欢迎第一次加入的伙伴(开会时请从下一页开始展示)

- 开放编辑，直接点击 request for edit 然后在东亚时区群里at吴伟
- 如果没有找到自己的内容分类，可以添加1-2页在最开始或中间
- 欢迎在开始的前5分钟进行自我介绍
- 日常八卦在东亚时区RISC-V双周同步微信群中，欢迎加入

东亚时区RISC-V双周会

2024年10月17日·第089次

https://github.com/cnrv/RISCV-East-Asia-Biweekly-Sync

Host: 张馥媛

Organizer: PLCT Lab plct-oss@iscas.ac.cn

会议议程（15:00 - 16:00）

- 自我介绍、等待参会者接入、非技术话题八卦（没有的话就直接跳过）
- RVI 的更新和八卦（基本上跟东亚双周会群内消息同步）
- Unratified Specs 的参考实现进展
- 东亚地区小伙伴的项目更新
- 自由讨论

RISC-V International 同步、全球开源社区八卦(陈逸轩)

- RISC-V 未来在物联网硬件方面的发展分析。Are IoT Hardware Vendors Finally Going Open Source?
- 红帽和其他发行版致力于定义一种处理器的可拓展性不需要重新编译内核的方式。Software-defined processors: the promise of RISC-V
- Renode在实时应用中引入RISC-V快速中断。Introducing fast RISC-V interrupts support in Renode for real time applications

RISC-V 中文社区的同步与八卦（张宇溪）

产品

- openKylin正式推出RISC-V统一镜像烧录工具，破解镜像碎片化 https://mp.weixin.qq.com/s/u2dEZQxIhFc2C88tJSDyWg
- 进迭时空与中国移动用芯共创AI+时代 https://mp.weixin.qq.com/s/oM8ObzRWdHwp4BXAt8AnVA
- 芯来RISC-V内核赋能格见高性能实时工业控制DSP https://mp.weixin.qq.com/s/OTEaeVjyiC0wabdqTY8Arw

生态

- 著名的 Box86/Box64 模拟器现在有了更好的RISC-V RVV 1.0 支持，性能提升显著 https://mp.weixin.qq.com/s/nVeUro0NKFjPhI4Mvdtc0w
- IAR全面支持国科环宇AS32X系列RISC-V车规MCU https://mp.weixin.qq.com/s/Nic6pteiUulooO_cIZmtrQ
- RISC-V AI技术正式纳入北京大学研究生课程 https://mp.weixin.qq.com/s/aM4KBF7getu8h-GgHw6JFw

活动

- RISC-V生态发展论坛将于10月17日盛大开幕，诚邀您共襄盛会！（10月17日 深圳）https://mp.weixin.qq.com/s/1t7m2s-wKC1JoO_18Gg_qA
- 2024世界消费电子展RISC-V专区参展邀请函（11月28-30日 深圳）https://mp.weixin.qq.com/s/__uG5kCe3rD1knETjJTH1w

RISC-V 韩语社区的同步与八卦

RISC-V 日语社区的同步与八卦

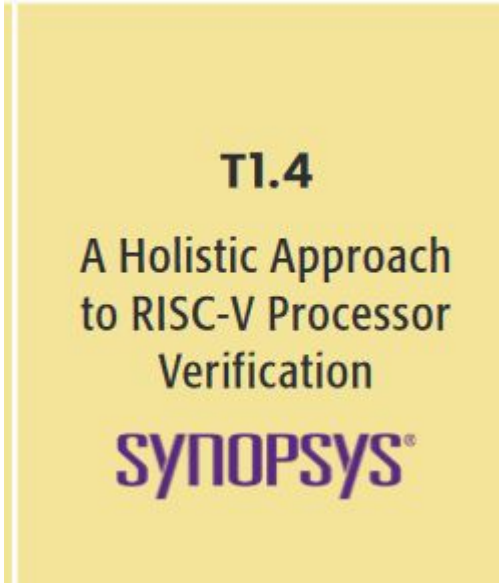
RISC-V 俄语社区的同步与八卦

无更新

RISC-V 德语社区的同步与八卦（罗云翔）

1. DVCon Europe 2024
   October 15 - October 16 Holiday Inn - Munich, Germany Munich, Germany

https://dvcon-europe.org/
https://confcats-siteplex.s3.us-east-1.amazonaws.com/dvconeurope/dvconeu24_program_v13_3f3b8401ca.pdf
https://dvcon-europe.org/program/2024-tutorials



Tutorial Program: Tuesday, October 15th (cont.)

T1.4: A Holistic Approach to RISC-V Processor Verification
4:00 PM – 5:30 PM
Forum 4
Speaker:
Larry Lapides, Synopsys

SYNOPSYS® Silicon to Software™

Processors using the open standard RISC-V instruction set architecture (ISA) are becoming increasingly common, with an estimated 30% of SoCs designed in 2023 containing at least one RISC-V core. Whether licensing RISC-V IP and adding custom instructions, using open-source RISC-V IP or building a RISC-V processor from scratch, verification of RISC-V processors is a task in the SoC project plan. With the variety of sources for the processor IP, the range of complexity and the span of use cases, a one-size-fits-all approach to RISC-V processor verification does not work.

This tutorial presents a holistic approach to RISC-V processor verification using various tools in the Synopsys portfolio. It will cover processor complexity from microcontrollers to application processors to arrays of processors for AI accelerators, different levels of integration from unit to individual processor to processing subsystem to SoC, and cover different scenarios depending on the source of the processor IP. Matching different technologies and methodologies to this multidimensional verification space is critical.

Figure 1 shows an overview of the technologies and products included in this holistic approach to processor verification. These can be separated at a high level into formal and dynamic verification technology groups, however, even within those groups there are multiple technologies, methodologies and use cases. For example, dynamic verification can include self-checking test, post-simulation trace-compare and lockstep continuous-compare methodologies, executed on RTL, hardware-assisted verification platforms or actual silicon.

The key metric for this holistic approach is functional coverage, driven by a comprehensive verification plan. Continuing the example above, the verification plan might utilize relatively simple post-simulation trace-compare for basic instruction verification. However, verification of asynchronous events such as interrupts, debug mode, privilege modes and more requires the lockstep continuous-compare flow (Figure 2), which utilizes the ImperasFPM RISC-V processor model, ImperasDV processor verification environment and ImperasFC functional coverage modules. The verification plan might drill down into specific units in the processor, for example using formal verification (VC Formal FPV plus RISC-V ISA AIP assertions) for the floating point unit or for micro-architectural features such as the pipeline (especially an Out-of-Order pipeline). It might also go to a higher level of integration, for example using PSS (VC PSS) for verification of high level caches in a multi-processor configuration.

The verification plan also needs to take into account processor complexity and the end use case. A simple microcontroller, e.g. RV32IMAC, that is going to have internally-developed software running on it (a limited use case) will need less verification than that same processor that will be exposed to end users running software that may exercise every feature of the core. Verifying custom instructions should be a task commensurate with the number and complexity of the custom instructions, however, there also needs to be some verification that adding the custom instructions did not add unexpected behaviors to the original processor.

Verification also takes a lot of cycles. One estimate is that verification of an application

Tutorial Program: Tuesday, October 15th (cont.)

T3.1: Unleash the Full Potential of Your Waveforms: From Extra-functional Analysis to Functional Debug via Programs on Waveforms
9:45 AM – 11:15 AM
Forum 6
Speakers:
Daniel Große, Johannes Kepler University Linz
Lucas Klemmer, Johannes Kepler University Linz

In the design phase, HDL simulation is the heart for functional and extra-functional verification. The HDL simulator produces a waveform for a simulation run. In case of a deviation from the expected behavior, the waveform has to be analyzed and understood. For this task, waveform viewers are utilized. However, they only allow for viewing signal relations visually which is a highly manual and tedious process. While advanced verification techniques have introduced automation and led to the generation of "better" waveforms (e.g. by employing formal methods, reducing the length of waveforms, or minimizing the signals involved in a failing trace), there has been almost no progress for automating the analysis of waveforms.

In this tutorial we bring automation to the analysis of waveforms and provide hands-on experience on the open-source Waveform Analysis Language (WAL); WAL allows to code your analysis tasks running on waveforms to answer questions like:

- What is the latency of my bus interfaces?
- What throughput is my bus achieving?
- When is the processor pipeline flushed or stalled during software execution?
- Which software basic blocks are executed on my processor?
- How can traditional waveform debugging be complemented with programmable waveform analysis?

WAL (**https://wal-lang.org** and **https://github.com/ics-jku/wal**) has been realized as a Domain Specific Language (DSL). In comparison to other programming languages, WAL programs have direct access to all signal values of a waveform. Accessing signals in WAL is similar to accessing variables with the difference that the value returned depends on the loaded waveform and the time at which the signal is accessed. The reference implementation of WAL is provided open-source in Python. In addition, WAL can be used as an Intermediate Representation (IR); this has been demonstrated by the implementation of WAWK. WAWK is making complex waveform analysis as easy as searching in text files. Moreover, recently a "SystemVerilog-Assertion-to-WAL compiler" has been developed, called WSVA, to check SVAs on simulation traces.

WAL has been used to analyze performance metrics of industrial RISC-V cores, estimate AXI performance in an industrial setting, analyze cache performance of configurable RISC-V cores, generate control flow graph for software running on a CPU, visualize pipeline instruction flows, and visualization of RISC-V programs on CPU diagrams in educational settings.

RISC-V 德语社区的同步与八卦（罗云翔）

1. DVCon Europe 2024
   October 15 - October 16 Holiday Inn - Munich, Germany Munich, Germany
   https://dvcon-europe.org/

Title	Author(s)	Year	Location	Type	
A Statistical and Model-Driven Approach for Comprehensive Fault Propagation Analysis of RISC-V Variants	Endri Kaja, Nicolas Gerlin, Ungsang Yun, Jad Al Halabi, Sebastian Prebeck, Dominik Stoffel, Wolfgang Kunz, Wolfgang Ecker	2024	United States	Paper	Download
Extending the RISC-V Verification Interface for Debug Module Co-Simulation	Michael Chan, Ravi Shethwala, Richa Singhal, Lee Moore, Aimee Sutton	2024	United States	Paper	Download
RISC-V Testing – status and current state of the art	Jon Taylor	2024	United States	Paper	Download
Unleashing the Power of Whisper for block-level verification in high performance RISC-V	Chenhui Huang, Yu Sun ysun, Joe Rahmeh	2024	United States	Paper	Download

2. From Device to Application - Integrating RRAM Accelerator Blocks into Large AI Systems
   https://ieeexplore.ieee.org/document/10682725

RISC-V 学习资源汇总整理计划(汪辰)

暂无更新

RISC-V GCC进展

- riscv-gnu-toolchain gcc更新至14.2，包含自动向量化特性

  https://github.com/riscv-collab/riscv-gnu-toolchain/commit/2fc58ccebf3c8eb325543bb12c2c58ee7cb1b11e

- 扩展包含关系处理目前正在讨论中(示例 c->zca，如果去掉c是否需要禁用zca)

  https://patchwork.sourceware.org/project/binutils/patch/TYZPR03MB70592C11773A6EAE7B64678B9D722@TYZPR03MB7059.apcprd03.prod.outlook.com/

- 支持了Smrnmi和Ssqosid扩展

  https://sourceware.org/pipermail/binutils/2024-September/136847.html
  https://sourceware.org/pipermail/binutils/2024-October/137113.html

- 提交了 TARGET_CLONE 相关的 patch

  [v2] Introduce TARGET_CLONES_ATTR_SEPARATOR for RISC-V
  [v8] RISC-V: Implement __init_riscv_feature_bits, __riscv_feature_bits, and __riscv_vendor_feature_bits
  [RFC] RISC-V: Implement riscv_minimal_hwprobe_feature_bits
  [v4] RISC-V: Implement TARGET_CAN_INLINE_P

- ESWIN提交了Bf16的自动向量化支持

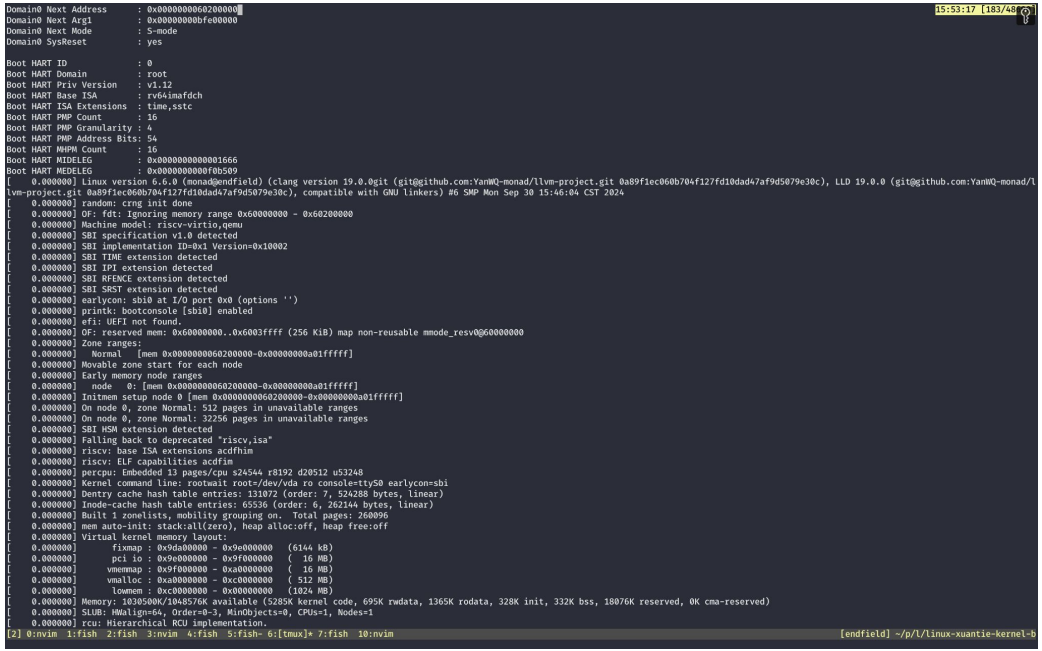
  RISC-V:Auto vect for vector bf16

- Palmer 提交了 LP64DV ABI的草案

  [RFC] RISC-V: Add support for LP64DV

Clang/LLVM 进展 (PLCT)

- LLVM 可以成功编译 rv64ilp32 Linux kernel（bf63582b08）并进入用户态
  - 测试运行 iPerf 时还有一些 bug
  - 未来的工作目标是将 rv64ilp32 纳入 psabi 规范
    https://github.com/ruyisdk/llvm-project/tree/rv64ilp32
- LLVM fortran 编译器 flang-new 更名 flang
  - llvm-project/pull/110023



SAIL进展 (PLCT)

ACT进展 (PLCT)

QEMU/Spike进展（提交者不在线）

- [PATCH 0/8] riscv: AIA userspace irqchip_split support,
  https://lists.gnu.org/archive/html/qemu-riscv/2024-10/msg00237.html

- [PATCH v3 0/8] target/riscv: Add support for Smdbltrp and Ssdbltrp extensions,
  https://lists.gnu.org/archive/html/qemu-riscv/2024-10/msg00281.html

- [PATCH v7 0/5] Add Smrnmi support,
  https://lists.gnu.org/archive/html/qemu-riscv/2024-10/msg00292.html

- [RFC v3 0/2] target/riscv: add endianness checks and atomicity guarantees.,
  https://lists.gnu.org/archive/html/qemu-riscv/2024-10/msg00299.html

- [PATCH] hw/riscv: Add Microblaze V 32bit virt board,
  https://lists.gnu.org/archive/html/qemu-riscv/2024-10/msg00303.html

V8 for RISC-V 更新（邱吉、陆亚涵）

Maglev(syntacore):

1. 5915617: [riscv][maglev] Re-enable Maglev by default for CP build | https://chromium-review.googlesource.com/c/v8/v8/+/5915617
2. 5891216: [riscv][codegen] Fix push/pop register order in AssertSmiOrHeapObjec... | https://chromium-review.googlesource.com/c/v8/v8/+/5891216
3. 5891216: [riscv][codegen] Fix push/pop register order in AssertSmiOrHeapObjec... | https://chromium-review.googlesource.com/c/v8/v8/+/5891216

1. 5923492: [riscv][maglev] Passing a right Condition to Assert | https://chromium-review.googlesource.com/c/v8/v8/+/5923492
2. 5933899: [riscv][tagged] Make FixedArray a HeapObjectLayout class | https://chromium-review.googlesource.com/c/v8/v8/+/5933899
3. 5912873: [riscv][sandbox][leaptiering] Validate argument count for optimized JS calls | https://chromium-review.googlesource.com/c/v8/v8/+/5912873
4. 5894765: [riscv] Add scratch reg in Abort | https://chromium-review.googlesource.com/c/v8/v8/+/5894765

Spidermonkey for RISC-V更新（邱吉、陆亚涵）

过去两周 暂无更新

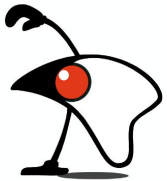
OpenJDK for RISC-V 杨飞

1. Authored JDK-mainline PRs:
- https://github.com/openjdk/jdk/pull/20805 (8339359: RISC-V: Use auipc explicitly in far_jump and far_call macro assembler routines)

2. Reviewed JDK-mainline PRs:
- https://github.com/openjdk/jdk/pull/20912 (8339741: RISC-V: C ABI breakage for integer on stack)
- https://github.com/openjdk/jdk/pull/20913 (8339771: RISC-V: Reduce icache flushes) -> Reduce System IPI (Linux PR_RISCV_SET_ICACHE_FLUSH_CTX syscall)
- https://github.com/openjdk/jdk/pull/20992 (8340102: Move assert-only loop in OopMapSort::sort under debug macro)
- https://github.com/openjdk/jdk/pull/20832 (8339466: Enumerate shared stubs and define static fields and names via declarations)
- https://github.com/openjdk/jdk/pull/20479 (8337753: Target class of upcall stub may be unloaded)
- https://github.com/openjdk/jdk/pull/21059 (8339992: RISC-V: some minor improvements of base64_vector_decode_round)
- https://github.com/openjdk/jdk/pull/20910 (8339738: RISC-V: Vectorize crc32 intrinsic)
- https://github.com/openjdk/jdk/pull/21105 (8340438: RISC-V: minor improvement in base64)
- https://github.com/openjdk/jdk/pull/21376 (8341586: RISC-V: build fail with gcc9)

3. Reviewed JDK-8u mainline PRs:
- https://github.com/openjdk/jdk8u-dev/pull/573 (8199138: Add RISC-V support to Zero)

4. Port JDK-8337511 Implement JEP-404: Generational Shenandoah (Experimental) to RISC-V
- RISC-V specific changes reviewed and merged into project shenandoah repo
- https://github.com/openjdk/shenandoah/pull/493 (8339643: Port JEP 404 to RISC-V)



Go community work update 蒙卓

Code

1. Plugin support [CI passed]
    a. don't merge symbols on riscv64 when dynamic linking
    b. add R_GOT_PCREL_ITYPE_RELOC for riscv64
    c. implement plugin mode for riscv64
2. Compress instructions [pending]
3. Bits (Zba/Zbb/Zbs) [reviewed]
4. Vector instructions [pending]
5. Go runtime syscall with ABI internal
    a. tools: Relax on writeResult
    b. runtime: using ABI on syscall for riscv64

Buidler/CI updates:

1. FreeBSD Go builders has updated to 14.1

RuyiSDK (Xi Jing, PLCT)

- Introduction
  - RuyiSDK: An Integrated Development Environment for RISC-V
  - Components of RuyiSDK
    - Ruyi Component Manager（also known as Ruyi Package Manager）
    - Ruyi Integrated Development Environment (Ruyi IDE)：Development toolkit for RISC-V software and applications
    - Developer Community(Ruyi Developer Community)：Documentation, tutorials, forums, technical discussions, blogs, etc.
  - URL：
    - **Website：https://ruyisdk.org/**
    - Documentation: https://ruyisdk.github.io/docs
    - OS support for RISC-V boards：https://github.com/ruyisdk/support-matrix/
    - Download:
      - GitHub Releases: https://github.com/ruyisdk/ruyi/releases/
      - iscas: https://mirror.iscas.ac.cn/ruyisdk/ruyi/releases/
    - GitHub: https://github.com/ruyisdk/ruyi

RuyiSDK (Xi Jing, PLCT)

- RuyiSDK V0.19 released, Package manager update:
  - Added multi-toolchain support, you can configure multiple different toolchain packages for a virtual environment.
  - Started basic telemetry function, user data is currently stored locally, and will support upload in the future.
  - Fixed the crash of `**ruyi news list**` when news contains UTF-8 character, thanks to external contributor RekiDunois!
  - Added `**ruyi admin format-manifest**` command to automatically format the package description file.
  - Added `**ruyi self clean**` command to clean up storage space.
- Linux distribution packaging:
  - Completed the packaging of multiple mainstream Linux distributions (such as Debian, Ubuntu, Fedora, openEuler, Arch Linux etc.) based on ruyi v0.18. link
- **Eclipse upstream initially supports RISC-V**, and daily build images can be downloaded for trial. link
- OS support matrix:
  - Rewritten the main document of the support matrix, categorizing it by Linux distributions, RTOS, BSD, and other types.
  - Add and modify several test reports.

openEuler RISC-V (周嘉诚)

Status / 20241017

- 🎉 openEuler 24.09 available now [Dnld]
- 🎉 24.09 Preview Release of *oE LLVM Parallel Universe Project* available now [Dnld]
- Following releases in 2H'24
  - Late Q4 - 24.03 follow-up community release for supporting more devices w/ *vendor kernels, proprietary drivers, etc*.
  - Late Q4. - 24.03 LTS Service Pack 1

- core packages in 24.03 LTS [Full List *in Chinese*]
  - glibc 2.38, binutils 2.41, gcc 12.3.1, llvm 17.0.6
  - openjdk 8u402-b08 / 11.0.23 / 17.0.11 / 21.0.3
  - python 3.11.6, perl 5.38.0
  - golang 1.21.4, rust 1.77.0

- Features:
  - 6.6-based common kernel for Qemu, SG2042 (Pioneer) & TH1520 (LPi4A)
  - UEFI-supported Hardware & QEMU images
  - Penglai TEE-enabled firmware variants

- Images:
  - UEFI Install ISO for SG2042 (Pioneer)
    - Standard & Netinst variants available
  - UEFI qcow2 Image w/ Penglai TEE
  - Legacy-boot Images for Pioneer & LPi4A
  - *Other images coming in the next community release*

Gentoo for RISC-V 的情况更新（Gentoo 小队）

暂无更新

Arch Linux RISC-V（潘瑞哲、Felix）

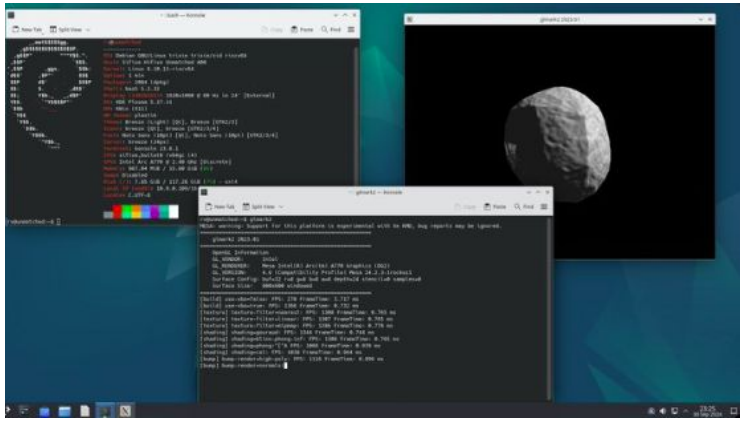
Fedora for RISC-V status update（20241017）

- **RPM packaging**
  - Koji Status: **Rawhide（F41）, GA on Oct 15**
    - **F39: 98.59% srpm [stop]**
    - **Rawhide/41: 22753/23689[96.04%] srprn**

      https://www.fedoravforce.com
- **main package version:**
  - Toolchain: gcc-14.0.1-0.15.3、**glibc-2.40-4**、**binutils-2.43.1-3**[up-to-date]
  - libffi-3.4.6-2(up-to-date)
  - **java-1.8.0-openjdk**
  - java-11-openjdk,java-17-openjdk,java-21-openjdk
  - java-latest-openjdk
  - perl-5.40.0-509(up-to-date)
  - **python3.13-3.13.0~rc1-3(up-to-date)**
  - **llvm-19.1.0-1(up-to-date)**
  - **golang-1.23.2-2**(up-to-date)
  - **rust-1.81.0-6**(up-to-date)

- Desktop support Fedora Rawhide:
  - **DONE: XFCE/LXDE/LXQT/Cinnamon/Sway/Budgie/Sugar/GNOME/Mate**
  - Testing:**KDE/Deepin**
  - **Key Desktop App**
    - **firefox-131.0-2[upstreamed]**
    - libreoffice-24.8.2.1-1**[upstreamed]**
    - Thunderbird-115.11.1-1[DONE]
    - chromium-126.0.6478.182-2[DONE]
- Image :
  - **https://images.fedoravforce.com/**
  - https://openkoji.iscas.ac.cn/pub/dist-repos/dl/
  - https://mirrors.iscas.ac.cn/fedora-riscv
  - https://dl.fedoraproject.org/pub/alt/risc-v/fedora-remix/
- ROS/ROS2 upgraded to F41
- Sail for rawhide[UPSTREAMING]
- function testing for F41:
  - **Podman[pass], Image: fedora-rv64（f41）**
  - Ceph[ONGOING]
  - K8s[ONGOING]

Debian for RISC-V（于波）



- Official port update
  1. perl 5.40 transition
  2. LO got upload
  3. PR for enable xe driver
- Debci Update
  1. Adding new three p550 as workers
  2. rust-* enable testing again
- Some works
  1. lem [got upload], eclipse [upstream supported]
  2. linksem [waiting to upload], jimtcl [transition], rush [got upload]
  3. netplan.io[debci issue], mangohub [backport commit], hol-light [build ok]

testing/riscv64

RevyOS (程龙灿)

- New image (20241008)
  - https://github.com/revyos/mkimg-k230/releases/tag/2024.10.08
  - The new LicheePi 4A image is currently under testing
    - SDK : 2.0.2
    - https://github.com/revyos/thead-u-boot/actions/runs/10965384532
      new uboot sdk2.0.2 - lpi4amain
    - https://github.com/revyos/th1520-linux-kernel/actions/runs/10970502815
      only lpi4a works.
    - https://github.com/revyos/mkimg-th1520/actions/runs/10973771394
      new lpi4a image.
- **ROS2**
  - Sync upstream, compile and fix newly updated packages
  - Jazzy cache generation keeps failing
  - CI test results:

    Pass: (39,428/39,496) > (39,312/39,568)

    Failed:146, Skipped:102 > 110

    Total time: 6.05 hours

RevyOS supported devices

Image download directory

1、LicheePi 4A

2、LicheePi Cluster 4A

3、beaglev-ahead

4、Milk-V Pioneer

5、Milk-V Meles

6、LicheeConsole4A

7、RISC-V Book

8、LicheeBook

SD card support

1、LicheePi 4A

2、beaglev-ahead

3、Milk-V Meles

4、LicheeConsole4A

Mainline support

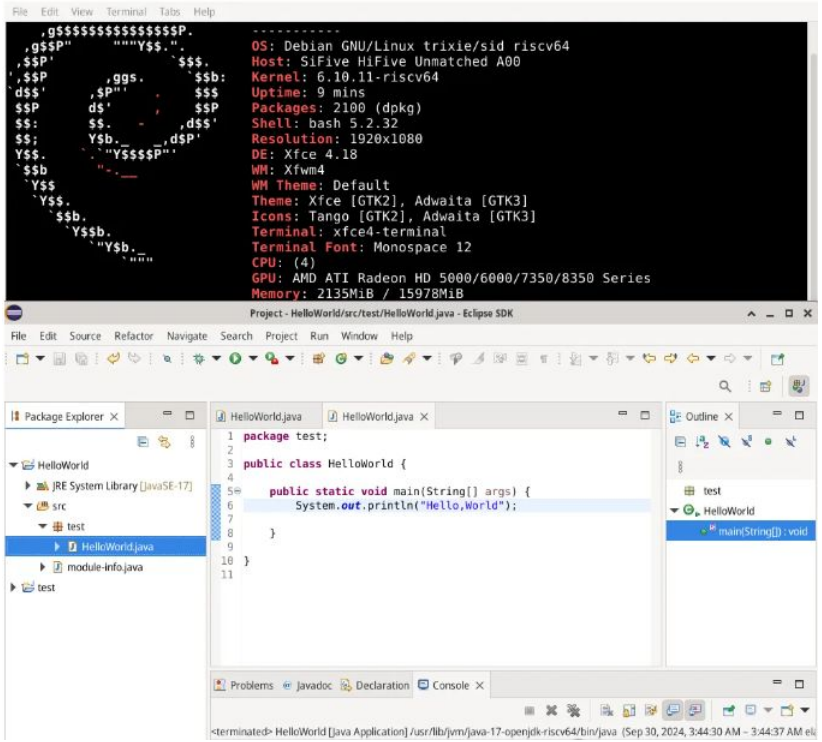
1、LicheePi 4A

2、Milk-V Pioneer

RevyOS (Longcan Cheng)

Eclipse upstream supports RISC-V ！

news release：https://mp.weixin.qq.com/s/qWvNienXYQU-PyAHP99Mdw



FW相关更新（王翔）

- opensbi
  - 修正sbi_cppc_write在RV32下丢失高32位的问题
  - 添加domain data支持，把domain context从静态数组修改为动态分配
  - pmp_set添加tor支持
  - 文档中添加构建内核的例子
  - 添加向量load/store的指令模拟
  - 修正is_region_subset中的整数移位溢出
  - 改进pmp和domain，移除rv32下4G物理内存的限制

固件相关更新(洛佳)

1. RustSBI Agent项目路线图
    a. 链接:参见
2. 新项目bouffaloader:博流芯片的异构引导程序
3. 请关注我们在GOSIM大会发表的演讲

香山开源RISC-V处理器 - ICT / PCL

- 功能
  - 前端
    - 修复一个取指块跨页导致的 gpaddr 被错误丢弃的 Bug (#3719)
    - 修复 Zcb 扩展中算术指令被识别为非法指令的问题 (#3721 OpenXiangShan/rocket-chip #14)
  - 后端
    - 修复连续 redirect 情况下 vtype 恢复错误 (#3705)
    - 修复 CSR 寄存器一系列读写以及权限检查相关错误 (#3717, #3701, #3700, #3703)
    - 修复 rtl 未正确指导 NEMU 中 mip.LCOFIP 更新的错误 (#3710, #572)
    - 修复向量访存异常处理相关错误 (#3722, #3720, #3714, #3704, #3702, #3695)
    - 修复 mstatus.FS 关闭时，vfslide1up/down 未报非法指令异常的错误 (#3696)
    - Debug 拓展支持 mcontrol6，废弃 mcontrol，以支持 H 拓展的硬件断点调试 (#3693)
    - Smrnmi 拓展：提供两种 NMI 中断，并提供中断线接口 (#3691)
  - 访存和缓存
    - 修复预取请求发生 guest page fault、以及发生 gpf 后出现重定向的处理逻辑 (#3697)
    - 修复非对齐请求未成功将 gpaddr 写入 htval 或 mtval2 寄存器的 Bug (#3699)
    - 修复非对齐 AMO 指令在 NEMU 上的 Bug
    - 继续完善向量 vstart、trigger 的支持、向量访存 fault-only-first 指令的实现，并修复一系列 Bug。目前已经合入主线 (#3690)
    - 性能计数器：Coupled L2 HPM 接入香山 CSR，并优化输出格式 (#3708)
    - CHIron (CHI Log)：完成二进制格式的设计实现，改善记录文件大小和解析速度
- 性能
  - OpenLLC：CHI-L3 接入 SoC 出现性能下降，原因定位到与转接桥并发度有关
  - CHI 转接桥：OpenNCB 支持大于 15 的内存请求并发；OpenNCB + NoC 框架下在双核情况下支持最大 45 的内存请求并发度
- 时序
  - 计划重构 LoadDependency 设计：通过重构 ldcancel 依赖链 cancel 逻辑，优化 issue queue 内时序，争取增加 issue queue 容量
  - 优化 MemBlock 中 TLB 物理摆放位置以及 TagArray 相关的关键路径，内部时序违例优化至 -47ps
  - 调整 CoupledL2 端口约束，L2 内部时序违例优化至约 -60ps

Chisel and Additional Technology / Sequencer

- t1
  - 缩减ci迭代至两组投片项目
  - 切换到 omreader 整合前中后端自动化流程
  - 持续优化permutation unit提高vgather执行带宽到原有的四倍左右
  - 进行批量后端评估
  - 提供更多的workloads评估性能
- chisel
  - 合入 SerializableModuleElaborator 为linking提供基建
  - 进行下一代chisel的feature定义
    - 整理出linking verification procedure等新需求的最小实现实行
    - 整理出下一代chisel的一系列 non compatible change 并进行讨论
    - Q4将提供一个Strict Chisel进行审核

OpenHW & OpenHW Aisa Working Group

ROCm bootstrapping for RISC-V (陆言, PLCT Tariser)

甲辰计划进展(吴伟)

- 签发人才背书至 017
- 安徽/合肥/HFUT学生交流、新增 RISC-V 实践点
- 南京 RISC-V Lab 建设进展
- RISC-V 开发板漂流继续活跃
- 甲辰计划组织筹办RISC-V芯片安全挑战赛
  - 主办方:苦芽科技、中科微 澜
  - ISCAS不再作为主办单位

自由讨论 / AOB

BACKUP

准备加入更多的国际开源组织进行同步观测

欢迎追加或提议