AXI4-Lite LED/Seven-Segment/IRQ Controller - UVM Test Plan

1. Document Information

DUT Name: axi_ledseg_irq
Protocol: AXI4-Lite
Author: Verification Team
Date: February 2026
Version: 1.0

---

2. Design Under Test (DUT) Overview

2.1 Functional Description

The `axi_ledseg_irq` module is an AXI4-Lite slave peripheral providing:

- 8-bit LED output control
- 8-bit seven-segment display control
- Edge-triggered interrupt generation when LED_OUT transitions to 0xFF
- Memory-mapped register interface
- Independent read/write channels

2.2 Key Features

- AXI4-Lite compliant
- Independent read and write FSMs (concurrent operations)
- One outstanding read and one outstanding write supported
- Byte-level write support via WSTRB
- Write-1-to-Clear (W1C) interrupt register
- Synchronous register access

2.3 Register Map

Offset	Name	Access	Description
0x00	LED_REG	R/W	LED output control [7:0]
0x04	SEVENSEG_REG	R/W	Seven-segment output [7:0]

Offset	Name	Access	Description
0x08	IRQ_STATUS	R/W1C	Interrupt status bit[0]

3. Verification Goals

3.1 Coverage Goals

- **Functional Coverage:** 100%
- **Code Coverage:** >95%
- **Toggle Coverage:** >90%
- **FSM Coverage:** 100%

3.2 Verification Objectives

1. Verify AXI4-Lite protocol compliance
2. Verify register read/write functionality
3. Verify byte-enable (WSTRB) operation
4. Verify IRQ generation and clearing
5. Verify concurrent read/write operations
6. Verify reset behavior
7. Verify edge cases and error conditions

4. Test Strategy

4.1 Verification Methodology

- **Framework:** UVM (Universal Verification Methodology)
- **Language:** SystemVerilog
- **Simulation:** Any UVM-compliant simulator (VCS/Questa/Xcelium)

4.2 Verification Components

- AXI4-Lite Master Agent (Driver, Monitor, Sequencer)
- Output Monitor (LED, SEVENSEG, IRQ)
- Scoreboard

- Coverage Collector
- Virtual Sequencer
- Reference Model

---

5. Test List

5.1 Sanity Tests

TEST-001: Reset Test

Objective: Verify reset functionality
Description:

- Assert reset
- Check all outputs are 0
- Check all registers read as 0
- Verify FSMs are in IDLE state

Pass Criteria:

- All registers = 0x00000000
- LED_OUT = 8'h00
- SEVENSEG_OUT = 8'h00
- IRQ_OUT = 1'b0

TEST-002: Basic Write Test

Objective: Verify basic AXI write operation
Description:

- Write to LED register (0x00)
- Write to SEVENSEG register (0x04)
- Verify outputs update correctly

Pass Criteria:

- LED_OUT matches written data
- SEVENSEG_OUT matches written data

- BRESP = 2'b00 (OKAY)

TEST-003: Basic Read Test

Objective: Verify basic AXI read operation
Description:

- Read from all three registers
- Verify read data matches expected values

Pass Criteria:

- Read data matches register contents
- RRESP = 2'b00 (OKAY)

5.2 Protocol Compliance Tests

TEST-004: Write Address Channel Test

Objective: Verify AWVALID/AWREADY handshake
Description:

- Test various AWVALID assertion timings
- Test AWREADY back-pressure scenarios
- Verify address capture on handshake

Pass Criteria:

- Address captured only when AWVALID && AWREADY
- No protocol violations

TEST-005: Write Data Channel Test

Objective: Verify WVALID/WREADY handshake
Description:

- Test various WVALID assertion timings
- Test WREADY back-pressure
- Verify data capture on handshake

Pass Criteria:

- Data captured only when WVALID && WREADY
- No protocol violations

TEST-006: Write Response Channel Test

Objective: Verify BVALID/BREADY handshake
Description:

- Test various BREADY assertion timings
- Verify BVALID generation
- Check BRESP is always OKAY

Pass Criteria:

- Response completes when BVALID && BREADY
- BRESP = 2'b00 always

TEST-007: Read Address Channel Test

Objective: Verify ARVALID/ARREADY handshake
Description:

- Test various ARVALID assertion timings
- Test ARREADY back-pressure
- Verify address capture

Pass Criteria:

- Address captured when ARVALID && ARREADY
- No protocol violations

TEST-008: Read Data Channel Test

Objective: Verify RVALID/RREADY handshake
Description:

- Test various RREADY assertion timings
- Verify RVALID generation
- Check RRESP is always OKAY

Pass Criteria:

- Data valid when RVALID && RREADY
- RRESP = 2'b00 always

5.3 Register Tests

TEST-009: LED Register Read/Write

Objective: Verify LED register functionality
Description:

- Write various patterns to LED register
- Read back and verify
- Check LED_OUT updates

Test Patterns:

- 0x00000000
- 0x000000FF
- 0x000000AA
- 0x00000055
- 0x0000007F

Pass Criteria:

- Read data matches written data
- LED_OUT[7:0] = written_data[7:0]
- Upper bits [31:8] read as 0

TEST-010: Seven-Segment Register Read/Write

Objective: Verify SEVENSEG register functionality
Description:

- Write various patterns
- Read back and verify
- Check SEVENSEG_OUT updates

Pass Criteria:

- Read data matches written data
- SEVENSEG_OUT[7:0] = written_data[7:0]

TEST-011: IRQ Status Register Read

Objective: Verify IRQ status register read
Description:

- Generate IRQ
- Read IRQ_STATUS register
- Verify bit[0] = 1

Pass Criteria:

- IRQ_STATUS[0] = 1 when IRQ active
- IRQ_STATUS[31:1] = 0

TEST-012: WSTRB Byte Enable Test

Objective: Verify byte-level writes
Description:

- Test all WSTRB combinations:
  - 4'b0001 (byte 0)
  - 4'b0010 (byte 1)
  - 4'b0100 (byte 2)
  - 4'b1000 (byte 3)
  - 4'b0011 (bytes 0-1)
  - 4'b1100 (bytes 2-3)
  - 4'b1111 (all bytes)

Pass Criteria:

- Only enabled bytes are written
- Disabled bytes retain previous values

5.4 Interrupt Tests

TEST-013: IRQ Generation - Edge Detection

Objective: Verify IRQ triggers on 0→0xFF transition
Description:

- Write non-0xFF value to LED register
- Write 0xFF to LED register
- Verify IRQ_OUT asserts

Test Cases:

- 0x00 → 0xFF
- 0x7F → 0xFF
- 0xFE → 0xFF

Pass Criteria:

- IRQ_OUT = 1 after transition
- IRQ_STATUS[0] = 1

TEST-014: IRQ No-Trigger Test

Objective: Verify IRQ doesn't trigger incorrectly
Description:

- Test scenarios that should NOT generate IRQ:
  - 0xFF → 0xFF (already high)
  - 0xFF → 0xFE (falling edge)
  - 0x00 → 0x7F (non-0xFF)

Pass Criteria:

- IRQ_OUT remains 0
- IRQ_STATUS[0] remains 0

TEST-015: IRQ Clear - Write-1-to-Clear

Objective: Verify W1C functionality
Description:

- Generate IRQ
- Write 1 to IRQ_STATUS[0]
- Verify IRQ clears
- Write 0 to IRQ_STATUS[0]
- Verify no effect

Pass Criteria:

- Writing 1 clears IRQ
- Writing 0 has no effect
- IRQ_OUT follows IRQ_STATUS[0]

TEST-016: Multiple IRQ Generation

Objective: Verify multiple IRQ cycles
Description:

- Generate IRQ → Clear → Generate again
- Repeat 10 times

Pass Criteria:

- Each cycle generates and clears correctly
- No stuck IRQ conditions

TEST-017: IRQ Persistence Test

Objective: Verify IRQ persists until cleared
Description:

- Generate IRQ
- Wait 100 clock cycles
- Verify IRQ still asserted
- Clear IRQ
- Verify IRQ deasserts

Pass Criteria:

- IRQ persists until explicitly cleared

5.5 Concurrent Operation Tests

TEST-018: Simultaneous Read and Write

Objective: Verify independent read/write channels
Description:

- Initiate write to LED register
- Simultaneously initiate read from SEVENSEG register
- Verify both complete correctly

Pass Criteria:

- Both transactions complete successfully
- No interference between channels
- Correct data integrity

TEST-019: Back-to-Back Writes

Objective: Verify sequential write handling
Description:

- Perform back-to-back writes to different registers
- Verify each write completes

Pass Criteria:

- All writes complete successfully

- Register values are correct

TEST-020: Back-to-Back Reads

Objective: Verify sequential read handling
Description:

- Perform back-to-back reads from different registers

Pass Criteria:

- All reads return correct data
- RRESP = OKAY for all

TEST-021: Interleaved Read/Write

Objective: Verify complex transaction ordering
Description:

- Interleave multiple reads and writes
- Sequence: W-R-W-R-W-R

Pass Criteria:

- All transactions complete correctly
- No deadlocks or hangs

5.6 Boundary and Corner Cases

TEST-022: Unaligned Address Test

Objective: Verify behavior with unaligned addresses
Description:

- Access registers with addresses 0x01, 0x02, 0x03
- Document behavior

Pass Criteria:

- Design handles gracefully (implementation defined)

TEST-023: Invalid Address Test

Objective: Verify behavior for unmapped addresses
Description:

- Read/write to addresses outside 0x00-0x08
- Test: 0x0C, 0x10, 0x100

Pass Criteria:

- No system hang
- Returns DEADBEEF or error response

TEST-024: Maximum Delay Test

Objective: Verify handling of slow master
Description:

- Insert maximum delays on VALID signals
- Insert maximum delays on READY responses

Pass Criteria:

- Transactions complete eventually
- No timeouts or hangs

TEST-025: Reset During Transaction

Objective: Verify reset handling mid-transaction
Description:

- Start write transaction
- Assert reset during data phase
- Verify clean recovery

Pass Criteria:

- All outputs deassert

- FSMs return to IDLE
- Next transaction works correctly

TEST-026: Stress Test

Objective: Verify sustained operation
Description:

- Run 1000 random transactions
- Mix of reads, writes, concurrent ops
- Random delays

Pass Criteria:

- All transactions complete successfully
- No errors or hangs
- Coverage metrics met

5.7 Negative Tests

TEST-027: BRESP Check

Objective: Verify write response is always OKAY
Description:

- Perform various writes
- Check BRESP = 2'b00 always

Pass Criteria:

- BRESP = 2'b00 for all writes

TEST-028: RRESP Check

Objective: Verify read response is always OKAY
Description:

- Perform various reads

- Check RRESP = 2'b00 always

Pass Criteria:

- RRESP = 2'b00 for all reads

---

6. Coverage Plan

6.1 Functional Coverage

Coverage Group 1: Register Access

systemverilog

```
covergroup cg_register_access;
    cp_reg_addr: coverpoint addr {
        bins led_reg = {32'h0};
        bins seg_reg = {32'h4};
        bins irq_reg = {32'h8};
        bins invalid = default;
    }
    cp_access_type: coverpoint access_type {
        bins read = {READ};
        bins write = {WRITE};
    }
    cx_reg_access: cross cp_reg_addr, cp_access_type;
endgroup
```

Coverage Group 2: WSTRB Patterns

systemverilog

```
covergroup cg_wstrb;
    cp_wstrb: coverpoint wstrb {
        bins byte0 = {4'b0001};
        bins byte1 = {4'b0010};
        bins byte2 = {4'b0100};
        bins byte3 = {4'b1000};
        bins halfword_low = {4'b0011};
        bins halfword_high = {4'b1100};
        bins word = {4'b1111};
        bins other = default;
    }
endgroup
```

Coverage Group 3: IRQ Scenarios

```systemverilog
covergroup cg_irq;
    cp_irq_trigger: coverpoint irq_triggered {
        bins triggered = {1};
        bins not_triggered = {0};
    }
    cp_irq_clear: coverpoint irq_cleared {
        bins cleared = {1};
        bins not_cleared = {0};
    }
    cp_led_value: coverpoint led_value {
        bins all_high = {8'hFF};
        bins other = default;
    }
endgroup
```

Coverage Group 4: AXI Protocol

```systemverilog
```

```
covergroup cg_axi_protocol;
    cp_write_delay: coverpoint awvalid_to_awready_delay {
        bins zero = {0};
        bins short = {[1:5]};
        bins long = {[6:20]};
    }
    cp_read_delay: coverpoint arvalid_to_arready_delay {
        bins zero = {0};
        bins short = {[1:5]};
        bins long = {[6:20]};
    }
endgroup
```

Coverage Group 5: Concurrent Operations

```systemverilog
covergroup cg_concurrent;
    cp_concurrent: coverpoint concurrent_rw {
        bins read_only = {0};
        bins write_only = {1};
        bins simultaneous = {2};
    }
endgroup
```

6.2 Code Coverage Targets

- **Line Coverage:** >95%
- **Branch Coverage:** >95%
- **Condition Coverage:** >90%
- **FSM Coverage:** 100%
- **Toggle Coverage:** >90%

---

7. Test Environment Architecture

```
+-------------------------------------------------+
|                   TEST BENCH                    |
| +---------------------------------------------+ |
| |             Virtual Sequencer               | |
|  +-------------------------------------------+  |
|                     |                        |
|  +------------------+--------------------+         |
| |                  |                  |         |
| | AXI Master    Output Monitor    Scoreboard    |
| | Agent           |                  |         |
| |  +---------+     |                  |         |
| |  | Driver  |     |                  |         |
| |  | Monitor |     |                  |         |
| |  | Seqr    |     |                  |         |
|  +--+---------+-----+-------------------+         |
|        |          |                  |         |
+---------|-----------|-------------------|----------+
          |          |                  |
      +---v-----------v-------------------v---+
      |            DUT Interface              |
      |     (axi_ledseg_irq)                  |
      +---------------------------------------+
```

8. Test Execution Plan

8.1 Regression Suite

Suite	Tests	Purpose
Sanity	TEST-001 to TEST-003	Quick smoke test
Protocol	TEST-004 to TEST-008	AXI compliance
Functional	TEST-009 to TEST-017	Feature verification
Stress	TEST-018 to TEST-026	Robustness
Full	All tests	Complete verification

8.2 Pass/Fail Criteria

- **Pass:** All assertions pass, functional coverage >95%, no X/Z propagation
- **Fail:** Any assertion failure, coverage gap, simulation error

9. Deliverables

1. UVM Testbench (all components)
2. Test sequences for all test cases
3. Coverage report
4. Test execution report
5. Bug report (if any)
6. Sign-off document

---

10. Schedule

Phase	Duration	Status
Testbench Development	2 weeks	In Progress
Test Development	2 weeks	Pending
Regression & Debug	2 weeks	Pending
Coverage Closure	1 week	Pending
Sign-off	1 week	Pending

---

11. Risks and Mitigation

Risk	Impact	Mitigation
Incomplete Spec	High	Clarify with designer
Coverage Gaps	Medium	Directed tests
Complex IRQ Logic	Medium	Focused IRQ test suite
Concurrent Ops	Medium	Stress testing

---

12. References

1. AXI4-Lite Specification Document
2. UVM 1.2 User Guide
3. Design RTL Source Code
4. Internal Verification Guidelines

End of Test Plan