



Architecture of the UltraSPARC processors

Jakub Jermar

Solaris Revenue Product Engineering

Sun Microsystems Czech



History of SPARC

- SPARC = Scalable Processor ARChitecture
- Processor architecture vs. processor implementation
- Designed by Sun in 1985, based on RISC I & II
 - > RISC project vs. MIPS project
- SPARC International
- SPARC V7 (1986), SPARC V8 (1990) – 32-bit
- SPARC V9 (1993) – 64-bit

Breakdown of SPARC V9

- SPARC V9 is architecture specification
 - > Pre-JPS1 processors:
 - UltraSPARC I, UltraSPARC II[ei]
 - > JPS1 (Joint Programming Specification 1)
 - UltraSPARC III[i, Cu], UltraSPARC IV[+]
 - SPARC64 V
 - > UltraSPARC Architecture 2005 (2007)
 - UltraSPARC T1, UltraSPARC T2
- } sun4u
 sun4us
 sun4v

Features of SPARC

- RISC design
- big endian (can access memory as little endian)
- 3-operand A/L instructions
- delay slots, anul bit
- branch prediction, hints
- conditional execution of instructions (ia64, arm32)
- out-of-order execution of instructions
- VIS instructions: SIMD

Alternate global registers

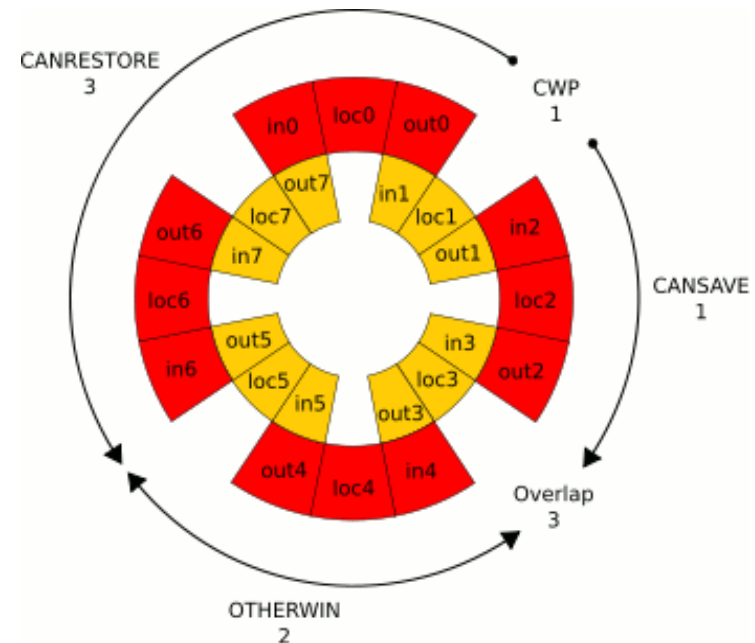
- $r[0] - r[7]$ or $g0 - g7$
- SPARC V9: normal and alternate globals (2 sets)
 - > Unprivileged software sees only normal globals
 - > Faster trap/interrupt handlers
 - > Kernel can store kernel stack address in an alternate register
 - > Similar feature found on IA-64, ARM
- UltraSPARC I – IV+: added IG and MG (4 sets)
- UltraSPARC T1, T2 ($MAXGL + 1$ sets)

Register windows I

- $r[8] - r[31]$ or $o0 - o7$ plus $l0 - l7$ plus $i0 - i7$
- Flat mode
 - > 32 general purpose registers $r[0] - r[31]$
 - > Function prologues use `SUB` instead of `SAVE`
 - > Function epilogues use `ADD` instead of `RESTORE`
 - > “MIPS-like behaviour”
 - > gcc 4.0.2: `-mflat`
 - > Not used

Register windows II

- $r[8] - r[31]$ or $o0 - o7$ plus $l0 - l7$ plus $i0 - i7$
- Windowed mode
 - > NWINDOW register windows (8 sets on UltraSPARC)
 - > IA-64 has a similar feature: RSE (Register Stack Engine)
 - > Window traps and OS support



Traps I

- Trap table contains 1024 entries with code
 - > Each trap type gets one entry
- Some traps that don't lead to a context switch don't have to save any state
 - > Alternate globals (different traps switch to different sets)
 - > Hardware stack for nested traps (`MAXTL = 5` on UltraSPARC) -> trap levels
- Handling traps that might do a context switch is very complicated thanks to the windowing mechanism and trap levels

Traps II

- Trap table
 - > 512 entries for traps from TL 0 + 512 entries for traps from TL above 0.
- 8 instructions per one trap type
- Frequent traps span multiple trap types (4) so that their handlers can more easily fit in the trap table.

ASI's

- Address Space Identifiers
 - > Not to be confused with 13-bit memory context ID
- ASI determines a namespace for addressing
- The default ASI is physical memory so that loads and stores work as expected
- Example: ASI 0x58 - 0x5f DMMU registers
 - > TLBs are programmed via using ASIs
- Example: Accessing secondary context (e.g. userspace with user privileges) via ASI 0x11
 - > kernel can have the whole 64-bit address space for itself

MMU

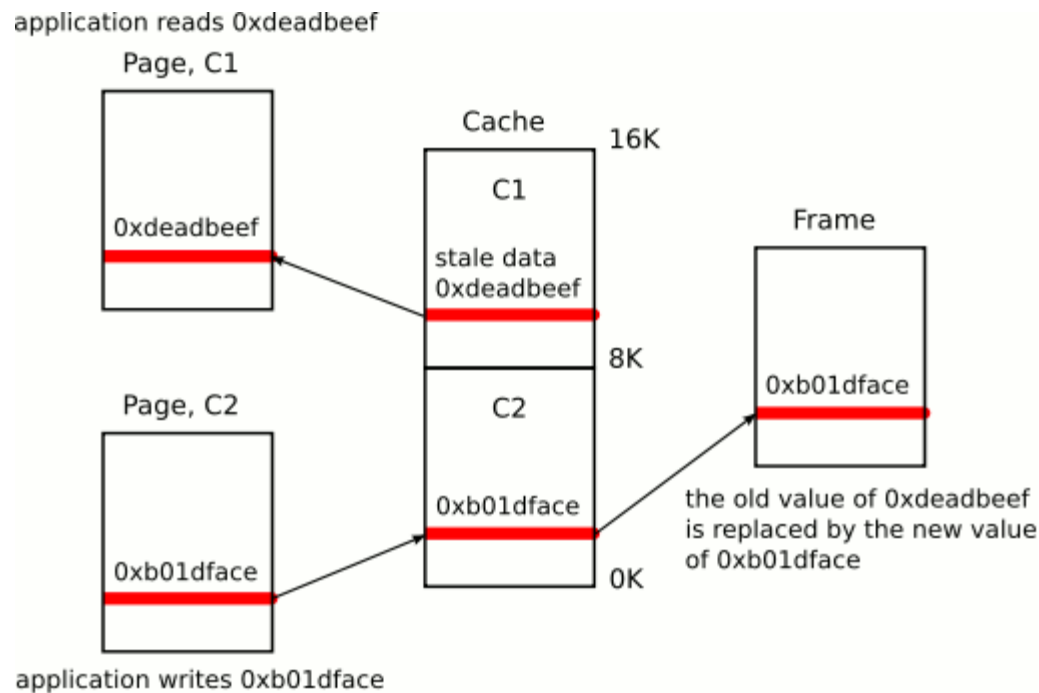
- Implementation dependent
- UltraSPARC I Spitfire MMU
- Separate I-TLB and D-TLB, each has 64-entries
- No hardware support for hierarchical page tables
- TLB-only with hardware assisted TSB index calculation
- TSB: software-managed cache of TLB entries
- Multiple page sizes: 8K, 64K, 512K and 4M
- 13-bit memory contexts

Caches I

- Implementation dependent
- Pre-T1 UltraSPARCs have 16K virtually indexed and physically tagged write-through direct-mapped instruction and data caches

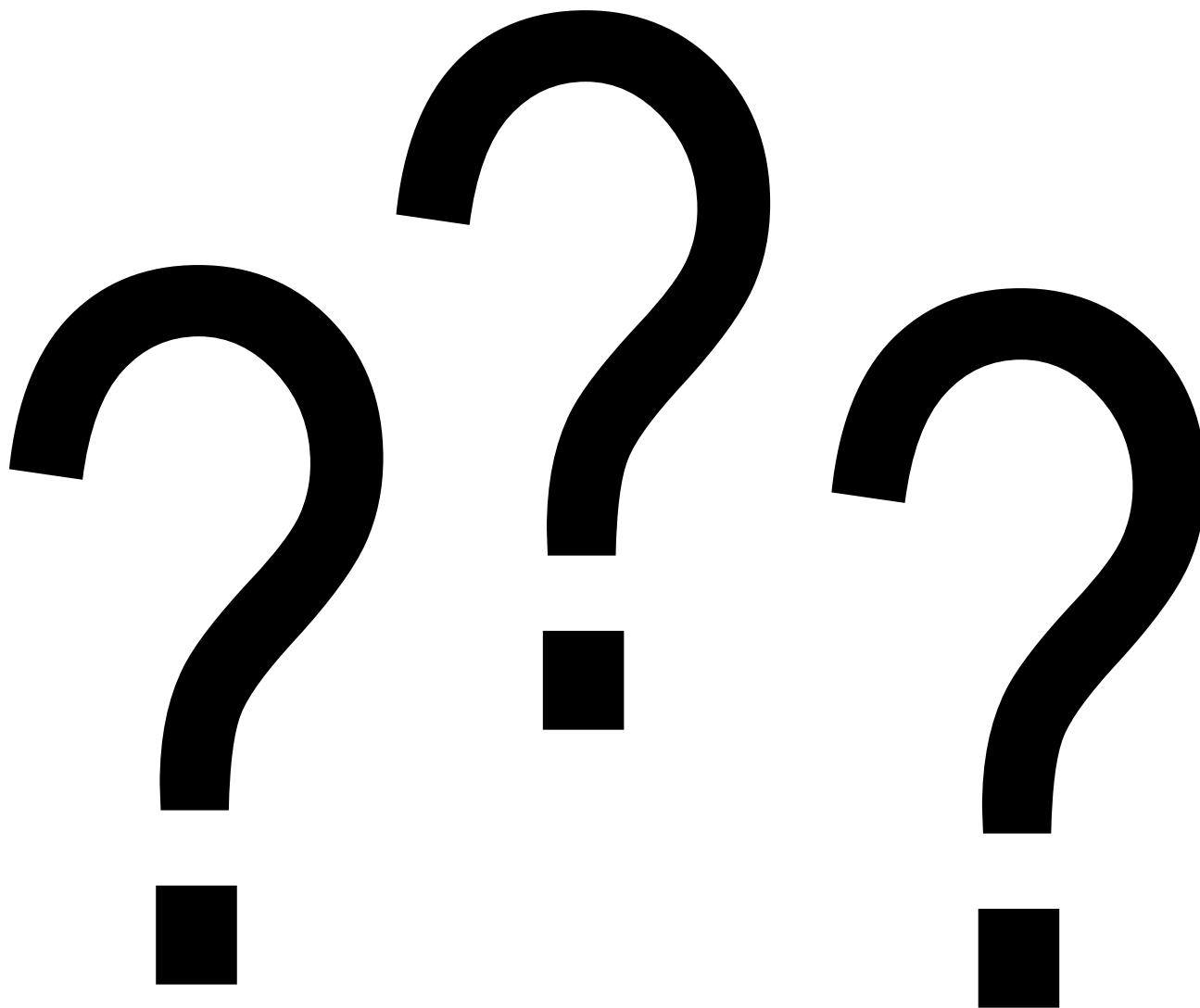
Caches II

- No hardware support for illegal virtual alias detection
 - > Must be dealt with in the OS



References

- <http://www.sun.com/processors/documentation.html>
- <http://www.sparc.org>
- <http://www.opensparc.net>
- Google for JPS1, first two links
- Google for ultrasparc architecture 200
- Porting SPARTAN kernel to SPARC V9 architecture





Architecture of the UltraSPARC processors

Jakub Jermar

Solaris Revenue Product Engineering

Sun Microsystem Czech