SPARC64™ +:
HAL's Second Generation 64-bit SPARC Processor
HAL Computer Systems
Fujitsu Limited

OUTLINE

- SPARC64+ Project Goals
- SPARC64 Overview
- Improvements over SPARC64
- SPARC64+ Performance Data
- Summary
SPARC64+ Project Goals

- Improvements over HAL's first generation SPARC64 Processor
  - Improved Performance.
  - Performance Monitor Support.
  - Size Reduction.
  - Increased Debug Visibility.
  - Complete software compatibility with SPARC64.

SPARC64 Overview (First Generation)

- SPARC64 Processor Module is a ceramic MCM with
  - One CPU
  - Four CACHE Chips
  - One MMU and
  - One CLOCK chip
SPARC64 Overview (First Generation)....

CPU

- Superscalar, 4 issue, true 64 bit, V9-SPARC Implementation.
- Four stage Fixed-Point Instruction Pipeline.
- Six stage Load Instruction Pipeline.

<table>
<thead>
<tr>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
<th>(6)</th>
<th>(7)</th>
<th>(8)</th>
<th>(9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fetch</td>
<td>issue</td>
<td>execute</td>
<td>complete</td>
<td>deactivate</td>
<td>commit</td>
<td>retire</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>load instruction (cache hit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fetch</td>
</tr>
</tbody>
</table>

Figure 1: Pipeline Stages in SPARC64 CPU
SPARC64 Overview (First Generation)...

- Up to 64 Active instructions tracked through Dataflow execution.
- Employs Register Renaming, Dynamic Branch Prediction.
- ISSUE in-order, EXECUTE/COMPLETE Out-of-order and COMMIT in-order.
- Contains 2 Load/Store Units, 2 Fixed Point Units, 2 Address Generation units, 1 Floating Point Multiply Add unit and a Self timed Floating Point Divide Unit.

 CACHE

- Non-Blocking, with virtual index and virtual tag.
- Each chip is 64kB, can service 2 independent requests from CPU.
- 4 way set associative.
- Services Speculative/out-of-order CPU requests.
- 2 Instruction Cache chips, 2 Data Cache Chips.

SPARC64 CPU Block Diagram

- BHT
- RPT
- Prefetch Buffers
- I Cache
- Branch Unit
- Issue Unit
- Fetch Unit
- R-Insts
- Precise State Unit
- Instructions
- Status
- Execution Results and Status
- Rename and Register Files
- Ren
- FP
- Ren
- Integer
- Ren
- CC
- Branch Mispredict Handler
- Multiple Execution Units
- LS units
- Load/Store Pipe1
- Load/Store Pipe2
- Reservation Stations
- Agen
- LS
- FP
- Integer
- FP
- Integer

297 mm$^2$
2.7 million transistors
SPARC64 Overview (First Generation)....

- MMU
  - Interfaces with Virtual Caches, Memory system, I/O, and Diagnostic Processor.
  - A Linear 64-Bit Address Space supported through 2 level translations and 3 level memory hierarchy (Virtual, Logical and Physical Addresses).

![Address Translation Flow Diagram]

- 3 "lookaside buffers" reduce time for translation
  - 128-entry, fully associative VLB (VA --> LA)
  - 1024 entry, 4 way associative TLB (LA --> PA) and
  - 8 entry, 3 port, fully associative microTLB (VA --> PA).

- Contains a 1024 entry Data Cache Real Address Table.
  - Stores Real Address Tags of all Data Cache lines.
  - An entry match => real hit but virtual cache miss => no memory access required and cache line is re-tagged.

![Virtual Address Translation Diagram]

3.2-10
SPARC64 Overview (First Generation)....

- CLOCK
  - Under Diagnostic Processor Control, provides SYS-CLK/SCAN-CLK for all other MCM chips and Memory Subsystem.
  - Freq: 15~280MHz, Jitter 80~160ps, Programmable Delay Lines to control skews.

- PROCESS
  - Fujitsu's twin-well 0.4 u, 4 layer-metal, CS55 CMOS process.
  - 3 Metal layers for Signal routing.
  - 4th Metal layer for Bonding Pads, Global Power, Clock Routing.
  - Solder Bump Bonding for MCM-chip interconnection.

SPARC64+ Improvements (Performance)

- Cycle Time speedup ~20% in CPU
  - Fujitsu's faster CS60-process, smaller die size.
  - Removed 2 levels of logic from SPARC64 CPU long paths.
  - Improved/Fine-tuned Critical Timing Paths using
    - Faster Fixed/Floating Point - Register Files (Aggressive Circuits).
    - Faster Caches, Macros(Circuits).
    - Wider Instruction recoding (changed from 38 to 44 bits - requires less decoding in the fetch cycle).
    - More aggressive Execution Selection Algorithm.
SPARC64+ Improvements (Performance).

- Micro Architectural Changes to improve IPC*
  - Fixed Point register file modifications: +2.9%
    - Physical register size increased from 116 to 128.
    - Number of Register Windows increased from 4 to 5.
  - CPU on-chip Instruction Cache Size increased from 1k to 2k instructions: +3.0%.
  - Branch History Table Size Increase, Improved Memory Instruction scheduling: +2.0%.

* measured by SPECINT92, All improvement predictions based on "Timer" - HAL developed Performance Evaluation Tool.

SPARC64+ Improvements (Performance).

- Logic/Timing Reduction: Example

[Diagram showing dataflow and register file write/read for SPARC64+ CPU]

SPARC64+ CPU: Register file read/write for Data forward

---

HAL Computer Systems

Hot Chips VII Stanford, California, August 14–15, 1995
SPARC64+ Improvements (Performance)

- N(egative), Z(ero) bits not stored in Register File.
- Dataforward results written into Reg.file one cycle earlier.
- Enables removal of bypass muxes/comparators in Reg.file read path and hence in timing. (10 read ports).

SPARC64+ Improvements....(Debug)

- Chips on MCM are flip mounted.
  - MCM chips can’t be probed for tests.
  - SPARC64 Processor chips provide dedicated MCM visible signals for Logic Analyser hookup during Debug.
  - Added more signals for tracking internal chip states.
SPARC64+ Improvements ..(Debug)

- CPU related Debug Bus (89 bits) has 4 views.
  - 1. Instruction Tracking Bus View.
  - 2. Instruction Fetch Bus View.
  - 3. Load Store Bus View.
  - 4. CPU- Internal State View

SPARC64+ Improvements (Perf.Monitoring)

- SPARC64 MMU incorporates Performance counters for
  - observing Lookaside Buffer(s) performance.
  - Cache hit/miss rates, Data cache replacement rates.
- In SPARC64+, CPU Performance Monitoring features added.
  - HAL implementation specific architecture feature.
  - Software visible and privileged.
  - Kept it simple though the system supports Speculative/ out-of-order executions, Retry/Block condition and Instruction/Data Pre-fetches.
SPARC64+ Improvements
(Perf. Monitoring)

- Performance Monitors/Counters in CPU provided to
  - Count Instruction Issue stalls due to Fetch, Resources
    (Execution units, Reservation Stations, Free Registers)
    and Precise State Exceptions.
  - Measure Instruction Issue/Commit rates.
  - Measure Total Latency of Memory accesses.
  - Measure Data Cache Hit Rates.
  - Measure Memory Access rates.
- Count Accumulation/Monitor interval period: 100msec.

SPARC64+ Improvements (Size Reduction)

- SPARC64+, CPU die size: \(~200\text{mm}^2\)
- 33% reduction over SPARC64 CPU due to
  - reduction in metal pitch.
  - reduced transistor feature size.
  - addition of 5th metal layer for routing.
SPARC64+ Performance Data

- Performance Improvement over SPARC64 (due to hardware only)

Estimated Performance change for SPARC64+

<table>
<thead>
<tr>
<th>Description</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle Time improvement</td>
<td>20%</td>
</tr>
<tr>
<td>IPC improvements</td>
<td>8%</td>
</tr>
<tr>
<td>Total (1.08X1.20)-1</td>
<td>-30%</td>
</tr>
</tbody>
</table>

1. refer to Functional Changes section.

SUMMARY

- 30% performance improvement over SPARC64 is achieved in hardware through Cycle Time Reduction, Addition of on-chip resources and Micro-architectural changes.
- Performance monitoring counters added in CPU to
  - observe/analyse the system behavior.
  - fine-tune/optimize Compilers.
- Chip Size reduced by 33%.
- Wider Debug Bus increased the observability of the system during Hardware Debug.
- Existing System Software/Binaries for SPARC64 will run efficiently on SPARC64+, without any changes.
- Restricted the micro-architectural changes to meet short development schedule requirements.
- Design Tool Flow changes, Verification Efforts due to the changes were kept minimal.
Acknowledgement :
Jonathan Chang, Anand Dharmaraj, Michael Filardo,
Atsushi Ike*, Bala Joshi, Takeshi Kitahara*,
Anand Krishnamoorthy, Simon Li, Sanjay Mansingh,
Osamu Moriyama*, Arvind Narayan, Kesiraju Rao,
Murugappan Ramaswami, Farnad Sajjadi, Mike Simone,
Gene Shen**, Ravi Swami, John Szeto, Vijji Thirumalaiswamy
Shalesh Thusoo, DeFrost Tovey

*Fujitsu Limited, **Exponential Technology(current affiliation)