

Automated, Simulation-Based Flow for Low-Cost FPGA-Accelerated Devices with Zephyr on BeagleV-Fire

Embedded Open Source Summit, Seattle, 2024-04-17

Piotr Zierhoffer, pzierhoffer@antmicro.com



ANTMICRO

- Open source SW-driven company with extensive HW know-how - developing complete systems
 - Heavily involved in RISC-V (Founding member), Zephyr (Platinum member), RISC-V maintainers
 - Also building tools for open source hardware/FPGA/ASIC development (CHIPS Alliance)



MICROCHIP

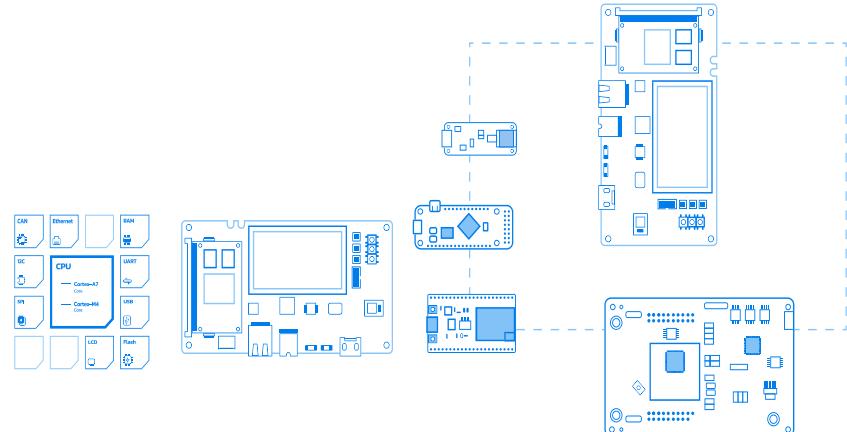
- Fellow RISC-V Founding member, original largest semiconductor (back then Microsemi) to commit to RISC-V
- PolarFire SoC - first RISC-V-based SoC FPGA
- 4+1 RISC-V cores (4 application cores and 1 monitor/boot core)
 - tightly integrated FPGA fabric
- Started cooperation with Antmicro on RISC-V support in Renode back in 2017, continues until today
- Renode has [support](#) for the PFSoC / Icicle Kit
 - [Integration](#) with Microchip SoftConsole
 - [Co-simulation](#) with Verilator



RENODE

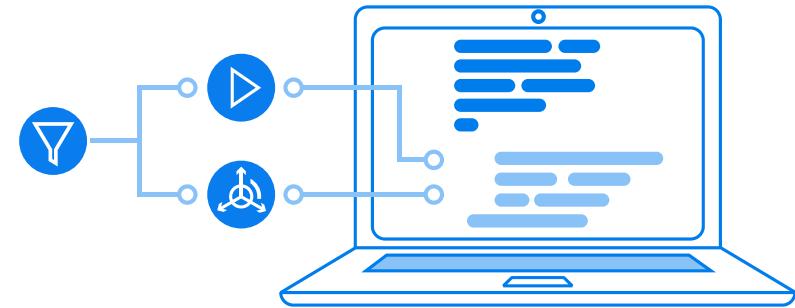
- **Renode** is Antmicro's flagship open-source system development framework featuring software/hardware/FPGA/ASIC co-simulation capability for the flexible and rapid design of heterogeneous systems
- Simulation of both low-power MCUs and multicore, 64-bit application platforms
- We offer commercial support and engineering services around Renode and integrating it into various interactive and CI workflows

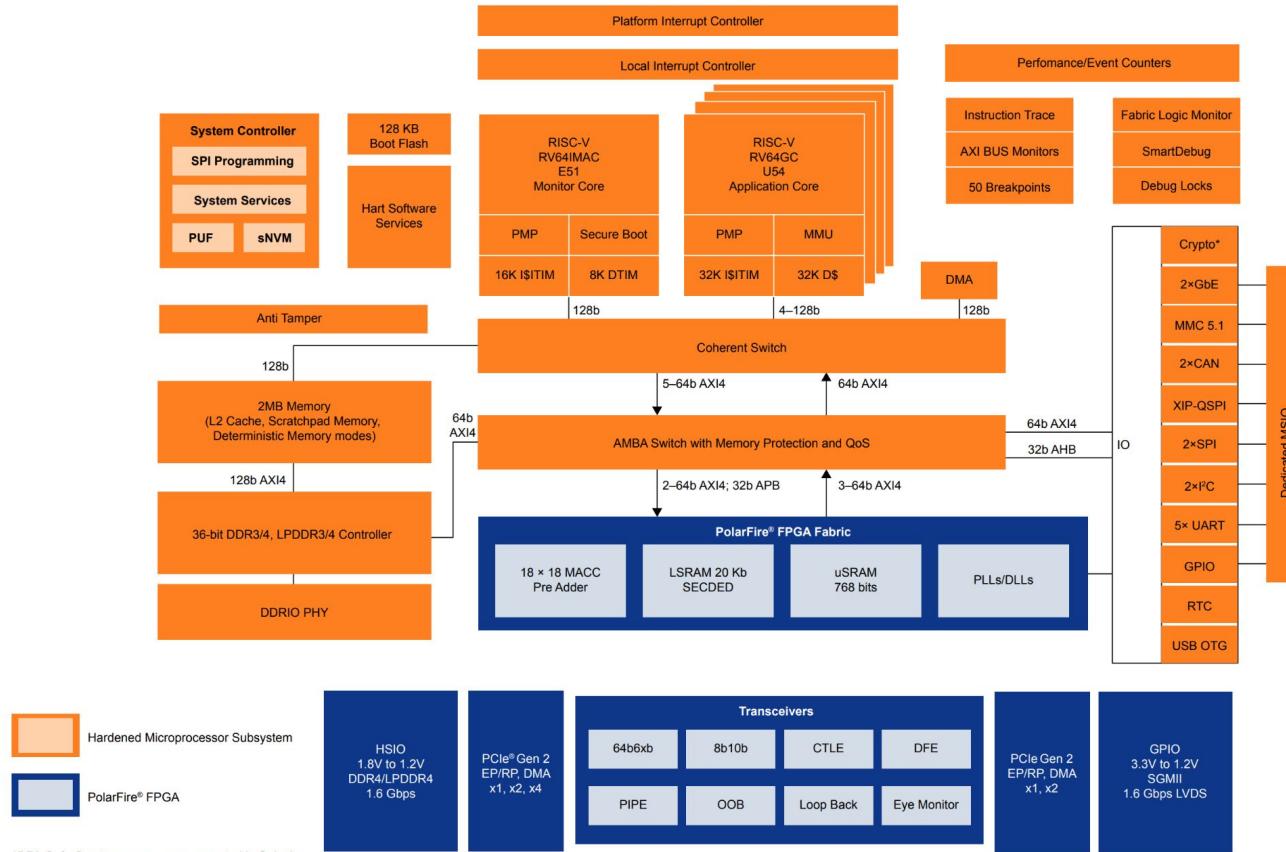
RENODE™



SUPPORTS COMPLEX, MULTI-NODE ENVIRONMENTS

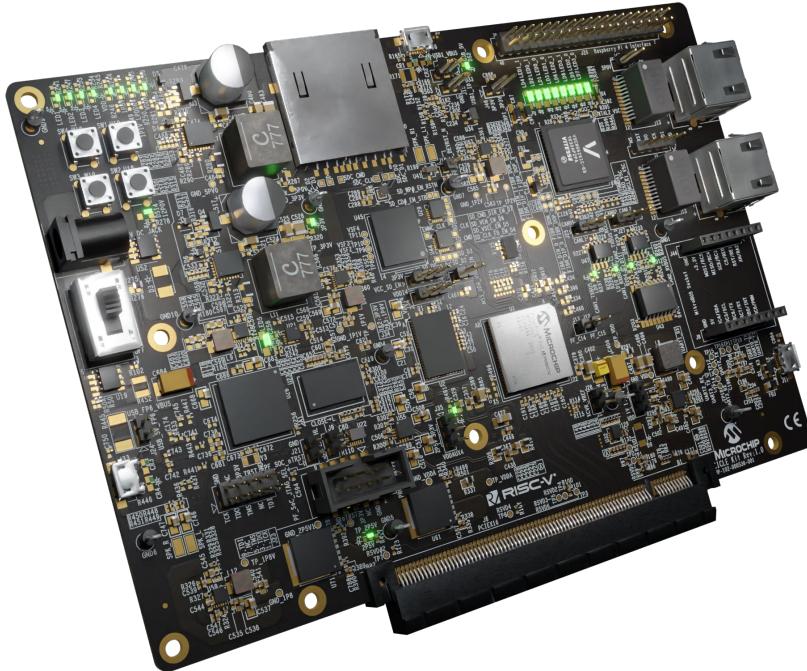
- Renode allows to design and test sophisticated **multi-node environments**, heterogeneous, multi-ISA; complete products
- **Broad ISA support**
- Full-system, binary-compatible simulation, software agnostic - but meshes best with open source like Zephyr!
- **Lets you debug simulated applications with GDB**, also for multiple nodes at the same time





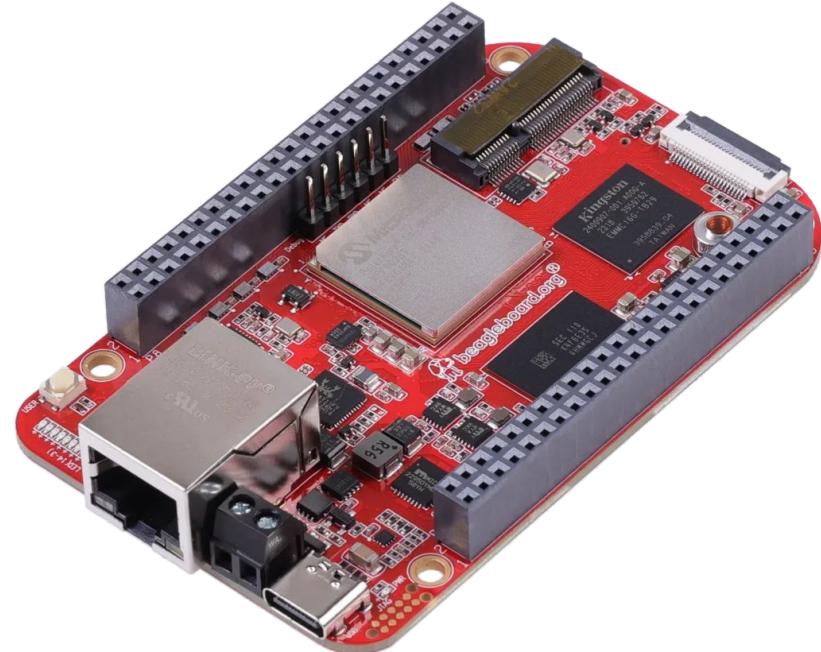
ORIGINAL PFSOC BOARDS

- Icicle Kit and Video Kit
- Aimed at professional users and early adopters
- First mass-market multi-core, Linux capable RISC-V silicon
 - this is actually the same CPU as the SiFive devboards



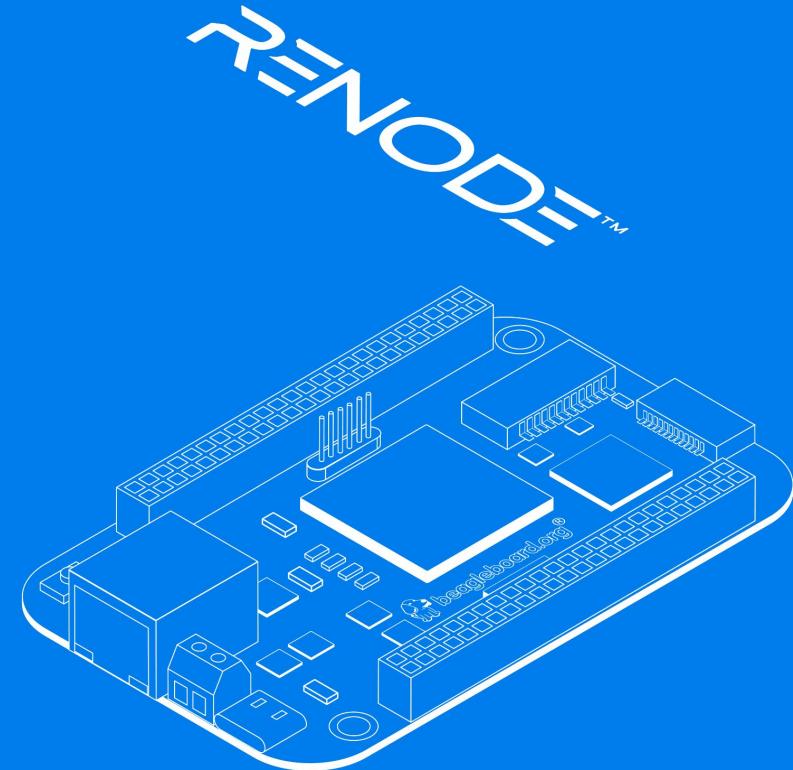
BEAGLE-V FIRE

- Much lower-cost solution from BeagleBoard.org
- Similar to typical “hobbyist” SBCs but with RISC-V processors and FPGA fabric
- BeagleBone Black header, M2 Key E
- Wide support and resources for users of the BeagleV-Fire from the BeagleBoard community
- **Open source design**
- Comprehensive [support in Renode Zephyr Dashboard/Renodepedia](#)



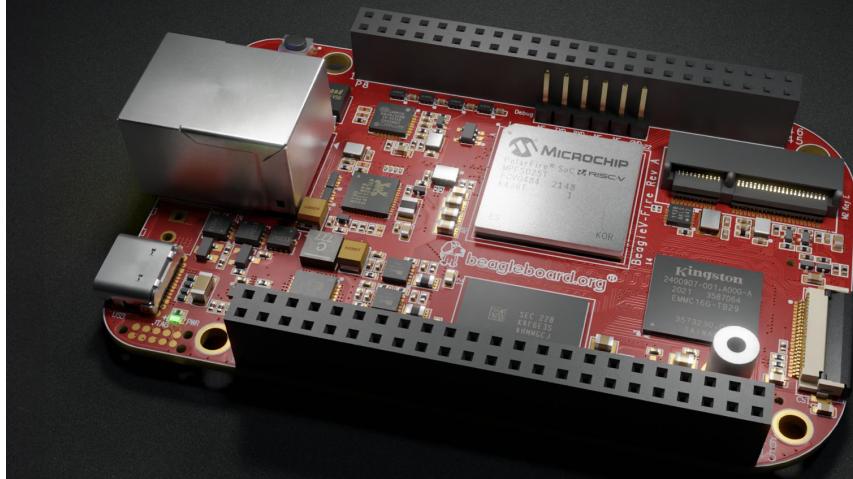
POLARFIRE SOC AND RENODE

- We've been working with Microchip for a long time to have first-class support for PFSoC in Renode
- Range of supported blocks and interfaces:
 - multicore RISC-V
 - DMA
 - Watchdog
 - RTC
 - Timer
 - GPIO
 - UART
 - SPI
 - I2C
 - CAN
 - Ethernet
 - eNVM
 - QSPI
 - SD/MMC
 - PCIe*



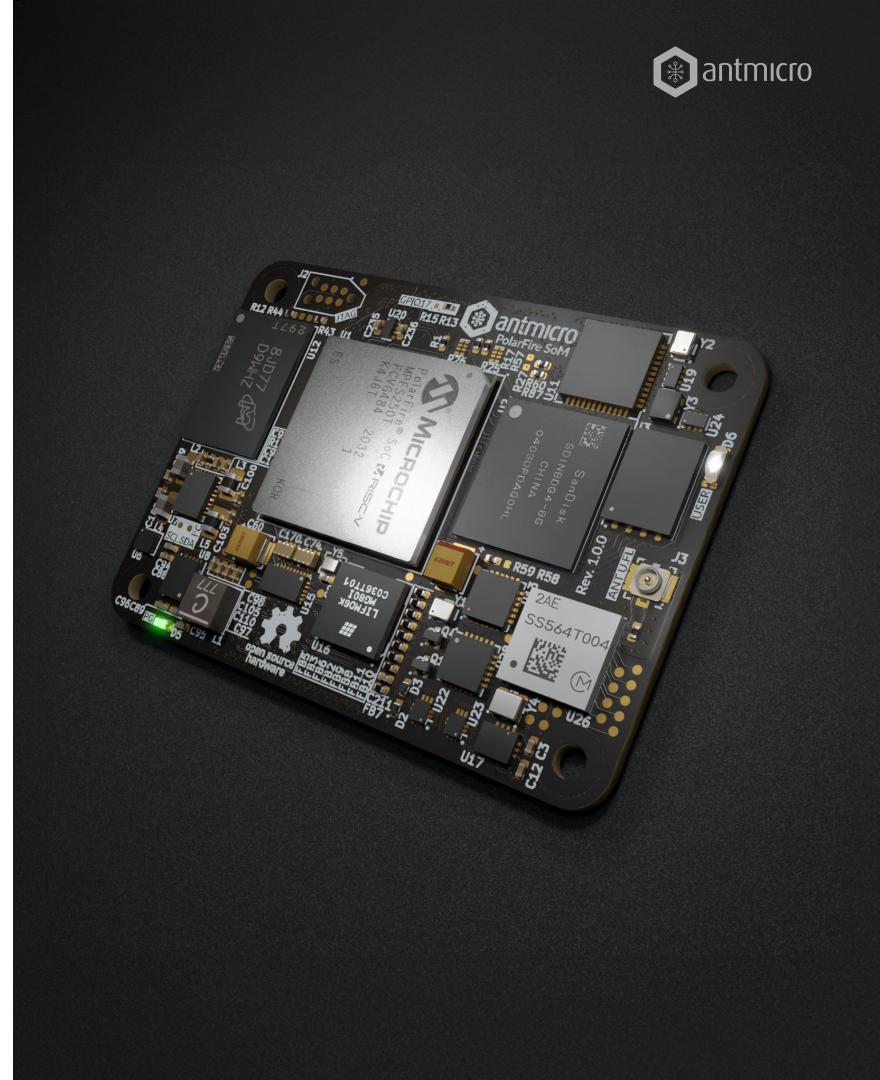
YET ANOTHER RISC-V PLATFORM?

- 1x SiFive E51 - RISC-V64 IMAC
- 4x SiFive U54 - RISC-V64 GC, SMP-capable
- 2x GigE MACs
- USB 2.0 OTG
- 5x multi-mode UARTs, 2x SPI, 2x I2C, 2x CAN 2.0 Controllers
- 2x PCIe Gen2 End Points/Root Ports
- **23K LE FPGA fabric**



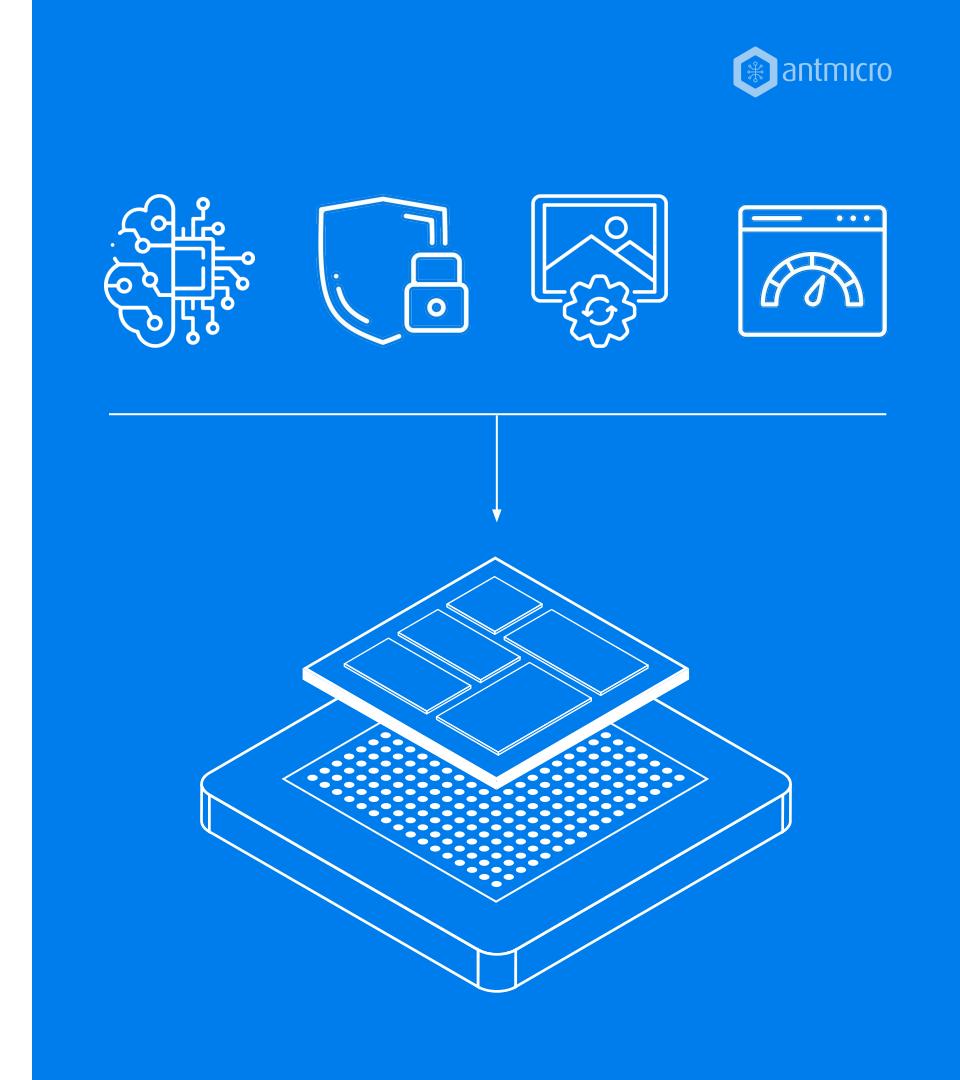
MORE AFFORDABLE RISC-V?

- At the time of submission, Beagle-V Fire was one of the few affordable options for low-cost powerful RISC-V
- Since then...
 - Microchip Discovery Kit - 95K LE
 - Antmicro PolarFire SoM:
openhardware.antmicro.com/boards/polarfire-som/
- 254K LE
- Multi-core RISC-V + FPGA adoption is no longer gated on cost, but ease of use



WHAT CAN YOU DO WITH AN FPGA SOC?

- Create specialized, low power embedded hardware platforms
- Best of both worlds: hard general purpose CPU and soft custom logic for:
 - Machine Learning accelerators
 - Security, crypto mechanisms
 - Image and video processing
 - Any other high-speed processing needing power-efficiency



“TRADITIONAL” FLOW: SOFTCONSOLE

- Microchip contributed the Beagle-V port to Zephyr
 - They offer an Eclipse-based IDE, SoftConsole
 - Integrated with Renode for many years now
 - Easy, “click to run and debug” flow
 - on hardware
 - in Renode
 - Does require you to program the FPGA with other tools
 - This, while convenient, is not the “typical” Zephyr development flow

The screenshot shows the Renode software interface, which includes a Project Explorer, a code editor, and a terminal window.

Project Explorer: Shows the project structure for "pse-blinky".

Code Editor: Displays the file `mss_gpio.c` with the following code snippet:

```
184 }
185 {
186     i
187     { (monitor) i S
188     { (machine-0) [ machine-0:sysbus.mmuart0
189         Setting outputs 0, 1 and 2 to high
190         * Setting
191         * output
192         Setting outputs 0, 1 and 2 to low
193         Setting outputs 0, 1 and 2 to high
194         if(value)
195             gpio
196         else
197             gpio
198     }
199
200     * -----
201     * MSS_GPIO_driver
202     * See "mss_gpio.h"
203     */
204     void MSS_GPIO_dri
205     (
206         GPIO_TypeDef
207         mss_gpio_id_t port_id,
208         mss_gpio_inout_state_t inout_state
209     )
210     {
211         if(0 == gpio_number_validate(gpio, port_id))
212         {
213             uint32_t config...
```

Terminal: Shows the log output from the PolarFire-SoC-Renode-emulation-platform:

```
14:23:46.0131 [DEBUG] gpio1: WriteUInt32 to 0xA4 (SetRegister), v
14:23:46.0131 [DEBUG] gpio1: WriteUInt32 to 0x00 (ClearRegister),
14:23:46.0131 [DEBUG] gpio1: WriteUInt32 to 0x00 (ClearRegister),
14:23:46.0141 [DEBUG] gpio1: WriteUInt32 to 0x00 (ClearRegister),
14:23:46.0141 [DEBUG] gpio1: WriteUInt32 to 0xA4 (SetRegister), v
14:23:46.0141 [DEBUG] gpio1: WriteUInt32 to 0xA4 (SetRegister), v
14:23:46.0141 [DEBUG] gpio1: WriteUInt32 to 0xA4 (SetRegister), v
```

DETOUR: IT'S ALL IN ZEPHYR

- Enable a broad set of features for available targets in a configurable manner - that's what a “batteries included” RTOS is about
- Zephyr RTOS exceeds in two aspects:
 - modularity
 - discoverability
 - now even more so with Hardware Model v2
- Device trees!
- How to leverage these?



ZEPHYR + RENODE = DASHBOARD

- Zephyr DTS descriptions and Renode used together for testing at scale:
- zephyr-dashboard.renode.io
- Presents auto-generated Zephyr examples and tests passing on 470+ platforms in Renode
 - approx. 80%
- Visualizes the data in a matrix view and allows for easy local reproduction
- Helps improve Zephyr as well - the RISC-V port, specific drivers, new demos, ...!

BOARD NAME	HELLO WORLD	PHILOSOPHERS	SHELL MODULE
RISCV64 (4)			
Beagleboard BeagleV-Fire	PASSED	PASSED	PASSED
Microchip PolarFire ICICLE kit	PASSED	PASSED	PASSED
SiFive HiFive Unleashed	PASSED	PASSED	PASSED
SiFive HiFive Unmatched	PASSED	PASSED	PASSED
RISCV32 (27)			
Andes ADP-XC7K AE350	PASSED	GENERATED	PASSED
ESP32-C3	BUILT	BUILT	BUILT
ESP32C3 LuatOS Core	BUILT	BUILT	BUILT
ESP32C3 LuatOS Core USB	BUILT	BUILT	BUILT
GigaDevice GD32VF103C-STARTER	GENERATED	GENERATED	GENERATED
GigaDevice GD32VF103V-EVAL	GENERATED	GENERATED	GENERATED
ICE-V Wireless	BUILT	BUILT	BUILT
INTEL FPGA Nios V/g general purpose processor	PASSED	GENERATED	GENERATED
INTEL FPGA niosv_m	PASSED	GENERATED	GENERATED

U-BOOT DASHBOARD

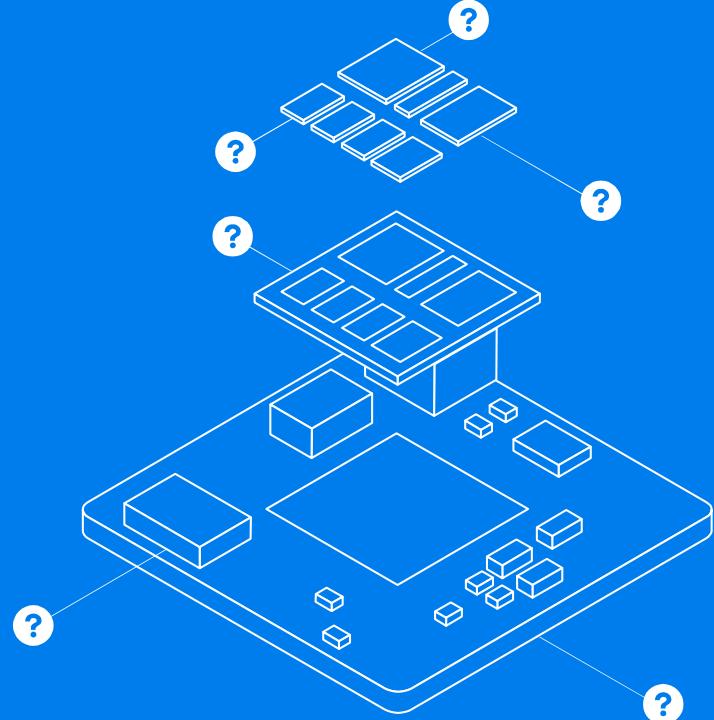
- This approach is not Zephyr-exclusive, it scales to other software as well!
- We need
 - a unified way to build software for different targets
 - machine-readable platform description
- u-boot-dashboard.renode.io
- Great source of extra data, especially for boards that are also supported in Zephyr

The screenshot shows a dashboard interface with a search bar at the top. In the top right corner, there is a green button labeled "148 PASSED". Below the search bar, there are two columns: "BOARD NAME" and "U-BOOT". A dropdown menu labeled "RISCV (21)" is open above the board list. The board list itself consists of 14 items, each with a name and a status indicator (green for passed, blue for built, or grey for not built). The boards listed are: SiFive HiFive Unleashed A00 (PASSED), Microchip PolarFire-SoC Icicle Kit (BUILT), openpiton_riscv64--openpiton-riscv64 (BUILT), openpiton_riscv64_spl--openpiton-riscv64 (BUILT), qemu-riscv64--qemu-virt64 (BUILT), qemu-riscv64_smode--qemu-virt64 (BUILT), qemu-riscv64_spl--qemu-virt64 (BUILT), SiFive HiFive Unmatched A00 (BUILT), Sipeed Lichee Pi 4A (BUILT), starfive_visionfive2--jh7110-starfive-visionfive-2 (BUILT), ae350_rv32--ae350_32 (NOT BUILT), ae350_rv32_spl--ae350_32 (NOT BUILT), ae350_rv32_spl_xip--ae350_32 (NOT BUILT), and ae350_rv32_xip--ae350_32 (NOT BUILT).

BOARD NAME	U-BOOT
SiFive HiFive Unleashed A00	PASSED
Microchip PolarFire-SoC Icicle Kit	BUILT
openpiton_riscv64--openpiton-riscv64	BUILT
openpiton_riscv64_spl--openpiton-riscv64	BUILT
qemu-riscv64--qemu-virt64	BUILT
qemu-riscv64_smode--qemu-virt64	BUILT
qemu-riscv64_spl--qemu-virt64	BUILT
SiFive HiFive Unmatched A00	BUILT
Sipeed Lichee Pi 4A	BUILT
starfive_visionfive2--jh7110-starfive-visionfive-2	BUILT
ae350_rv32--ae350_32	NOT BUILT
ae350_rv32_spl--ae350_32	NOT BUILT
ae350_rv32_spl_xip--ae350_32	NOT BUILT
ae350_rv32_xip--ae350_32	NOT BUILT

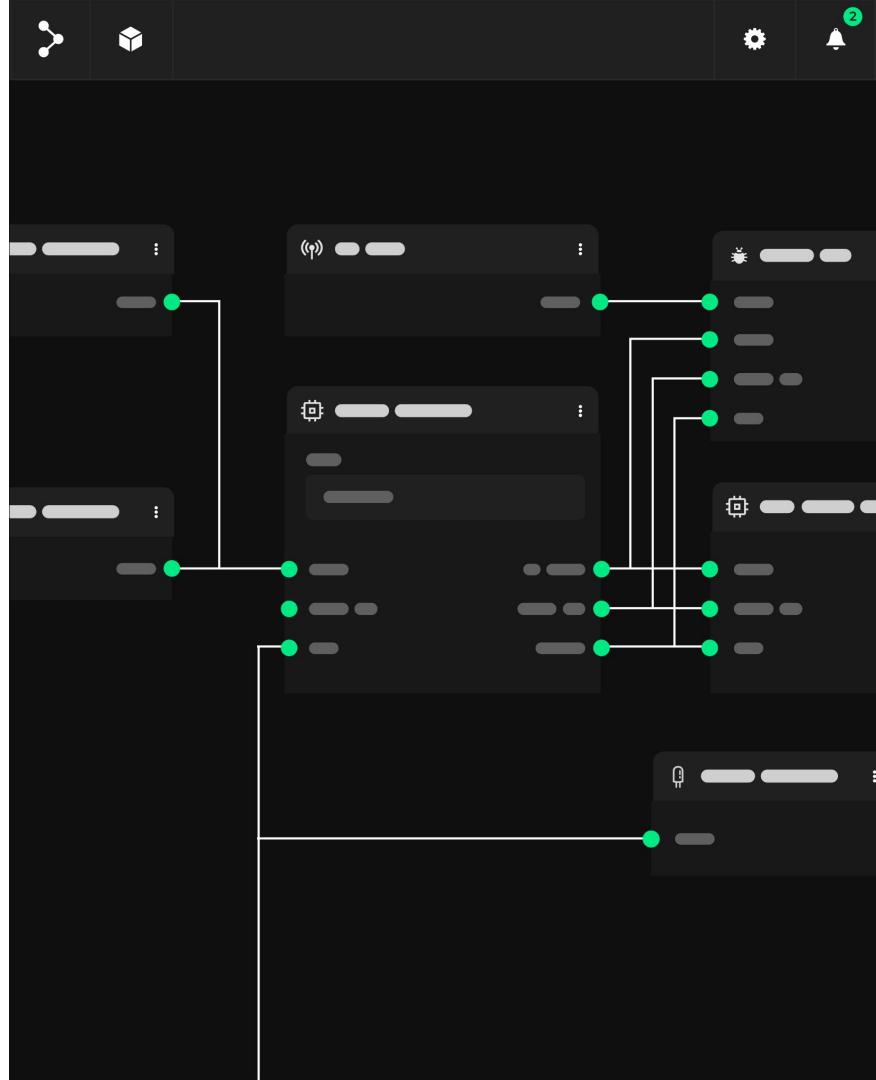
RENODEPEDIA

- [Renodepedia](#) - Antmicro's encyclopedia of hardware - SoCs, boards, IP blocks
- Cataloging HW ecosystem, exposing structure - especially great for open source hardware and RISC-V ecosystem
- Based on automated data e.g. device trees from the [Zephyr Dashboard CI](#):
 - Discover connections between components
 - Navigate dependencies
 - Show commonalities and SW payloads
 - Simulate in Renode



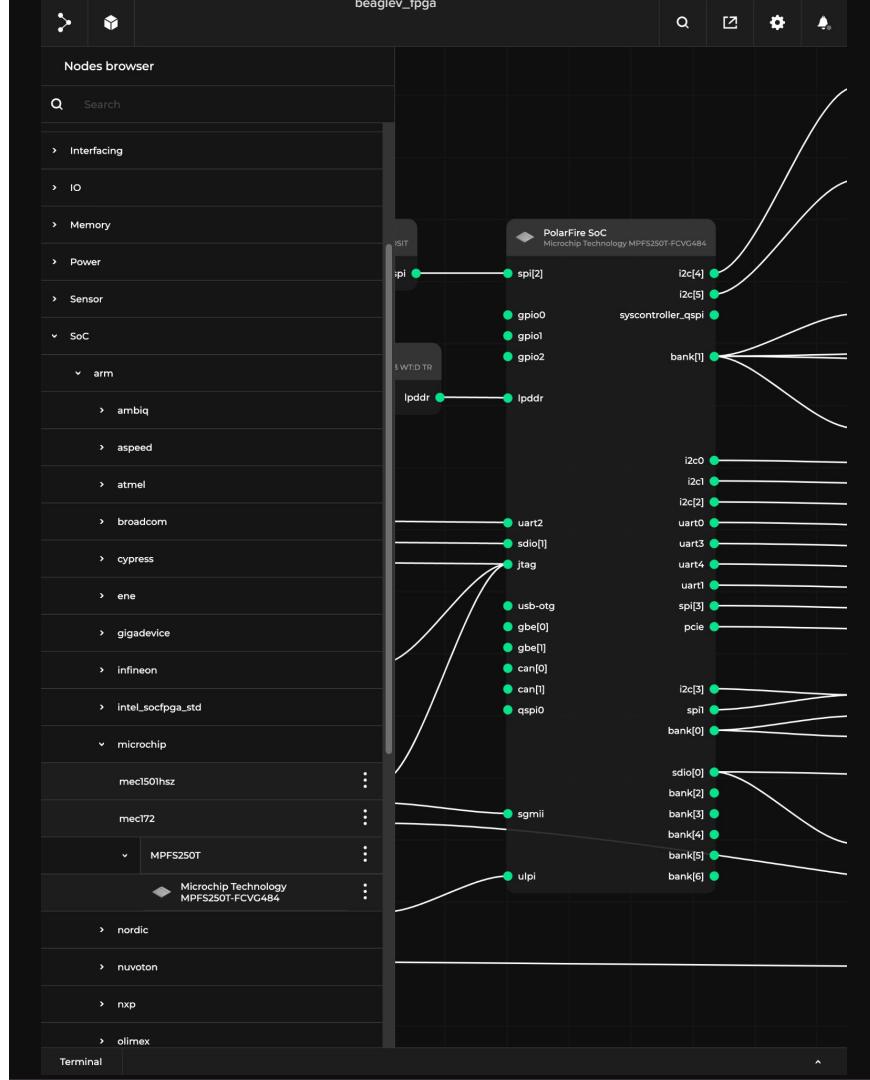
VISUAL SYSTEM DESIGNER

- We went pro-active with the available data and created VSD
- [Visual System Designer](#) enables a visual, formalized way to build embedded systems from defined building blocks
- An open source tool, based on Antmicro's open source component and platform database, data from Zephyr and other projects, and using our [Pipeline Manager visual diagramming tool](#)
- Design hardware block diagrams that use a unified specification file containing different component types



WORKING WITH BEAGLEV-FIRE

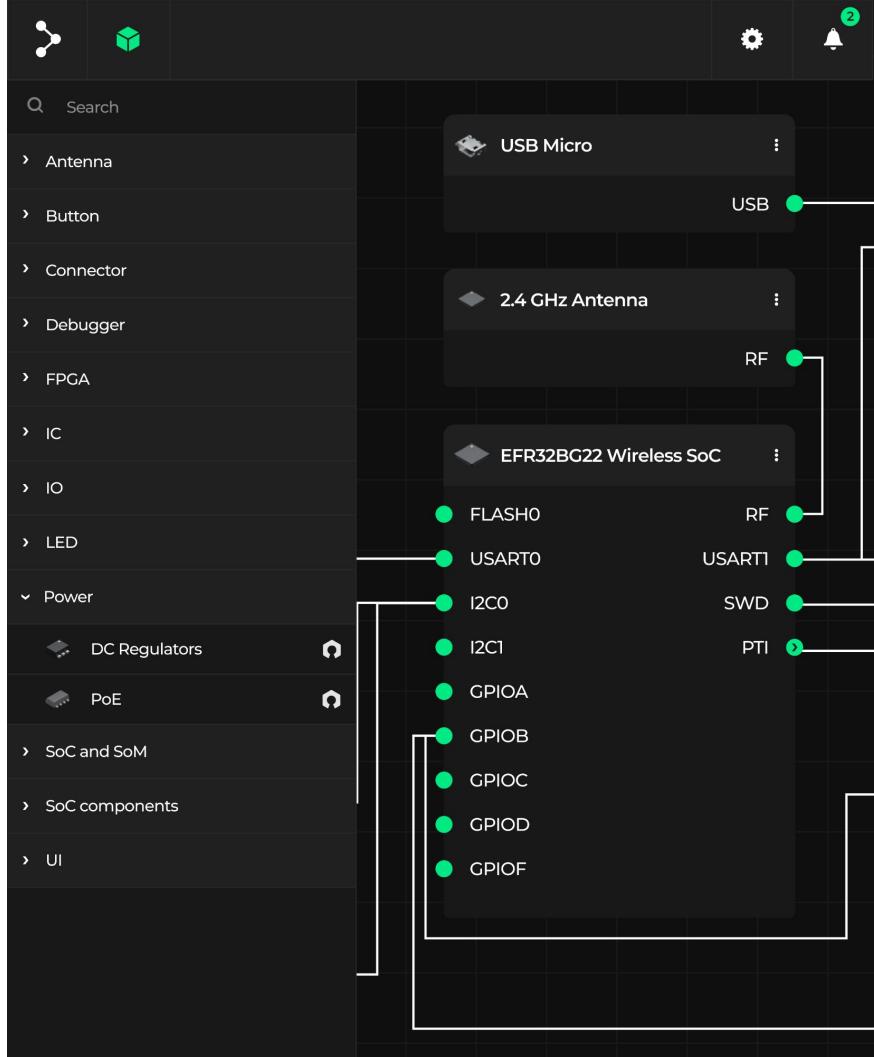
- Software engineers are not always comfortable with working with FPGAs
- Ever-growing library of open-source IP blocks helps us bring these worlds together
- FPGAs are programmed using RTL, such as Verilog
- Verilog can be compiled with Verilator and co-simulated with Renode
- It's not all that difficult - but what if we could hide the complexity?



DEMO

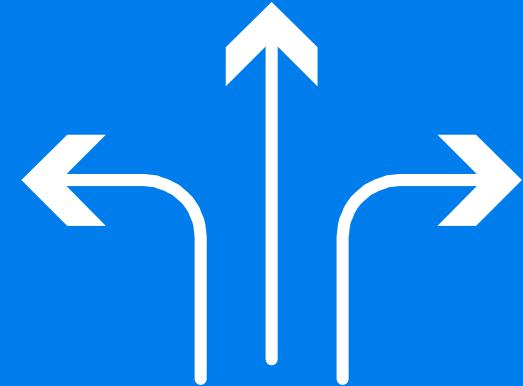
WHAT YOU JUST SAW?

- Leveraging FPGA capabilities of BeagleV-Fire
 - making the software part easier
- Software built specifically for target hardware configuration
- No vendor-specific flows, pure reliance on Zephyr data
- Open-source data and processing
- Easily expandable with your own data



WHERE FROM HERE?

- Expose diagram-based design / software generation capability to more users via a dedicated portal
- Enable more software scenarios
- IDE integration
- Vendor independence!
- Multiple levels of abstraction
 - SoC - drivers, build system,
 - board, device - sensors, interfaces initialization
 - complex system - network setup, services
- Reach out! contact@antmicro.com





antmicro

**THANK YOU
FOR YOUR ATTENTION!**

