Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)
MoE, UGC & AICTE Approved
NAAC A++ Accredited

An internship report submitted by

NIVETHITHA R – URK21CS1001

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

under the supervision of

Dr. Esther Daniel
Mr. Abhishek Nandy

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec-3 of the UGC Act, 1956)
Karunya Nagar, Coimbatore - 641 114. INDIA

Team Name: Singleton

Team Members

The project was carried out by :

- Nivethitha R – nivethithar21@karunya.edu.in

Mentor

I would like to express my gratitude to my mentor for providing constant support and guidance throughout the internship.

- Dr.Esther Daniel(Academic Mentor) -estherdaniel@karunya.edu
- Mr. Abhishek Nandy(Industrial Mentor) - abhisheknandy@theprograms.in

ABSTRACT

ATPs are designed to collect payment from consumers by cash, cheque, or DD. It will be unmanned and can be operated by the customers 24/7. It accepts cash/cheque/DD/pay order, issues an acknowledgment on every payment made and is a touchscreen and multimedia-based system. When the customer places the voucher/bill in the designated slot under the barcode scanner, the ATP will automatically get started. Suitable prompts are provided for guidance.

The ATP captures data from the voucher/bill and will display parameters on the monitor. A customer needs to choose the mode of payment. Once the amount is confirmed by the customer, the ATP will give directions on cash/cheque insertion. Parameters such as cheque number etc are read from the MICR fields and an acknowledgment is issued to the customer with the bill.

Introduction:

The ATP machine is an automatic machine which is used for the payment of electricity bill. ATPs are designed to collect payment from consumers by cash, cheque, or DD. It will be unmanned and can be operated by the customers 24/7. It accepts cash/cheque/DD/pay order, issues an acknowledgment on every payment made and is a touchscreen and multimedia-based system. When the customer places the voucher/bill in the designated slot under the barcode scanner, the ATP will automatically get started. Suitable prompts are provided for guidance. The ATP captures data from the voucher/bill and will display parameters on the monitor. A customer needs to choose the mode of payment. Once the amount is confirmed by the customer, the ATP will give directions on cash/cheque insertion. Parameters such as cheque number etc are read from the MICR fields and an acknowledgment is issued to the customer with the bill.

The ATP Machine has a touch screen, through which the consumers can enter their Service Number and pay the amount and get a receipt/statement of the amount paid. The machine has the provision to take currency notes. A separate slot will accept cheques in the same manner.

Machines of this new category are generally called Anytime payment kiosks. Apparently similar to the development of traditional mobile phones into smartphones, ATP machines have also progressively, though at a much slower pace, evolved into smart machines. Newer technologies at a lower cost of adoption, such as the large digital touch display, internet connectivity, cameras and various types of sensors, more cost-effective embedded computing power, digital signage, various advanced payment systems, and a wide range of identification technology (NFC, RFID, etc) have contributed to this development.. Integrated sensors and cameras also represent a source of such data as customer demographics and other locality-specific information. It also enables better customer engagement.

APPROACH TO SOLVE THE PROBLEM

Define the System Requirements:

- Determine the input and output requirements of the controller.
- Identify the necessary control signals and data paths.
- Specify any specific functionalities or constraints.

Design the Finite State Machine (FSM):

- Identify the different states the controller can be in.
- Define the state transition conditions and state outputs.
- Implement the FSM using Verilog constructs like always block and case statement.

Implement the Input Handling:

- Identify the input sources
- Design input registers to capture and store the input data.
- Implement the necessary logic to validate and process the inputs.

Implement the Output Handling:

- Identify the output devices
- Design output registers to store the data to be displayed or printed.
- Implement the necessary logic to control the output devices.

Implement the Billing Logic:

- Determine the billing algorithm and calculations.
- Design registers to store relevant billing information.
- Implement the necessary logic to calculate the bill amount.

Implement the Payment Processing:

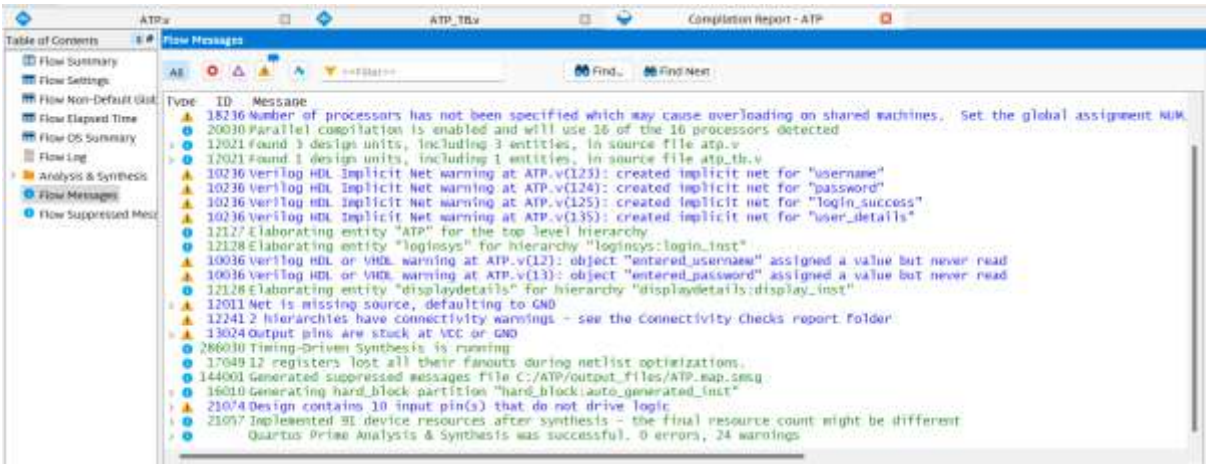
- Design registers to store payment information.
- Implement the logic to verify payment methods.
- Update the billing information based on the payment received.

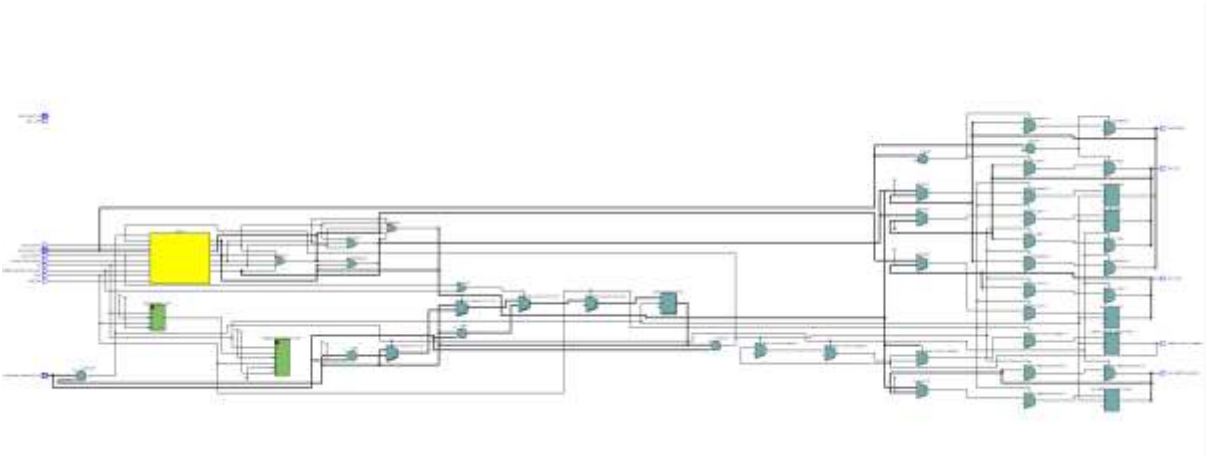
Test and Debug:

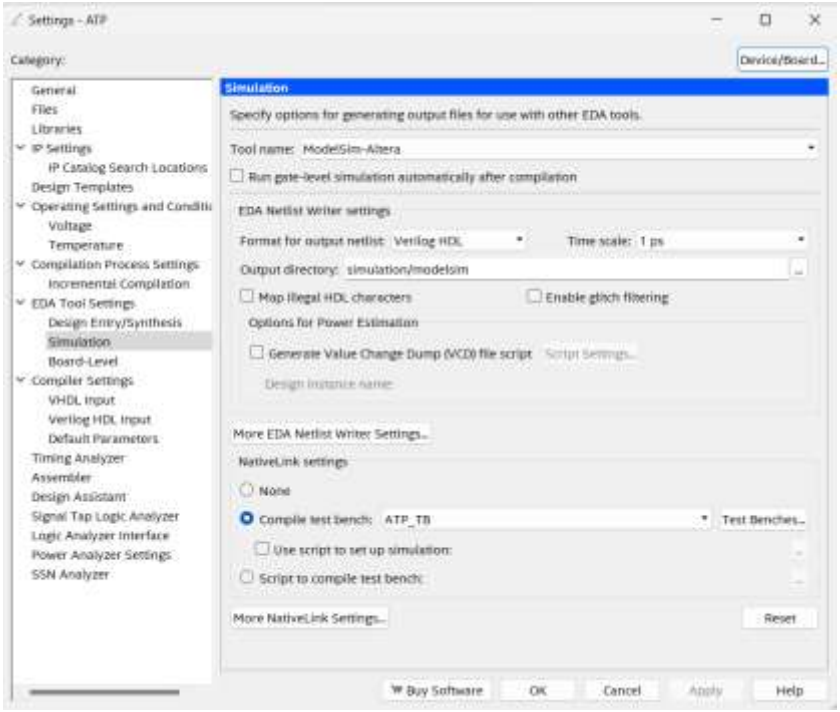
- Create testbench modules to simulate the controller's functionality.
- Generate test cases to verify different scenarios.
- Simulate and debug the design to ensure correct behavior.

Synthesize and Implement:

- Use a synthesis tool (such as Xilinx Vivado or Intel Quartus) to convert the Verilog code into a gate-level netlist.
- Implement the design on the target FPGA device or ASIC.


ATP.v
ATP_TB.v
Compilation Report - ATP
Table of Contents
Flow Summary
Flow Settings
Flow Non-Default Glob
Flow Elapsed Time
Flow OS Summary
Flow Log
Analysis & Synthesis
Flow Messages
Flow Suppressed Mes
Flow Messages
All
Find...
Find Next
Type ID Message
18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM
20030 Parallel compilation is enabled and will use 16 of the 16 processors detected
12021 Found 3 design units, including 3 entities, in source file atp.v
12021 Found 1 design units, including 1 entities, in source file atp_tb.v
10236 Verilog HDL Implicit Net warning at ATP.v(123): created implicit net for "username"
10236 Verilog HDL Implicit Net warning at ATP.v(124): created implicit net for "password"
10236 Verilog HDL Implicit Net warning at ATP.v(125): created implicit net for "login_success"
10236 Verilog HDL Implicit Net warning at ATP.v(135): created implicit net for "user_details"
12127 Elaborating entity "ATP" for the top level hierarchy
12128 Elaborating entity "loginsys" for hierarchy "loginsys:login_inst"
10036 Verilog HDL or VHDL warning at ATP.v(12): object "entered_username" assigned a value but never read
10036 Verilog HDL or VHDL warning at ATP.v(13): object "entered_password" assigned a value but never read
12128 Elaborating entity "displaydetails" for hierarchy "displaydetails:display_inst"
12011 Net is missing source, defaulting to GND
12241 2 hierarchies have connectivity warnings - see the Connectivity Checks report folder
13024 Output pins are stuck at VCC or GND
286030 Timing-Driven Synthesis is running
17049 12 registers lost all their fanouts during netlist optimizations.
144001 Generated suppressed messages file C:/ATP/output_files/ATP.map.smsg
16010 Generating hard_block partition "hard_block:auto_generated_inst"
21074 Design contains 10 input pin(s) that do not drive logic
21057 Implemented 91 device resources after synthesis - the final resource count might be different
Quartus Prime Analysis & Synthesis was successful. 0 errors, 24 warnings




Settings - ATP
Category:
Device/Board...
General
Files
Libraries
IP Settings
IP Catalog Search Locations
Design Templates
Operating Settings and Conditi
Voltage
Temperature
Compilation Process Settings
Incremental Compilation
EDA Tool Settings
Design Entry/Synthesis
Simulation
Board-Level
Compiler Settings
VHDL Input
Verilog HDL Input
Default Parameters
Timing Analyzer
Assembler
Design Assistant
Signal Tap Logic Analyzer
Logic Analyzer Interface
Power Analyzer Settings
SSN Analyzer
Simulation
Specify options for generating output files for use with other EDA tools.
Tool name: ModelSim-Altera
Run gate-level simulation automatically after compilation
EDA Netlist Writer settings
Format for output netlist: Verilog HDL
Time scale: 1 ps
Output directory: simulation/modelsim
Map illegal HDL characters
Enable glitch filtering
Options for Power Estimation
Generate Value Change Dump (VCD) file script
Script Settings...
Design instance name:
More EDA Netlist Writer Settings...
NativeLink settings
None
Compile test bench: ATP_TB
Test Benches...
Use script to set up simulation:
Script to compile test bench:
More NativeLink Settings...
Reset
Buy Software
OK
Cancel
Apply
Help

DESIGN METHODOLOGY

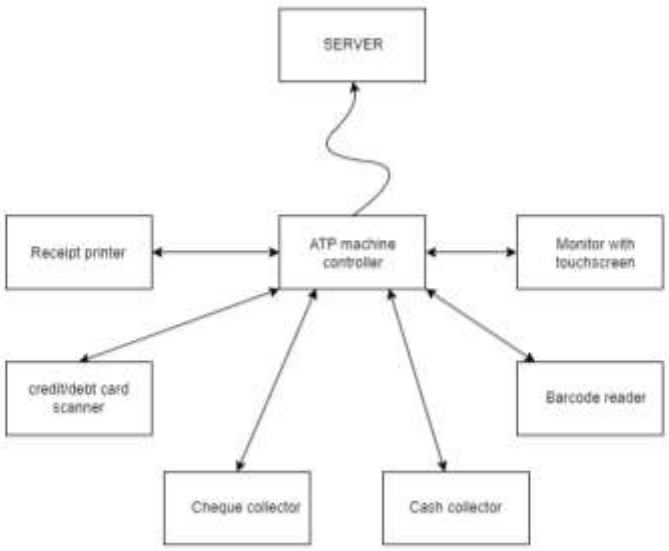
A) PROBLEM FORMULATION

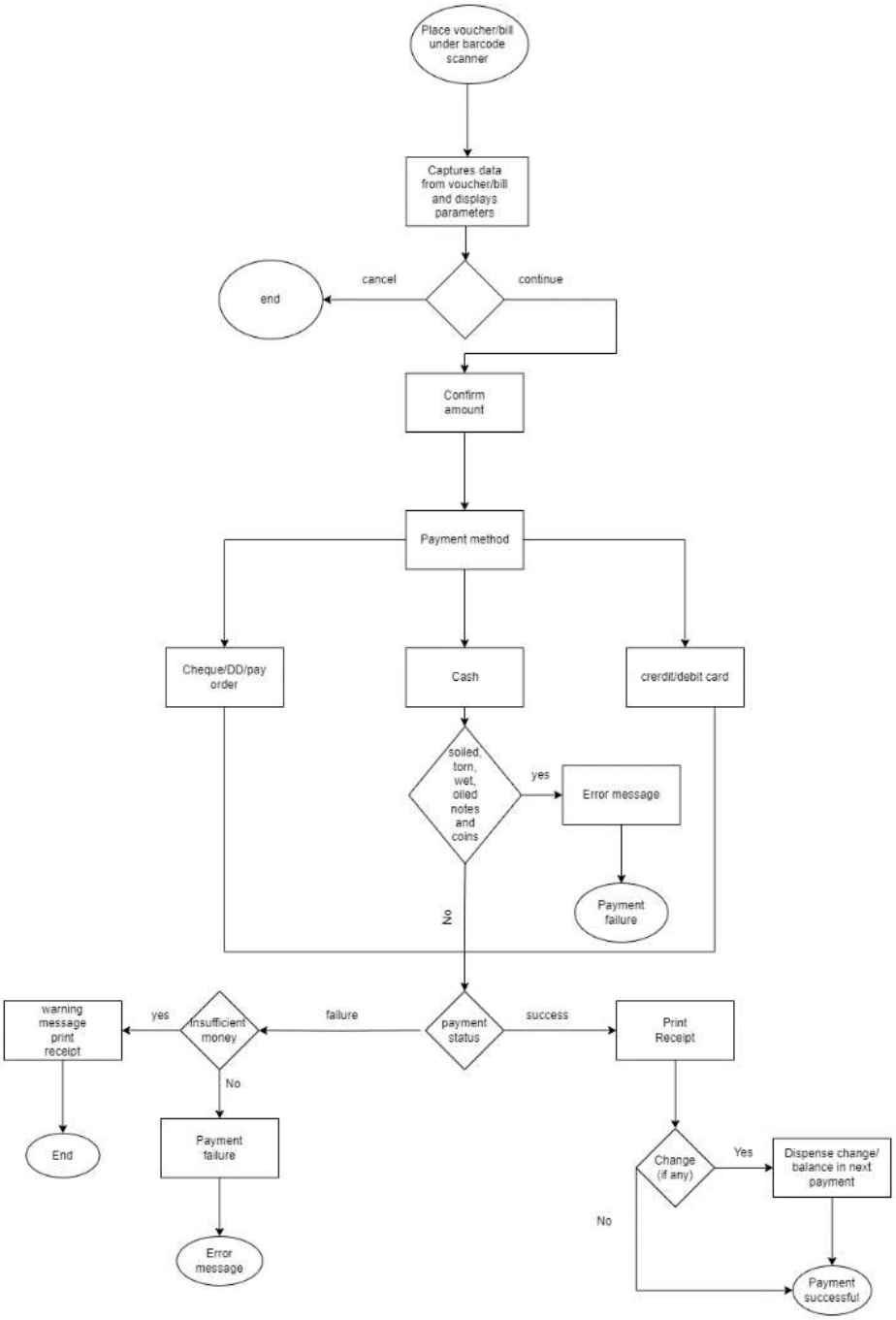
The problem is defined as the design of an ATP machine that accepts cash, cheque/DD/Pay orders. The machine should possess additional features of returning change when more money is paid or reduced in subsequent payment returning and money when the request is canceled. The machine asks the customer to select the payment method. Therefore, the payment is one of the inputs. There is also a cancel option which is another input. The machine checks the paid amount with the total amount to be paid and prints the receipt if both are equal. If the customer inserts more money than the total amount, the machine gives the receipt along with the change or adds it to a subsequent cycle. Receipt and change are therefore outputs of the machine. Money is returned when the request is canceled. Therefore, return money is another output. Two registers are needed, one to keep track of the amount of the present transaction and the other to keep track of the total amount in the machine. The registers are taken to be ten-bit width each. When a higher amount is inserted, change can be given only if the total amount is higher than the present transaction. In this way, the machine checks whether the change is available or not.

In order to meet the above specifications, a finite state machine approach is adopted. The ASM chart and state diagram are drawn, and the corresponding Verilog code is formulated using behavioral modeling.

The subsequent sections deal with elaborate descriptions of the flowchart and state diagram of the ATP machine

B) BLOCK DIAGRAM AND FLOWCHART OF ATP MACHINE:



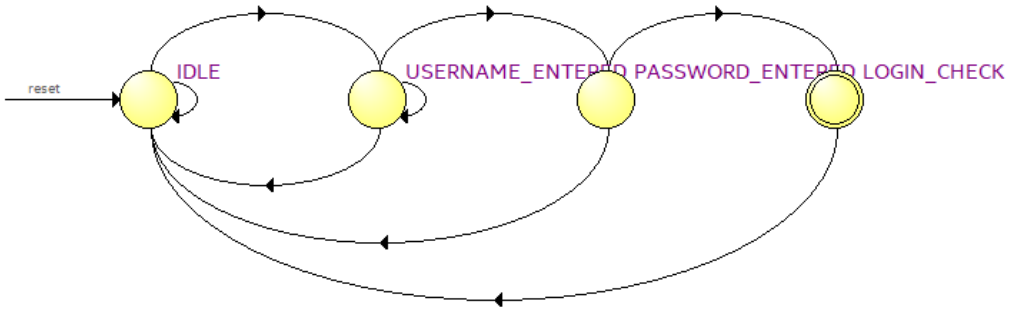

Place voucher/bill under barcode scanner
Captures data from voucher/bill and displays parameters
cancel
end
continue
Confirm amount
Payment method
Cheque/DD/pay order
Cash
crerdit/debit card
soiled, torn, wet, oiled notes and coins
yes
Error message
Payment failure
No
payment status
failure
success
yes
warning message print receipt
Insufficient money
No
End
Payment failure
Error message
Print Receipt
Change (if any)
Yes
Dispense change/ balance in next payment
No
Payment successful

C) STATE DIAGRAM OF ATP MACHINE:

The flow of signals that lead to various states is shown in subsequent figure.

STATES:

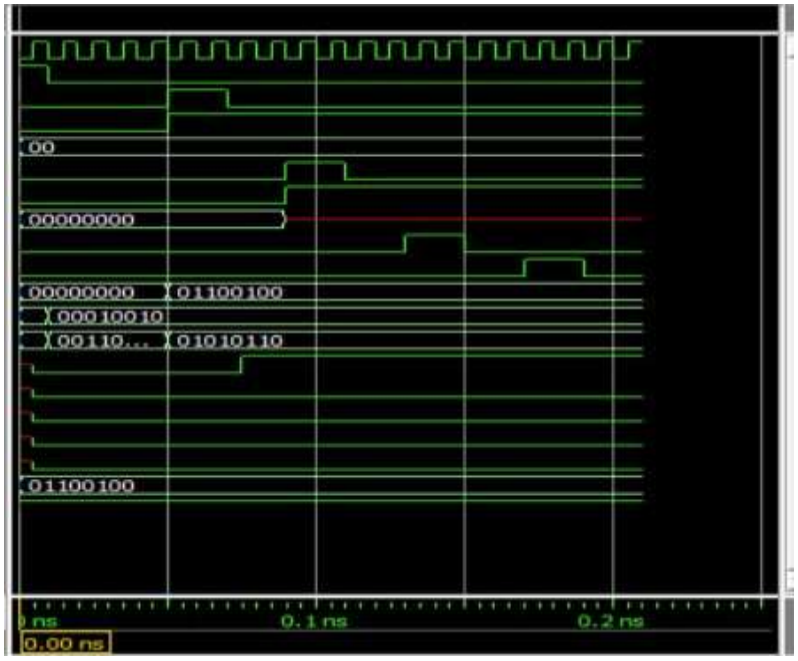
1. IDLE: The initial state of the ATP.
2. STARTED: The ATP has been started, and an acknowledgment is issued with the voucher data.
3. CONFIRM_AMOUNT: The customer confirms the amount to be paid.
4. INSERT_CASH_CHEQUE: The customer inserts cash/cheque.
5. PROCESSING: The ATP processes the cash/cheque.
6. COMPLETED: The transaction is completed, and an acknowledgment is issued with the total amount and electricity charge.



The state diagram through RTL viewer.

RESULTS AND DISCUSSION

The Verilog code has been formulated for the vending machine using behavioral modeling. The test bench has been written. The simulation has been performed. The simulation waveform is obtained as shown in the subsequent figure.



Compilation Report:

Flow Status	Successful - Fri Jul 14 19:46:49 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	ATP
Top-level Entity Name	ATP
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	27
Total pins	39
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

	Source State	Destination State	Condition
1	AUTHORIZED_STATE	IDLE_STATE	(!LessThan1).(!LessThan0).(reset) + (!LessThan1).(LessThan0) + (LessThan1)
2	AUTHORIZED_STATE	AUTHORIZED_STATE	(!LessThan1).(!LessThan0).(!reset)
3	CARD_PRESENT_STATE	PIN_VALID_STATE	(valid_pin).(!reset)
4	CARD_PRESENT_STATE	IDLE_STATE	(!valid_pin) + (valid_pin).(reset)
5	CASH_PRESENT_STATE	IDLE_STATE	(reset)
6	CASH_PRESENT_STATE	AUTHORIZED_STATE	(!reset)
7	DIGITAL_PAYMENT_S...	IDLE_STATE	(reset)
8	DIGITAL_PAYMENT_S...	AUTHORIZED_STATE	(!reset)
9	IDLE_STATE	MICR_VALID_STATE	(cheque_dd_present).(!card_present).(!reset)
10	IDLE_STATE	IDLE_STATE	(!digital_payment_present).(!cash_present).(!cheque_dd_present).(!card_present) + (!digital_payment_present).(!cash_present).(!cheque_dd_present).(card_present).(reset) + (!digital_payment_present).(!cash_present).(cheque_dd_present).(reset) + (!digital_payment_present).(cash_present).(reset) + (digital_payment_present).(reset)
11	IDLE_STATE	DIGITAL_PAYMENT...	(digital_payment_present).(!cash_present).(!cheque_dd_present).(!card_present).(!reset)
12	IDLE_STATE	CASH_PRESENT_S...	(cash_present).(!cheque_dd_present).(!card_present).(!reset)
13	IDLE_STATE	CARD_PRESENT_S...	(card_present).(!reset)
14	MICR_VALID_STATE	IDLE_STATE	
15	PIN_VALID_STATE	IDLE_STATE	(!card_type[1]).(reset) + (card_type[1])
16	PIN_VALID_STATE	AUTHORIZED_STATE	(!card_type[1]).(!reset)

CONCLUSION

The ATP machine was successful, with the additional features of printing receipts along with returning change when a higher amount is paid and returning total money when the request is canceled. The ATP machine is successful in meeting the specifications laid out prior to the design.

References:

https://www.apspdcl.in/atp-machine.jsp

https://www.intel.com/content/www/us/en/programmable/customertraining/webex/Verilog/presentation_html5.html

BOOK reference:

Verilog HDL: A Guide to Digital Design and Synthesis