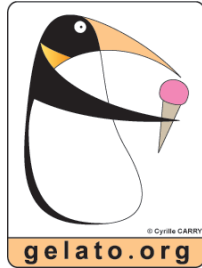


ULTRASPARC T1

THE PROCESSOR FORMERLY KNOWN AS "NIAGARA"



This work supported by UNSW and HP through the Gelato Federation

Slide 1

SPARC HISTORY

- Scalable Processor **ARCH**itecture
- 1985 – Sun Microsystems
 - Berkeley RISC – 1980-1984
 - MIPS – 1981-1984

Slide 2 Architecture v Implementation:

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

SUN + SPARC = ULTRASPARC

Processor	Cores	Threads/Core	Clock	L1D	L1I	L2 Cache
UltraSPARC III	1	1	550Mhz, 650Mhz	16KIB	16KIB	512KIB
UltraSPARC III	1	1	1.593Ghz	I	D	1MIB ^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+ (?)	?	?	?

Slide 3

- ^aOn-chip
- ^bExternal, on chip tags
- ^cUltraSPARC III cores
- ^d8MIB per core
- ^e32MIB off chip L3
- ^f1/D Cache per core
- ^g4 way banked
- ^hSecond-half 2007

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

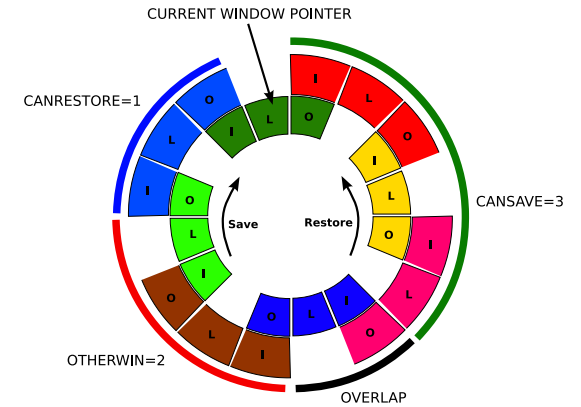
CODE EXAMPLE

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

Slide 5

```
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
```

V9 REGISTER WINDOWS



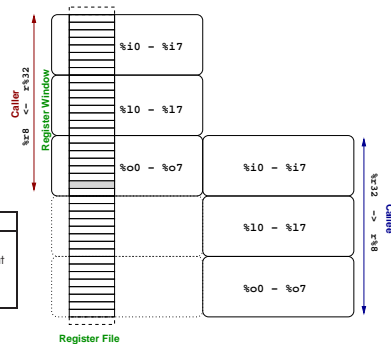
Slide 7

REGISTERS

- Large Register File
 - Fixed **register windows**
 - Registers *renamed*
- **save** and **restore**

Slide 6

General	Windowed	Description
%0 - %7	%g0 - %g7	Global (all)
%8 - %15	%o0 - %o7	Window output
%16 - %23	%l0 - %l7	Window local
%24 - %31	%i0 - %i7	Window input

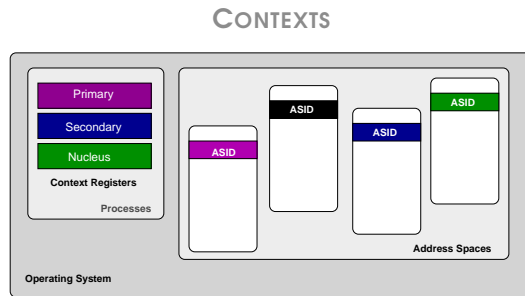


REGISTER WINDOWS TODAY

- ✗ (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 8

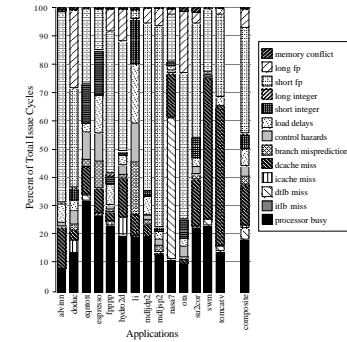
Slide 9



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

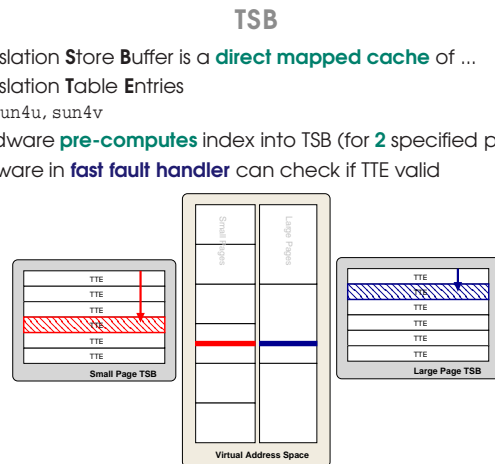
Slide 11

THROUGHPUT COMPUTING



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

Slide 10



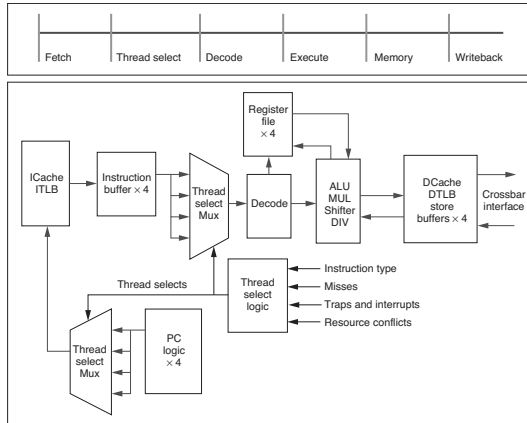
- Translation Store Buffer is a direct mapped cache of ...
- Translation Table Entries
 - sun4u, sun4v
- Hardware pre-computes index into TSB (for 2 specified page sizes)
- Software in fast fault handler can check if TTE valid

Slide 12

ULTRASPARC T1

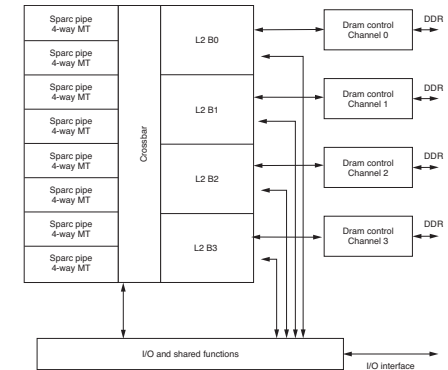
- 8 cores
- 4 threads / core – thread group
- 32 way multi-threaded
- 5.76 IPC (CPI 0.17, efficiency 71%)
- Good luck finding the clock speed (1.2Ghz)
- 70 Watts – “Green Processor”
- UltraSPARC Architecture 2005
 - Same underlying principles as SPARC V9

ULTRASPARC T1 PIPELINE



Slide 13

CHIP RESOURCES (2)



Slide 15

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 14

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 16

HYPERVISOR

Slide 17

- 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*

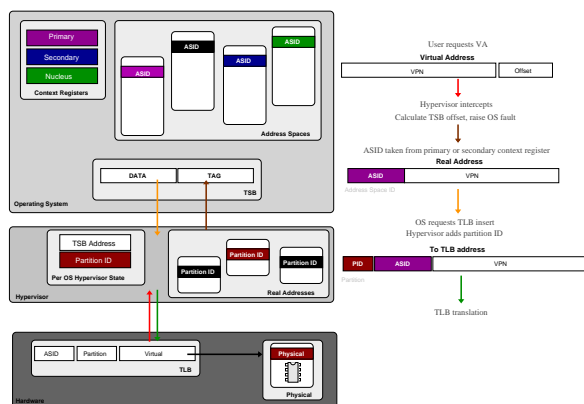
OPEN SOURCE

Slide 19

- ✓ First open source processor
- <http://opensparc.sunsource.net/>
- Mailing Lists, Forums
- Bug reports and feedback

HYPERVISOR IN ACTION

Slide 18



Slide 20

THANK YOU
QUESTIONS?