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

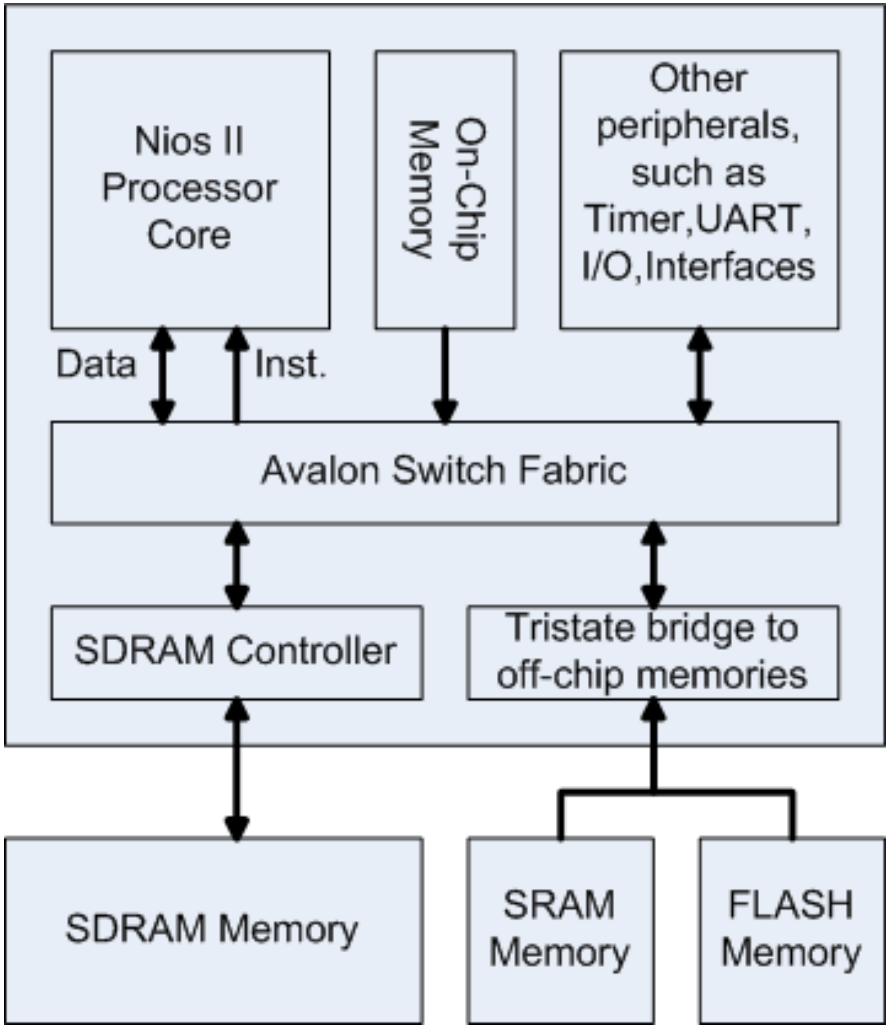
Yan Bao
Mats Brorsson

# Introduction

- 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

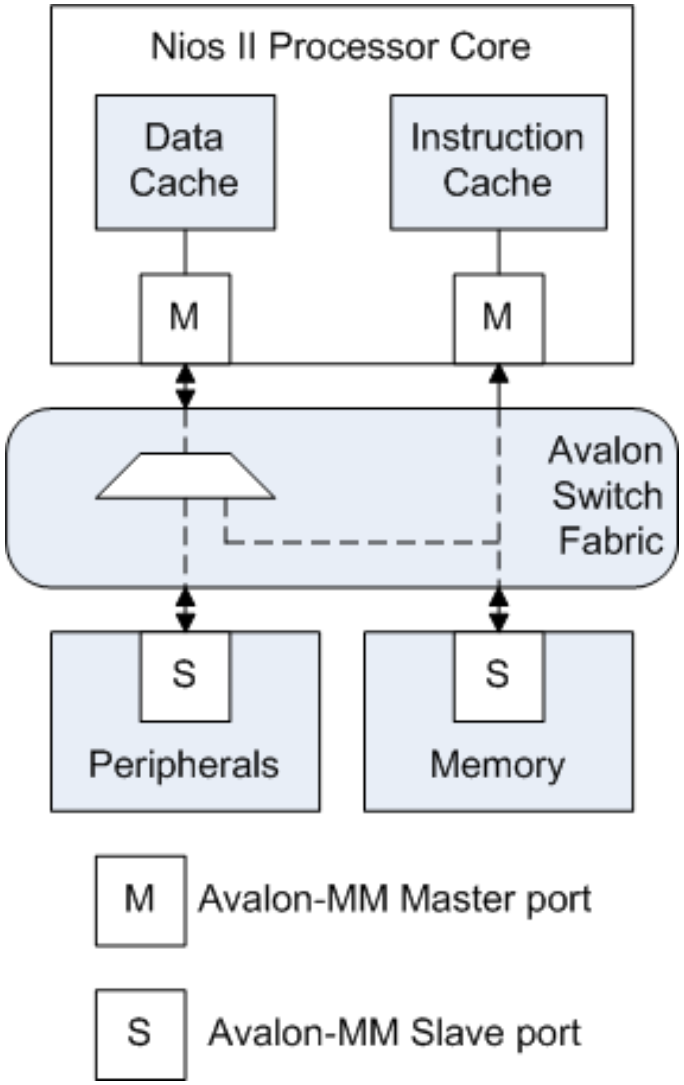
# Nios II Processor System

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



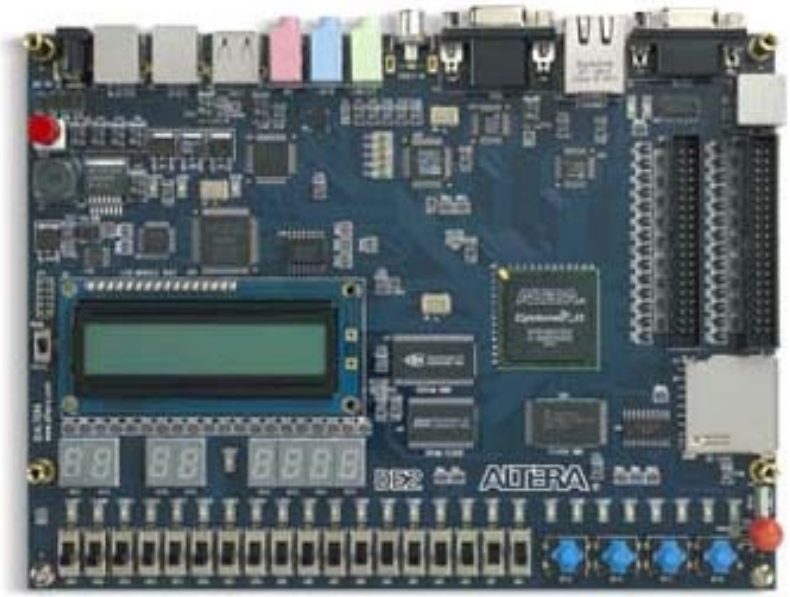
# Nios II Processor Core

- 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



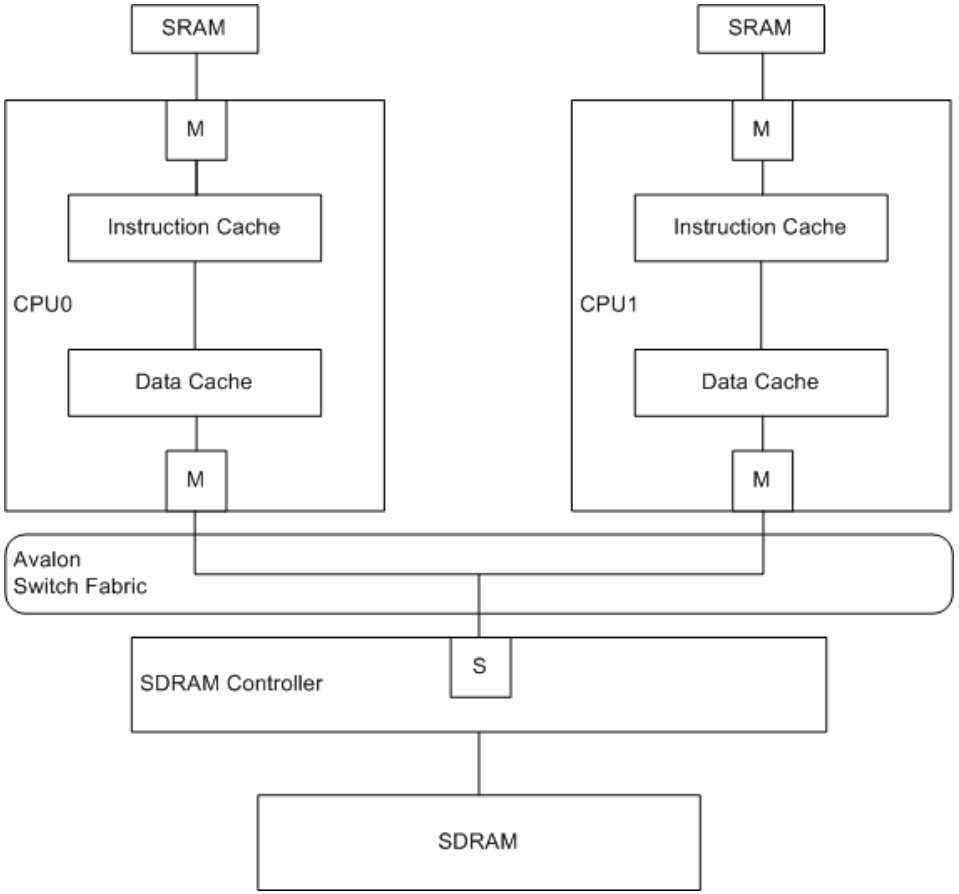
# DE2 Development and Education Board

- 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

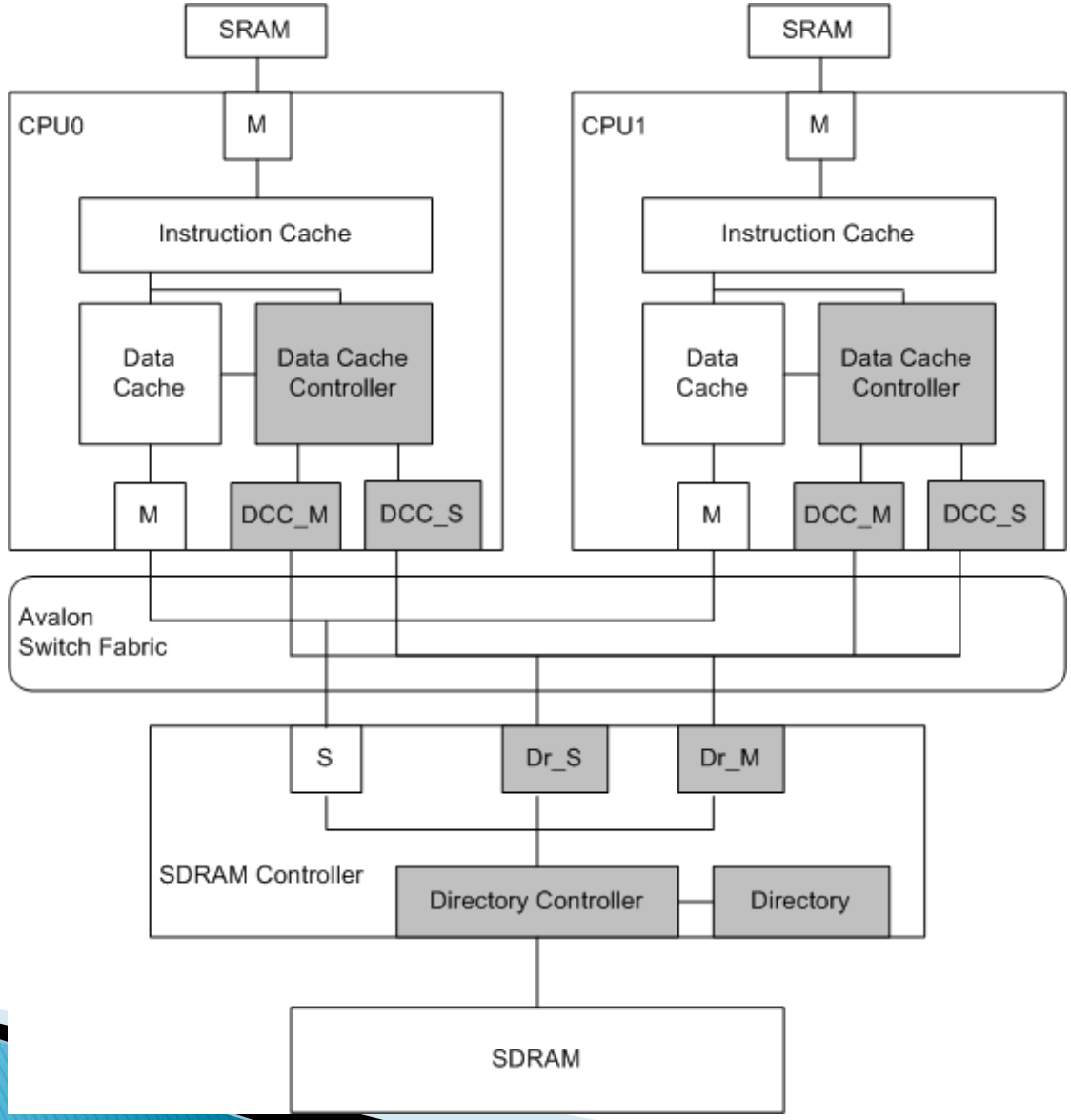


# Architecture of the original system

- Generated by SOPC Builder
- Two cores system
- SDRAM is the shared memory
- Do not support hardware cache coherency solutions



# Hardware cache coherency interface

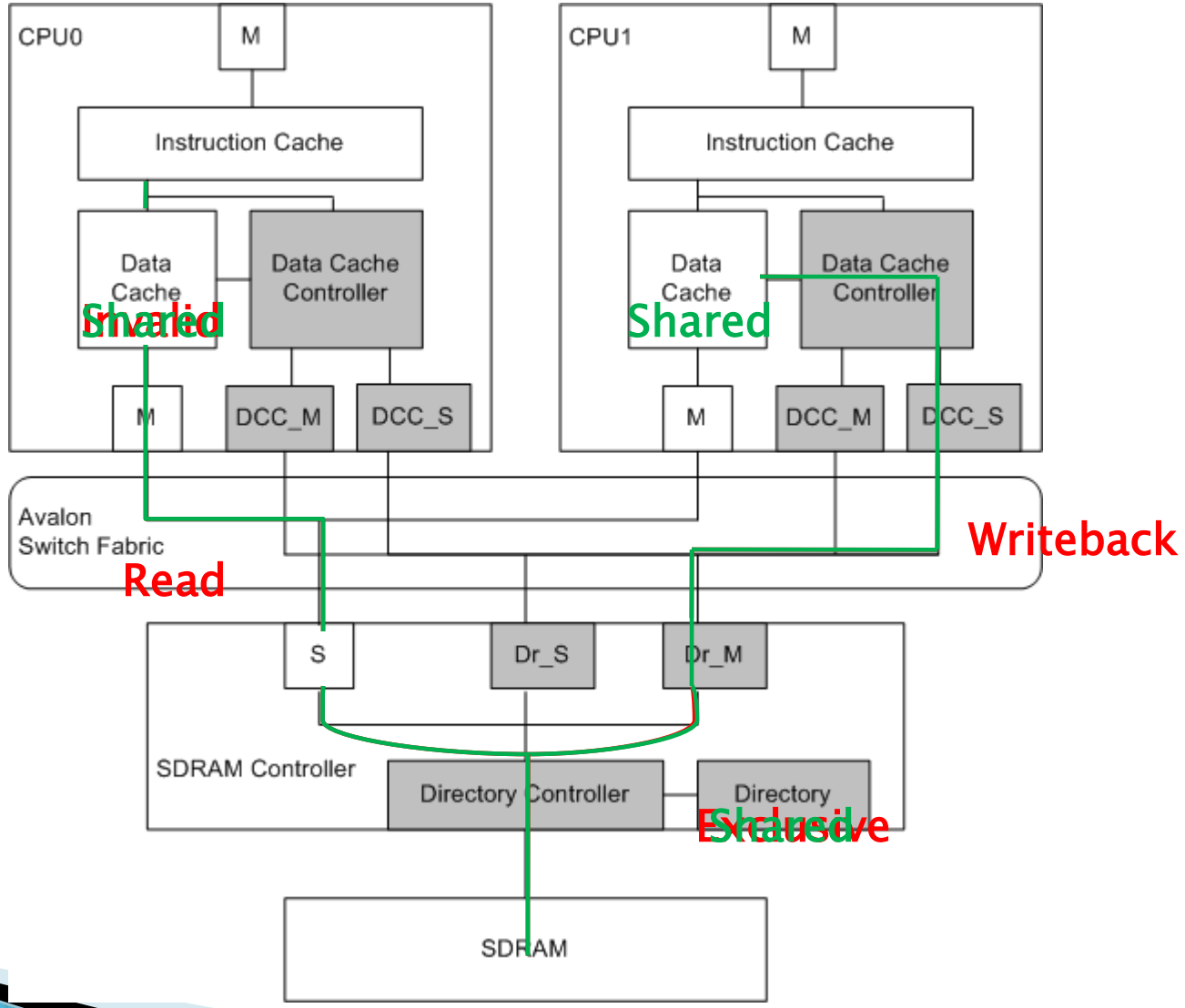


# States in cache and directory

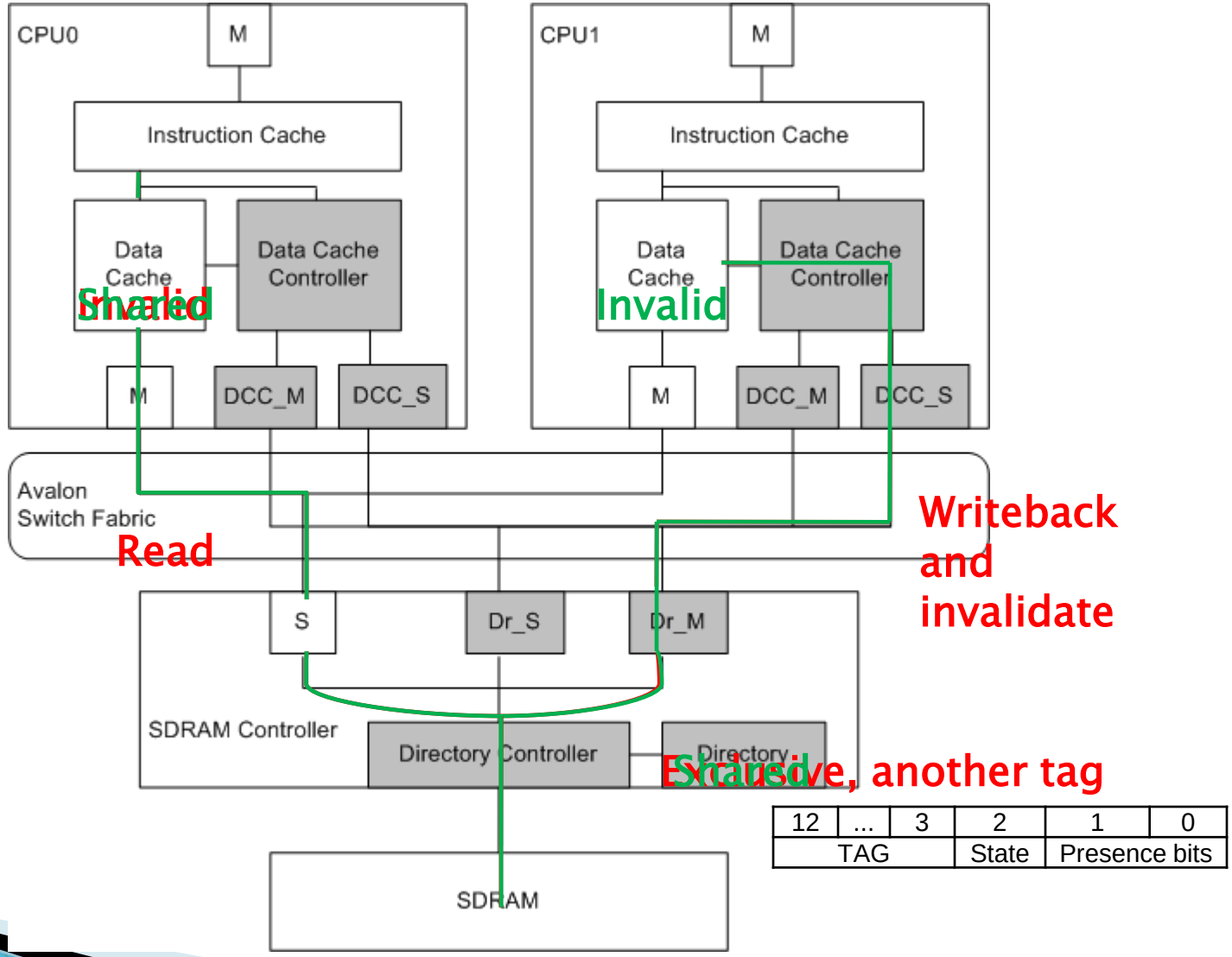
- States in data cache: *Modified*, *Shared* and *Invalid*
- States in directory: *Exclusive, Shared, Uncached*
- Directory only contains valid cache lines' information, not the whole SDRAM. Because of resource limit on DE2 board.
- Directory Byte Data field:

| 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TAG | | | | | | | | | | State | Presence bits | |

# Protocols of a load miss in data cache

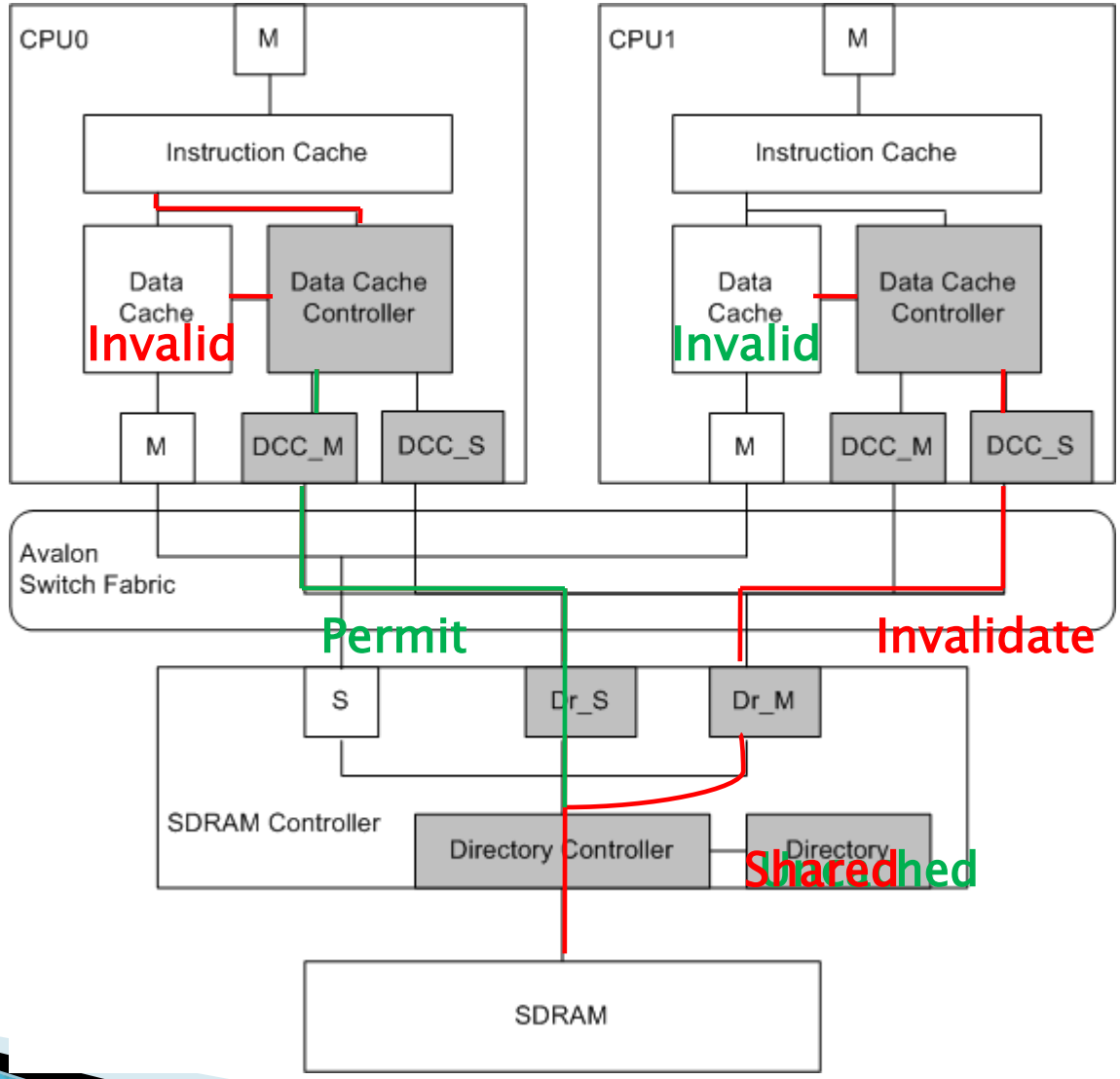


# Protocols of a load miss in data cache



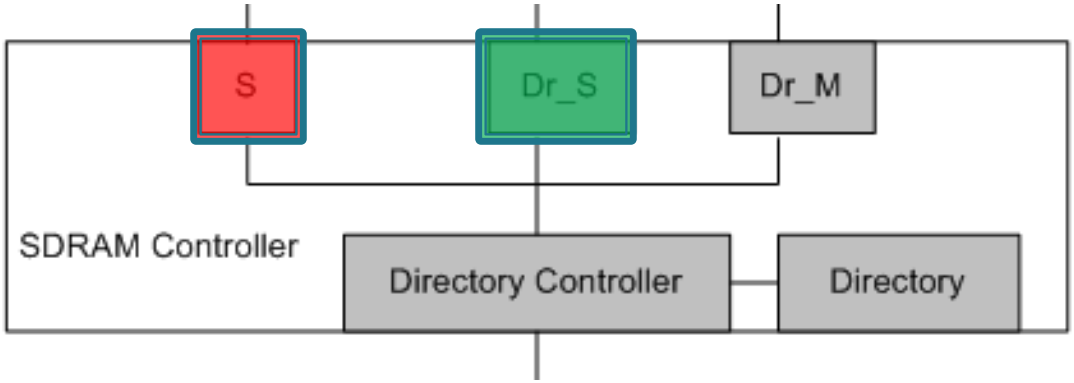

| 12 | ... | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| TAG | | | State | Presence bits | |

# Protocols of a store miss in data cache



# 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



# Test

- 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:

| | Original design | New design |
|---|---|---|
| Total logic elements | 5 642 | 7 282 |
| Memory bits used | 230 656 | 243 072 |
| Max clock frequency | 187.34 MHz | 72.35 MHz |

- 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.

# Conclusions and Future Work

- 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.