**Hardware Implementation  
SAHITYA KUSHWAHA  
May 2025**

**1. Introduction**

In this project, I implemented a basic Electronic Vending Machine using FPGA. The goal was to simulate the functionality of a vending machine that accepts inputs (like coin insertion or item selection) and produces appropriate outputs (like dispensing items or showing balance) using digital logic. This was done through both:

1. Using physical input/output ports on an FPGA board.
2. Using virtual I/O for simulation and verification.

**2. Tools and Requirements**

**2.1 Xilinx Vivado**

Xilinx Vivado is the core development tool used to design, simulate, and implement the vending machine’s logic. Vivado allows us to write Verilog code, perform RTL simulation, analyse timing, assign I/O pins, and generate the final bitstream needed for FPGA programming.

**2.2 Real Digital Boolean Board**

The Boolean Board is used to deploy and test the vending machine logic in real-time. It provides physical buttons, switches, and LEDs that serve as inputs and outputs during hardware testing.

**3. What Did I Do?**

**3.1 Basic Overview**

I developed Verilog code to model the core logic of a vending machine. The machine accepts coin inputs, processes item selection, verifies balance, and dispenses items accordingly. After verifying the logic through simulation, I mapped the I/O to physical switches (for inputs like coins and selections) and LEDs (for outputs like "item dispensed" or "insufficient balance").

I generated the constraints file (.xdc) specifying voltage levels (e.g., 3.3V), ran synthesis and implementation, and created a bitstream. After uploading the bitstream to the FPGA board, I validated functionality using actual hardware buttons and LEDs.

**3.2 Step-by-Step Process**

**Step 1: Code and Testbench**

Wrote Verilog modules to implement the vending machine logic. Created a testbench to simulate scenarios like inserting coins, choosing items, and handling insufficient balance.

**Step 2: Simulation**

Ran behavioural simulations to verify correctness of the vending logic. Checked if the machine dispenses only when the correct amount is entered.

**Step 3: RTL Analysis & I/O Planning**Used RTL analysis to understand the gate-level design. Created an .xdc constraints file to assign appropriate FPGA pins to buttons (inputs) and LEDs (outputs). Set voltage standards using board specifications.

**Step 4: Synthesis and Implementation**Performed synthesis to generate a netlist from RTL code. Ran implementation to map the design to the physical FPGA logic blocks.

**Step 5: Connect Hardware**Connected the FPGA board to the system via USB and ensured power supply was adequate.

**Step 6: Generate Bitstream & Program Device**Generated the bitstream from Vivado and uploaded it to the board using the Hardware Manager.

**Step 7: Test on Hardware**Inserted different input combinations using buttons and verified output behaviour on LEDs—confirming the vending machine logic works in real-time.

**4. What Did I Learn?**

* How to write modular Verilog code for a multi-state digital system like a vending machine.
* How to simulate real-world scenarios using a testbench.
* How to interpret RTL schematics and plan I/O pin mapping.
* How to generate a hardware-compatible bitstream from HDL code.
* How to interact with physical hardware and debug logic in real-time.
* How digital systems handle logic flow, state transitions, and error handling.