

2019年中国RISC-V论坛

思沃 (SERVE)

面向RISC-V生态的系统级原型验证服务平台

张科 常轶松 余子濠 唐丹 王海喆 于磊 张旭 赵然 包云岗

2019.11.13



中国科学院计算技术研究所
Institute of Computing Technology, Chinese Academy of Sciences



先进计算机系统研究中心

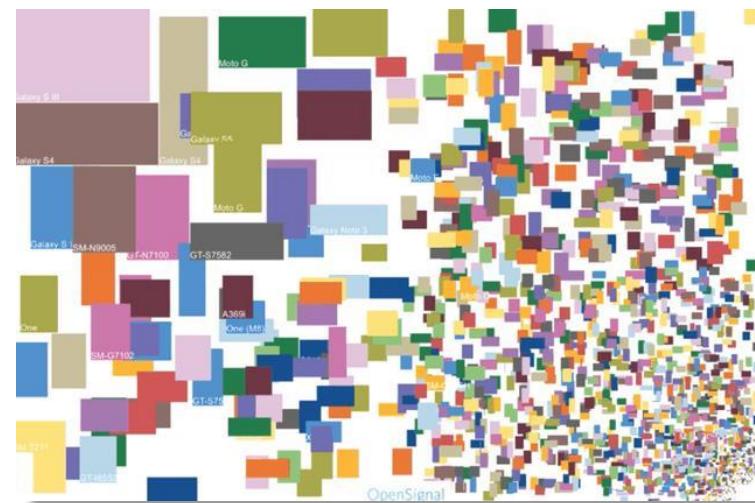
基于RISC-V的软硬件协同开发需求

领域专用体系结构

turing lecture



处理器芯片碎片化需求



RISC-V
软硬件深度融合优化

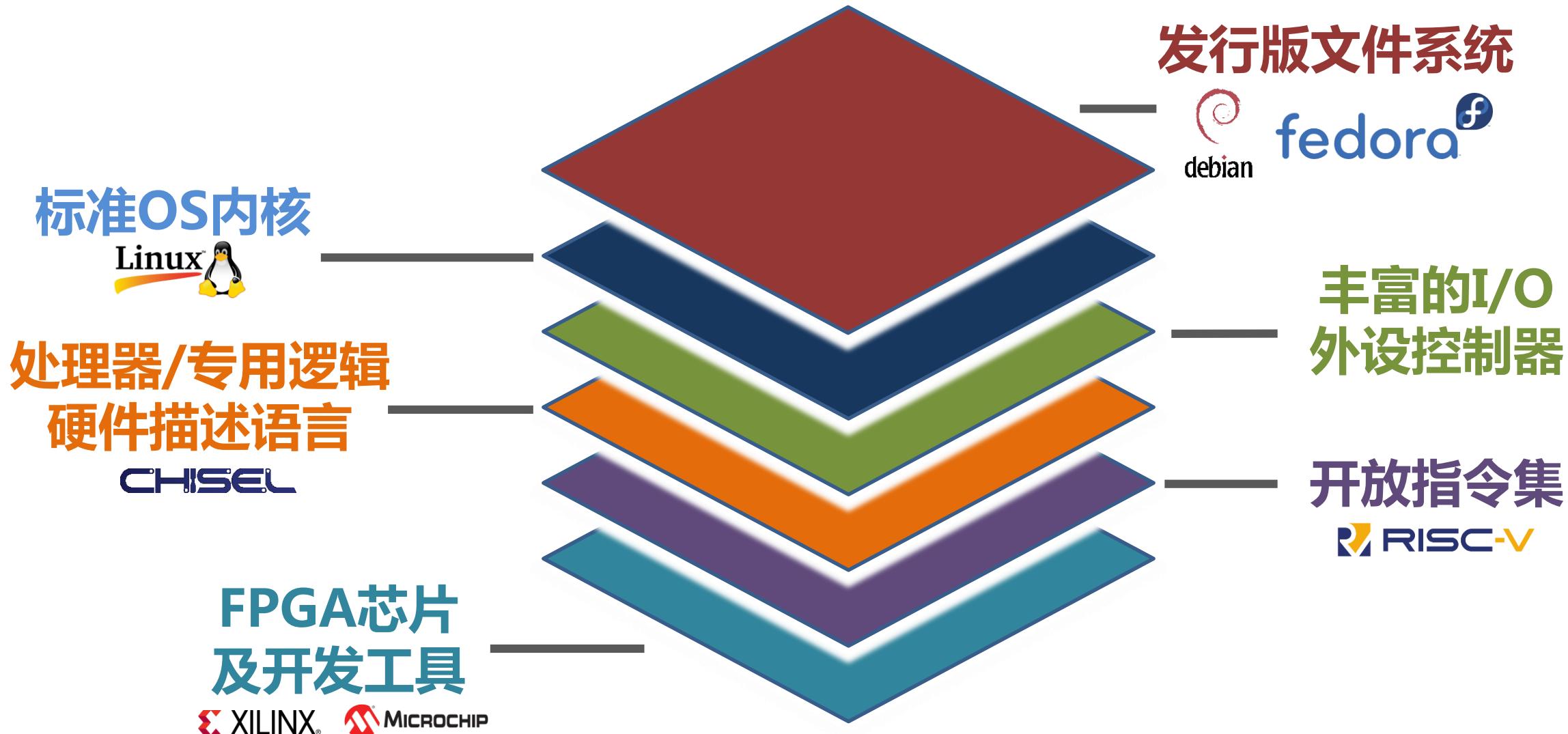
专用逻辑 + 领域软件

需要在芯片设计早期快速开展软硬件协同设计

软硬件协同设计平台

	软件模拟器 (QEMU、Spike)	FPGA 原型平台
成本开销		
使用门槛		
目标软件运行速度		
真实硬件	(C/C++模型)	可综合 RTL/Chisel代码
可扩展性		

面向RISC-V的FPGA原型平台生态



已有支持RISC-V的FPGA原型平台

	基于Zynq* 芯片的板卡	基于传统FPGA 芯片的板卡	新兴FPGA云 (亚马逊AWS F1公有云 实现x86服务器+传统 FPGA芯片板卡)
RISCV-V能否 使用板卡外设	不能使用 (外设由ARM控制)	可使用 (需额外添加控制器软IP 或实现低速控制器)	不可使用 (非真实硬件 : x86软件 模拟或RTL外设模型)
能否支持 发行版文件系统	需扩展NVMe SSD (成本高)	需扩展NVMe SSD (成本高)	虚拟块设备
RISC-V能否 上电自动启动	不支持 需额外配置	支持	不支持 需额外配置
典型代表	ETH PULP HERO 剑桥lowRISC UC Berkeley Rocket-Chip	SiFive Freedom ETH PULP Ariane	UC Berkeley FireSim

*Xilinx Zynq : 在同一个芯片封装内，包含一个ARM处理器硬核SoC和FPGA可编程逻辑

我们的FPGA原型平台设计目标

- RISC-V处理器可直接通过硬核IP访问板卡真实外设
- 使用低成本存储介质（如SD卡）支持发行版文件系统
- RISC-V处理器上电自启动，无需其他额外配置
- 涵盖从单板到云的多种平台形态，支撑不同设计规模

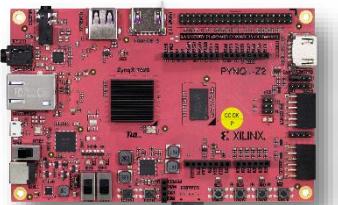
思沃(SERVE) 面向RISC-V生态的系统级原型验证服务平台

System **E**mulation and **P**rototyping for **R**ISC-**V** **E**nvironment



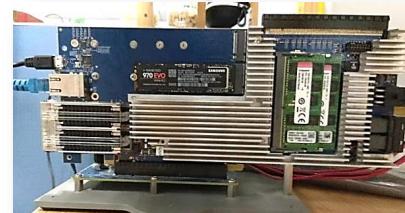
SERVE.r

基于Xilinx
PYNQ-Z2低成本
精简普及版



SERVE.i

基于FIDUS
Sidewinder-100
高性能增强版



SERVE.s

多节点边缘集群版
(精简/高性能集群)



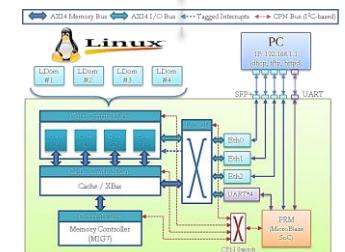
SERVE.c

云服务版
(精简/高性能云)

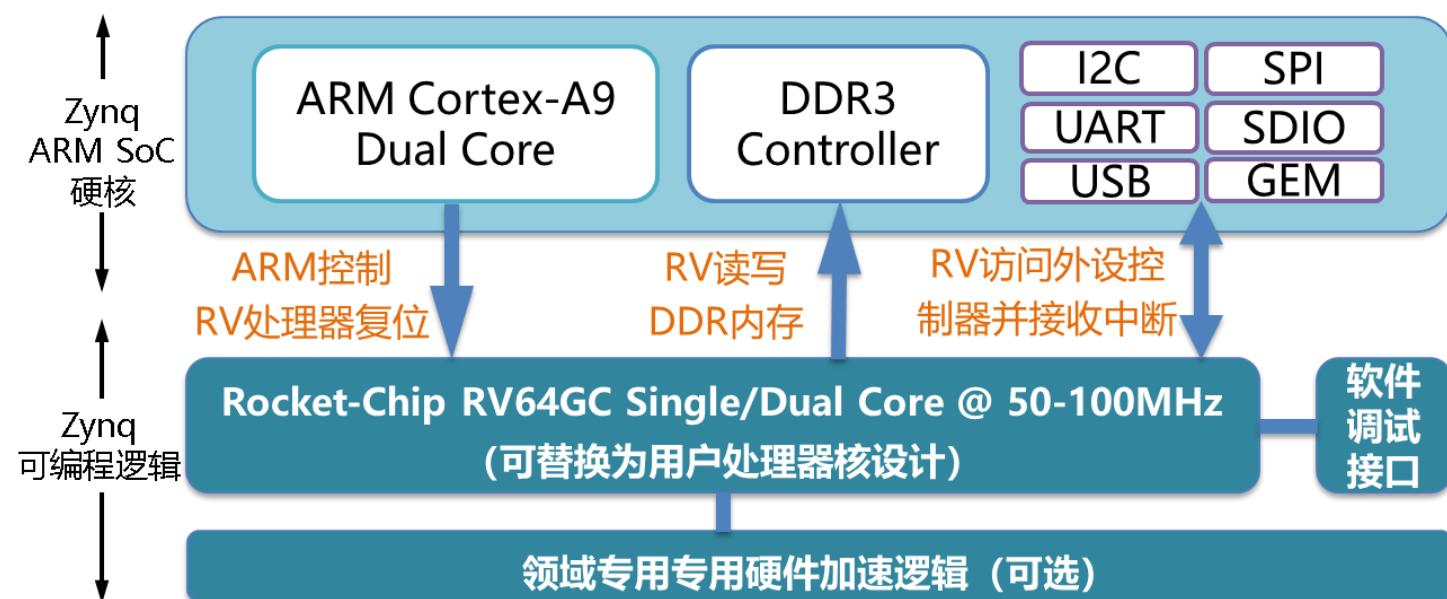
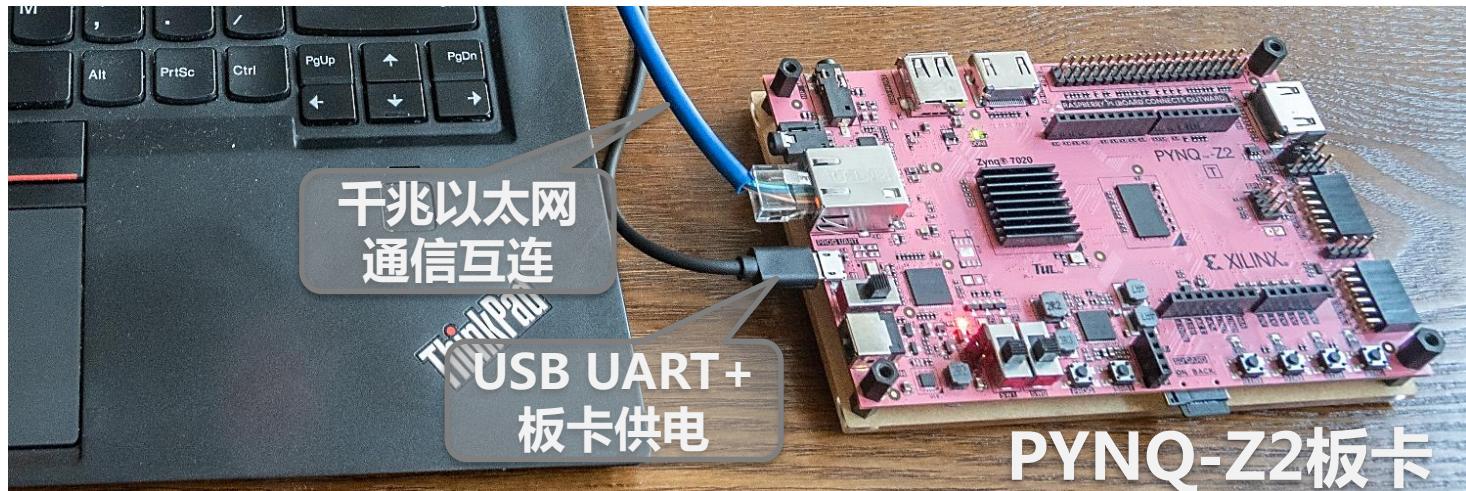


SERVE.v

标签化冯·诺伊曼
体系结构版本
(LvNA)



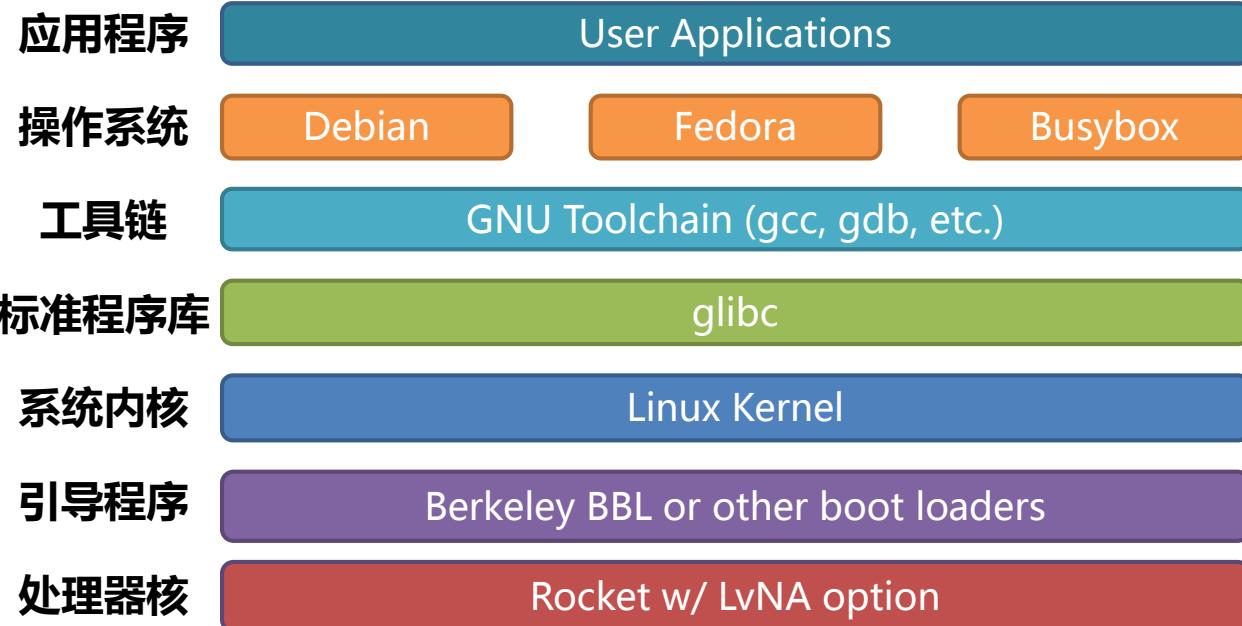
SERVE.r : 低成本RISC-V全系统原型验证平台



- **低成本**：价格低于 ¥ 1000
- **低功耗**：USB接口直接供电
- **低门槛**：开源软硬件设计
(整理中，陆续开放)
- **高可靠**：直接使用ARM SoC外设控制器IP硬核
- **在线使用**：可通过标准以太网 ssh 远程登陆

SERVE.r : 低成本RISC-V全系统原型验证平台

开源开放的全系统软硬件栈环境



The programs included with the **Debian GNU/Linux** system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

root@rv-hfu-01:~# uname -a
Linux rv-hfu-01 4.19.0-00056-g3735af2 #1 SMP Wed Jul 24 22:08:39 CST 2019 riscv64 GNU/Linux

启动标准Linux内核并挂载Debian文件系统

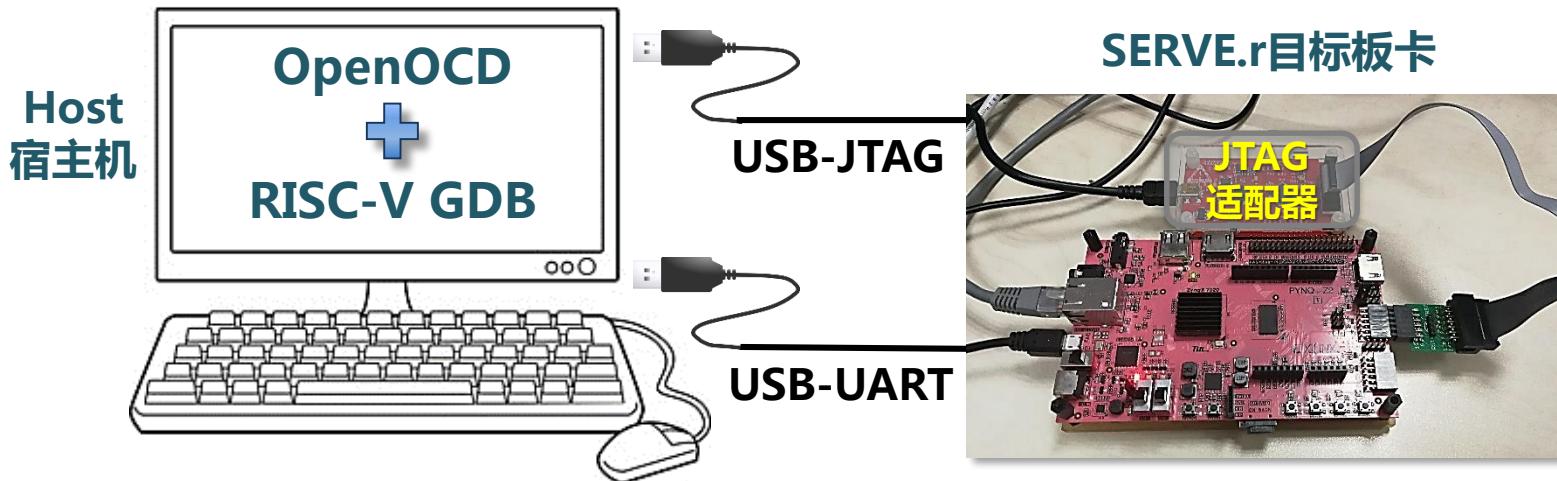
```
root@rv-hfu-01:~# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    link/ether a6:b2:91:23:3b:e5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.10/24 brd ff:ff:ff:ff:ff:ff scope global eth0
        valid_lft forever preferred_lft forever
        inet6 fe80::a6b2:91ff:fe23:3be5/64 scope link
            valid_lft forever preferred_lft forever
root@rv-hfu-01:~# ping 192.168.100.20
PING 192.168.100.20 (192.168.100.20) 56(84) bytes of data.
64 bytes from 192.168.100.20: icmp_seq=1 ttl=128 time=0.672 ms
64 bytes from 192.168.100.20: icmp_seq=2 ttl=128 time=0.542 ms
64 bytes from 192.168.100.20: icmp_seq=3 ttl=128 time=0.531 ms
```

以太网适配器 以太网:	连接特定的 DNS 后缀	:	fe80::511e:57d0:aefc:1eac%6
本地链接 IPv6 地址	:	192.168.100.20	
IPv4 地址	:	255.255.255.0	
子网掩码	:		
默认网关.	:		

无线局域网适配器 本地连接* 1:	媒体状态	:	媒体已断开连接
连接特定的 DNS 后缀	:		

千兆以太网通路测试

SERVE.r软件调试环境



调试 Bare Metal 程序

```
free@free-ThinkPad-T450:~/workspace/pynq
(gdb) load
Loading section .text, size 0xc2 lma 0x10000000
Loading section .data, size 0x34 lma 0x100010c8
Start address 0x10000000, load size 246
Transfer rate: 16 KB/sec, 123 bytes/write.
(gdb) set $pc=0x10000
(gdb) b _start
Breakpoint 1 at 0x10000000: file baremetal.S, line 4.
(gdb) c
Continuing.

Breakpoint 1, _start () at baremetal.S:4
4      jal main
(gdb) s
main () at baremetal.c:24
24      init_uart();
(gdb) n
27      for(i = 0;array[i];i++)
(gdb)
28          _putc(array[i]);
(gdb) p i
$1 = 0
(gdb) p array
$2 = "\r\n\r\n", '*' <repeats 14 times>, "\r\n Hello World!\r\n", '*' <repeats 1
4 times>, "\r\n"
(gdb) c
Continuing.
```

GDB 交互

```
free@free-ThinkPad-T450:~
RISC-V Execution in BootROM...
Passing Control to RISC-V Berkeley Boot Loader(BBL)...

*****
Hello World!
*****
```

串口输出

调试 Linux 内核

```
free@free-ThinkPad-T450:~/workspace/pynq/riscv-os-on-pynq/software/riscv-linux
Loading section __param, size 0x1e28 lma 0x107cf38
Loading section __modver, size 0x20 lma 0x107d1b0
Loading section __rodata, size 0x3b0 lma 0x107d1b80
Loading section __data, size 0x39fe0 lma 0x107d2000
Loading section __bug_table, size 0x5070 lma 0x1080bfe0
Loading section __sdata, size 0x298c lma 0x10811050
Loading section __ex_table, size 0x1880 lma 0x108dd610
Loading section __notes, size 0x3c lma 0x108dee90
Start address 0xffffffffe00000000, load size 6374900
Transfer rate: 246 KB/sec, 15624 bytes/write.
(gdb) set $pc=0x10200000
(gdb) d
Delete all breakpoints? (y or n) y
(gdb) hb start_kernel
Hardware assisted breakpoint 2 at 0xffffffffe00000650: file /home/free/workspace/pynq/riscv-os-on-pynq/software/riscv/linux/init/main.c, line 536.
(gdb) disp $pc
1: x/5l $pc
=> 0x10200000: csrw    sie,zero
  0x10200004: auipc   gp,0x612
  0x10200008: addi    gp,sp,-1972
  0x1020000c: lui     t0,0x6
  0x1020000e: csrrc   sstatus,t0
(gdb) c
Continuing.

Breakpoint 2, start_kernel ()
  at /home/free/workspace/pynq/riscv-os-on-pynq/software/riscv/linux/init/main.c:536
536      set_task_stack_end_magic(&init_task);
1: x/5l $pc
=> 0xffffffffe00000650 <start_kernel+12>:    auipc   a0,0x5d9
  0xffffffffe00000654 <start_kernel+16>:    addi    a0,a0,1648
  0xffffffffe00000658 <start_kernel+20>:
  jal ra,0xffffffffe000035700 <set_task_stack_end_magic>
  0xffffffffe0000065c <start_kernel+24>:
  jal ra,0xffffffffe000020be <smp_setup_processor_id>
  0xffffffffe00000660 <start_kernel+28>:
  jal ra,0xffffffffe000006522 <cgroup_init_early>
(gdb) []
```

SERVE.r镜像文件在开源托管平台提供下载

The screenshot shows the iHub platform interface for the SERVE.r project. The top navigation bar includes links for Activities (1), 任务, Resources, Repositories (3), Pull Requests, Roadmap, Calendar, and Gantt. The main content area displays a detailed description of the project, highlighting its features such as System-level Verification and Prototyping for RISC-V Environment. It also lists the main characteristics of the platform. Below this, there is a section for the SERVE.r Platform User Manual, contact information (changyisong@ict.ac.cn), and a QR code for quick access.

鹏城实验室iHub

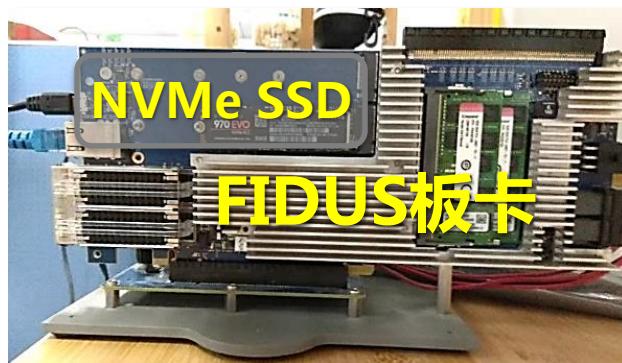
The screenshot shows the GitHub repository page for ict-accel-team/SERVE.r. The repository summary indicates 3 commits, 1 branch, 0 releases, and 1 contributor. The repository description states: "The reduced version of System Emulation and Prototyping for RISC-V Environment (SERVE) with Xilinx PYNQ-z2 board". The repository topics are managed under "Manage topics". A list of recent commits is shown, including updates to README and bin files. Below the repository details, there is a section for the SERVE.r Platform User Manual, contact information (changyisong@ict.ac.cn), and a QR code.

GitHub

SERVE.r应用实例

- **2019秋季中国科学院大学操作系统实验课程基础教学平台**
 - 实现基于RV64GC指令集的简易操作系统内核
- **2019年国际人工智能系统大赛**
 - 赛道1：基于RISC-V的卷积神经网络图片分类软件算法移植优化
(<http://www.benchcouncil.org/competition/cn/index.html>)

SERVE.i：基于FIDUS的高性能RISC-V全系统平台



PCIe
扫描

```
xilinx-pcie 40000000.axi-pcie: PCIe Link is UP
xilinx-pcie 40000000.axi-pcie: host bridge /amba_pl@0/axi-pcie@70000
xilinx-pcie 40000000.axi-pcie: No bus range found for /amba_pl@0/
xilinx-pcie 40000000.axi-pcie: MEM 0x70000000..0x7fffffff -> 0x70000000
xilinx-pcie 40000000.axi-pcie: PCI host bridge to bus 0000:00
pci_bus 0000:00: root bus resource [bus 00-ff]
pci_bus 0000:00:00:00:0: BAR 0: assigned [mem 0x70000000-0x73ffffffff 64bit]
pci 0000:00:00.0: BAR 8: assigned [mem 0x74000000-0x740fffff]
pci 0000:00:00.0: BAR 0: assigned [mem 0x74000000-0x74003fff 64bit]
pci 0000:00:00.0: PCI bridge to [bus 01]
pci 0000:00:00.0: bridge window [mem 0x74000000-0x740fffff]
nvme nvme0: pci function 0000:01:00.0
```

```
[ OK ] Started Remove Stale Online Ext4 Metadata Cache
[ OK ] Reached target Graphical Interface.
      Starting Update UTMP about System Runlevel Change...
[ OK ] Started Update UTMP about System Runlevel Change...
```

```
Debian GNU/Linux bullseye/sid label-riscv hvc0

label-riscv login: root
Password:
Last login: Thu Feb 14 10:16:45 UTC 2019 on hvc0
Linux label-riscv 4.18.0-ga57318a4-dirty #41 SMP Wed
```

```
The programs included with the Debian GNU/Linux system,
the exact distribution terms for each program are
individual files in /usr/share/doc/*copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY,
permitted by applicable law.
root@label-riscv [10:15:21 ~] $ cat /proc/cpuinfo
hart : 0
isa : rv64imac
mmu : sv39
uarch : sifive,rocket0
root@label-riscv [10:15:27 ~] $
```

Debian
登录

cpuinfo
输出

```
root@fidus-88:~/master# ifconfig eth0 192.168.1.1
root@fidus-88:~/master# ssh root@192.168.1.2
root@192.168.1.2's password:
Linux label-riscv 4.18.0-ga57318a4-dirty #41 SMP Wed
```

```
The programs included with the Debian GNU/Linux system,
the exact distribution terms for each program are
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY,
permitted by applicable law.
Last login: Thu Feb 14 10:15:37 2019
root@label-riscv [10:16:22 ~] $
```

```
root@label-riscv [10:18:55 ~] $ cd gcc-example/
root@label-riscv [10:18:58 ~/gcc-example] $ ls
root@label-riscv [10:18:58 ~/gcc-example] $ vim hello.c
root@label-riscv [10:19:29 ~/gcc-example] $ cat hello.c
#include <stdio.h>
int main() {
    printf("Hello Labeled RISC-V!\n");
    return 0;
}
root@label-riscv [10:19:33 ~/gcc-example] $ gcc hello.c
root@label-riscv [10:19:41 ~/gcc-example] $ ./a.out
Hello Labeled RISC-V!
root@label-riscv [10:19:44 ~/gcc-example] $

root@label-riscv [10:20:00 ~] $ python3
Python 3.7.4 (default, Jul 11 2019, 10:43:21)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello Labeled RISC-V!')
Hello Labeled RISC-V!
>>>
root@label-riscv [10:20:20 ~] $
```

ssh登录

gcc编译

运行python

```
root@label-riscv [10:20:52 ~/java-example] $ ls
Hello.java
root@label-riscv [10:20:56 ~/java-example] $ cat Hello.java
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello Labeled RISC-V!");
    }
}
root@label-riscv [10:21:04 ~/java-example] $ javac Hello.java
root@label-riscv [10:32:41 ~/java-example] $ java Hello
Hello Labeled RISC-V!
root@label-riscv [10:33:09 ~/java-example] $
```

```
root@label-riscv [10:33:14 ~] $ ls
gcc-example hello-x86-native java-example microbench-x86-native nexus-am qemu
root@label-riscv [10:33:18 ~] $ file hello-x86-native
hello-x86-native: ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), st
ed
root@label-riscv [10:33:26 ~] $ ./qemu/i386-linux-user/qemu-i386 hello-x86-native
Hello World from a(n) x86 program
```

通过qemu运行x86程序

java编译

SERVE.s : 集群版本RISC-V全系统平台



- 集成多个通过标准千兆以太网络互连的SERVE.r或SERVE.s节点
- 便于多人同时同地开展RISC-V系统级验证
- 基于Debian生态，尝试构建并行编程环境

SERVE.c : 云平台版本

基于网络
7 x 24小时
不间断服务

真实硬件
Scale-Out

提供更多FPGA
逻辑资源及
异构加速能力

提供更加真实的
数据中心与云计算
应用负载评估环境



32路基于
Zynq SoC
FPGA的
RISC-V
全系统节点

动态弹性分配
硬件板卡资源

高密度
低功耗

支持更大规模
的并发验证
和远程调试

RV-Prototype-
as-a-Service

SERVE.v：标签化冯诺依曼体系结构 LvNA

Labeled RISC-V FPGA原型系统配置

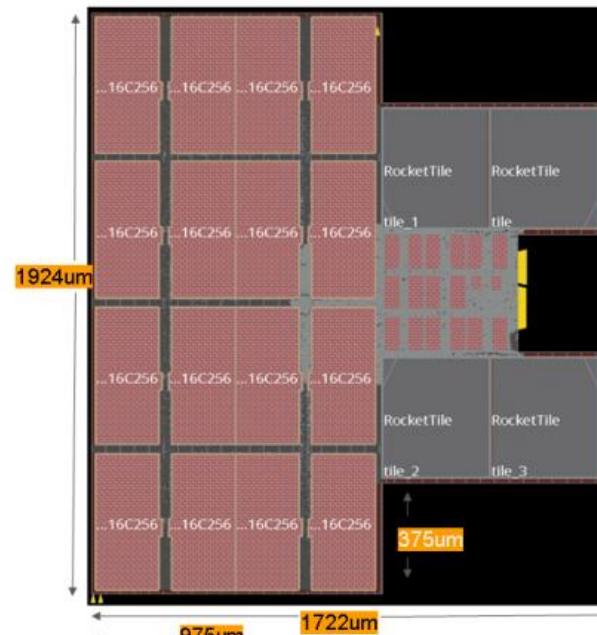
- rocket核心 * 4
- 16KB L1 I\$, 16KB L1 D\$, 2MB L2\$
- 千兆以太网
- 可运行Redis、Memcached等应用
- **支持标签化特色的性能调控技术**



基于
FIDUS
板卡部署

GF 14nm ASIC流片评估结果

- Slow Corner, 0.72V, 125C
- 核心频率1.25GHz
- L2 Cache频率625MHz
- 芯片面积1.9mm x 1.7mm



Yungang Bao, Sa Wang, **Labeled von Neumann Architecture for Software-Defined Cloud**. *Journal of Computer Science and Technology (JCST)*, 32(2): 219-223, 2017.

硬件应用案例：我们的开源微结构实现改进计划

均在 SERVE.i 或 SERVE.v 上开展

已完成
进行中
计划中

▶ Bug修复

- FIRRTL编译性能bug
- RAS性能bug
- BOOM的LSQ功能bug

▶ 创新尝试

- 资源低开销的OoO调度器设计
- 自动化末级缓存容量划分
- 多bank的寄存器堆
- LSQ内存模型

▶ 性能改进

- 神经网络分支预测器
- DRRIP末级缓存替换算法
- Non-blocking的流水化末级缓存
- L1/LLC预取
- Load-to-use延迟优化

▶ 测试验证和设计方法

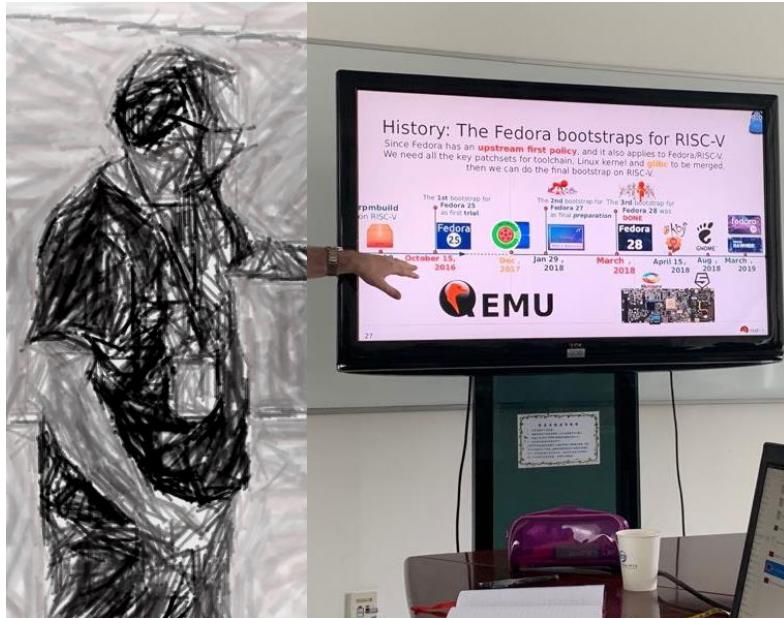
- 基于Lock-step的差分测试
- BOOM的断点支持
- 自动流水化设计范式

- ▶

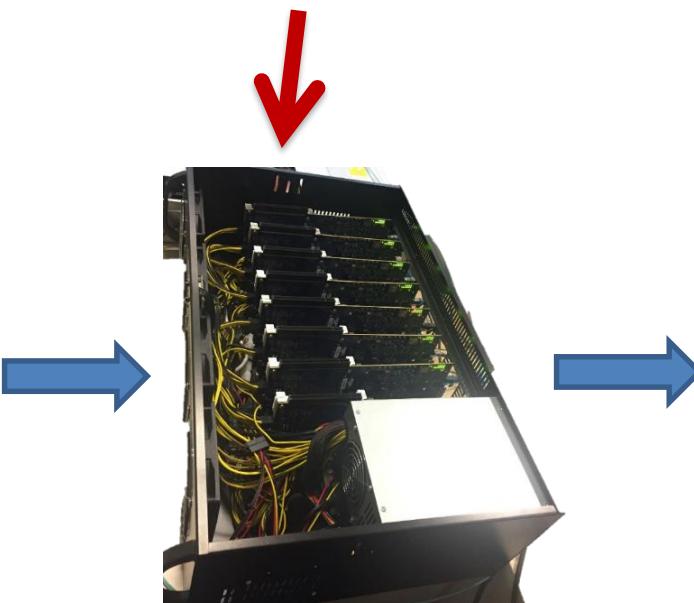
软件应用案例：帮助RedHat移植Fedora

RedHat工程师远程登录使用**火苗**进行Fedora的移植

火苗 = SERVE.c + SERVE.v



RedHat工程师到访计算所



```
[ OK ] Starting Hostname Service...
[ OK ] Started Permit User Sessions.
Starting Terminate Plymouth Boot Screen...
Starting Hold until boot process finishes up...
[ OK ] Started Terminate Plymouth Boot Screen.
[ OK ] Started Hold until boot process finishes up.

Welcome to the Fedora/RISC-V disk image
https://fedoraproject.org/wiki/Architectures/RISC-V

Build date: Wed Jul  3 20:19:49 UTC 2019

Kernel 4.18.0-ga57318a4-dirty on an riscv64 (hvc0)

The root password is ..riscv...
To install new packages use 'dnf install ...'
To upgrade disk image use 'dnf upgrade --best'
If DNS isn't working, try editing ..etc/yum.repos.d/fedora-riscv.repo...
For updates and latest information read:
https://fedorapeople.org/groups/risc-v/disk-images/readme.txt

Fedora/RISC-V
-----
Koji: http://fedora-riscv.tranquillity.se/koji/
SCM: http://fedora-riscv.tranquillity.se:3000/
Distribution rep.: http://fedora-riscv.tranquillity.se/repos-dist/
Koji internal rep.: http://fedora-riscv.tranquillity.se/repos/
[ 230.410000] tx_irq = 4
[ 230.410000] rx_irq = 5
fedora-riscv login: root
Password:
[root@fedora-riscv ~]# passwd

```

在火苗上成功登录Fedora

感谢关注

SERVE.r开源软硬件环境

<https://code.ihub.org.cn/projects/373>

<https://github.com/ict-accel-team/SERVE.r>

标签化冯·诺伊曼体系结构开源项目

<https://code.ihub.org.cn/projects/159>

<https://github.com/LvNA-system/labeled-RISC-V>

加入我们



联系我们

crva@ict.ac.cn



中国开放指令生态 (RISC-V) 联盟
China RISC-V Alliance



中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES



先进计算机系统研究中心
Center for Advanced Computer Systems



鹏城实验室
Peng Cheng Laboratory