The J&T processor

- Overview of the Architecture
- Technological details
- Present status
Overview of the J&T Architecture

J&T is the basic (only) building block of the apeNEXT processor. It performs several (all needed) tasks:

- (Mostly) floating point number crunching
- Program flow control
- Memory addressing
- Node-to-node communications (6 + 1 links)
- Slow control interface (mostly for debugging purposes)
Overview of the J&T Architecture

All functions required by the architecture are handled by J&T...
Overview of the J&T Architecture

J&T is a technology improved replacement for the APEmille chipset (3 elements). Project goals:

- Peak floating point performance of about 1.6 Gflops (IEEE compliant double precision)
- Integer arithmetic performance of about 400 Mips
- Link bandwidth of about 200 Mbyte/sec each (full duplex)
- Support for current generation DDR memory
- Memory bandwidth of 3.2 Gbyte/sec (400 Mword/sec)
Overview of the J&T Architecture

Changes w.r.t APEmille are dictated by technology constraints:
- Common program/data memory => Program cache/fifo
- Large memory latency => Data prefetch queues
- Hi clock frequency => Independent flow control
- Hi clock frequency => Self synchronizing nodes
- Hi clock frequency => Self-clocked link
- Memory bandwidth of 3.2 Gbyte/sec (400 Mword/sec)
Overview of the J&T Architecture

Selected highlights: the Memory queue

- Local prefetch queue (replace cache)
- Remote prefetch queue
- Automatic reordering of arriving packets.
Overview of the J&T Architecture

The processor design is based on:

- VHDL description of the system (~55000 VHDL lines)
- Synopsys based synthesis on a CMOS technology
- Memory macrocells
- Synopsys based placement (Physical compiler)
- Cadence based routing (SE) and back-annotation
- Mostly functional test-vectors
- Limited use of TetraMax
Overview of the J&T Architecture

Some figures concerning the physical implementation:

- 0.18 micron/5metal(Al) CMOS technology by Atmel
- 450 signal pins
- 600 total pins (Bga package)
- LVDS (differential) signalling for communication links
- SSTL signalling for DDR memory interface
- 16.0 x 16.0 mmsq (heavily pad limited)
- 5.0 Watt estimated power at 200 Mhz
Some more figures concerning the physical implementation:

- Approx. 20 mmsq memory macrocells
- Approx. 520K equivalent gates random logic
  - Number crunching: 160K
  - Register file: 240K
  - Glue logic: 120K
- 450 signal pins
- 600 total pins (Bga package)
- 16.0 x 16.0 mmsq (heavily pad limited)
- 5.0 Watt estimated power at 200 Mhz
Placement problems

Power Distribution was grossly underestimated
Floorplanning ....
Design completed around July 20th, 2003
Protos delivered by Sankta Klaus
Functional tests started around January 10th
'large' application cores running after just a few days
Crossing fingers.... everything logically OK
Have to measure actual operating frequency (< 200 Mhz)
Photo Gallery: proto #1
Photo Gallery: apeNEXT module
Photo Gallery: testing # 1
Photo Gallery: the 'eyes' of the LVDS link
Photo Gallery: our competitors
A list of errors (with hindsight...)

Should we restart the project today from scratch (which we will NOT do), we would:

- Reduce glue-logic complexity (even more focus)
- Reduce the size of the fastest clock partition
- Start placement at an earlier step
- Select a different silicon foundry (confidential)
The arithmetic unit

Performance Estimates (clock cycle of 5 ns):

<table>
<thead>
<tr>
<th>Type</th>
<th>Performance</th>
<th>Operands in RF</th>
<th>Latency (clock cycles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex</td>
<td>1600Mflops</td>
<td>256</td>
<td>10</td>
</tr>
<tr>
<td>Real</td>
<td>400Mflops</td>
<td>512</td>
<td>7</td>
</tr>
<tr>
<td>Real. vect</td>
<td>800Mflops</td>
<td>256</td>
<td>7</td>
</tr>
<tr>
<td>Integer</td>
<td>200Mips</td>
<td>512</td>
<td>4</td>
</tr>
<tr>
<td>Int. vect</td>
<td>400Mips</td>
<td>256</td>
<td>4</td>
</tr>
</tbody>
</table>