
ORACLE®

sparcv9

New archicture for Valgrind

Ivo Raisr
Tomáš Jedlička

ORACLE

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

THE FOLLOWING IS INTENDED TO OUTLINE OUR GENERAL PRODUCT DIRECTION. IT IS INTENDED FOR INFORMATION PURPOSES ONLY, AND MAY NOT BE INCORPORATED INTO ANY CONTRACT. IT IS NOT A COMMITMENT TO DELIVER ANY MATERIAL, CODE, OR FUNCTIONALITY, AND SHOULD NOT BE RELIED UPON IN MAKING PURCHASING DECISIONS. THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITY DESCRIBED FOR ORACLE'S PRODUCTS REMAINS AT THE SOLE DISCRETION OF ORACLE.

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

Porting Valgrind to sparcv9

sparcv9/Linux and sparcv9/Solaris

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

Overview

- The story
- Hackers
- Why SPARC?
- sparcv9 ISA highlights
- Design decisions and problems to solve
- Current status

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

The Story

- Gained traction after upstreaming Solaris port
- Initially sparcv9/Solaris
- Recently sparcv9/Linux
- Maintained as separate forks

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

Hackers

- Collaborative effort:
- Ivo Raisr
- Tomáš Jedlička

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

Why SPARC?

- SPARC
- History begins in 1980’s
- Joint effort of Sun (now Oracle) and Fujitsu
- Designed for enterprise workloads
- Several terabytes of memory
- Several thousands of CPUs

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

sparcv9 ISA Highlights I.

- RISC architecture
- load/store memory access, properly aligned
- Explicit stack support (nothing like push/pop)
- Instruction format is `OP %r1, %r2/imm, %rd`
- Hypervisor in the firmware

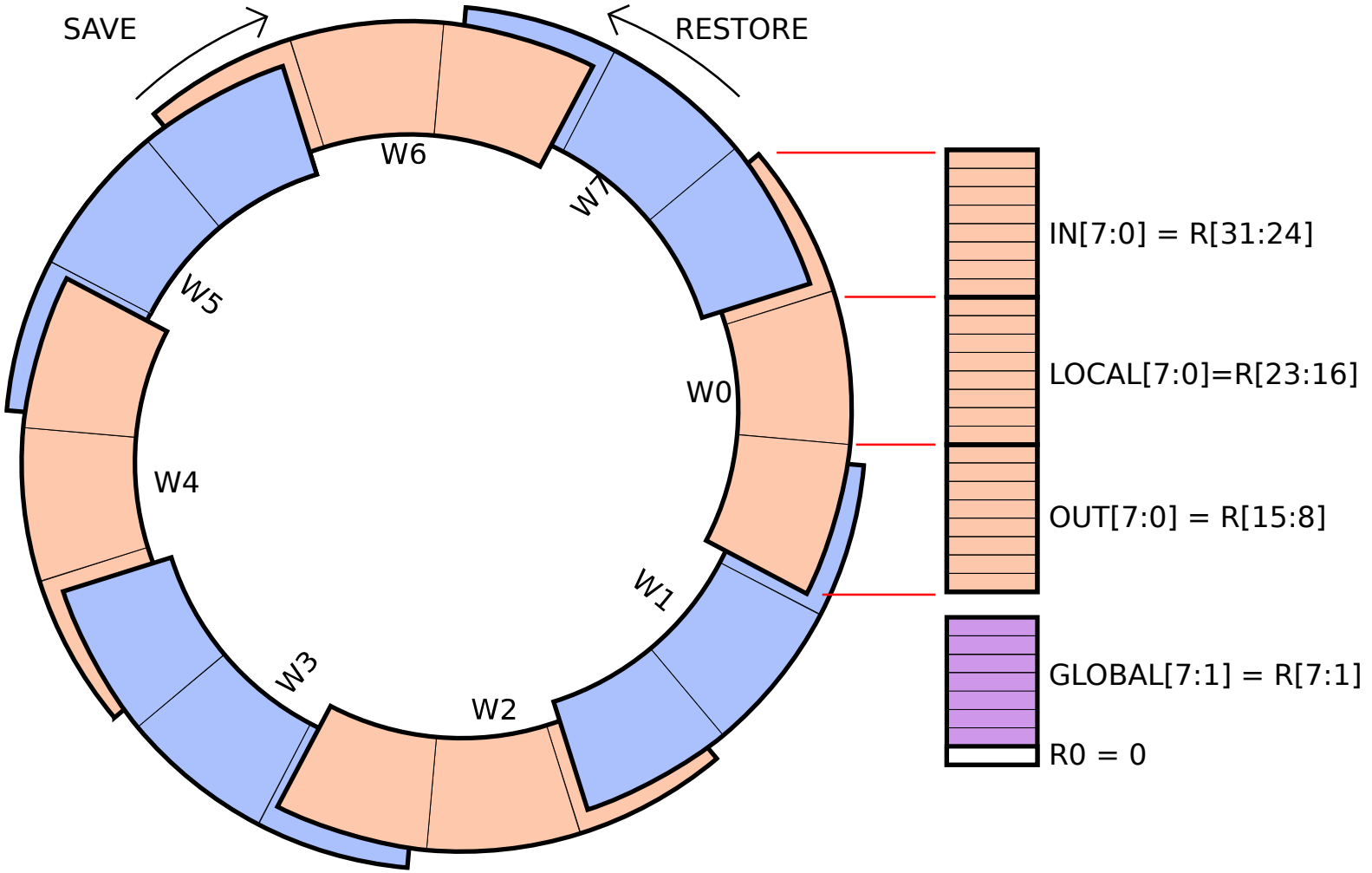
Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

sparcv9 ISA Highlights II.

- **Registers**
- General purpose registers and aliasing into `%i, %o, %l`
- Widowing state register + spill/fill traps
- Floating point registers and its variants `%f, %d, %q` (4, 8, 16 bytes)

ORACLE

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |



Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

sparcv9 ISA Highlights III.

- **Delayed Control Transfer Instructions** (dCTI)
- Control transfer instructions (jmp, branch...) have a delay slot
- Sequence of instructions:
  - `bcc %xcc/%icc, <address>`
  - `add %o0, 1, %o0`
- behaves actually as:
  - Evaluate the branch condition
  - Execute `add %o0, 1, %o0`
  - Set nPC to <address>
  - Transfer control to <address>, setting PC to <address> and nPC to <address+4>
- Possibility to annul the delayed instruction if the branch is not taken.

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

sparcv9 ISA Highlights IV.

- **Address Space Identifier** (ASI)
- Alters MMU behaviour during load/store operations
- One instruction can do many different things, based on *ASI* value
- Example: LDDF loads 8 bytes by default, however with a different *ASI*:
  - *LDSHORTF* loads 1 byte with *ASI* D0
  - *LDSHORTF* loads 2 bytes with *ASI* D2
  - *LDBLOCKF* loads 64 bytes with *ASI* F0
  - *LDDFA* loads 8 bytes with other *ASI*s
- *ASI* can be immediate or register (%asi)
- Sharing the same opcode: `ldda [address] %asi, register`
- Actual instruction is known only at runtime, not at VEX translation time
- Used in: hardware specialized optimizations, OpenSSL, stack unwinding

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

sparcv9 ISA Highlights V.

- **Application Data Integrity** (ADI)
- Dynamic Tainting technique implemented in sparc chip MMU and memory allocators.
- Taint mark (version) stored both in memory/cache and in the pointer passed from the allocator.
- Taint mark (version) match is checked by MMU with every load or store.
- Trap generated and signal sent to the process when versions do not match.

ORACLE

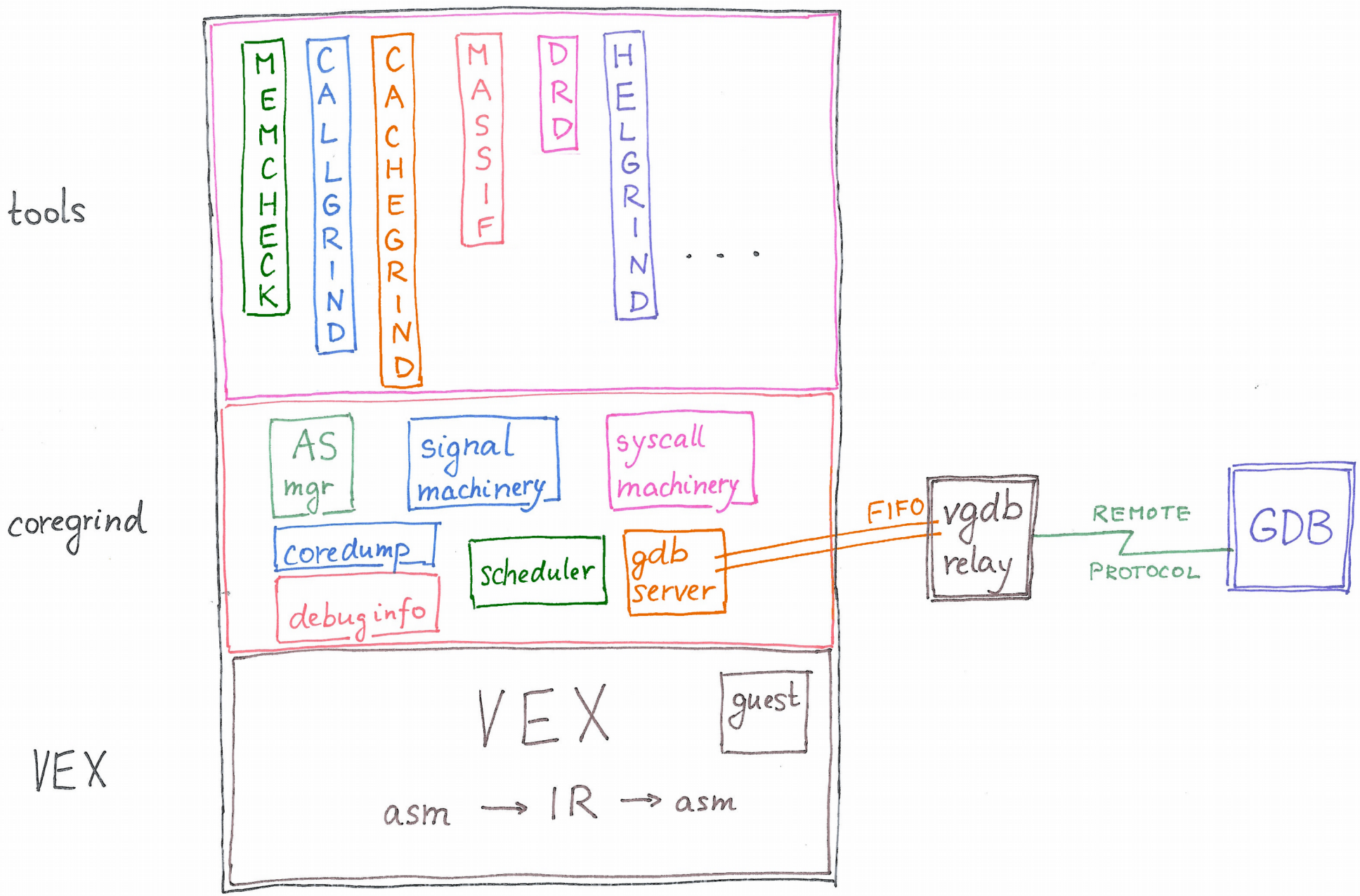
Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

sparcv9 ISA Highlights VI.

- **Syscalls and Fast traps**
- Syscall handling via trap table
- Fast trap based syscall
- OS specific feature

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

VALGRIND ARCHITECTURE



sparcv9 guest state layout

EvC_FAILADDR	EvC_COUNTER	%g0	%g1	%g2	%g3
%g4	%g5	%g6	%g7	%o0	%o1
%o2	%o3	%o4	%o5	%o6	%o7
%l0	%l1	%l2	%l3	%l4	%l5
%l6	%l7	%i0	%i1	%i2	%i3
%i4	%i5	%i6	%i7	%f0	%f2
%f4	%f6	%f8	%f10	%f12	%f14
%f16	%18	%f20	%f22	%f24	%f26
%f28	%f30	%d32	%d34	%d36	%d38
%d40	%d42	%d44	%d46	%d48	%d50
%d52	%d54	%d56	%d58	%d60	%d62
%pc	%npc	%y	%asi	%fprs	%gsr
CMSTART	CMLEN	CC_OP	CC_DEP1	CC_DEP2	CC_NDEP
FSR_RD	FSR_FCC	FSR_CEXC_OP	FSR_CEXC_DEP1_HI	FSR_CEXC_DEP1_LO	FSR_CEXC_DEP2_HI
FSR_CEXC_DEP2_LO	FSR_CEXC_NDEP	EMNOTE	scratchpad		

ORACLE

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

Design Decisions and Problems I.

- Register windows support
- Instructions in branch delay slot
- Lazy evaluation of CCR and FSR.cexc
- Instructions with %asi register in opcode
- Basic emulation of 128 bit integer operations
- Program counters PC and nPC
- Hundreds of misaligned warnings from compiler
- Represent a syscall returning 5 return values
- Can Valgrind leverage ADI?

ORACLE

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

Design Decisions and Problems II.

- Decorating `Iex_Load` and `Iex_Store` with *ASI*
- Teach Memcheck about `ASI_PRIMARY_NOFAULT`

```
struct {
   IREndness end;    /* Endian-ness of the load */
   IRType    ty;     /* Type of the loaded value */
   IRExpr*   addr;   /* Address being loaded from */
   IRExpr*   asi;    /* SPARCv9 address space indetifier */
} Load;
```

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

Current status

- Synced regularly with SVN upstream
- Support for sparcv9/Solaris stable
- memcheck and none tests:
- `== 731 tests, 35 stderr failures, 8 stdout failures, 0 stderrB failures, 0 stdoutB failures, 1 post failure ==`
- Support for sparcv9/Linux at the beginning
- Both ports live at: https://bitbucket.org/iraisr/valgrind-solaris

ORACLE

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |

Q&A Session

Copyright © 2017 Oracle and/or its affiliates. All rights reserved. |


ORACLE®