Hardware/Software Co-Design with the Open Source Renode Framework and RISC-V

Getting Started With RISC-V NA Tour, 2019
Michael Gielda, mgielda@antmicro.com
Why HW/SW co-design?

- Design cycles are radically decreasing, and new HW platforms become more complex: features vs security vs safety vs ease-of-use
- More and more software is expected to be provided for out of the box experience
- Configurability and openness of RISC-V offers the promise of “software-driven” design...
Why HW/SW co-design?

• ... but new tools and methods are needed to deliver on the promise
• You can’t just build your platform sequentially any more, many elements need to be set in motion simultaneously and iterated on
• Cost of bug caught early is 1000x less than one caught further down the line
Develop your IoT product with Renode™:

GET STARTED
Continuous Integration based development

Company Environment

Local PC

Interactive test and debug in Renode

Get help from colleagues

Commit code

Develop with favorite IDE/compiler

Tests pass?

Tests with various configurations

CI e.g. with Robot + Renode

Merge changes

Push to server

Plane tests / deployment
Renode in short

- rapid development framework for building software including real, end-user applications
- open source Instruction Set Simulator (ISS) with a multi-layered framework on top
- strong background in practical use across multiple architectures (mainly but not only ARM & RISC-V)
- commercial backing from Antmicro - we use it ourselves extensively
Renode – why?

• system emulator - mimic entire boards or even multiple connected nodes
• most interesting name I heard: “hierarchical functional simulator” - ‘building-block’ nature
• scriptable, API-oriented, extremely flexible
• software agnostic - runs Zephyr, Linux, bare metal, proprietary SW: binary compatible (no special compilation targets)
**Easy to develop & prototype**

- lots of useful abstraction and interfaces
- human readable, modular and extensible platform description format
- plug-and-play blocks, Python stubs

```plaintext
uart: UART.MiV_CoreUART @ sysbus 0x70001000
clockFrequency: 66000000

cpu: CPU.RiscV @ sysbus
cpuType: "rv32g"

plic: Interrupts.PlatformLevelInterruptController @ sysbus 0x40000000
IRQ -> cpu@1
numberOfSources: 31 //based on release notes
```
Layer #3: Complex system

- Sensor nodes
- Gateway
Layer #2: The device
Layer #1: System-on-Chip

- CPU
  - U54 RV64GC Core
  - E51 RV64IMAC Core

- CAN
- Ethernet
- RAM
- UART
- SPI
- USB
- PCIe
- Flash
Enabling the Freedom to Innovate:
PolarFire SoC FPGA architecture

Microsemi
a Microchip company
Before – a USD 3000 development platform (hard to fit in carry-on luggage)
Now – Renode, a free and open source framework that’s in your PC (or server)

Get Renode™ for:

- **Debian**
  - Linux, (.deb)
  - Download

- **Fedora**
  - Linux, (.rpm)
  - Download

- **Arch**
  - Linux, (.pkg.tar.xz)
  - Download

- **macOS**
  - Download

- **Windows**
  - Download
PFSoC support

- Entire SoC complex
- include lots of I/O like USB, PCIe, CAN, I2C, SPI, GPIO, Eth...
- can model additional peripherals in the FPGA - easy to add new models as blocks
- integration with Verilator to actually co-simulate the IPs
PFSoC support highlights

• interfaces for multi-node connectivity
• also host-guest networking for Eth
• networking includes TSN/PTP
• analyze protocols, debug entire system at the same time
**PFSoc support highlights**

- interfaces for connecting e.g. sensors and actuators
- quite useful for building real boards that interact with the external world
It Works!

Temperature: 37.000
PFSoc support highlights

- interfaces for connecting with interesting external elements
- enable to really explore the flexibility of Renode
SoftConsole integration

- Standard IDE, comes bundled
- Linux and Windows
- examine the entire system as you’re developing code
- new and exciting abilities
SoftConsole integration

- Renode is extremely extendible
- Debug, tracing, visualisation - we have all the data
Significance of PolarFire SoC platform

- as an FPGA SoC, ideal for developing and prototyping new RISC-V hardware solutions
- custom accelerators in FPGA fabric
- Linux + real-time systems
- showing the way with an ultra-flexible pre-silicon development platform for all developers - Renode
Co-simulation with Verilator

- For IP you care about, you can simulate the real RTL with Verilator
- The co-simulated block works as a regular Renode block (of course much slower), driven from Renode with all APIs, only a small shim needed
- Divide and conquer your problem - simulate most of the system fast and limit HDL simulation to minimum
- Developing and extending this feature with multiple partners
Example: LiteX soft SoC

- open source configurable SoC, RISC-V option
- runs 32-bit Linux now!
- Renode model of core and peripherals e.g. Ethernet, UART etc.
- can build, for example, a simulated setup with multiple Ethernets
- easily add extra HDL peripherals through Verilator integration
HW/SW co-development:
Dover Microsystems

- Dover is developing CoreGuard™ cybersecurity silicon IP, used by customers such as NXP
- Renode was extended by Antmicro for Dover with fine-grained control of execution and debugging
- Dover uses Renode both in internal development as well as externally
Renode - a new approach to complex embedded systems development

HW/SW co-development: Dover Microsystems - uses

- Design space exploration
- Implementation
- Testing
- Demonstration and evaluation
Integrates with RISC processors to provide separate, sentry logic.

Extracts a set of trace signals from host processor

Monitors and evaluates every instruction in the host processor in real time

Provide mechanism for CoreGuard to affect a stall on the host when needed to evaluate policies

Exceptions thrown from CoreGuard when policy violated
A set of rules that express the allowed combinations of metadata for each possible CPU operation.

CoreGuard uses the metadata information today’s processors have been throwing away.

Metadata defines …

if the value is **private**

if the value is a **pointer**

what a pointer has **access** to

if the data is **executable**

if the reference word in memory is a **return address**
TWO CONTEXTS OF DOVER USING RENODE

CUSTOMER DELIVERABLE

- SoC has AP CPU + peripherals
- CoreGuard is entirely simulated in C# and C++
  - Policy code runs on host (e.g. X86)
- Via a Renode Plugin
  - Registers BlockBeginHook and BlockEndHook Renode hooks
  - Hooks call generated C++ code

INTERNAL DEVELOPMENT

- SoC has AP CPU + peripherals + CG PEX (RISC-V CPU), CG “accelerator”
- Policy code runs on simulated RISC-V
- Used to debug boot and runtime CG API
PARALLEL HW/SW DEVELOPMENT

Use Case

- HW spec developed between HW and SW teams
- SW team implements spec in Renode and writes firmware against spec, testing on Renode
- In parallel, HW is implementing and testing HW design
- Integration test via FPGA and/or HW simulator

EXAMPLE

HDMI device. We had SW working against spec, under Renode, in advance of HW.
Non ELF formats (scatter-load) used by HW boot ROMs
  ▪ Easy to test in Renode
Need full SoC simulated; copying from flash to RAM, initializing devices
  ▪ Can instrument Renode to flag erroneous bus traffic
  ▪ Full debugging under Renode (rarely the case at boot time on HW)
Peripherals (UARTs, PICs, our CoreGuard interface) can be instrumented to check for correct initialization and use
Playing with memory map
Before hardware design, can profile different cache hierarchies, data representations, etc.

Get “macro” numbers – cache hits vs. misses, number of indirections (memory pressure), etc.
Dover – effects

• Renode allows Dover to run thousands of tests daily, preventing regressions (particularly bad for security IP)
• Decreased turnaround for new feature prototyping from days/weeks to hours
• Enabled customers to easily test-drive the technology
Interested? Just get it from GitHub!

... and be sure to see our demo + talk to us
Renode development services

- Embedded systems development services powered by the RENODE methodology
- Professional support, implementing new platforms
- Building customized tools, user interfaces and integrations
- Setting up CI and improving development workflows in your organization
Summary

- Renode allows practical HW/SW co-design on many levels
- Its flexibility enables easy prototyping and practical adoption for RISC-V & mixed systems
- Microsemi (Microchip) has shown the way for enabling users with pre-silicon development
- Features like Verilator integration are designed to further strengthen the HW/SW co-design use-case
- Dover case shows how you can use Renode throughout the entire development cycle
A sneak peek into the future