Aligned on RISC-V

Daniel Maslowski
Agenda

- Intro and current status
- Aligning with others
- Unaligned memory access
Hey Again OSFC!
Remember oreboot?
Remember oreboot?

*oreboot is a fork of coreboot…*
oreboot is platform initialization firmware in Rust!

oreboot is a fork of coreboot, with C removed, written in Rust.

https://github.com/oreboot
State of coreboot on RISC-V in 2023 / Q4

Looking at the docs…

Whoops, a bit out of sync with the current spec — won’t work!
State of coreboot on RISC-V in 2023 / Q4

Looking at the docs…
State of coreboot on RISC-V in 2023 / Q4

Looking at the docs…

QEMU RISC-V emulator

Building coreboot and running it in QEMU

- Configure coreboot and run \texttt{make} as usual
- Run \texttt{util/riscv/make-spike-elf.sh} \texttt{build/coreboot.rom} \texttt{build/coreboot.elf} to convert coreboot to an ELF that QEMU can load
- Run \texttt{qemu-system-riscv 64 -M virt -m 1024M -nographic -kernel build/coreboot.elf}

Whoops, a bit out of sync with the current spec - won’t work!
WIP for coreboot on RISC-V\(^1\)

We had a long road to go for GSoC 2023…

A lot of work still remains open.

\(^1\)Screenshot taken on 2023-10-06
State of oreboot on RISC-V in 2023 / Q4

- Support for 3 SoCs
  - Allwinner D1, now also with SD card
  - StarFive JH7100 - we have DRAM init
  - StarFive JH7110 - DRAM init, SMP is WIP, we can boot Linux
- RISC-V SBI (Supervisor Binary Interface) factored out from D1
- WIP for wider use on other SoCs, e.g., JH7110
- Now getting HSM (Hart State Machine) compression library factored out for reuse
- Build system based on xtask extended
- Website and documentation are still open.
State of oreboot on RISC-V in 2023 / Q4

- Support for 3 SoCs
  - Allwinner D1, now also with SD card
  - StarFive JH7100 - we have DRAM init
  - StarFive JH7110 - DRAM init, SMP is WIP, we can boot Linux

Website and documentation are still open.
State of oreboot on RISC-V in 2023 / Q4

- support for 3 SoCs
  - Allwinner D1, now also with SD card
  - StarFive JH7100 - we have DRAM init
  - StarFive JH7110 - DRAM init, SMP is WIP, we can boot Linux

- RISC-V SBI (*Supervisor Binary Interface*) factored out from D1
  - [https://github.com/riscv-non-isa/riscv-sbi-doc](https://github.com/riscv-non-isa/riscv-sbi-doc)
  - WIP for wider use on other SoCs, e.g., JH7110
  - now getting HSM (Hart State Machine)
State of oreboot on RISC-V in 2023 / Q4

- support for 3 SoCs
  - Allwinner D1, now also with SD card
  - StarFive JH7100 - we have DRAM init
  - StarFive JH7110 - DRAM init, SMP is WIP, we can boot Linux

- RISC-V SBI (*Supervisor Binary Interface*) factored out from D1
  - [https://github.com/riscv-non-isa/riscv-sbi-doc](https://github.com/riscv-non-isa/riscv-sbi-doc)
  - WIP for wider use on other SoCs, e.g., JH7110
  - now getting HSM (Hart State Machine)

- compression library factored out for reuse
- build system based on *xtask* extended

Website and documentation are still open.
State of oreboot on RISC-V in 2023 / Q4

- support for 3 SoCs
  - Allwinner D1, now also with SD card
  - StarFive JH7100 - we have DRAM init
  - StarFive JH7110 - DRAM init, SMP is WIP, we can boot Linux

- RISC-V SBI (Supervisor Binary Interface) factored out from D1
  - https://github.com/riscv-non-isa/riscv-sbi-doc
  - WIP for wider use on other SoCs, e.g., JH7110
  - now getting HSM (Hart State Machine)

- compression library factored out for reuse
- build system based on xtask extended

Website and documentation are still open.
Aligning with others
Aligning with mainline

Why aligning with open source mainline is the way to go

By Neill Whillans

You’ve made a choice to use open-source software as part of your product release. That’s a great start. Open-source software projects usually have large contributing communities that help improve the software’s quality and functionality over time. They are also usually willing to provide suggestions on how to proceed if you happen to run into difficulties during your development process.

https://www.codethink.co.uk/articles/2021/why-aligning-with-open-source-mainline-is-the-way-to-go/
Once upon a time, you could design and build a device, flash a well-engineered firmware onto it and support it for 20 years, with minimal changes to the core software. Those days are over. A new approach is needed to provide long-term support for devices while managing the complexity of today's embedded software stacks.

https://www.codethink.co.uk/articles/2023/ltm-2023/
Once upon a time, you could design and build a device, flash a well-engineered firmware onto it and support it for 20 years, with minimal changes to the core software. Those days are over. A new approach is needed to provide long-term support for devices while managing the complexity of today’s embedded software stacks.

https://www.codethink.co.uk/articles/2023/ltm-2023/

A common understanding, common base, and common concepts are crucial.

Let’s talk and find those!
1.2 RISC-V Software Execution Environments and Harts

The behavior of a RISC-V program depends on the execution environment in which it runs. A RISC-V execution environment interface (EEI) defines the initial state of the program, the number and type of harts in the environment including the privilege modes supported by the harts, the accessibility and attributes of memory and I/O regions, the behavior of all legal instructions executed on each hart (i.e., the ISA is one component of the EEI), and the handling of any interrupts or exceptions raised during execution including environment calls. Examples of EEIs include the Linux application binary interface (ABI), or the RISC-V supervisor binary interface (SBI). The implementation of a RISC-V execution environment can be pure hardware, pure software, or a combination of hardware and software. For example, opcode traps and software emulation can be used to implement functionality not provided in hardware. Examples of execution environment implementations include:

- “Bare metal” hardware platforms where harts are directly implemented by physical processor threads and instructions have full access to the physical address space. The hardware platform defines an execution environment that begins at power-on reset.
Stacks as per Spec

Figure 1.1: Different implementation stacks supporting various forms of privileged execution.

Note: Lower privilege modes have less access to the platform.

https://riscv.org/technical/specifications/
Our stack: oreboot and LinuxBoot

Porting oreboot to the VisionFive1 board / JH7100 SoC

- MMIO mapped
  - mask ROM
  - second boot
  - DDR init
  - OpenSBI
  - U-Boot proper

- 128K SRAM 1
  - mask ROM
  - oreboot bt0
  - DDR init
  - OpenSBI
  - U-Boot proper

- 128K SRAM 2
  - mask ROM
  - oreboot bt0
  - DDR init
  - OpenSBI
  - U-Boot proper

- 8G DRAM
  - mask ROM
  - oreboot bt0 with DRAM init and RustSBI
  - OpenSBI
  - U-Boot proper
  - LinuxBoot

oreboot + RustSBI = SEE (Supervisor Execution Environment)

---

Rising Interest in oreboot and LinuxBoot

Why don't they collaborate with Oreboot project?

Someone from Intel contacted me:

ByteDance and many hyperscalers are using LinuxBoot.
Rising Interest in oreboot and LinuxBoot

https://www.reddit.com/r/RISCV/comments/13ksvsz/comment/jkm63qh/
 Why don’t they collaborate with Oreboot project?
Rising Interest in oreboot and LinuxBoot

https://www.reddit.com/r/RISCV/comments/13ksvsz/comment/jkm63qh/
Why don’t they collaborate with Oreboot project?

https://forum.rvspace.org/t/oreboot-visionfive-2-support-effort/2211
Rising Interest in oreboot and LinuxBoot

https://www.reddit.com/r/RISCV/comments/13ksvsz/comment/jkm63qh/
*Why don’t they collaborate with Oreboot project?*

https://forum.rvspace.org/t/oreboot-visionfive-2-support-effort/2211

Someone from Intel contacted me:

> Now I am also curious what your interest in the project is.
> Are you working with Rust, and would you like to contribute?
> Or are you mainly looking for integrating with LinuxBoot?

We are enabling UEFI on RISC-V, currently most work are based on the u-boot-spl + edk2 solution.
At the same time, the coreboot + linuxboot solution is also very popular, intel has full reference firmware stack on x86 platform
That's why I'm also trying to understand the current status of these firmware solutions on RISC-V
We are a new team and we are at the very beginning in firmware solution development, and our work will open source and contribute to software ecosystem project with community.
I believe RUST is an important language for firmware development.
And I think it is very be a very exciting thing to have full a RUST based UEFI firmware solution for high-end RISC-V PC and server market, especially for security features!
Let's see

---

3https://www.youtube.com/watch?v=gB3wgOuvLJQ
Rising Interest in oreboot and LinuxBoot

https://www.reddit.com/r/RISCV/comments/13ksvsz/comment/jkm63qh/
Why don’t they collaborate with Oreboot project?

https://forum.rvspace.org/t/oreboot-visionfive-2-support-effort/2211

Someone from Intel contacted me:

Now I am also curious what your interest in the project is.
Are you working with Rust, and would you like to contribute?
Or are you mainly looking for integrating with LinuxBoot?

We are enabling UEFI on RISC-V, currently most work are based on the u-boot-spl + edk2 solution.
At the same time, the coreboot + linuxboot solution is also very popular, intel has full reference firmware stack on x86 platform
That's why I'm also trying to understand the current status of these firmware solutions on RISC-V
We are a new team and we are at the very beginning in firmware solution development, and our work will open source and contribute to software ecosystem project with community.
I believe RUST is an important language for firmware development.
And I think it is very be a very exciting thing to have full a RUST based UEFI firmware solution for high-end RISC-V PC and server market, especially for security features!
Let's see

ByteDance\(^3\) and many hyperscalers are using LinuxBoot.

\(^3\)https://www.youtube.com/watch?v=gB3wgOuvLJQ
Community tracking upstream

Overview

The idea is to submit the code needed to run the Linux kernel on Allwinner SoC upstream, i.e. to the official Linux kernel.
This can be achieved by following the concept outlined in the Your new ARM SoC Linux support checklist article published by Thomas Petazzoni from Beets.
Where relevant, I have attempted to include who is currently working on an item, mostly separate from any particular mainlining goal.

Status

The Mainline Kernel backports contains the currently used backports for the mainlining process. The U-Boot repository and toxchain is described in the Mainline U-Boot backports. The Mainline Kernel category gives an overview of currently supported devices.

Status Matrix

The goal of this matrix is to give an easy view of work on each SoC worked on by Linus.

<table>
<thead>
<tr>
<th>Model</th>
<th>F525</th>
<th>X86</th>
<th>A10</th>
<th>A13</th>
<th>AM3</th>
<th>ORB</th>
<th>A20</th>
<th>T2</th>
<th>R40</th>
<th>T3</th>
<th>A48H</th>
<th>A80</th>
<th>A11</th>
<th>A23</th>
<th>A23</th>
<th>R16</th>
<th>A10T</th>
<th>A5</th>
<th>H3</th>
<th>A64</th>
<th>H5</th>
<th>H8</th>
<th>A50</th>
<th>V10</th>
<th>V93</th>
<th>H108</th>
<th>A100</th>
<th>A133</th>
<th>R129</th>
<th>D14</th>
<th>D11</th>
<th>T13</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1ADC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thermal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Touch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC97</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analog Codec</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio Hub</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPI0F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camera</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BT666</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPU C0-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paschal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAN bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Codec</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPUInfo (DVF5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPUInfo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Each component - kernel, U-Boot and OpenSBI - is tracked, driver for driver, with a link to the respective patch series in the mailing list.

https://rvspace.org/en/project/JH7110_Upstream_Plan
Unaligned Memory Access
When it goes wrong - userspace
When it goes wrong - userspace

Note: fixed in Go 1.21
When it goes wrong - kernel space
When it goes wrong - kernel space

Note: both `kexec` and module loader perform unaligned access as of now
Alignment on Real Devices
Alignment on Real Devices

D1 / T-Head can handle unaligned access in hardware, opt-out possible
Alignment on Real Devices

- D1 / T-Head can handle unaligned access in hardware, opt-out possible
- JH7110 / SiFive does not handle it and raises misaligned exceptions

OpenSBI has a trap handler for it, meaning big performance penalty

https://github.com/riscv-software-src/opensbi/blob/0ad866067d7853683d88c10ea9269ae6001bcf6f/lib/sbi/sbi_misaligned_ldst.c#L162

▶ It runs in M-mode, so context switches are involved
▶ Someone in SiFive forums measured 350x impact

https://forums.sifive.com/t/ld-sd-alignment/5530/2

We delegate all exceptions in no reboots so it is up to the OS

▶ Users may choose more suitable hardware per case
▶ Not every OS would want to do lots of probing etc.
Alignment on Real Devices

- D1 / T-Head can handle unaligned access in hardware, opt-out possible
- JH7110 / SiFive does not handle it and raises misaligned exceptions
- OpenSBI has a trap handler for it, meaning big performance penalty
  https://github.com/riscv-software-src/opensbi/blob/0ad866067d7853683d88c10ea9269ae6001bdf6f/lib/sbi/sbi_misaligned_ldst.c#L162
  ▶️ It runs in M-mode, so context switches are involved
  ▶️ Someone in SiFive forums measured 350x impact
  https://forums.sifive.com/t/ld-sd-alignment/5530/2

We delegate all exception in oes reboot so it is up to the OS
▶️ Users may choose more suitable hardware per case
▶️ Not every OS would want to do lot of probing etc.
Alignment on Real Devices

- D1 / T-Head can handle unaligned access in hardware, opt-out possible
- JH7110 / SiFive does not handle it and raises misaligned exceptions
- OpenSBI has a trap handler for it, meaning big performance penalty
  
  https://github.com/riscv-software-src/opensbi/blob/0ad866067d7853683d88c10ea9269ae6001bcf6f/lib/sbi/sbi_misaligned_ldst.c#L162
  
  ▶ It runs in M-mode, so context switches are involved
  ▶ Someone in SiFive forums measured 350x impact
    https://forums.sifive.com/t/ld-sd-alignment/5530/2

We delegate all exceptions in preboot so it is up to the OS

▶ Users may choose more suitable hardware per case
▶ Not every OS would want to do lots of probing etc
Not everything worked out yet: Platform

Proposed SBI extension to turn delegation on or off

Question raised how that would work with hypervisors

Pending PR for platform spec

Describemisaligneddataaccessscenarios

Optimization not worth the penalty?

RISC-V GCC RFC: "-mstrict-align" argument, and unaligned access tuning

https://lists.riscv.org/g/tech-prs/message/569

https://github.com/riscv/riscv-platform-specs/pull/2

https://groups.google.com/a/groups.riscv.org/g/sw-dev/c/bmQJ8HVmquY/m/QDmPndFlCwAJ
Not everything worked out yet: Platform

To the OS, it matters which traps it needs to handle.
Not everything worked out yet: Platform

To the OS, it matters which traps it needs to handle.

Proposed SBI extension to turn delegation on or off

Question raised how that would work with hypervisors
https://lists.riscv.org/g/tech-prs/message/569
Not everything worked out yet: Platform

To the OS, it matters which traps it needs to handle.

Proposed SBI extension to turn delegation on or off

Question raised how that would work with hypervisors
https://lists.riscv.org/g/tech-prs/message/569

Pending PR for platform spec
https://github.com/riscv/riscv-platform-specs/pull/2

*Describe misaligned data access scenarios*
Not everything worked out yet: Platform

To the OS, it matters which traps it needs to handle.

Proposed SBI extension to turn delegation on or off

Question raised how that would work with hypervisors
https://lists.riscv.org/g/tech-prs/message/569

Pending PR for platform spec
https://github.com/riscv/riscv-platform-specs/pull/2
  Describe misaligned data access scenarios

Optimization not worth the penalty?
https://groups.google.com/a/groups.riscv.org/g/sw-dev/c/bmQJ8HVmquY/m/QDmPndFlCwAJ
  RISC-V GCC RFC: "-mstrict-align" argument, and unaligned access tuning
Unaligned access in S-Mode / Linux
Unaligned access in S-Mode / Linux

Documentation/riscv/uabi.rst

Misaligned accesses are supported in userspace, but they may perform poorly.
Misaligned accesses are supported in userspace, but they may perform poorly.

...supported?

[RFC V2 PATCH 0/9] Add support to handle misaligned accesses in S-mode

https://lore.kernel.org/all/20230704140924.315594-1-cleger@rivosinc.com/
Misaligned accesses are supported in userspace, but they may perform poorly.

...supported?

[RFC V2 PATCH 0/9] Add support to handle misaligned accesses in S-mode

https://lore.kernel.org/all/20230704140924.315594-1-cleger@rivosinc.com/

Got a short remark, stuck for more than 2 months.
Unaligned access in S-Mode / Linux

Documentation/riscv/uabi.rst

Misaligned accesses are supported in userspace, but they may perform poorly.

…supported?

[RFC V2 PATCH 0/9] Add support to handle misaligned accesses in S-mode

https://lore.kernel.org/all/20230704140924.315594-1-cleger@rivosinc.com/

Got a short remark, stuck for more than 2 months.

So I talked to people at Kernel Recipes. :-(
Linux: Discussion picked up again

From: Clément Léger @ 2023-10-04 15:13 UTC (permalink / raw)
To: Paul Walmsley, Palmer Dabbel, Albert Ou
Cc: Clément Léger, Atish Patra, Andrew Jones, Evan Green,
    Björn Topel, linux-riscv, linux-kernel, Ron Minnich,
    Daniel Maslowski, Conor Dooley

Since commit 51cadb9 ("Provide new description of misaligned load/store behavior compatible with privileged architecture.") in the RISC-V ISA manual, it is stated that misaligned load/store might not be supported. However, the RISC-V kernel uABI describes that misaligned accesses are supported. In order to support that, this series adds support for S-mode handling of misaligned accesses as well support for prctl(PR_UNALIGN).

Handling misaligned access in kernel allows for a finer grain control of the misaligned accesses behavior, and thanks to the prctl() call, can allow disabling misaligned access emulation to generate SIGBUS. User space can then optimize its software by removing such access based on SIGBUS generation.

This series is useful when using a SBI implementation that does not handle misaligned traps as well as detecting misaligned accesses generated by userspace application using the prctl(PR_SET_UNALIGN) feature.

This series can be tested using the spike simulator[1] and a modified openSBI version[2] which allows to always delegate misaligned load/store to S-mode. A test[3] that exercise various instructions/registers can be executed to verify the unaligned access support.


https://lore.kernel.org/linux-riscv/20231004151405.521596-1-cleger@rivosinc.com/T/
Alignment in RISC-V ISA Spec: What may and may not

An EEI may guarantee that misaligned loads and stores are fully supported, and so the software running inside the execution environment will never experience a contained or fatal address-misaligned trap. In this case, the misaligned loads and stores can be handled in hardware, or via an invisible trap into the execution environment implementation, or possibly a combination of hardware and invisible trap depending on address.

An EEI may not guarantee misaligned loads and stores are handled invisibly. In this case, loads and stores that are not naturally aligned may either complete execution successfully or raise an exception. The exception raised can be either an address-misaligned exception or an access-fault exception. For a memory access that would otherwise be able to complete except for the misalignment, an access exception can be raised instead of an address-misaligned exception if the misaligned access should not be emulated, e.g., if accesses to the memory region have side effects. When an EEI does not guarantee misaligned loads and stores are handled invisibly, the EEI must define if exceptions caused by address misalignment result in a contained trap (allowing software running inside the execution environment to handle the trap) or a fatal trap (terminating execution).

Awesome Demo?
Keep in mind that many people are working on many layers. Props to Clément Léger for the Linux patches; it's a big change. Special thanks to Björn Töpelf for ping Clément: ‑) … and to everyone contributing to fixing the gaps!
Keep in mind that many people are working on many layers.
Keep in mind that many people are working on many layers.

Props to Clément Léger for the Linux patches; it’s a big change.
Keep in mind that many people are working on many layers.

Props to Clément Léger for the Linux patches; it’s a big change.

Special thanks to Björn Töpel for pinging Clément :-)

Keep in mind that many people are working on many layers.

Props to Clément Léger for the Linux patches; it’s a big change.

Special thanks to Björn Töpel for pinging Clément :-)

… and to everyone contributing to fixing the gaps!
kexec still needs some work
VisionFive 2 trouble with Device Tree + MDIO

Many files are involved - hard to find the issue.

MDIO is the communication between ethernet MAC and PHY.
Misaligned accesses are occasionally required when porting legacy code, and help performance on applications when using any form of packed-SIMD extension or handling externally packed data structures. Our rationale for allowing EEIs to choose to support misaligned accesses via the regular load and store instructions is to simplify the addition of misaligned hardware support. One option would have been to disallow misaligned accesses in the base ISA and then provide some separate ISA support for misaligned accesses, either special instructions to help software handle misaligned accesses or a new hardware addressing mode for misaligned accesses. Special instructions are difficult to use, complicate the ISA, and often add new processor state (e.g., SPARC VIS align address offset register) or complicate access to existing processor state (e.g., MIPS LWL/LWR partial register writes). In addition, for loop-oriented packed-SIMD code, the extra overhead when operands are misaligned motivates software to provide multiple forms of loop depending on operand alignment, which compiles code generation and adds to loop startup overhead. New misaligned hardware addressing modes take considerable space in the instruction encoding or require very simplified addressing modes (e.g., register indirect only).
Alignment in RISC-V Privileged Spec

3.6.3.3 Alignment

Memory regions that support aligned LR/SC or aligned AMOs might also support misaligned LR/SC or misaligned AMOs for some addresses and access widths. If, for a given address and access width, a misaligned LR/SC or AMO generates an address-misaligned exception, then all loads, stores, LRs/SCs, and AMOs using that address and access width must generate address-misaligned exceptions.

The standard “A” extension does not support misaligned AMOs or LR/SC pairs. Support for misaligned AMOs is provided by the standard “Zam” extension. Support for misaligned LR/SC sequences is not currently standardized, so LR and SC to misaligned addresses must raise an exception.

Mandating that misaligned loads and stores raise address-misaligned exceptions wherever misaligned AMOs raise address-misaligned exceptions permits the emulation of misaligned AMOs in an M-mode trap handler. The handler guarantees atomicity by acquiring a global mutex and emulating the access within the critical section. Provided that the handler for misaligned loads and stores uses the same mutex, all accesses to a given address that use the same word size will be mutually atomic.

Implementations may raise access-fault exceptions instead of address-misaligned exceptions for some misaligned accesses, indicating the instruction should not be emulated by a trap handler. If, for a given address and access width, all misaligned LRs/SCs and AMOs generate access-fault exceptions, then regular misaligned loads and stores using the same address and access width are not required to execute atomically.
Thank you!
Follow Me

https://github.com/orangecms
https://twitter.com/orangecms
https://mastodon.social/cyrevolt
https://twitch.tv/cyrevolt
https://youtube.com/@cyrevolt

Daniel Maslowski
https://github.com/oreboot/oreboot