The Implementation of a 2-core, Multi-threaded Itanium® Family Processor

Samuel Naffziger
Blaine Stackhouse
Tom Grutkowski
Intel Corp. Fort Collins, CO
Outline

• Processor overview
• Core improvements
  – Cache hierarchy
  – Multi threading
  – Power reduction
• Handling 4 voltage and frequency domains
• Power management and reduction
• Conclusion
Itanium’s Progress

• 90nm implementation optimized for server workloads
  – Two dual-threaded high performance 64b EPIC cores on a single chip
  – (very) Large on-chip caches totaling 26.5MB
    • The same latencies as previous Itaniums (paper 26.8)
• 1.72 Billion transistors, 596mm²
• Legacy system bus at higher frequency (666MT/s)
• High frequency operation representing at least a 20% increase over the 130nm implementation of the same core micro-architecture
• Power efficiency improvements that deliver a >2 fold compute capability increase at 23% less power consumption
Business Critical Features

- 1.72 Billion transistors
- 1MB L2I
- 2-Way Multi-Threading
- 2 X 12MB L3 Caches with Pellston
- Soft Error Detection and Correction
- 21.5 mm
- 27.72 mm
- Foxton Power Controller

Dual Cores
Core Design Changes

- Temporal Multi-threading
  - Very low cost throughput performance boost
- New level of cache – 6 cycle 1MB L2I
  - Addresses largest CPI component for transaction processing (Instruction misses)
  - Frees 256KB L2D to be dedicated for data
- ECC add in L2T tags, and parity added to L1I TLB
- Parity in FP and Integer register files
- Second Shift/merge unit for encryption performance
- Power reduction with support for dynamic frequency and voltage
Temporal Multi-Threading

Overlap memory latency stall of Thread 1 with execution of Thread 2

15 cycle penalty

2% core area impact and 0% cycle time impact

Stalled Cycles

15-35% Performance Gain

- SpecInt CPI
- SpecFP CPI
- TPC-C Single Thread CPI
- TPC-C Dual-thread CPI
Multi-Threading Implementation

- 15% register file growth with TMT using mux scheme
  - See Paper 20.5
- Key architectural state threaded, rest is flushed on a switch
  - 2X latch area for 1.3% of total latches

```
<table>
<thead>
<tr>
<th>12 read ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 write ports</td>
</tr>
<tr>
<td>Thread sel</td>
</tr>
<tr>
<td>Thread 0</td>
</tr>
<tr>
<td>Thread 1</td>
</tr>
<tr>
<td>12 read ports</td>
</tr>
<tr>
<td>10 write ports</td>
</tr>
</tbody>
</table>
```
Core Power Reduction

- To improve power efficiency, we focused the team on reducing power consumption for typical applications
  - Developed a vector based tool to accurately measure switching and leakage power
  - Achieved a 28% overall reduction from the ported core even with the new features.
Multiple Voltage and Frequency Domains

- To ensure minimal latency impact, a high gain resolver is needed to cross clock domains

Failure probability

\[ P = \exp\left(-\frac{T_w}{T_r}\right) \]

Where \( T_w \) is the time window for resolution, \( T_r \) is the resolver time constant \( \approx 5\text{ps} \)

- Determinism for debug / test provided with a fixed frequency mode (FFM) enabled with dynamic deskew
Multiple Voltage and Frequency Domains

- To ensure reliable crossing of voltage domains, with up to 400mV offsets, a robust level shifter is used
  - \( V_{\text{cache}} \leftrightarrow V_{\text{core}} \)
  - \( V_{\text{core}} \leftrightarrow V_{\text{fixed}} \)
  - \( V_{\text{fixed}} \leftrightarrow V_{\text{tt}} \)

- A tool was developed to identify all domain crossings and check for the presence of level shifters
Power Efficiency Improvements

• A primary design goal of Montecito was the improvement of power efficiency
  – Performance / Watt
  – Lower acquisition and operating costs, reduced form factor, better compute density

• One must measure what is being optimized
  – Temperature measurements do not suffice – an environmentally dependent proxy for power
    ➔ An ammeter is required

• To avoid wasted watts, have the chip instantaneously optimize operating point for variations in {voltage, temperature, workload, silicon processing}
  ➔ Requires self selection of voltage with dynamic frequency in conjunction with temperature and current measurement
Power Management Loop

• For further details, see paper 16.7

  100s of μs

  Thermal Sensor

  Controller

  Supply VREF

  Voltage Sensor

  Voltage to Freq. Converter

  Clock

  100s of ps

• Key enablers:
  – High accuracy ammeter
  – Dynamic voltage control
  – Fast frequency synthesis as a function of Vcc
  – Accurate thermal measurement
Power Consumption Contour

Optimization point is for typical integer applications which have .6X the switching power of the worst case ➔ Amdahl’s law

Manufacturing test is accomplished by observing the self measured power, and the self-generated frequency for typical code at the power limit
Per-Part Self-Optimization

Power Distribution at Fixed V/F

Power Range with Foxton

Resulting Vdd Distribution
The clock system generates frequency as a function of measured voltage (papers 16.1 & 16.2)
<table>
<thead>
<tr>
<th>Feature</th>
<th>Frequency Gain @ 100W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage to application power vs. max power</td>
<td>12%</td>
</tr>
<tr>
<td>Adapt Vcc to optimal value for each part</td>
<td>5%</td>
</tr>
<tr>
<td>Manage junction temperature to the minimum possible</td>
<td>3%</td>
</tr>
<tr>
<td>Adapt frequency to Vcc to operate at frequency($V_{\text{avg}}$) vs. frequency($V_{\text{min}}$)</td>
<td>7%</td>
</tr>
<tr>
<td>Optimize circuits for low switching and low leakage</td>
<td>11%</td>
</tr>
<tr>
<td>Cache power optimization (low V, asynch. etc.)</td>
<td>5%</td>
</tr>
</tbody>
</table>

For Power, since $P \sim F^3$:

$$P = \frac{P_{\text{orig}}}{(1+.12+.05+.03+.07+.11+.05)^3} = \frac{P_{\text{orig}}}{2.92}$$

This improvement is for a *typical* application.
High power floating point code may see up to a 10% reduction in frequency.
**Shmoo**

<table>
<thead>
<tr>
<th>FREQ</th>
<th>VCORE 1</th>
<th>VCORE 1.05</th>
<th>VCORE 1.1</th>
<th>VCORE 1.15</th>
<th>VCORE 1.2</th>
<th>VCORE 1.25</th>
<th>VCORE 1.3</th>
<th>VCORE 1.35</th>
<th>VCORE 1.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2200</td>
<td>H-FW</td>
<td>H-FW</td>
<td>H-FW</td>
<td>MCA-0</td>
<td>MCA-0</td>
<td>HANG</td>
<td>110W</td>
<td></td>
<td>2200</td>
</tr>
<tr>
<td>2100</td>
<td>H-FW</td>
<td>H-FW</td>
<td>H-FW 0-86165392</td>
<td>95W</td>
<td>104W</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2000</td>
<td>H-FW</td>
<td>H-FW</td>
<td>MCA-0</td>
<td>H-FW</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1900</td>
<td>H-FW</td>
<td>0-86165392</td>
<td>0-86165400</td>
<td>71W</td>
<td>77W</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1800</td>
<td>MCA-0</td>
<td></td>
<td></td>
<td>63W</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1700</td>
<td>57W</td>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Power Consumption is for all 4 voltage domains at ~40C Tj**
Summary

• Design builds on Itanium’s performance strengths:
  – Improves leadership cache hierarchy
  – Improves instruction throughput further with dual-core, multi-threading and new execution units

• Bolsters reliability
  – Parity in register files, improved ECC
  – Seamless adaptation to power or thermal overages

• Achieves leadership performance / Watt
  – Breakthrough power measurement and management technology provides flexibility and efficiency