



OpenSPARC™

Chapter Three

OPENSPARC ARCHITECTURE GENERATIONS

David L. Weaver

Principal Engineer, UltraSPARC Architecture

Principal Evangelist, OpenSPARC

Sun Microsystems



University Roots

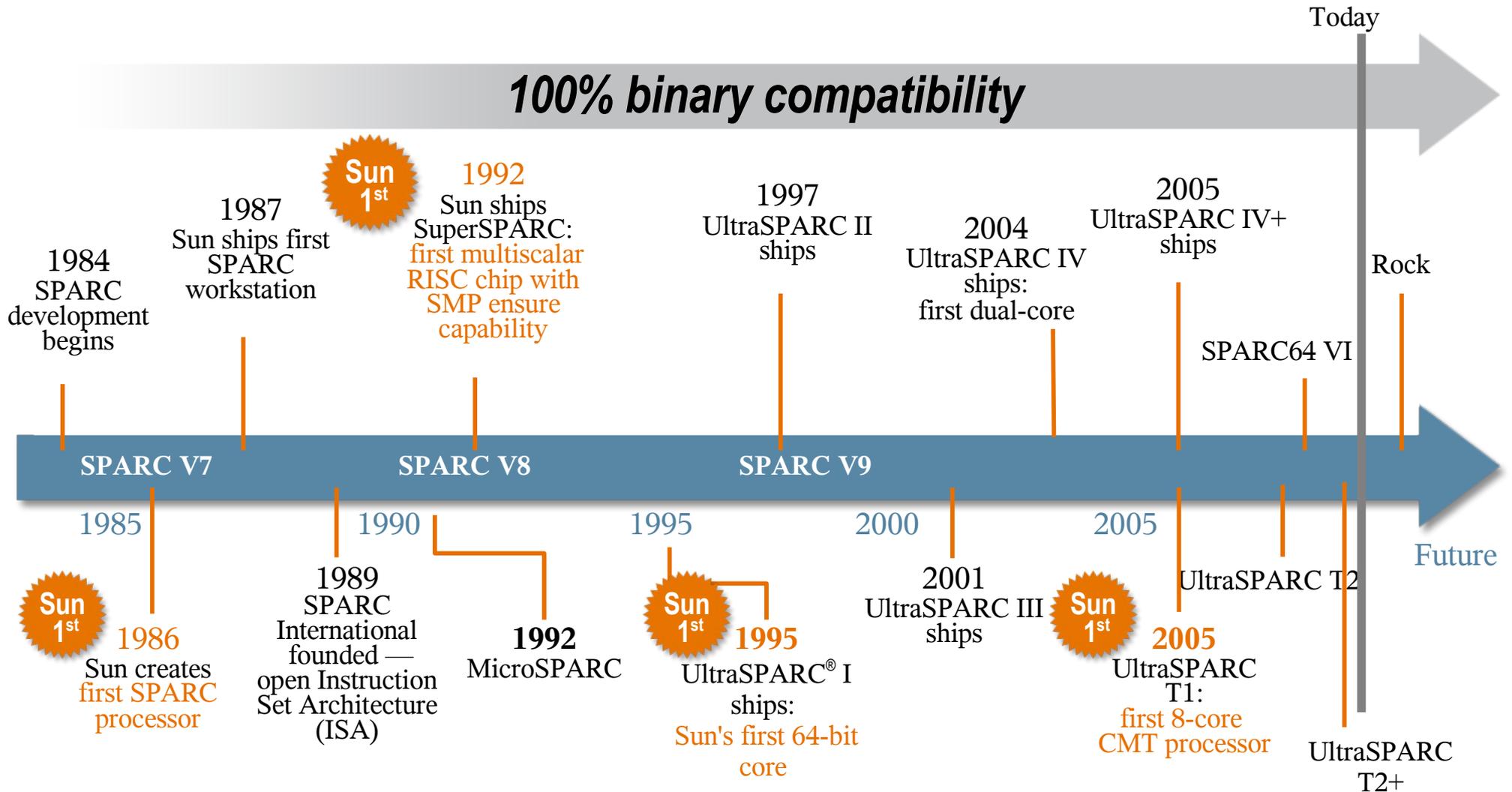
- SUN
 - > Founders three from Stanford (Business and Hardware) and one from Berkeley (Software, BSD-Unix)
 - > **Stanford Unix Network**
- SPARC
 - > **Scalable Processor ARChitecture**
 - > Based on Berkeley's RISC projects
 - > Different than MIPS (Stanford's project) mainly in register windows and MIPS pipeline focus
 - > First successful commercial RISC processor design clearly demonstrated the power of RISC concept and helped ushered in the generation of RISC/modern processor design until now

Generations of SPARC

- **SPARC V8** (SPARC Int'l, 1989): 32-bit
 - > MicroSPARC I and SuperSPARC I, 1992
 - > MicroSPARC II and SuperSPARC II, 1994
- **SPARC V9** (SPARC Int'l, 1994): 64-bit addr+data
 - > UltraSPARC I, 1995 – VIS-1 instructions
 - > UltraSPARC III, ~2000 – VIS-2 instructions
 - > UltraSPARC IV, 2004 – Dual core, basic CMT
- **UltraSPARC Architecture 2005** (Sun,2005): full CMT, hyperprivileged mode
 - > UltraSPARC T1, 2005 → OpenSPARC T1
- **UltraSPARC Architecture 2007** (Sun,2007):
 - > UltraSPARC T2, 2007 → OpenSPARC T2

The History of SPARC®

More than two decades of continuous technical innovation



Specification Differences, V9 → UA 2005

- Formatting improvements
- More complete and more precise than SPARC V9; for example:
 - > lists the specific conditions under which each exception may be raised, for every instruction
 - > clarifies relative trap priorities
 - > closes many old implementation dependencies
 - > specifies many extensions to architecture
- Document Design:
 - > Architecture Spec + Implementation Supplements

Architecture Extensions, V9 → UA 2005

- Sun's VIS1 and VIS2 instructions
- GSR register
- Privileged register-window management instructions ALLCLEAN, OTHERW, NORMALW, and INVALIDW
- “Deferred” traps split into two categories
 - > **SPARC V9 deferred traps are now "restartable deferred" traps**
 - > **Termination Deferred Traps**

Architecture Changes, V9 → UA 2005

- **Hyperprivileged** mode has been added, including:
 - > several hyperprivileged registers
 - > a few hyperprivileged instructions
 - > notably RDHPR and WRHPR (hyperprivileged register access)
 - > effects on the Tcc instruction
 - > effects on the trap model
 - > SIR instruction is now hyperprivileged
 - > VER register is now the hyperprivileged (**HVER**)
 - > full control of Chip MultiThreading (CMT) features

Architecture Changes, Earlier UltraSPARCs → UA 2005

- For Block Store instructions, an intermediate "zero" state is allowed to be observed during execution

Feature Classification in UA 2005

- Architectural features are now classified and tagged
 - > Software Class (letter)
 - > Implementation Class (digit)
 - > allows smooth long-term architectural evolution (addition and deprecation of features)

Why Hyperprivileged Mode?

- Allows running multiple simultaneous guest OSs
 - > (and/or multiple versions of the same OS)
- Allows running older OS (that uses hypervisor API) on newer hardware, without need to port the OS
- Simplifies OS ports (Linux in 2 months!)
- Allows implementation of logical domains
- Allows *virtualization*

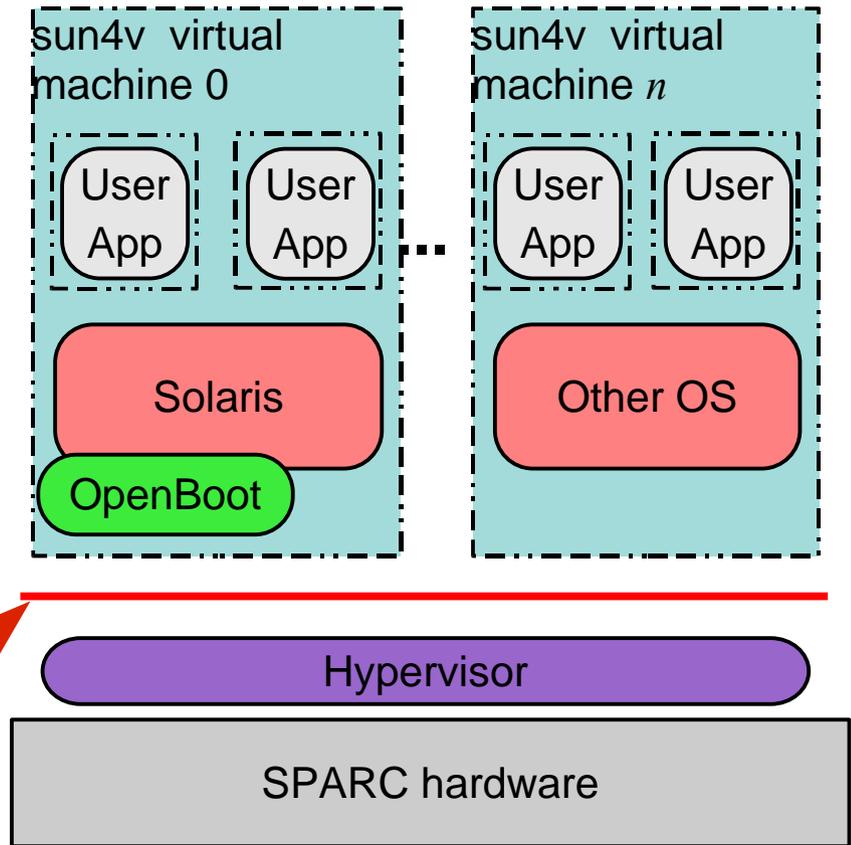
Why Virtualization?

- Insulates higher levels of software from underlying hardware, by adding another software abstraction layer
 - > Protects customers' investment in application software from changes in underlying software (OS)
 - > Buying new, faster HW no longer requires running a new version of the OS
- Allows ability to "oversubscribe" resources (run multiple top-level software)

Virtualization

- Thin software layer between OS and platform hardware
- Para-virtualized OS
- Hypervisor + sun4v interface
 - Virtualizes machine HW and isolates OS from register-level
 - Delivered with *platform*, not with OS
 - Not itself an OS

stable interface "sun4v"





OpenSPARC™

OpenSPARC Slide-Cast

In 12 Chapters

Presented by OpenSPARC designers,
developers, and programmers

- to guide users as they develop their own OpenSPARC designs and
- to assist professors as they teach the next generation

This material is made available under
Creative Commons Attribution-Share 3.0 United States License

[Creative Commons Attribution-Share 3.0 United States License](https://creativecommons.org/licenses/by-sa/3.0/)

