EC5018 Project

AHB2APB BRIDGE VERIFICATION

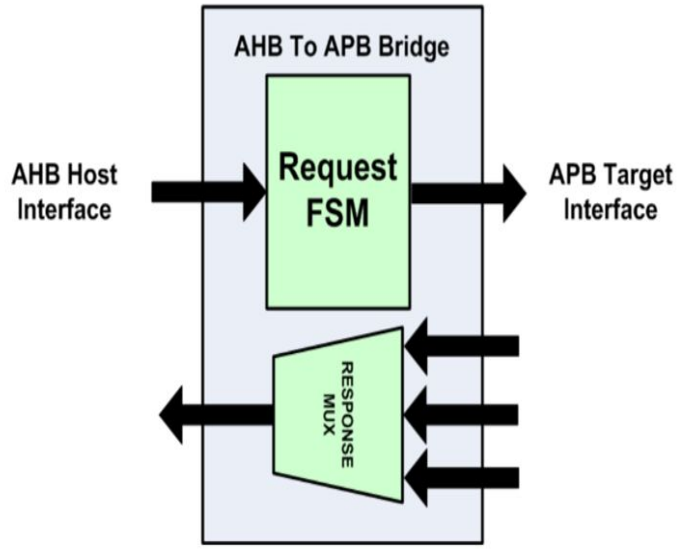
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING,
KANCHEEPURAM

DONE BY -

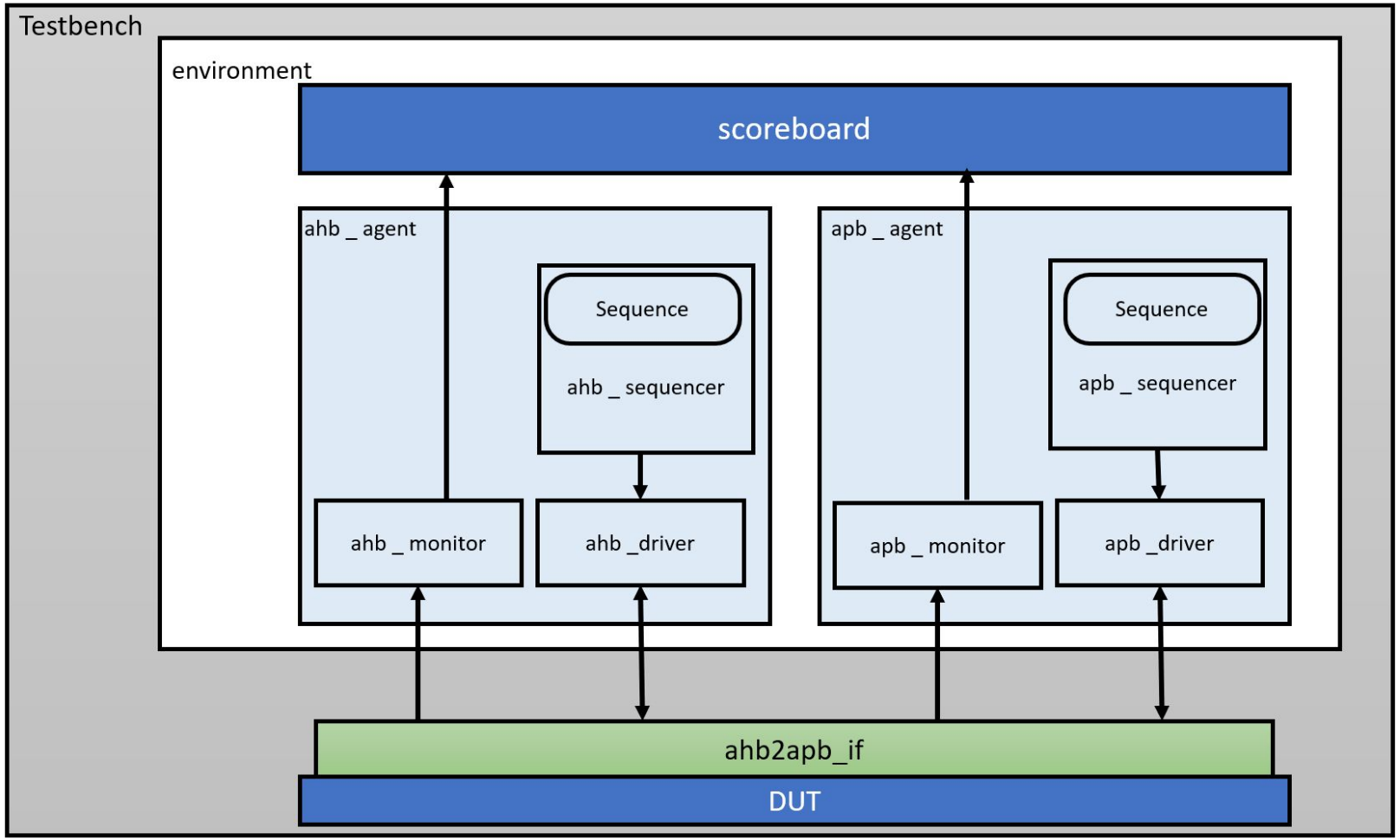
VRSHA SOLACHI R (EC24M2013)
DELPHINA JERUSHA (EC24M2019)

INTRODUCTION

- The **AHB2APB Bridge** acts like a translator between two different types of buses:
  - **AHB** (Advanced High-performance Bus): used for fast, high-bandwidth communication in processors.
  - **APB** (Advanced Peripheral Bus): used for simple, low-power communication with peripherals (like UART, GPIO, Timers).
- The bridge receives **read/write requests** from the AHB side, processes them, and converts them into **APB-compatible signals** so that peripheral devices can understand.
- Internally, the bridge uses a **Finite State Machine (FSM)** to manage the transfer flow, ensuring smooth handshakes between the AHB and APB sides.
- The design ensures that **data integrity, timing, and bus protocols** are followed, even if AHB and APB work at different speeds or have different control signals.



TESTBENCH ARCHITECTURE



UVM REPORT SUMMARY

```
# UVM_INFO /usr/share/questa/questasim/verilog_src/uvm-1.2/src/base/uvm_report_server.svh(847) @ 225: reporter [UVM/REPORT/SERVER]
# --- UVM Report Summary ---
# UVM_INFO :   60
# UVM_WARNING :   45
# UVM_ERROR :    2
# UVM_FATAL :    0
# [Questa UVM]     2
# [UVM/RELNOTES]     1
```

```
# ** Report counts by id
# [AHB_ASSERTION]     1
# [AHB_DRIVER]    36
# [AHB_MONITOR]    23
# [AHB_SEQUENCE]     1
# [APB_AGENT]     4
# [APB_DRIVER]     1
# [APB_MONITOR]     1
# [MY_TEST]     1
# [RNTST]     1
# [SCOREBOARD]    22
# [SEQUENCE]    11
# [UVM/RELNOTES]     1
#
```

AHB DRIVER

```
[l210@l210 FINAL_AHB_APB_BRIDGE_2]$ grep AHB_DRIVER qrun.log
# UVM_ERROR ahb_driver.sv(55) @ 5: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] HTRANS should not be IDLE during active transfer.
# UVM_INFO ahb_driver.sv(59) @ 5: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] WRITE Transaction: Addr=0x7e5, Data=0x534bce33
# UVM_WARNING ahb_driver.sv(67) @ 5: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x7e5, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 15: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0x7e5, HRESP=1
# UVM_INFO ahb_driver.sv(61) @ 25: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] READ Transaction: Addr=0x36b
# UVM_WARNING ahb_driver.sv(67) @ 25: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x36b, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 35: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0x36b, HRESP=1
# UVM_INFO ahb_driver.sv(59) @ 45: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] WRITE Transaction: Addr=0x235, Data=0xaaeaaa50
# UVM_WARNING ahb_driver.sv(67) @ 45: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x235, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 55: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0x235, HRESP=1
# UVM_INFO ahb_driver.sv(59) @ 65: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] WRITE Transaction: Addr=0x4fe, Data=0x55924419
# UVM_WARNING ahb_driver.sv(67) @ 65: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x4fe, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 75: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0x4fe, HRESP=1
# UVM_INFO ahb_driver.sv(59) @ 85: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] WRITE Transaction: Addr=0x13a, Data=0xf2f528f2
# UVM_WARNING ahb_driver.sv(67) @ 85: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x13a, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 95: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0x13a, HRESP=1
# UVM_INFO ahb_driver.sv(59) @ 105: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] WRITE Transaction: Addr=0x561, Data=0x5f78ea8f
# UVM_WARNING ahb_driver.sv(67) @ 105: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x561, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 115: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0x561, HRESP=1
# UVM_INFO ahb_driver.sv(61) @ 125: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] READ Transaction: Addr=0x21f
# UVM_WARNING ahb_driver.sv(67) @ 125: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x21f, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 135: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0x21f, HRESP=1
# UVM_INFO ahb_driver.sv(61) @ 145: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] READ Transaction: Addr=0x4d6
# UVM_WARNING ahb_driver.sv(67) @ 145: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x4d6, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 155: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0x4d6, HRESP=1
# UVM_INFO ahb_driver.sv(61) @ 165: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] READ Transaction: Addr=0x6c4
# UVM_WARNING ahb_driver.sv(67) @ 165: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x6c4, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 175: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0x6c4, HRESP=1
# UVM_INFO ahb_driver.sv(61) @ 185: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] READ Transaction: Addr=0x2b1
# UVM_WARNING ahb_driver.sv(67) @ 185: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x2b1, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 195: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0x2b1, HRESP=1
# UVM_INFO ahb_driver.sv(61) @ 205: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] READ Transaction: Addr=0xfff
# UVM_WARNING ahb_driver.sv(67) @ 205: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0xfff, HRESP=1
# UVM_WARNING ahb_driver.sv(26) @ 215: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Warning: Invalid HRESP during transaction. Addr=0xfff, HRESP=1
# UVM_INFO ahb_driver.sv(61) @ 225: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] READ Transaction: Addr=0x7b6
# UVM_WARNING ahb_driver.sv(67) @ 225: uvm_test_top.env.ahb_agnt.hdriver [AHB_DRIVER] Error Response at Addr=0x7b6, HRESP=1
# [AHB_DRIVER]    36
```

AHB MONITOR

```
[l210@l210 FINAL_AHB_APB_BRIDGE_2]$ grep AHB_MONITOR qrun.log
# UVM_INFO ahb_monitor.sv(20) @ 0: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Build phase completed
# UVM_INFO ahb_monitor.sv(40) @ 15: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured WRITE transaction: Addr=0x7e5, Data=0x534bce33, HRESP=1
# UVM_INFO ahb_monitor.sv(40) @ 25: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured WRITE transaction: Addr=0x7e5, Data=0x534bce33, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 35: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0x36b, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 45: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0x36b, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(40) @ 55: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured WRITE transaction: Addr=0x235, Data=0xaaeaaa50, HRESP=1
# UVM_INFO ahb_monitor.sv(40) @ 65: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured WRITE transaction: Addr=0x235, Data=0xaaeaaa50, HRESP=1
# UVM_INFO ahb_monitor.sv(40) @ 75: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured WRITE transaction: Addr=0x4fe, Data=0x55924419, HRESP=1
# UVM_INFO ahb_monitor.sv(40) @ 85: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured WRITE transaction: Addr=0x4fe, Data=0x55924419, HRESP=1
# UVM_INFO ahb_monitor.sv(40) @ 95: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured WRITE transaction: Addr=0x13a, Data=0xf2f528f2, HRESP=1
# UVM_INFO ahb_monitor.sv(40) @ 105: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured WRITE transaction: Addr=0x13a, Data=0xf2f528f2, HRESP=1
# UVM_INFO ahb_monitor.sv(40) @ 115: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured WRITE transaction: Addr=0x561, Data=0x5f78ea8f, HRESP=1
# UVM_INFO ahb_monitor.sv(40) @ 125: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured WRITE transaction: Addr=0x561, Data=0x5f78ea8f, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 135: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0x21f, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 145: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0x21f, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 155: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0x4d6, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 165: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0x4d6, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 175: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0x6c4, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 185: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0x6c4, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 195: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0x2b1, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 205: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0x2b1, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 215: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0xfff, Data=0x0, HRESP=1
# UVM_INFO ahb_monitor.sv(43) @ 225: uvm_test_top.env.ahb_agnt.hmonitor [AHB_MONITOR] Captured READ transaction: Addr=0xfff, Data=0x0, HRESP=1
# [AHB_MONITOR]    23
```

AHB SEQUENCE

```
[l210@l210 FINAL_AHB_APB_BRIDGE_2]$ grep AHB_SEQUENCE qrun.log
# UVM_INFO ahb_sequence.sv(19) @ 225: uvm_test_top.env.ahb_agnt.hsequencer@@aseq [AHB_SEQUENCE] Generated transaction:
# [AHB_SEQUENCE]     1
```

```
# UVM_INFO ahb_sequence.sv(19) @ 225: uvm_test_top.env.ahb_agnt.hsequencer@@aseq [AHB_SEQUENCE] Generated transaction:
# ----------------------------------------------------------------------------------
# Name                            Type          Size  Value
# ----------------------------------------------------------------------------------
# req                             cmn_seq_item  -     @1223
#   begin_time                    time          64    215
#   end_time                      time          64    225
#   depth                         int           32    'd2
#   parent sequence (name)        string        4     aseq
#   parent sequence (full name)   string        41    uvm_test_top.env.ahb_agnt.hsequencer.aseq
#   sequencer                     string        36    uvm_test_top.env.ahb_agnt.hsequencer
#   ADDR                          integral      32    'h2c9
#   DATA                          integral      32    'h8d0bdc80
#   SLAVE_NUMBER                  integral      3     'h2
#   operation                     integral      1     'h1
#   BURSTMODE                     integral      3     'h1
#   HSIZE                         integral      2     'h2
#   HRESP                         integral      1     'h0
# ----------------------------------------------------------------------------------
```

APB AGENT

```
[l210@l210 FINAL_AHB_APB_BRIDGE_2]$ grep APB_AGENT qrun.log
# UVM_INFO apb_agent.sv(18) @ 0: uvm_test_top.env.apb_agnt [APB_AGENT] Starting build phase...
# UVM_INFO apb_agent.sv(33) @ 0: uvm_test_top.env.apb_agnt [APB_AGENT] Build phase completed
# UVM_INFO apb_agent.sv(40) @ 0: uvm_test_top.env.apb_agnt [APB_AGENT] Connecting driver and sequencer...
# UVM_INFO apb_agent.sv(42) @ 0: uvm_test_top.env.apb_agnt [APB_AGENT] Driver and sequencer connected
# [APB_AGENT]     4
```

APB DRIVER

```
[l210@l210 FINAL_AHB_APB_BRIDGE_2]$ grep APB_DRIVER qrun.log
# UVM_INFO apb_driver.sv(21) @ 0: uvm_test_top.env.apb_agnt.pdriver [APB_DRIVER] Build phase completed
# [APB_DRIVER]     1
```

APB MONITOR

```
[l210@l210 FINAL_AHB_APB_BRIDGE_2]$ grep APB_MONITOR qrun.log
# UVM_INFO apb_monitor.sv(20) @ 0: uvm_test_top.env.apb_agnt.pmonitor [APB_MONITOR] Build phase completed
# UVM_INFO apb_monitor.sv(26) @ 0: uvm_test_top.env.apb_agnt.pmonitor [APB_MONITOR] Run phase started
# [APB_MONITOR]     2
```

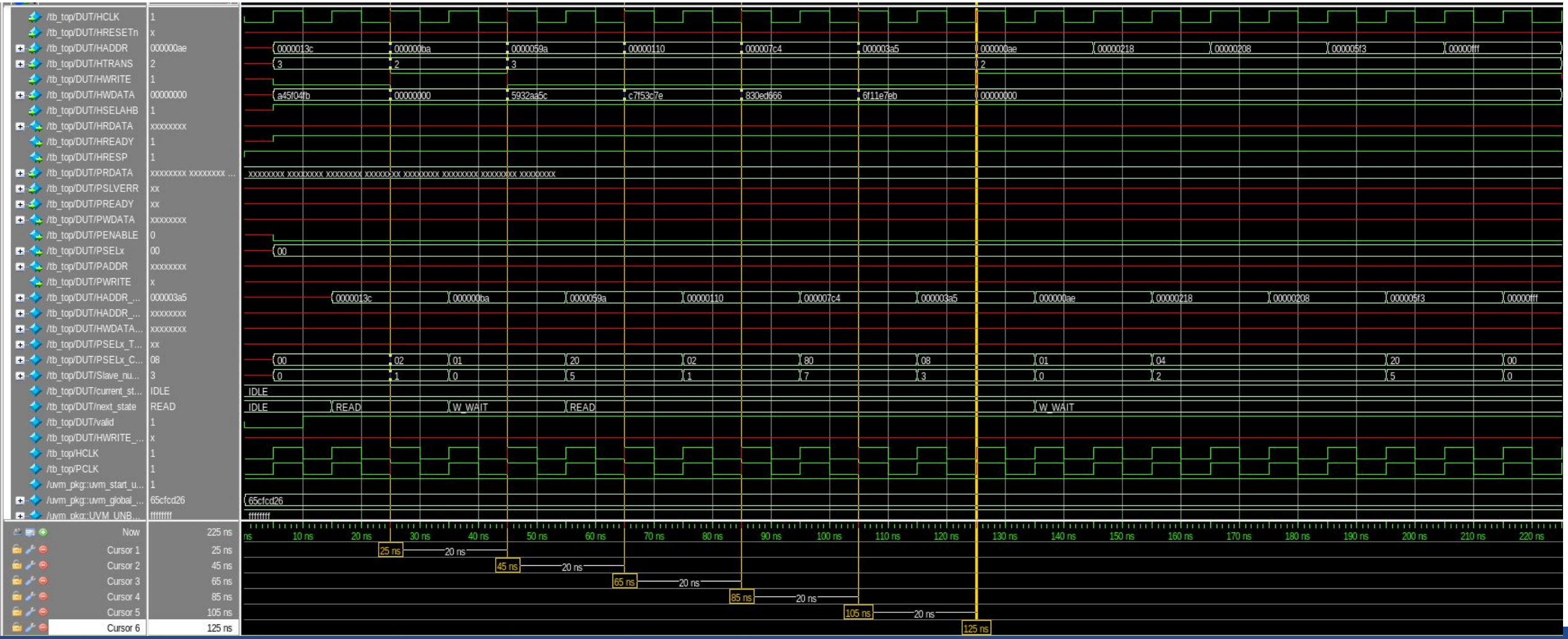
SEQUENCE

```
[l210@l210 FINAL_AHB_APB_BRIDGE_2]$ grep SEQUENCE qrun.log
# UVM_INFO single_write_seq.sv(26) @ 0: uvm_test_top.env.ahb_agnt.hsequencer@@sw_seq [SEQUENCE] Single Write: Addr=0x7e5 Data=0x534bce33 Slave=7
# UVM_INFO single_read_seq.sv(21) @ 25: uvm_test_top.env.ahb_agnt.hsequencer@@sr_seq [SEQUENCE] Single Read: Addr=0x36b Slave=3
# UVM_INFO burst_write_seq.sv(33) @ 45: uvm_test_top.env.ahb_agnt.hsequencer@@bw_seq [SEQUENCE] Burst Write [0]: Addr=0x235 Data=0xaaeaaa50 Slave=2
# UVM_INFO burst_write_seq.sv(33) @ 65: uvm_test_top.env.ahb_agnt.hsequencer@@bw_seq [SEQUENCE] Burst Write [1]: Addr=0x4fe Data=0x55924419 Slave=4
# UVM_INFO burst_write_seq.sv(33) @ 85: uvm_test_top.env.ahb_agnt.hsequencer@@bw_seq [SEQUENCE] Burst Write [2]: Addr=0x13a Data=0xf2f528f2 Slave=1
# UVM_INFO burst_write_seq.sv(33) @ 105: uvm_test_top.env.ahb_agnt.hsequencer@@bw_seq [SEQUENCE] Burst Write [3]: Addr=0x561 Data=0x5f78ea8f Slave=5
# UVM_INFO burst_read_seq.sv(24) @ 125: uvm_test_top.env.ahb_agnt.hsequencer@@br_seq [SEQUENCE] Burst Read [0]: Addr=0x21f Slave=2
# UVM_INFO burst_read_seq.sv(24) @ 145: uvm_test_top.env.ahb_agnt.hsequencer@@br_seq [SEQUENCE] Burst Read [1]: Addr=0x4d6 Slave=4
# UVM_INFO burst_read_seq.sv(24) @ 165: uvm_test_top.env.ahb_agnt.hsequencer@@br_seq [SEQUENCE] Burst Read [2]: Addr=0x6c4 Slave=6
# UVM_INFO burst_read_seq.sv(24) @ 185: uvm_test_top.env.ahb_agnt.hsequencer@@br_seq [SEQUENCE] Burst Read [3]: Addr=0x2b1 Slave=2
# UVM_INFO slave_access_fail_seq.sv(21) @ 205: uvm_test_top.env.ahb_agnt.hsequencer@@saf_seq [SEQUENCE] Slave Access Fail Test: Addr=0xfff Slave=7
# UVM_INFO ahb_sequence.sv(19) @ 225: uvm_test_top.env.ahb_agnt.hsequencer@@aseq [AHB_SEQUENCE] Generated transaction:
# [AHB_SEQUENCE]     1
# [SEQUENCE]    11
```

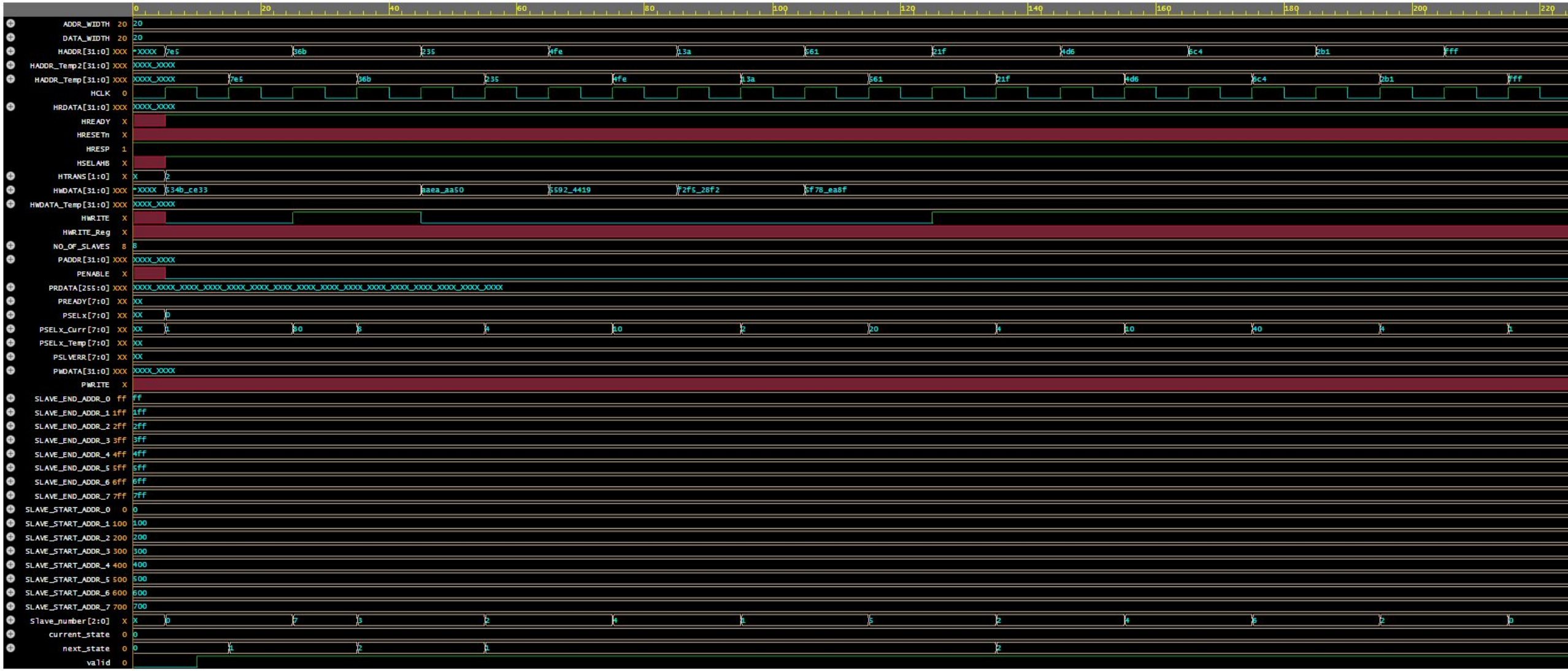
SCOREBOARD

```
[l210@l210 FINAL_AHB_APB_BRIDGE_2]$ grep SCOREBOARD qrun.log
# UVM_WARNING scoreboard.sv(64) @ 15: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 25: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 35: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 45: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 55: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 65: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 75: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 85: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 95: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 105: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 115: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 125: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 135: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 145: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 155: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 165: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 175: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 185: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 195: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 205: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 215: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# UVM_WARNING scoreboard.sv(64) @ 225: uvm_test_top.env.scb [SCOREBOARD] APB Queue is empty, no APB transaction to compare.
# [SCOREBOARD]    22
```

WAVEFORM IN QUESTA_GUI



WAVEFORM IN EDA PLAYGROUND



SEQUENCE MAPPING TABLE

Transaction Type	Index	Address	Data	Slave
Single Write	–	0x7e5	0x534bce33	7
Single Read	–	0x36b	–	3
Burst Write	0	0x235	0xaaeaaa50	2
Burst Write	1	0x4fe	0x55924419	4
Burst Write	2	0x13a	0xf2f528f2	1
Burst Write	3	0x561	0x5f78ea8f	5
Burst Read	0	0x21f	–	2
Burst Read	1	0x4d6	–	4
Burst Read	2	0x6c4	–	6
Burst Read	3	0x2b1	–	2
Slave Access Fail	–	0xfff	–	-

TEST SCENARIOS

S No	Test Scenario	Purpose
1	**Single Write**	Check: AHB Master writes single data to APB Slave.
2	**Single Read**	Check: AHB Master reads single data from APB Slave.
3	**Burst Write**	Check: AHB Master writes a burst (multiple data) to APB.
4	**Burst Read**	Check: AHB Master reads a burst from APB Slave.
5	**Slave Access Failure / Idle**	Check: AHB Master accesses invalid / unreachable address (e.g. address out of range) and APB doesn't respond, and proper HRESP error is raised.

LIST OF ASSERTION

Assertion Name	Description	Purpose	Location Example
HRESP Assertion	When HREADY is high, HRESP must be 2'b00 (OKAY response).	To ensure that HRESP is valid when HREADY is high.	if (vif.HREADY && (vif.HRESP != 2'b00)) uvm_warning("AHB_DRIVER", $sformatf("Warning: Invalid HRESP..."));
HREADY Assertion	After a write operation, HREADY should remain high for at least one cycle.	To ensure that HREADY stays high after a transaction.	assert (vif.HREADY !== 1'b0) else uvm_error("AHB_DRIVER", "HREADY unexpectedly low after transaction.");
HTRANS Assertion	HTRANS should not be T_IDLE during an active transfer.	To ensure that an active transfer is not in IDLE state.	assert (vif.HTRANS != T_IDLE) else uvm_error("AHB_DRIVER", "HTRANS should not be IDLE during active transfer.");`

LIST OF ASSERTION

Assertion Name	Description	Purpose	Location Example
AHB Valid Transfer Assertion	Ensures that when HSELAHB and HREADY are high, the HTRANS is either 2'b10 (Non-sequential) or 2'b11 (Sequential).	To validate that the transfer type is correct when both HSELAHB and HREADY are active.	`assert (!(vif.HSELAHB && vif.HREADY)
APB WRITE Handshake Assertion	Ensures that during a write operation, PSELx, PWRITE, and PENABLE are high.	To validate that the control signals for a write transaction are correct before proceeding.	assert (vif.PSELx[slave] && vif.PWRITE && vif.PENABLE) else uvm_error("APB_DRIVER", "WRITE handshake failed: Invalid control signals");
APB READ Handshake Assertion	Ensures that during a read operation, PSELx, PWRITE (low), and PENABLE are high.	To validate that the control signals for a read transaction are correct before proceeding.	assert (vif.PSELx[slave] && !vif.PWRITE && vif.PENABLE) else uvm_error("APB_DRIVER", "READ handshake failed: Invalid control signals");

ASSERTION GRAPH

Instance	Design unit	Design unit type	Top Category	Visibility	Total coverage	Covergroup %	Assertions count	Assertions hit	Assertions missed	Assertion %	Assertion graph
uvm_root	uvm_root	SVClassItem	TB Component	+acc=<f...							
uvm_test_top	my_test	SVClassItem	TB Component	+acc=<f...							
tb_top	tb_top(fast)	Module	DU Instance	+acc=<f...							
hpvif	ahb2apb_if(f...	Interface	DU Instance	+acc=<f...							
DUT	ahb2apb(fast)	Module	DU Instance	+acc=<f...							
#ALWAYS#15	tb_top(fast)	Process	-	+acc=<f...							
#ALWAYS#16	tb_top(fast)	Process	-	+acc=<f...							
#INITIAL#47	tb_top(fast)	Process	-	+acc=<f...							
#ALWAYS#57	tb_top(fast)	Process	-	+acc=<f...							
uvm_pkg	uvm_pkg(fast)	VlPackage	Package	+acc=<f...	100.00%		1	1	0	100.00%	
pkg	pkg(fast)	VlPackage	Package	+acc=<f...	25.00%	0.00%	2	1	1	50.00%	
questa_uvm_pkg	questa_uvm...	VlPackage	Package	+acc=<f...							
std	std	VlPackage	Package	+acc=<f...							
#vsim_capacity#		Capacity	Statistics	+acc=<n...							

COVERAGE EXPLANATION

Cross Coverage	Involved Coverpoints	Description
CROSS	operation x HSIZE	Verifies combinations of read/write with data sizes
CROSS_ADDR_OP	address x operation	Verifies address interaction with read/write
CROSS_ADDR_HSIZE	address x HSIZE	Verifies address interaction with data size

Coverpoint	Expression
operation	ahb_item.operation
HSIZE	ahb_item.HSIZE
address	ahb_item.ADDR
read_write_cycles	ahb_item.operation
address_ranges	ahb_item.ADDR

COVERAGE

MODULE CODE COVERAGE	STATEMENT	TOGGLE	CONDITION	FSM STATE	FSM TRANSITION	BRANCH	REMARKS
DUT	31.86 %	19.74 %	40.74 %	12.50 %	0.00 % (IDLE STATE)	36.78 %	Partial coverage; improvement needed in toggle, FSM, and branch

COVERAGE

FUNCTIONAL COVERAGE	TOTAL BINS	BINS HIT	BINS MISS	% COVERED	REMARKS
COVERGROUP 1	338	181	157	53.55 %	Moderate coverage; scope for better testing

ASSERTION COVERAGE

```
Assertion Coverage:
    Assertions                        4         1         3    25.00%
--------------------------------------------------------------------------------------------------------------------------------
Name                 File(Line)                    Failure      Pass     Vacuous    Disable    Attempt     Active Peak Active ATV
                                                   Count        Count    Count      Count      Count       Count  Count
--------------------------------------------------------------------------------------------------------------------------------
/pkg/ahb_monitor/run_phase/assert_AHB_valid_transfer
                     ahb_monitor.sv(28)                1         22          0          0         23          0           0 off
/pkg/apb_monitor/run_phase/#ublk#30487#25/immed__47
                     apb_monitor.sv(47)               23          0          0          0         23          0           0 off
/pkg/apb_monitor/run_phase/#ublk#30487#25/immed__46
                     apb_monitor.sv(46)               23          0          0          0         23          0           0 off
/pkg/ahb_sequence/body/immed__17
                     ahb_sequence.sv(17)               0          1          0          0          1          0           0 off
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                       226        47       179    20.79%
```

OBSERVATIONS

- The UVM testbench is built with various components.
- All AHB transactions (single read /single write/burst read/burst write) are returning HRESP = 1, which indicates an error response from the AHB side — a clear sign that the AHB interface is not functioning properly or the DUT isn't responding correctly.
- The APB monitor has not captured any transaction.
- The scoreboard consistently warns that the APB transaction queue is empty — meaning the bridge either didn't generate any APB transactions or APB slaves did not respond.

DESIGN ISSUES AND SUGGESTED FIXES

Issue No.	Observation	Cause	Suggested Fix
1	Transactions with ADDR values like 0xFFF, 0x7BE, 0x71F, 0x1F3, 0x6C6 triggered errors (HRESP=1).	Address falls outside defined **valid slave ranges (0x000 to 0x7FF)**.	Enhance your stimulus constraints to limit address generation to 0x000–0x7FF. You can enforce this in your sequences.
2	Multiple UVM log errors: **"PSELx is not set for any slave while PENABLE is high"** in apb_monitor.	Indicates the APB phase was invalid: no slave selected but PENABLE=1.	Fix design or monitor logic to ensure PENABLE is only asserted when PSELx ≠ 0. Consider adding assertions to catch this earlier.
3	Repeated **"Write operation with zero data at Addr=0xXX"** errors in APB Monitor.	Write transaction triggered but PWDATA not driven or tr.DATA uninitialized.	Review AHB-to-APB write path; ensure PWDATA is properly latched from HWDATA. Also check sequence items set tr.DATA.

DESIGN ISSUES AND SUGGESTED FIXES

Issue No.	Observation	Cause	Suggested Fix
4	HRESP continuously reported as 1 (error) in valid address ranges.	DUT may flag the transaction as invalid due to incorrect HTRANS or timing errors.	Add assertions on HTRANS and HSELAHB to validate they are correct during transaction setup. Adjust timing if needed.
5	Scoreboard Warning: **"APB Queue is empty, no APB transaction to compare"**.	AHB and APB transactions are out of sync; monitor may miss sampling due to timing mismatches.	Review monitor sampling conditions. Use `@(posedge clk)` consistently and ensure APB monitor samples after the APB enable phase.

CONCLUSION

- The testbench for AHB2APB Bridge was implemented using UVM.
- Functional coverage at 53.55% – indicates moderate verification completeness.
- Low FSM & toggle coverage in DUT – more test scenarios needed.
- Code coverage gaps suggest limited corner case exploration.
- Dynamic Address Validation was implemented in the design and verification environment to detect and handle out-of-range address accesses.
- The test sequence triggers AHB single and burst read/write transactions to various slave addresses, but due to HRESP=1 (error) and no APB response, all transactions fail and the scoreboard finds no matching APB data.
- Next Steps: Enhance testbench to improve coverage across FSM, condition, and branch metrics.

IP ADDRESS : 172.16.15.3

EDA PLAYGROUND: https://www.edaplayground.com/x/XJS3