Jaguar: A Next-Generation Low-Power x86-64 Core

Teja Singh, Joshua Bell, Shane Southard, Deepesh John
AMD, Austin TX
Outline

- Motivation
- Architecture
- Technology
- Implementation
- Circuits
- Clocking
- Timing
- Power
- Reliability
- Conclusion
Motivation

• First AMD 28nm quad-core x86-64
• Build unit to deploy into a wide variety of SoCs for different applications
• Span wide array of applications from sub 5W to 25W
• Worthy successor to “Bobcat” x86-64 core
Target Markets

• Build SoC to fit range of markets
  – Tablet, hybrids
  – Value notebook
  – Ultrathin notebook
  – Value desktop
## Core Comparison

<table>
<thead>
<tr>
<th></th>
<th>&quot;Bobcat&quot; (BT)</th>
<th>&quot;Jaguar&quot; (JG)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Process</strong></td>
<td>40nm bulk</td>
<td>28nm bulk</td>
</tr>
<tr>
<td><strong># Cores</strong></td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td><strong>L2 Cache Size</strong></td>
<td>1MB (512KB dedicated 16-way)</td>
<td>2MB (shared, 4x 512KB 16-way)</td>
</tr>
<tr>
<td><strong>Core Size</strong></td>
<td>4.9mm(^2)</td>
<td>3.1mm(^2)</td>
</tr>
<tr>
<td><strong>Core Flop Count</strong></td>
<td>159900</td>
<td>194490</td>
</tr>
<tr>
<td><strong>Machine Width</strong></td>
<td>2-wide</td>
<td>2-wide</td>
</tr>
<tr>
<td><strong>Physical Address</strong></td>
<td>36-bit</td>
<td>40-bit</td>
</tr>
<tr>
<td><strong>L1 Instruction Cache</strong></td>
<td>32kB, 2-way 64B line</td>
<td>32kB, 2-way 64B line</td>
</tr>
<tr>
<td><strong>L1 Data Cache</strong></td>
<td>32KB, 8-way 64B line</td>
<td>32KB, 8-way 64B line</td>
</tr>
<tr>
<td><strong>Load/Store Bandwidth</strong></td>
<td>8B/cycle, Write Combine</td>
<td>16B/cycle, Write Combine</td>
</tr>
<tr>
<td><strong>FPU Datapath</strong></td>
<td>64-bit</td>
<td>128-bit</td>
</tr>
<tr>
<td><strong>EX Scheduler</strong></td>
<td>16 entries</td>
<td>20 entries</td>
</tr>
<tr>
<td><strong>AGU Scheduler</strong></td>
<td>8 entries</td>
<td>12 entries</td>
</tr>
</tbody>
</table>
Architecture

- ISA enhancements added
  - SSE4.1, SSE4.2
  - Advanced Vector Extensions
  - AES, CLMUL
  - MOVBE
  - XSAVE/XSAVEOPT
  - F16C, BMI1
- 4x32B Instruction Cache loop buffer for power
- Improved Instruction Cache prefetcher for IPC
- Added hardware integer divider
- L2 prefetcher
- Improved C6 and CC6 entry/exit latencies
- Estimated typical IPC improvement over “Bobcat”: >15%*
- Clock gate >92% flops in typical applications

* Estimates based on internal AMD modeling using benchmark simulations. This information is preliminary and subject to change without notice.
**Technology**

- **TSMC 28nm bulk HKMG**
- **3 Vt solution:** HVT/RVT/LVT
- **Longer lengths for each Vt**
- **BT had 10 metal stack**
- **JG uses 11 metal stack**
  - stdcells block most of M2
  - additional 2x layer added to offset loss of tracks

<table>
<thead>
<tr>
<th>Layer</th>
<th>BT Type</th>
<th>BT Pitch</th>
<th>JG Type</th>
<th>JG Pitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>1x</td>
<td>126nm</td>
<td>1x</td>
<td>90nm</td>
</tr>
<tr>
<td>M2-M8</td>
<td>1x</td>
<td>126nm</td>
<td>1x</td>
<td>90nm</td>
</tr>
<tr>
<td>M9</td>
<td>14x</td>
<td>900nm</td>
<td>2x</td>
<td>180nm</td>
</tr>
<tr>
<td>M10</td>
<td>14x</td>
<td>900nm</td>
<td>10x</td>
<td>900nm</td>
</tr>
<tr>
<td>M11</td>
<td>n/a</td>
<td>n/a</td>
<td>10x</td>
<td>900nm</td>
</tr>
</tbody>
</table>

*Reference: Wuu, Shien-Yan, et al.. 2009 Symposium on VLSI Technology Digest. pp 210-211*
Implementation Overview

• Focus on density
  – Use high density 9 track library
  – Use 1x metals to increase routing resources
  – Implemented using large units to reduce boundary cases
    • Core is 1.25 million placed instances
    • L2I is 0.6 million placed instances
• Standard auto place and route design style
• JG Core has 2 unique custom arrays
• Achieved silicon frequency >1.85Ghz*
• Integrated Power Gating
• Power supply via towers oriented based on route congestion

* Estimates based on internal AMD modeling using benchmark simulations. This information is preliminary and subject to change without notice.
Compute Unit Floorplan
Core Floorplan

- Branch
- Ucode ROM
- Integer
- Data Tag/TLB
- Bus Unit
- Data Cache
- Inst. Tag/TLB
- Inst Cache
- x86 Decode
- Debug
- ROB
- Load/Store
- FP
- Test
Core Power Gating

- Headers have 4 independent enables to control strength: \{\frac{1}{15}, \frac{2}{15}, \frac{4}{15}, \frac{8}{15}\} of total width
- Diagram showing highlighted headers within the JG core
- Area overhead is \sim 3\%
Custom Array Power Gating

- Arrays have integrated power header columns
- Control section with more columns to reduce IR drop
- ROM array shown above ~390k bits
Core Power Gating

- Contour IR map of power headers on the JG core
- Showing worst case pattern during a dynamic IR analysis
- Header IR drop is <20mV*; total IR drop within design limits

* Estimates based on internal AMD modeling using benchmark simulations. This information is preliminary and subject to change without notice.
Compute Unit IR Map

- IR map using a worst case pattern highlighting areas with larger drops
- Showing worst case pattern during a dynamic IR analysis
Circuit Overview

- Reduce custom array count from BT
  - RAM array module
  - ROM array module
- Focus on process portability
- Used high speed flops in top critical timing paths
- Arrays utilize fuse programmability for flexibility and reuse
High Speed Flop

<table>
<thead>
<tr>
<th></th>
<th>Clk2q Optimized High Speed Flop vs Nominal Flop</th>
<th>Setup Optimized High Speed Flop vs Nominal Flop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup/Tcycle</td>
<td>6.7% speedup</td>
<td>14.0% speedup</td>
</tr>
<tr>
<td>ClkToQ/Tcycle</td>
<td>6.1% speedup</td>
<td>0.3% speedup</td>
</tr>
<tr>
<td>Hold</td>
<td>220%</td>
<td>360%</td>
</tr>
<tr>
<td>Cac</td>
<td>172%</td>
<td>207%</td>
</tr>
<tr>
<td>Area</td>
<td>160%</td>
<td>180%</td>
</tr>
</tbody>
</table>
High Speed Flop

Used in critical paths

~70 high speed flop variants based on
#2, #3: Combinational function
#5: Output buffer drive strength
#4, #5: Edge rate control on outputs
  – Rising edge skewed
  – Falling edge skewed
  – Balanced edges
#1, Input clock pin buffering
  – Unbuffered to favor Clk2Q
  – Buffered to favor setup
RAM Fuse Capabilities

- RAM array reuse was a goal; 51 instantiations within the JG Core, 276 instantiations within the Compute Unit
- Utilize fuse capabilities to tune the design
Array Read Timing Fuses

- FUSE1 (Read Address) and FUSE3 (Read Data) are used to modulate a half cycle access/write time.
- These fuses control programmable delay cells and can be set per macro instantiation.
Array Read Timing Fuses

<table>
<thead>
<tr>
<th>Settings</th>
<th>Read Address delay (normalized to clock period)</th>
<th>Read Data delay (normalized to clock period)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High Voltage</td>
<td>Low Voltage</td>
</tr>
<tr>
<td></td>
<td>High Voltage</td>
<td>Low Voltage</td>
</tr>
<tr>
<td>00</td>
<td>14%</td>
<td>12%</td>
</tr>
<tr>
<td></td>
<td>11%</td>
<td>9%</td>
</tr>
<tr>
<td>01</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td>7%</td>
<td>6%</td>
</tr>
<tr>
<td>10</td>
<td>10%</td>
<td>9%</td>
</tr>
<tr>
<td></td>
<td>15%</td>
<td>12%</td>
</tr>
<tr>
<td>11</td>
<td>18%</td>
<td>15%</td>
</tr>
<tr>
<td></td>
<td>18%</td>
<td>15%</td>
</tr>
</tbody>
</table>

- Four settings for both sets of fuses
- Delay ranges from 5-18% of clock period
Keeper Enable Fuse

Keeper Enable signal can be delayed to improve performance or can be turned to an *Always ON* state for improved noise immunity.
### Keeper Enable Fuse

<table>
<thead>
<tr>
<th>Settings</th>
<th>Bitline to Keeper Enable Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High Voltage</td>
</tr>
<tr>
<td></td>
<td>(Normalized to clock period)</td>
</tr>
<tr>
<td>00</td>
<td>1%</td>
</tr>
<tr>
<td>01</td>
<td>6%</td>
</tr>
<tr>
<td>10</td>
<td>11%</td>
</tr>
<tr>
<td>11</td>
<td>Always ON</td>
</tr>
</tbody>
</table>

- In the default case Keeper Enable turns on just after the bitline falls
- The keeper device is always on for 11 setting
Write Wordline Pulse Width Fuse

- WWL pulse width is chopped based on fuse settings
- Allows silicon measurement of write margin
Write Wordline Pulse Width Fuse

<table>
<thead>
<tr>
<th>Settings</th>
<th>Write Wordline (Normalized to clock period)</th>
<th>Pulse Width (Normalized to clock period)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>56%</td>
<td>52%</td>
</tr>
<tr>
<td>01</td>
<td>34%</td>
<td>31%</td>
</tr>
<tr>
<td>10</td>
<td>28%</td>
<td>25%</td>
</tr>
<tr>
<td>11</td>
<td>18%</td>
<td>16%</td>
</tr>
</tbody>
</table>

- Pulse width is ~50% of clock period for the default setting
- Pulse width is controlled by combining write clock and its delayed inverted version
- Pulse width for non default settings are frequency independent
CU Level Clock Distribution

- Matched clock delay to all endpoints to minimize latency
- Each unit’s clock independently gated to reduce dynamic power
- L2D half frequency operation supported without adding additional stages to clock path

© 2013 IEEE

IEEE International Solid-State Circuits Conference
• Clock dividing for various operating modes
• Duty cycle adjuster for independent control of duty cycle within each block
Core clock distribution & CTS

- Low skew recombinant mesh design
- Mesh driven by configurable custom cells to enable faster design closure and tunability
- Multipoint CTS start points created by preplacing 2 levels of inverters
- Delete unused S1/S2 levels
Timing Methodology

• Primary design optimization uses all Low Vt for speed and area
• Multi-Vt optimization done multiple times post-placement and in eco to reduce leakage
• Use Monte Carlo simulations to calculate Vt derates applied to High Vt and Regular Vt cells based on their variation relative to Low Vt
  – Ensure cells with large variation get sufficient margin
  – Ensure Si-critical paths are set by Low Vt
• Exclude cells with sigma/mean ratio worse than a set floor
  – Enable operation at lower voltages and expedite hold timing closure
Silicon Results

Normalized Frequency

Normalized Voltage
Power

• Dynamic
  – Reduced number of clock spines versus BT
  – Remove unused S1/S2 clock inverters
  – Move clock spine to Low Vt versus BT
  – Gate L2 clock when L2 not accessed

• Static
  – Always ON buffer tree for power gate enables use longer length Hvt
  – Vt usage tuned within custom arrays
  – Measured silicon shows JG power gated leakage <10mW*

* Estimates based on internal AMD modeling using benchmark simulations. This information is preliminary and subject to change without notice.
Power Breakdown

JG Core Vt Cell Usage

- High Vt, long length: 27%
- Low Vt: 13%
- Low Vt, long length: 12%
- Nominal Vt: 23%
- Nominal Vt, long length: 4%

JG Core Total Power Breakdown

- Clock Grid: 21%
- Custom Macros: 13%
- Stdcell Logic: 28%
- Registers: 21%
- Clock Gaters: 17%
- Clock: 21%
Reliability

- Design for superset of usage model conditions
- Numerous challenges for 28nm Vmin/Vmax support:
  - Time dependent and intra-metal dielectric breakdown
  - Bias Temperature Instability (BTI)
    - Use foundry calculator to determine Vt shift for given usage model
    - Use Vt shift in critical path simulations to gauge frequency degradation
    - Margin timing paths across units with different usage conditions via clock uncertainty
    - Compare pre-silicon to measured Si degradation
  - Electro-migration
    - Require statistical EM budgeting to close longest lifetime parts
    - Thermal solve used to reduce self heat pessimism for Irms calculations
    - Thermal map of RAM array shown
Conclusion

- “Jaguar” is first AMD 28nm bulk CPU
- Quad core with shared L2
- Substantially higher IPC and frequency than BT
- Unit built for reuse in multiple SoCs
- Design methods increase process portability
- Focus on high density and smaller chip area
- Low power and low skew configurable clock tree
- Highly utilize SAPR design flow but customize for high speed flops and programmable custom arrays