Tiny Hardware OTP Authenticator Verification

TEAM - 4

Varun G

Ayan B M

Prathiksha

Kanthimathi C

Sharath Kumar U K

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: PROJECT OVERVIEW AND SPECIFICATIONS

An OTP (One-Time Password) Authenticator is a critical security system used for reliable authentication in digital applications. Unlike static password systems, where authentication codes remain constant, OTP systems generate temporary passwords that change dynamically, making them essential for secure access control requiring time-sensitive authentication.

The OTP Authenticator operates as a multi-clock domain security system with independent LFSR-based password generation, user input validation, timing enforcement, and dual 7-segment display management. It implements state machine-controlled authentication logic with configurable timing parameters for expiry, attempt tracking, and security lockout mechanisms.

1.1 Key Features

1. LFSR-based OTP generation verified using a 16-bit LFSR with polynomial ($x^{16}+x^{14}+x^{13}+x^{11}+x^{9}+x^{7}+1$) and seed 16’hACE1, generating secure 4-digit BCD OTPs.
2. Dual 7-segment display confirmed with independent OTP and user outputs, featuring a 2-second data/status toggle using anode multiplexing.
3. Four-state FSM (IDLE, GENERATE_OTP, ENTER_OTP, UNLOCK) verified for correct OTP flow, user input handling, and state transitions.
4. All timing parameters derived from the 50 MHz clock verified for accuracy, including 50-second expiry, 2-second toggle, and 5-second status hold.
5. Three-attempt security limit validated with automatic lockout on failures and ‘U’/‘L’ indications displayed for authentication status.
6. User input logic verified for sequential 4-digit entry using edge-detected user_latch and real-time display updates.

7. Multi-clock domain operation validated with derived 2 kHz, 0.5 Hz, and timer clocks showing stable synchronization.
8. Display format verified for both data mode (OTP/user input) and status mode [STATUS][-][ATTEMPT][A] with U/L/E indicators.
9. Edge detection confirmed clean single-cycle triggers for otp_latch and user_latch, eliminating noise-induced activations.
10. Asynchronous active-low reset verified to correctly initialize FSM, timers, and displays to known safe states.

1.2 Verification Objectives

- Functional correctness verified by ensuring accurate LFSR-based OTP generation, correct 4-digit BCD conversion, and proper authentication unlocking only on valid OTP entry.
- UVM testbench developed with modular components—driver, sequencer, monitor, agent, scoreboard, coverage, and assertions—for complete verification of authentication logic.
- Authentication sequence validated for correct unlock on valid OTP, attempt count increment on mismatch, and lockout after three failed attempts with proper status display.
- Timing system verified for accurate 50-second OTP expiry, 2-second display toggle, and 5-second status hold; all timing relationships confirmed against the 50 MHz clock.
- Security policy validated for correct attempt tracking, enforced lockout, OTP expiry handling, and proper reset-based recovery to secure default state.
- Display system verified for correct 7-segment encoding of digits and status symbols, stable multiplexing, and reliable data/status mode transitions.
- LFSR reference model implemented for bit-accurate comparison; DUT output matched reference sequence for correctness, randomness, and periodicity.
- Multi-clock domain testing confirmed accurate clock dividers (2 kHz, 0.5 Hz) and synchronized operation without timing violations or metastability.
- Edge and corner case testing ensured robustness under resets, boundary timings, rapid inputs, and concurrent operations across FSM states.

- Assertion-based verification applied to detect invalid state transitions, timing violations, display format errors, and security rule breaches.

1.3 DUT Interfaces

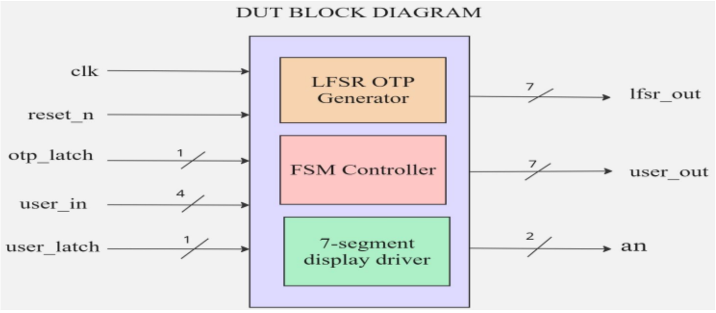


Figure 1 Block Diagram

Signal	**Direction**	**Width**	**Description**
clk	Input	1 Bit	System clock
reset_n	Input	1 Bit	Active-low Asynchronous reset
user_in	Input	4 Bit	User-entered digit (0-9)
otp_latch	Input	1 Bit	Captures current LFSR
user_latch	Input	1 Bit	Captures user digit into user register
lfsr_out	Output	7 Bit	Decoded output of LFSR for 7-segment display
user_out	Output	7 Bit	Decoded output of USER for 7-segment display
an	Output	7 Bit	Anode Output for 7-segment display

1.4 Finite State Machine (FSM)

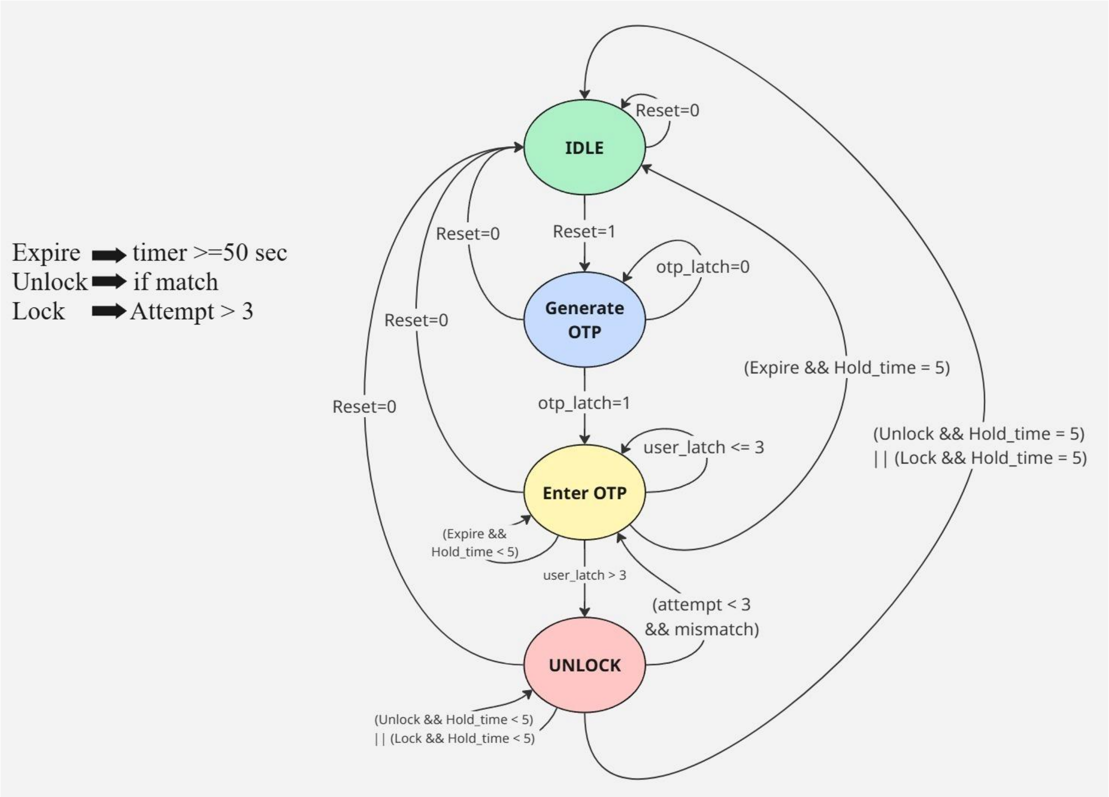

Expire ➡ timer >=50 sec
Unlock ➡ if match
Lock ➡ Attempt > 3
IDLE
Reset=0
Reset=1
Reset=0
Reset=0
Reset=0
Generate OTP
otp_latch=0
otp_latch=1
(Expire && Hold_time = 5)
Enter OTP
user_latch <= 3
(Unlock && Hold_time = 5) || (Lock && Hold_time = 5)
(Expire && Hold_time < 5)
user_latch > 3
(attempt < 3 && mismatch)
UNLOCK
(Unlock && Hold_time < 5) || (Lock && Hold_time < 5)

Figure 2 FSM

IDLE:

Default reset state where all registers, timers, and displays are initialized; system waits for reset release to start OTP generation.

GENERATE OTP:

LFSR runs continuously generating OTP values; OTP is latched on otp_latch=1 to proceed for user entry.

ENTER OTP:

Accepts sequential 4-digit user inputs via user_latch; compares entries, tracks attempts, and monitors expiry timer.

UNLOCK:

Displays authentication result — unlock if OTP matches, lock after 3 failed attempts, or expire after timeout — then returns to IDLE after 5 seconds.

Chapter 2: TESTBENCH ARCHITECTURE AND METHODOLOGY

2.1 Testbench Architecture

The figure below represents a UVM-based Testbench Architecture built using SystemVerilog for verifying the OTP Authenticator Design Under Verification (DUV). This methodology follows a structured, layered approach to enable effective verification through stimulus generation, response monitoring, timing validation, and security policy checking. The testbench is hierarchically divided into several key components under a TOP test module, which encapsulates the TEST, ENVIRONMENT, and AGENTS.

The UVM architecture provides reusability, configurability, and systematic verification coverage through modular components that interact via Transaction-Level Modeling (TLM) interfaces. The dual-agent architecture (active agent for inputs, passive agent for outputs) reflects the input/output nature of the OTP authenticator, with comprehensive monitoring of both stimulus and response signals.

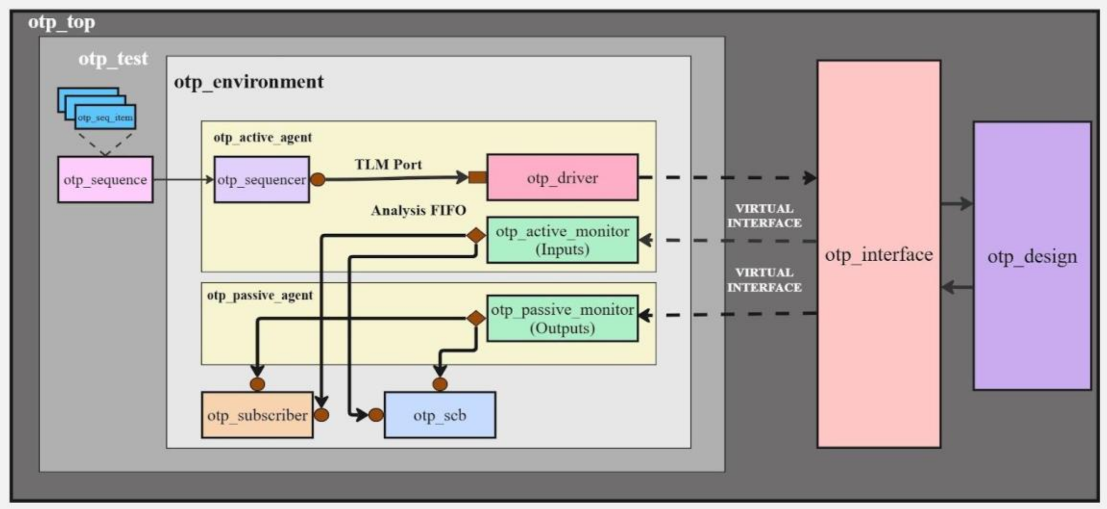


Figure 3 Testbench Architecture

2.2 Architecture and Component Details

The OTP testbench implements a complete UVM verification environment with active agent for stimulus generation and passive agent for response monitoring. The architecture ensures proper stimulus generation, monitoring, checking, and coverage collection for comprehensive OTP authenticator verification.

2.2.1 Sequence item

Defines transaction-level data exchanged between sequence, driver, and monitor. Includes randomizable inputs *(*otp_latch, user_latch, user_in) and observed outputs (lfsr_out, user_out, an). Implements uvm_field_int macros for automated copy, compare, and debug support.

2.2.2 Sequence

Generates OTP-related stimulus patterns for various test scenarios like latch, input, mismatch, and expiry. Controls transaction flow via body() task ensuring realistic timing and sequencing. Validates authentication and timing responses through directed and random sequences.

2.2.3 Sequencer

Acts as the transaction manager between sequence and driver using TLM seq_item_port. Implements uvm_sequencer for otp_sequence_item ensuring safe data transfer. Supports single or concurrent sequence arbitration based on test configuration.

2.2.4 Driver

Converts sequence items to DUT pin-level activity using clocking blocks for synchronization. Drives signals (*otp_latch, user_latch, user_in*) with precise timing and hold durations. Ensures stable stimulus generation aligned with DUT clock domain.

2.2.5 Monitor

Passively observes and captures DUT input/output transactions through clocking blocks. Active monitor tracks inputs; passive monitor tracks DUT responses. Broadcasts transactions via analysis ports to scoreboard and coverage component

2.2.6 Scoreboard

Checks DUT functionality against reference models for LFSR, authentication, and timing. Validates OTP accuracy, expiry timing, lockout policy, and display correctness. Reports test results with detailed pass/fail statistics.

2.2.7 Subscriber

Collects functional coverage from input and output monitors to assess verification completeness. Implements covergroups for digits, latches, display states, and anode control. Generates coverage reports summarizing overall test completeness.

2.2.8 Agent

Encapsulate sequencer, driver, and monitor components for modularity and reuse. Active agent drives stimulus; passive agent observes DUT outputs.
Provide configurable setup enabling clear separation of input and output verification.

2.2.9 Environment

Top-level container connecting all verification components—agents, scoreboard, and subscriber. Handles build and connect phases linking TLM analysis ports for transaction flow. Coordinates checking and coverage collection across all components.

2.2.10 Test

Controls execution of verification scenarios by instantiating environment and sequences. Raises objections, starts OTP stimulus, and ensures full simulation duration. Supports reusable test extensions for authentication, timing, and edge cases.

2.2.11 Top

Integrates DUT, UVM testbench, and interface connections under a single testbench.
Generates master clock, reset signals, and launches *run_test()* execution.
Passes virtual interface handles through *uvm_config_db* and enforces simulation timeout.

Chapter 3: VERIFICATION PLAN

3.1 Test Plan and Test Cases

3.1.1 Directed Testcases

S.NO	TESTCASE NAME	DESCRIPTION
1	Initial_reset	Active-low reset; FSM goes to IDLE state
2	check_latched_otp	Verifies that when otp_latch is asserted, the output lfsr_out correctly reflects the latched OTP value generated by the LFSR.
3	match_otp_A1	Validates successful authentication when user-entered OTP matches lfsr_out on the first attempt before expiry time.
4	match_otp_A2	Checks OTP match condition on the second attempt within expiry time, ensuring correct unlock response.
5	match_otp_A3	Confirms correct unlock behavior when OTP matches lfsr_out on the third and final allowed attempt before expiry.
6	mismatch_otp_A1	Verifies authentication failure when OTP mismatch occurs on the first attempt within the valid time window.
7	mismatch_otp_A2	Tests system response for OTP mismatch during the second attempt, ensuring proper attempt increment and no unlock.
8	attempt_cnt_reached	Validates lockout condition when the user exceeds three unsuccessful OTP attempts before expiry.
9	expiry_time	Checks system behavior when timer exceeds the 50-second expiry limit before completing OTP entry, ensuring expiry status activation.

3.1.2 Corner Testcases

S.NO	TESTCASE NAME	DESCRIPTION
10	user_in_check	Validates that user_in accepts only valid BCD digits (0–9) and rejects or ignores values outside the range.
11	user_latch_high	Checks system behavior when user_latch is held high continuously while user_in changes, ensuring only one valid capture per edge.
12	no_otp_latch	Verifies that without asserting otp_latch, user inputs are not accepted for authentication and OTP comparison does not occur.
13	otp_latch_in_between	Tests the case where otp_latch is asserted again after one attempt, confirming system stability and prevention of mid-sequence re-latching.

3.2 Coverage Plan

S.NO	COVERPOINT NAME	DESCRIPTION
1	OTP_LATCH	Verifies latch control signal transitions (0→1 and 1→0) for correct OTP capture on positive edge.
2	USER_IN	Validates user_in value range and classification: Small (0–2), Medium (3–5), Large (6–9), and Illegal (10–15).
3	USER_LATCH	Checks latch control behavior for user input capture on valid posedge and negedge transitions.
4	LFSR_OUT	Confirms generated OTP and status display values: digits (0–9), Locked (10), Unlocked (11), Expired (12), Dash (13), Attempt (14), and Off (15).

5	USER_OUT	Validates user-entered digits displayed correctly within range (0–9) and categorized as S (0–2), M (3–5), L (6–9).
6	AN	Verifies correct anode multiplexing pattern for display selection (00, 01, 10, 11).
7	USER_IN_x_USER_LATCH	Cross coverage between posedge of user_latch and user_in categories (S, M, L); checks all valid and illegal combinations.

Chapter 4: DESIGN AND SPECIFICATION BUGS

4.1 Specification Bugs

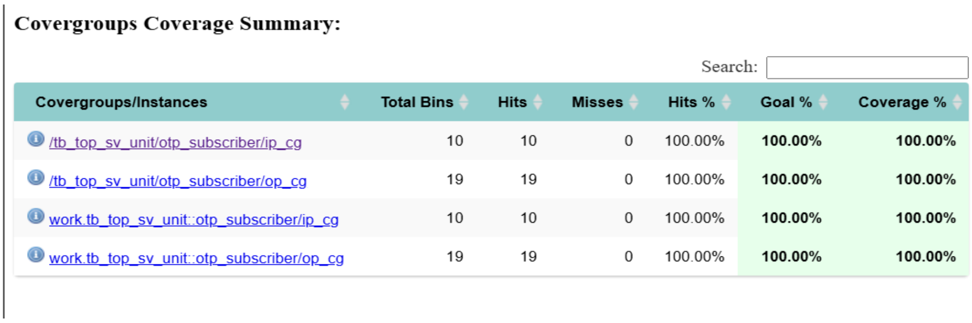
S.NO	DESCRIPTION
1	DUT works on posedge detection logic for otp_latch and user_latch signals, but this was not mentioned in the specification.
2	The master clock frequency used in the design was specified.
3	The application and usage of the HOLD_TIME parameter within the design are specified.
4	The representation of FSM outputs (LOCKED, UNLOCKED, EXPIRED) on the 7-segment display is documented.
5	The tap bit positions for the 16-bit LFSR implementation are detailed.
6	The behavior of the 50-second inactivity timer is clarified — whether it resets after each user attempt or applies cumulatively across all attempts.
7	The clock divider frequencies used for driving the 7-segment display are defined.
8	The FSM transition condition from IDLE to GENERATE_OTP state is explicitly defined.

4.2 Design Bugs

S.NO	DESCRIPTION
1	The expiry flag timing as per the specification is 50 seconds, but during testing, the flag was raised at 50 sec 20 ns — one clock cycle later than expected.

Chapter 5: RESULTS AND ANALYSIS

5.1 Functional Coverage Details:



Covergroups Coverage Summary:

Search:

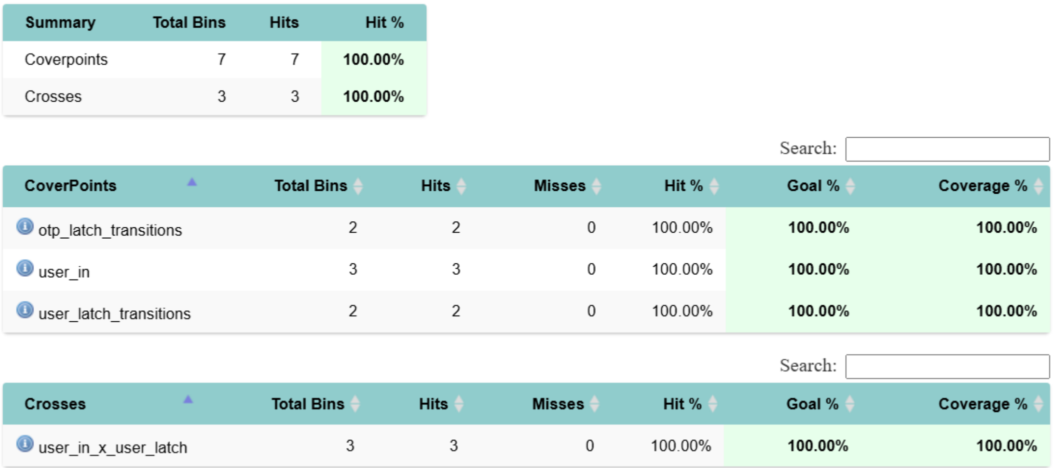
Covergroups/Instances	Total Bins	Hits	Misses	Hits %	Goal %	Coverage %
/tb_top_sv_unit/otp_subscriber/ip_cg	10	10	0	100.00%	**100.00%**	**100.00%**
/tb_top_sv_unit/otp_subscriber/op_cg	19	19	0	100.00%	**100.00%**	**100.00%**
work.tb_top_sv_unit::otp_subscriber/ip_cg	10	10	0	100.00%	**100.00%**	**100.00%**
work.tb_top_sv_unit::otp_subscriber/op_cg	19	19	0	100.00%	**100.00%**	**100.00%**

Figure 4. Functional coverage

5.1.1 INPUT COVERAGE:

Covergroup type:

ip_cg



Summary	Total Bins	Hits	Hit %
Coverpoints	7	7	**100.00%**
Crosses	3	3	**100.00%**

Search:

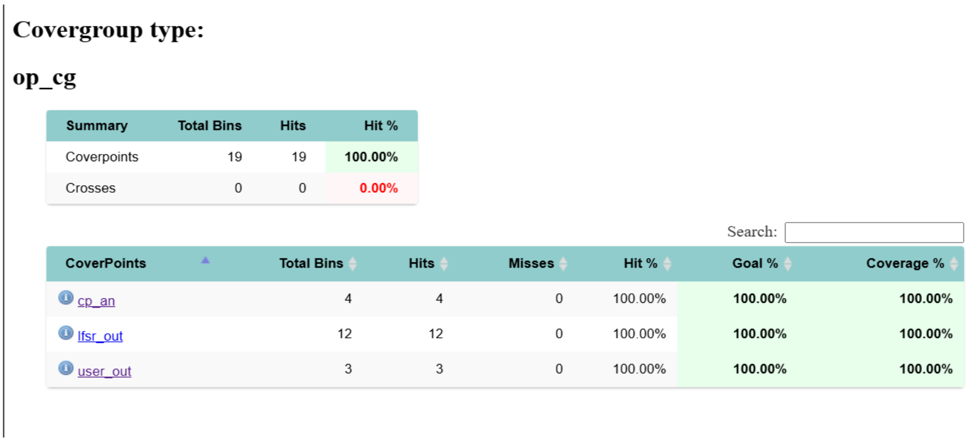
CoverPoints	Total Bins	Hits	Misses	Hit %	Goal %	Coverage %
otp_latch_transitions	2	2	0	100.00%	**100.00%**	**100.00%**
user_in	3	3	0	100.00%	**100.00%**	**100.00%**
user_latch_transitions	2	2	0	100.00%	**100.00%**	**100.00%**

Search:

Crosses	Total Bins	Hits	Misses	Hit %	Goal %	Coverage %
user_in_x_user_latch	3	3	0	100.00%	**100.00%**	**100.00%**

Figure 5 Input Coverage

5.1.2 OUTPUT COVERAGE:



Covergroup type:

op_cg

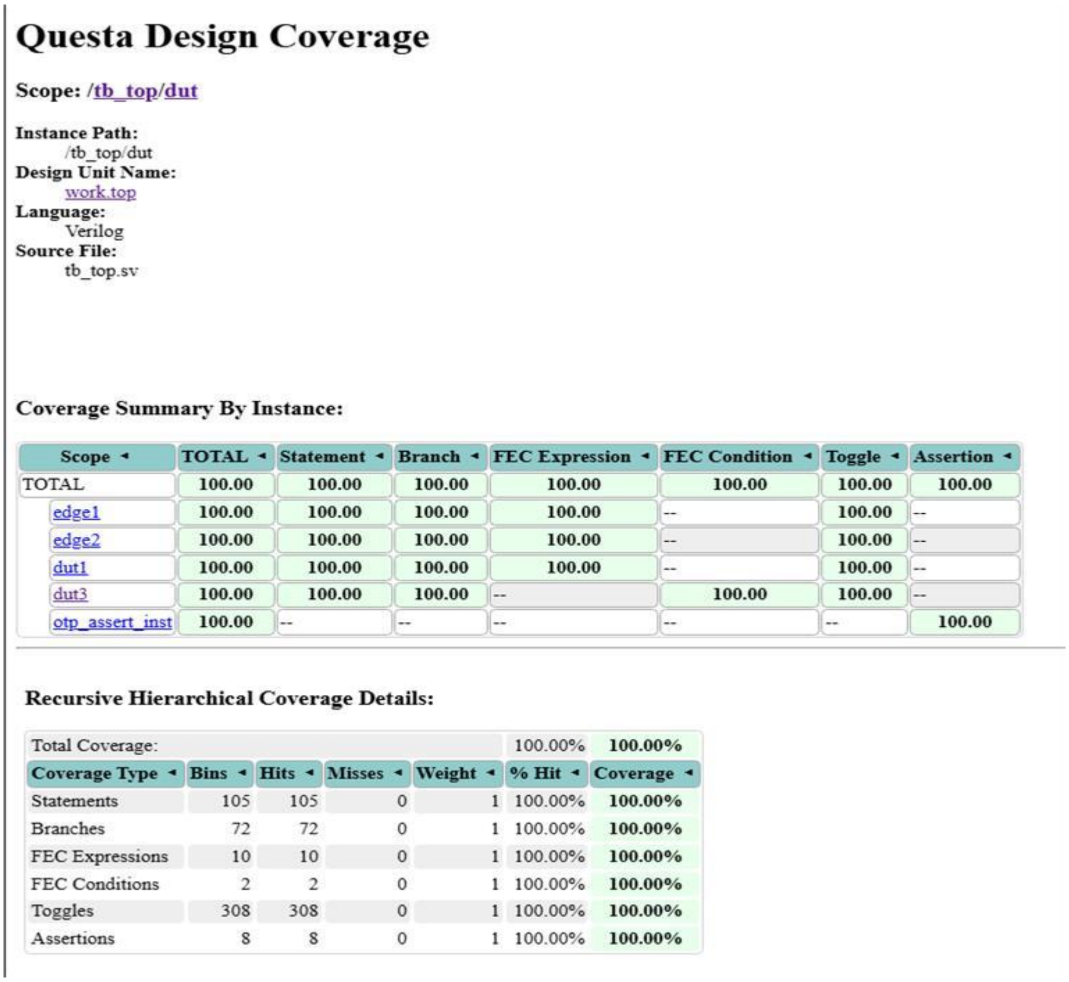
Summary	Total Bins	Hits	Hit %
Coverpoints	19	19	100.00%
Crosses	0	0	0.00%

Search:

CoverPoints	Total Bins	Hits	Misses	Hit %	Goal %	Coverage %
cp_an	4	4	0	100.00%	100.00%	100.00%
lfsr_out	12	12	0	100.00%	100.00%	100.00%
user_out	3	3	0	100.00%	100.00%	100.00%

Figure 6 Output coverage

5.2 Code Coverage Results:



Questa Design Coverage

Scope: /tb_top/dut

Instance Path:
/tb_top/dut
Design Unit Name:
work.top
Language:
Verilog
Source File:
tb_top.sv

Coverage Summary By Instance:

Scope	TOTAL	Statement	Branch	FEC Expression	FEC Condition	Toggle	Assertion
TOTAL	100.00	100.00	100.00	100.00	100.00	100.00	100.00
edge1	100.00	100.00	100.00	100.00	--	100.00	--
edge2	100.00	100.00	100.00	100.00	--	100.00	--
dut1	100.00	100.00	100.00	100.00	--	100.00	--
dut3	100.00	100.00	100.00	--	100.00	100.00	--
otp_assert_inst	100.00	--	--	--	--	--	100.00

Recursive Hierarchical Coverage Details:

Total Coverage:					100.00%	100.00%
Coverage Type	**Bins**	**Hits**	**Misses**	**Weight**	**% Hit**	**Coverage**
Statements	105	105	0	1	100.00%	100.00%
Branches	72	72	0	1	100.00%	100.00%
FEC Expressions	10	10	0	1	100.00%	100.00%
FEC Conditions	2	2	0	1	100.00%	100.00%
Toggles	308	308	0	1	100.00%	100.00%
Assertions	8	8	0	1	100.00%	100.00%

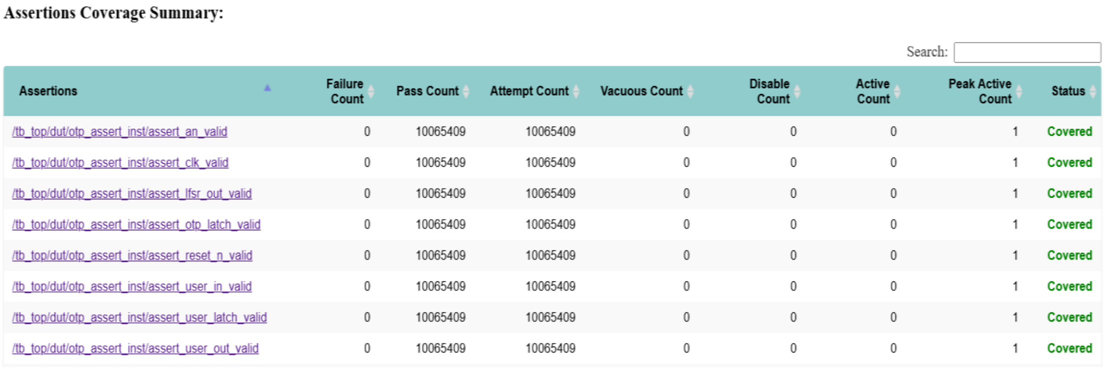
Figure 7 Code coverage

The following coverage metrics were achieved during verification execution:

- Overall Functional Coverage: 100%
- Code Coverage: 100%
- Assertion Coverage: 100%

5.3 Assertion Results

SystemVerilog assertions continuously monitor OTP behavior throughout simulation, catching protocol violations and security policy breaches in real-time. The assertion suite covers signal validity, protocol compliance, and timing requirements.



Assertions Coverage Summary:

Search:

Assertions	Failure Count	Pass Count	Attempt Count	Vacuous Count	Disable Count	Active Count	Peak Active Count	Status
/tb_top/dut/otp_assert_inst/assert_an_valid	0	10065409	10065409	0	0	0	1	Covered
/tb_top/dut/otp_assert_inst/assert_clk_valid	0	10065409	10065409	0	0	0	1	Covered
/tb_top/dut/otp_assert_inst/assert_lfsr_out_valid	0	10065409	10065409	0	0	0	1	Covered
/tb_top/dut/otp_assert_inst/assert_otp_latch_valid	0	10065409	10065409	0	0	0	1	Covered
/tb_top/dut/otp_assert_inst/assert_reset_n_valid	0	10065409	10065409	0	0	0	1	Covered
/tb_top/dut/otp_assert_inst/assert_user_in_valid	0	10065409	10065409	0	0	0	1	Covered
/tb_top/dut/otp_assert_inst/assert_user_latch_valid	0	10065409	10065409	0	0	0	1	Covered
/tb_top/dut/otp_assert_inst/assert_user_out_valid	0	10065409	10065409	0	0	0	1	Covered

Figure 8 Assertions

The scoreboard maintains reference models and validates all DUT operations against expected behavior. It tracks authentication, timing, security, and display system validation

5.4 Waveform Analysis

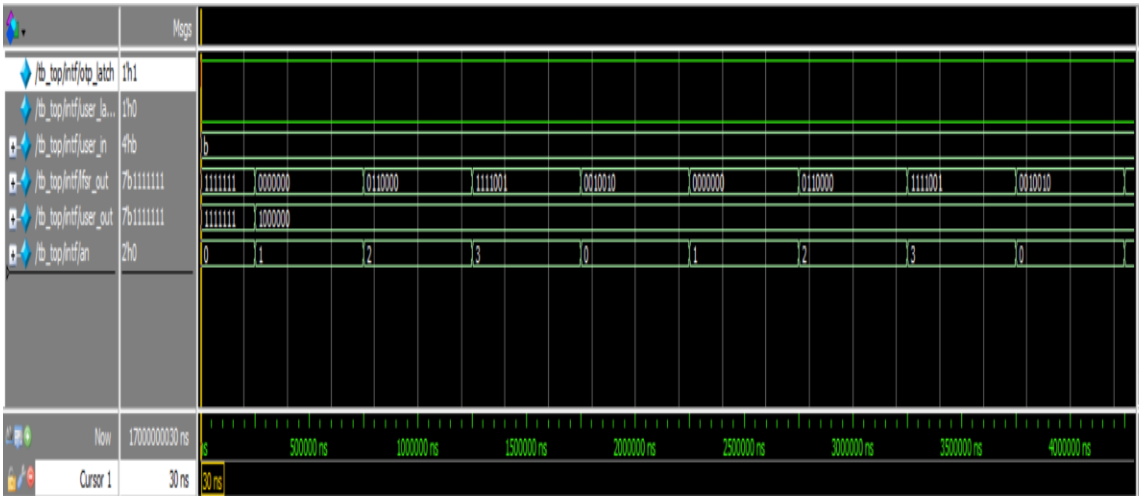

Msgs
/tb_top/intf/otp_latch 1'h1
/tb_top/intf/user_la... 1'h0
/tb_top/intf/user_in 4'hb
/tb_top/intf/lfsr_out 7'b1111111
/tb_top/intf/user_out 7'b1111111
/tb_top/intf/an 2'h0
b
1111111 0000000 0110000 1111001 0010010 0000000 0110000 1111001 0010010
1111111 1000000
0 1 2 3 0 1 2 3 0
Now 17000000030 ns
Cursor 1 30 ns
500000 ns 1000000 ns 1500000 ns 2000000 ns 2500000 ns 3000000 ns 3500000 ns 4000000 ns
30 ns

Figure 9 Otp Latch high

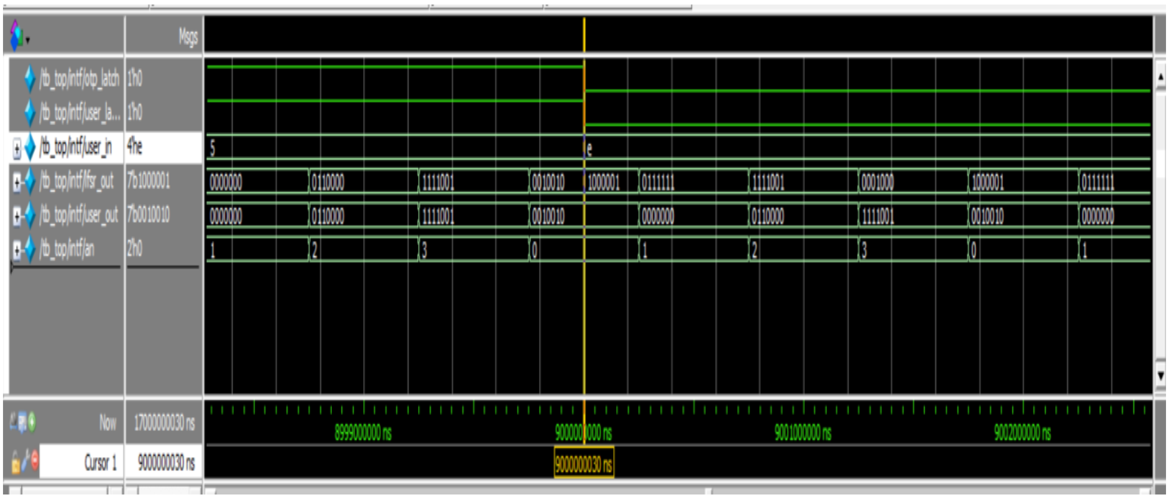


Figure 10 LFSR status

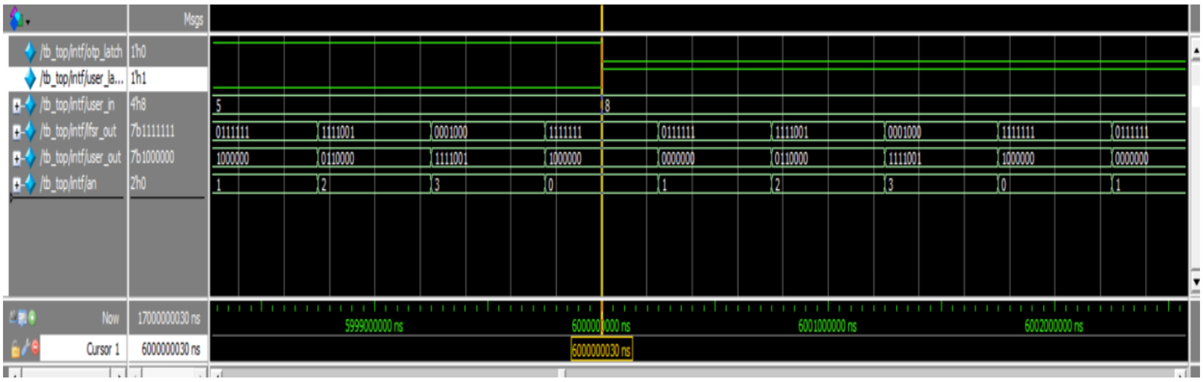


Figure 11 User input 8

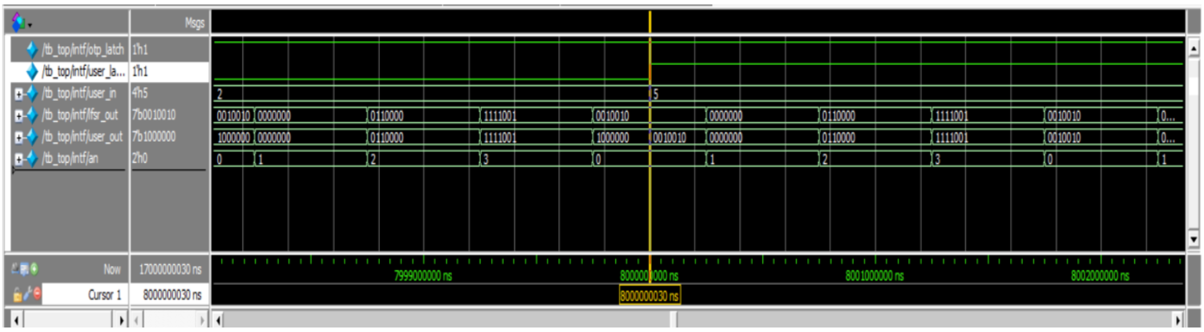

Msgs
/tb_top/intf/otp_latch 1'h1
/tb_top/intf/user_la... 1'h1
/tb_top/intf/user_in 4'h5
/tb_top/intf/lfsr_out 7'b0010010
/tb_top/intf/user_out 7'b1000000
/tb_top/intf/an 2'h0
Now 17000000030 ns
Cursor 1 8000000030 ns
7999000000 ns
8000000000 ns
8001000000 ns
8002000000 ns
8000000030 ns

Figure 12 User input 5

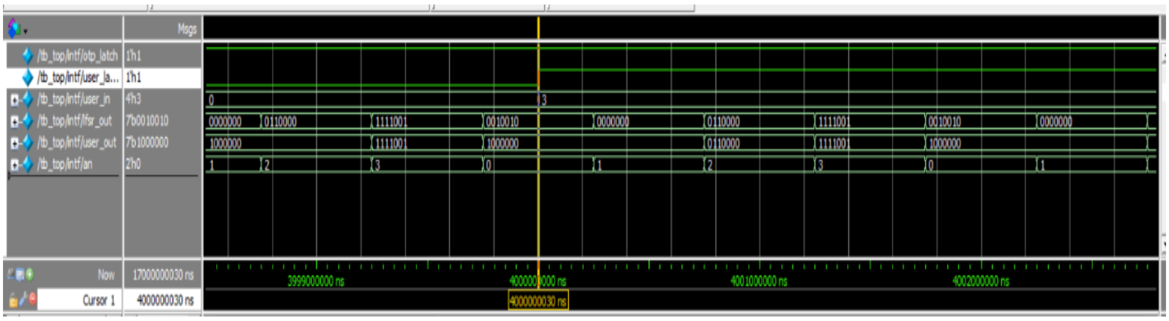

Msgs
/tb_top/intf/otp_latch 1'h1
/tb_top/intf/user_la... 1'h1
/tb_top/intf/user_in 4'h3
/tb_top/intf/lfsr_out 7'b0010010
/tb_top/intf/user_out 7'b1000000
/tb_top/intf/an 2'h0
Now 17000000030 ns
Cursor 1 4000000030 ns
3999000000 ns
4000000000 ns
4001000000 ns
4002000000 ns
4000000030 ns

Figure 13 User input 3

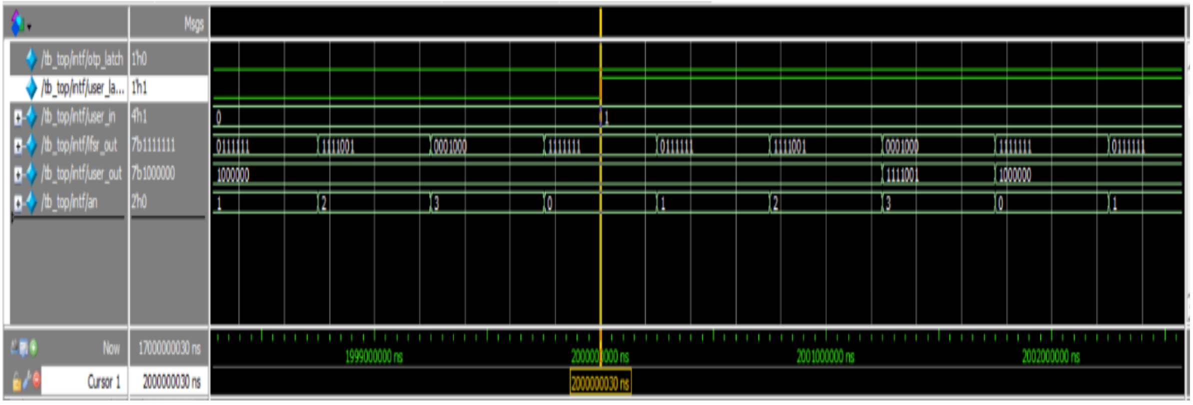

Msgs
/tb_top/intf/otp_latch 1'h0
/tb_top/intf/user_la... 1'h1
/tb_top/intf/user_in 4'h1
/tb_top/intf/lfsr_out 7'b1111111
/tb_top/intf/user_out 7'b1000000
/tb_top/intf/an 2'h0
Now 17000000030 ns
Cursor 1 2000000030 ns
1999000000 ns
2000000000 ns
2001000000 ns
2002000000 ns
2000000030 ns

Figure 14 User input 1

Chapter 6: CONCLUSION

This document presented a comprehensive verification approach for an OTP Authenticator design using UVM methodology. The verification environment successfully validated the security authentication functionality across independent clock domains, ensuring robust OTP generation between master (50 MHz) and display (2 kHz, 0.5 Hz) timing domains.

The verification strategy employed multiple complementary techniques to ensure comprehensive coverage:

- UVM-based testbench architecture with active/passive agents for input stimulus and output monitoring
- Reference model-based scoreboard implementing LFSR, authentication, timing, and security checking
- Functional coverage collection through subscriber component measuring scenario completeness
- SystemVerilog assertions for continuous protocol, security, and timing monitoring
- Comprehensive test plan covering basic functionality, authentication, security, timing, and edge cases
- Waveform analysis validating authentication sequences, timing relationships, and display behavior

APPENDICES

Appendix A: File Structure

The verification environment consists of the following SystemVerilog files:

File Name	Description
tb_top.sv	Top-level testbench integrating OTP DUT, interface, and UVM test execution.
otp_if.sv	Interface defining OTP DUT connections with clocking blocks for stimulus and sampling.
otp_sequence_item.sv	Transaction class containing OTP-related fields for sequence-driver communication.
otp_sequence.sv	UVM sequences generating stimulus patterns for OTP latch, input, mismatch, and expiry scenarios.
otp_sequencer.sv	Sequencer managing transaction flow between sequence and driver.
otp_driver.sv	Driver converting sequence transactions into pin-level DUT signals with proper timing.
otp_active_monitor.sv	Monitor capturing input-side signals (otp_latch, user_latch, user_in) for checking and coverage.
otp_passive_monitor.sv	Monitor observing DUT outputs (lfsr_out, user_out, an) for validation and coverage.
otp_active_agent.sv	Active agent encapsulating sequencer, driver, and input monitor for stimulus generation.
otp_passive_agent.sv	Passive agent containing only output monitor for DUT response observation.
otp_scoreboard.sv	Reference model and checker validating LFSR logic, authentication, and security behavior.
otp_subscriber.sv	Functional coverage collector measuring input/output coverage and verification completeness.
otp_env.sv	UVM environment integrating agents, scoreboard, and coverage subscriber.
otp_test.sv	Test classes defining OTP verification scenarios such as authentication, expiry, and security policy tests.

otp_assertions.sv	SystemVerilog assertions monitoring protocol, timing, and state transition violations.
otp_bind.sv	Bind file linking assertion modules to the DUT for active checking during simulation.
project_configs.sv	Global configuration file defining OTP parameters, data types, and shared constants.

Appendix B: Test plan Links

Test plan: Test Plan

Coverage Plan: Coverage Plan

Assertion Plan: Assertion Plan

DUT Bugs: Corrections in DUT