MachineWare

SIM-V

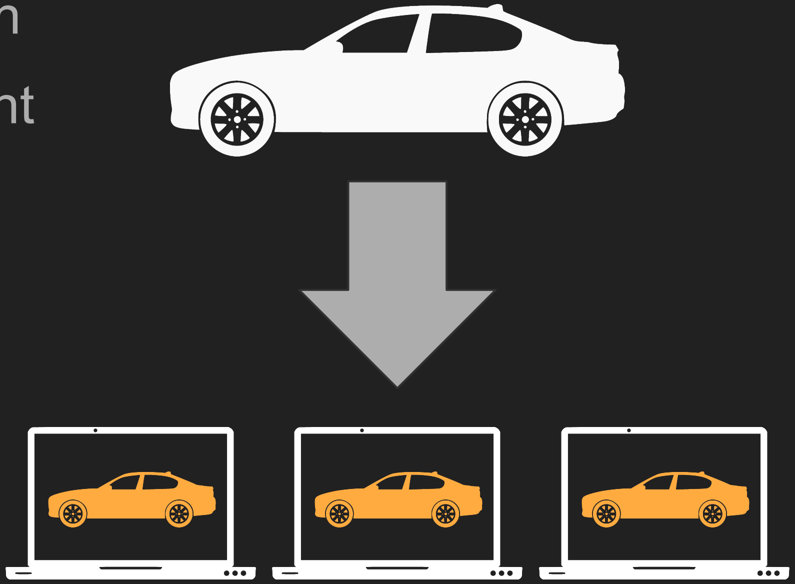
Fast, Parallel RISC-V Simulation for Rapid Software Verification

Lukas Jünger, MachineWare GmbH
Jan Henrik Weinstock, MachineWare GmbH
Rainer Leupers, RWTH Aachen University

As presented at DVCON Europe 2022

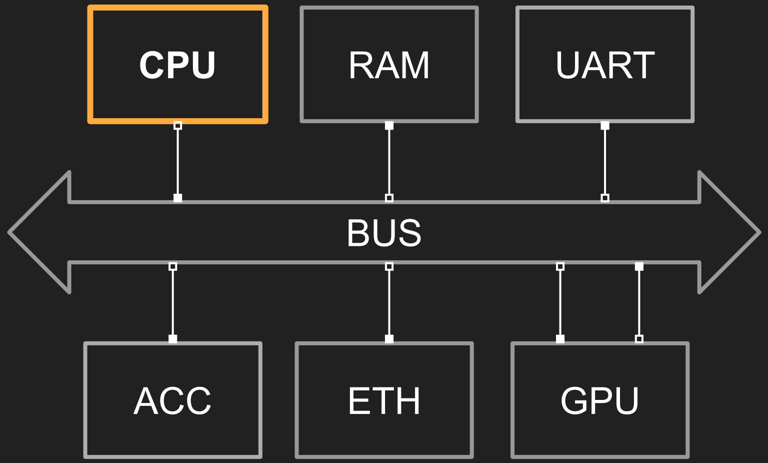
Virtual Prototyping

- Virtual Platform: Full System Simulation
- **Indispensable** in software development
  - Everising SW and HW complexity
- Advantages over physical prototypes
  - Available earlier (shift-left methodology)
  - Full flexibility, deep introspection
  - Non-intrusive debug
  - More scalable (e.g. in the cloud)



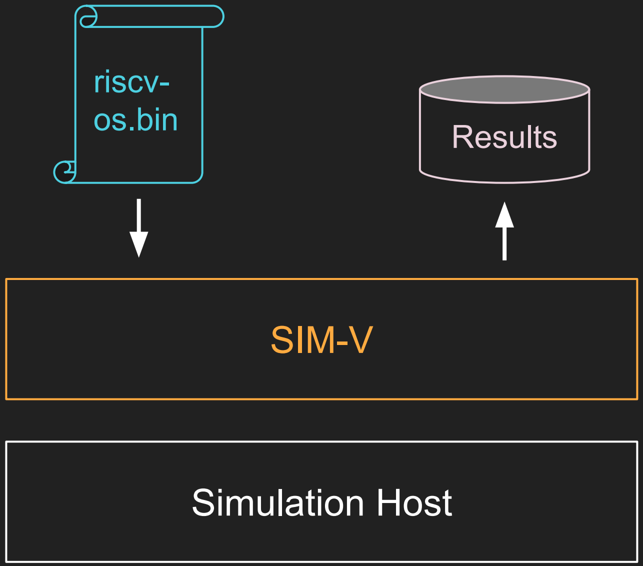
Virtual Platform Challenges

- Insufficient performance
- High complexity
- Limited vendor interoperability
- Limiting factors
  - Modern simulation infrastructure
  - Efficient standardization
  - **Fast models (especially CPU)**

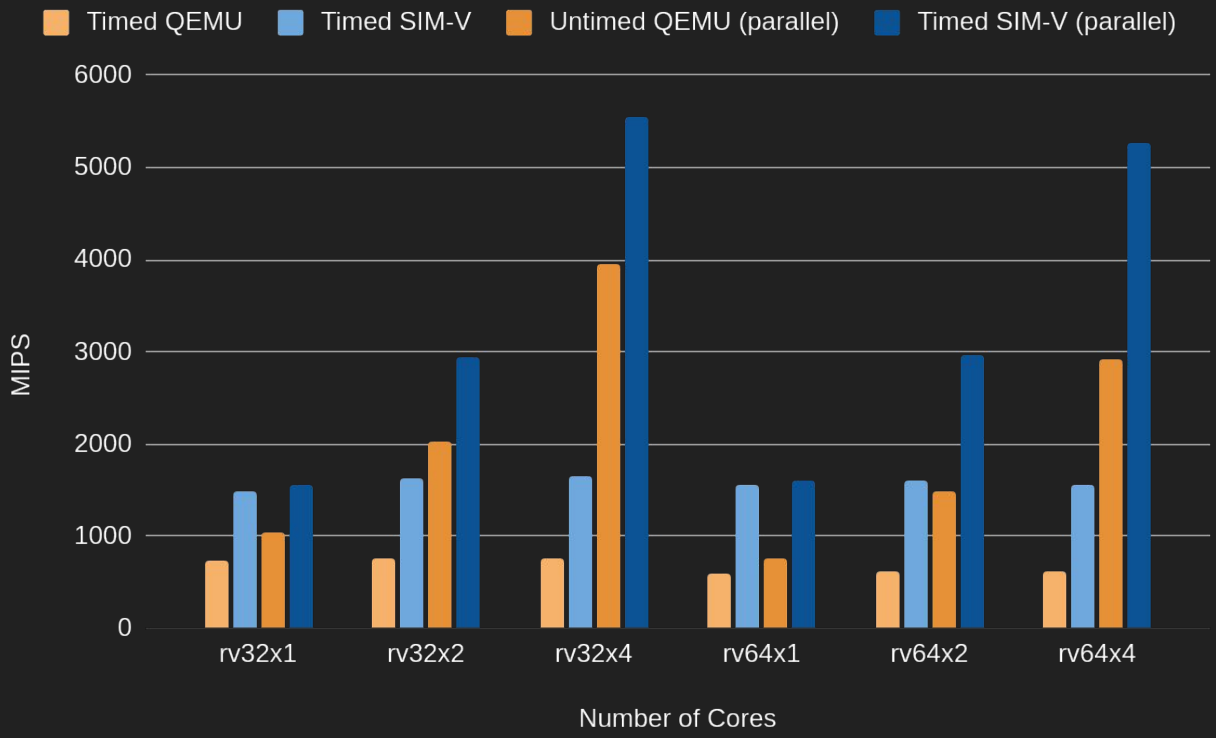


MachineWare SIM-V

- Fast, functional RISC-V simulator
- **2.5x faster** than state-of-the-art
- Shift-left: Better software earlier
- Easy to use and integrate



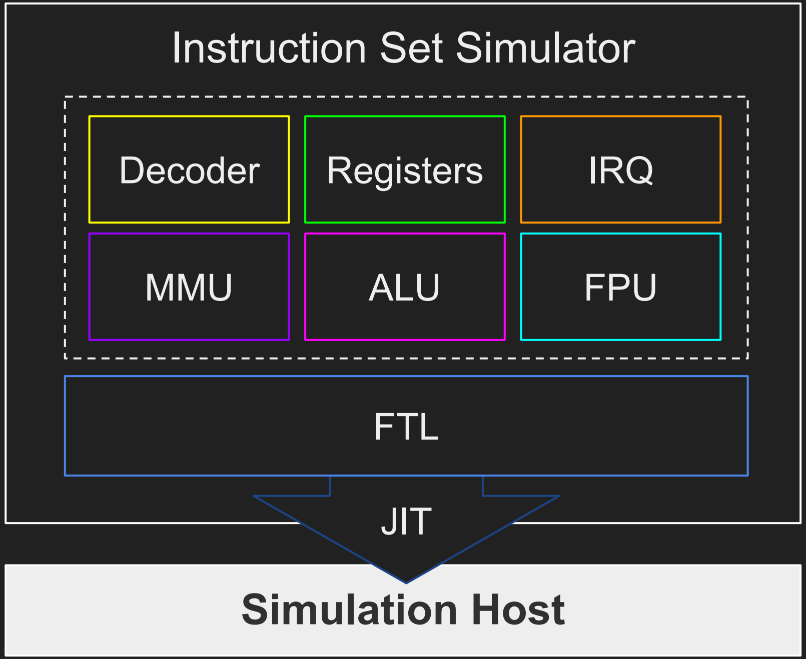
SIM-V vs. QEMU: Dhrystone



FTL - Fast Translator Library

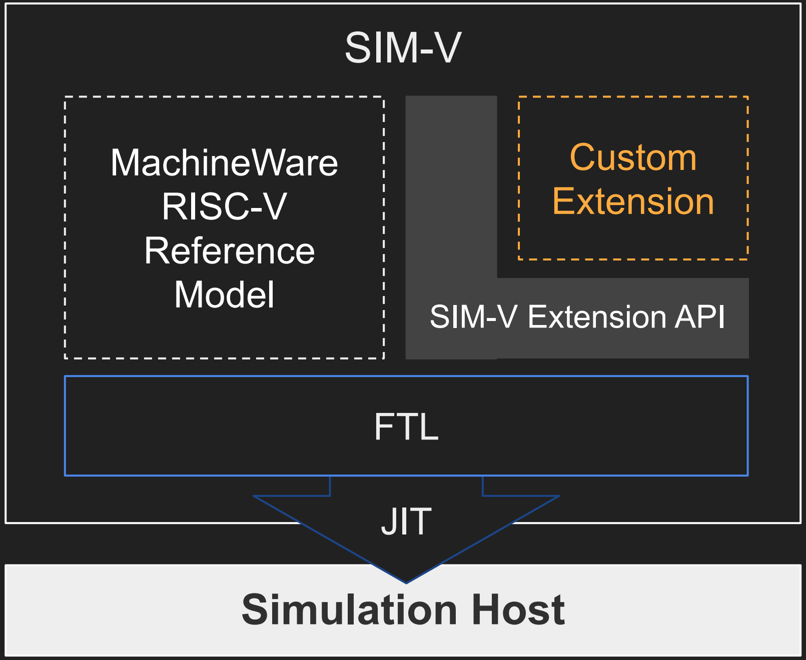
Processor Modeling and High-Speed Dynamic Binary Translation Toolkit

- Model your processor in C++
- **Fast JIT** binary translation
- No target software limitations

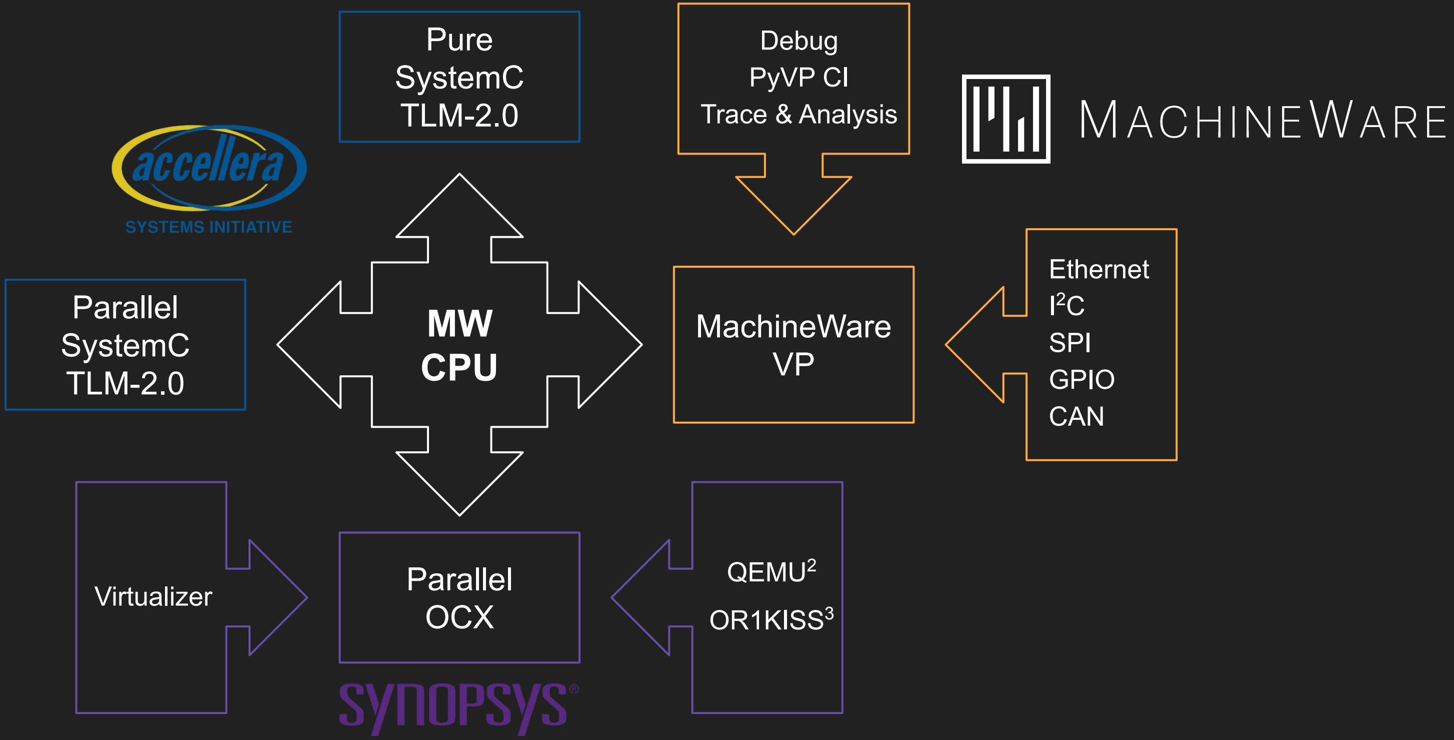


SIM-V Custom Extensions

- Add instructions, registers, …
- **SDK** for extension development
- Automatically loaded by SIM-V
- Leverage FTL for performance



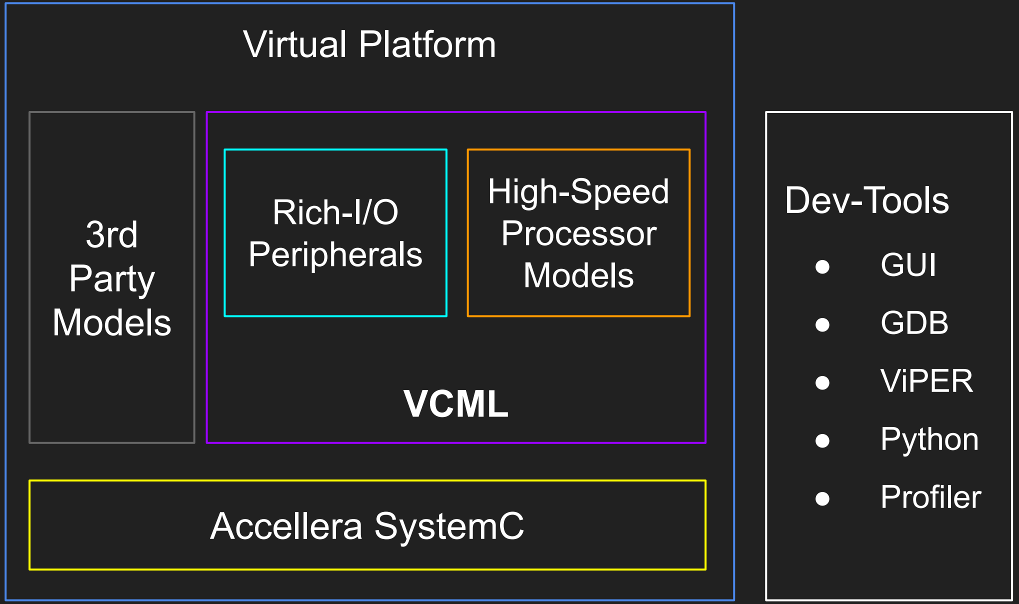
Interoperability



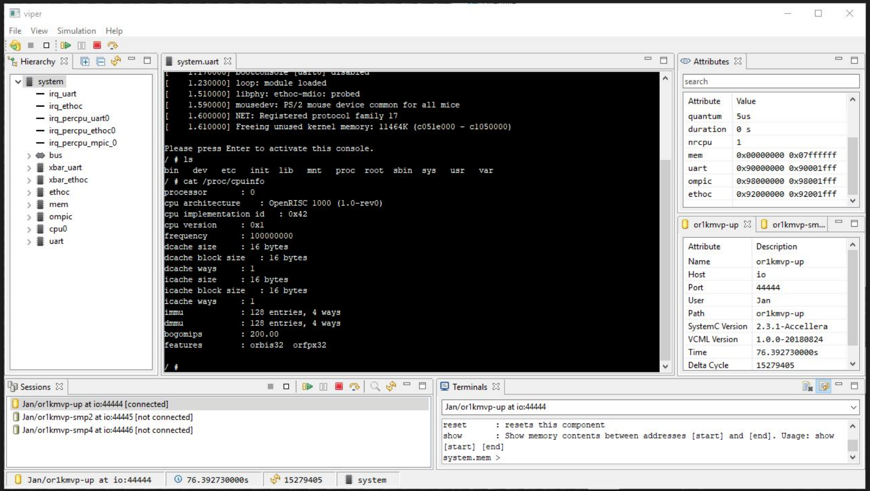
VCML - Virtual Components Modeling Library

Rapid Virtual Platform construction and universal tool integration

- **Open-source** license
- Component models
- Tool integrations
- TLM protocols
- Modeling primitives



MachineWare Virtual Platform ExploreR (ViPER)



```
[    1.170000] bootconsole [uart0] disabled
[    1.230000] loop: module loaded
[    1.510000] libphy: ethoc-mdio: probed
[    1.590000] mousedev: PS/2 mouse device common for all mice
[    1.600000] NET: Registered protocol family 17
[    1.610000] Freeing unused kernel memory: 11464K (c051e000 - c1050000)

Please press Enter to activate this console.
/ # ls
bin   dev   etc   init  lib   mnt   proc  root  sbin  sys   usr   var
/ # cat /proc/cpuinfo
processor       : 0
cpu architecture    : OpenRISC 1000 (1.0-rev0)
cpu implementation id   : 0x42
cpu version     : 0x1
frequency       : 100000000
dcache size     : 16 bytes
dcache block size   : 16 bytes
dcache ways     : 1
icache size     : 16 bytes
icache block size   : 16 bytes
icache ways     : 1
immu            : 128 entries, 4 ways
dmmu            : 128 entries, 4 ways
bogomips        : 200.00
features        : orbis32  orfpx32

/ #
```

Attribute	Value
quantum	5us
duration	0 s
nrcpu	1
mem	0x00000000 0x07ffffff
uart	0x90000000 0x90001fff
ompic	0x98000000 0x98001fff
ethoc	0x92000000 0x92001fff

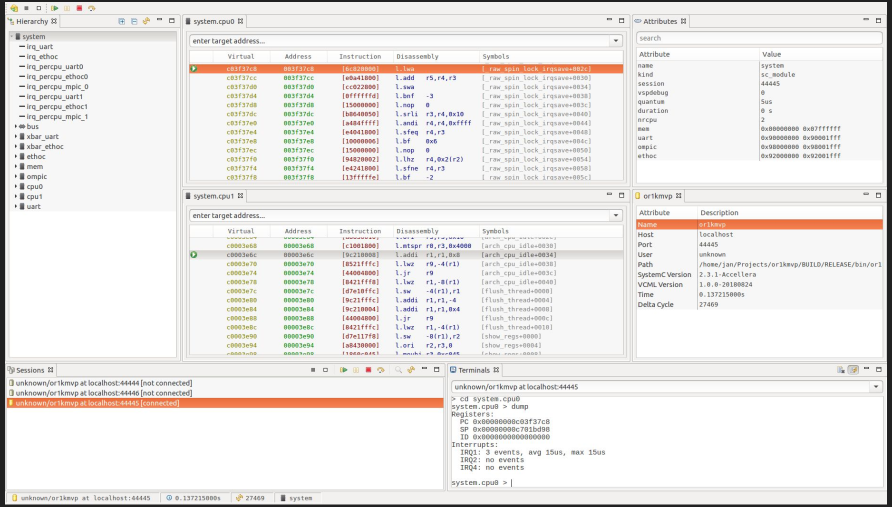
Attribute	Description
Name	or1kmvp-up
Host	io
Port	44444
User	Jan
Path	or1kmvp-up
SystemC Version	2.3.1-Accellera
VCML Version	1.0.0-20180824
Time	76.392730000s
Delta Cycle	15279405

Jan/or1kmvp-up at io:44444 [connected]
Jan/or1kmvp-smp2 at io:44445 [not connected]
Jan/or1kmvp-smp4 at io:44446 [not connected]

Jan/or1kmvp-up at io:44444

```
reset      : resets this component
show       : Show memory contents between addresses [start] and [end]. Usage: show
[start] [end]
system.mem >
```

Jan/or1kmvp-up at io:44444 | 76.392730000s | 15279405 | system



Virtual	Address	Instruction	Disassembly	Symbols
c03f37c8	003f37c8	[6c820000]	l.lwa	[_raw_spin_lock_irqsave+002c]
c03f37cc	003f37cc	[e0a41800]	l.add r5,r4,r3	[_raw_spin_lock_irqsave+0030]
c03f37d0	003f37d0	[cc022800]	l.swa	[_raw_spin_lock_irqsave+0034]
c03f37d4	003f37d4	[0ffffffd]	l.bnf -3	[_raw_spin_lock_irqsave+0038]
c03f37d8	003f37d8	[15000000]	l.nop 0	[_raw_spin_lock_irqsave+003c]
c03f37dc	003f37dc	[b8640050]	l.srli r3,r4,0x10	[_raw_spin_lock_irqsave+0040]
c03f37e0	003f37e0	[a484ffff]	l.andi r4,r4,0xffff	[_raw_spin_lock_irqsave+0044]
c03f37e4	003f37e4	[e4041800]	l.sfeq r4,r3	[_raw_spin_lock_irqsave+0048]
c03f37e8	003f37e8	[10000006]	l.bf 0x6	[_raw_spin_lock_irqsave+004c]
c03f37ec	003f37ec	[15000000]	l.nop 0	[_raw_spin_lock_irqsave+0050]
c03f37f0	003f37f0	[94820002]	l.lhz r4,0x2(r2)	[_raw_spin_lock_irqsave+0054]
c03f37f4	003f37f4	[e4241800]	l.sfne r4,r3	[_raw_spin_lock_irqsave+0058]
c03f37f8	003f37f8	[13fffffe]	l.bf -2	[_raw_spin_lock_irqsave+005c]

Virtual	Address	Instruction	Disassembly	Symbols
c0003e68	00003e68	[c1001800]	l.mtspr r0,r3,0x4000	[arch_cpu_idle+0030]
c0003e6c	00003e6c	[9c210008]	l.addi r1,r1,0x8	[arch_cpu_idle+0034]
c0003e70	00003e70	[8521fffc]	l.lwz r9,-4(r1)	[arch_cpu_idle+0038]
c0003e74	00003e74	[44004800]	l.jr r9	[arch_cpu_idle+003c]
c0003e78	00003e78	[8421fff8]	l.lwz r1,-8(r1)	[arch_cpu_idle+0040]
c0003e7c	00003e7c	[d7e10ffc]	l.sw -4(r1),r1	[flush_thread+0000]
c0003e80	00003e80	[9c21fffc]	l.addi r1,r1,-4	[flush_thread+0004]
c0003e84	00003e84	[9c210004]	l.addi r1,r1,0x4	[flush_thread+0008]
c0003e88	00003e88	[44004800]	l.jr r9	[flush_thread+000c]
c0003e8c	00003e8c	[8421fffc]	l.lwz r1,-4(r1)	[flush_thread+0010]
c0003e90	00003e90	[d7e117f8]	l.sw -8(r1),r2	[show_regs+0000]
c0003e94	00003e94	[a8430000]	l.ori r2,r3,0	[show_regs+0004]

Attribute	Value
name	system
kind	sc_module
session	44445
vspdebug	0
quantum	5us
duration	0 s
nrcpu	2
mem	0x00000000 0x07ffffff
uart	0x90000000 0x90001fff
ompic	0x98000000 0x98001fff
ethoc	0x92000000 0x92001fff

Attribute	Description
Name	or1kmvp
Host	localhost
Port	44445
User	unknown
Path	/home/jan/Projects/or1kmvp/BUILD/RELEASE/bin/or1
SystemC Version	2.3.1-Accellera
VCML Version	1.0.0-20180824
Time	0.137215000s
Delta Cycle	27469

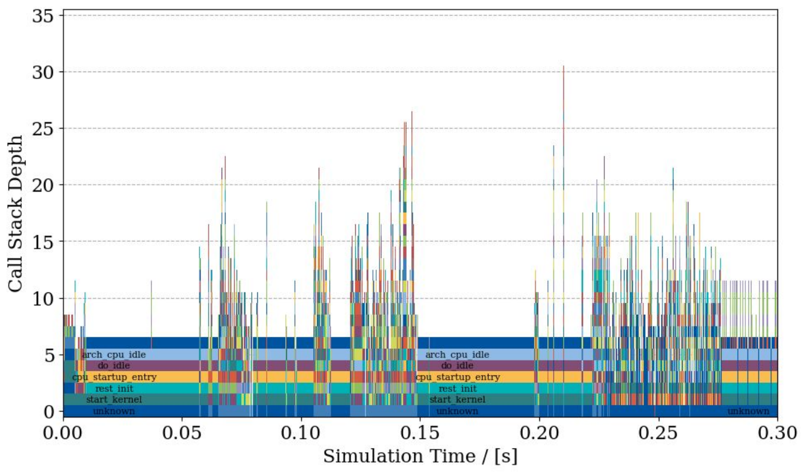
unknown/or1kmvp at localhost:44444 [not connected]
unknown/or1kmvp at localhost:44446 [not connected]
unknown/or1kmvp at localhost:44445 [connected]

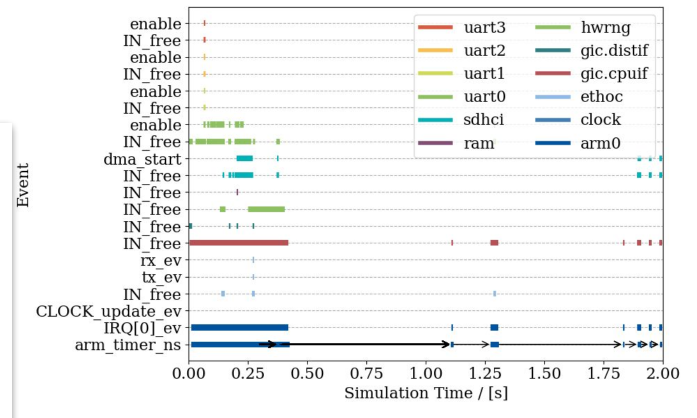
unknown/or1kmvp at localhost:44445

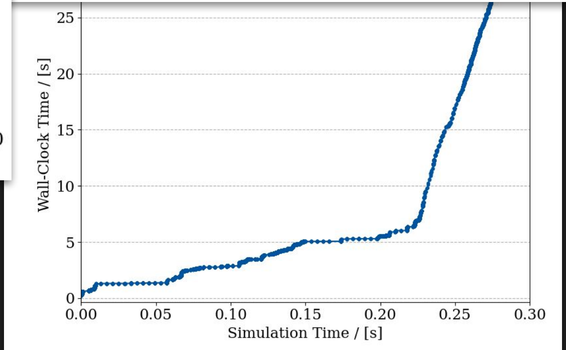
```
> cd system.cpu0
system.cpu0 > dump
Registers:
  PC 0x00000000c03f37c8
  SP 0x00000000c701bd98
  ID 0x0000000000000000
Interrupts:
  IRQ1: 3 events, avg 15us, max 15us
  IRQ2: no events
  IRQ4: no events

system.cpu0 > |
```

unknown/or1kmvp at localhost:44445 | 0.137215000s | 27469 | system

ViPER Trace and Analysis

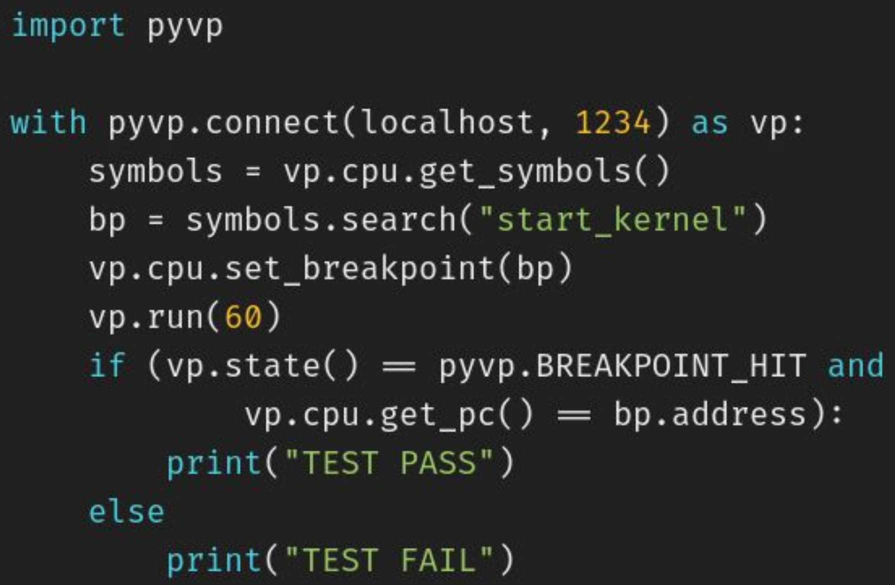






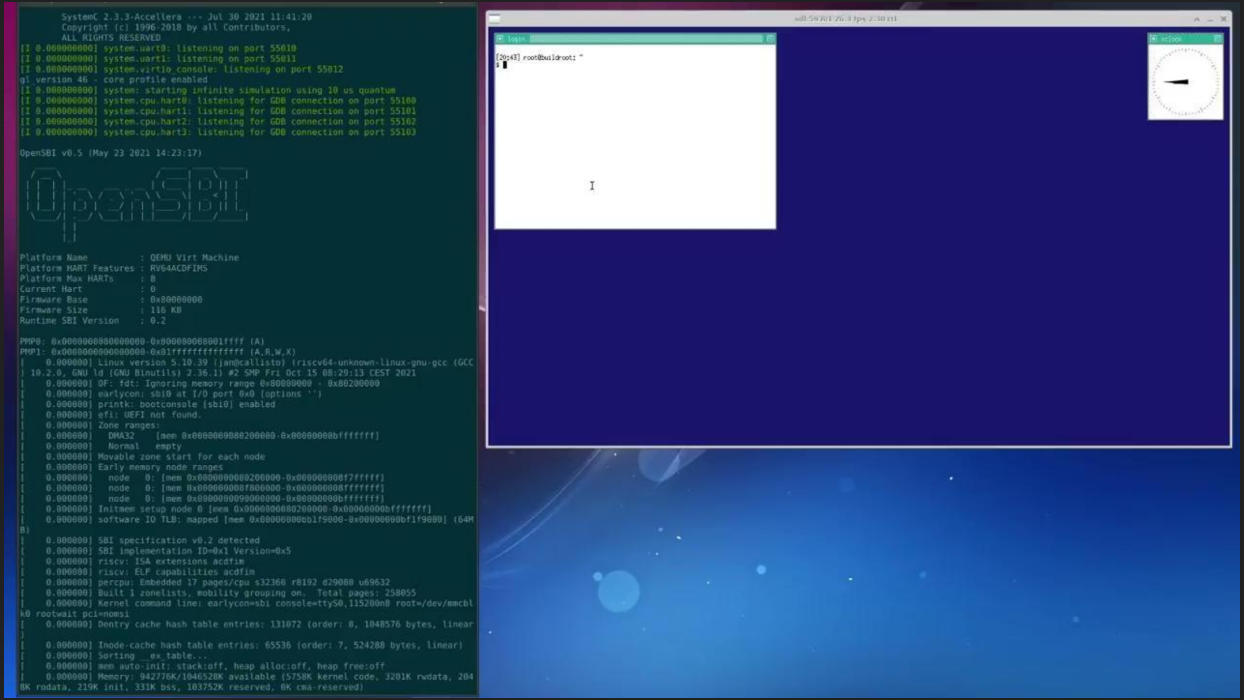
PyVP

- Python scripting for CI and test
- Connect to VP over network
- Control, configure, inspect VP
- **Easy integration** with Jenkins, Gitlab CI, …



```python
import pyvp

with pyvp.connect(localhost, 1234) as vp:
    symbols = vp.cpu.get_symbols()
    bp = symbols.search("start_kernel")
    vp.cpu.set_breakpoint(bp)
    vp.run(60)
    if (vp.state() == pyvp.BREAKPOINT_HIT and
            vp.cpu.get_pc() == bp.address):
        print("TEST PASS")
    else
        print("TEST FAIL")
```

Demo



```
        SystemC 2.3.3-Accellera --- Jul 30 2021 11:41:20
        Copyright (c) 1996-2018 by all Contributors,
        ALL RIGHTS RESERVED
[I 0.000000000] system.uart0: listening on port 55010
[I 0.000000000] system.uart1: listening on port 55011
[I 0.000000000] system.virtio_console: listening on port 55012
gl version 46 - core profile enabled
[I 0.000000000] system: starting infinite simulation using 10 us quantum
[I 0.000000000] system.cpu.hart0: listening for GDB connection on port 55100
[I 0.000000000] system.cpu.hart1: listening for GDB connection on port 55101
[I 0.000000000] system.cpu.hart2: listening for GDB connection on port 55102
[I 0.000000000] system.cpu.hart3: listening for GDB connection on port 55103

OpenSBI v0.5 (May 23 2021 14:23:17)
   ____                    _____ ____ _____
  / __ \                  / ____|  _ \_   _|
 | |  | |_ __   ___ _ __ | (___ | |_) || |
 | |  | | '_ \ / _ \ '_ \ \___ \|  _ < | |
 | |__| | |_) |  __/ | | |____) | |_) || |_
  \____/| .__/ \___|_| |_|_____/|____/_____|
        | |
        |_|

Platform Name          : QEMU Virt Machine
Platform HART Features : RV64ACDFIMS
Platform Max HARTs     : 8
Current Hart           : 0
Firmware Base          : 0x80000000
Firmware Size          : 116 KB
Runtime SBI Version    : 0.2

PMP0: 0x0000000080000000-0x000000008001ffff (A)
PMP1: 0x0000000000000000-0x01ffffffffffffff (A,R,W,X)
[    0.000000] Linux version 5.10.39 (jan@callisto) (riscv64-unknown-linux-gnu-gcc (GCC
) 10.2.0, GNU ld (GNU Binutils) 2.36.1) #2 SMP Fri Oct 15 08:29:13 CEST 2021
[    0.000000] OF: fdt: Ignoring memory range 0x80000000 - 0x80200000
[    0.000000] earlycon: sbi0 at I/O port 0x0 (options '')
[    0.000000] printk: bootconsole [sbi0] enabled
[    0.000000] efi: UEFI not found.
[    0.000000] Zone ranges:
[    0.000000]   DMA32    [mem 0x0000000080200000-0x00000000bfffffff]
[    0.000000]   Normal   empty
[    0.000000] Movable zone start for each node
[    0.000000] Early memory node ranges
[    0.000000]   node   0: [mem 0x0000000080200000-0x000000008f7fffff]
[    0.000000]   node   0: [mem 0x000000008f800000-0x000000008fffffff]
[    0.000000]   node   0: [mem 0x0000000090000000-0x00000000bfffffff]
[    0.000000] Initmem setup node 0 [mem 0x0000000080200000-0x00000000bfffffff]
[    0.000000] software IO TLB: mapped [mem 0x00000000bb1f9000-0x00000000bf1f9000] (64M
B)
[    0.000000] SBI specification v0.2 detected
[    0.000000] SBI implementation ID=0x1 Version=0x5
[    0.000000] riscv: ISA extensions acdfim
[    0.000000] riscv: ELF capabilities acdfim
[    0.000000] percpu: Embedded 17 pages/cpu s32360 r8192 d29080 u69632
[    0.000000] Built 1 zonelists, mobility grouping on.  Total pages: 258055
[    0.000000] Kernel command line: earlycon=sbi console=ttyS0,115200n8 root=/dev/mmcbl
k0 rootwait pci=nomsi
[    0.000000] Dentry cache hash table entries: 131072 (order: 8, 1048576 bytes, linear
)
[    0.000000] Inode-cache hash table entries: 65536 (order: 7, 524288 bytes, linear)
[    0.000000] Sorting __ex_table...
[    0.000000] mem auto-init: stack:off, heap alloc:off, heap free:off
[    0.000000] Memory: 942776K/1046528K available (5758K kernel code, 3201K rwdata, 204
8K rodata, 219K init, 331K bss, 103752K reserved, 0K cma-reserved)
```

```
[20:43] root@buildroot: ~
$ 
```

SIM-V Use Cases

- Early software development and test
    - Utilize in VP or stand-alone to target bare-metal, hypervisor, OS, or user-space
- Reference model for comparison
    - Execute in lock-step and compare states
- Target software analysis
    - Code coverage, hot spot analysis, …
- Continuous Integration
    - Automate tests with SIM-V in Jenkins, Gitlab, …
    - Scale easily on-premise or in the cloud

MachineWare

SIM-V is an ultra-fast RISC-V Simulator (2x faster than Qemu)

MachineWare FTL JIT engine for customizable processor modeling

MachineWare VCML SystemC-TLM 2.0 modeling infrastructure

Reach out:

E-Mail: contact@mwa.re

www.machineware.de