

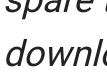
## News

← All news

 Search the archive

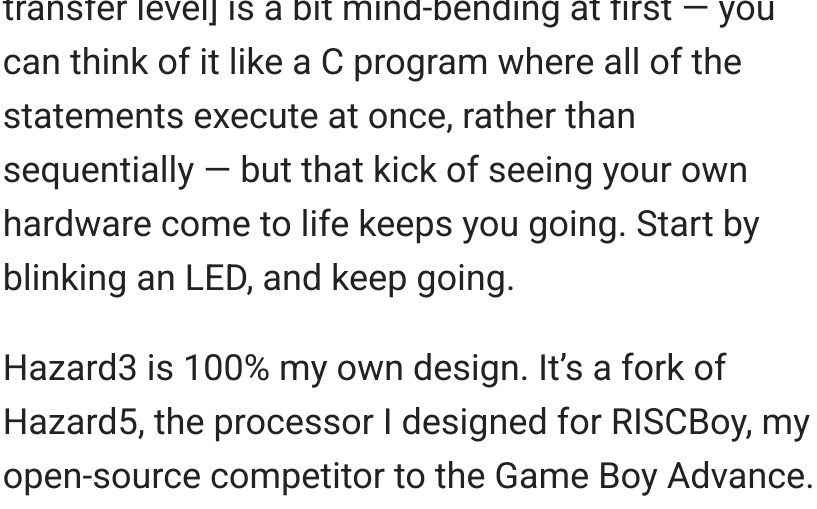
 RSS feed

# RISC-V on Raspberry Pi Pico 2

 25th Sep 2024 Luke Wren 3 comments

*The Hazard 3 RISC-V cores on the RP2350 were designed by Raspberry Pi's own Luke Wren in his spare time – and as they're open source, you can download the design files yourself and start poking around in the very same chip that will eventually be in use on millions of units out in the wild. As Eben Upton puts it: "In adding Hazard3 to [RP2350](#), we're aiming to give software developers a chance to experiment with the RISC-V architecture in a stable, well-supported environment, and to popularise Hazard3 as a clean, open core, suitable for verbatim use in other devices, or as a basis for further development." Luke's reflections first appeared in [issue 145 of The MagPi](#).*

I've been doing logic design in my spare time since I was a student. It's highly addictive, and I think it's more accurate to say I'm a hobbyist who works in chip design than a chip designer with a hobby! It's an open-source processor design that anyone can put in their chip and use to run RISC-V code anywhere. You can also run it on an FPGA board, or run the simulator on your own machine. It's all built using open-source tools like yosys, nextpnr and gtkwave.



The best way to get started is to get an FPGA board and just get hacking. Writing RTL [register transfer level] is a bit mind-bending at first – you can think of it like a C program where all of the statements execute at once, rather than sequentially – but that kick of seeing your own hardware come to life keeps you going. Start by blinking an LED, and keep going.

Hazard3 is 100% my own design. It's a fork of Hazard5, the processor I designed for RISCBoy, my open-source competitor to the Game Boy Advance. Hazard5 is a five-stage pipeline – therefore having many hazards: data flow, control flow and structural – and a hazard is also a kind of 'risk', like the instruction set.

Hazard5 was meant to run at the highest possible frequency on an iCE40 FPGA, so I could run the RISCBoy graphics core at a higher frequency too. Hazard3 on the other hand is a production-grade processor which delivers as much performance as possible in its small area envelope and within the range of frequencies I expect to see on microcontroller designs. It's a productionised version of Hazard5 with a shorter pipeline, hardware debug, and some security and memory protection features that people expect in real systems.

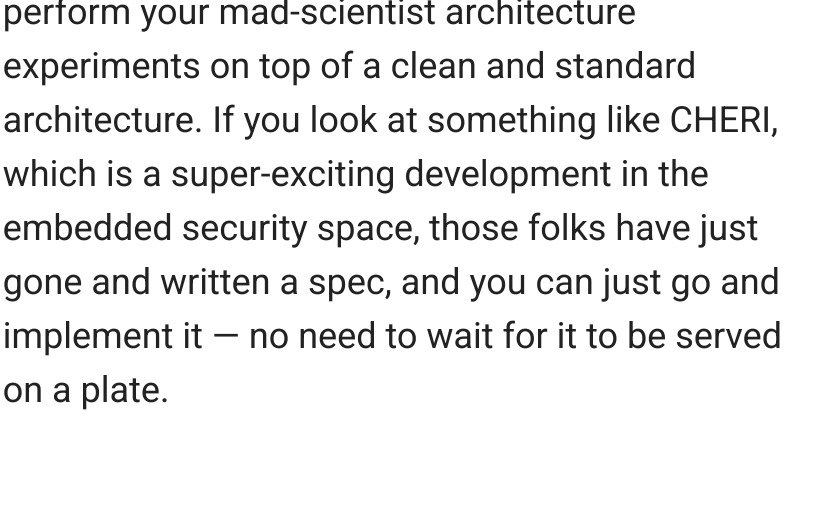


RP2350 on [Raspberry Pi Pico 2](#)

From forking Hazard5 to having Hazard3 running CoreMark took less than a week. From that point until the first [RP2350](#) tapeout was around two years, working on it on-and-off throughout. There is still ongoing maintenance work, and plans for future expansion – it will never be 'finished', just transition from development to stable releases.

Before I started working on RISCBoy I had a project called Tarantula which was an eight-thread barrel processor implementing the Armv6-M instruction set, because that was the ISA I was most familiar with at the time, having written some Assembly during a summer internship. I abandoned the project because I realised I would never be able to share it with anybody, and I don't think I even have that source code any more.





That experience changed how I looked at things from that point forward. When I decided I wanted to build a games console from scratch, including the processor, I looked around the instruction sets available at this point, this was around 2018, and there were a few interesting ones – Hitachi SuperH had just become much less legally restrictive – but RISC-V stood out as an instruction set where I could implement it fairly easily.



The base instruction set is quite clean and simple, and you can add more complexity from a menu of extensions. I could share that with other people, and they could actually use it, and I could program using a real production-grade compiler like GCC or LLVM.

That was a long time ago, and RISC-V has come a long way since, both technically and as a community. There are other instruction sets that have become more open in the wake of RISC-V but I think it's clear where the momentum is. It's easy to criticise some of the technical decisions made in the base ISA – did we really need 31 link registers? – but the community is the most important thing in my eyes.

I am excited about RISC-V because it lets you perform your mad-scientist architecture experiments on top of a clean and standard architecture. If you look at something like CHERI, which is a super-exciting development in the embedded security space, those folks have just gone and written a spec, and you can just go and implement it – no need to wait for it to be served on a plate.

Share this post    

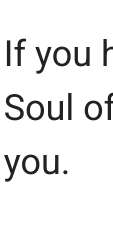
Raspberry Pi Pico 2

RISC-V

RP2350

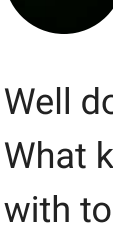
## 3 comments

[Jump to the comment form](#)

- 


**NEUBER JOSE DE SOUSA**  
26th September 2024, [2:20 am](#)

Hi  
What about RemoteProc  
Bro

[Reply to NEUBER JOSE DE SOUSA](#)
- 

**A Marchini**  
5th October 2024, [12:59 am](#)

If you haven't already, you should read "The Soul of a New Machine". It might be fun for you.

[Reply to A Marchini](#)
- 

**Anthony King**  
1st February 2025, [2:00 am](#)

Well done Luke.  
What kind of building blocks do you work with to create a processor core?

[Reply to Anthony King](#)

## Leave a Comment

**Name (Required)**

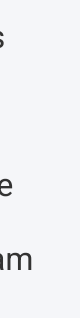
**Email address (also required)**

**Website URL**

**Your comment (required)**

☐


I am human

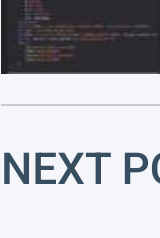


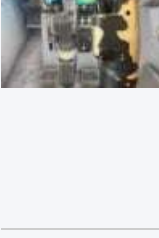
hCaptcha  
Privacy - Terms

**Post Comment**

## RELATED POSTS


- 

**RP2350: the brains of Raspberry Pi Pico 2**
- 

**What's new in Raspberry Pi Pico 2**
- 

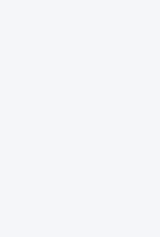
**Rust on RP2350**

## NEXT POST

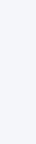
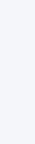
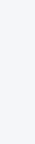
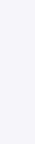
- 

**Raspberry Pi Pico brings junked joysticks back to life | The MagPi #146**


## PREVIOUS POST

- 


**OUT NOW: The Official Raspberry Pi Handbook 2025 has landed**


Share this post    


### Follow us


 Sign up to newsletter


 X


 Threads

 TikTok

 YouTube

 Instagram

 LinkedIn

 Facebook

### About Raspberry Pi

- News
- Investor relations
- Contact us
- Trademark
- About us
- Our Approved Resellers
- Jobs
- Accessibility
- Modern slavery statement
- Site use terms and conditions
- Acceptable use
- Cookies
- Licensing
- Terms and conditions of sale
- Privacy
- Security
- Verify our bank details

### Software

- Raspberry Pi Connect
- Raspberry Pi Desktop
- Raspberry Pi Imager
- Raspberry Pi OS

### Documentation

- All categories
- Product information portal
- Datasheets

### Community

- Forums
- Events

### Raspberry Pi Store

- Store information

### Raspberry Pi Press

- About Raspberry Pi Press
- Raspberry Pi Official Magazine
- Books

### For home

- Raspberry Pi for home
- Tutorials

### For industry

- Raspberry Pi for industry
- Thin clients
- Raspberry Pi in space
- Powered by Raspberry Pi
- Design partners
- Success stories

### Hardware

- Computers and microcontrollers
- Cameras and displays
- Add-on boards
- Power supplies and cables
- Cases
- Peripherals