



10

s



SIMULATION - Behavioral Simulation - Functional - sim_1 - vending_machine_tb

vending_machine.v

Untitled 1

vending_machine_tb.v



Name

Value

clk

0

reset

0

> coin[1:0]

0

select_item

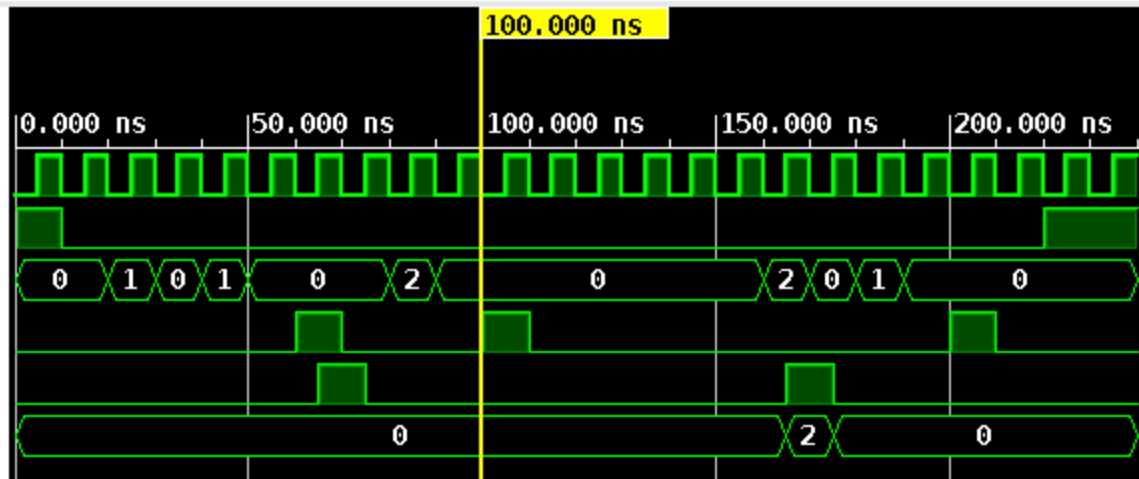
1

dispense

0

> change[3:0]

0



Tcl Console

Messages

Log



Time	reset	coin	select_item	dispense	change
Time=50000	0	00	0	0	0
Time=60000	0	00	1	0	0
Time=65000	0	00	1	1	0
Time=70000	0	00	0	1	0
Time=75000	0	00	0	0	0
Time=80000	0	10	0	0	0
Time=90000	0	00	0	0	0
Time=100000	0	00	1	0	0
Time=110000	0	00	0	0	0
Time=160000	0	10	0	0	0
Time=165000	0	10	0	1	2
Time=170000	0	00	0	1	2
Time=175000	0	00	0	0	0
Time=180000	0	01	0	0	0
Time=190000	0	00	0	0	0
Time=200000	0	00	1	0	0
Time=210000	0	00	0	0	0
Time=220000	1	00	0	0	0

Type a Tcl command here

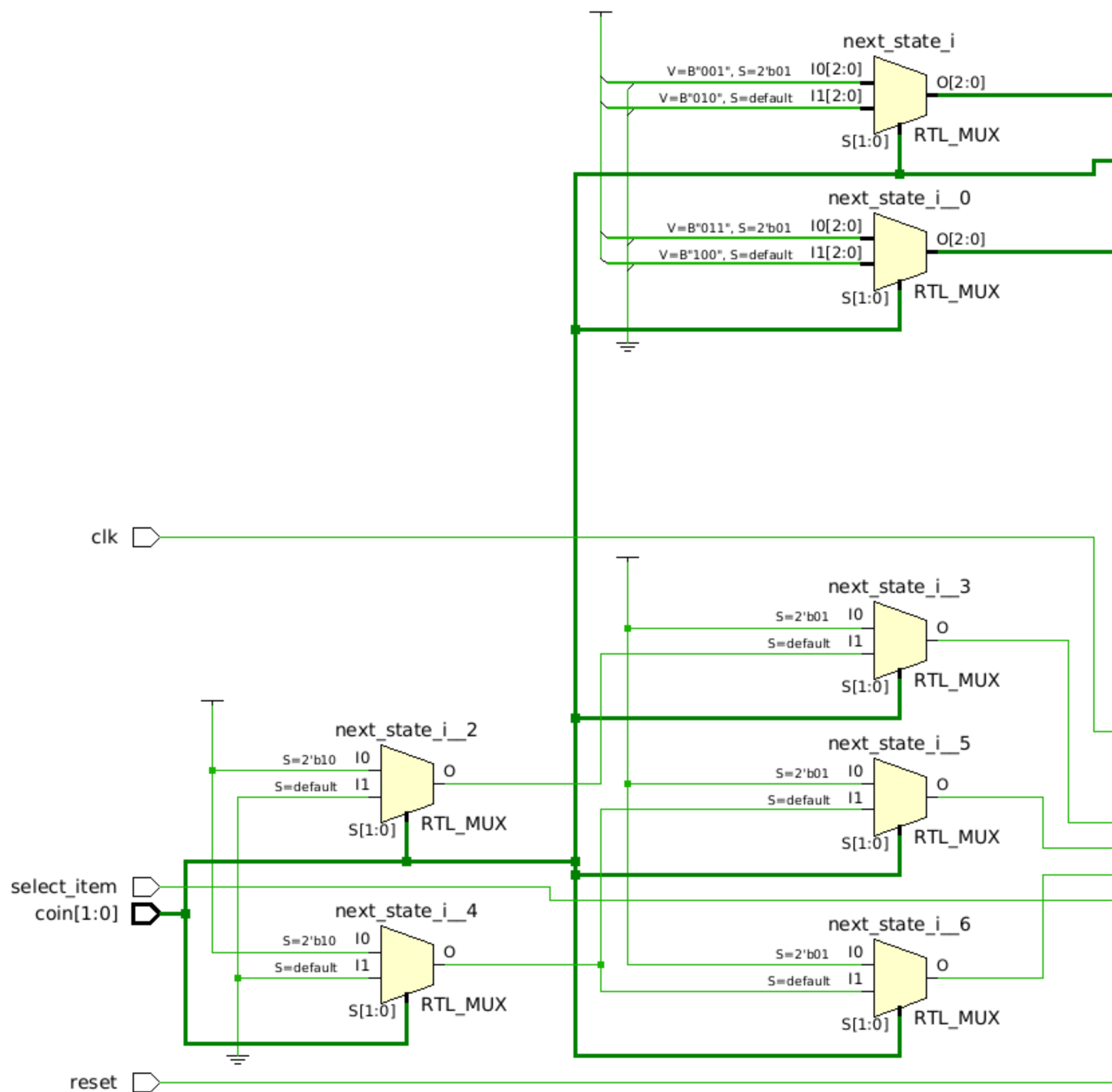
Schematic

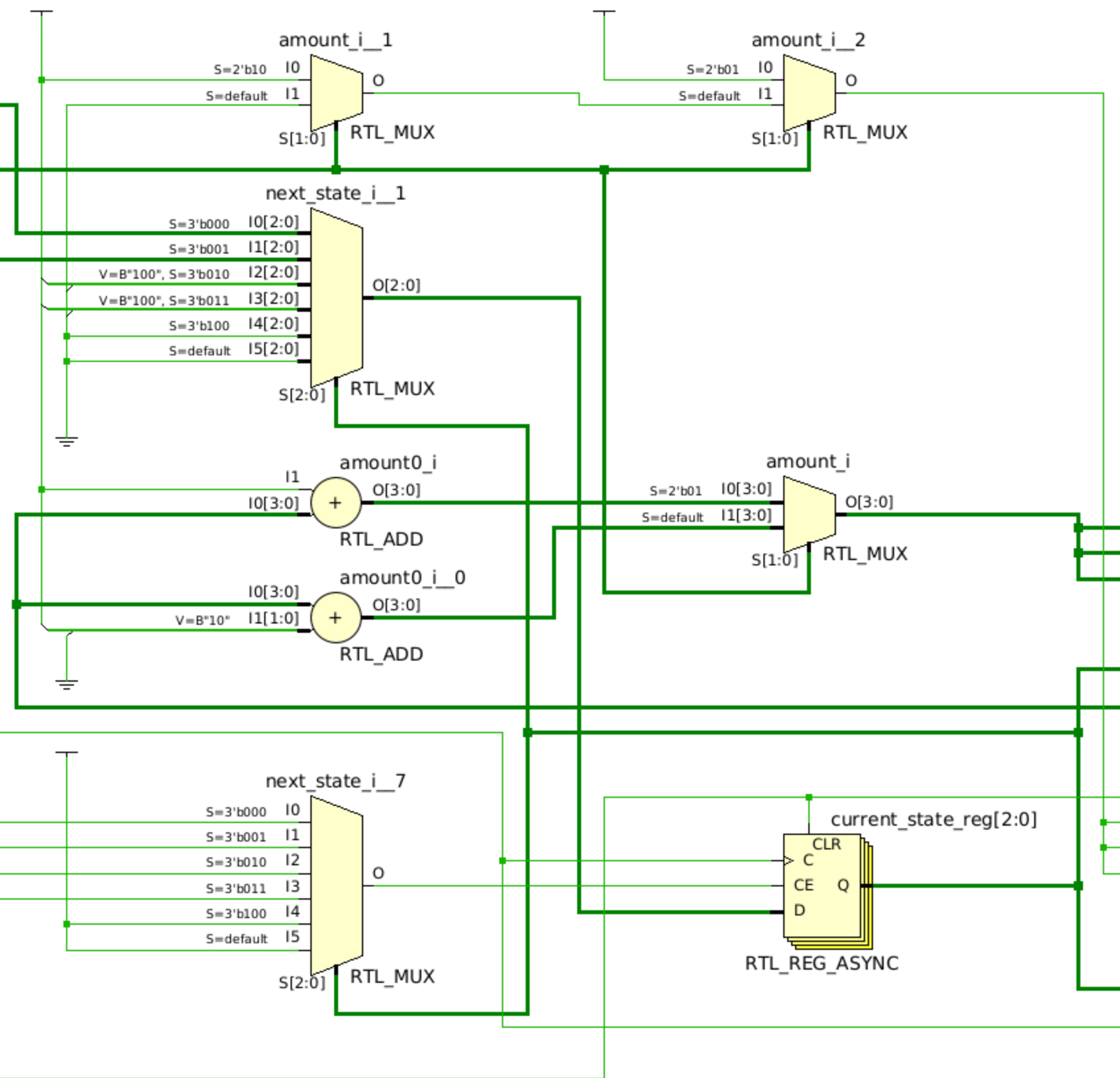


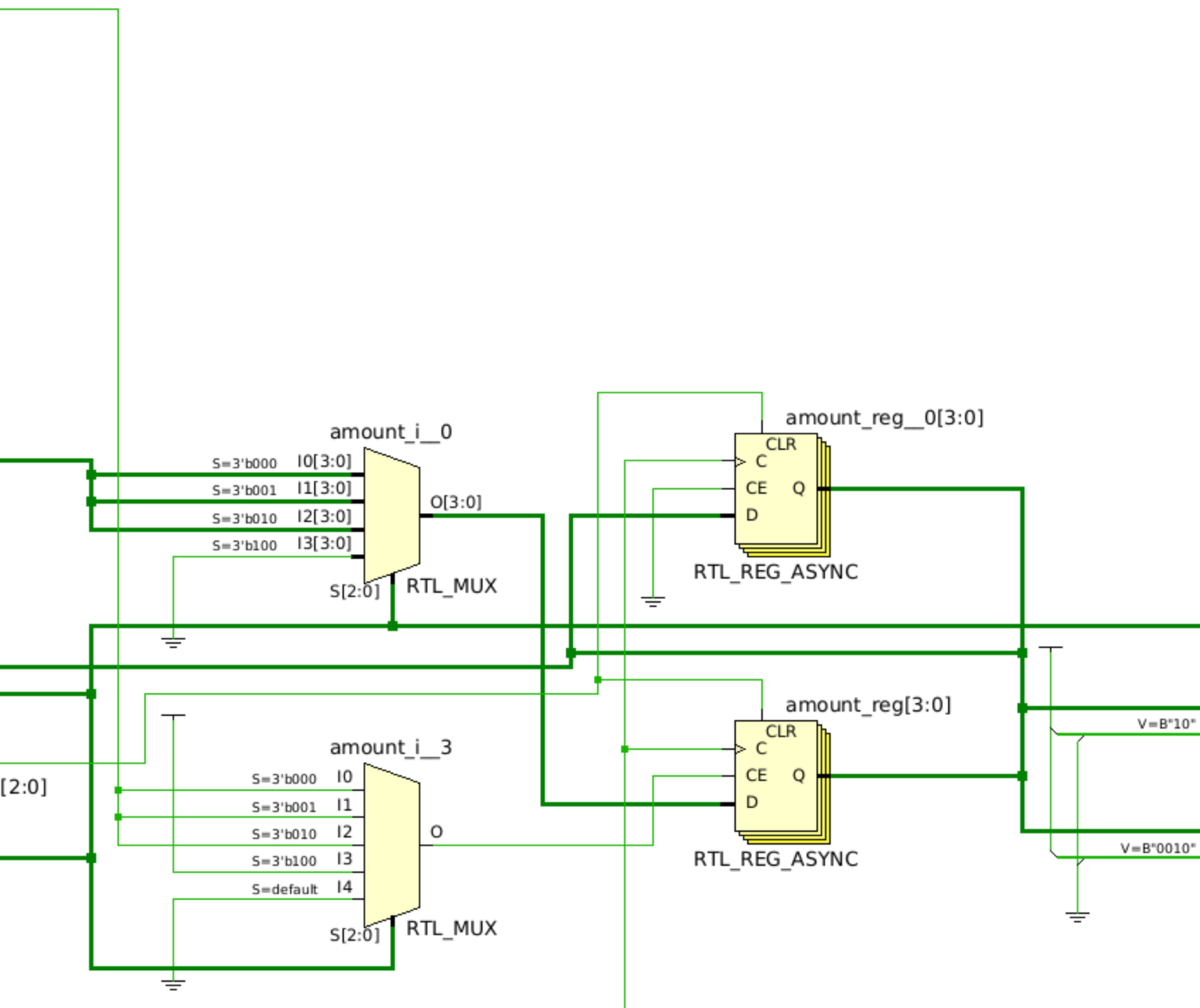
32 Cells

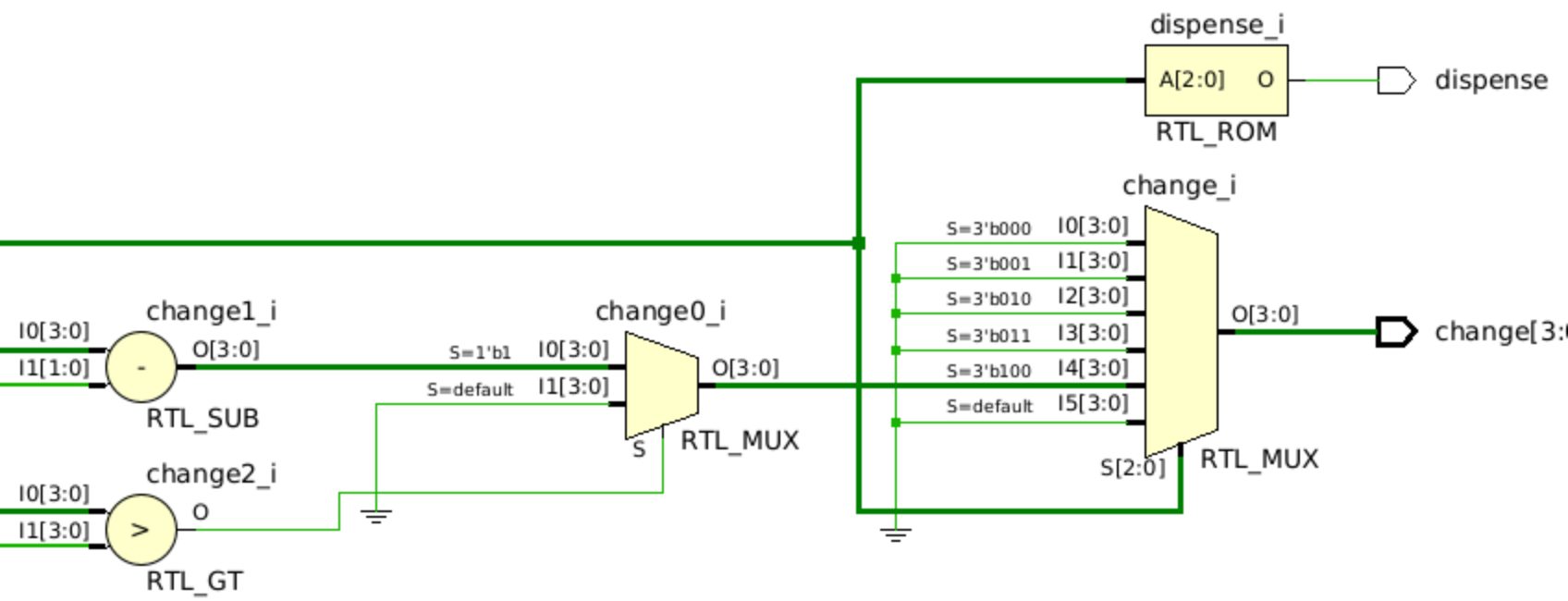
10 I/O Ports

62 Nets











```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
3  // Engineer: Anjan Prasad
4  // Create Date: 12/24/2024 12:10:52 AM
5  // Module Name: vending_machine
6  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
7
8  module vending_machine (
9      input clk,reset,select_item,
10     input [1:0] coin,      // Input coin: 2'b01 for 1 unit, 2'b10 for 2 units
11     output reg dispense,
12     output reg [3:0] change
13 );
14
15     parameter IDLE      = 3'b000,
16                ONE_UNIT = 3'b001,
17                TWO_UNITS = 3'b010,
18                ITEM_READY = 3'b011,
19                DISPENSE  = 3'b100;
20
21     reg [2:0] current_state, next_state;
22     reg [3:0] amount; // Tracks the inserted amount
23
24     always @(posedge clk or posedge reset) begin
25         if (reset) begin
26             current_state <= IDLE;
27             amount <= 0;
28         end else begin
29             current_state <= next_state;
30         end
31     end
32
33     always @(*) begin
34         next_state = current_state;
35         dispense = 0;
36         change = 0;
37
38         case (current_state)
39             IDLE: begin
40                 if (coin == 2'b01) begin
41                     next_state = ONE_UNIT;
42                 end else if (coin == 2'b10) begin
43                     next_state = TWO_UNITS;
44                 end
45             end
46             ONE_UNIT: begin
47                 if (coin == 2'b01) begin
48                     next_state = ITEM_READY;
49                 end else if (coin == 2'b10) begin
50                     next_state = DISPENSE;
```



```
51         end
52     end
53     TWO_UNITS: begin
54         if (coin == 2'b01) begin
55             next_state = DISPENSE;
56         end else if (coin == 2'b10) begin
57             next_state = DISPENSE;
58         end
59     end
60     ITEM_READY: begin
61         if (select_item) begin
62             next_state = DISPENSE;
63         end
64     end
65     DISPENSE: begin
66         dispense = 1;
67         change = (amount > 2) ? amount - 2 : 0; // Calculate change
68         next_state = IDLE;
69     end
70     default: next_state = IDLE;
71 endcase
72 end
73
74 always @(posedge clk or posedge reset) begin // Update Amount Logic
75     if (reset) begin
76         amount <= 0;
77     end else begin
78         case (current_state)
79             IDLE: begin
80                 if (coin == 2'b01) amount <= amount + 1;
81                 else if (coin == 2'b10) amount <= amount + 2;
82             end
83
84             ONE_UNIT: begin
85                 if (coin == 2'b01) amount <= amount + 1;
86                 else if (coin == 2'b10) amount <= amount + 2;
87             end
88
89             TWO_UNITS: begin
90                 if (coin == 2'b01) amount <= amount + 1;
91                 else if (coin == 2'b10) amount <= amount + 2;
92             end
93
94             DISPENSE: amount <= 0;
95
96             default: amount <= amount;
97         endcase
98     end
99 end
100 endmodule
```



```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////
3  // Engineer: Anjan Prasad
4  // Create Date: 12/24/2024 12:12:27 AM
5  // Module Name: vending_machine_tb
6  //////////////////////////////////////
7
8  module vending_machine_tb;
9
10     reg clk;
11     reg reset;
12     reg [1:0] coin;
13     reg select_item;
14     wire dispense;
15     wire [3:0] change;
16
17     vending_machine DUT (
18         .clk(clk),
19         .reset(reset),
20         .coin(coin),
21         .select_item(select_item),
22         .dispense(dispense),
23         .change(change)
24     );
25
26     always #5 clk = ~clk;
27
28     initial begin
29
30         clk = 0;
31         reset = 1;
32         coin = 0;
33         select_item = 0;
34
35         #10 reset = 0;
36
37         // Scenario 1: Insert two 1-unit coins and select an item
38         #10 coin = 2'b01;
39         #10 coin = 0;
40         #10 coin = 2'b01;
41         #10 coin = 0;
42         #10 select_item = 1;
43         #10 select_item = 0;
44
45         // Scenario 2: Insert a 2-unit coin directly
46         #10 coin = 2'b10;
47         #10 coin = 0;
48         #10 select_item = 1;
49         #10 select_item = 0;
50         #40;
```




```
50      ○      #40;
51
52      // Scenario 3: Insert extra coins and get change
53      ○      #10 coin = 2'b10;
54      ○      #10 coin = 0;
55      ○      #10 coin = 2'b01;
56      ○      #10 coin = 0;
57      ○      #10 select_item = 1;
58      ○      #10 select_item = 0;
59
60      ○      #10 reset = 1;
61      ○ →     #20 $finish;
62      ○      end
63
64      ○      initial begin
65      ○ $monitor("Time=%0t | reset=%b | coin=%b | select_item=%b | dispense=%b | change=%d",
66                $time, reset, coin, select_item, dispense, change);
67      ○      end
68
69      ○      endmodule
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```