BUG REPORT FOR PHASE – 4

Submitted by Amarnath Mahadevuni (UIN: 225003187) , Abhishek Gupta (UIN: 324006634)

IMPORTANT: The term ‘ Core 0’ denotes the Processor with P1\_DL/P1\_IL cache and so on for others.

Bug Number : 1

Line Number : 284 in cache\_block\_0.v

Bug Type : syntax

Test Number : N/A. Syntax checking by just compiling 'cache\_multi\_config\_1.v' with nc-verilog

Bug Description : The equality comparison operator must be '==' and not '=' .

Fix Implemented : Change '=' to '==' at line 284 in cache\_block\_0.v

Bug Number : 2

Line Number : 569 in cache\_block\_0.v

Bug Type : syntax

Test Number : N/A. Syntax checking by just compiling 'cache\_multi\_config\_1.v' with nc-verilog

Bug Description : Net used as an lvalue. CPU\_stall is not declared as a reg.

Fix Implemented : Declare a reg for CPU\_stall as 'reg CPU\_stall'

Bug Number : 3

Line Number : 17 in cache\_def\_0.v

Bug Type : Syntax

Test Number : N/A. Syntax checking by just compiling 'cache\_multi\_config\_1.v' with nc-verilog

Bug Description : Illegal expression in LRU\_size macro definition. Incorrect operator used.

Fix Implemented : Replace '<>' with the correct operator '<<'

Bug Number : 4

Line Number : 69 in cache\_controller\_I\_0.v

Bug Type : Syntax

Test Number : N/A. Syntax checking by just compiling 'cache\_multi\_config\_1.v' with nc-verilog

Bug Description : Macro TAG\_MSB is incorrectly written.

Fix Implemented : Replace TAG\_MSB with `TAG\_MSB

Bug Number : 5

Line Number : 568 in cache\_block\_3.v

Bug Type : Syntax

Test Number : N/A. Syntax checking by just compiling 'cache\_multi\_config\_1.v' with nc-verilog

Bug Description : Cache\_var register size is given incorrectly as CACHE\_DEPTH also Cache\_var array size is incorrectly given as CACHE\_DATA\_SIZE.

Fix Implemented :

define Cache\_var reg as 'reg [`CACHE\_DATA\_SIZE-1 : 0] Cache\_var [0 : `CACHE\_DEPTH-1];'

Bug Number : 6

Line Number : 221, 222 in cache\_block\_3.v

Bug Type : Functionality

Test Number : 1. TopReadMiss .

Bug Description : BusRd and BusRdX assignments are interchanged.

Fix Implemented : assign BusRd to BusRd\_reg and BusRdX to BusRdX\_reg appropriately

Bug Number : 6

Line Number : 771 in cache\_block\_0.v

Bug Type : Functionality

Test Number : 1

Bug Description : Com\_Bus\_Req\_snoop of Proc 1 is asserted while it should remain de-asserted

Bug Number : 7

Line Number : 647 in cache\_block\_1.v

Bug Type : Functionality

Test Number : 15 .P1\_DL, P2\_DL share a block. Core 1 writes to that block in P2\_DL

Bug Description : Core 1 does not send Invalidate signal upon writing to a Shared Block.

Fix Implemented : Add Invalidate logic at 647 in cache\_block\_1.v

Bug Number : 8

Line Number : 470 in cache\_block\_0.v

Bug Type : Functionality

Test Number : 15 .P1\_DL, P2\_DL share a block. Core 1 writes to that block in P2\_DL

Bug Description : Core 0 does not send Invalidate it’s block upon assertion of invalidate signal.

Fix Implemented : In the If condition at line 470 in cache\_block\_0, include ‘|| Invalidate’ to fix this.

Bug Number : 9

Line Number : around 650 in cache\_block\_1.v

Bug Type : Functionality

Test Number : 15 .P1\_DL, P2\_DL share a block. Core 1 writes to that block in P2\_DL

Bug Description : Core 1 Updates MESI State without checking All\_invalidation Signal.

Fix Implemented : Add Invalidate logic at 647 in cache\_block\_1.v

Bug Number : 10

Line Number : 654 in cache\_block\_0.v

Bug Type : Functionality

Test Number : 15

Bug Description : Core 1 does not assert Invalidation\_done signal upon invalidating its shared block.

Fix Implemented : Uncomment the logic for asserting Invalidation\_done.

Bug Number : 11

Line Number : 127 in cache\_multi\_config\_1.v

Bug Type : Functionality

Test Number : 15 . P1\_DL, P2\_DL share a block. Core 0 writes to that block in P1\_DL

Bug Description : All\_Invalidation\_done is not asserted after Invalidation\_done is set by core 1.

Fix Implemented : Use the commented logic and not the workaround at 129 in cache\_multi\_config\_1.v

Bug Number : 12

Line Number : around 685 in cache\_block\_1.v

Bug Type : Functionality

Test Number : 15 .P1\_DL, P2\_DL share a block. Core 1 writes to that block in P2\_DL

Bug Description : Core 1 asserts BusRdX\_reg even though this is a hit in Core 1.

Fix Implemented : Could not figure out the fix.

Bug Number : 13

Line Number : around 272 in cache\_block\_2.v

Bug Type : Functionality

Test Number : 2 . Read Miss on Core 2 in an attempt to create and Exclusive state block.

Bug Description : 3 out of 4 lines are being loaded with Data. Occurs due to wrong assignment of Access\_blk\_proc[\*] . All 4 of them are reversed.

Fix Implemented : Assign the Access\_blk\_proc[\*] correctly.

Bug Number : 14

Line Number : around 102 in cache\_controller\_2.v

Bug Type : Functionality

Test Number : 16. Checking LRU Functionality on Core 2 i.e P3\_DL

Bug Description : wrong lines are being replaced when the set is full. This is due to incorrect assignment of parameters BLK\*\_REPLACEMENT.

Fix Implemented : Assign the BLK\*\_REPLACEMENT correctly as per PLRU in HAS3.0

Bug Number : 15

Line Number : 348 in cache\_block\_I\_1.v

Bug Type : Functionality

Test Number : 17. Checking Read Miss Functionality on Core 1 Instruction Cache i.e P2\_IL

Bug Description : Cache\_var is not written with the data from Data\_Bus\_Com due to incorrect assignment at 348 in cache\_block\_I\_1.v

Fix Implemented : Assign Cache\_var to Data\_Bus\_Com appropriately

Bug Number : 16

Line Number : 331 in cache\_block\_I\_0.v

Bug Type : Functionality

Test Number : 18. Checking Pseudo LRU Functionality on Core 0 Instruction Cache i.e P1\_IL

Bug Description : LRU\_var for a set does not change because Blk\_accessed is hard coded to 0 at 331 in cache\_block\_I\_0.v

Fix Implemented : Assign Blk\_accessed to Blk\_access\_proc;