           555555555           55555
   55555           55555           55555
     55555           5           55555
       55555                   55555
         55555               55555
           55555           55555
             55555       55555
               55555   55555
                 555555555
                   55555
                     5


Welcome to the E31 Coreplex IP FPGA Evaluati
```

Build and Run the Software (Change to New Flow)

- File - Import - DevKit Examples - Browse
- Select the zip that matches your core
- Select the desired examples and click Finish
- Control-B will build the entire workspace
- **Run - Debug - OpenOCD** starts a JTAG Debug Session and Loads the program


wsFreedomStudio - coreplexip_welcome/coreplexip_welcome.c - Freedom Studio
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Project Explorer
Registers
coreplexip_welcome
Binaries
Includes
bsp
Debug
coreplexip_welcome.c
coreplexip_welcome JLINK.launch
coreplexip_welcome OpenOCD.launch
sifive-coreplexip-arty.cfg
dhrystone
global_interrupts
local_interrupts
performance_counters
vectored_interrupts
Variables
Expressions
Memory
Name
Type
Console
Problems
CDT Build Console [vectored_interrupts]
Debug
Debugger
Breakpoint
<terminated>coreplexip_welcome OpenOCD [G
<terminated, exit value: -1>openocd
<terminated, exit value: 0>riscv64-unknown-
coreplexip_welcome.c
Outline
Disassembly
Terminal
/dev/ttyUSB1
coreplexip_welcome

```
Invoking: GNU RISC-V Cross Create Flash Image
riscv64-unknown-elf-objcopy -O ihex "vectored_interrupts.elf"
"vectored_interrupts.hex"
Finished building: vectored_interrupts.hex

Invoking: GNU RISC-V Cross Print Size
riscv64-unknown-elf-size --format=berkeley "vectored_interrupts.elf"
   text    data     bss     dec     hex filename
  11526     132    2340   13998    36ae vectored_interrupts.elf
Finished building: vectored_interrupts.siz

15:21:07 Build Finished (took 1s.67ms)
```

```c
  }
  return 0;
}

static void _puts(const char * s) {
  while (*s != '\0'){
    _putc(*s++);
  }
}

int main (void){

  // 115200 Baud Rate at (65 / 2) MHz
  UART0_REG(UART_REG_DIV) = 282;
  UART0_REG(UART_REG_TXCTRL) = UART_TXEN;
  UART0_REG(UART_REG_RXCTRL) = UART_RXEN;

  // Wait a bit because we were changing the GPIOs
  volatile int i=0;
  while(i < 10000){i++;}

  _puts(sifive_msg);

  _puts(welcome_msg);

  uint16_t r=0x3F;
  uint16_t g=0;
  uint16_t b=0;
  // Set up RGB PWM

  PWM0_REG(PWM_CFG)   = 0;
  PWM0_REG(PWM_CFG)   = (PWM_CFG_ENALWAYS) | (PWM_CFG_ZEROCMP) | (PWM_CFG_DEGLIT
  PWM0_REG(PWM_COUNT) = 0;
```

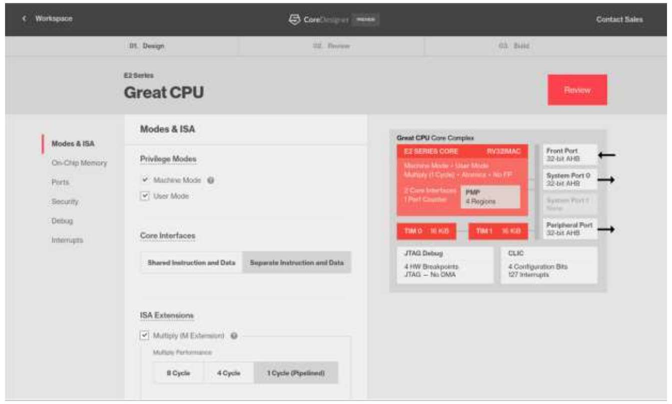
```
                SIFIVE, INC.

         5555555555555555555555555
        5555                   5555
       5555                     5555
      5555                       5555
     5555       5555555555555555555555
    5555       555555555555555555555555
   5555                             5555
  5555                               5555
 5555                                 5555
5555555555555555555555555555          55555
 55555           555555555           55555
   55555           55555           55555
     55555           5           55555
       55555                   55555
         55555               55555
           55555           55555
             55555       55555
               55555   55555
                 555555555
                   55555
                     5



Welcome to the E31 Coreplex IP FPGA Evaluati
```

Demo - From Custom CPU to Hello World in 30 minutes



E2 Series

Great CPU

Modes & ISA

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)

Great CPU 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

TIM 0 16 KiB

TIM 1 16 KiB

Front Port
32-bit AHB

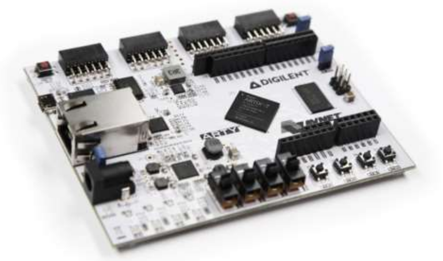
System Port 0
32-bit AHB

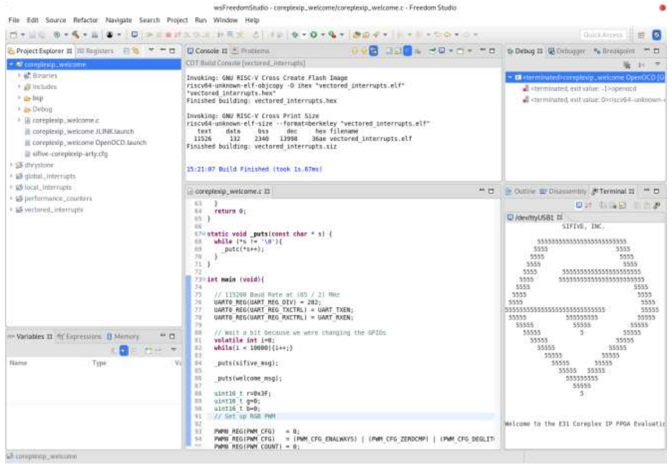
System Port 1
None

Peripheral Port
32-bit AHB

JTAG Debug
4 HW Breakpoints
JTAG — No DMA

CLIC
4 Configuration Bits
127 Interrupts





CDT Build Console [vectored_interrupts]

```
Invoking: GNU RISC-V Cross Create Flash Image
riscv64-unknown-elf-objcopy -O ihex "vectored_interrupts.elf"
"vectored_interrupts.hex"
Finished building: vectored_interrupts.hex

Invoking: GNU RISC-V Cross Print Size
riscv64-unknown-elf-size --format=berkeley "vectored_interrupts.elf"
   text    data     bss     dec     hex filename
  11526     132    2340   13998    36ae vectored_interrupts.elf
Finished building: vectored_interrupts.siz


15:21:07 Build Finished (took 1s.67ms)
```

<terminated>coreplexip_welcome OpenOCD [G

<terminated, exit value: -1>openocd

<terminated, exit value: 0>riscv64-unknown-

```
  }
  return 0;
}

static void _puts(const char * s) {
  while (*s != '\0'){
    _putc(*s++);
  }
}

int main (void){

  // 115200 Baud Rate at (65 / 2) MHz
  UART0_REG(UART_REG_DIV) = 282;
  UART0_REG(UART_REG_TXCTRL) = UART_TXEN;
  UART0_REG(UART_REG_RXCTRL) = UART_RXEN;

  // Wait a bit because we were changing the GPIOs
  volatile int i=0;
  while(i < 10000){i++;}

  _puts(sifive_msg);

  _puts(welcome_msg);

  uint16_t r=0x3F;
  uint16_t g=0;
  uint16_t b=0;
  // Set up RGB PWM

  PWM0_REG(PWM_CFG)   = 0;
  PWM0_REG(PWM_CFG)   = (PWM_CFG_ENALWAYS) | (PWM_CFG_ZEROCMP) | (PWM_CFG_DEGLIT
  PWM0_REG(PWM_COUNT) = 0;
```

```
                SIFIVE, INC.

Welcome to the E31 Coreplex IP FPGA Evaluati
```

Step 1

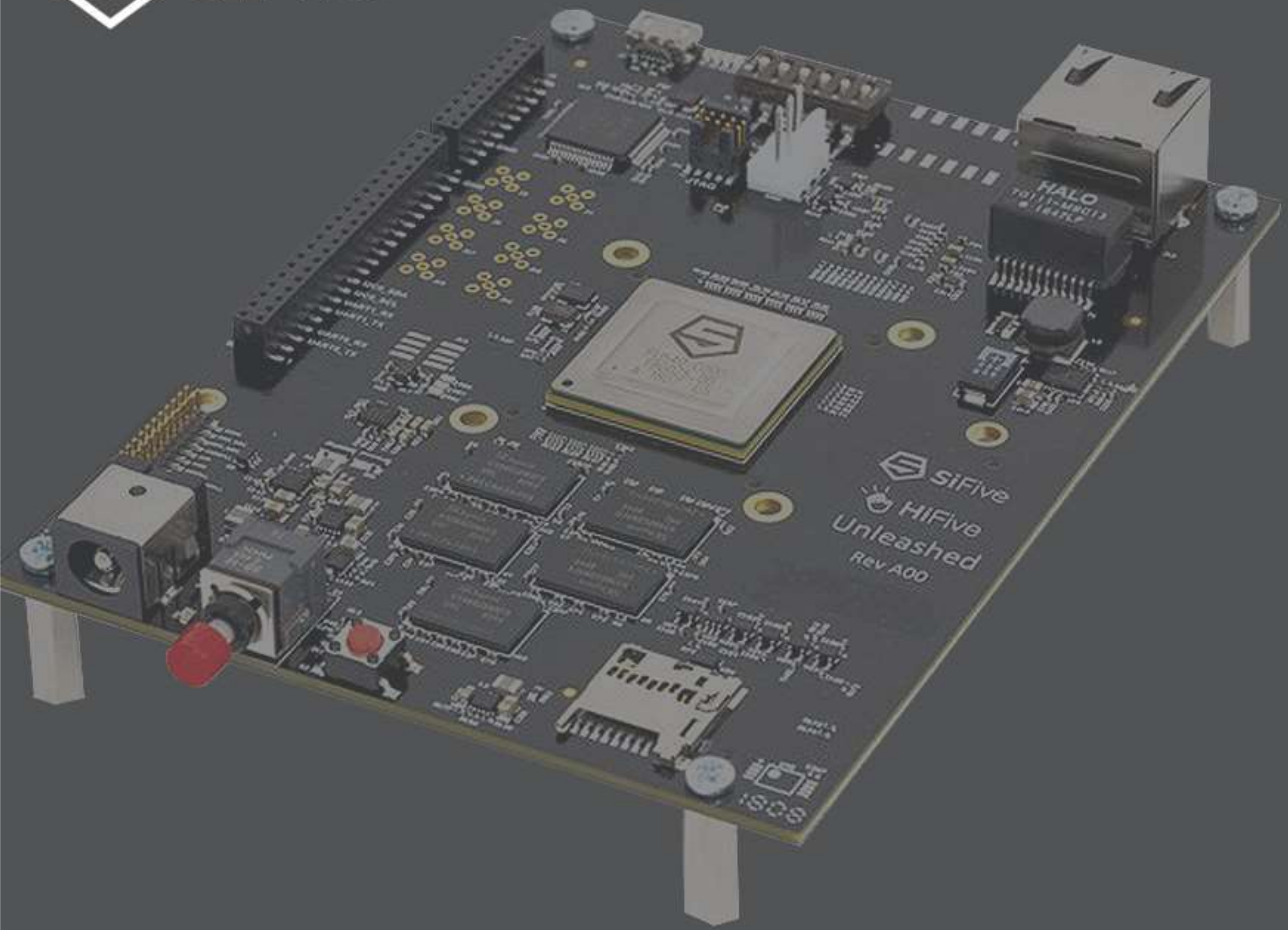
Configure a custom SiFive RISC-V Core using SiFive Core Designer

Step 2

Use the FPGA bitstream from Step 1 to program a Digilent Arty FPGA board with the configured CPU

Step 3

Use Freedom Studio and the SiFive SDK to program and run Hello World



Silicon verified. Market proven.

The most advanced configurable core IP and silicon solutions from the inventors of RISC-V.

Microcontrollers ■ Embedded ■ Linux ■ Multicore

■ Networking ■ Storage ■ Computing ■ AI
■ Industrial ■ IoT ■ Consumer ■ Automotive

www.sifive.com