SPARC64™ VII
Fujitsu’s Next Generation Quad-Core Processor

August 26, 2008
Takumi Maruyama
LSI Development Division
Next Generation Technical Computing Unit
Fujitsu Limited
SPARC64™ VII Chip

- **Architecture Features**
  - 4core x 2threads (SMT)
  - Embedded 6MB L2$
  - 2.5GHz
  - Jupiter Bus

- **Fujitsu 65nm CMOS**
  - 20.31mm x 20.86mm
  - 600M transistors
  - 456 signal pins

- **135 W (max)**
  - 44% power reduction per core from SPARC64™ VI
SPARC64™ VII Design Target

• Upgradeable from current SPARC64™ VI on SPARC Enterprise Servers
  ➔ Keep single thread performance & high reliability by reusing SPARC64™ VI core as much as possible
  ➔ Same system I/F: Jupiter-Bus

• More Throughput Performance
  ➔ Dual core to Quad-core
  ➔ VMT (Vertical Multithreading) to SMT (Simultaneous Multithreading)

• Technical Computing
  ➔ Shared L2$ & Hardware Barrier
  ➔ Increased Bus frequency (on Fujitsu ‘FX1’ server)
Simultaneous Multi-Threading

- **Fine Grain Multi-thread**
  - Fetch/Issue/Commit stage: Alternatively select one of the two thread each cycle
  - Execute stage: Select instructions based on oldest-ready independently from threads.

  ➔ **SMT throughput increase**
  - x 1.2 INT/FP
  - x 1.3 Java
  - x 1.3 OLTP

- **Split HW Queues**
  - To avoid interaction between threads.
  - Automatically combined if the other thread is idle.

**SPARC64 VI**

Core0: t0 t1 t0 t1
Core1: t2 t3 t2 t3

**SPARC64 VII**

Core0: t1 t0 t1
Core1: t3 t2 t3
Core2: t5 t4 t5
Core3: t7 t6 t7

The other thread is idle

<table>
<thead>
<tr>
<th>#Active threads</th>
<th>IB</th>
<th>Reservation Station</th>
<th>Rename Registers</th>
<th>Port</th>
<th>CSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two</td>
<td>4+4</td>
<td>8x2</td>
<td>8x2</td>
<td>24+24</td>
<td>24+24</td>
</tr>
<tr>
<td>One</td>
<td>8</td>
<td>8x2</td>
<td>8x2</td>
<td>48</td>
<td>48</td>
</tr>
</tbody>
</table>

All Rights Reserved, Copyright © FUJITSU LIMITED 2008
Integrated Multi-core Parallel Architecture

- **L2 cache**
  - Shared by 4 Cores to avoid false-sharing
  - B/W between L2 cache and L1 cache
    (2x of SPARC64™ VI)
    - L2$\rightarrow$L1$: 32byte/cycle/2core x 4core
    - L2$\leftarrow$L1$: 16byte/cycle/core x 4core

- **Hardware Barrier**
  - High speed synchronization mechanism between cores in a CPU chip.
  - Reduce overhead for parallel execution

→ Handles the multi-core CPU as one equivalent fast CPU with Compiler Technology (Fujitsu’s Parallelnavi Language Package 3.0)
  - Automatic parallelization
  - Optimize the innermost loop

```
DO J=1,N
  DO I=1,M
    A(I,J)=A(I,J+1)*B(I,J)
  END
END
```
Hardware Barrier

- **Barrier resources**
  - BST: Barrier STatus
  - BST_mask: Select Bits in BST
  - LBSY: Last Barrier synchronization status
  - Synchronization is established when all BST bits selected by BST_mask are set to the same value.

- **Usage**
  - Each core updates BST
  - Wait until LBSY gets flipped

> Synchronization time: 60ns
  - 10 times faster than SW

Sample Code of Barrier Synchronization
```c
/*
 * %r1: VA of a window ASI, %r2:, %r3: work
 */
ldxa [%r1]ASI_LBSY, %r2 ! read current LBSY
not %r2 ! inverse LBSY
and %r2, 1, %r2 ! mask out reserved bits
stxa %r2, [%r1]ASI_BST ! update BST
membar #storeload ! to make sure stxa is complete
loop:
ldxa [%r1]ASI_LBSY, %r3 ! read LBSY
and %r3, 1, %r3 ! mask reserved bits
subcc %r3, %r2, %g0 ! check if status changed
bne,a loop
sleep ! if not changed, sleep for a while
```
Other performance enhancements

• Faster FMA (Fused Multiply-Add)
  – 7cycles → 6cycles
  – DGEMM efficiency: 93% on 4 cores
  → LINPACK=2,023Gflops with 64CPU

• Prefetch Improvement
  – HW prefetch algorithm further refined
  – SW is now able to specify “strong” prefetch to avoid prefetch lost.

• Shared context
  – Virtual address space shared by two or more processes
  – Effective to save TLB entries

• New Instruction
  – Integer Multiply-Add with FP registers

• etc…

⇒ Performance relative to SPARC64™ VI
  – Single thread performance: x1.0 - x1.2
  – Throughput: x1.7 - x1.9
# Reliability, Availability, Serviceability

## Existing RAS features

<table>
<thead>
<tr>
<th>Cache Protection</th>
<th>Tag</th>
<th>ECC (L2$) Duplicate &amp; Parity (L1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Degradation</td>
<td>Data</td>
<td>ECC (L2$, L1D$) Parity (L1I$)</td>
</tr>
<tr>
<td>ALU/Register</td>
<td></td>
<td>ECC (INT Regs) Parity (FP Regs, etc)</td>
</tr>
<tr>
<td>HW Instruction Retry</td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HW Instruction Retry</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

## New RAS features of SPARC64™ VII

- Integer registers are ECC protected
- Number of checkers increased to ~3,400

![Diagram showing RAS features and execution flow](image)
RAS coverage

- Green: 1bit error Correctable
- Yellow: 1bit error Detectable
- Gray: 1bit error harmless

More than 70% of Transient States are 1bit error detectable.
⇒ Recoverable through HW Instruction Retry

- All RAMs are ECC protected or Duplicated
- Most latches are parity protected

⇒ Guaranteed Data Integrity
Servers based on SPARC64™ VII

**SPARC Enterprise M8000**
- SPARC64™ VI or VII
- 46GB/s (for 4CPUs)

**FX1**
- Single CPU node
- Increased Jupiter Bus freq = 1.26GHz
- System chip is newly designed to realize
  - Higher memory throughput
  - Lower Memory latency
  - Smaller footprint
- Performance
  - FP throughput/socket: x1.5
  - STREAM benchmark: >13.5GB/s

**SPARC Enterprise: UNIX**
- Max 64CPUs / SMP
- Upgradeable from SPARC64™ VI
- Possible to mix SPARC64™ VI and SPARC64™ VII within the same SB
Performance Analysis

• Performance Analyzer
  – About 100 performance events can be monitored
  – Available through cputrack() and cpustat()
  – 8 performance events can be gathered at the same time.

• Commit base performance events
  – Number of commit instructions each cycle.
  – The cause of no commit
    • L2$miss
    • L1D$miss
    • Fetch miss
    • Execution Unit busy
    • …
Memory access cost has been reduced on FX1.
SPARC64™ Future

- Design History: Evolution rather than revolution
- SPARC64™ V (1core)
  - RAS
  - Single thread performance
- SPARC64™ VI (2core x 2VMT)
  - Throughput
- SPARC64™ VII (4core x 2SMT)
  - More Throughput
  - High Performance Computing
- What’s next?
SPARC64™ VII Summary

• The same system I/F with existing SPARC64™ VI to protect customer’s investment

• 4core x 2SMT design realizes high throughput without sacrificing single thread performance

• Shared L2$ and HW barrier makes 4 cores behave as a single processor with compiler technology.

• The new system design has fully exploited potential of SPARC64™ VII.

• Fujitsu will continue to develop SPARC64™ series to meet the needs of a new era.
Venus

- For PETA-scale Computing Server
  - 8core
  - Embedded Memory Controller
- SPARC-V9 + extension (HPC-ACE)
  - SIMD
- Fujitsu 45nm CMOS
- \( \rightarrow \) 128GF@socket
Abbreviations

• **SPARC64™ VII**
  - RSA: Reservation Station for Address generation
  - RSE: Reservation Station for Execution
  - RSF: Reservation Station for Floating-point
  - RSBR: Reservation Station for Branch
  - GUB: General Update Buffer
  - FUB: Floating point Update Buffer
  - GPR: General Purpose Register
  - FPR: Floating Point Register
  - CSE: Commit Stack Entry
  - FP: Fetch Port
  - SP: Store Port

• **Chip-sets**
  - SC: System Controller
  - MC: Memory Controller
  - JSC: Jupiter System Controller
  - SB: System Board

• **Others**
  - DGEMM: Double precision matrix multiply and add