UNIVERSITY OF CAMBRIDGE
Computer Laboratory

FreeBSD/RISC-V updates

Ruslan Bukin

University of Cambridge Computer Laboratory

BSDCam ☺ 2017

Approved for public release; distribution is unlimited. This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL), under contract FA8750-10-C-0237. The views, opinions, and/or findings contained in this article/presentation are those of the author(s)/presenter(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

RISC-V: general information

- Designed in Berkeley
- Fifth implementation of RISC
  - RISC-I, RISC-II from Berkeley
  - Berkeley research machines SOAR/SPUR are considered RISC-III and RISC-IV
- Fully open source, royalty free
- Designed for Research & Education
  - Strong industry support
- Extensible ISA
- Privilege and User specs available
- Growing ecosystem

FreeBSD/RISC-V: General information

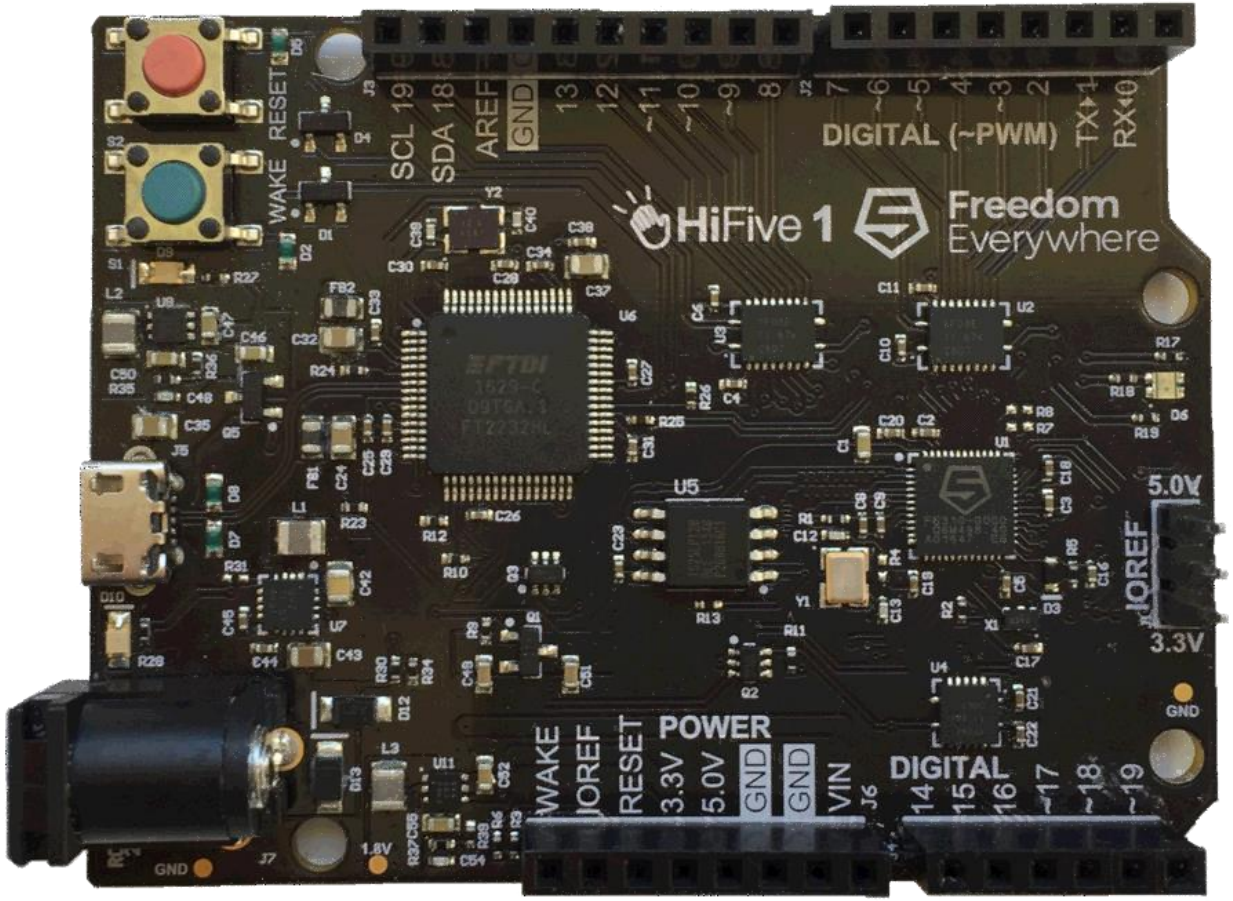
- Started summer 2015
- First chars printed in console Aug 2015
- Fully works December 2015
- Committed January 2016
- Jan 2016 – Aug 2016 : fixing bugs
  - SMP, KDB, DTrace, Modules, ISA v1.9, BBL support
- Late 2016: floating point unit support
- August 2017: GENERIC kernel

FreeBSD/RISC-V

- Full stack BSD license: RISC-V / LLVM / FreeBSD
  - Perfect for research
  - Technology transition

SiFive HiFive1

- TSMC 180nm
- 320+ MHZ
- 16kb SRAM
- 16kb I-cache
- Designed in Chisel
- RTL is open source
- Board available $59
- Chips are not available ☹



Highlights

- HiFive1 board released
- GCC 7 target upstreamed
- NVIDIA is to ship all of their GPUs with RISC-V processor
- Next RISC-V workshop November 2017, organized by Western Digital in San Jose
- v1.10 of privilege specification

Privileged Architecture v1.10: changes

- Not compatible on S-mode with v1.9
- Next versions “should” be compatible with v1.10 on S-mode
- Changes:
  - Built-in macros and compiler arguments changed
  - SBI interface changed
  - VM changes, BBL changed
  - Physical Memory Protection Unit introduced
  - Support for FDT

RISC-V privilege levels

- ❑Machine (M-mode) - Berkeley BootLoader (BBL)
- ❑Hypervisor (H-mode) - September 2017
- ❑Supervisor (S-mode) – OS kernel
- ❑User (U-mode) - /sbin/init

- Supported combination of modes:
  - M simple embedded
  - M, U simple embedded with protection
  - M, S, U UNIX like OS (FreeBSD, Linux)

Berkeley Boot Loader

- Firmware/bootloader
- Operates in M-mode
- Provides access to hardware
- Does not build initial pages tables
- All traps switching mode to M
  - However we can delegate some traps directly to S-mode using mtdeleg

Hypervisor Spec

- Fourth privilege level H above S
  - M, H, S, U
- Designed for Type-1 hypervisors
- Work in progress for Type-2 hypervisors (Bhyve, KVM)
- September 2017

Compiler arguments changed

-mno-float, -msoft-float removed

-march=rv64imafdc -mabi=lp64

“a” – atomic
“m” – multiplication
“fd” – double precision floating point unit
“c” – compressed
“g” == “imafd”

Compiler built-in defines

- __riscv__, __riscv64 removed!

__riscv
__riscv_compressed
__riscv_atomic
__riscv_mul
__riscv_div
__riscv_muldiv
__riscv_fdiv
__riscv_fsqrt

__riscv_float_abi_soft
__riscv_float_abi_single
__riscv_float_abi_double
__riscv_cmodel_medlow
__riscv_cmodel_medany
__riscv_cmodel_pic
__riscv_xlen == 64

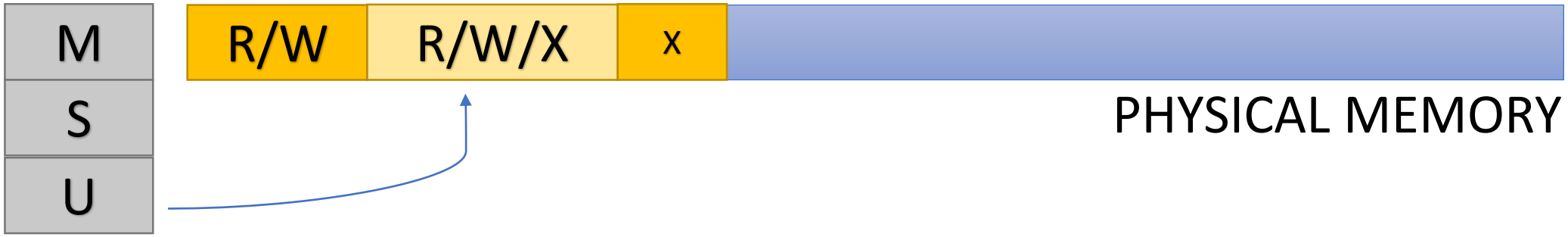
SBI interface changed

- Old way: pre-defined function address in physical memory
- New way: ecall to upper (machine) privileged level
- Required for:
  - Timer
  - Console
  - IPI
  - Shutdown
  - Hart ID

PMP unit (1 of 2)

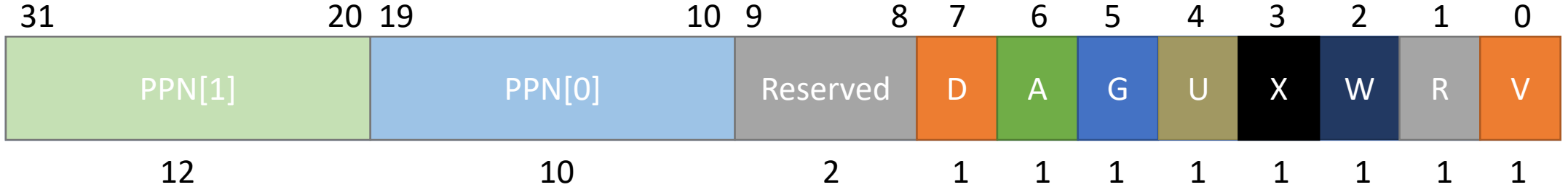
- When enabled, modes below M have no memory permissions
- Can grant R/W/X on 4-bytes granularity
- Up to 16 memory regions

PMP unit (2 of 2)



- Composable with MMU: address translation happens first, then PMP checks translated address
- Can be locked in M-mode
- Useful when modes below M is untrusted

Virtual Memory changes (1 of 2)



- Supervisor can’t access user pages by default
- R, W, X controlled separately
  - Support for X-only pages
  - Combination W & ~R reserved

VM changes (2 of 2)

- Supervisor can’t access user pages by default
  - We need to access (e.g. on syscall)
  - Solution: set SUM bit to SSTATUS register
- Supervisor can’t read eXecute-only pages
  - Do we care ?
  - But solution: set MXR bit in SSTATUS register
- VM on/off controlled by SATP (sptbr) register by Supervisor
- BBL does not build initial page tables for us anymore

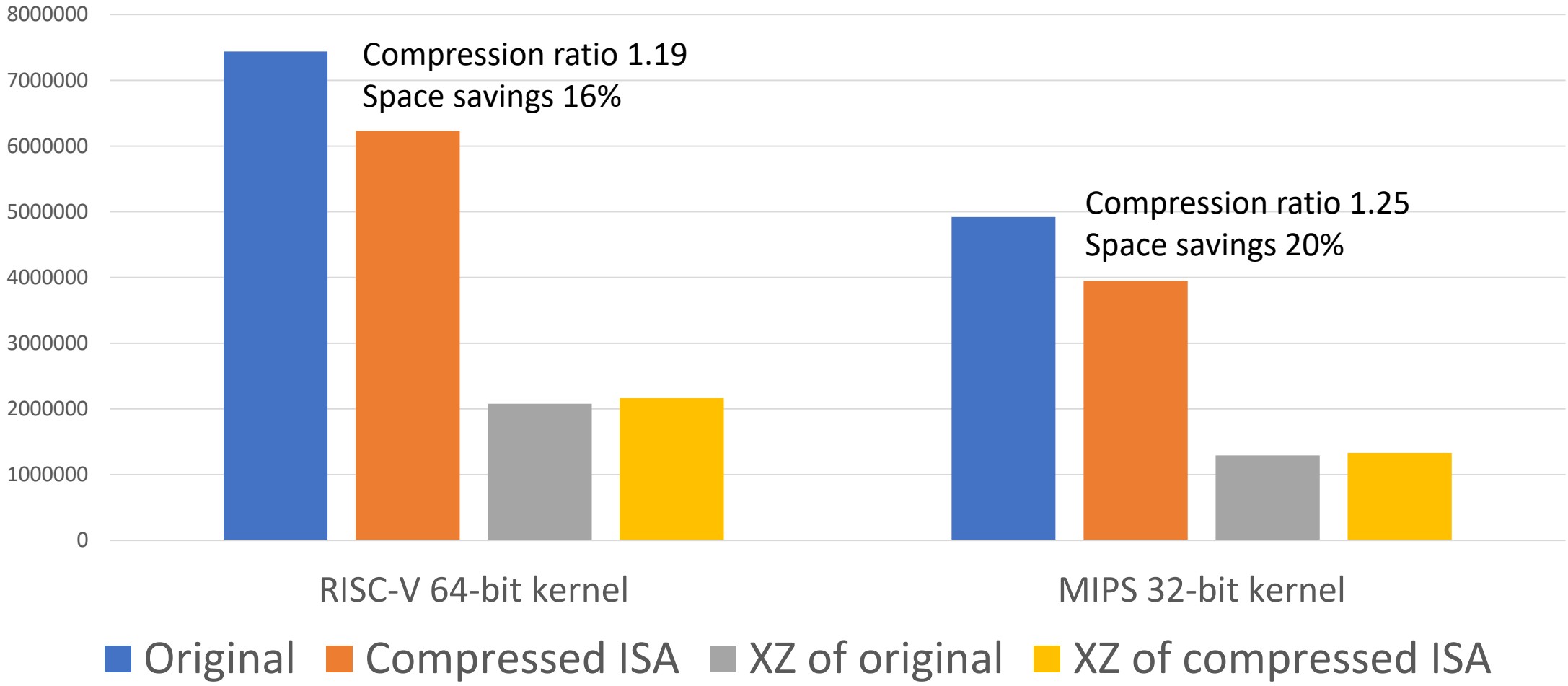
Switch VM on

1. Build page tables
2. Set exception vector register (STVEC) to some function in virtual address space
3. Set pointer to page tables and enable MMU in sptbr
4. Wait few CPU cycles
5. Here we are. Restore exception vector (STVEC) to real exception handler

“C”- Compressed Extension

- Low-end deeply embedded to save space
- High-end workload to save cache footprint
- Compressing some instructions to 2 bytes
- Declared 25-30% smaller code with extension turned on

“C”- Compressed Extension (continued)



Compressed extension assembly

```
/home/br/obj/riscv.riscv64/usr/home/br/dev/freebsd-riscv/sys/GENERIC/kernel:     file format e


Disassembly of section .text:

ffffffc000200000 <_start>:
ffffffc000200000:       000402b7                lui     t0,0x40
ffffffc000200004:       1002a073                csrs    sstatus,t0
ffffffc000200008:       8d2a                    mv      s10,a0
ffffffc00020000a:       8dae                    mv      s11,a1
ffffffc00020000c:       000d0463                beqz    s10,ffffffc000200014 <_start+0x14>
ffffffc000200010:       1b00406f                j       ffffffc0002041c0 <mpentry>
ffffffc000200014:       00461497                auipc   s1,0x461
ffffffc000200018:       fec48493                addi    s1,s1,-20 # ffffffc000661000 <pagetabl
ffffffc00020001c:       00462917                auipc   s2,0x462
ffffffc000200020:       fe490913                addi    s2,s2,-28 # ffffffc000662000 <pagetabl
ffffffc000200024:       00c95913                srli    s2,s2,0xc
ffffffc000200028:       fff0079b                addiw   a5,zero,-1
ffffffc00020002c:       02679793                slli    a5,a5,0x26
ffffffc000200030:       83f9                    srli    a5,a5,0x1e
ffffffc000200032:       1ff7f793                andi    a5,a5,511
ffffffc000200036:       4e85                    li      t4,1
ffffffc000200038:       00a91f13                slli    t5,s2,0xa
ffffffc00020003c:       01eeefb3                or      t6,t4,t5
ffffffc000200040:       4821                    li      a6,8
ffffffc000200042:       030787bb                mulw    a5,a5,a6
ffffffc000200046:       00f482b3                add     t0,s1,a5
ffffffc00020004a:       01f2b023                sd      t6,0(t0) # 40000 <kernbase-0xffffffbf
ffffffc00020004e:       00462497                auipc   s1,0x462
ffffffc000200052:       fb248493                addi    s1,s1,-78 # ffffffc000662000 <pagetabl
:
```

FDT support

- GENERIC kernel only
- timer/console moved from OFW bus to nexus bus

./spike -dump-dts

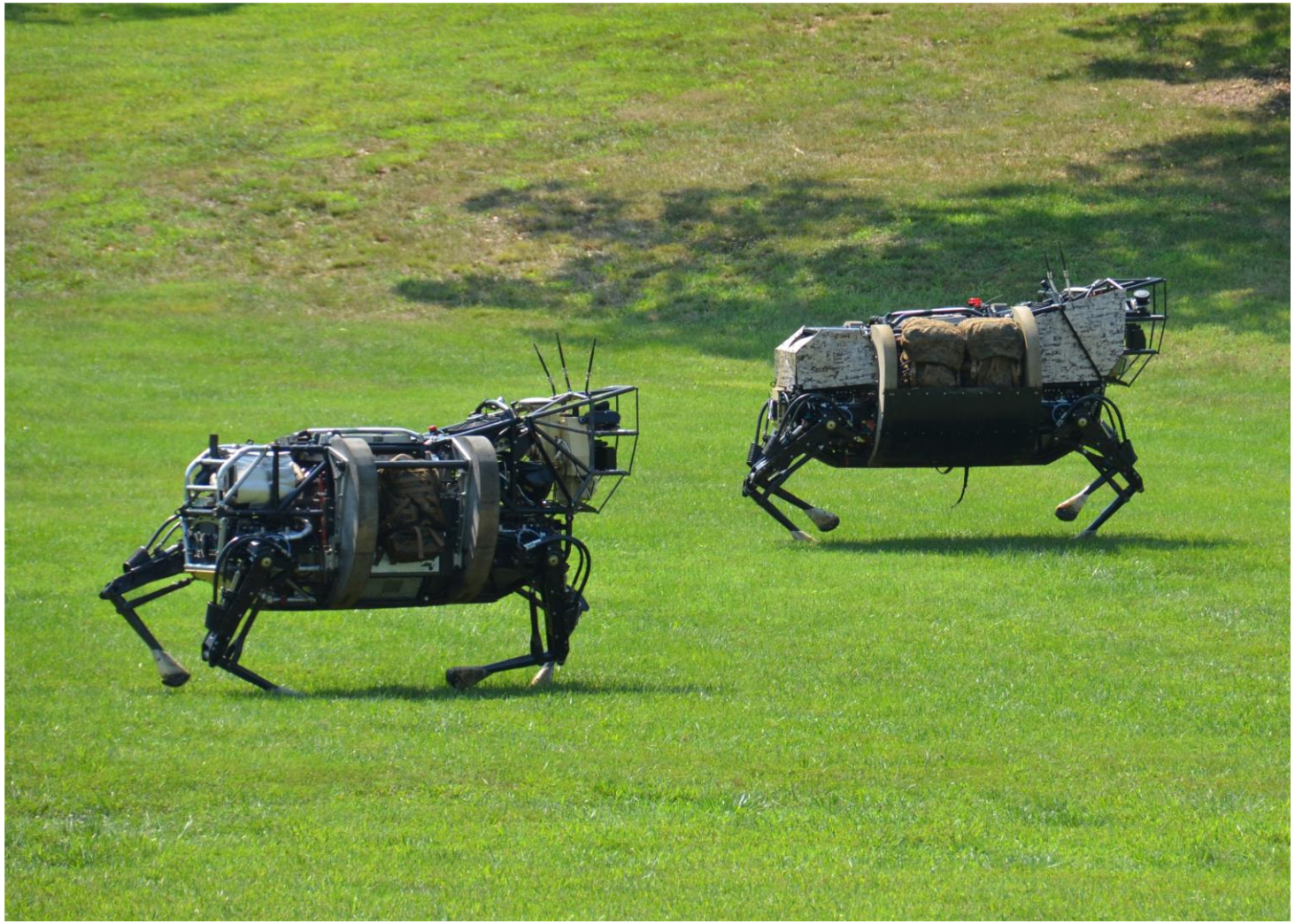
./spike -m2048 -p8 /path/to/bbl

Hardware support for v1.10 privilege spec

- No real hardware available for us yet
  - due to changeable spec?
- Spike
  - golden model simulator
- Rocket Chip
  - Not tried yet
- lowRISC, QEMU
  - Status is unknown

Plans

- Real chips?
- LLVM support?
- GCC freebsd patches upstream
- QEMU usermode?
- BHYVE/RISC-V ?
- CHERI/RISC-V ?



Review

D11800 (only 65 files changed)

Questions ?