The Effect and Technique of System Coherence in ARM Multicore Technology

John Goodacre
Senior Program Manager
ARM Processor Division
Cambridge, UK

Cortex™-A9 Microarchitecture (single core variant)
Addressing system performance

- System performance is not only defined by the processor
  - Speed/width and latency of DRAM and processor system interface
  - Connectivity and efficiency interacting with system components
    - Offchip Bridged components
    - On chip system components
    - Tightly coupled components

- Important to consider system design, especially interaction with other system components such as DMA and accelerators
Traditional Accelerator SoC Integration

- Analysis of traditional SoC accelerator SoC integration
  - Inefficient usage of CPU cache
  - Significant performance and power implications from data movement
  - High signalling latencies due to mailbox access and interrupt latencies

Enhanced Accelerator SoC Integration

- ARM MPCore: Accelerator Coherence Port (ACP)
  - Simplified software and reduces cache flush overheads
  - Accelerators gain access to CPU cache hierarchy, increasing system performance and reducing overall power
  - Uses AMBA® 3 AXI™ technology for compatibility with standard un-cached peripherals and accelerators
IPSEC Acceleration Using I/O Coherency

ACP provides WAN sub-system with IPSEC

1. WAN port receives packets and allocates packet into L2 coherently
2. Core looks at header, decides it is encrypted and interrupts IPSEC block
3. IPSEC block processes the packet and modifies coherent memory
4. ARM core performs NAT/QoS/Firewall and sends packet through LAN port

ACP - Access to Shared Caches

- Example: CRC engine for TCP packet forwarding on 64 byte packet
- Using typical system latency, ignoring common processing overhead
- Assumed writes are fully buffered

<table>
<thead>
<tr>
<th>Algorithm Stage</th>
<th>Approximate Cycle Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design style</td>
<td>Traditional shared memory with mailbox communication</td>
</tr>
<tr>
<td>Packet received and processed by CPU</td>
<td>0</td>
</tr>
<tr>
<td>Flush cache to make data visible to accelerator</td>
<td>20</td>
</tr>
<tr>
<td>Accelerator notified of data availability</td>
<td>&gt; 4</td>
</tr>
<tr>
<td>Accelerator Reads data from cache line</td>
<td>Typical - 120 [read data from off-chip]</td>
</tr>
<tr>
<td>Accelerator Write data (assuming buffered)</td>
<td>8</td>
</tr>
<tr>
<td>Processor reads processes cache line</td>
<td>Typical - 120</td>
</tr>
<tr>
<td><strong>Total latency overhead</strong></td>
<td>~272 cycles</td>
</tr>
</tbody>
</table>

ACP solution is appropriate for cycle-offload accelerators executing in 100’s of cycles with cache resident workloads. For example in low latency situations required by audio echo cancelation.
Summary

- ARM Cortex-A9 has now exposed the MPCore coherence technology to the wider SoC
  - Interface is known as “Accelerator Coherence Port (ACP)”
  - We’re hearing about ~25% reduction in memory transactions due to reduction in cache flushing

- Software no longer needs to be concerned with cache flush, which can be particularly troublesome on a multicore

- First devices expected be in sample around end of this year