RISC-V开放指令集和软硬件生态

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

宋威

2018年8月16日

内容简介

- RISC-V指令集
  计算机体系结构与微结构、指令集分类、扩展、指令集设计思路等等。

- RISC-V基金会
  RISC-V基金会、RISC-V的商业模式、RISC-V商标和兼容性测试。

- RISC-V的软硬件生态
  在全球和中国的发展。

什么是RISC-V

- RISC-V是加州伯克利大学研发的第五代精简指令集（RISC）。
  - Krste Asanovic，加州伯克利教授，RISC-V基金会主席。
  - David Patterson，量化分析一书的作者，RISC概念的提出者，图灵奖获得者。
  - Andrew Waterman，SiFive首席工程师，Rocket处理器的作者。
  - Yunsup Lee，SiFive首席技术官，Rocket处理器的作者。

- RISC-V是一个开放指令集。
  - 体系结构≠处理器设计。
  - RISC-V只定义了软件和硬件的接口，而并不定义具体处理器实现。
  - RISC-V的设计是仔细考虑了硬件和软件实现的代价。
  - RISC-V是开放的，免费的，但是遵循RISC-V标准设计的CPU不一定开放并免费。

- RISC-V是一个全新的指令集
  - RISC-V在加州伯克利内部使用了几年，然后于2014年公开。
  - RISC-V指令集的设计吸取了过去几十年来各类指令集（Alpha，ARM, MIPS, Power，x86等）成功和失败的经验教训
  - RISC-V是一个模块化可扩展的指令集。能够支撑面向从IoT到高性能计算的所有应用。

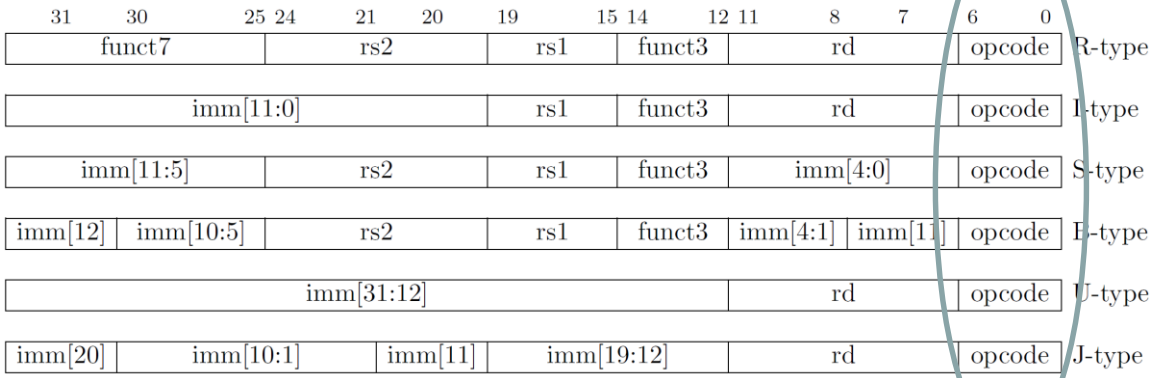
RISC-V的设计思路

- 简单的硬件支持最核心的指令功能。
  - 基本指令集的编码设计支持高效的指令解码硬件。
  - 所有基本指令都是单周期指令（不支持多周期指令）。
  - 不使用条件执行指令，简化处理器控制逻辑。
  - 非强制的缓存控制指令，简化一致性处理。
  - 支持安全的新特性（SMAP/SMEP/KPTI）。
  - 简单的通过16/32-bit混和指令长度就可以达到和X86、ARMv8同样的指令密度

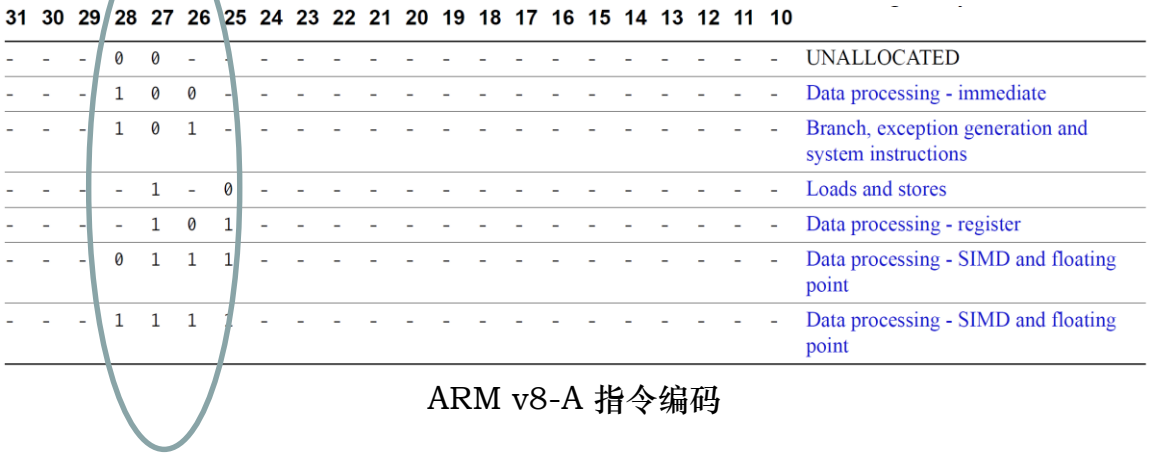
- 可扩展的指令集设计
  - 两个基本指令集（RV32和RV64）支持所有的基本功能。
  - 支持多种指令长度（16/32/48/64）
  - C：压缩指令集,16比特比特指令，面向超小系统和高性能系统。
  - A：原子操作指令集；M：整数乘除指令集。
  - F：单精度浮点计算；D：64位双精度浮点计算；Q：128位浮点计算指令集。
  - B：比特处理指令集；P：Packed SIMD指令集；V：vector指令集。
  - J：动态翻译（Dynamic Binary Translation）指令集。

RISC-V的编码设计

- 指令功能编码在地位（提前辨别16/32/48比特指令）。
- 源/目的寄存器地址位置恒定，便于硬件提取。
- 立即数比特位置尽量对齐，便于硬件提取。



RISC-V 指令编码



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	
-	-	-	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	UNALLOCATED
-	-	-	1	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Data processing - immediate
-	-	-	1	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Branch, exception generation and system instructions
-	-	-	-	1	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Loads and stores
-	-	-	-	1	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Data processing - register
-	-	-	0	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Data processing - SIMD and floating point
-	-	-	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Data processing - SIMD and floating point

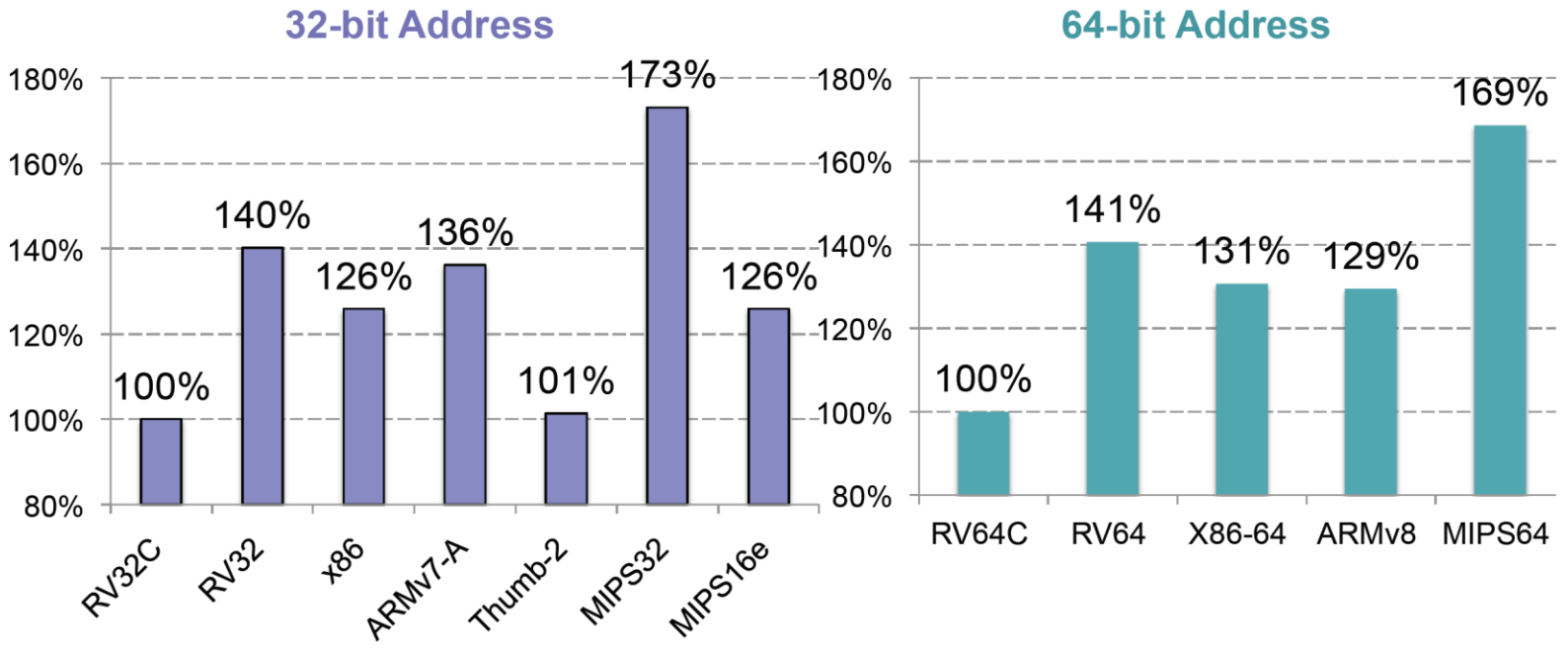
ARM v8-A 指令编码

RISC-V指令的选取

- RV32/64I 指令数<50, 简单的硬件支持最核心的指令功能。
- 支持高性能乱序处理器
  - 所有的指令都是单周期指令
    方便指令调度。
  - 没有条件执行指令（Predicated instructions）
    条件执行指令将控制引入了数据依赖，调度复杂，只在短小的条件执行片段有优势。当分支预测准确度提升之后，没有性能提高。
  - 没有多寄存器压栈和取栈指令
    多周期指令使得调度更加复杂，硬件更加复杂，可以通过millicode达到类似的代码密度，没有实际优势。
- 支持嵌入式处理器
  - 所有的指令都是单周期指令
    执行时间确定。
  - 支持压缩指令
    通过benchmark选择最常用的压缩指令子集。
  - 支持RV32E嵌入式指令集
    通过减少通用寄存器数量至16减小指令编码空间。

RVC压缩指令

- 每一条RVC指令对应一条RV指令
  - 硬件只需要扩展处理器缓存对RVC的翻译即可，不需更改解码和执行单元
  - RVC指令的插入变成了简单替换，可由连接器完成
  - 支持RVC和RV指令的混放



SPEC2006，Krste: “Why RISC-V ISAs Want to be Free”, 2017

支持micro-op fusion

- 处理器在一个周期执行两条指令
  - 不引入新的依赖关系
  - 不引入新的寄存器读口
  - 当作一条新的指令处理，只需扩展指令译码器
- 例子：
  - 远函数调用 jal 0xFFFFF804
    lui ra, 0xFFFF; jalr ra, ra, 0x804

  - 数组下标取值
    // rd = array[offset], x1 = &(array), x2 = offset
    add x1, x1, x2; ld rd, 0(x1)
    c.add x1, x1, x2; c.ld rd, 0(x1) // 32-bit

- 使用micro-op fusion + RVC可以构造新的32bit指令

支持系统页表分离KPTI

- 为了防止攻击者利用Meltdown漏洞读取系统页数据，RISC-V将在RISC-V priv (特权)标准 v1.2中加入KPTI支持，分离系统页表和用户也表。
  - 所有的进程共享一个系统页表(satp)
  - 每一个用户进程使用独立的用户也表(suatp)
  - 仍然支持共享页表系统

- RISC-V的实现方式
  - 系统写satp时会同时写suatp: 当系统页表被初始化时，默认用户页表失效，选择共享页表模式
  - 系统写suatp则只改变用户也表：支持用户也表只需要多一条CSR写操作
  - 如果使用共享页表方式，使用satp即可
  - 如果使用分离页表，进程上下文保存suatp即可（上下文操作没有本质改变）
  - 系统和用户使用不同的ASID

- RISC-V在定义扩展时会充分考虑兼容现有处理器实现

缓存控制指令

- RISC-V现在并没有缓存控制指令：flush, prefetch
- 用途：微处理器（scratchpad）和高性能服务器（榨取cache性能）
- 缓存控制指令的草案第6版
  - FENCE(有的架构也叫barrier)
  - 预取
  - 缓存锁定(把部分缓存的区域变成scratchpad)
  - 缓存清理(flush)
  - 其他破坏性缓存操作
    - MEM.DISCARD 直接失效缓存区域（数据丢失，用于抛弃无用数据而避免写回）
    - MEM.REWRITE 为写操作直接初始化缓存区域而不读取数据（已知数据会被彻底覆盖时，可以直接创建缓存数据同时标记已修改）
- 基本原则
  - 所有的缓存指令都必须是hint，可以当作NOP处理

其他已接受/讨论中的扩展

- 调试指令扩展
  - 支持semi-host，使用调试器当I/O
- 内存模型定义
  - 默认RVWMO 弱内存模型，可支持RVSTO 强模型
- Vector/Packed-SIMD
- Microcontroller profile (RV32E)
- 快速中断

RISC-V的组织形式

开放和商业化的有效折衷

- Krste Asanovic 是RISC-V商标的所有者。
  - Krste授权RISC-V基金会使用RISC-V商标。
  - 防止RISC-V基金会被大公司或者政府控制，最后的杀手锏。
- RISC-V基金会是RISC-V标准的管理机构
  - 在美国成立的非盈利组织，采取会员制。
  - RISC-V董事会成员在RISC-V的白金会中差额选举产生。
  - RISC-V董事会审核由工作小组提交的指令集扩展草案并组织投票。
  - RISC-V下设技术委员会和市场委员会。
- RISC-V技术委员会和市场委员会
  - 由RISC-V的可投票会员选举产生，负责管理RISC-V工作小组和市场营销。
- RISC-V工作小组
  - 由RISC-V会员组成，讨论并起草RISC-V指令集扩展草案。

RISC-V的会员

- RISC-V独立会员（Individual）
  - 可以参加工作小组讨论。
  - 99美元每年。
- RISC-V旁观会员（Audit）
  - 可以参加工作小组和委员会，但不可以投票。
  - 2,500美元每年。
- RISC-V白银会员（Silver）
  - 可以参与投票。
  - 5,000美元每年。
- RISC-V黄金会员（Gold）
  - 可以成为工作小组主席。
  - 10,000美元每年。
- RISC-V白金会员（Platinum）
  - 可以入选为RISC-V董事会成员。
  - 25,000美元每年。

RISC-V的会员（非官方解释）

- 所有的RISC-V会员都必须签署RISC-V的会员协议（by-law）。
  - 会员协议规定了会员的年费、投票权、参与权等等。
  - 会员协议规定了会员不可以保护其对RISC-V标准的贡献。即会员不可以专利化其对RISC-V标准的贡献，从而在将来索取专利费。
  - 会员之间不可以互诉对方侵犯其关于RISC-V的专利。

RISC-V扩展指令集的产生

- RISC-V技术委员会或市场委员会成立工作小组。
- 工作小组选取自己的主席（必须是黄金或白金会员）和副主席，制定标准扩展草案的轮廓。
- 当工作小组完成标准草案，草案向公众（非基金会成员）公布45天，搜集意见。
- 45天后，工作小组根据搜集的意见，选择做出改动然后递交基金会审阅，或重新编写草案重新公布。
- 如果提交基金会，工作小组必须对公示期内的所有意见提供反馈，并且对组内的反对意见给出解释。
- 如果草案被提交基金会，基金会董事会将根据草案和意见反馈决定是否展开投票。（董事会每月会举行一次）
- 如果投票通过，草案成为标准的一部分。

RISC-V的商业模式

- RISC-V是一个非盈利组织。
  - RISC-V基金会管理RISC-V指令集标准的制定，本身不从标准获利。
  - RISC-V基金会保证RISC-V指令集标准是开放并免费获取的。
  - RISC-V指令集的使用是免费的（有条件，见下）。
  - 声称兼容RISC-V指令集的处理器需要通过RISC-V兼容性测试。
  - 在商业产品中使用RISC-V商标需要成为RISC-V基金会成员。
  - （例外）在非商业研究和学术项目中可以免费使用RISC-V商标。
  - RISC-V对处理器本身的商业形式**没有任何限制**。

RISC-V的主要会员（2018 1Q：>90）

- 白金会员：
  - 谷歌：TPU
  - Microsemi：FPGA
  - nVidia: GPU controller, (**安全扩展**)
  - 西部数据: SD controller
  - 高通, Rambus, Marvell, Micron Technology, NXP, 三星, SiFive, UC Berkeley, 中兴微电子
- IC公司
  - BAE Systems， 晶心科技(**快速中断，ABI,GCC,LLVM**)， MediaTek，Lattice，西门子，华为，VeriSilicon，CEVA， Dolphin Integration，GlobalFoundries， IAR Systems， QuickLogic，希捷，C-sky
- EDA公司
  - Bluespec，UltreSoC，Imperas，Mentor
- 其他
  - IBM，特斯拉
- 中国公司
  - 华为， VeriSilicon，中科院计算所，中兴微电子，C-sky

一些较有名的RISC-V处理器实现

- Rocket
  - UC Berkeley, SiFive, lowRISC
  - 6级有序流水线，Chisel实现，SiFive的主打产品，对标ARM Cortex-A53
  - 多次流片
- BOOM
  - UC Berkeley
  - 乱序流水线，Chisel实现，对标ARM Cortex-A57
  - 多次流片
- PULP
  - ETH Zurich
  - 从1级到5级流水线，SystemVerilog实现，AXI总线，低功耗，微控制器。
  - 多次流片验证。

一些较有名的RISC-V处理器实现

- GRVI
  - Jan Gray针对FPGA高度优化的微处理器
  - Verilog设计，250~320 LUT
  - 在一个FPGA上实现1240个处理器的互联。
  - 商用授权，非开源。

- PicoRV32
  - Clifford Wolf设计的可流片的微处理器小核
  - Verilog设计，700~2000 LUT
  - >1GHz 主频，但是CPI在4~6之间。

- SHAKTI
  - 印度的国家处理器计划，选择RISC-V为国家指令集
  - 从微处理到服务器级别的处理器设计
  - Bluespec设计
  - Intel 22nm FinFET Low-power (FFL) 320MHz 0.7V CoreMark 2.2

软件生态环境

- 编译器相关
  - GNU GCC 进入主线 > 7.1
  - GNU Binutils 进入主线 > 2.28
  - GNU Glibc 进入主线 > 2.27
  - Newlib C library 进入主线 > 2.5.0
  - LLVM 进入主线 （8个月）
- 操作系统
  - Linux kernel 进入主线 > 4.15
  - Fedora 部分进入主线，可在QEMU上启动Fedora 27
  - Debian 部分进入9.5 unstable分支
  - FreeBSD 进入主线 > 11.0
- 仿真器、调试器
  - GDB 即将发布 （bare metal） > 8.2
  - QEMU 进入主线 2.12.0
  - Gem5 进入主线 SE模式
  - Imperas 全面支持
  - OVPSim 全面支持
  - MCU Eclipse 主持 bare metal

商业应用（除了SiFive）

- 西部数据
  - 全面使用RISC-V处理器替代现有产品中的控制器，1 billion cores per year。
- nVidia
  - 在下一代GPU中使用自制RISC-V控制器
- Rambus
  - 基于RISC-V的可信启动芯片
- MicroSemi
  - 携带RISC-V软核的FPGA开发平台
- 使用RISC-V处理器的公司(2018的新闻)
  - eSilicon，SerDes
  - Mobiveil，SSD controller
  - Delphin，RISC-V处理器IP
  - UltraSoC，全面支持RISC-V在线调试
  - Andes，全面兼容RISC-V指令集
  - Espranto，使用RISC-V做AI加速器
  - Dover，基于RISC-V的安全处理器
  - Codasip，GreenWaves，VectorBlox，Antmicro，Trinamic，CEVA

教学和研究

- 书籍
  - J. Hennessy, D. Patterson: *Computer Architecture: A Quantitative Approach*
  - D. Patterson, J. Hennessy: *Computer Organization and Design RISC-V Edition*
  - D. Patterson, A. Waterman: *The RISC-V Reader: An Open Architecture Atlas*
- 课程
  - 加州伯克利大学 CS61C EECS151
  - 麻省理工学院 6.175 Constructive Computer Architecture
  - 卡耐基梅隆大学 447 Introduction to Computer Architecture
  - 康奈尔大学 ECE 4750 Computer Architecture
  - 丹麦技术大学(DTU) Computer Architecture and Engineering course
- 研究
  - lowRISC: 每月被下载250余次，20多所国际大学和研究机构（2017年10月）

RISC-V在中国的发展情况

- 大家都很关心，讨论热烈。

- 已有自己的能跑RISC-V的处理器
  - 蜂鸟200系列（胡振波）https://github.com/SI-RISCV/e200_opensource
  - 对标ARM Cortex-M0

- 已有自己的中文书
  - 胡振波：手把手教你设计CPU——RISC-V处理器篇

- 已有公司发布了或即将发布使用RISC-V处理器的产品
  - 不过他们并不公开说他们没用ARM

- 6月30日，上海 RISC-V Day!
  - https://tmt.knect365.com/risc-v-day-shanghai/
  - 16个主题演讲

CNRV

- CNRV: RISC-V @ China
  - CNRV通过微信群、网站和公众号，与大部分国内RISC-V的使用者、参与者和爱好者建立了联系，包括软件/IC/硬件开发者、爱好者、相关公司高管、高校研究人员以及投资人等。
  - 每两周通过网站和公众号发布《CNRV双周简报》，向中文读者介绍和总结RISC-V相关领域的最新进展，可理解为RISC-V大参考～
  - CNRV网站搜集关于RISC-V的各种信息，为国内的RISC-V使用人员提供帮助。

- CNRV的主要成员
  - **郭雄飞**：景略半导体（上海）设计总监，RISC-V独立会员，RISC-V亚太区委员会成员，CNRV微信群的管理员，RISC-V双周简报编辑，国内最早接触和推广RISC-V的人。
  - **黄柏玮**：国立台湾大学咨询网路与多媒体研究所硕士，RISC-V独立会员，RISC-V调试小组和原比特操作小组成员，脸书RISC-V@Taiwan小组管理员。
  - **宋威**: 中科院信工所副研究员，原lowRISC工程硬件负责人， RISC-V双周简报编辑。

CNRV的网站

- https://cnrv.io/
- RISC-V双周简报
- 资源列表
- RISC-V镜像服务

- 访问量：
  每月访问次数~1100次

CNRV的资源列表

- https://cnrv.io/resource

- 已经搜集了26个RISC-V处理器设计。其中开源设计23个。

- 操作系统
- 开发工具
- 验证环境
- 文档

CNRV
为推广RISC-V尽些薄力

View On GitHub

RISC-V资源列表

处理器实现

- BOOM: Christopher Celio的RV64乱序处理器实现。Chisel, BSD Licensed。[GitHub] [Doc]
- BottleRocket: RV32IMC微处理器。Chisel，Apache Licensed。[GitHub]
- bwitherspoon: RV32微处理器。SystemVerilog, ISC Licensed。[GitHub]
- Clarvi: 剑桥大学教学用RISC-V处理器。SystemVerilog, BSD Licensed。[GitHub]
- F32: 针对FPGA的RV32微处理器，VHDL，BSD Licensed。[GitHub]
- GRVI: Gray Research LLC. 针对FPGA优化的RV32微处理器，commercial licensed。[Web]
- Hummingbird E200. 二级流水线，目标替代Cortex-M0/8051, Verilog, Apache 2.0 licensed。[GitHub]
- invicta: 一级流水线的RV32微处理器。Verilog，BSD Licensed。[GitHub]
- Kamikaze: RV32微处理器。Verilog，MIT Liencensed。[GitHub]
- KCP53000: Samuel A. Falvo II的RV64处理器实现。Verilog, MPL Licensed。[GitHub]
- nanorv32: 2机流水线的RV32实现。Verilog, GPLv2 Licensed。[GitHub]
- OpenV: 支持RV32的开源微处理器，Verilog，MIT Licensed，OnChipUIS，来源于哥伦比亚的Universidad Industrial de Santander。[GitHub]
- ORCA: 支持RV32的开源微处理器，VHDL，BSD Licensed，VectorBlox。[Github]
- PicoRV32: Clifford Wolf设计的(针对FPGA)RV32微处理器，Verilog，ISC Licensed。[GitHub]
- Potato: 针对FPGA的RV32微处理器。VHDL，BSD Licensed。[GitHub]
- RI5CY：支持RV32的开源微处理器
  - PULPino: SystemVerilog，Solderpad Licensed, 来源于苏黎世理工和博洛尼亚大学的PULP项目。[GitHub] [Web]
- RIDERCORE: RISC-V乱序处理器设计。Verilog, BSD Licensed。[GitHub]
- River: GNSS Senor Ltd.基于Rocket架构开发的RV64处理器。VHDL, BSD Licensed。[GitHub]
- Rocket: 支持RV64/32的开源处理器
  - Rocket-Chip: Chisel，BSD Licensed, Free chips project, UC Berkeley分离的开源工程。[GitHub]
  - Freedom: Chisel，Apache Licensed, SiFive, UC Berkeley分离的初创企业。[GitHub] [Web]
  - lowRISC：Chisel+SystemVerilog，Solderpad Licensed, 从剑桥大学发起的非盈利组织。[GitHub] [Web]
  - RoCC: the Rocket customized coprocessor interface 和Rocket处理器紧密互联的的协处理器接口。[BSG]
- RV12: RoaLogic的RV32微处理器。Verilog, RoaLogic non-commercial Licensed。[GitHub]
- SCR1: Syntacore的RV32开源微处理器。SystemVerilog，Solerpad Licensed。[GitHub]

CNRV的微信群和双周简报

- 受500人人数上限限制，现在已经有3个子群，约1000人。
- 每天的活跃用户~100人，消息约500~2000条每天。

- 微信公众号的关注到~1100人，日增长~10人。

- 双周简报的阅读数量在~600人，已经有了29期。

- 订阅双周简报，
  请扫二维码---------->

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

Thank You!