

# CVSD #hw5 Report

R06943087 葉玲瑜

1. 請指出 path.v 裡有 bug 的地方，該如何修正，解釋原因，以及會使那些 assertion 有 counter example。

(1)

```
assign stop1_o = priority_flag ? ((full && !gnt_i) || !enable_i) : 1;  
assign stop2_o = priority_flag ? ((full && !gnt_i) || !enable_i) : 1;
```

根據 pdf 裡面對系統的說明，stop1\_o 的判斷條件應該是 !priority\_flag，利用此變數來做 master 的切換。

```
assign stop1_o = !priority_flag ? ((full && !gnt_i) || !enable_i) : 1;  
assign stop2_o = priority_flag ? ((full && !gnt_i) || !enable_i) : 1;
```

這個 bug 影響到的 assert 為：

✓ !	Assert	path.inst_vcomp_path.assert_valid1_to_req1_bb
✓ !	Assert	path.inst_vcomp_path.assert_valid2_to_req2_bb
✓	Assert	path.inst_vcomp_path.assert_only_one_slave_request
✓ !	Assert	path.inst_vcomp_path.assert_valid1_to_stop1_bb
✓ !	Assert	path.inst_vcomp_path.assert_valid2_to_stop2_bb
✓ !	Assert	path.inst_vcomp_path.assert_data_bypass_bb
✓	Assert	path.inst_vcomp_path.assert_never_full_empty_wb
✓	Assert	path.inst_vcomp_path.assert_noPushRemainEmpty_wb
✓ !	Assert	path.inst_vcomp_path.assert_stop_when_full_wb
✗	Assert	path.inst_vcomp_path.assert_nonfull_stop_check_wb
✓ !	Assert	path.inst_vcomp_path.assert_emptyData1_wb
✓ !	Assert	path.inst_vcomp_path.assert_emptyData2_wb
✓ !	Assert	path.inst_vcomp_path.assert_emptyDataBypass_wb
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk1.data_integrity
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk2.no_overflow
✓ !	Assert (live)	path.inst_vcomp_path.assert_data1_flow_check_bb
✓ !	Assert (live)	path.inst_vcomp_path.assert_data2_flow_check_bb

(2)

```
assign req1_o = (gnt1_i)? 0 : valid1_o;
```

req1\_o 跟要輸出的 data 是否 valid 有關，所以條件只有 valid1\_o 而沒有 gnt1\_i。

```
assign req1_o = valid1_o;
```

這個 bug 影響到的 assert 為：

✓ !	Assert	path.inst_vcomp_path.assert_emptyData1_wb
✓	Assert	path.inst_vcomp_path.assert_emptyData2_wb
✓	Assert	path.inst_vcomp_path.assert_emptyDataBypass_wb

(3)

```
assign req2_o == valid1_i;
```

req2\_o 跟要輸出的 data 是否 valid 有關，所以條件只有 valid2\_o，與 valid1\_o 無關。

```
assign req2_o = valid2_o;
```

這個 bug 影響到的 assert 為：

✗	Assert	path.inst_vcomp_path.assert_valid2_to_req2_bb
✗	Assert	path.inst_vcomp_path.assert_only_one_slave_request
✓	Assert	path.inst_vcomp_path.assert_valid1_to_stop1_bb
✓	Assert	path.inst_vcomp_path.assert_valid2_to_stop2_bb
✓	Assert	path.inst_vcomp_path.assert_data_bypass_bb
✓	Assert	path.inst_vcomp_path.assert_never_full_empty_wb
✓	Assert	path.inst_vcomp_path.assert_noPushRemainEmpty_wb
✓	Assert	path.inst_vcomp_path.assert_stop_when_full_wb
✓	Assert	path.inst_vcomp_path.assert_nonfull_stop_check_wb
✓	Assert	path.inst_vcomp_path.assert_emptyData1_wb
✓	Assert	path.inst_vcomp_path.assert_emptyData2_wb
✓	Assert	path.inst_vcomp_path.assert_emptyDataBypass_wb
✗	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk1.data_integrity
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk2.no_overflow
✗	Assert (live)	path.inst_vcomp_path.assert_data1_flow_check_bb
✗	Assert (live)	path.inst_vcomp_path.assert_data2_flow_check_bb

2. 請指出 fifo.v 裡有 bug 的地方，該如何修正，解釋原因，以及會使那些 assertion 有 counter example。

(1)

```
wr_ptr <= wr_ptr == FDEPTH-2 ? {ADDR_WIDTH{1'b0}} : wr_ptr + 2;
```

wr\_ptr 有增加的時候應該是要+1，如果達到 wr\_ptr 的上限 FDEPTH-1 的時候則是要-1，但這邊寫成+2, -2所以要改成：

```
wr_ptr <= wr_ptr == FDEPTH-1 ? {ADDR_WIDTH{1'b0}} : wr_ptr + 1;
```

這個 bug 影響到的 assert 為：

✗	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk1.data_integrity
✗	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk2.no_overflow
✗	Assert (live)	path.inst_vcomp_path.assert_data1_flow_check_bb
✗	Assert (live)	path.inst_vcomp_path.assert_data2_flow_check_bb

(2)

```
rd_ptr <= (rd_ptr == FDEPTH - 2) ? {ADDR_WIDTH{1'b0}} : rd_ptr + 1;
```

rd\_ptr 有增加的時候應該是要+1，如果達到 rd\_ptr 的上限 FDEPTH-1 的時候則是要-1，但這邊寫成+2, -2 所以要改成：

```
rd_ptr <= (rd_ptr == FDEPTH - 1) ? {ADDR_WIDTH{1'b0}} : rd_ptr + 1;
```

這個 bug 影響到的 assert 為：

✗	Assert	path.inst_vcomp_path.assert_emptyDataBypass_wb
✗	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk1.data_integrity
✗	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk2.no_overflow
✗	Assert (live)	path.inst_vcomp_path.assert_data1_flow_check_bb
✗	Assert (live)	path.inst_vcomp_path.assert_data2_flow_check_bb

3. 請附上用 12 條 assertion 的 verify (1) 初始有 bug 的 RTL 檔案以及(2)修掉 bug 的 RTL 檔案在 JasperGold 上 prove 的結果(2 張截圖)。

(1)

✓	Assert	path.inst_vcomp_path.assert_never_underflow_bb
✓	Assert	path.inst_vcomp_path.assert_never_overflow_bb
✓	Assert	path.inst_vcomp_path.assert_cross_stop_bb
✓ !	Assert	path.inst_vcomp_path.assert_valid1_to_req1_bb
✓ !	Assert	path.inst_vcomp_path.assert_valid2_to_req2_bb
✓	Assert	path.inst_vcomp_path.assert_only_one_slave_request
✓ !	Assert	path.inst_vcomp_path.assert_valid1_to_stop1_bb
✓ !	Assert	path.inst_vcomp_path.assert_valid2_to_stop2_bb
✓ !	Assert	path.inst_vcomp_path.assert_data_bypass_bb
✓	Assert	path.inst_vcomp_path.assert_never_full_empty_wb
✓	Assert	path.inst_vcomp_path.assert_noPushRemainEmpty_wb
✓ !	Assert	path.inst_vcomp_path.assert_stop_when_full_wb
✗	Assert	path.inst_vcomp_path.assert_nonfull_stop_check_wb
✓ !	Assert	path.inst_vcomp_path.assert_emptyData1_wb
✓ !	Assert	path.inst_vcomp_path.assert_emptyData2_wb
✓ !	Assert	path.inst_vcomp_path.assert_emptyDataBypass_wb
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk1.data_integrity
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk2.no_overflow

✓ !	Assert (live)	path.inst_vcomp_path.assert_data1_flow_check_bb
✓ !	Assert (live)	path.inst_vcomp_path.assert_data2_flow_check_bb
●	Assume	path.inst_vcomp_path.assume_no_valid1_if_stall1
●	Assume	path.inst_vcomp_path.assume_no_valid2_if_stall2
●	Assume	path.inst_vcomp_path.assume_no_gnt1_without_req1
●	Assume	path.inst_vcomp_path.assume_no_gnt2_without_req2
●	Assume	path.inst_vcomp_path.assume_gnt_cannot_rise_together
●	Assume	path.inst_vcomp_path.assume_data1_sample_hold_bb
●	Assume	path.inst_vcomp_path.assume_data2_sample_hold_bb
●	Assume	caselasm1
●	Assume	caselasm2
● !	Assume (live)	path.inst_vcomp_path.assume_arbitration_is_fair1
● !	Assume (live)	path.inst_vcomp_path.assume_arbitration_is_fair2
✗	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_in
✗	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_out
✗	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_in
✗	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_out
✗	Cover (related)	path.inst_vcomp_path.assume_arbitration_is_fair1:precondition1
✗	Cover (related)	path.inst_vcomp_path.assume_arbitration_is_fair2:precondition1
✓	Cover (related)	path.inst_vcomp_path.assume_no_valid1_if_stall1:precondition1
✓	Cover (related)	path.inst_vcomp_path.assume_no_valid2_if_stall2:precondition1
✓	Cover (related)	path.inst_vcomp_path.assume_no_gnt1_without_req1:precondition1
✓	Cover (related)	path.inst_vcomp_path.assume_no_gnt2_without_req2:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_valid1_to_req1_bb:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_valid2_to_req2_bb:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_valid1_to_stop1_bb:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_valid2_to_stop2_bb:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_data1_flow_check_bb:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_data2_flow_check_bb:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_data_bypass_bb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_noPushRemainEmpty_wb:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_stop_when_full_wb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_nonfull_stop_check_wb:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_emptyData1_wb:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_emptyData2_wb:precondition1
✗	Cover (related)	path.inst_vcomp_path.assert_emptyDataBypass_wb:precondition1

(2)

✓	Assert	path.inst_vcomp_path.assert_never_underflow_bb	Hp (1)
✓	Assert	path.inst_vcomp_path.assert_never_overflow_bb	Hp (1)
✓	Assert	path.inst_vcomp_path.assert_cross_stop_bb	PRE (1)
✓	Assert	path.inst_vcomp_path.assert_valid1_to_req1_bb	PRE
✓	Assert	path.inst_vcomp_path.assert_valid2_to_req2_bb	PRE
✓	Assert	path.inst_vcomp_path.assert_only_one_slave_request	PRE (1)
✓	Assert	path.inst_vcomp_path.assert_valid1_to_stop1_bb	PRE (1)
✓	Assert	path.inst_vcomp_path.assert_valid2_to_stop2_bb	PRE (1)
✓	Assert	path.inst_vcomp_path.assert_data_bypass_bb	Hp (1)
✓	Assert	path.inst_vcomp_path.assert_never_full_empty_wb	Hp (2)
✓	Assert	path.inst_vcomp_path.assert_noPushRemainEmpty_wb	Hp (2)
✓	Assert	path.inst_vcomp_path.assert_stop_when_full_wb	PRE (1)
✓	Assert	path.inst_vcomp_path.assert_nonfull_stop_check_wb	PRE (1)
✓	Assert	path.inst_vcomp_path.assert_emptyData1_wb	Hp (1)
✓	Assert	path.inst_vcomp_path.assert_emptyData2_wb	Hp (1)
✓	Assert	path.inst_vcomp_path.assert_emptyDataBypass_wb	M (7)
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk1.data_integrity	R (14)
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk2.no_overflow	Tri (11)

✓	Assert (live)	path.inst_vcomp_path.assert_data1_flow_check_bb
✓	Assert (live)	path.inst_vcomp_path.assert_data2_flow_check_bb
●	Assume	path.inst_vcomp_path.assume_no_valid1_if_stall1
●	Assume	path.inst_vcomp_path.assume_no_valid2_if_stall2
●	Assume	path.inst_vcomp_path.assume_no_gnt1_without_req1
●	Assume	path.inst_vcomp_path.assume_no_gnt2_without_req2
●	Assume	path.inst_vcomp_path.assume_gnt_cannot_rise_together
●	Assume	path.inst_vcomp_path.assume_data1_sample_hold_bb
●	Assume	path.inst_vcomp_path.assume_data2_sample_hold_bb
●	Assume	<a href="#">caselasm1</a>
●	Assume	<a href="#">caselasm2</a>
●	Assume (live)	path.inst_vcomp_path.assume_arbitration_is_fair1
●	Assume (live)	path.inst_vcomp_path.assume_arbitration_is_fair2
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_in
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_out
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_in
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_out
✓	Cover (related)	path.inst_vcomp_path.assume_arbitration_is_fair1:precondition1

✓	Cover (related)	path.inst_vcomp_path.assume_arbitration_is_fair2:precondition1
✓	Cover (related)	path.inst_vcomp_path.assume_no_valid1_if_stall1:precondition1
✓	Cover (related)	path.inst_vcomp_path.assume_no_valid2_if_stall2:precondition1
✓	Cover (related)	path.inst_vcomp_path.assume_no_gnt1_without_req1:precondition1
✓	Cover (related)	path.inst_vcomp_path.assume_no_gnt2_without_req2:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_valid1_to_req1_bb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_valid2_to_req2_bb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_valid1_to_stop1_bb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_valid2_to_stop2_bb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_data1_flow_check_bb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_data2_flow_check_bb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_data_bypass_bb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_noPushRemainEmpty_wb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_stop_when_full_wb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_nonfull_stop_check_wb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_emptyData1_wb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_emptyData2_wb:precondition1
✓	Cover (related)	path.inst_vcomp_path.assert_emptyDataBypass_wb:precondition1

4. 請完整地列出(包含 code)哪幾條 assertion，即使是有 bug 的 RTL code，也會被 proved。

```
// should never see overflow or underflow
assert_never_underflow_bb : assert property (@(posedge clk) disable iff(!rst_n) not (underflow_o));
assert_never_overflow_bb : assert property (@(posedge clk) disable iff(!rst_n) not (overflow_o));

// stop1 and stop2 cannot be low at same time
assert_cross_stop_bb: assert property ( @(posedge clk) disable iff(!rst_n) (stop1_o | stop2_o) );
```

```
//TODO// only one slave (memory) would be requested
assert_only_one_slave_request : assert property ( @(posedge clk) disable iff(!rst_n) !(req1_o && req2_o) );
```

```
// full and empty asserted at the same time would be an error
assert_never_full_empty_wb : assert property (@(posedge clk) disable iff(!rst_n) not (full && empty) );

// if empty and no writes, it must stay empty
assert_noPushRemainEmpty_wb : assert property (@(posedge clk) disable iff(!rst_n) (empty && !write_i) | => empty );
```

5. 請截圖 scoreboard 的六個 property 被 proved 的情形

✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_out
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_in
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_out
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_in
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk2.no_overflow
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk1.data_integrity