

# **INTENSIVE TRAINING PROGRAM 2025 REPORT**

## **GROUP 4 – BAYMAX**

### **VENDING MACHINE'S SPECIFICATION REPORT**

*10/03/2025 – 13/03/2025*

## General purpose

### 1. Overview

| Design Description   |                       |     |     |     |                 |
|--|-----------------------|-----|-----|-----|-----------------|
| <p>Design a vending machine capable of accepting multiple deposits of three denominations: 5, 10, and 20. When a user selects a product, the machine will check the total amount deposited:</p> <ul style="list-style-type: none"> <li>• If the deposited amount is greater than or equal to the product price, the machine will dispense the product and return any change (if applicable).</li> <li>• If insufficient, the machine will prompt the user to add more money or cancel the transaction.</li> <li>• The user can cancel the transaction any time after depositing money, and the machine will refund the full amount.</li> </ul> |                       |     |     |     |                 |
| No.  | Parameter description | Min | Typ | Max | Units           |
| 1.   | Process               |     |     |     |                 |
| 2.   | Voltage               |     |     |     | V               |
| 3.   | Temperature           |     |     |     | °C              |
| 4.   | Power Dissipation     |     |     |     | mW              |
| 5.   | Die Area              |     |     |     | um <sup>2</sup> |
| 6.   | Clock frequency       |     |     |     | GHz             |

Table 1: Specification of Vending Machine

### 2. Functional behavior

The vending machine is designed to dispense items based on user selection and payment. It operates using a finite state machine (FSM) and interfaces with input and output signals for control and monitoring.

#### Item Selection and Payment:

- The user presses start to begin.
- The user selects an item using item\_in.
- The user inserts money (money signal).
- The system waits until done\_money = 1 before moving forward.

#### Comparison:

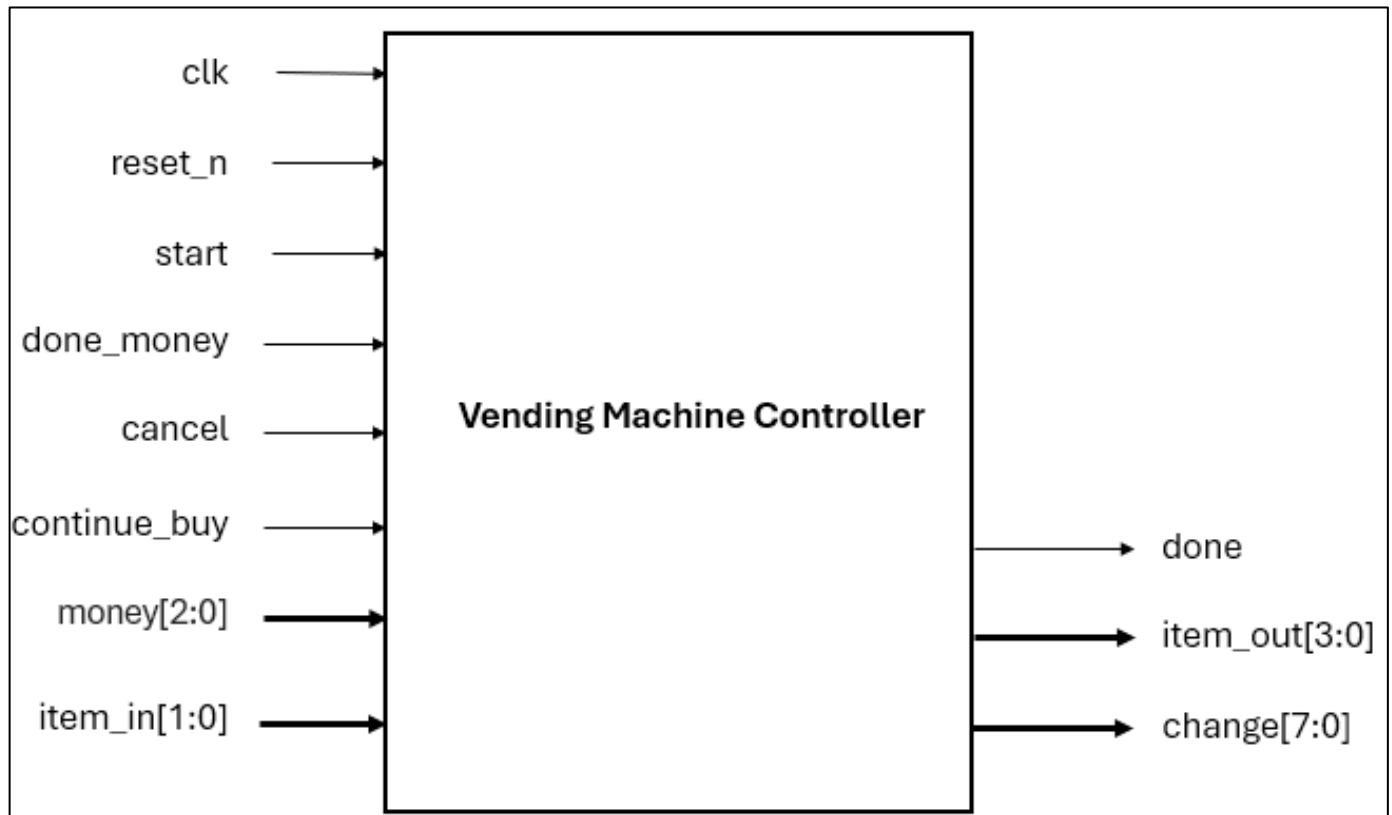
- If the inserted money is sufficient, the system proceeds to dispense the item.
- If the money is insufficient, the transaction is canceled.

#### Dispensing Items and Returning Change:

- Once sufficient money is received, item\_out is activated to dispense the selected item.
- If excess money is inserted, change is calculated and returned.

**Cancellation Handling:**

- If cancel = 1, the transaction is aborted.
- If money was inserted, the system returns the money before resetting.

**I. Top module**

| signal name | width | default value | direction | description  |
|-------------|-------|---------------|-----------|--|
| clk         | 1     | 0             | input     | clock  |
| reset_n     | 1     | 1             | input     | Reset active low (asynchronous)                                    |
| start       | 1     | 0             | input     | Start signal to begin item selection.                              |
| done_money  | 1     | 0             | input     | Confirmation signal indicating sufficient money has been inserted. |
| cancel      | 1     | 0             | input     | Signal to cancel the transaction and refund money.                 |

|              |   |               |        |   |
|--------------|---|---------------|--------|---|
| continue_buy | 1 | 0             | input  | Continue to buy the next item when returned change    |
| money        | 3 | 000           | input  | Denomination of money inserted by the user.           |
| item_in      | 2 | 00            | input  | Code of the item chosen by the user.                  |
| done         | 1 | 0             | output | Indicating the transaction was completed successfully |
| item_out     | 4 | 0000          | output | Code representing the dispensed item.                 |
| change       | 8 | 0000_<br>0000 | output | Amount of change to be returned to the user.          |

Table 2: The signal pins of block Top design

| Signal     | Value       | Action/ Meaning   |
|------------|-------------|---|
| clk        | 1'b0 / 1'b1 | Signal clock  |
| reset_n    | 1'b0        | Reset the circuit, reload default values for variables                            |
|            | 1'b1        | The circuit operates normally   |
| start      | 1'b0        | In wait mode  |
|            | 1'b1        | Start transaction   |
| done_money | 1'b0        | After inserting money, continue inserting more                                    |
|            | 1'b1        | After inserting money, stop adding more and proceed to the next step              |
| cancel     | 1'b0        | Continue the transaction process.   |
|            | 1'b1        | Stop the transaction, return the inserted money (if any), and revert to wait mode |

|              |         |  |
|--------------|---------|--|
| continue_buy | 1'b0    | After returning the change, stop and do not proceed with the transaction (must remain in standby mode until signal start = 1 to begin a new transaction) |
|              | 1'b1    | After returning the change, continue the transaction at the item selection step  |
| money        | 3'b001  | Corresponding to the inserted denomination of 5  |
|              | 3'b010  | Corresponding to the inserted denomination of 10   |
|              | 3'b100  | Corresponding to the inserted denomination of 20   |
|              | else    | The denomination is not accepted, and no additional money is inserted into the machine   |
| item_in      | 2'b00   | Corresponding to selecting item 1  |
|              | 2'b01   | Corresponding to selecting item 2  |
|              | 2'b10   | Corresponding to selecting item 3  |
|              | 2'b11   | Corresponding to selecting item 4  |
| done         | 1'b0    | No item was purchased  |
|              | 1'b1    | One item was purchased   |
| item_out     | 4'b0000 | No item was purchased  |
|              | 4'b1000 | The first item was purchased   |
|              | 4'b0100 | The second item was purchased  |
|              | 4'b0010 | The third item was purchased   |
|              | 4'b0001 | The fourth item was purchased  |

|        |                                 |                               |
|--------|---------------------------------|-------------------------------|
|        | else                            | This output should not appear |
| change | [8'b0000_0000<br>:8'b1111_1111] | Total change amount           |

Table 3: The meaning of the signal for each corresponding value of Top design

II. Submodule

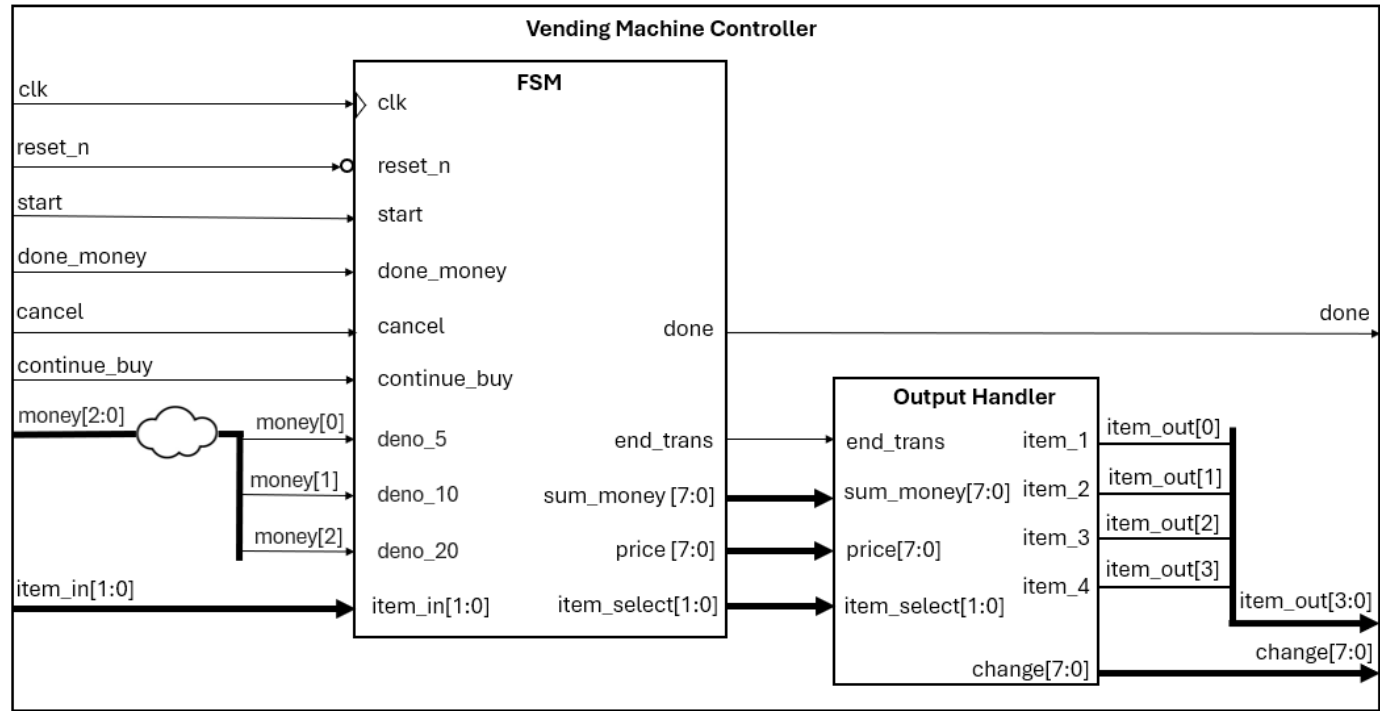
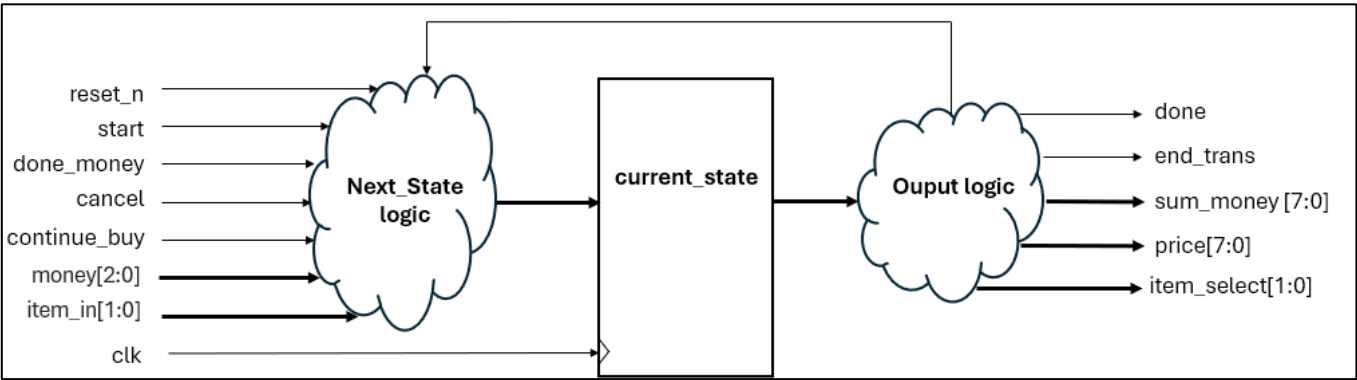
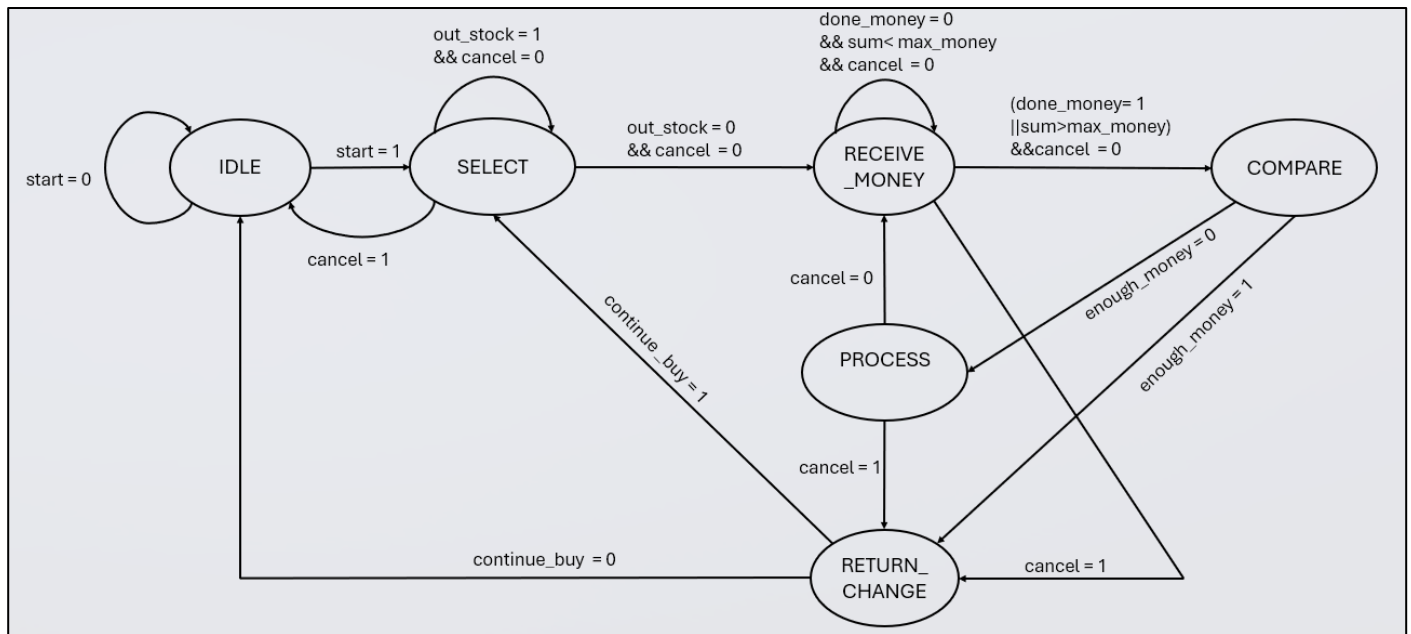


Figure 1: Detail of the top design

1. FSM





| current_state | Signal / Condition |                    |                        | next_state    |
|---------------|--------------------|--------------------|------------------------|---------------|
| IDLE          | start = 1'b0       |                    |                        | IDLE          |
|               | start = 1'b0       |                    |                        | SELECT        |
| SELECT        | cancel = 1'b1      | out_stock = 1'bx   |                        | IDLE          |
|               | cancel = 1'b0      | out_stock = 1'b0   |                        | RECEIVE_MONEY |
|               | cancel = 1'b0      | out_stock = 1'b1   |                        | SELECT        |
| RECEIVE_MONEY | cancel = 1'b1      | done_money = 1'b x | sum > max_money = 1'bx | RETURN_CHANGE |
|               | cancel = 1'b0      | done_money = 1'b 0 | sum > max_money = 1'b0 | RECEIVE_MONEY |
|               | cancel = 1'b0      | done_money = 1'b 0 | sum > max_money = 1'b1 | COMPARE       |
|               | cancel = 1'b0      | done_money = 1'b 1 | sum > max_money = 1'b0 | COMPARE       |
|               | cancel = 1'b0      | done_money = 1'b 1 | sum > max_money = 1'b1 | COMPARE       |
| COMPARE       | enough_money = 0   |                    |                        | PROCESS       |
|               | enough_money = 1   |                    |                        | RETURN_CHANGE |

|               |                  |               |
|---------------|------------------|---------------|
| PROCESS       | cancel = 0       | RECEIVE_MONEY |
|               | cancel = 1       | RETURN_CHANGE |
| RETURN_CHANGE | continue_buy = 0 | IDLE          |
|               | continue_buy = 1 | SELECT        |

Table 4: Truth table of FSM

| No. | state         | Output      |      |
|-----|---------------|-------------|------|
| 1   | IDLE          | done        | 1'b0 |
|     |               | end_trans   | 1'b0 |
|     |               | sum_money   | 8'b0 |
|     |               | price       | 8'b0 |
|     |               | item_select | 2'b0 |
| 2   | SELECT        | done        | 1'b0 |
|     |               | end_trans   | 1'b0 |
|     |               | sum_money   | 8'b0 |
|     |               | price       | 8'b0 |
|     |               | item_select | 2'b0 |
| 3   | RECEIVE_MONEY | done        | 1'b0 |
|     |               | end_trans   | 1'b0 |
|     |               | sum_money   | 8'b0 |
|     |               | price       | 8'b0 |
|     |               | item_select | 2'b0 |
| 4   |               | done        | 1'b0 |



|   |               |             |  |
|---|---------------|-------------|--|
|   | COMPARE       | end_trans   | 1'b0   |
|   |               | sum_money   | 8'b0   |
|   |               | price       | 8'b0   |
|   |               | item_select | 2'b0   |
| 5 | PROCESS       | done        | 1'b0   |
|   |               | end_trans   | 1'b0   |
|   |               | sum_money   | 8'b0   |
|   |               | price       | 8'b0   |
|   |               | item_select | 2'b0   |
| 6 | RETURN_CHANGE | done        | = 1'b1 if buy successful 1 item, else = 1'b0 |
|   |               | end_trans   | 1'b1   |
|   |               | sum_money   | Value of sum_money                           |
|   |               | price       | Price of the selected item                   |
|   |               | item_select | Selected item                                |

Table 5: The output in each state

| Signal       | Width | Default value | Direction | Description  |
|--------------|-------|---------------|-----------|--|
| clk          | 1     | 0             | input     | Clock  |
| reset_n      | 1     | 1             | input     | Reset active low (asynchronous)                                    |
| start        | 1     | 0             | input     | Start signal to begin item selection.                              |
| done_money   | 1     | 0             | input     | Confirmation signal indicating sufficient money has been inserted. |
| cancel       | 1     | 0             | input     | Signal to cancel the transaction and refund money.                 |
| continue_buy | 1     | 0             | input     | Continue to buy the next item                                      |

|             |   |   |        |   |
|-------------|---|---|--------|---|
| deno_5      | 1 | 0 | input  | Is the denomination 5?                                |
| deno_10     | 1 | 0 | input  | Is the denomination 10?                               |
| deno_20     | 1 | 0 | input  | Is the denomination 20?                               |
| item_in     | 2 | 0 | input  | Code of the item chosen by the user.                  |
| done        | 1 | 0 | output | Indicating the transaction was completed successfully |
| end_trans   | 1 | 0 | output | End a transaction                                     |
| sum_money   | 8 | 0 | output | Total of money  |
| price       | 8 | 0 | output | Price of the item that has been selected              |
| item_select | 2 | 0 | output | The item has been selected                            |

Table 6: The signal pins of block Top design

| Signal       | Value       | Action/Meaning  |
|--------------|-------------|---|
| clk          | 1'b0 / 1'b1 | Signal clock  |
| reset_n      | 1'b0        | The circuit resets and reloads the default values for the variables.  |
|              | 1'b1        | The circuit operates normally.  |
| start        | 1'b0        | Stay at state IDLE  |
|              | 1'b1        | Transition to state SELECT  |
| done_money   | 1'b0        | If in the RECEIVE-MONEY state, remain in the same state at the next clock cycle                                 |
|              | 1'b1        | If in the RECEIVE-MONEY state, stop inserting money and transition to the next state.                           |
| cancel       | 1'b0        | Maintain the transaction process  |
|              | 1'b1        | Stop the transaction and transition to the RETURN-CHANGE state at the next clock cycle                          |
| continue_buy | 1'b0        | In the RETURN-CHANGE state, the transaction will stop and transition to the IDLE state at the next clock cycle. |

|             |                              |   |
|-------------|------------------------------|---|
|             | 1'b1                         | In the RETURN-CHANGE state, the transaction will continue and transition to the SELECT state at the next clock cycle. |
| deno_5      | 1'b0                         | Do not increase the total amount.   |
|             | 1'b1                         | If state = RECEIVE-MONEY, increase the total amount by 5  |
| deno_10     | 1'b0                         | Do not increase the total amount.   |
|             | 1'b1                         | If state = RECEIVE-MONEY, increase the total amount by 10   |
| deno_20     | 1'b0                         | Do not increase the total amount.   |
|             | 1'b1                         | If state = RECEIVE-MONEY, increase the total amount by 20   |
| item_in     | 2'b00                        | Corresponding to selecting item 1 (retrieve the price and quantity of the product)                                    |
|             | 2'b01                        | Corresponding to selecting item 2 (retrieve the price and quantity of the product)                                    |
|             | 2'b10                        | Corresponding to selecting item 3 (retrieve the price and quantity of the product)                                    |
|             | 2'b11                        | Corresponding to selecting item 4 (retrieve the price and quantity of the product)                                    |
| done        | 1'b0                         | No item was purchased or the transaction is not yet complete  |
|             | 1'b1                         | One item has been purchased   |
| end-trans   | 1'b0                         | The transaction is not yet complete.  |
|             | 1'b1                         | The transaction has ended in the RETURN-CHANGE state  |
| sum-money   | [8'b0000_0000 :8'b1111_1111] | Total amount inserted.  |
| price       | [8'b0000_0000 :8'b1111_1111] | Price of the product.   |
| item-select | 2'b00                        | The first item has been selected.   |
|             | 2'b01                        | The second item has been selected.  |
|             | 2'b10                        | The third item has been selected.   |

|  |       |                                   |
|--|-------|-----------------------------------|
|  | 2'b11 | The fourth item has been selected |
|--|-------|-----------------------------------|

*Table 7: The meaning of the signal for each corresponding value of FSM design*

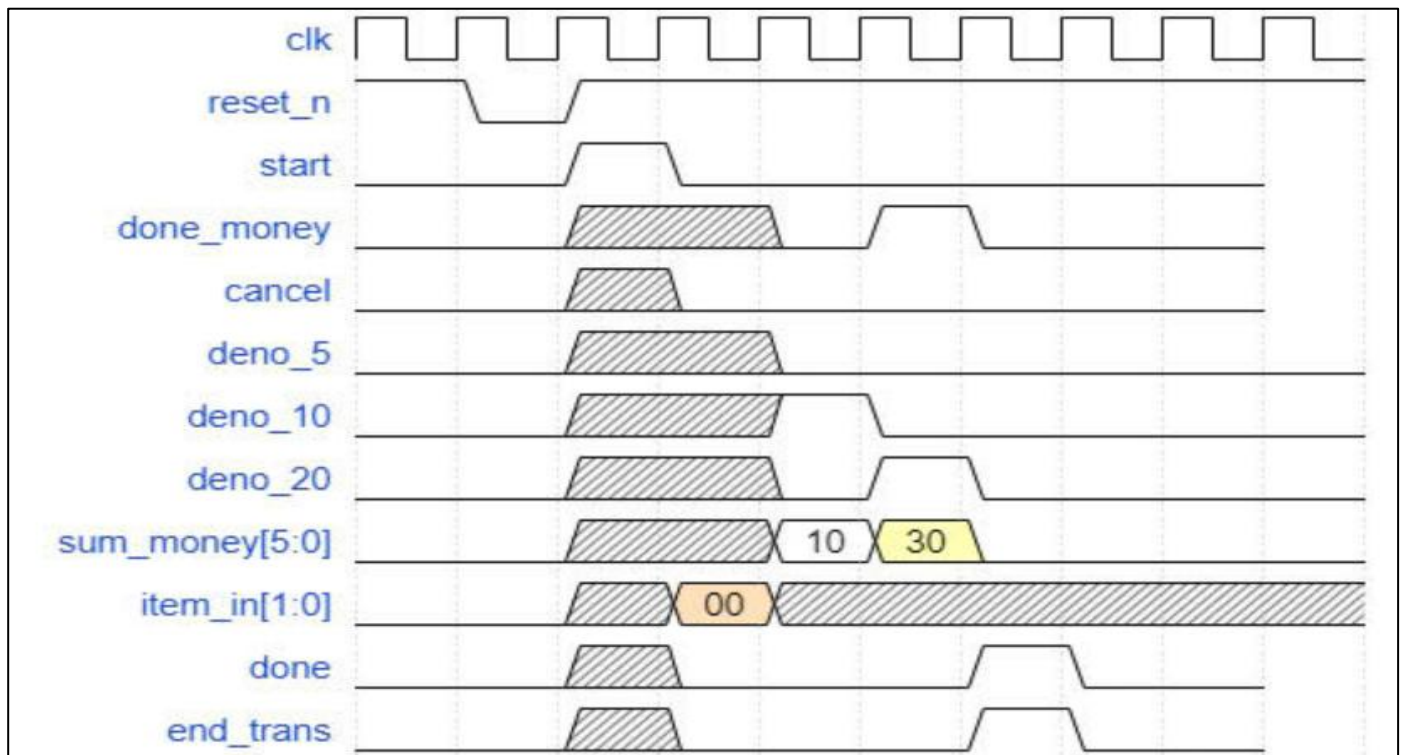
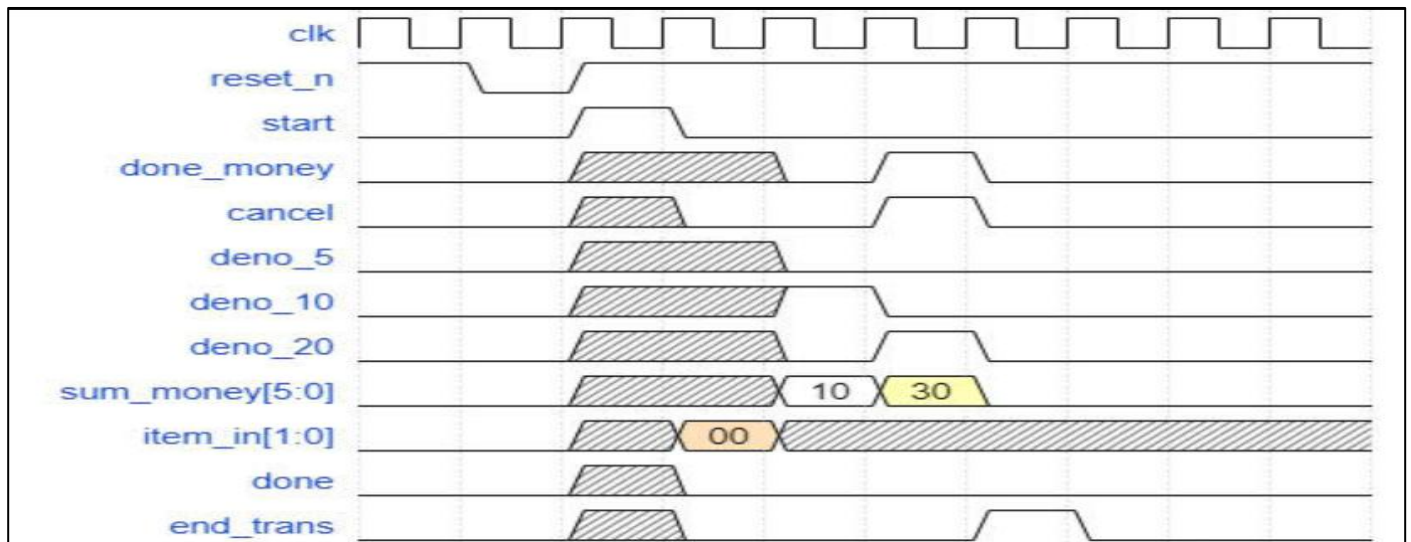
This is the main state machine of the design. It consists of six states:

- IDLE – Initialization stage.
- SELECT – Selecting the type of goods.
- RECEIVE\_MONEY – Receiving money.
- COMPARE – Checking whether the inserted amount is sufficient for the purchase.
- PROCESS – Adding more money or finalizing the transaction.
- RETURN\_CHANGE – Returning the remaining balance.

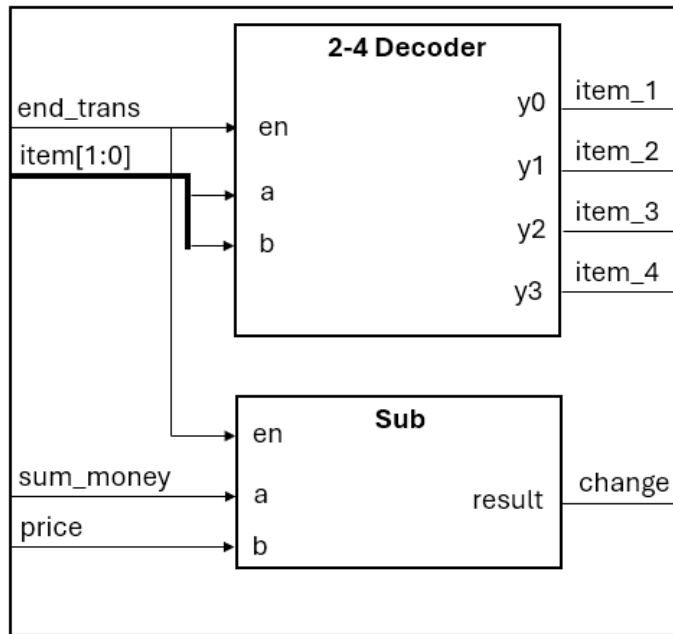
Initially, the machine remains in the IDLE state until the start signal is activated, at which point it transitions to the SELECT state. In the SELECT state, the machine receives an item input to choose the desired product. If the quantity of the selected item is greater than zero, the machine transitions to the RECEIVE\_MONEY state. Money is inserted based on the money input, which corresponds to specific values:

- money = 100 → Adds 5
- money = 010 → Adds 10
- money = 001 → Adds 20

This process continues until either done\_money is activated or the total inserted amount exceeds max\_money, at which point the machine transitions to the COMPARE state. In the COMPARE state, the inserted money is compared with the price of the selected product: If the amount is insufficient, the machine transitions to the PROCESS state. Depending on the cancel input, two scenarios arise: if cancel = 0, the machine returns to RECEIVE\_MONEY to accept additional money. if cancel = 1, the machine moves to RETURN\_CHANGE to fully refund the inserted amount. If the inserted money is sufficient for the purchase, the machine transitions to RETURN\_CHANGE to return any excess money. Additionally, the cancel = 1 signal is also valid in the RECEIVE\_MONEY state to stop and refund the money, or in the SELECT state to cancel the transaction and return to IDLE. When the refunded amount is not equal to zero, the end\_trans output signal is activated.



## 2. Output Handler



| signal      | width | default value | direction | description  |
|-------------|-------|---------------|-----------|--|
| end_trans   | 1     | 0             | input     | end a transaction  |
| sum_money   | 8     | 0             | input     | Represents the total amount of money received by the vending machine from the user.                          |
| price       | 8     | 0             | input     | price of the item that has been selected   |
| item_select | 2     | 0             | input     | the selected item  |
| item_1      | 1     | 0             | output    | Indicates that item 1 has been dispensed.  |
| item_2      | 1     | 0             | output    | Indicates that item 2 has been dispensed.  |
| item_3      | 1     | 0             | output    | Indicates that item 3 has been dispensed.  |
| item_4      | 1     | 0             | output    | Indicates that item 4 has been dispensed.  |
| change      | 8     | 0             | output    | Represents the amount of change to be returned to the customer if the inserted money exceeds the item price. |

Table 8: The pin signals of block *Output\_Handler*

| Signal      | Value                           | Action/ Meaning                      |
|-------------|---------------------------------|--------------------------------------|
| end_trans   | 1'b0                            | All output = 0                       |
|             | 1'b1                            | The output ports can produce values. |
| sum_money   | [8'b0000_0000<br>:8'b1111_1111] | Total amount inserted                |
| price       | [8'b0000_0000<br>:8'b1111_1111] | Price of the item                    |
| item_select | 1'b00                           | Select item 1                        |
|             | 1'b01                           | Select item 2                        |
|             | 1'b10                           | Select item 3                        |
|             | 1'b11                           | Select item 4                        |
| item_1      | 1'b0                            | Item 1 was not selected              |
|             | 1'b1                            | Item 1 has been selected             |
| item_2      | 1'b0                            | Item 2 was not selected              |
|             | 1'b1                            | Item 2 has been selected             |
| item_3      | 1'b0                            | Item 3 was not selected              |
|             | 1'b1                            | Item 3 has been selected             |
| item_4      | 1'b0                            | Item 4 was not selected              |
|             | 1'b1                            | Item 4 has been selected             |
| change      | [8'b0000_0000<br>:8'b1111_1111] | Total of change                      |

Table 9: The meaning of the signal for each corresponding value

This block takes the item input to determine which of the four-item types (item\_0, item\_1, item\_2, item\_3) has been purchased. If an item is purchased, its corresponding value is set to 1, otherwise, it is 0. Additionally, the block outputs the change amount, which represents the remaining balance

after deducting the purchase price. If the purchase is unsuccessful, the change output may be the entire inserted amount. When `end_trans = 0`, the output value will always be 0.

