An Implementation of Cache–Coherence for the Nios II Soft–core Processor

Yan Bao
Mats Brorsson
FPGA has been broadly used in system design area

Soft-core programmable processors mapped onto FPGA can be considered as equivalents to a microcontroller.

Nios II soft-core processors are designed for uniprocessor system, not for multiprocessor system
Nios II Processor System

- Generated automatically by SOPC Builder
- Avalon Switch Fabric is a point-to-point interconnection
- Snooping cannot be used
Nios II Processor Core

- Nios II core is a 32-bit processor
- The data cache is a direct mapped write-back cache
- Caches have their own Avalon Memory Mapped (Avalon-MM) master ports connected with slave peripherals
DE2 Development and Education Board

- An Altera Cyclone II 2C35 FPGA device
- USB Blaster (on board) for programming and user API control
- 512Kbyte SRAM
- 8Mbtye SDRAM
- LEDs, LCD and 7-segment displays
- Switches and buttons
Architecture of the original system

- Generated by SOPC Builder
- Two cores system
- SDRAM is the shared memory
- Do not support hardware cache coherency solutions
Hardware cache coherency interface
States in data cache: *Modified, Shared* and *Invalid*

States in directory: *Exclusive, Shared, Uncached*

Directory only contains valid cache lines’ information, not the whole SDRAM. Because of resource limit on DE2 board.

Directory Byte Data field:

<table>
<thead>
<tr>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>TAG</td>
<td>State</td>
<td>Presence bits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Protocols of a load miss in data cache

- CPU0
  - Instruction Cache
  - Data Cache
  - DCC_M
  - DCC_S
  - Avalon Switch Fabric

- CPU1
  - Instruction Cache
  - Data Cache
  - DCC_M
  - DCC_S

- SDRAM Controller
  - Directory Controller

- Read
- Shared
- Writeback
- Exclusive
Protocols of a load miss in data cache

<table>
<thead>
<tr>
<th>TAG State Presence bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 ... 3 2 1 0</td>
</tr>
</tbody>
</table>

Invalid

Exclusive, another tag

Writeback and invalidate

Exclusive
Protocols of a store miss in data cache

Invalid

Invalidate

Permission
Protocol of Busy State in Directory Controller

- To maintain write atomicity, only one write operation is permitted at the same time.
- Before the permitted request completes, directory controller is at its busy state.
- ’S’ port will always be blocked.
- ’Dr_S’ port can receive write requests, but requests will always be refused.
Quartus II is the software used as testing tool.
The test of the system includes two parts: Simulation and Debug
Simulation is based on software and debug is based on hardware.
Single processor read and write test
Multi-processor read and write test
Program tests in the future
# Implementation Aspects

Key number differences between the original design and our cache coherent design:

<table>
<thead>
<tr>
<th></th>
<th>Original design</th>
<th>New design</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total logic elements</strong></td>
<td>5,642</td>
<td>7,282</td>
</tr>
<tr>
<td><strong>Memory bits used</strong></td>
<td>230,656</td>
<td>243,072</td>
</tr>
<tr>
<td><strong>Max clock frequency</strong></td>
<td>187.34 MHz</td>
<td>72.35 MHz</td>
</tr>
</tbody>
</table>

- Memory overhead is negligible
- The Max clock frequency decreased much, since critical path is longer.
- Because before each read or write operation get access to SDRAM, the directory controller holds it, and checks and changes the information in directory.
Compared to a dual-core design without cache coherence, our design has negligible overhead in memory bits and logic elements.

The design is extendable to the system with more than two cores.

Our future work will be to implement it on a large FPGA board and add more functional performances.