# Anytime Electricity Bill Payment (ATP)

**Team Name: -** Intelligent insights

**Team Size: -**3

**College Name: -**Malla Reddy College of Engineering and Technology

Description: -

* Design an ATP to collect payment in different mode such as cash/DD/cheque from the customer 24/7.
* ATP capture data from bill/voucher and display on monitor.
* ATP able to show the payment modes (i.e Cash/DD/cheque).
* Once the payment is confirmed by the user/customer, customer should able to know the bill status.
* ATP should accept 10,20,50,100,200,500 notes only (In cash Mode).
* ATP should not accept soiled, torn, wet, oiled notes and coins.
* Parameters such as cheque number etc are read from MICR field and an acknowledgment is issued to the customer with the bill.
* Excess amount paid will be adjected in the subsequent cycles and any short payment will lead to disconnection of your electricity line without any information.

**State machine: -**

* The state machine has 5 states: -

i)Idle state

ii)Prompt state

iii)Scan state

iv) Payment state

v) Acknowledgment state

**i) Idle state: -** The system is in an idle state, waiting for user input.

**ii) Prompt State: -** The system displays a prompt on the touchscreen and waits for a response.

**iii) Scan State: -** The system is scanning a barcode for voucher detection.

**iv) Payment state: -** The system is in the payment phase, handling cash insertion and confirming payment.

**v)Acknowledgment state: -** The system acknowledges the completion of the transaction.

**->ATP functionality divide into 6 module: -**

* User Interface Module
* Barcode Scanner Module
* Cash Handling Module
* Payment Processing Module
* Acknowledgment Module
* ATP System Module
* **User Interface Module:** -

The `User Interface` module handles the user interface of the ATP system. It captures inputs from the touchscreen and buttons and provides outputs for touchscreen prompts and button selections. On each rising edge of the clock or reset signal, it updates the `touchscreen prompt` and `button selection` based on the corresponding input values.

* **Barcode Scanner Module**: -

The `Barcode Scanner` module represents a barcode scanner. It takes inputs for barcode detection and barcode data. When a barcode is detected (`barcode detection` is high), it sets the `voucher detected` signal to 1 and captures the barcode data in the `captured data` signal. If no barcode is detected, both `voucher detected` and `captured data` are set to 0.

* **Cash Handling Module: -**

The `Cash Handling` module handles the cash handling functionality of the ATP system. It takes inputs for cash insertion, denomination selection, and cash confirmation. On each rising edge of the clock or reset signal, it checks if cash is inserted (`cash insertion` is not 0). If cash is inserted, it sets the `cash accepted` signal based on the `cash confirmation` signal and stores the inserted cash amount. The `inserted cash amount output` is updated with the current inserted cash amount.

* **Payment Processing Module:** -

The `Payment Processing` module handles the payment processing logic. It takes inputs for the confirmed amount, inserted cash amount, and payment confirmation. On each rising edge of the clock or reset signal, it calculates the adjusted amount by subtracting the inserted cash amount from the confirmed amount. It then checks if the payment is confirmed. If the payment is confirmed, it checks the adjusted amount and sets the appropriate flags (`excess payment`, `short payment`, `disconnection`) based on the adjusted amount. If the payment is not confirmed, it sets the `disconnection` flag to 1.

* **Acknowledgment Module: -**

The `Acknowledgment` module handles the acknowledgment functionality. It takes inputs for transaction completion and transaction amount. On each rising edge of the clock or reset signal, it checks if a transaction is complete. If a transaction is complete, it sets the `acknowledgment issued` signal to 1 indicating that an acknowledgment is issued.

* **ATP System Module: -**

The ATP system operates by capturing inputs from the user interface, barcode scanner, and cash handling modules. It processes the inputs using the payment processing module and generates appropriate outputs for touchscreen prompts, button selections, voucher detection, captured barcode data, cash acceptance, inserted cash amount, adjusted amount, excess payment, short payment, disconnection, and acknowledgment issuance.

**Approach: -**

Approach to the project

We have approached the project with proper guidance from the college mentors as well as from Intel. We have gone through the internet for more information which made us complete the project successfully. following are the steps that we have gone through.

step 1: basic knowledge of the problem statement (ATP).

step 2: learned the language in which the code is to be written (Verilog HDL)

step 3: Developed the Block diagram of ATP.

step 4: Developed the FSM for ATP.

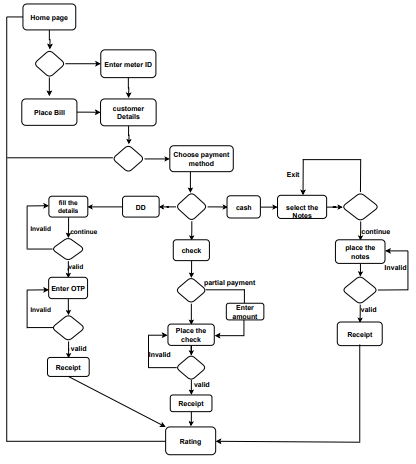
step 5: Code and Testbench generation.

step 6: compilation of the code.

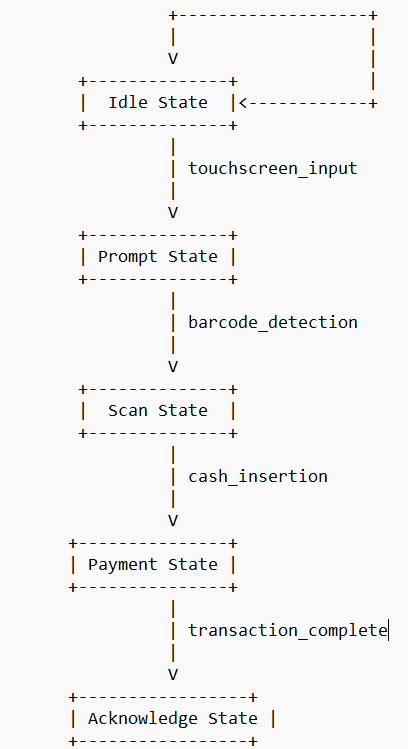
step 7: Preparing proper documentation.

step 8: Submission of the code in GitHub.

**Block Diagram: -**

****

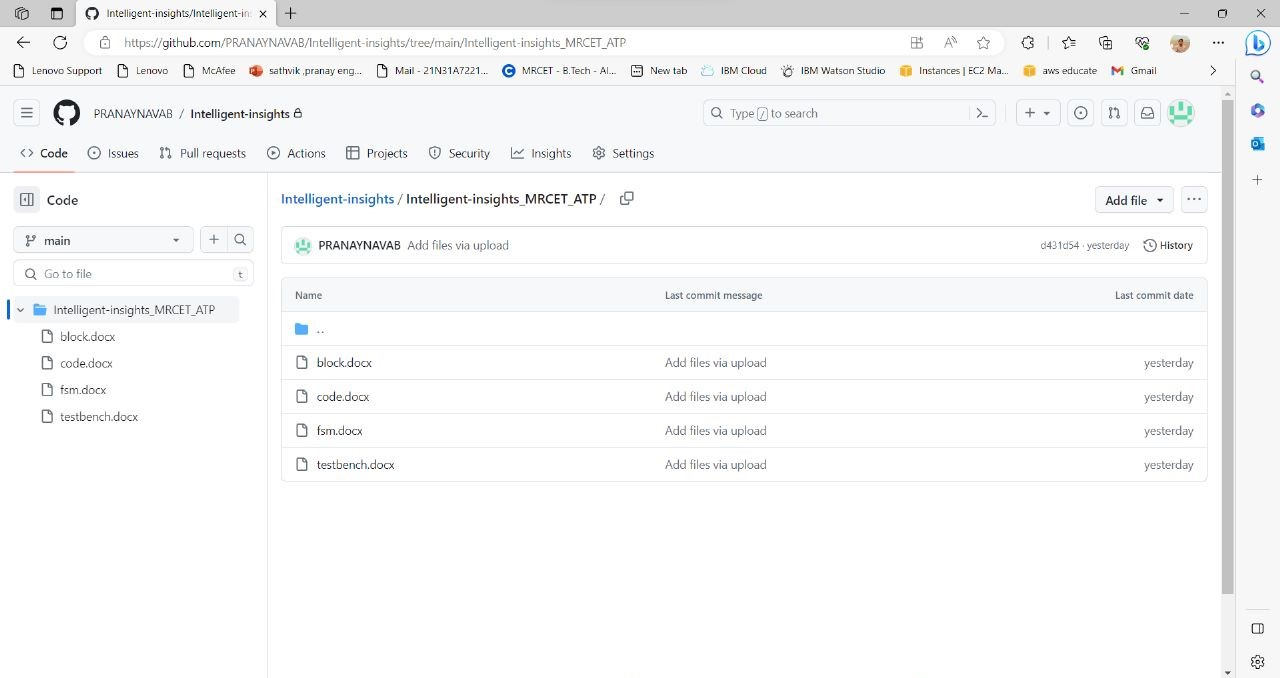
**FSM Model: -**

****

**Git Hub: -**

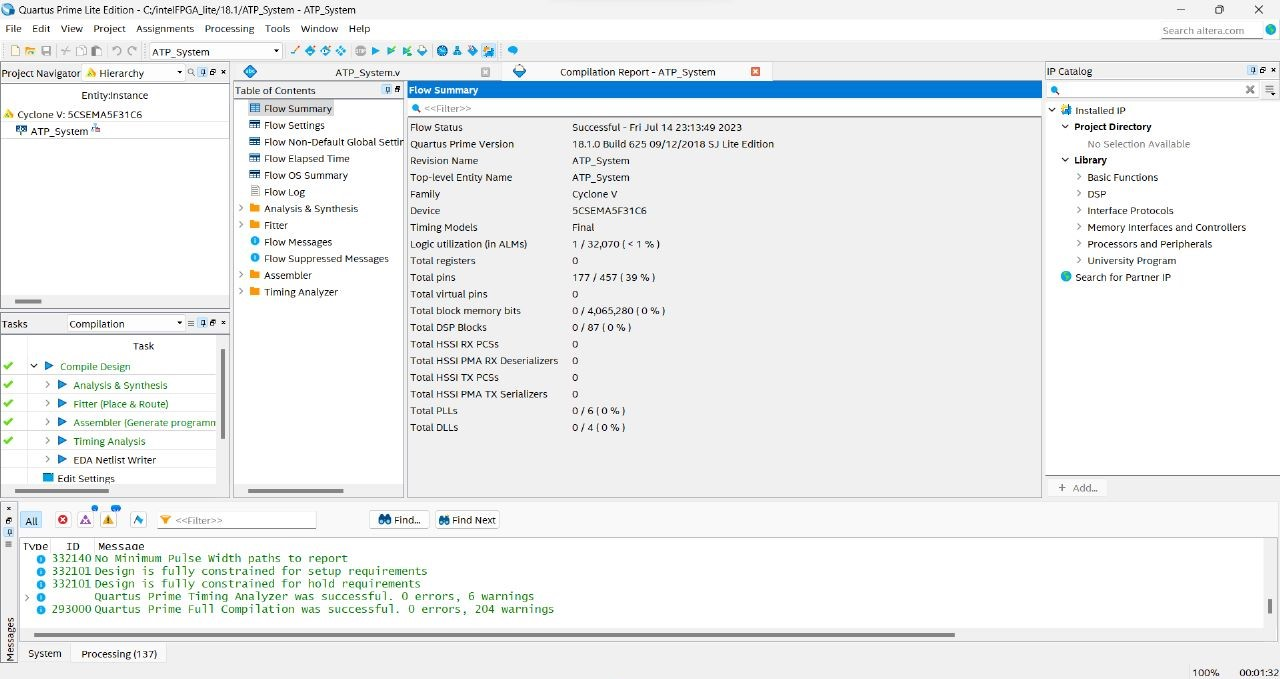
**Link: -** https://github.com/PRANAYNAVAB/Intelligent-insights.git

**Screenshot: -**



**Result: -**

* It took 1 Logic utilization which is less than 1%
* It consists of 177 pins.



**Implementation: -**

1. Initialize all signals and variables to their default values.

2. User Interface:

- On each rising edge of the clock or reset, check if the system is in a reset state.

- If in a reset state, set the touchscreen prompt and button selection to their default values (0).

- If not in a reset state, update the touchscreen prompt and button selection based on the inputs from the touchscreen and buttons.

3. Barcode Scanner:

- On each rising edge of the clock or reset, check if the system is in a reset state.

- If in a reset state, set the voucher detected and captured data to their default values (0).

- If not in a reset state, check if a barcode is detected.

- If a barcode is detected, set the voucher detected to 1 and capture the barcode data.

- If no barcode is detected, set the voucher detected and captured data to 0.

4. Cash Handling:

- On each rising edge of the clock or reset, check if the system is in a reset state.

- If in a reset state, set the cash accepted, inserted cash amount, and inserted cash amount output to their default values (0).

- If not in a reset state, check if cash is inserted.

- If cash is inserted, set the cash accepted based on the cash confirmation and store the inserted cash amount.

- Update the inserted cash amount output with the current inserted cash amount.

5. Payment Processing:

- On each rising edge of the clock or reset, check if the system is in a reset state.

- If in a reset state, set the adjusted amount, excess payment, short payment, disconnection, and adjusted amount variables to their default values (0).

- If not in a reset state, calculate the adjusted amount by subtracting the inserted cash amount from the confirmed amount.

- Check if payment is confirmed.

- If payment is confirmed, check the adjusted amount:

- If the adjusted amount is greater than 0, set the short payment flag to 1 and clear the excess payment and disconnection flags.

- If the adjusted amount is less than 0, set the excess payment flag to 1 and clear the short payment and disconnection flags.

- If the adjusted amount is equal to 0, clear all payment-related flags.

- If payment is not confirmed, set the disconnection flag to 1 and clear all other payment-related flags.

6. Acknowledgment:

- On each rising edge of the clock or reset, check if the system is in a reset state.

- If in a reset state, set the acknowledgment issued flag to its default value (0).

- If not in a reset state, check if a transaction is complete.

- If a transaction is complete, set the acknowledgment issued flag to 1.

- If a transaction is not complete, clear the acknowledgment issued flag.

7. Connect the inputs and outputs of the internal modules within the top-level ATP\_System module.