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。

修正處 1:

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

改成

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

原因:

否則會有啟動問題，stop1_o、stop2_o 都會是 1，會無法啟動

這個會讓所有的 assertion 都沒有波型可以搜尋，因為電路是不會正常跑的，但會讓一些 assertion 瞎矇混過。

Counter example:

✖	Assert	path.inst_vcomp_path.assert_nonfull_stop_check_wb	L	1	0.0	<embedded>
✖	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_in	PRE	Infinite	0.0	<embedded>
✖	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_out	AB (5)	Infinite	0.0	<embedded>
✖	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_in	PRE	Infinite	0.0	<embedded>
✖	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_out	AB (2)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assume_arbitration_is_fair1:precondition1	AB (3)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assume_arbitration_is_fair2:precondition1	AB (3)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assume_no_gnt1_without_req1:precondition1	AB (2)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assume_no_gnt2_without_req2:precondition1	AB (3)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_valid1_to_req1_bb:precondition1	AB (3)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_valid2_to_req2_bb:precondition1	AB (3)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_valid1_to_stop1_bb:precondition1	PRE (1)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_valid2_to_stop2_bb:precondition1	PRE (1)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_data1_flow_check_bb:precondition1	PRE	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_data2_flow_check_bb:precondition1	PRE	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_data_bypass_bb:precondition1	PRE	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_stop_when_full_wb:precondition1	PRE	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_emptyData1_wb:precondition1	PRE (1)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_emptyData2_wb:precondition1	PRE (1)	Infinite	0.0	<embedded>
✖	Cover (related)	path.inst_vcomp_path.assert_emptyDataBypass_wb:precondition1	PRE (1)	Infinite	0.0	<embedded>

修正處 2:

```
assign req1_o   = (gnt1_i)? 0 : valid1_o;
assign req2_o   = valid1_o;
assign valid1_o = !data_o[IDWIDTH+DWIDTH-1] && ( bypass1 || !empty );
assign valid2_o = data_o[IDWIDTH+DWIDTH-1]  && ( bypass2 || !empty );
```

改成

```
assign req1_o   = !data_o[IDWIDTH+DWIDTH-1] && ( bypass1
|| !empty ) ;
assign req2_o   = data_o[IDWIDTH+DWIDTH-1]  && ( bypass2
|| !empty ) ;
```

```
assign valid1_o = !data_o[IDWIDTH+DWIDTH-1] && ( bypass1 || !empty )
&& gnt1_i;
assign valid2_o = data_o[IDWIDTH+DWIDTH-1]  && ( bypass2 || !empty )
&& gnt2_i;
```

原因:
因為 req_o 不是由 gnt_i 驅動的，是 gnt_i 由 req_o 由驅動
另外當 gnt_i 來之後，valid_o 要起來

Counter example:
會讓下列的 assertion 有反例
fairness: (1) assume_arbitration_is_fair1
(2) assume_arbitration_is_fair2
grant signals: (1) assume_no_gnt1_without_req1
(2)asume_no_gnt2_without_req2
if valid_o is asserted, the request should be asserted in the same cycle (1)assert_valid1_to_req1_bb
(2)assert_valid2_to_req2_bb
only one slave (memory) would be requested
assert_only_one_slave_request

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 。
修正處:

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

改成

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

原因:
這樣 ptr 存取的資料才符合 circular queue，才會是對的

Counter example: 會讓下列的 assertion 有反例
assert_data1_flow_check_bb 、 assert_data2_flow_check_bb
if FIFO is empty, valid is asserted, but no gnt. Then input data should be seen on the next cycle on the output
assert_emptyDataBypass_wb
score_board 的 data_integrity 和 no_overflow 也會沒過

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）初始有 bug 的 RTL 檔案

	Type	Name	Engine
✗	Assert	path.inst_vcomp_path.assert_nonfull_stop_check_wb	B
✗	Cover (related)	path.inst_vcomp_path.assert_data1_flow_check_bb:precondition1	PRE
✗	Cover (related)	path.inst_vcomp_path.assert_data2_flow_check_bb:precondition1	PRE
✗	Cover (related)	path.inst_vcomp_path.assert_data_bypass_bb:precondition1	PRE
✗	Cover (related)	path.inst_vcomp_path.assert_emptyData1_wb:precondition1	PRE
✗	Cover (related)	path.inst_vcomp_path.assert_emptyData2_wb:precondition1	PRE (1)
✗	Cover (related)	path.inst_vcomp_path.assert_emptyDataBypass_wb:precondition1	PRE (1)
✗	Cover (related)	path.inst_vcomp_path.assert_stop_when_full_wb:precondition1	PRE
✗	Cover (related)	path.inst_vcomp_path.assert_valid1_to_req1_bb:precondition1	PRE (1)
✗	Cover (related)	path.inst_vcomp_path.assert_valid1_to_stop1_bb:precondition1	PRE (1)
✗	Cover (related)	path.inst_vcomp_path.assert_valid2_to_req2_bb:precondition1	PRE (1)
✗	Cover (related)	path.inst_vcomp_path.assert_valid2_to_stop2_bb:precondition1	PRE (1)
✗	Cover (related)	path.inst_vcomp_path.assume_arbitration_is_fair1:precondition1	PRE (1)
✗	Cover (related)	path.inst_vcomp_path.assume_arbitration_is_fair2:precondition1	PRE (1)
✗	Cover (related)	path.inst_vcomp_path.assume_no_gnt1_without_req1:precondition1	PRE
✗	Cover (related)	path.inst_vcomp_path.assume_no_gnt2_without_req2:precondition1	PRE (1)
✗	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_in	PRE
✗	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_out	PRE (1)
✗	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_in	PRE
✗	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_out	PRE (1)
✓	Cover (related)	path.inst_vcomp_path.assert_noPushRemainEmpty_wb:precondition1	B
✓	Cover (related)	path.inst_vcomp_path.assert_nonfull_stop_check_wb:precondition1	B
✓	Cover (related)	path.inst_vcomp_path.assume_no_valid1_if_stall1:precondition1	B
✓	Cover (related)	path.inst_vcomp_path.assume_no_valid2_if_stall2:precondition1	B
✓ !	Assert (live)	path.inst_vcomp_path.assert_data1_flow_check_bb	PRE
✓ !	Assert (live)	path.inst_vcomp_path.assert_data2_flow_check_bb	PRE
✓ !	Assert (live)	path.inst_vcomp_path.assert_emptyDataBypass_wb	PRE
✓	Assert	path.inst_vcomp_path.assert_cross_stop_bb	PRE
✓ !	Assert	path.inst_vcomp_path.assert_data_bypass_bb	PRE
✓ !	Assert	path.inst_vcomp_path.assert_emptyData1_wb	PRE
✓ !	Assert	path.inst_vcomp_path.assert_emptyData2_wb	PRE (1)
✓	Assert	path.inst_vcomp_path.assert_never_full_empty_wb	PRE
✓	Assert	path.inst_vcomp_path.assert_never_overflow_bb	PRE
✓	Assert	path.inst_vcomp_path.assert_never_underflow_bb	PRE (1)
✓	Assert	path.inst_vcomp_path.assert_noPushRemainEmpty_wb	PRE (1)
✓	Assert	path.inst_vcomp_path.assert_only_one_slave_request	PRE (1)
✓ !	Assert	path.inst_vcomp_path.assert_stop_when_full_wb	PRE
✓ !	Assert	path.inst_vcomp_path.assert_valid1_to_req1_bb	PRE
✓ !	Assert	path.inst_vcomp_path.assert_valid1_to_stop1_bb	PRE (1)
✓ !	Assert	path.inst_vcomp_path.assert_valid2_to_req2_bb	PRE (1)
✓ !	Assert	path.inst_vcomp_path.assert_valid2_to_stop2_bb	PRE (1)
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk1.data_integrity	PRE (1)
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk2.no_overflow	PRE (1)
!	Assume (live)	path.inst_vcomp_path.assume_arbitration_is_fair1	?
!	Assume (live)	path.inst_vcomp_path.assume_arbitration_is_fair2	?
	Assume	caselasm1	?
	Assume	caselasm2	?
	Assume	path.inst_vcomp_path.assume_data1_sample_hold_bb	?
	Assume	path.inst_vcomp_path.assume_data2_sample_hold_bb	?
	Assume	path.inst_vcomp_path.assume_data3_sample_hold_bb	?
	Assume	path.inst_vcomp_path.assume_data4_sample_hold_bb	?
	Assume	path.inst_vcomp_path.assume_gnt_cannot_rise_together	?
!	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_no_valid1_if_stall1	?
	Assume	path.inst_vcomp_path.assume_no_valid2_if_stall2	?

(2) 修掉 bug 的 RTL 檔案在 JasperGold 上 prove 的結果

	Type	Name	Engine	B
●	Assume	path.inst_vcomp_path.assume_data3_sample_hold_bb	?	
●	Assume	path.inst_vcomp_path.assume_data4_sample_hold_bb	?	
●	Assume (live)	path.inst_vcomp_path.assume_arbitration_is_fair1	?	
✓	Cover (related)	path.inst_vcomp_path.assume_arbitration_is_fair1:precondition1	Q3	
●	Assume (live)	path.inst_vcomp_path.assume_arbitration_is_fair2	?	
✓	Cover (related)	path.inst_vcomp_path.assume_arbitration_is_fair2:precondition1	Q3	
●	Assume	path.inst_vcomp_path.assume_no_valid1_if_stall1	?	
✓	Cover (related)	path.inst_vcomp_path.assume_no_valid1_if_stall1:precondition1	Q3	
●	Assume	path.inst_vcomp_path.assume_no_valid2_if_stall2	?	
✓	Cover (related)	path.inst_vcomp_path.assume_no_valid2_if_stall2:precondition1	Q3	
●	Assume	path.inst_vcomp_path.assume_no_gnt1_without_req1	?	
✓	Cover (related)	path.inst_vcomp_path.assume_no_gnt1_without_req1:precondition1	Q3	
●	Assume	path.inst_vcomp_path.assume_no_gnt2_without_req2	?	
✓	Cover (related)	path.inst_vcomp_path.assume_no_gnt2_without_req2:precondition1	Q3	
●	Assume	path.inst_vcomp_path.assume_gnt_cannot_rise_together	?	
✓	Assert	path.inst_vcomp_path.assert_never_underflow_bb	K (3)	
✓	Assert	path.inst_vcomp_path.assert_never_overflow_bb	R (3)	
✓	Assert	path.inst_vcomp_path.assert_cross_stop_bb	PRE (1)	
✓	Assert	path.inst_vcomp_path.assert_valid1_to_req1_bb	PRE	
✓	Cover (related)	path.inst_vcomp_path.assert_valid1_to_req1_bb:precondition1	Q3	
✓	Assert	path.inst_vcomp_path.assert_valid2_to_req2_bb	PRE	
✓	Cover (related)	path.inst_vcomp_path.assert_valid2_to_req2_bb:precondition1	Q3	
✓	Assert	path.inst_vcomp_path.assert_only_one_slave_request	PRE (1)	
✓	Assert	path.inst_vcomp_path.assert_valid1_to_stop1_bb	PRE (1)	
✓	Cover (related)	path.inst_vcomp_path.assert_valid1_to_stop1_bb:precondition1	Q3	
✓	Assert	path.inst_vcomp_path.assert_valid2_to_stop2_bb	PRE (1)	
✓	Cover (related)	path.inst_vcomp_path.assert_valid2_to_stop2_bb:precondition1	Q3	
●	Assume	path.inst_vcomp_path.assume_data1_sample_hold_bb	?	
●	Assume	path.inst_vcomp_path.assume_data2_sample_hold_bb	?	
✓	Assert (live)	path.inst_vcomp_path.assert_data1_flow_check_bb	C (9)	
✓	Cover (related)	path.inst_vcomp_path.assert_data1_flow_check_bb:precondition1	L	
✓	Assert (live)	path.inst_vcomp_path.assert_data2_flow_check_bb	C (9)	
✓	Cover (related)	path.inst_vcomp_path.assert_data2_flow_check_bb:precondition1	K	
✓	Assert	path.inst_vcomp_path.assert_data_bypass_bb	Hp (1)	
✓	Cover (related)	path.inst_vcomp_path.assert_data_bypass_bb:precondition1	K	
✓	Assert	path.inst_vcomp_path.assert_never_full_empty_wb	Hp (2)	
✓	Assert	path.inst_vcomp_path.assert_noPushRemainEmpty_wb	R (3)	
✓	Cover (related)	path.inst_vcomp_path.assert_noPushRemainEmpty_wb:precondition1	Q3	
✓	Assert	path.inst_vcomp_path.assert_stop_when_full_wb	PRE (1)	
✓	Cover (related)	path.inst_vcomp_path.assert_stop_when_full_wb:precondition1	Hp	
✓	Assert	path.inst_vcomp_path.assert_nonfull_stop_check_wb	PRE (1)	
✓	Cover (related)	path.inst_vcomp_path.assert_nonfull_stop_check_wb:precondition1	Q3	
✓	Assert	path.inst_vcomp_path.assert_emptyData1_wb	R (3)	
✓	Cover (related)	path.inst_vcomp_path.assert_emptyData1_wb:precondition1	L	
✓	Assert	path.inst_vcomp_path.assert_emptyData2_wb	R (3)	
✓	Cover (related)	path.inst_vcomp_path.assert_emptyData2_wb:precondition1	Q3	
✓	Assert (live)	path.inst_vcomp_path.assert_emptyDataBypass_wb	Tri	
✓	Cover (related)	path.inst_vcomp_path.assert_emptyDataBypass_wb:precondition1	L	
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk1.data_integrity	R (16)	
✓	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk2.no_overflow	R (16)	
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_in	K	
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_out	K	
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_in	Q3	
✓	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_out	Q3	
●	Assume	caselasm1	?	
●	Assume	caselasm2	?	

4. 請完整地列出(包含 code)哪幾條 assertion，即使是有 bug 的 RTLcode，也會被 proved。
因為一開始沒有啟動，所以下面有些會碰巧矇對

assert_never_underflow_bb
assert_never_overflow_bb
assert_cross_stop_bb
assert_valid1_to_req1_bb
assert_valid2_to_req2_bb
assert_only_one_slave_request
assert_valid1_to_stop1_bb
assert_valid2_to_stop2_bb
assert_data_bypass_bb
assert_never_full_empty_wb
assert_noPushRemainEmpty_wb
assert_stop_when_full_wb
assert_emptyData1_wb
assert_emptyData2_wb
assert_data1_flow_check_bb
assert_data2_flow_check_bb
assert_emptyDataBypass_wb

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

✔	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk1.data_integrity	R (16)
✔	Assert	path.inst_vcomp_path.sc3_1.genblk6.core.genblk5.genblk2.no_overflow	R (16)
✔	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_in	K
✔	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[0].data_out	K
✔	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_in	Q3
✔	Cover	path.inst_vcomp_path.sc3_1.genblk6.core.genblk7.COVER[1].data_out	Q3