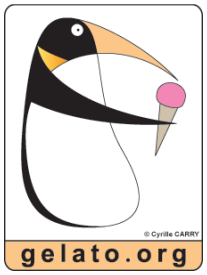
Slide 1

UltraSPARC T1

The processor formerly known as "Niagara"



This work supported by UNSW and HP through the Gelato Federation

Slide 2

SPARC History

- ➜ **S**calable **P**rocessor **ARCH**itecture
- ➜ 1985 – Sun Microsystems
  - ➜ Berkeley RISC – 1980-1984
  - ➜ MIPS – 1981-1984

Architecture v Implementation:

- ➜ SPARC Architecture
  - ➜ SPARC V7 – 1986
  - ➜ SPARC Interntaional, Ltd – 1989
  - ➜ SPARC V8 – 1990
  - ➜ **SPARC V9** – 1994
- ➜ Privileged v Non-Privileged

Slide 3

Sun + SPARC = UltraSPARC

Processor	Cores	Threads/Core	Clock	L1D	L1I	L2 Cache
UltraSPARC IIi	1	1	550Mhz, 650Mhz	16KiB	16KiB	512KiB
UltraSPARC IIIi	1	1	1.593Ghz	I	D	1MB[a]
UltraSPARC III	1	1	1.05-1.2GHz	64KiB	32KiB	8MiB[b]
UltraSPARC IV	2[c]	1	1.05-1.35Ghz	64KiB	32KiB	16MiB[d]
UltraSPARC IV+	1	2	1.5Ghz	I	D	2MiB[e]
UltraSPARC T1	8	4	1.2Ghz	32KiB	16KiB[f]	3MiB[g]
UltraSPARC T2[h]	16 (?)	8	2Ghz+ (?)	?	?	?

[a] On-chip
[b] External, on chip tags
[c] UltraSPARC III cores
[d] 8MiB per core
[e] 32MiB off chip L3
[f] I/D Cache per core
[g] 4 way banked
[h] Second-half 2007

Slide 4

Instruction Set

- ➜ **RISC!**
- ➜ Load–store only through registers
- ➜ Fixed size instructions (32 bits)
  - ➜ *register + register*
  - ➜ *register + 13 bit immediate*
- ➜ Branch delay slot
- ✗ Condition Codes
  - ☑ (V9) CC and non-CC instructions
  - ☑ (V9) Compare on integer registers
- ➜ Synthesised instructions

Code Example

Slide 5

```
void addr(void) {
  int i = 0xdeadbeef;
}

00000054 <addr>:
  54:   9d e3 bf 90     save  %sp, -112, %sp
  58:   03 37 ab 6f     sethi  %hi(0xdeadbc00), %g1
  5c:   82 10 62 ef     or  %g1, 0x2ef, %g1
  60:   c2 27 bf f4     st  %g1, [ %fp + -12 ]
  64:   81 e8 00 00     restore
  68:   81 c3 e0 08     retl
  6c:   01 00 00 00     nop
```

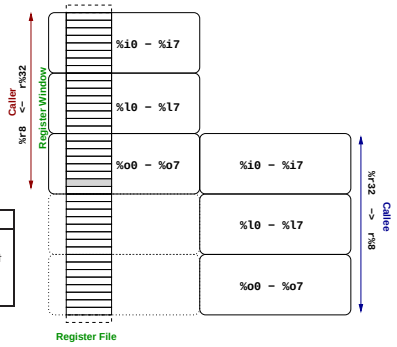
Registers

Slide 6

- Large Register File
- Fixed **register windows**
- Registers *renamed*

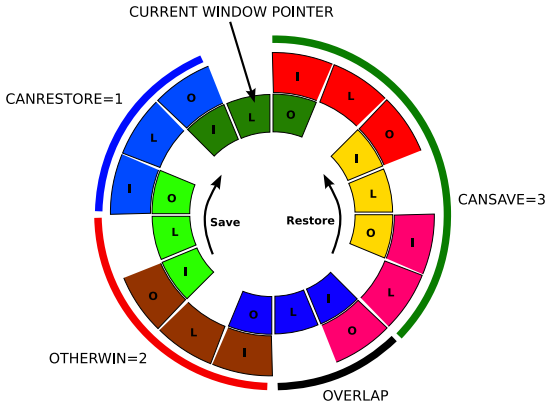
- **save** and **restore**

General	Windowed	Description
%r0 - %r7	%g0 - %g7	Global (all)
%r8 - %r15	%o0 - %o7	Window output
%r16 - %r23	%l0 - %l7	Window local
%r24 - %r31	%i0 - %i7	Window input



V9 Register Windows

Slide 7



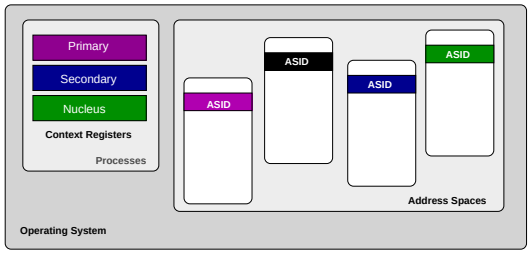
Register Windows Today

Slide 8

- ☒ (V8) Window buffer needs to be flushed
- ☒ Kernel code has deep call chains
  - ➜ Walks up and down a lot
- ☒ Question over studies showing advantages
  - ➜ State required for C compared to higher-level languages
- ☒ Less is **not** more
  - ➜ Superscalar needs a lot of registers
- ➜ Itanium
  - ☑ Variable sized windows
  - ☑ RSE deals with fill/spill
  - ☑ Allows for growth of underlying register file

Slide 9

Contexts

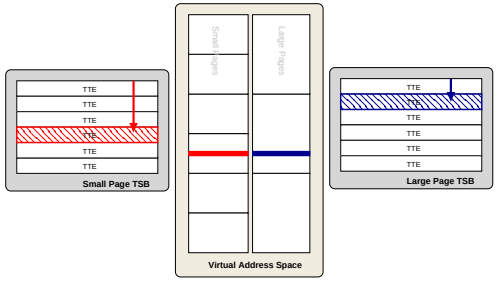


- Multiple Address spaces
- Primary, Alternate and explicit load instructions

Slide 10

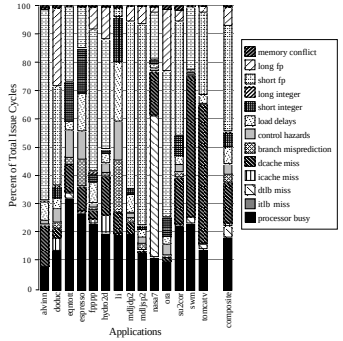
TSB

- **T**ranslation **S**tore **B**uffer is a **direct mapped cache** of ...
- **T**ranslation **T**able **E**ntries
  - `sun4u`, `sun4v`
- Hardware **pre-computes** index into TSB (for **2** specified page sizes)
- Software in **fast fault handler** can check if TTE valid



Slide 11

Throughput Computing



- 8 issue machine requires $\frac{1}{8}^{th}$ (12.5%) filled for CPI < 1

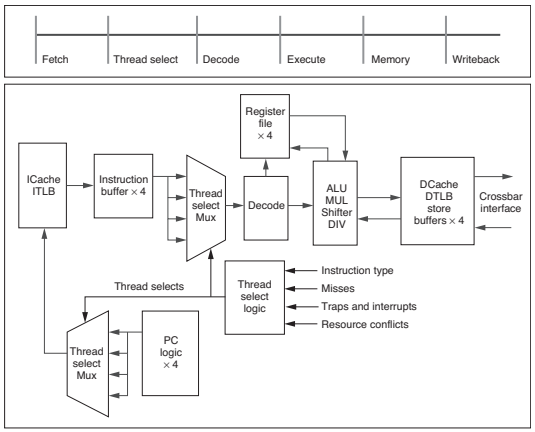
Slide 12

UltraSPARC T1

- **8** cores
- **4** threads / core – *thread group*
- **32** way multi-threaded
- **5.76** IPC (CPI 0.17, efficiency 71%)
- Good luck finding the clock speed (1.2Ghz)
- 70 Watts – "Green Processor"
- UltraSPARC Architecture 2005
  - Same underlying principles as SPARC V9

Slide 13

UltraSPARC T1 Pipeline



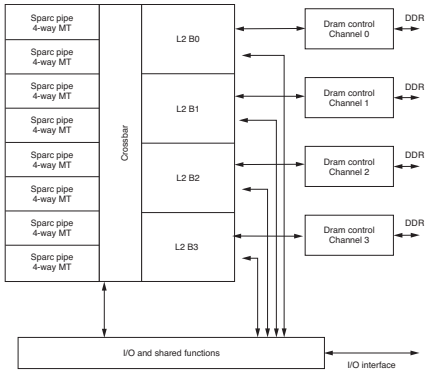
Slide 14

Chip Resources

- Per Thread
  - Registers
    - *Working Set* and *Architectural Set*
  - Instruction Buffer
- Per Core
  - **L1I** 16Kib, 4-way set associative, 32 byte lines
  - **L1D** 8KiB, 4-way set associative, 16 byte lines
  - **I/DTLB** 64-entry, fully associative
  - **Execution Units**
- Shared
  - **L2** 3MiB, 12-way set associative, 4-way banked
  - I/O

Slide 15

Chip Resources (2)



Slide 16

Thread Switching

- **Default** – switch per cycle, LRU
- Other heuristics go into thread swapping logic
  - **Predecoded Information** – long latency instructions
  - **Traps** – system calls, exceptions
  - **Resource Conflicts** – execution resources
  - **Cache Miss**

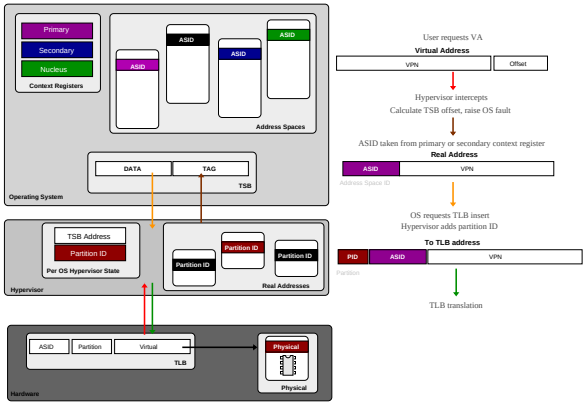
Slide 17

HYPERVISOR

- ➜ Unprivileged, Privileged, **Hyperprivileged**
- ➜ The hypervisor is the "hardware"
  - ➜ API and source code published
- ➜ Hyperprivileged resources
  - ➜ MMU
  - ➜ Interrupts
  - ➜ PCI
- ➜ Machine Description mechanism
- ➜ *Fast* and *Slow* trap mechanisms to privileged mode
- ➜ VA extended with partition ID

Slide 18

HYPERVISOR IN ACTION



Slide 19

OPEN SOURCE

- ☑ First open source processor
- ➜ `http://opensparc.sunsource.net/`
- ➜ Mailing Lists, Forums
- ➜ Bug reports and feedback

Slide 20

THANK YOU
QUESTIONS?