Aligned on RISC-V

Daniel Maslowski

Agenda

- Intro and current status
- Aligning with others
- Unaligned memory access

Hey Again OSFC!

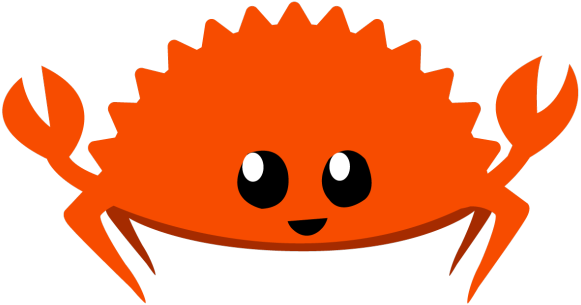
Remember oreboot?

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

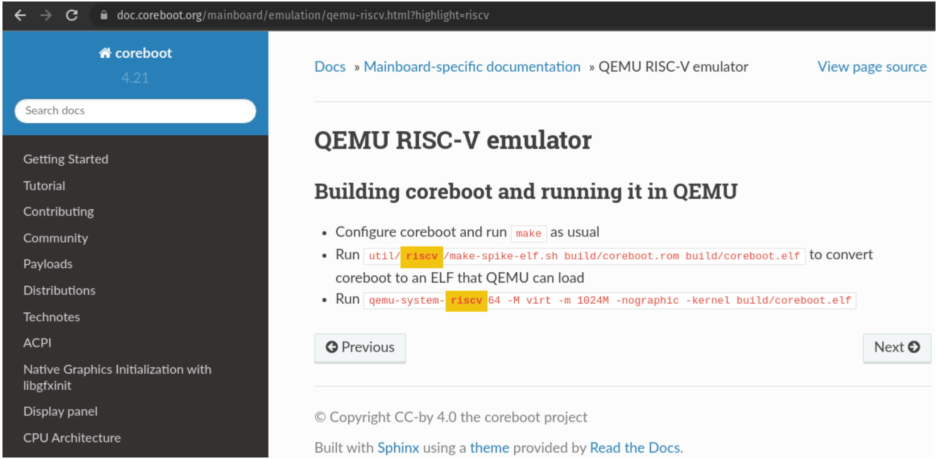
State of coreboot on RISC-V in 2023 / Q4

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…



doc.coreboot.org/mainboard/emulation/qemu-riscv.html?highlight=riscv

coreboot

4.21

Search docs

- Getting Started
- Tutorial
- Contributing
- Community
- Payloads
- Distributions
- Technotes
- ACPI
- Native Graphics Initialization with libgfxinit
- Display panel
- CPU Architecture

Docs » Mainboard-specific documentation » QEMU RISC-V emulator

View page source

QEMU RISC-V emulator

Building coreboot and running it in QEMU

- Configure coreboot and run `make` as usual
- Run `util/riscv/make-spike-elf.sh build/coreboot.rom build/coreboot.elf` to convert coreboot to an ELF that QEMU can load
- Run `qemu-system-riscv64 -M virt -m 1024M -nographic -kernel build/coreboot.elf`

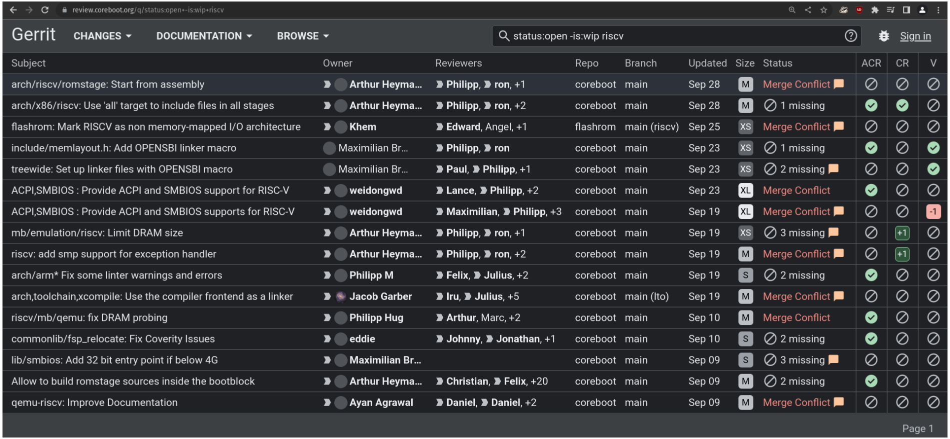
© Copyright CC-by 4.0 the coreboot project

Built with Sphinx using a theme provided by Read the Docs.

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...



review.coreboot.org/q/status:open+-is:wip+riscv

Subject	Owner	Reviewers	Repo	Branch	Updated	Size	Status	ACR	CR	V
arch/riscv/romstage: Start from assembly	Arthur Heyma...	Philipp, ron, +1	coreboot	main	Sep 28	M	Merge Conflict	⊘	⊘	⊘
arch/x86/riscv: Use 'all' target to include files in all stages	Arthur Heyma...	Philipp, ron, +2	coreboot	main	Sep 28	M	⊘ 1 missing	✓	✓	⊘
flashrom: Mark RISCV as non memory-mapped I/O architecture	Khem	Edward, Angel, +1	flashrom	main (riscv)	Sep 25	XS	Merge Conflict	⊘	⊘	⊘
include/memlayout.h: Add OPENSBI linker macro	Maximilian Br...	Philipp, ron	coreboot	main	Sep 23	XS	⊘ 1 missing	✓	⊘	✓
treewide: Set up linker files with OPENSBI macro	Maximilian Br...	Paul, Philipp, +1	coreboot	main	Sep 23	XS	⊘ 2 missing	⊘	⊘	✓
ACPI,SMBIOS : Provide ACPI and SMBIOS support for RISC-V	weidongwd	Lance, Philipp, +2	coreboot	main	Sep 23	XL	Merge Conflict	✓	⊘	⊘
ACPI,SMBIOS : Provide ACPI and SMBIOS supports for RISC-V	weidongwd	Maximilian, Philipp, +3	coreboot	main	Sep 19	XL	Merge Conflict	⊘	⊘	-1
mb/emulation/riscv: Limit DRAM size	Arthur Heyma...	Philipp, ron, +1	coreboot	main	Sep 19	XS	⊘ 3 missing	⊘	+1	⊘
riscv: add smp support for exception handler	Arthur Heyma...	Philipp, ron, +2	coreboot	main	Sep 19	M	Merge Conflict	⊘	+1	⊘
arch/arm* Fix some linter warnings and errors	Philipp M	Felix, Julius, +2	coreboot	main	Sep 19	S	⊘ 2 missing	✓	⊘	⊘
arch,toolchain,xcompile: Use the compiler frontend as a linker	Jacob Garber	Iru, Julius, +5	coreboot	main (lto)	Sep 19	M	Merge Conflict	⊘	⊘	⊘
riscv/mb/qemu: fix DRAM probing	Philipp Hug	Arthur, Marc, +2	coreboot	main	Sep 10	M	Merge Conflict	✓	⊘	⊘
commonlib/fsp_relocate: Fix Coverity Issues	eddie	Johnny, Jonathan, +1	coreboot	main	Sep 10	S	⊘ 2 missing	✓	⊘	⊘
lib/smbios: Add 32 bit entry point if below 4G	Maximilian Br...		coreboot	main	Sep 09	S	⊘ 3 missing	⊘	⊘	⊘
Allow to build romstage sources inside the bootblock	Arthur Heyma...	Christian, Felix, +20	coreboot	main	Sep 09	M	⊘ 2 missing	✓	⊘	⊘
qemu-riscv: Improve Documentation	Ayan Agrawal	Daniel, Daniel, +2	coreboot	main	Sep 09	M	Merge Conflict	⊘	⊘	⊘

Page 1

A lot of work still remains open.

[1] Screenshot taken on 2023-10-06

State of oreboot on RISC-V in 2023 / Q4

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

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)

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

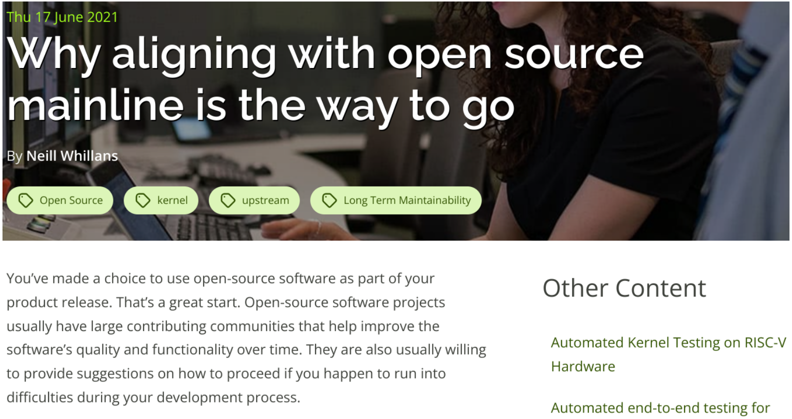
State of oreboot on RISC-V in 2023 / Q4

- 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



Thu 17 June 2021

Why aligning with open source mainline is the way to go

By Neill Whillans

Open Source · kernel · upstream · Long Term Maintainability

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.

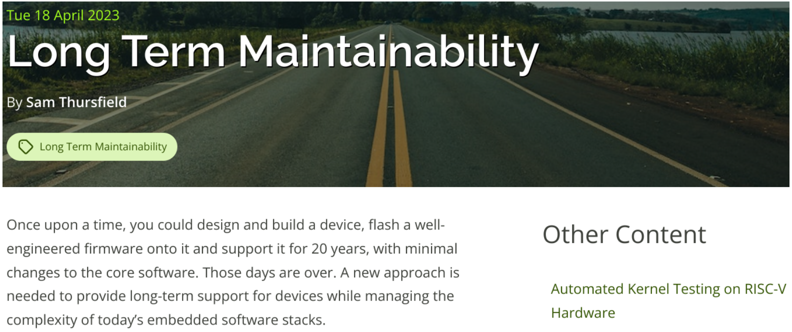
Other Content

Automated Kernel Testing on RISC-V Hardware

Automated end-to-end testing for

https://www.codethink.co.uk/articles/2021/why-aligning-with-open-source-mainline-is-the-way-to-go/

Long Term Maintainability



Tue 18 April 2023

Long Term Maintainability

By Sam Thursfield

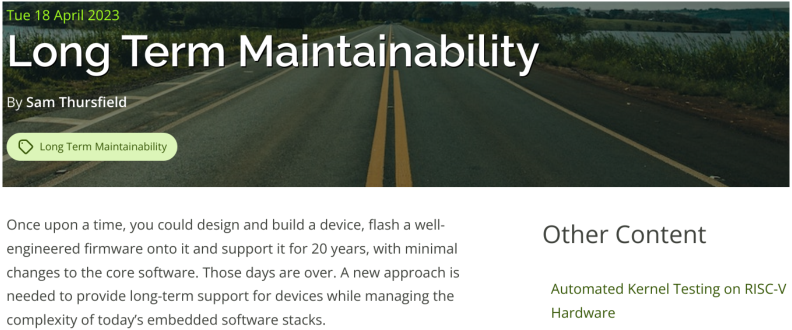
Once upon a time, you could design and build a device, flash a well-engineered firmware onto it and support it for 20 years, with minimal changes to the core software. Those days are over. A new approach is needed to provide long-term support for devices while managing the complexity of today's embedded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

Other Content

Automated Kernel Testing on RISC-V Hardware

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

Long Term Maintainability



Tue 18 April 2023

Long Term Maintainability

By Sam Thursfield

Long Term Maintainability

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.

Other Content

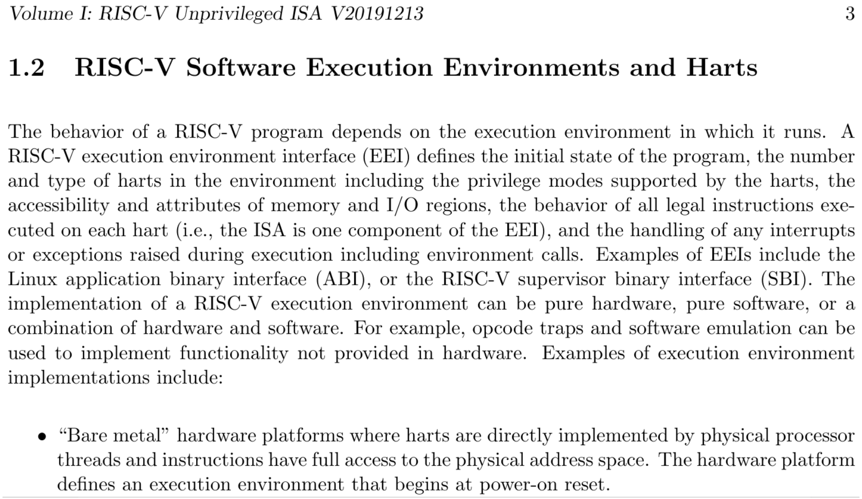
Automated Kernel Testing on RISC-V Hardware

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!

RISC-V Execution Environments



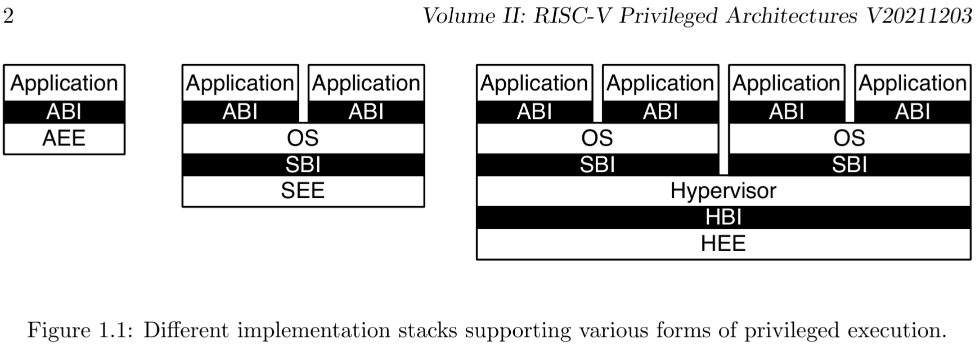
Volume I: RISC-V Unprivileged ISA V20191213 3

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



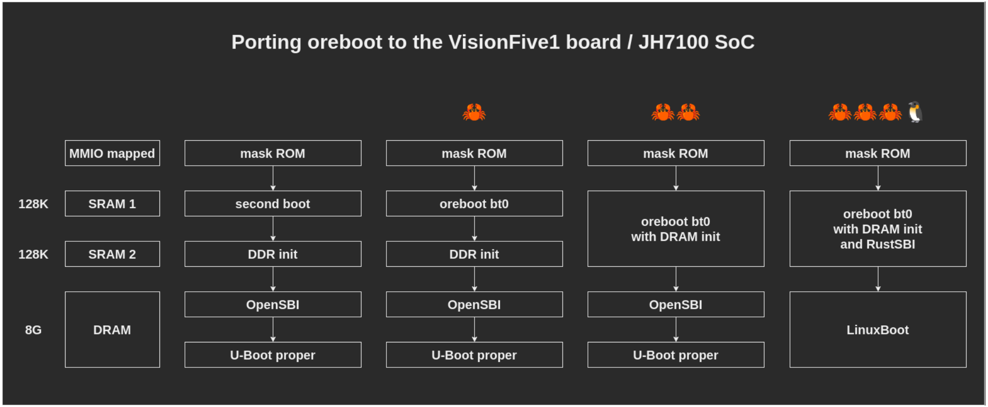
2 *Volume II: RISC-V Privileged Architectures V20211203*

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[2]



Porting oreboot to the VisionFive1 board / JH7100 SoC

			🦀	🦀🦀	🦀🦀🦀🐧
	MMIO mapped	mask ROM	mask ROM	mask ROM	mask ROM
128K	SRAM 1	second boot	oreboot bt0	oreboot bt0 with DRAM init	oreboot bt0 with DRAM init and RustSBI
128K	SRAM 2	DDR init	DDR init		
8G	DRAM	OpenSBI	OpenSBI	OpenSBI	LinuxBoot
		U-Boot proper	U-Boot proper	U-Boot proper	

oreboot + RustSBI = SEE (Supervisor Execution Environment)

[2] https://github.com/oreboot/oreboot/blob/main/Documentation/boot-flow.md

Rising Interest in oreboot and LinuxBoot

Rising Interest in oreboot and LinuxBoot

https://www.reddit.com/r/RISCV/comments/13ksvsz/comment/jkm63qh/

Why dom’t they collaborate with Oreboot project?

Rising Interest in oreboot and LinuxBoot

https://www.reddit.com/r/RISCV/comments/13ksvsz/comment/jkm63qh/

Why dom't they collaborate with Oreboot project?

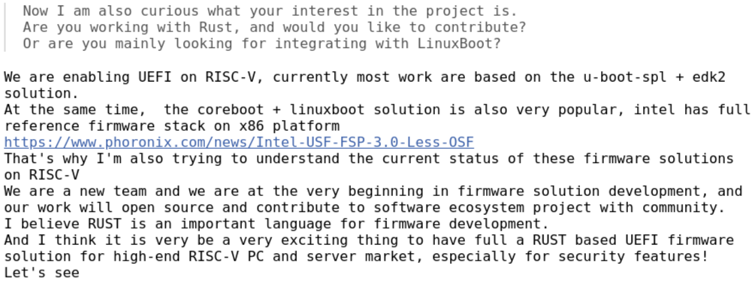
https://forum.rvspace.org/t/oreboot-visionfive-2-support-effort/2211

Rising Interest in oreboot and LinuxBoot

https://www.reddit.com/r/RISCV/comments/13ksvsz/comment/jkm63qh/
Why dom'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
https://www.phoronix.com/news/Intel-USF-FSP-3.0-Less-OSF
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
```

[3] https://www.youtube.com/watch?v=gB3wgOuvLJQ

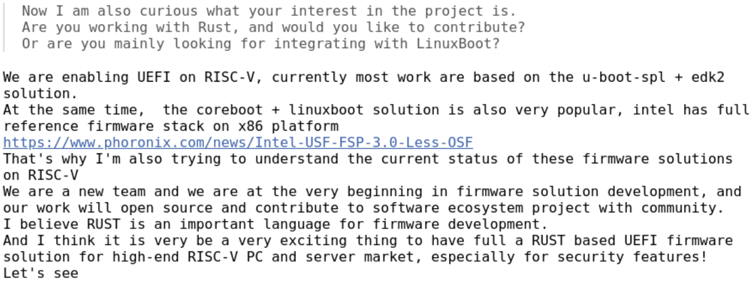
Rising Interest in oreboot and LinuxBoot

https://www.reddit.com/r/RISCV/comments/13ksvsz/comment/jkm63qh/

Why dom't they collaborate with Oreboot project?

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

Someone from Intel contacted me:



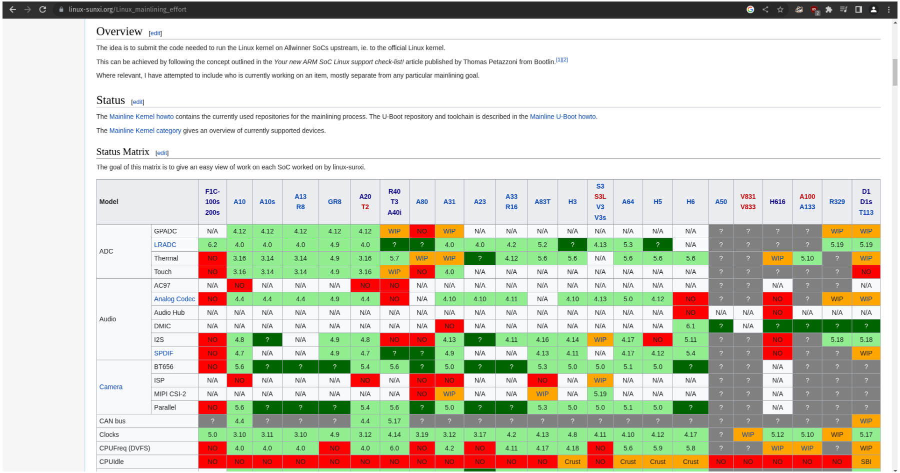
> Now I am also curious what your interest in the project is.
> Are you working with Rust, and would you like to contribute?
> Or are you mainly looking for integrating with LinuxBoot?

```
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
https://www.phoronix.com/news/Intel-USF-FSP-3.0-Less-OSF
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



linux-sunxi.org/Linux_mainlining_effort

Overview [edit]

The idea is to submit the code needed to run the Linux kernel on Allwinner SoCs upstream, ie. to the official Linux kernel.

This can be achieved by following the concept outlined in the *Your new ARM SoC Linux support check-list!* article published by Thomas Petazzoni from Bootlin.[1][2]

Where relevant, I have attempted to include who is currently working on an item, mostly separate from any particular mainlining goal.

Status [edit]

The Mainline Kernel howto contains the currently used repositories for the mainlining process. The U-Boot repository and toolchain is described in the Mainline U-Boot howto.

The Mainline Kernel category gives an overview of currently supported devices.

Status Matrix [edit]

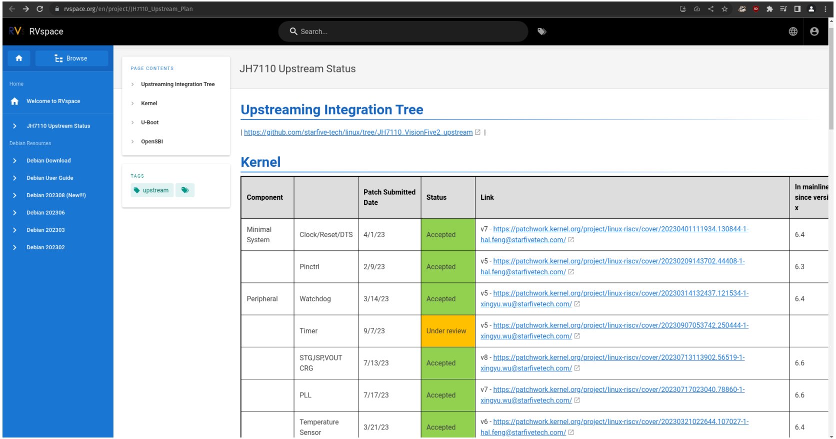
The goal of this matrix is to give an easy view of work on each SoC worked on by linux-sunxi.

Model		F1C- 100s 200s	A10	A10s	A13 R8	GR8	A20 T2	R40 T3 A40i	A80	A31	A23	A33 R16	A83T	H3	S3 S3L V3 V3s	A64	H5	H6	A50	V831 V833	H616	A100 A133	R329	D1 D1s T113
ADC	GPADC	N/A	4.12	4.12	4.12	4.12	4.12	WIP	NO	WIP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	?	?	?	?	WIP	WIP
	LRADC	6.2	4.0	4.0	4.0	4.9	4.0	?	?	4.0	4.0	4.2	5.2	?	4.13	5.3	?	N/A	?	?	?	?	5.19	5.19
	Thermal	NO	3.16	3.14	3.14	4.9	3.16	5.7	WIP	WIP	?	4.12	5.6	5.6	N/A	5.6	5.6	5.6	?	?	WIP	5.10	?	WIP
	Touch	NO	3.16	3.14	3.14	4.9	3.16	WIP	NO	4.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	?	?	?	?	?	NO
Audio	AC97	N/A	NO	N/A	N/A	N/A	NO	NO	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	?	?	N/A	?	?	N/A
	Analog Codec	NO	4.4	4.4	4.4	4.9	4.4	NO	N/A	4.10	4.10	4.11	N/A	4.10	4.13	5.0	4.12	NO	?	?	NO	?	WIP	WIP
	Audio Hub	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	NO	N/A	N/A	NO	N/A	N/A	N/A
	DMIC	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	NO	N/A	N/A	N/A	N/A	N/A	N/A	N/A	6.1	?	N/A	?	?	?	?
	I2S	NO	4.8	?	N/A	4.9	4.8	NO	NO	4.13	?	4.11	4.16	4.14	WIP	4.17	NO	5.11	?	?	NO	?	5.18	5.18
	SPDIF	NO	4.7	N/A	N/A	4.9	4.7	?	?	4.9	N/A	N/A	4.13	4.11	N/A	4.17	4.12	5.4	?	?	NO	?	?	WIP
Camera	BT656	NO	5.6	?	?	?	5.4	5.6	?	5.0	?	?	5.3	5.0	5.0	5.1	5.0	?	?	?	N/A	?	?	?
	ISP	N/A	NO	N/A	N/A	N/A	NO	N/A	NO	NO	N/A	N/A	NO	N/A	WIP	N/A	N/A	N/A	?	?	N/A	?	?	?
	MIPI CSI-2	N/A	N/A	N/A	N/A	N/A	N/A	N/A	NO	WIP	N/A	N/A	WIP	N/A	5.19	N/A	N/A	N/A	?	?	N/A	?	?	?
	Parallel	NO	5.6	?	?	?	5.4	5.6	?	5.0	?	?	5.3	5.0	5.0	5.1	5.0	?	?	?	N/A	?	?	?
CAN bus		?	4.4	?	?	?	4.4	5.17	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	WIP
Clocks		5.0	3.10	3.11	3.10	4.9	3.12	4.14	3.19	3.12	3.17	4.2	4.13	4.8	4.11	4.10	4.12	4.17	?	WIP	5.12	5.10	WIP	5.17
CPUFreq (DVFS)		NO	4.0	4.0	4.0	NO	4.0	6.0	NO	4.2	NO	4.11	4.17	4.18	NO	5.6	5.9	5.8	?	?	WIP	WIP	?	WIP
CPUIdle		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	Crust	NO	Crust	Crust	Crust	NO	NO	NO	NO	NO	SBI

Each SoC and feature is being tracked, driver for driver.

https://linux-sunxi.org/Linux_mainlining_effort

Vendor tracking upstream



JH7110 Upstream Status

Upstreaming Integration Tree

| https://github.com/starfive-tech/linux/tree/JH7110_VisionFive2_upstream |

Kernel

Component		Patch Submitted Date	Status	Link	In mainline since versi x
Minimal System	Clock/Reset/DTS	4/1/23	Accepted	v7 - https://patchwork.kernel.org/project/linux-riscv/cover/20230401111934.130844-1-hal.feng@starfivetech.com/	6.4
	Pinctrl	2/9/23	Accepted	v5 - https://patchwork.kernel.org/project/linux-riscv/cover/20230209143702.44408-1-hal.feng@starfivetech.com/	6.3
Peripheral	Watchdog	3/14/23	Accepted	v5 - https://patchwork.kernel.org/project/linux-riscv/cover/20230314132437.121534-1-xingyu.wu@starfivetech.com/	6.4
	Timer	9/7/23	Under review	v5 - https://patchwork.kernel.org/project/linux-riscv/cover/20230907053742.250444-1-xingyu.wu@starfivetech.com/	
	STG,ISP,VOUT CRG	7/13/23	Accepted	v8 - https://patchwork.kernel.org/project/linux-riscv/cover/20230713113902.56519-1-xingyu.wu@starfivetech.com/	6.6
	PLL	7/17/23	Accepted	v7 - https://patchwork.kernel.org/project/linux-riscv/cover/20230717023040.78860-1-xingyu.wu@starfivetech.com/	6.6
	Temperature Sensor	3/21/23	Accepted	v6 - https://patchwork.kernel.org/project/linux-riscv/cover/20230321022644.107027-1-hal.feng@starfivetech.com/	6.4

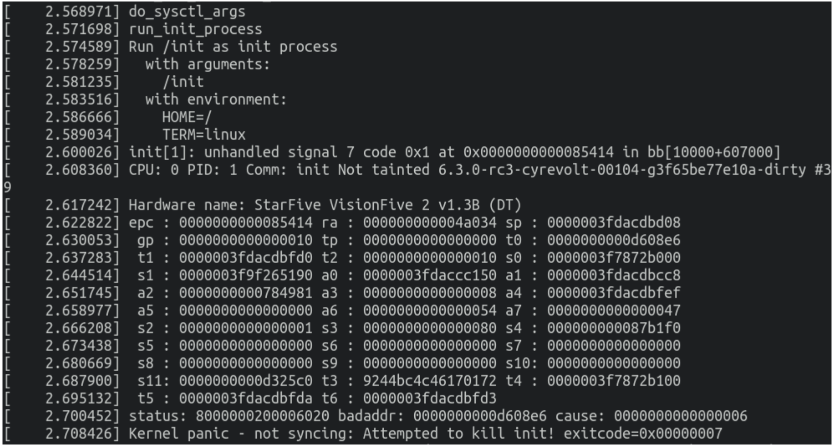
Each component - kernel, U-Boot and OpenSBI - is tracked, driver for driver, with a link to the respective patch series in the mailing list.

https://rvspace.org/en/project/JH7110_Upstream_Plan

Unaligned Memory Access

When it goes wrong - userspace

When it goes wrong - userspace

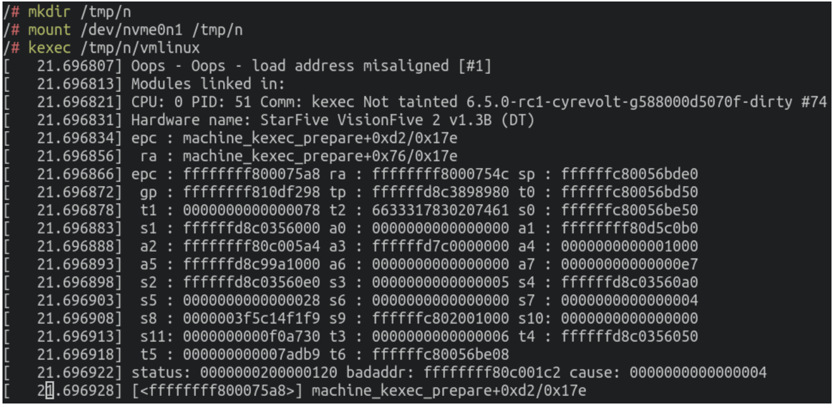


```
[    2.568971] do_sysctl_args
[    2.571698] run_init_process
[    2.574589] Run /init as init process
[    2.578259]   with arguments:
[    2.581235]     /init
[    2.583516]   with environment:
[    2.586666]     HOME=/
[    2.589034]     TERM=linux
[    2.600026] init[1]: unhandled signal 7 code 0x1 at 0x0000000000085414 in bb[10000+607000]
[    2.608360] CPU: 0 PID: 1 Comm: init Not tainted 6.3.0-rc3-cyrevolt-00104-g3f65be77e10a-dirty #3
9
[    2.617242] Hardware name: StarFive VisionFive 2 v1.3B (DT)
[    2.622822] epc : 0000000000085414 ra : 000000000004a034 sp : 0000003fdacdbd08
[    2.630053]  gp : 0000000000000010 tp : 0000000000000000 t0 : 0000000000d608e6
[    2.637283]  t1 : 0000003fdacdbfd0 t2 : 0000000000000010 s0 : 0000003f7872b000
[    2.644514]  s1 : 0000003f9f265190 a0 : 0000003fdaccc150 a1 : 0000003fdacdbcc8
[    2.651745]  a2 : 0000000000784981 a3 : 0000000000000008 a4 : 0000003fdacdbfef
[    2.658977]  a5 : 0000000000000000 a6 : 0000000000000054 a7 : 0000000000000047
[    2.666208]  s2 : 0000000000000001 s3 : 0000000000000080 s4 : 000000000087b1f0
[    2.673438]  s5 : 0000000000000000 s6 : 0000000000000000 s7 : 0000000000000000
[    2.680669]  s8 : 0000000000000000 s9 : 0000000000000000 s10: 0000000000000000
[    2.687900]  s11: 0000000000d325c0 t3 : 9244bc4c46170172 t4 : 0000003f7872b100
[    2.695132]  t5 : 0000003fdacdbfda t6 : 0000003fdacdbfd3
[    2.700452] status: 8000000200006020 badaddr: 0000000000d608e6 cause: 0000000000000006
[    2.708426] Kernel panic - not syncing: Attempted to kill init! exitcode=0x00000007
```

Note: fixed in Go 1.21

When it goes wrong - kernel space

When it goes wrong - kernel space



```
/# mkdir /tmp/n
/# mount /dev/nvme0n1 /tmp/n
/# kexec /tmp/n/vmlinux
[   21.696807] Oops - Oops - load address misaligned [#1]
[   21.696813] Modules linked in:
[   21.696821] CPU: 0 PID: 51 Comm: kexec Not tainted 6.5.0-rc1-cyrevolt-g588000d5070f-dirty #74
[   21.696831] Hardware name: StarFive VisionFive 2 v1.3B (DT)
[   21.696834] epc : machine_kexec_prepare+0xd2/0x17e
[   21.696856]  ra : machine_kexec_prepare+0x76/0x17e
[   21.696866] epc : ffffffff800075a8 ra : ffffffff8000754c sp : ffffffc80056bde0
[   21.696872]  gp : ffffffff810df298 tp : ffffffd8c3898980 t0 : ffffffc80056bd50
[   21.696878]  t1 : 0000000000000078 t2 : 6633317830207461 s0 : ffffffc80056be50
[   21.696883]  s1 : ffffffd8c0356000 a0 : 0000000000000000 a1 : ffffffff80d5c0b0
[   21.696888]  a2 : ffffffff80c005a4 a3 : ffffffd7c0000000 a4 : 0000000000001000
[   21.696893]  a5 : ffffffd8c99a1000 a6 : 0000000000000000 a7 : 00000000000000e7
[   21.696898]  s2 : ffffffd8c03560e0 s3 : 0000000000000005 s4 : ffffffd8c03560a0
[   21.696903]  s5 : 0000000000000028 s6 : 0000000000000000 s7 : 0000000000000004
[   21.696908]  s8 : 0000003f5c14f1f9 s9 : ffffffc802001000 s10: 0000000000000000
[   21.696913]  s11: 0000000000f0a730 t3 : 0000000000000006 t4 : ffffffd8c0356050
[   21.696918]  t5 : 000000000007adb9 t6 : ffffffc80056be08
[   21.696922] status: 0000000200000120 badaddr: ffffffff80c001c2 cause: 0000000000000004
[   21.696928] [<ffffffff800075a8>] machine_kexec_prepare+0xd2/0x17e
```

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

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

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 oreboot 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

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.*

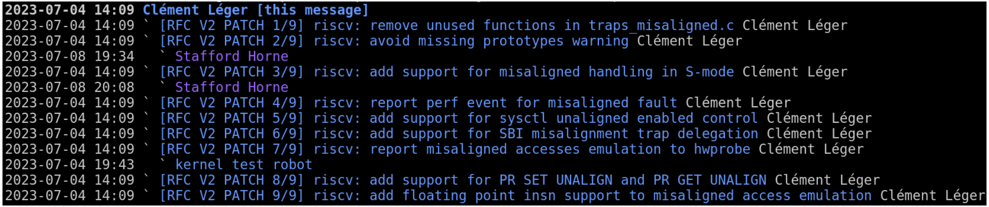
Unaligned access in S-Mode / Linux

`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*



```
2023-07-04 14:09 Clément Léger [this message]
2023-07-04 14:09 ` [RFC V2 PATCH 1/9] riscv: remove unused functions in traps_misaligned.c Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 2/9] riscv: avoid missing prototypes warning Clément Léger
2023-07-08 19:34   ` Stafford Horne
2023-07-04 14:09 ` [RFC V2 PATCH 3/9] riscv: add support for misaligned handling in S-mode Clément Léger
2023-07-08 20:08   ` Stafford Horne
2023-07-04 14:09 ` [RFC V2 PATCH 4/9] riscv: report perf event for misaligned fault Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 5/9] riscv: add support for sysctl unaligned enabled control Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 6/9] riscv: add support for SBI misalignment trap delegation Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 7/9] riscv: report misaligned accesses emulation to hwprobe Clément Léger
2023-07-04 19:43   ` kernel test robot
2023-07-04 14:09 ` [RFC V2 PATCH 8/9] riscv: add support for PR SET UNALIGN and PR GET UNALIGN Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 9/9] riscv: add floating point insn support to misaligned access emulation Clément Léger
```

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

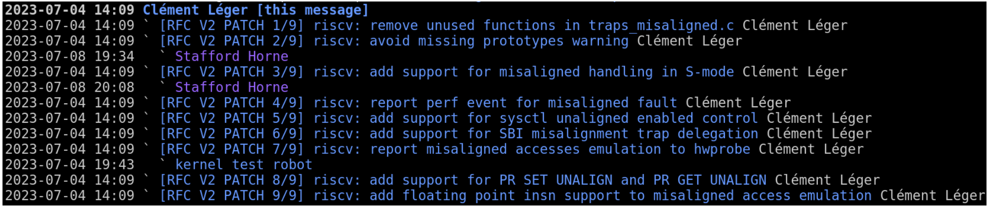
Unaligned access in S-Mode / Linux

`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



```
2023-07-04 14:09 Clément Léger [this message]
2023-07-04 14:09 ` [RFC V2 PATCH 1/9] riscv: remove unused functions in traps_misaligned.c Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 2/9] riscv: avoid missing prototypes warning Clément Léger
2023-07-08 19:34   ` Stafford Horne
2023-07-04 14:09 ` [RFC V2 PATCH 3/9] riscv: add support for misaligned handling in S-mode Clément Léger
2023-07-08 20:08   ` Stafford Horne
2023-07-04 14:09 ` [RFC V2 PATCH 4/9] riscv: report perf event for misaligned fault Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 5/9] riscv: add support for sysctl unaligned enabled control Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 6/9] riscv: add support for SBI misalignment trap delegation Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 7/9] riscv: report misaligned accesses emulation to hwprobe Clément Léger
2023-07-04 19:43   ` kernel test robot
2023-07-04 14:09 ` [RFC V2 PATCH 8/9] riscv: add support for PR SET UNALIGN and PR GET UNALIGN Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 9/9] riscv: add floating point insn support to misaligned access emulation Clément Léger
```

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

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

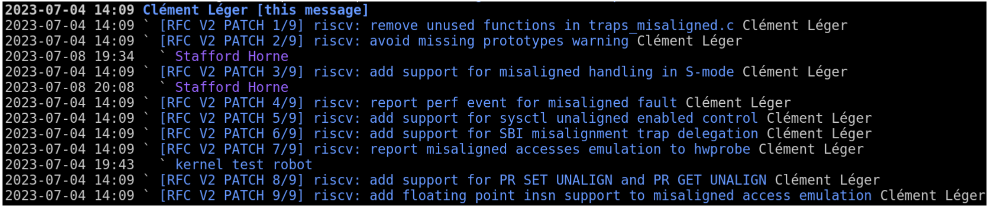
Unaligned access in S-Mode / Linux

`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*



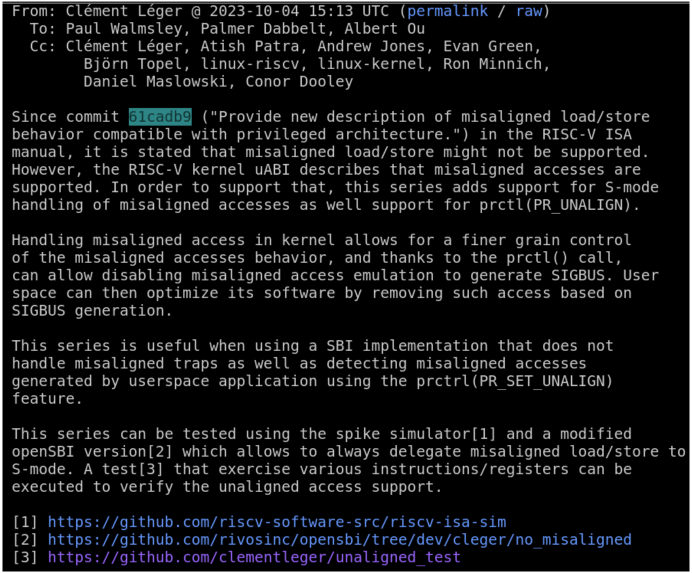
```
2023-07-04 14:09 Clément Léger [this message]
2023-07-04 14:09 ` [RFC V2 PATCH 1/9] riscv: remove unused functions in traps_misaligned.c Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 2/9] riscv: avoid missing prototypes warning Clément Léger
2023-07-08 19:34   ` Stafford Horne
2023-07-04 14:09 ` [RFC V2 PATCH 3/9] riscv: add support for misaligned handling in S-mode Clément Léger
2023-07-08 20:08   ` Stafford Horne
2023-07-04 14:09 ` [RFC V2 PATCH 4/9] riscv: report perf event for misaligned fault Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 5/9] riscv: add support for sysctl unaligned enabled control Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 6/9] riscv: add support for SBI misalignment trap delegation Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 7/9] riscv: report misaligned accesses emulation to hwprobe Clément Léger
2023-07-04 19:43   ` kernel test robot
2023-07-04 14:09 ` [RFC V2 PATCH 8/9] riscv: add support for PR SET UNALIGN and PR GET UNALIGN Clément Léger
2023-07-04 14:09 ` [RFC V2 PATCH 9/9] riscv: add floating point insn support to misaligned access emulation Clément Léger
```

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. :-)

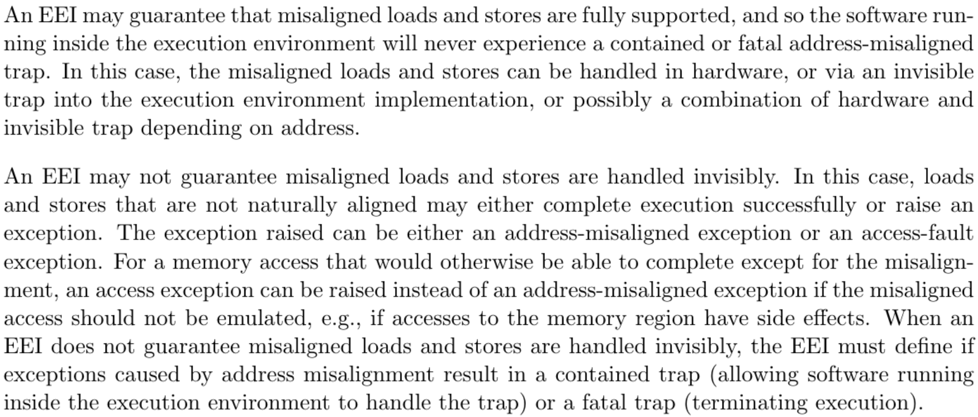
Linux: Discussion picked up again[4]



```
From: Clément Léger @ 2023-10-04 15:13 UTC (permalink / raw)
  To: Paul Walmsley, Palmer Dabbelt, 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 61cadb9 ("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 prctrl(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.

[1] https://github.com/riscv-software-src/riscv-isa-sim
[2] https://github.com/rivosinc/opensbi/tree/dev/cleger/no_misaligned
[3] https://github.com/clementleger/unaligned_test
```

[4] 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).

added in https://github.com/riscv/riscv-isa-manual/commit/61cadb9

Awesome Demo?

Credits

Credits

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

Credits

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

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

Credits

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 :-)

Credits

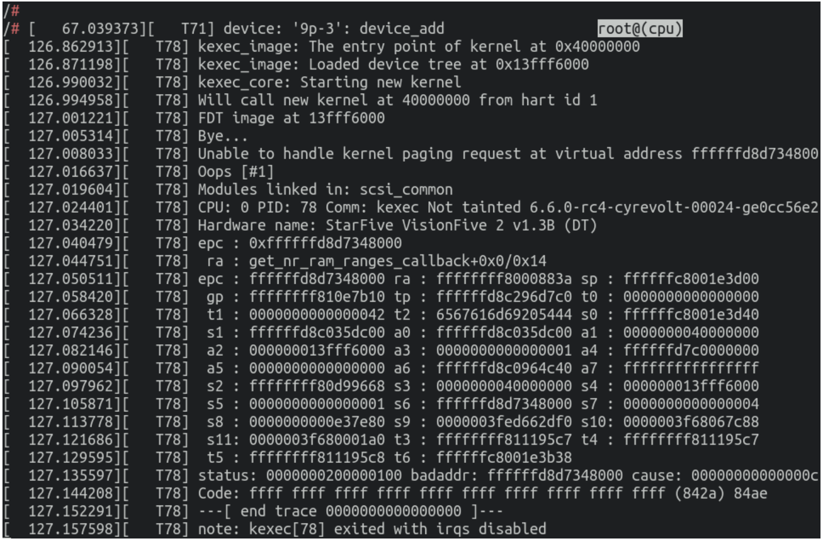
Keep in mind that many people are working on many layers.

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!

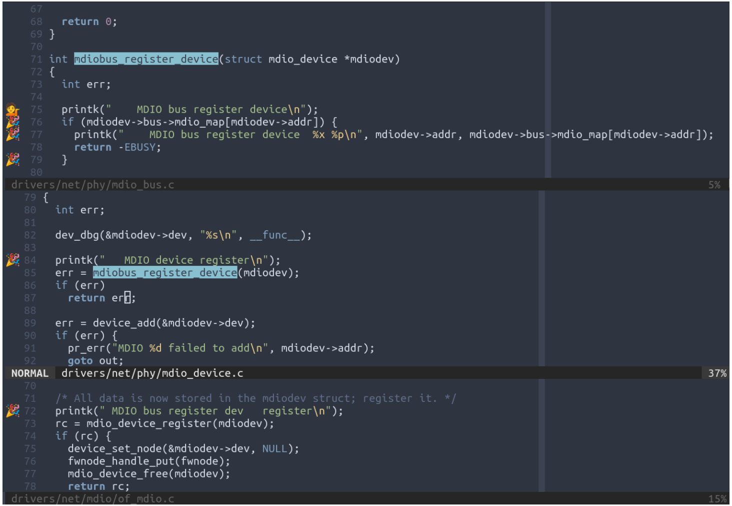
`kexec` still needs some work



```
/#
/# [   67.039373][   T71] device: '9p-3': device_add                  root@(cpu)
[  126.862913][   T78] kexec_image: The entry point of kernel at 0x40000000
[  126.871198][   T78] kexec_image: Loaded device tree at 0x13fff6000
[  126.990032][   T78] kexec_core: Starting new kernel
[  126.994958][   T78] Will call new kernel at 40000000 from hart id 1
[  127.001221][   T78] FDT image at 13fff6000
[  127.005314][   T78] Bye...
[  127.008033][   T78] Unable to handle kernel paging request at virtual address ffffffd8d734800
[  127.016637][   T78] Oops [#1]
[  127.019604][   T78] Modules linked in: scsi_common
[  127.024401][   T78] CPU: 0 PID: 78 Comm: kexec Not tainted 6.6.0-rc4-cyrevolt-00024-ge0cc56e2
[  127.034220][   T78] Hardware name: StarFive VisionFive 2 v1.3B (DT)
[  127.040479][   T78] epc : 0xffffffd8d7348000
[  127.044751][   T78]  ra : get_nr_ram_ranges_callback+0x0/0x14
[  127.050511][   T78] epc : ffffffd8d7348000 ra : ffffffff8000883a sp : ffffffc8001e3d00
[  127.058420][   T78]  gp : ffffffff810e7b10 tp : ffffffd8c296d7c0 t0 : 0000000000000000
[  127.066328][   T78]  t1 : 0000000000000042 t2 : 6567616d69205444 s0 : ffffffc8001e3d40
[  127.074236][   T78]  s1 : ffffffd8c035dc00 a0 : ffffffd8c035dc00 a1 : 0000000040000000
[  127.082146][   T78]  a2 : 000000013fff6000 a3 : 0000000000000001 a4 : ffffffd7c0000000
[  127.090054][   T78]  a5 : 0000000000000000 a6 : ffffffd8c0964c40 a7 : ffffffffffffffff
[  127.097962][   T78]  s2 : ffffffff80d99668 s3 : 0000000040000000 s4 : 000000013fff6000
[  127.105871][   T78]  s5 : 0000000000000001 s6 : ffffffd8d7348000 s7 : 0000000000000004
[  127.113778][   T78]  s8 : 0000000000e37e80 s9 : 0000003fed662df0 s10: 0000003f68067c88
[  127.121686][   T78]  s11: 0000003f680001a0 t3 : ffffffff811195c7 t4 : ffffffff811195c7
[  127.129595][   T78]  t5 : ffffffff811195c8 t6 : ffffffc8001e3b38
[  127.135597][   T78] status: 0000000200000100 badaddr: ffffffd8d7348000 cause: 000000000000000c
[  127.144208][   T78] Code: ffff ffff ffff ffff ffff ffff ffff ffff ffff ffff (842a) 84ae
[  127.152291][   T78] ---[ end trace 0000000000000000 ]---
[  127.157598][   T78] note: kexec[78] exited with irqs disabled
```

VisionFive 2 trouble with Device Tree + MDIO

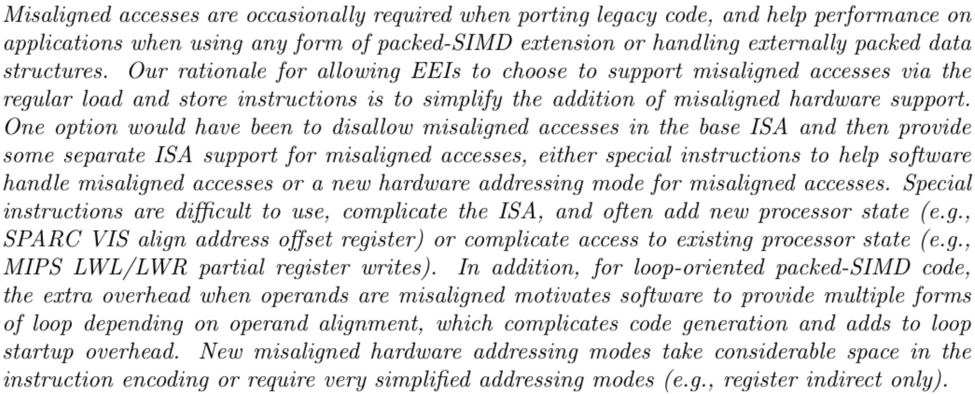
Many files are involved - hard to find the issue.



```
  return 0;
}

int mdiobus_register_device(struct mdio_device *mdiodev)
{
  int err;

  printk("    MDIO bus register device\n");
  if (mdiodev->bus->mdio_map[mdiodev->addr]) {
    printk("    MDIO bus register device  %x %p\n", mdiodev->addr, mdiodev->bus->mdio_map[mdiodev->addr]);
    return -EBUSY;
  }

drivers/net/phy/mdio_bus.c                                                                    5%
{
  int err;

  dev_dbg(&mdiodev->dev, "%s\n", __func__);

  printk("   MDIO device register\n");
  err = mdiobus_register_device(mdiodev);
  if (err)
    return err;

  err = device_add(&mdiodev->dev);
  if (err) {
    pr_err("MDIO %d failed to add\n", mdiodev->addr);
    goto out;
NORMAL  drivers/net/phy/mdio_device.c                                                         37%

  /* All data is now stored in the mdiodev struct; register it. */
  printk(" MDIO bus register dev   register\n");
  rc = mdio_device_register(mdiodev);
  if (rc) {
    device_set_node(&mdiodev->dev, NULL);
    fwnode_handle_put(fwnode);
    mdio_device_free(mdiodev);
    return rc;
drivers/net/mdio/of_mdio.c                                                                    15%
```

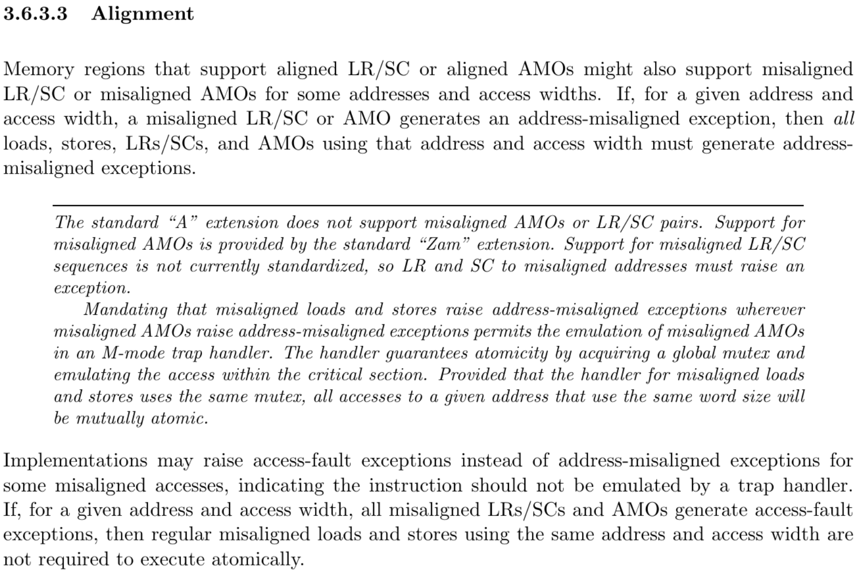
MDIO is the communication between ethernet MAC and PHY.

Alignment in RISC-V ISA Spec: Reasoning



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 complicates 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

https://metaspora.org/aligned-on-riscv-osfc2023.pdf