One size doesn’t fit all

SPARC chip technology available in three broad product categories:

- S-Series (servers, highest performance)
- I-Series (price/performance, ease-of-use)
- E-Series (lowest cost)

Tired: Marginal micro-optimizations

Wired: Interfaces and integration.
Desktop/High end embedded system issues

- Ease of design-in
- Low-cost
- Simple motherboard designs
- Power
- Higher performance graphics interfaces
- Low latency to memory and I/O
- Upgrades
- The J word

UltraSPARC-IIi System Example

- SRAM
- UltraSPARC III
- APB
- Advanced PCI Bridge (optional)
- 66 or 33 MHz
- 32-bit
- 3.3v
- 72-bit DIMM
- 75MHz/72-bit
- XCVRs
- 72-bit DIMM
- 100MHz/64-bit
- UPA64S device
- example: Creator3D graphics
- 33MHz/32-bit
- 5v/3.3v
- Can use PC-style SUPERIO chips
  (boot addressing / INT_ACK features)
- 33MHz/32-bit
  5v/3.3v
Highlights

- **TI Epic4 GS2, 5-layer CMOS, 2.6v core, 3.3v I/O.**
- **Sun’s Visual Instruction Set - VIS\(^{(TM)}\)**
  - 2-D, 3-D graphics, image processing, real-time compression/decompression, video effects
  - block loads and stores (64-byte blocks) sustain 300 MByte/s to/from main memory, with arbitrary src/dest alignment
- **Four-way superscalar instruction issue**
  - 6 pipes: 2 integer, 2 fp/graphics, 1 load/store, 1 branch
- **High bandwidth / Low latency interfaces**
  - Synchronous, external L2 cache, 0.25MB - 2MB
  - 8-byte (+parity) data bus to L2 cache (1.2 GByte/s sustained)
  - 8-byte (+ECC) data bus to DRAM XCVRs (400 MByte/s sustained) or UPA64S (800 Mbyte/s sustained)
  - 4-byte PCI/66 interface. (190 Mbyte/s sustained)

Highlights (continued)

- **Instruction prefetching**
  - in-cache, dynamic branch prediction
  - 1 branch per cycle
  - 2-way, 16kB Instruction Cache
- **Non-blocking Data Cache**
  - 8-entry load buffer: pipe stalls only if load data needed
  - 8-entry (8 bytes per entry) store buffer for write-thru to L2 cache
  - single-cycle load-use penalty for D-cache hit
  - throughput of one load/store per 2 cycles to L2 cache
- **Non-blocking software prefetch to L2 cache**
  - Prefetch completion isn’t constrained by the memory model.
  - Compiler uses to further hide the latency of L2 miss.
- **Simultaneous support of little/big endian data**
Instruction Pipelines

**Integer**

- **F** Fetch: Instructions are fetched from i-cache
- **D** Decode: Instructions are decoded and placed in the instruction buffer
- **G** Group: Up to 4 instructions are grouped and dispatched. RF accessed
- **E** Execute: Integer instructions are executed and virtual addresses calculated
- **C** Cache Access: Dcache/TLB accessed. Branch resolved
- **N1**: Dcache hit or miss determined. Deferred load enters load buffer
- **N2**: Integer pipe waits for floating-point/graphics pipe
- **N3**: Traps are resolved
- **W** Write: All results are written to the register files. Instructions are committed

**Floating-point/Graphics**

- **R** Register: Floating-point/graphics instructions are further decoded. RF accessed
- **X1** Start execution
- **X2** Execution continued
- **X3** Finish execution

**Block Diagram**

- iCache
- IMMU
- Prefetch and Dispatch Unit
- Branch Prediction
- Branch Unit
- Integer Execution Unit
- Load/Store Unit
- Floating Point/Graphics Unit
- Load Buffer
- DCache
- DMMU
- Store Buffer
- L2 Cache, DRAM, UPA64S, PCI
- 64-byte DRAM write buffer
- IOMMU
- Separate 64-byte PIO and DMA buffers
- Interfaces
- Second-Level Cache: 64x8 (parity)
- TI XCVRs
- DRAM Data: 128x16 (ECC)
- PCI 66MHz
- DRAM
- PCI 66MHz
- 36+1 (parity)
- 64x8 (ECC)
- 64x8 (parity)
Prefetch and Dispatch Unit

- 16kB ICache, 2-way set associative, 32B line size w/pre-decoded bits
- 2kB Next Field RAM which contains 1k branch target addresses and 2k dynamic branch predictions
- 4-entry Return Address Stack for fetch prediction
- Branch prediction (software/hardware)
- 64-entry, fully associative ITLB backs up 1-entry µTLB
- 12-entry Instruction Buffer fed by ICache or second-level cache
- Single-cycle dispatch logic considers “top” 4 instructions
- Controls Integer operand bypassing

Integer Execution Unit

- Integer Register File has 7 read ports/3 write ports
- 8 windows plus 4 sets of 8 global registers
- Dual Integer Pipes
- ALU0 executes shift instructions
- ALU1 executes register-based CTIs, Integer multiply/divide, and condition code-setting instructions
- Integer multiply uses 2-bit Booth encoding w/“early out” -- 5-cycle latency for typical operands
- Integer divide uses 1-bit non-restoring subtraction algorithm
- Completion Unit buffers ALU/load results before instructions are committed, providing centralized operand bypassing
- Precise exceptions and 5 levels of nested traps
Floating-Point/Graphics Unit

- Floating-point/Graphics Register File has 5 read ports/3 write ports
- 32 single-precision/32 double-precision registers
- 5 functional units, all fully pipelined except for Divide/Square Root unit
- High bandwidth: 2 FGops per cycle
- Completion Unit buffers results before instructions are committed, supporting rich operand bypassing

- Short latencies
  - Floating-point compares: 1 cycle
  - Floating-point add/subtract/convert: 3 cycles
  - Floating-point multiplies: 3 cycles
  - Floating-point divide/square root(sp): 12 cycles
  - Floating-point divide/square root(dp): 22 cycles
  - Partitioned add/subtract, align, merge, expand, logical: 1 cycle
  - Partitioned multiply, pixel compare, pack, pixel distance: 3 cycles

Load/Store Unit

- 16 kB DCache (D$), direct-mapped, 32B line size w/16B sub-blocks
- 64-entry, fully associative DTLB supports multiple page sizes
- D$ is non-blocking, supported by 9-entry Load Buffer
- D$ tags are dual-ported, allowing a tag check from the pipe and a line fill/snoop to occur in parallel
- Sustained throughput of 1 load per 2 cycles from second-level cache (E$)
- Pipelined stores to D$ and E$ via decoupled data and tag RAMs
- Store compression reduces E$ bus utilization
External Cache Control

- E$ sizes from 0.25 MB to 2MB
- E$ is direct-mapped, physically indexed and tagged, w/64B line size
- Uses synchronous SRAMs with delayed write
- 8B (+ parity) interface to E$ supports 1.2 GB/s sustained bandwidth
- Supports two sram modes, compatible with US-I or US-II srams
- ad and dt are 2 cycles each. ac stage doesn’t exist for 22 mode srams
- PCI activity is fully cache coherent
- 16-byte fill to L1 data cache, desired 8 bytes first

Pins (balls)

- 90 - L2 cache data and tag
- 37 - L2 cache addr/control
- 8 - L2 cache byte write enables
- 53 - PCI plus arbiter
- 9 - Interrupt interface
- 35 - UPA64S address/control interface
- 34 - dram and transceiver control
- 72 - dram/UPA64S data
- 18 - clock input and reset/modes
- 10 - jtag/test

total with power: 587
## Performance (300mhz)

### SPEC

<table>
<thead>
<tr>
<th></th>
<th>0.5mbyte L2</th>
<th>2.0 Mbyte L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPECint95</td>
<td>11.0</td>
<td>11.6</td>
</tr>
<tr>
<td>SPECfp95</td>
<td>12.8</td>
<td>15.0</td>
</tr>
</tbody>
</table>

### Memory

<table>
<thead>
<tr>
<th>Operation</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2-Cache read</td>
<td>1.2 GBytes/S</td>
</tr>
<tr>
<td>L2-Cache write</td>
<td>1.2 GBytes/S</td>
</tr>
<tr>
<td>DRAM read (random)</td>
<td>350 Mbytes/S</td>
</tr>
<tr>
<td>DRAM write</td>
<td>350 Mbytes/S</td>
</tr>
<tr>
<td>DRAM read (same page)</td>
<td>400 Mbytes/S</td>
</tr>
<tr>
<td>DRAM, memcopy</td>
<td>303 Mbytes/S</td>
</tr>
<tr>
<td>DRAM, memcopy to UPA</td>
<td>550 Mbytes/S</td>
</tr>
</tbody>
</table>

**STREAM (Compiled, and with VIS block store)**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy</td>
<td>199 Mbytes/S</td>
</tr>
<tr>
<td>Scale</td>
<td>199 Mbytes/S</td>
</tr>
<tr>
<td>Add</td>
<td>224 Mbytes/S</td>
</tr>
<tr>
<td>Triad</td>
<td>210 Mbytes/S</td>
</tr>
</tbody>
</table>

## Performance

### PCI (66 MHz/32 bits, Single Device)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI to DRAM (DMA)</td>
<td>151 Mbytes/S</td>
</tr>
<tr>
<td>PCI from DRAM (DMA)</td>
<td>132 Mbytes/S</td>
</tr>
<tr>
<td>PCI to L2 Cache (DMA)</td>
<td>186 Mbytes/S</td>
</tr>
<tr>
<td>PCI from L2 Cache (DMA)</td>
<td>163 Mbytes/S</td>
</tr>
<tr>
<td>CPU to PCI (PIO)</td>
<td>200 Mbytes/S</td>
</tr>
</tbody>
</table>

### UPA64S (100 MHz/64 bits)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU to UPA (PIO)</td>
<td>800 Mbytes/S</td>
</tr>
<tr>
<td>CPU to UPA (PIO)</td>
<td>600 Mbytes/S</td>
</tr>
</tbody>
</table>
Memory

- Control/data transitions align to cpu clock
- Registered Transceivers provide 144-bit path to DRAM
- Standard DIMMs
- Data transfers at 75MHz
- ECC supported, but not required
- Page mode support: 3 outstanding requests
- Equal performance for range of CPU clocks

Memory Latency (300mhz)

Additional delay a sequential load-use pair sees:

1 cycle to L1 cache: 3.3ns
+ 8 cycles to L2 cache: 26.5ns
+ 41 cycles to DRAM: 135.3ns (16 bytes)

L2 fill continues for 22 cycles more: 72.6ns (48 bytes)

Optimized for page miss, but can take advantage of page hits if next request arrives in time.

(page miss numbers above)
UPA64S

- Simple split transaction protocol. Addr/Data/Preply/Sreply pins
- 1/3 of Processor clock rate (100MHz)
- Sequential store compression for improved bandwidth
- Overlaps 64-byte transfer with DRAM access. Two protocols on one data bus
- 92% data bus utilization during memcopy

Instruction and Data MMUs

- 44 bit Virtual to 41 bit Physical translation
- 64 entry iTLB and dTLB
- Fully associative with LRU replacement
- Software miss processing with acceleration for software TSB
- Multiple page sizes (8KB, 64KB, 0.5M, 4MB)
- Invert endian bit
- Nucleus, Primary, Secondary Contexts
- Nested trap support (trap in trap handler)
PCI

- Registered Outputs, non-registered Inputs
- Clock domain independent from core CPU
- Runs internally at 132MHz (2x interface speed)
- IOMMU provides address translation/protection

PCI MMU

- 32-bit IO VA to 34-bit PA translations
- Supports single-level hardware tablewalk
- 8K & 64K (as well as mixed) page sizes supported
- Based on 16-entry fully associative TLB
Advanced PCI Bridge

- 66 MHz/32 bit + Two 33MHz/32 bit
- Queueing of stores
- Prefetch behavior
- Ordering interrupts and prior DMA writes

Flexible Interrupts

- Level interrupts encoded as 6-bit INT_NUMs, up to 48 unique.
- US-IIi hardware FSMs filter duplicate INT_NUMs until software clears the interrupt.
- 11-bit value in “mondo vector” registers identifies interrupt precisely.
- Pending internal “mondo vector” causes trap to handler. Special global registers available to avoid state save before processing.
- Software unloads “mondo vector” and clears out the appropriate state to allow another interrupt of that type to occur.
- SOFTINT register can be used to create delayed handling of interrupts with priority levels
Chip Summary

- 0.25u poly, 0.21u Leffective, 5-layer CMOS, 2.6v core, 3.3v I/O.
- 12mm x 12.5mm die. 5.75 million transistors
- 24 watts at 300mhz, with integrated I/O, memory, graphics and interrupt interfaces.
- 66MHz PCI
- High bandwidth Graphics port (UPA64S)
- Low latency to memory (16MB-1GB)
- 11.6/15.0 SPEC95 with 2.0Mbyte L2 cache
- Lower frequency/power options available.
- PC system components with >PC performance.
- Fully UltraSPARC. (VIS etc)

UltraSPARC-IIi

Continued integration of system functions onto a single die

Benefits: performance, cost, ease of use

Higher and lower frequencies coming

UltraSPARC performance into a broader range of products