Vending Machine Controller Verification Plan

1. Objective

The objective of this verification plan is to verify the **functional correctness**, **timing accuracy**, **interface compliance**, and **configurability** of the **Vending Machine Controller IP** using the **Universal Verification Methodology (UVM)**. The goal is to ensure the design behaves as specified in all possible use cases, including edge cases and error conditions, with a reusable and coverage-driven verification environment.

2. Scope

- Validate operation in **Reset**, **Configuration**, and **Operation** modes.
- Ensure correct behavior of **APB-based register interface**.
- Check **item selection** and **currency validation logic**.
- Verify **dispense and change generation** logic.
- Ensure **assertion coverage**, **functional coverage**, and **code coverage (statement & branch)** are achieved for sign-off.

3. Design Interfaces to Verify

Interface	Signal Group	Clock Domain
System Interface	clk, rstn, cfg_mode	clk (100MHz)
APB Configuration Bus	pclk, prstn, paddr, psel, pwrite, pwdata, prdata, pready	pclk (50MHz)
Currency Interface	currency_valid, currency_value	Asynchronous (10KHz–50MHz)
Item Select Interface	item_select, item_select_valid	Asynchronous (10KHz–50MHz)
Output Interface	item_dispense_valid, item_dispense, currency_change	clk (100MHz)

4. Verification Strategy

Aspect	Strategy
Methodology	UVM (Universal Verification Methodology)
Language	SystemVerilog
Checker	Reference Model & Scoreboard
Coverage	Functional + Code (line, toggle, condition, FSM transitions)
Stimulus Generation	Constrained-random for currency and item input
Reusability	Configurable and reusable UVM testbench agents for each functional interface

5. Verification Components

Component	Description
APB Agent	Configurable master for write/read
Currency Agent	Generates valid asynchronous currency_value and currency_valid pulses
Item Selector Agent	Generates item selection requests to DUT.
Dispense Agent	Passive monitor checking output accuracy and timing.
Monitor	Captures item selection, currency, and output decisions
Scoreboard	Compares expected and actual dispense/change outputs
Checker/Assertions	Ensures timing, mode transitions, change calculation, latency, and FSM correctness
Coverage Collector	Records coverage points for functional and corner cases

6. Key Functional Checks

Feature	Description
Reset Behavior	All registers and state machines must initialize to default
APB Configuration	Proper data transfer and address decoding.
Item Setup	Proper count and value configuration for items
Currency Logic	Accepts supported values only (5, 10, 15, 20, 50, 100)
Dispense Logic	Dispense only when enough value is inserted
Change Logic	Proper change dispensed if excess amount inserted
Out-of-stock Case	Emits empty indication and refunds currency
Latency	10 system clocks from last valid currency to output
Single Dispense	Only one output decision per transaction

7. Agent Statergy

Agent Name	Interface	Activity Description	Type
cfg_agent	APB Configuration Interface	• Drives APB transactions (read/write) • Monitor checks address decoding, write-read alignment, pready timing	Active
currency_agent	Currency Input Interface	• Sends valid and random currency values asynchronously • Stimulates FSM with valid, underpaid, overpaid, and noisy currency inputs	Active
item_agent	Item Selection Interface	• Sends valid and invalid item selections • Triggers FSM decisions related to item availability and price	Active

dispense_agent	Dispense Output Interface	• Passively observes item_dispense_valid, item_dispense, and currency_change • Reports outputs to scoreboard and timing checker	Passive

8. TEST Cases

1. Sanity Tests

Testcase Name	Description
tc_sanity	Power-on Reset → Config mode → Add 2 items → Select & buy item

2. Configuration Mode Tests

Testcase Name	Description
tc_cfg_write_read	Write to and read back from APB registers
tc_cfg_max_items	Configure up to 1024 items and verify correctness
tc_cfg_clear_dispense_count	Validate disp_items is zeroed after reset

3. Functional Operation Mode Tests

Testcase Name	Description
tc_single_item_buy	Buy a single item with exact currency
tc_multiple_currency_insert	Insert currency in multiple parts (5 + 10 + 5)
tc_item_unavailable	Select item with 0 stock, validate change refund
tc_overpay_with_change	Insert extra currency and verify correct change
tc_random_currency_sequence	Insert random valid sequences for multiple items

4. Negative and Corner Cases

Testcase Name	Description
tc_unsupported_currency	Insert unsupported currency (e.g., 30)
tc_double_dispense_check	Ensure only one dispense per transaction
tc_asynchronous_violation	Insert currency while APB config is happening (should be ignored or flagged)
tc_invalid_item_select	Select item index beyond configured no_of_items

5. Performance & Latency

Testcase Name	Description
tc_latency_check	Validate dispense within 10 clocks from last currency
tc_back_to_back_operations	Multiple transactions in short duration

6. Coverage Goals

Coverage Type	Target
Functional Coverage	100% (Currency, Item, Config, Modes)
Code Coverage	100% (Statement, Branch)

7. Waveform Checks & Logs

- Use of log messages and waveform markers in:
    - Currency transaction
    - Item selection & dispense
    - Change calculation
- Record APB access trace

8. Directed Testcases:

Testcase Name	Description
tc_exact_currency_match	Select an item of value ₹20, insert ₹20 → Check item dispensed, no change
tc_excess_currency	Select item ₹15, insert ₹20 → Check item dispensed and ₹5 returned
tc_item_out_of_stock	Select item with avail_items = 0 → Check no dispense, full change returned
tc_config_max_items	Configure 1024 items and select item at last index → Should behave normally
tc_change_zero	Select item ₹50, insert ₹50 → Item should dispense, zero change

Testbench architecture