Bug Number	#3
Bug Location	Main_func_lv1_il.sv **Line 127**
Bug Type	Functional / Design Bug
SV Test Run to uncover the bug	**SEED**: 1 I created a test **(read_miss_eviction_icache)** that performs five reads to the same set in a 4-way set-associative cache. This way, the first four reads fill up the set, and the fifth one triggers a miss with no free block, which should lead to a replacement. Essentially I wanted to check the behavior when a read happens to a fully filled set in a cache.

Checker/Assertion Failed	**<u>Assertions Failed:</u>**
	
	master_access_penable_stable: assert property ((access_state && !pready) |=> penable);
	
	master_access_psel_stable: assert property ((access_state && !pready) |=> $stable(psel));
	
	master_access_paddr_stable: assert property ((access_state && !pready) |=> $stable(paddr));
	
	<u>Error Message:</u>
	
	`    (access_state && !pready) |=> penable);`
	`                               |`
	xmsim: *E,ASRTST (../uvm/cpu_apb_interface.sv,216): (time 2365 NS) Assertion top.inst_cpu_apb_if[1].master_access_penable_stable has failed (2 cycles, starting 2355 NS)
	`    (access_state && !pready) |=> $stable(psel));`
	`                               |`
	xmsim: *E,ASRTST (../uvm/cpu_apb_interface.sv,219): (time 2365 NS) Assertion top.inst_cpu_apb_if[1].master_access_psel_stable has failed (2 cycles, starting 2355 NS)
	`    (access_state && !pready) |=> $stable(paddr));`
	`                               |`
	xmsim: *E,ASRTST (../uvm/cpu_apb_interface.sv,222): (time 2365 NS) Assertion

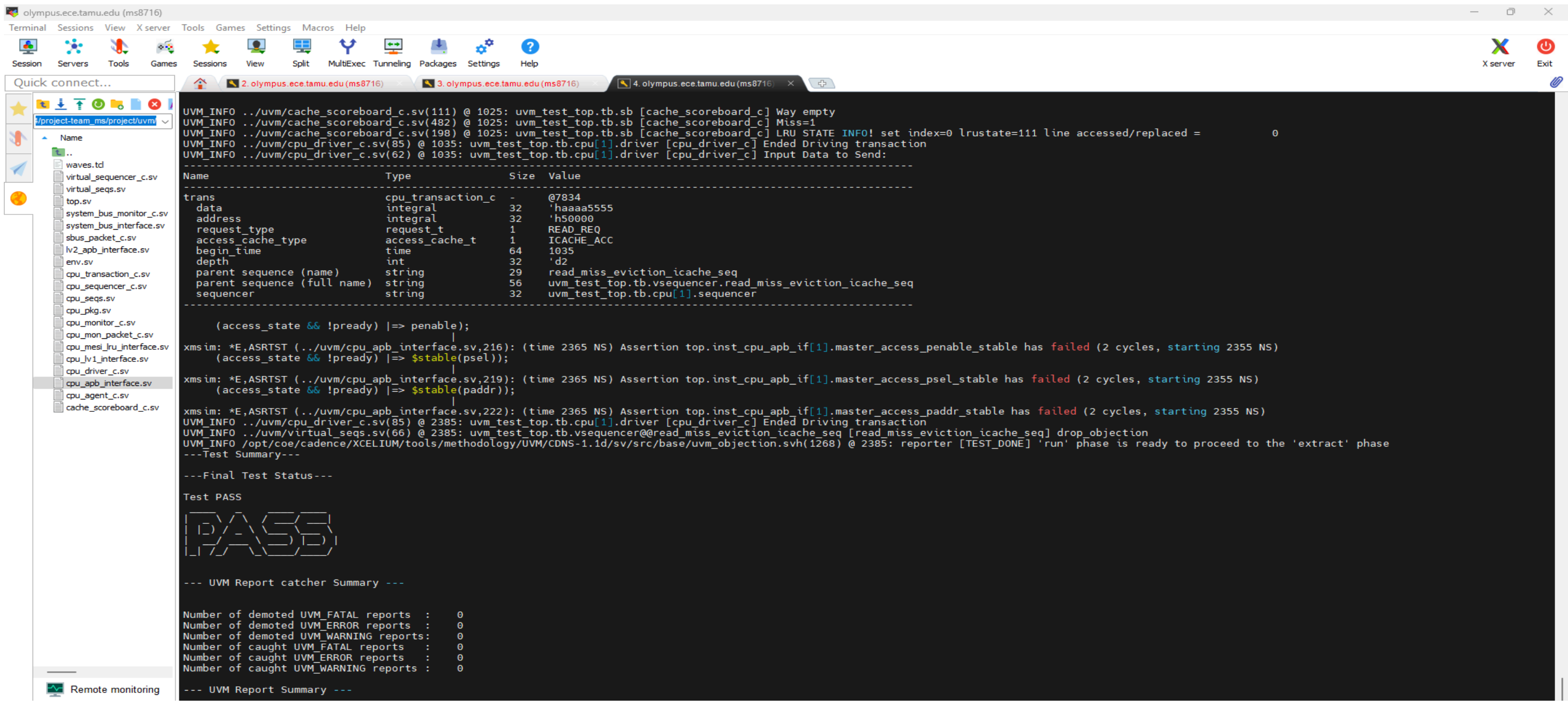
	top.inst_cpu_apb_if[1].master_access_paddr_stable has failed (2 cycles, starting 2355 NS) UVM_INFO ../uvm/cpu_driver_c.sv(85) @ 2385: uvm_test_top.tb.cpu[1].driver [cpu_driver_c] Ended Driving transaction UVM_INFO ../uvm/virtual_seqs.sv(66) @ 2385: uvm_test_top.tb.vsequencer@@read_miss_eviction_icache_seq [read_miss_eviction_icache_seq] drop_objection
Checker Description	These assertions check if the APB control signals (penable, psel, and paddr) remain stable during an ongoing transaction. Specifically, they ensure that once a transfer begins, the address being accessed and the control signals don't change mid-transaction, which is required for APB protocol behavior.
Bug Description/Debug Process	**Bug Description:** When there's a read miss in the lv1 ICACHE and no free block is available, the design sets the MESI state to VALID without doing any eviction or fetching data from L2. So the cache basically claims it has valid data when it doesn't. **Debug Process:** I created a test that performs five reads to the same set of the cache. This way, the first four reads fill up the set, and the fifth one triggers a miss with no free block, which should lead to a replacement. When I ran this test, I didn't see functional errors in the cache logic, but instead got APB protocol assertion failures. Specifically, assertions checking the stability of penable, psel, and paddr. These signals are supposed to remain stable during an APB read transaction, but here they were changing mid-transaction, which usually points to some kind of issue in how the bus is being driven. What caught my attention was that none of the previous ICACHE tests triggered these, only this eviction test did. So I knew the issue had to be tied to this specific case. For the master_access_paddr_stable assertion, I saw that paddr transitioned from a valid address to 'hZZZZZZZZ at the same moment that access_state went low. paddr should remain stable until the access

	fully completes or at least until pready goes high. The fact that it changed immediately as access_state dropped suggests the design may be deasserting the access too early or not holding the signal long enough, leading to unstable bus behavior. So to understand what was going wrong, I checked the behavior of the signals that are supposed to be involved during a read miss with no free block. According to the specification in HAS, the cache should initiate a writeback by asserting bus_lv1_lv2_req_proc, driving the address onto addr_bus_lv1_lv2, placing the data on data_bus_lv1_lv2, raising lv2_wr, and finally waiting for lv2_wr_done from L2 before marking the block as invalid. This frees up space for the incoming data. However, when I looked at the waveform at the time the assertions failed (~2364ns), none of these expected signals were active. lv2_wr remained low, both addr_bus_lv1_lv2 and data_bus_lv1_lv2 were high-Z (zzzz), and lv2_wr_done was never asserted. This clearly indicated me that no replacement or writeback process was initiated. This confirmed that the cache did not handle the miss correctly and led to invalid bus signal, which caused the APB assertions to fail. Looking at the behaviour for read miss in the RTL when the block is full, I saw that the MESI state is going to valid without selecting a victim block or performing any eviction or replacement logic. This gives the illusion that the cacheline contains valid data, even though no data has been fetched from L2 or replaced from an existing block. As a result, when the CPU attempts to read from the cache, the system proceeds with invalid or undefined data, which leads to unstable behavior on the APB interface.

Original Code	`Line 127:`
	`` `CACHE_CURRENT_MESI <= VALID; ``

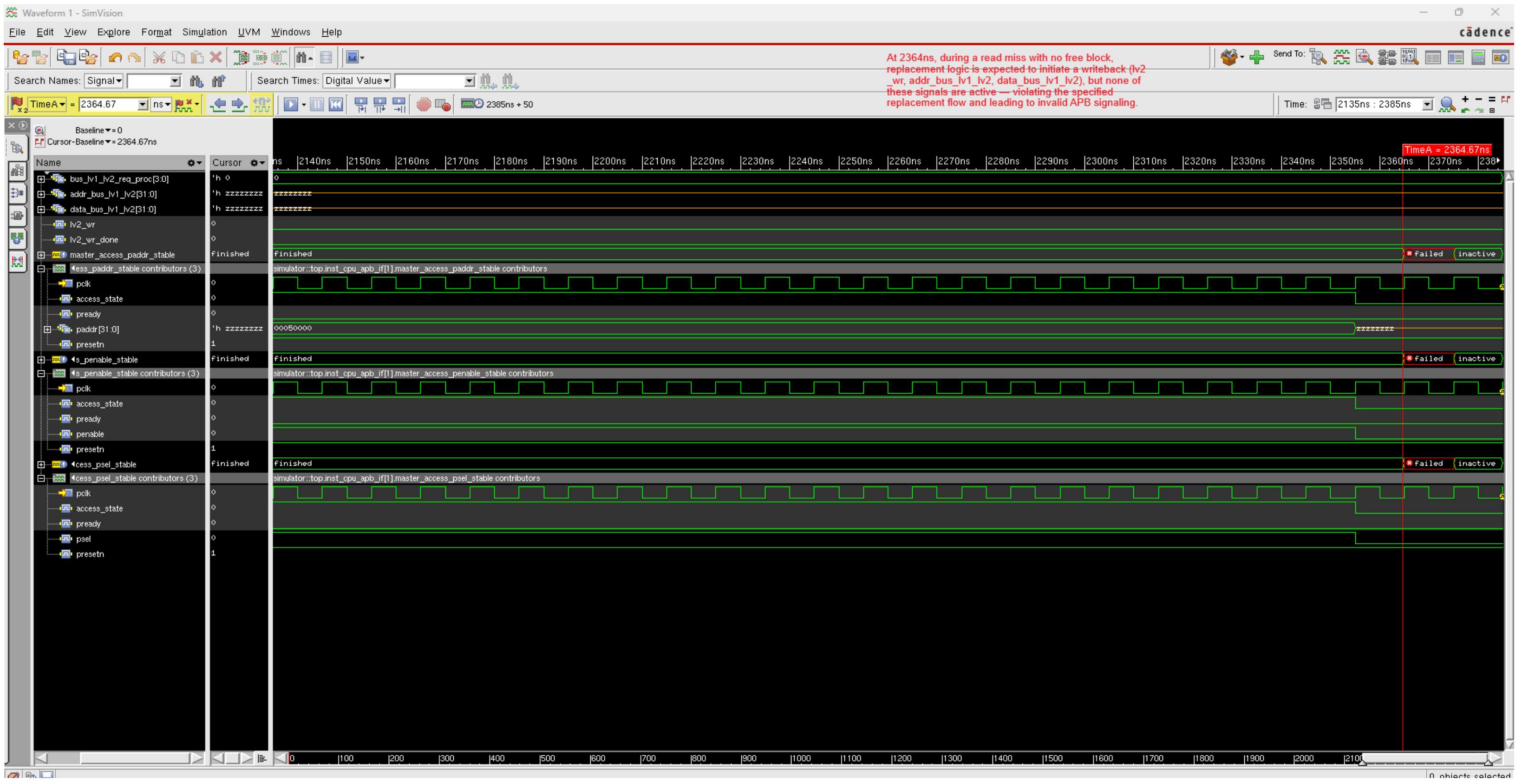
Code after Fix	`Line 127:`
	`` `CACHE_CURRENT_MESI <= INVALID; ``

Error Log for read_miss_eviction_icache test (Assertion failure)



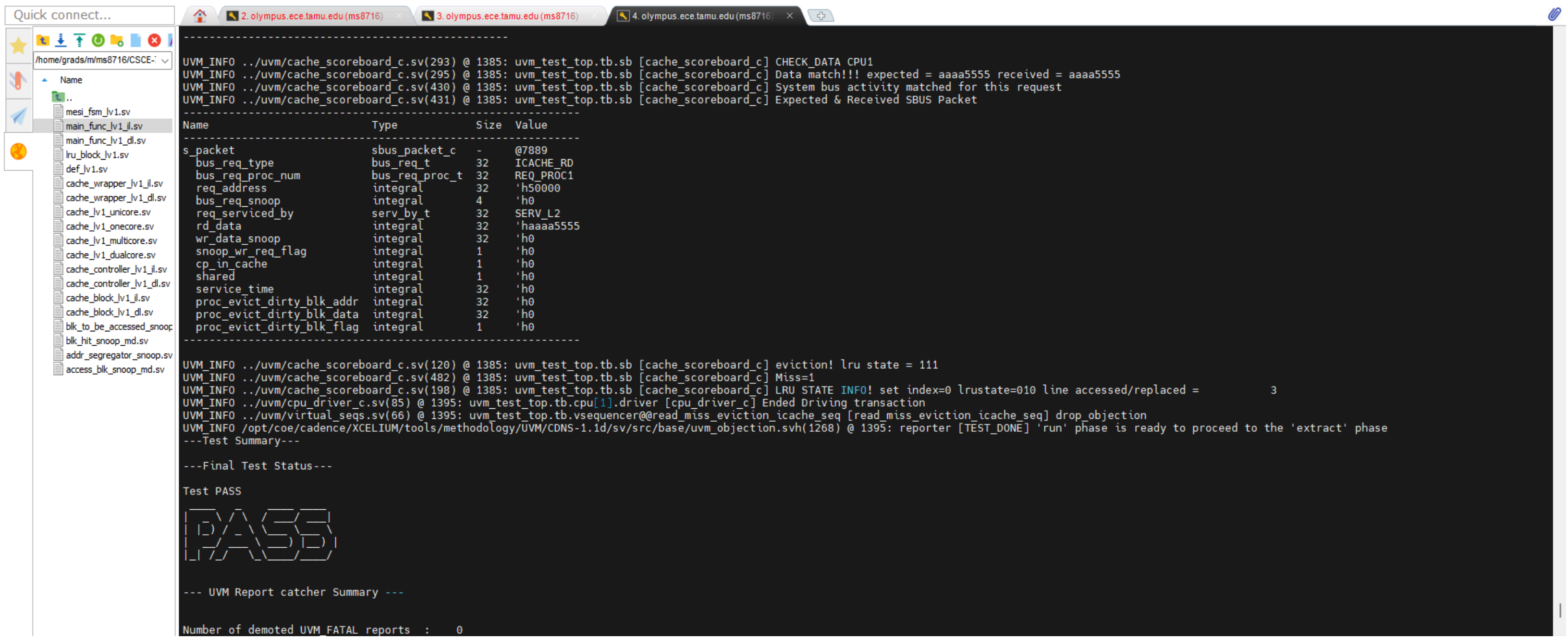
```
UVM_INFO ../uvm/cache_scoreboard_c.sv(111) @ 1025: uvm_test_top.tb.sb [cache_scoreboard_c] Way empty
UVM_INFO ../uvm/cache_scoreboard_c.sv(482) @ 1025: uvm_test_top.tb.sb [cache_scoreboard_c] Miss=1
UVM_INFO ../uvm/cache_scoreboard_c.sv(198) @ 1025: uvm_test_top.tb.sb [cache_scoreboard_c] LRU STATE INFO! set index=0 lrustate=111 line accessed/replaced =          0
UVM_INFO ../uvm/cpu_driver_c.sv(85) @ 1035: uvm_test_top.tb.cpu[1].driver [cpu_driver_c] Ended Driving transaction
UVM_INFO ../uvm/cpu_driver_c.sv(62) @ 1035: uvm_test_top.tb.cpu[1].driver [cpu_driver_c] Input Data to Send:
-------------------------------------------------------------------------------------------------------
Name                          Type                Size  Value
-------------------------------------------------------------------------------------------------------
trans                         cpu_transaction_c   -     @7834
  data                        integral            32    'haaaa5555
  address                     integral            32    'h50000
  request_type                request_t           1     READ_REQ
  access_cache_type           access_cache_t      1     ICACHE_ACC
  begin_time                  time                64    1035
  depth                       int                 32    'd2
  parent sequence (name)      string              29    read_miss_eviction_icache_seq
  parent sequence (full name) string              56    uvm_test_top.tb.vsequencer.read_miss_eviction_icache_seq
  sequencer                   string              32    uvm_test_top.tb.cpu[1].sequencer
-------------------------------------------------------------------------------------------------------

     (access_state && !pready) |=> penable);
                                    |
xmsim: *E,ASRTST (../uvm/cpu_apb_interface.sv,216): (time 2365 NS) Assertion top.inst_cpu_apb_if[1].master_access_penable_stable has failed (2 cycles, starting 2355 NS)
     (access_state && !pready) |=> $stable(psel));
                                    |
xmsim: *E,ASRTST (../uvm/cpu_apb_interface.sv,219): (time 2365 NS) Assertion top.inst_cpu_apb_if[1].master_access_psel_stable has failed (2 cycles, starting 2355 NS)
     (access_state && !pready) |=> $stable(paddr));
                                    |
xmsim: *E,ASRTST (../uvm/cpu_apb_interface.sv,222): (time 2365 NS) Assertion top.inst_cpu_apb_if[1].master_access_paddr_stable has failed (2 cycles, starting 2355 NS)
UVM_INFO ../uvm/cpu_driver_c.sv(85) @ 2385: uvm_test_top.tb.cpu[1].driver [cpu_driver_c] Ended Driving transaction
UVM_INFO ../uvm/virtual_seqs.sv(66) @ 2385: uvm_test_top.tb.vsequencer@@read_miss_eviction_icache_seq [read_miss_eviction_icache_seq] drop_objection
UVM_INFO /opt/coe/cadence/XCELIUM/tools/methodology/UVM/CDNS-1.1d/sv/src/base/uvm_objection.svh(1268) @ 2385: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
---Test Summary---

---Final Test Status---

Test PASS
 ____   _    ____ ____
|  _ \ / \  / ___/ ___|
| |_) / _ \ \___ \___ \
|  __/ ___ \ ___) |__) |
|_| /_/   \_\____/____/


--- UVM Report catcher Summary ---


Number of demoted UVM_FATAL reports  :    0
Number of demoted UVM_ERROR reports  :    0
Number of demoted UVM_WARNING reports:    0
Number of caught UVM_FATAL reports   :    0
Number of caught UVM_ERROR reports   :    0
Number of caught UVM_WARNING reports :    0

--- UVM Report Summary ---
```

Waveform before fix for **read_miss_eviction_icache**



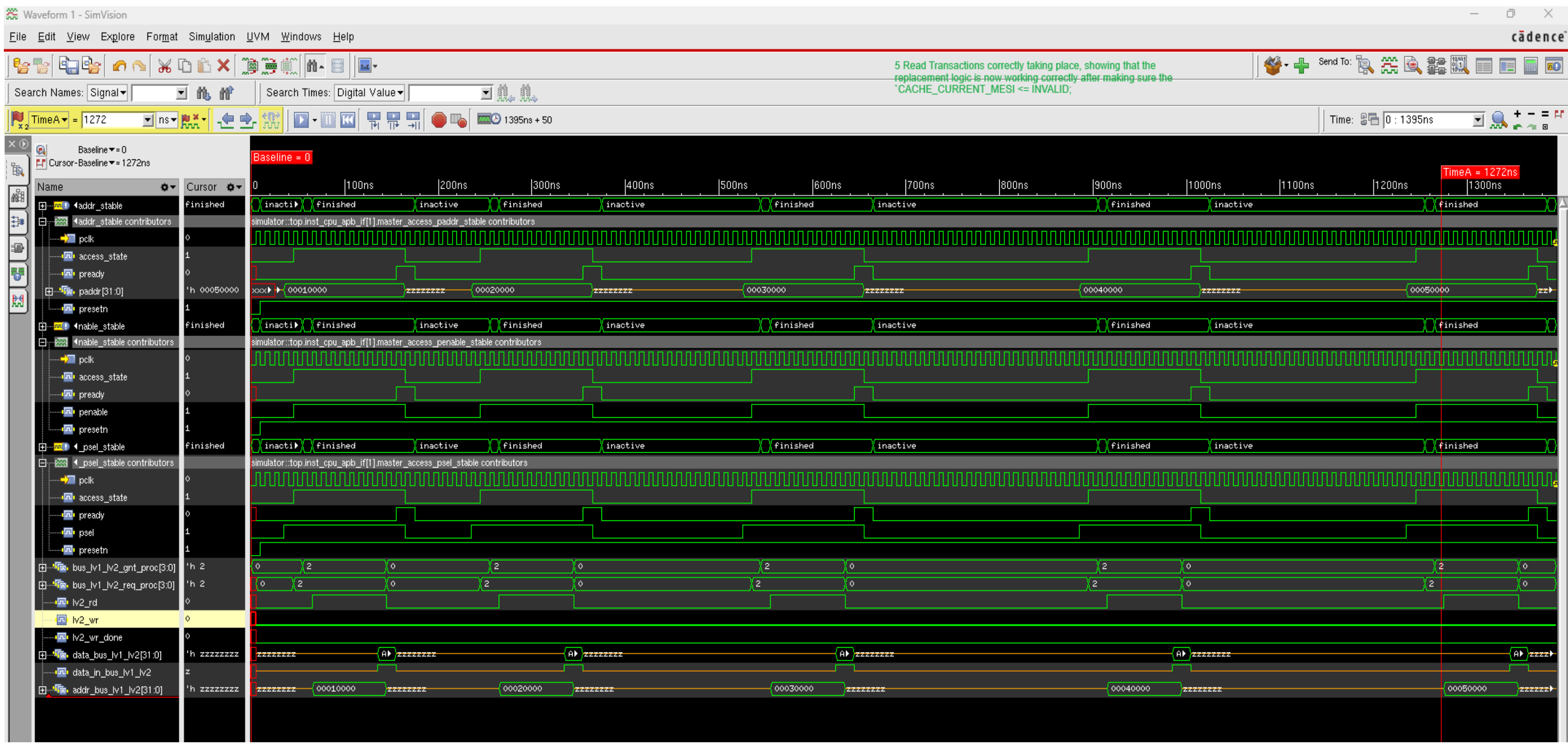
At 2364ns, during a read miss with no free block, replacement logic is expected to initiate a writeback (lv2 _wr, addr_bus_lv1_lv2, data_bus_lv1_lv2), but none of these signals are active — violating the specified replacement flow and leading to invalid APB signaling.

No APB Assertion Failure AFTER FIX



```
--------------------------------------------------

UVM_INFO ../uvm/cache_scoreboard_c.sv(293) @ 1385: uvm_test_top.tb.sb [cache_scoreboard_c] CHECK_DATA CPU1
UVM_INFO ../uvm/cache_scoreboard_c.sv(295) @ 1385: uvm_test_top.tb.sb [cache_scoreboard_c] Data match!!! expected = aaaa5555 received = aaaa5555
UVM_INFO ../uvm/cache_scoreboard_c.sv(430) @ 1385: uvm_test_top.tb.sb [cache_scoreboard_c] System bus activity matched for this request
UVM_INFO ../uvm/cache_scoreboard_c.sv(431) @ 1385: uvm_test_top.tb.sb [cache_scoreboard_c] Expected & Received SBUS Packet
-------------------------------------------------------------
Name                         Type             Size  Value
-------------------------------------------------------------
s_packet                     sbus_packet_c    -     @7889
  bus_req_type               bus_req_t        32    ICACHE_RD
  bus_req_proc_num           bus_req_proc_t   32    REQ_PROC1
  req_address                integral         32    'h50000
  bus_req_snoop              integral         4     'h0
  req_serviced_by            serv_by_t        32    SERV_L2
  rd_data                    integral         32    'haaaa5555
  wr_data_snoop              integral         32    'h0
  snoop_wr_req_flag          integral         1     'h0
  cp_in_cache                integral         1     'h0
  shared                     integral         1     'h0
  service_time               integral         32    'h0
  proc_evict_dirty_blk_addr  integral         32    'h0
  proc_evict_dirty_blk_data  integral         32    'h0
  proc_evict_dirty_blk_flag  integral         1     'h0
-------------------------------------------------------------

UVM_INFO ../uvm/cache_scoreboard_c.sv(120) @ 1385: uvm_test_top.tb.sb [cache_scoreboard_c] eviction! lru state = 111
UVM_INFO ../uvm/cache_scoreboard_c.sv(482) @ 1385: uvm_test_top.tb.sb [cache_scoreboard_c] Miss=1
UVM_INFO ../uvm/cache_scoreboard_c.sv(198) @ 1385: uvm_test_top.tb.sb [cache_scoreboard_c] LRU STATE INFO! set index=0 lrustate=010 line accessed/replaced =           3
UVM_INFO ../uvm/cpu_driver_c.sv(85) @ 1395: uvm_test_top.tb.cpu[1].driver [cpu_driver_c] Ended Driving transaction
UVM_INFO ../uvm/virtual_seqs.sv(66) @ 1395: uvm_test_top.tb.vsequencer@@read_miss_eviction_icache_seq [read_miss_eviction_icache_seq] drop_objection
UVM_INFO /opt/coe/cadence/XCELIUM/tools/methodology/UVM/CDNS-1.1d/sv/src/base/uvm_objection.svh(1268) @ 1395: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
---Test Summary---

---Final Test Status---

Test PASS
 ____   _    ____ ____
|  _ \ / \  / ___/ ___|
| |_) / _ \ \___ \___ \
|  __/ ___ \ ___) |__) |
|_| /_/   \_\____/____/


--- UVM Report catcher Summary ---


Number of demoted UVM_FATAL reports  :    0
```

Waveform after fix for **read_miss_eviction_icache**


5 Read Transactions correctly taking place, showing that the replacement logic is now working correctly after making sure the 'CACHE_CURRENT_MESI <= INVALID;