



SiFive Core Designer

**From Custom CPU to Hello World in 30 Minutes**

---



# Silicon at the speed of software.

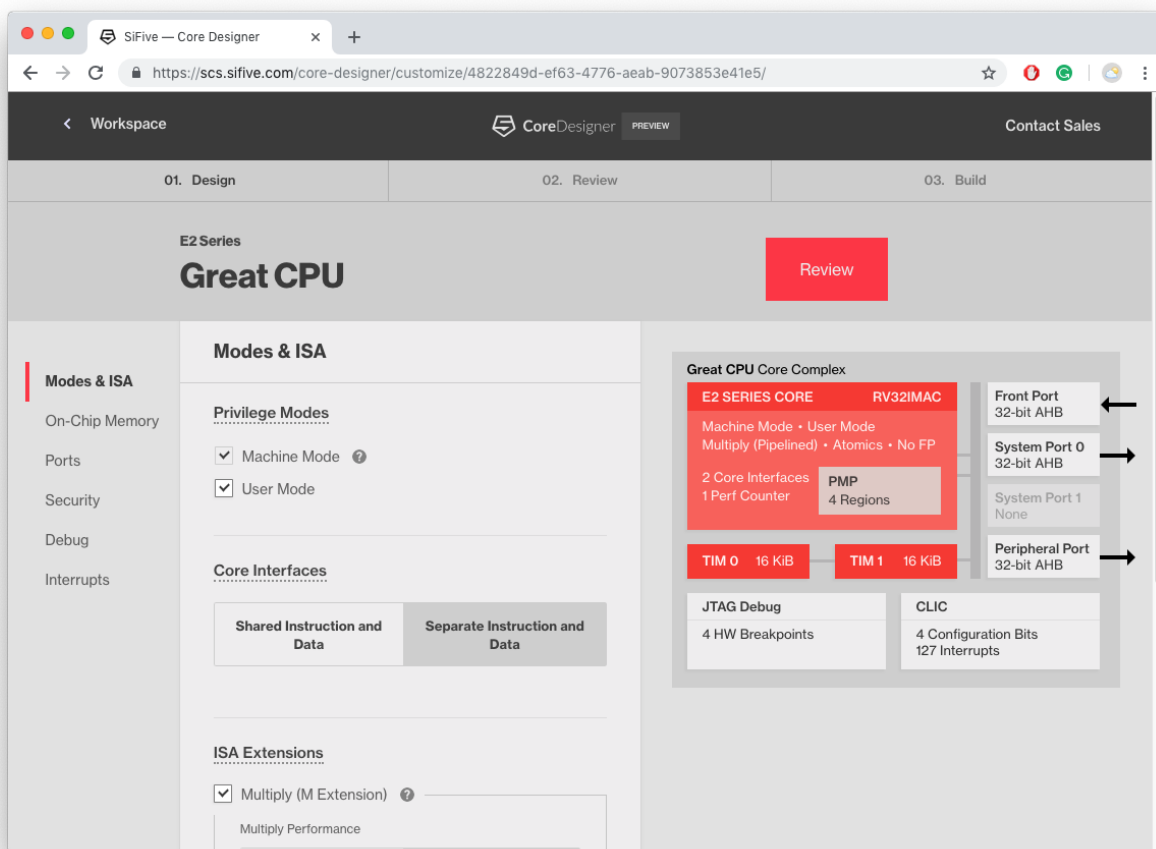
Design RISC-V CPUs in an hour. Get custom SoCs in weeks, not months. Impossible? Not anymore.

[Start Designing](#)



# SiFive Core Designer

Your interface to SiFive RISC-V Core IP



- All SiFive Core IP is configured and delivered via the SiFive Core Designer Web Portal
  - Simple, Easy to Use, Web Interface
- **Variants** are generated with click of a button and available from the Workspace
- **Variants** 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

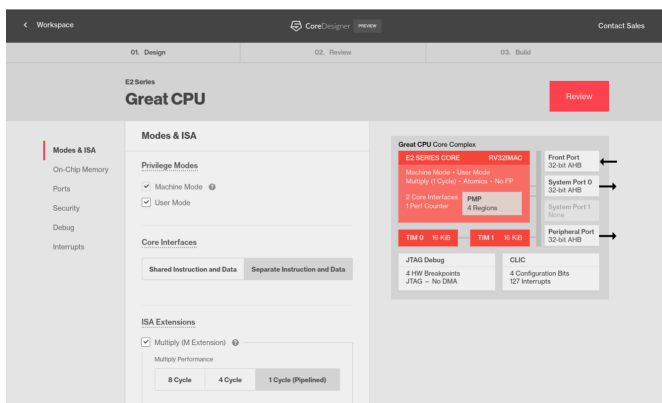


# In-house IP: SiFive RISC-V Core IP Product Offering

	<b>E Cores</b> 32-bit embedded cores MCU, edge computing, AI, IoT	<b>S Cores</b> 64-bit embedded cores Storage, AR/VR, machine learning	<b>U Cores</b> 64-bit application cores Linux, datacenter, network baseband
<b>7 Series</b>  Highest performance: 8-stage, dual-issue superscalar pipeline	<b>E7 Series</b> <ul style="list-style-type: none"> <li>&gt; <b>E76-MC</b> Quad-core 32-bit embedded processor</li> <li>&gt; <b>E76</b> High performance 32-bit embedded core</li> </ul>	<b>S7 Series</b> <ul style="list-style-type: none"> <li>&gt; <b>S76-MC</b> Quad-core 64-bit embedded processor</li> <li>&gt; <b>S76</b> High-performance 64-bit embedded core</li> </ul>	<b>U7 Series</b> <ul style="list-style-type: none"> <li>&gt; <b>U74-MC</b> Multicore: four U74 cores and one S76 core</li> <li>&gt; <b>U74</b> High performance Linux-capable processor</li> </ul>
<b>3/5 Series</b>  Efficient performance: 5-6-stage, single- issue pipeline	<b>E3 Series</b> <ul style="list-style-type: none"> <li>&gt; <b>E34</b> E31 features + single-precision floating point</li> <li>&gt; <b>E31</b> Balanced performance and efficiency</li> </ul>	<b>S5 Series</b> <ul style="list-style-type: none"> <li>&gt; <b>S54</b> S51 features + single-precision floating point</li> <li>&gt; <b>S51</b> Low-power 64-bit MCU core</li> </ul>	<b>U5 Series</b> <ul style="list-style-type: none"> <li>&gt; <b>U54-MC</b> Multicore application processor with four U54 cores and one S76 core</li> <li>&gt; <b>U54</b> Linux-capable application processor</li> </ul>
<b>2 Series</b>  Power & area optimized: 2-3-stage, single- issue pipeline	<b>E2 Series</b> <ul style="list-style-type: none"> <li>&gt; <b>E24</b> E21 + single-precision floating point</li> <li>&gt; <b>E21</b> E20 + User Mode, Atomics, Multiply, TIM</li> <li>&gt; <b>E20</b> Our smallest, most efficient core</li> </ul>	<b>S2 Series</b> <ul style="list-style-type: none"> <li>&gt; <b>S21</b> Area-efficient 64-bit MCU core</li> </ul>	

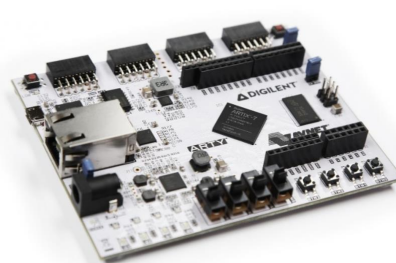


# From Custom CPU to Hello World in 30 minutes



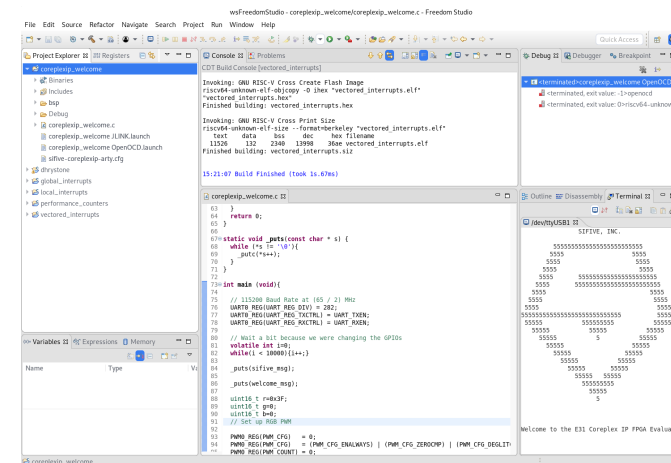
## Step 1

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

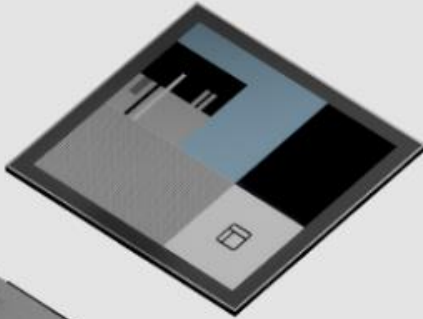
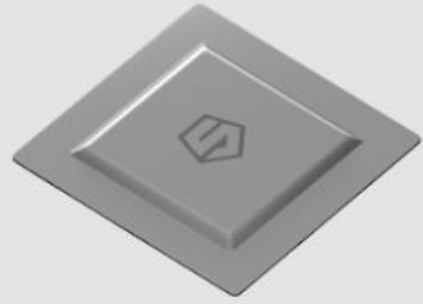
Use Freedom Studio and the SiFive SDK to program and run Hello World



# 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.

Design Core

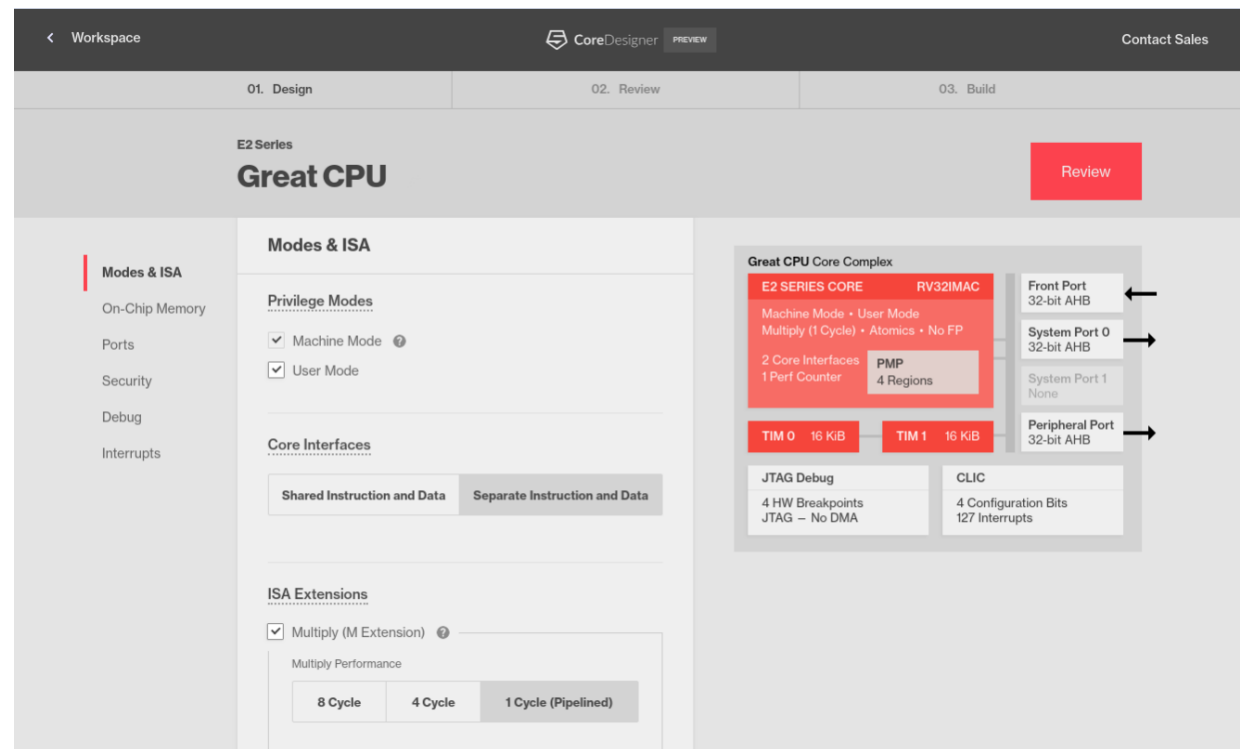




# Configure a SiFive RISC-V CPU

## 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**







# 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

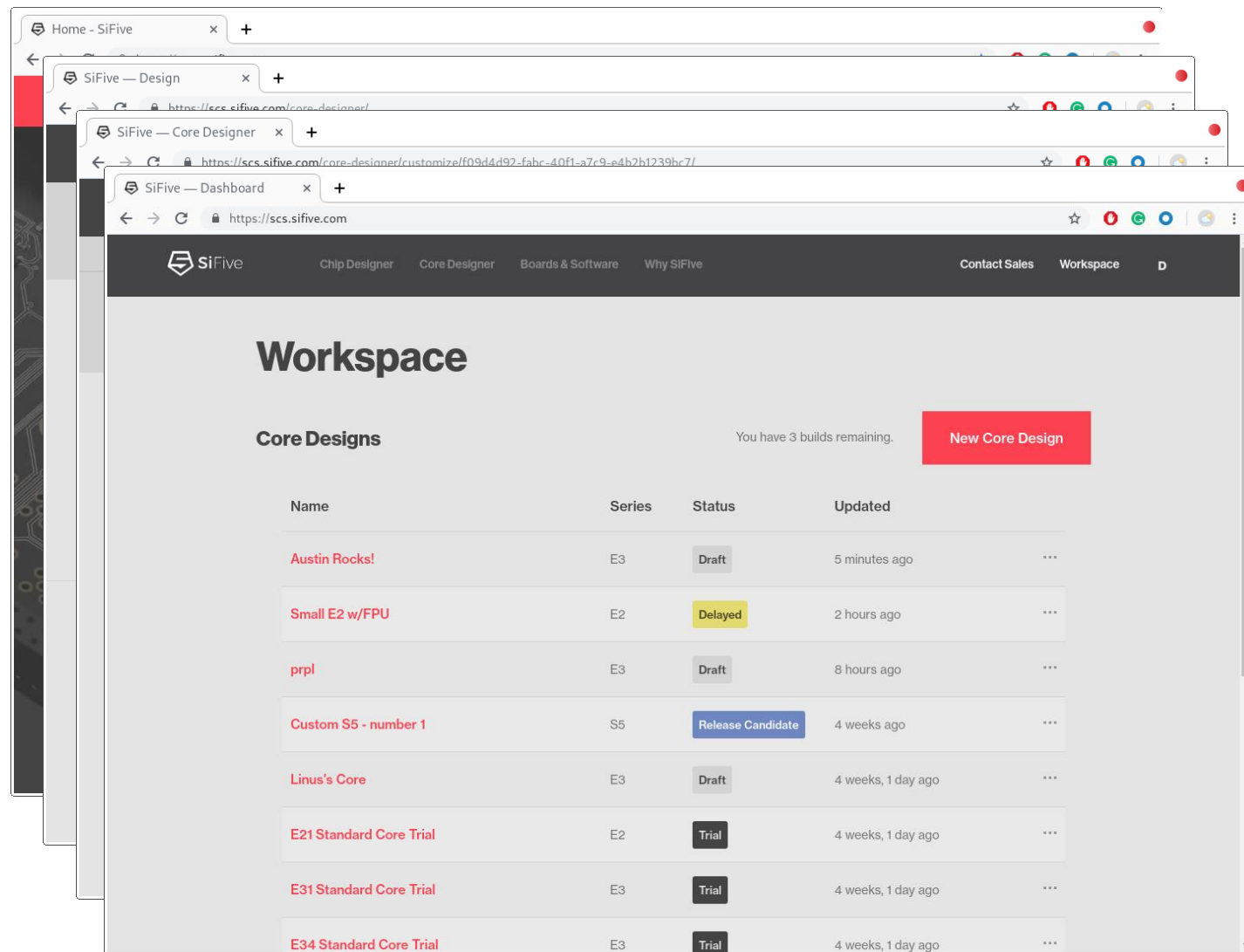
Name the Design and Start Clicking!

- 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

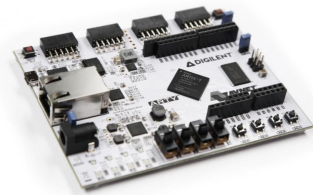
Download from your SiFive Workspace





# Too Many Choices? Start with a Standard Core

## E3 Series



FPGA Evaluations



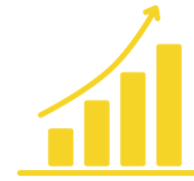
RTL Evaluations

**E31 Standard Core Evaluation Core Complex**

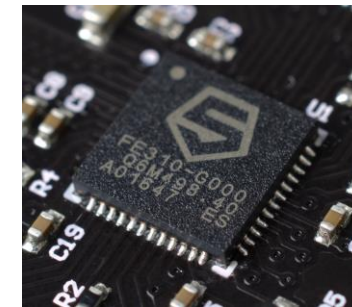
<b>E3 SERIES CORE</b>		<b>RV32IMAC</b>
Machine Mode • User Mode Multiply (1 Cycle) • Atomics • No FP 16 Local Interrupts • 2 Perf Counters		
28 BTB Entries • 512 BHT Entries 6 Ret Addr Stack		
Physical Memory Protection 8 Regions		
No ECC Support		
Instr Cache 16KiB • 2-way	Data TIM 64KiB	
JTAG Debug 4 HW Breakpoints JTAG – DMA	PLIC 7 Priority Levels 127 Global Int.	CLINT

Front Port 32-bit AHB ←  
System Port 32-bit AHB →  
Peripheral Port 32-bit AHB →  
Memory Port None

E31 Standard Core Definition



Benchmarks



FE310 Silicon

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



## Step 2 - Download the Deliverables and Program the FPGA

---



# Download the Deliverables from your SCD Workspace

The screenshot shows the SiFive Workspace dashboard. At the top, there's a navigation bar with the SiFive logo and links for Chip Designer, Core Designer, Boards & Software, Why SiFive, Contact Sales, and Workspace. The main heading is "Workspace". Below it, there's a section for "Core Designs" with a sub-header "You have 3 builds remaining." and a red "New Core Design" button. A table lists several core designs with columns for Name, Series, Status, and Updated.

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

The screenshot shows the SiFive Deliverable page for "Custom S5 - number 1". The page has a "Workspace" breadcrumb and a "CoreDesigner" button. The main heading is "Custom S5 - number 1" with a "Release Candidate" button. Below the heading, there are tabs for "Diagram" and "Settings". The "Diagram" tab is active, showing a block diagram of the "Custom S5 - number 1 Core Complex". The diagram includes a central "S5 SERIES CORE" block with "RV64IMAC" architecture, and various peripheral blocks like "Front Port 32-bit AXI4", "System Port 32-bit AXI4", "Peripheral Port 32-bit AXI4", and "Memory Port None". A red box on the right side of the diagram highlights the "Custom S5 - number 1 Dev Kit (4.6 MB)". Below the diagram, there's a "Support" section with a link to "Open a Service Ticket" and a "Reference Build ID: fbd5f3a99831". At the bottom, there's a "Software Development Tools" section with links for "SDK" and "More Tools".



# Deploy the bitstream to the FPGA

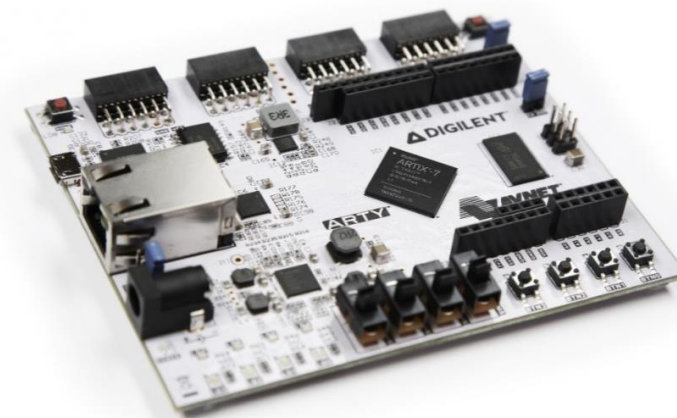
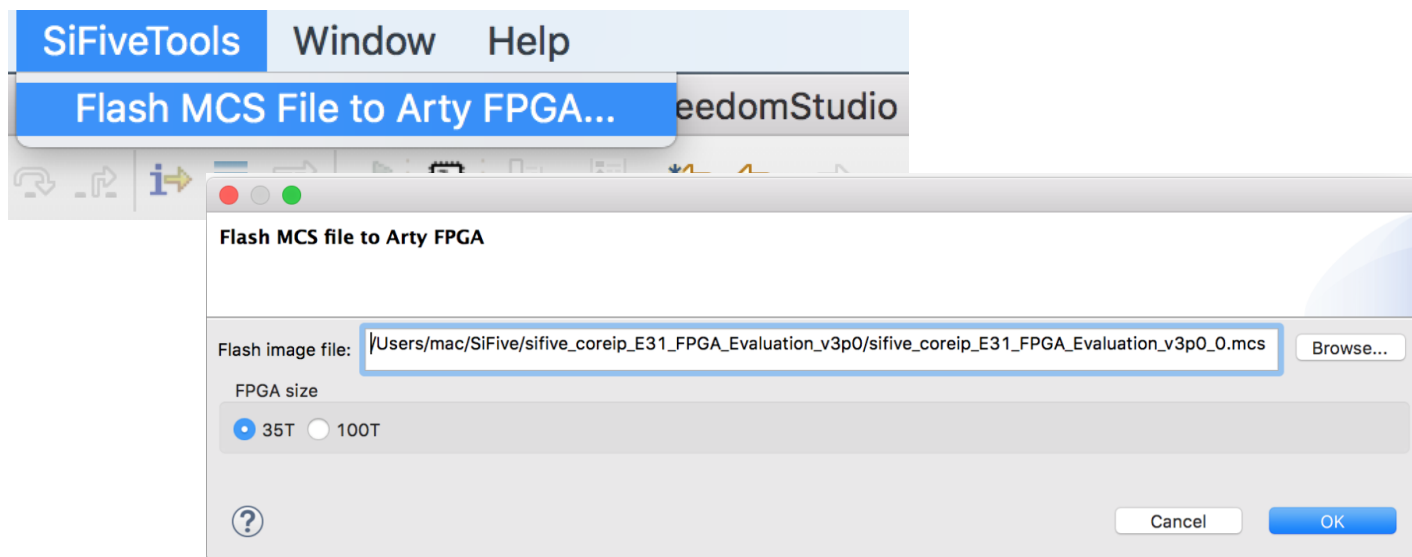
## 1. Purchase a Digilent Arty

<https://store.digilentinc.com/artix-a7-artix-7-fpga-development-board-for-makers-and-hobbyists/>

## 2. Download Freedom Studio

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

## 3. Use Freedom Studio's Integrated Arty Flashing utility



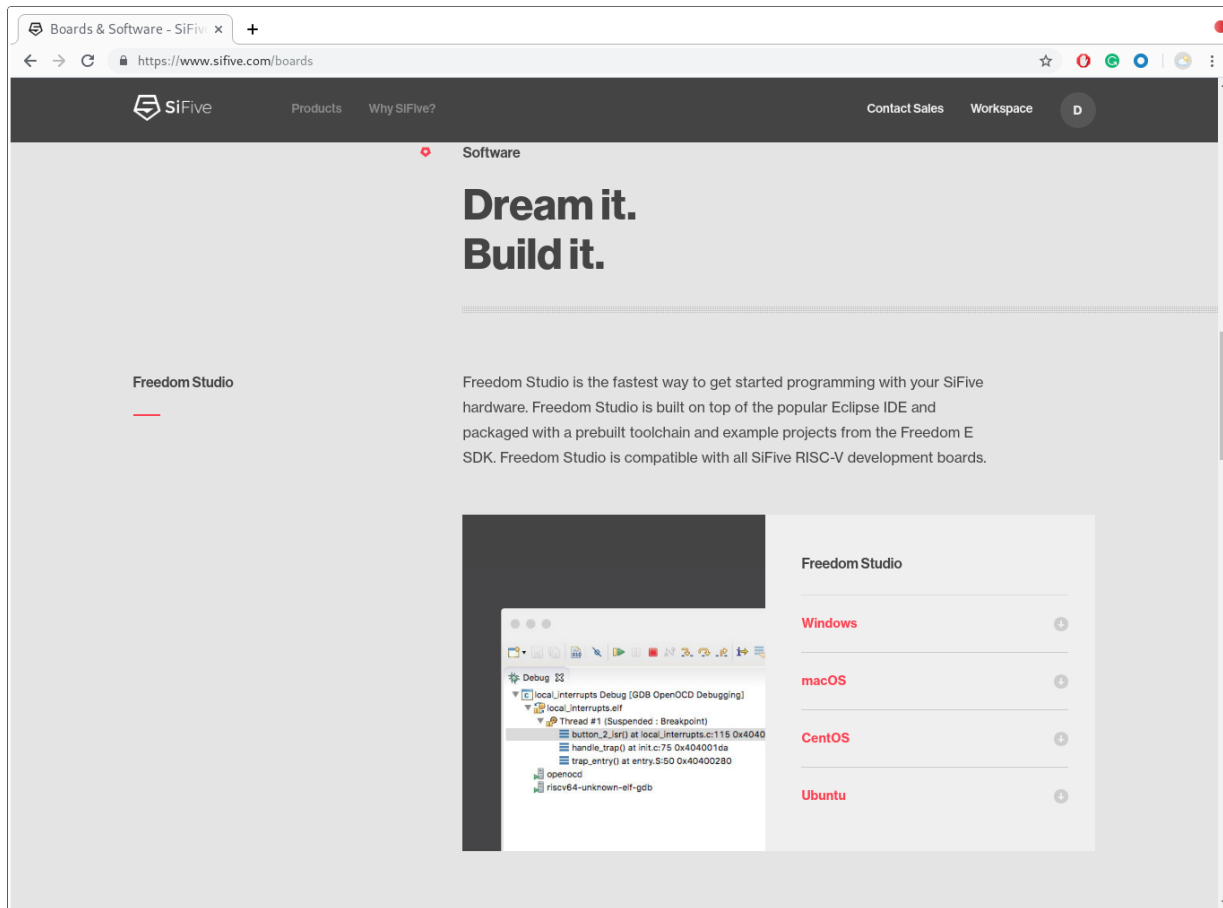


## Step 3 - Hello World!

---



# Download 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

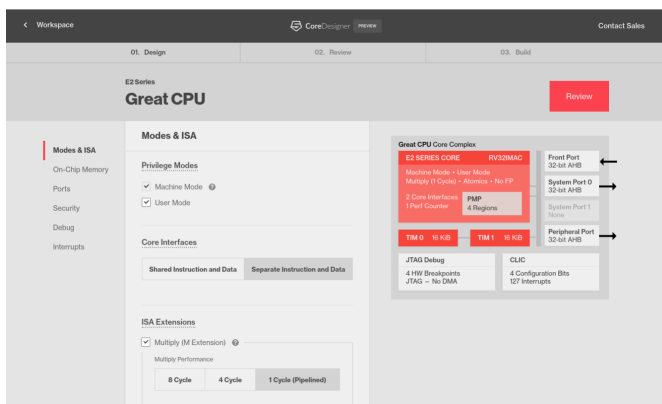
- File - New – C Project
- Select the Freedom E SDK Makefile Project
- Select the desired device and example and click Finish
- The program will automatically build and launch the debugger!

The screenshot displays the Freedom Studio IDE interface for a C project named 'coreplexip\_welcome'. The Project Explorer on the left shows the project structure, including source files like 'coreplexip\_welcome.c'. The Console window shows the build process, including the invocation of GNU RISC-V Cross Create Flash Image and GNU RISC-V Cross Print Size, resulting in a successful build of 'vectored\_interrupts.hex' and 'vectored\_interrupts.siz' in 1.67ms. The main editor window shows the source code for 'coreplexip\_welcome.c', which includes a main function that prints a message and sets up hardware registers. The Debug Console on the right shows the program's output, which consists of a series of 'S' characters and a welcome message: 'Welcome to the E31 Coreplex IP FPGA Evaluation'.



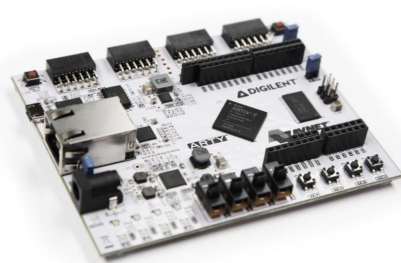


# Demo - From Custom CPU to Hello World in 30 minutes



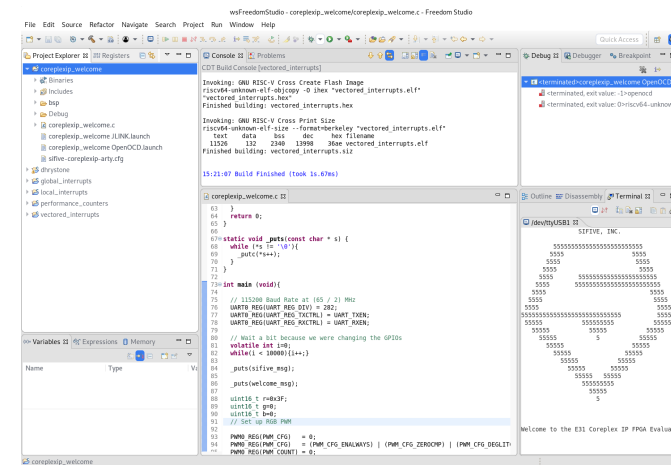
## Step 1

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



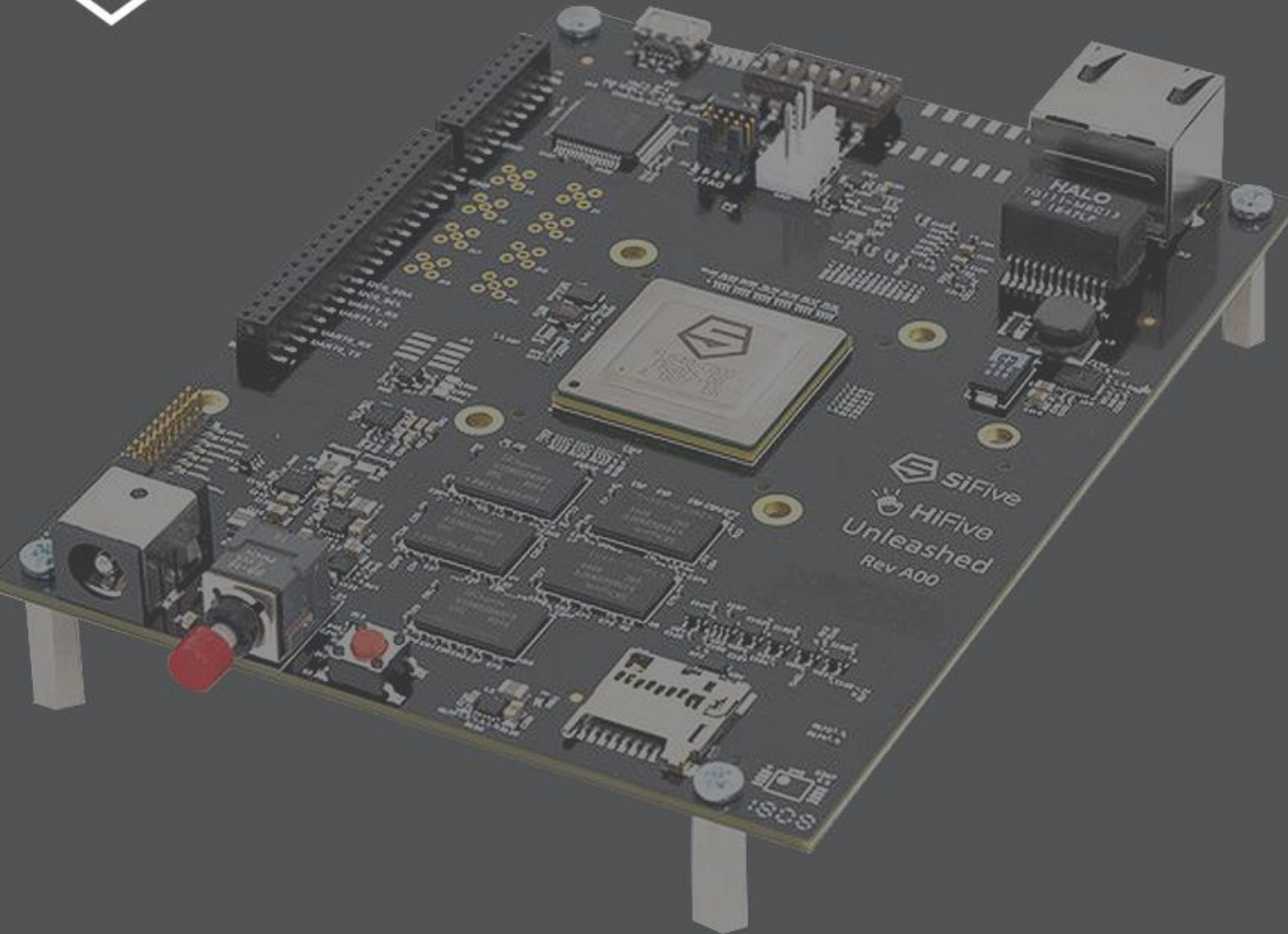
## Step 2

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](http://www.sifive.com)