

Title : Logical circuit design _ vending machine

Author : 김정환(Kim jeonghwan)

Datetime : 2022.01.12.

1. 목표

전제) 유한상태머신 (FSM)으로 구현

1_1) Total 금액 300원을 초과해서 사용하지 못하도록 제한.

ex) 150원의 커피를 뽑은 이후, 150원 초과해서 돈을 투입 못하도록 구현

1-2) s0,s50...s300 7개의 상태를 뒤서 300원까지 화폐를 투입할 수 있는 기능을 구현, 화폐를 투입하거나 소비하면 State를 알맞은 위치로 이동.

2_1) 입력 신호가 들어오면 50원 동전과 100원 동전 투입 명령을 주는 버튼 구현.

2_2) 반환 버튼을 눌렀을때, 지금까지 넣은 화폐를 반환해주는 기능을 구현. 화폐는 100원과 50원만을 사용하므로 출력 역시 두 종류만 나옴, 다만 중복되는 화폐가 여러개일 수 있으니 Integer 형태로 구현. 금액이 큰 화폐를 최대한 반환하는 쪽으로 프로그래밍.

3) 150원의 커피와, 100원의 코코아를 뽑을 수 있는 기능 구현. 커피와 코코아를 중복해서 뽑을 수 있으므로 Integer 형태로 출력.

4) 동전을 2개 넣거나, 금액을 초과해서 넣거나, 상품 구매에 필요한 금액이 부족할 경우 모두 else 구문으로 처리해서 명령 실행이 되지않도록 설계.

5) rst(reset) 신호가 0일 경우 프로그램을 초기화한다.

6) 커피와 코코아는 손에 들고 있기 때문에, 출력 신호를 계속해서 유지하지만, 반환되는 동전의 수는 나오는 순간의 출력만 보여줄 것임.

2. 결과

전제) 유한상태머신 (FSM)으로 구현 성공

1_1) **total**은 수중에 들고 있는 돈을 나타낸다.

Module파일에 Variable **total** := 300을 지정하고, 화폐를 자판기에 투입할때마다 **total**에서 차감되도록 설정하였음. **Return_button**에 신호가 오면 동전 반환과 동시에 **state**의 크기만큼 **total**을 증가. (만약 150원을 커피 사는데에 사용했다면 후에 아무리 반환하더라도 **total**은 150원까지 밖에 안 오름)

1_2) **s0,s50...s300** 7개의 상태를 만들어 금액을 투입하거나, 반환 또는 상품을 구매 할 때마다 상태를 이동하도록 설정하였음.

2_1) 입력 포트인 **coin50** 버튼과 **coin100** 버튼을 각각 구현, **coin50**에 신호가 들어오면 상태를 50증가시키고 **coin100**에 신호가 들어오면 상태가 100증가함. 입력 받은 금액은 총합 300을 넘지 못하도록 설계하였음.

2_2) 입력 포트인 **Return_button**을 구현. **Return_button**에 신호가 들어오면 현재 상태 (0,50...300)에 맞는 금액을 반환해줌. 반환해주는 화폐의 종류는 큰 금액을 우선으로 반환함. 예를 들어 250원 반환 -> 100원 2개, 50원 1개.

화폐 개수 출력 신호 포트 명 : **coin50out, coin100out : Integer**

자료형을 Integer로 주었기 때문에 반환받는 화폐의 개수를 성공적으로 나타냄.

3) 입력 포트인 **coffee_button, cocoa_button**을 구현. 각각의 가격은 150원, 100원으로 책정되어서 신호를 받을 경우, 금액이 충분한 상태일때 상태 (0,50...300)에서 각각 150, 100씩 감소시키도록 구현하였음.

출력 포트인 **coffee_out, cocoa_out : Integer** 를 지정해서 최종적으로 몇개의 커피와 코코아를 주문했는지 출력하도록 함.

주문은 순차적으로 이루어지기 때문에 Module 파일의 process 내부에 Variable **coffee_value, cocoa_value : Integer := 0** 을 지정함. 주문이 성공적으로 이루어졌을 경우 각각의 값을 증가시키고, process의 마지막에 Out Port에 각각의 값들을 전달해줌으로서 뽑은 커피와 코코아의 개수를 성공적으로 출력함.

4) 각각의 When절마다 **Else**구문을 뒤서 상태에 맞는 예외 상황은 처리하지 않도록 설계하였음. 예를 들어 동전 두 개를 동시에 입력 받거나, 코코아와 커피를 동시에 주문 받거나, 수중에 있는 돈 **Total**을 초과하는 금액을 투입하거나 금액이 모자란데 주문을 하는 경우 등을 예외 처리하였음.

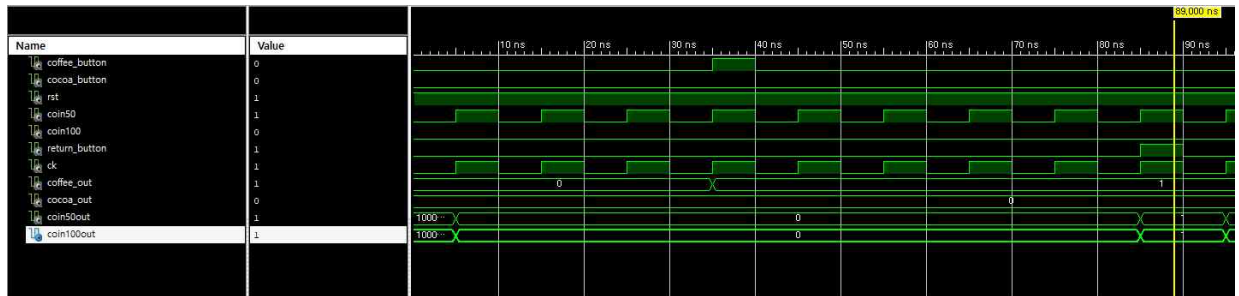
5) 입력 포트 **Rst(reset)** 을 추가함, Rst 신호가 0일 경우 프로그램을 초기화하고 **total, coffee_out, cocoa_out**을 모두 초기값으로 돌려줌

6) 커피와 코코아는 출력 신호가 유지되고, 반환되는 동전은 잠시 보여줬다가 0으로 바뀌게 구현이 되었음.

3. 시뮬레이션

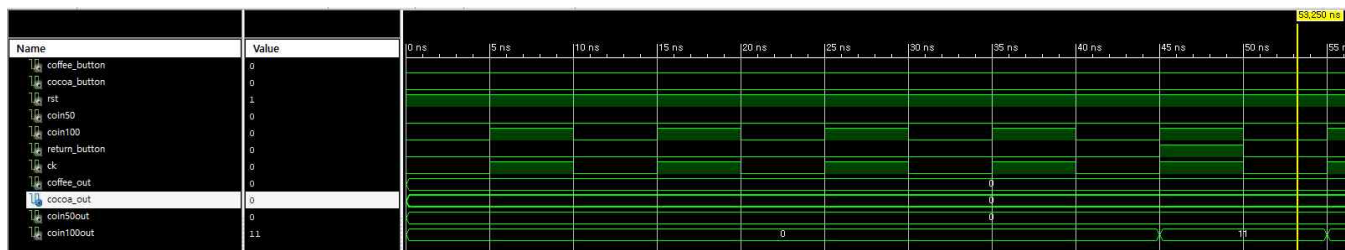
Simulation radix 설정이 안 되서 부득이 Integer 출력이 Binary 형태로 출력됩니다.
 미리 양해 부탁드립니다.
 integer 자료형을 사용하는 포트는 coffee_out, cocoa_out, coin50out, coin100out 입니다.

Case 1) total 금액을 초과해서 사용하지 못하도록 설정.



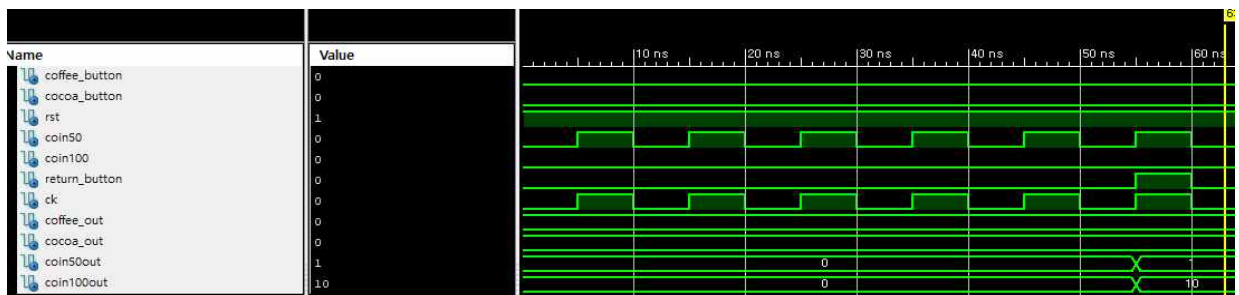
10ns마다 자판기에 50원을 투입, 35ns에 커피 주문을 받자 coffee_out이 1이 되었음을 확인 가능함. 이후 50원을 5번 추가로 투입했지만 85ns에 Return_button에 1 신호가 들어왔을 때 coin100out에 1(2)과 coin50out에 1(2), 총 150원만 반환되는 것을 알 수 있음.

Case 2) 100원을 3번 입력받고 300원을 반환



10ns마다 100원을 투입하고, 45ns에 동전 반환을 요구.
 100원 입력 신호를 총 4번 받았지만, coin100out에는 11(2)만 반환되는 것을 볼 수 있음.

Case 3) 50원을 5번 입력받고 250원을 반환



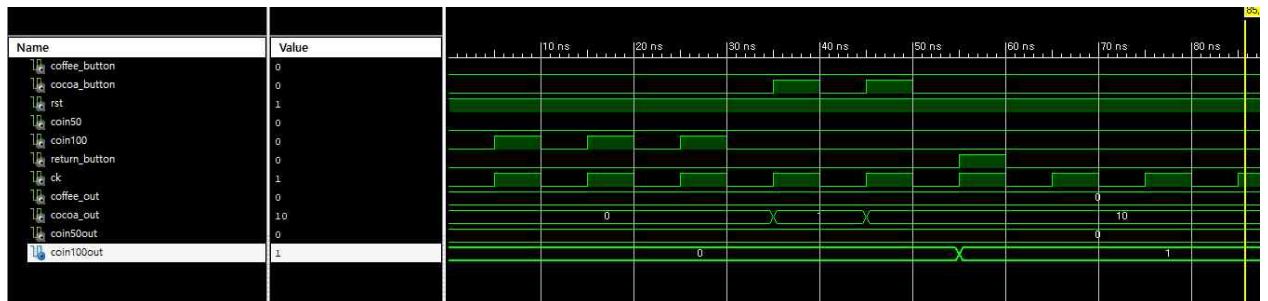
10ns마다 50원을 투입하고 55ns에 반환요구.
 coin50out에 1(2), coin150out에 10(2), 도합 250원이 반환되는 것을 볼 수 있음.

Case 4) 100원을 계속 투입하면서 커피 2개를 뽑았을 경우



10ns마다 100원을 투입하고, 25ns와 45ns에 각각 coffee_button에 신호 1을 주었음.
 coffee_button에 1신호가 올때마다 coffee_out의 값이 각각 1₍₂₎, 10₍₂₎으로 변화함.
 이후 75ns, 155ns에 동전 반환을 요구했지만 수중의 돈을 모두 써버렸기에 반환되는 동전
 이 없음을 볼 수 있음.

Case 5) 300원을 투입하고 코코아 2개를 뽑은 뒤 반환 요구한 경우



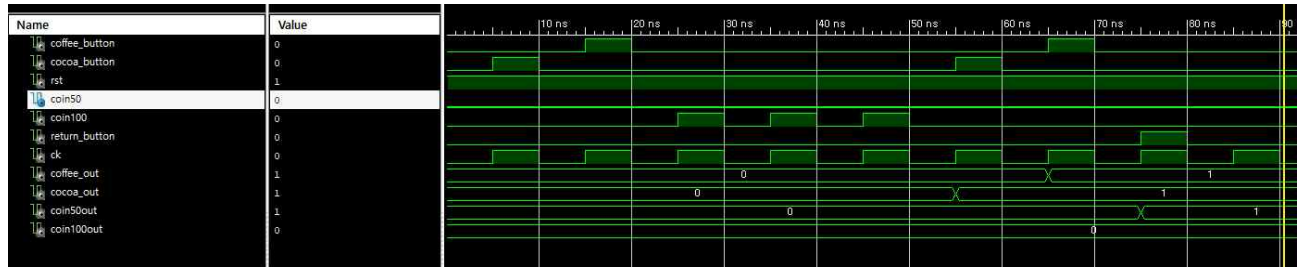
100원을 3번, 300원을 투입 받은 후 35ns 45ns에 코코아 주문을 했을 경우
 cocoa_out의 출력이 1₍₂₎, 10₍₂₎으로 변화하는 것을 확인 할 수 있고
 55ns에 return_button에 1신호를 주자 coin100out에 1이 반환되는 것을 볼 수 있다.

Case 6) 300원을 투입하고 cocoa_button과 coffee_button에 동시에 신호를 줄 경우



10ns 주기로 100원을 투입해서 300원이라는 충분한 예산이 있음에도 불구하고
 2개의 제품을 한번에 출력할 수 없으므로 예외처리가 되는 모습

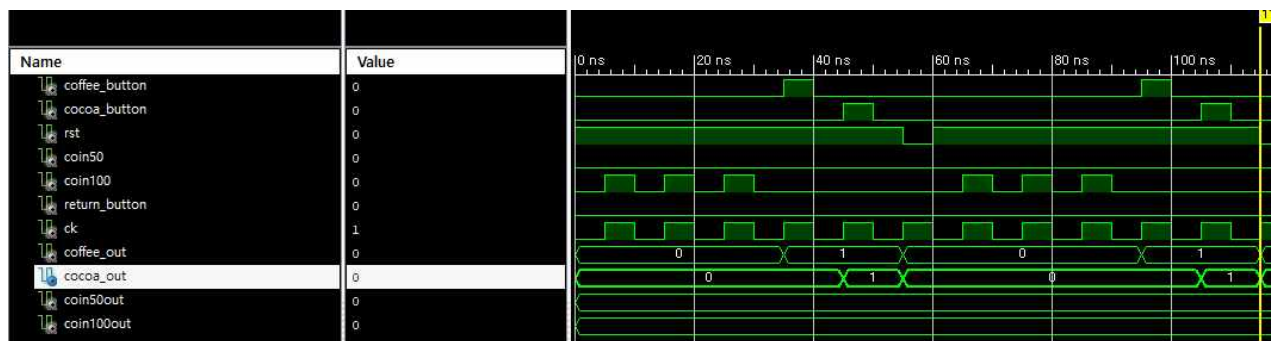
Case 7) 넣은 돈이 0인데 주문시 예외처리, 이후 다시 돈을 넣고 주문



5ns에 cocoa_button에 1신호, 15ns에 coffee_button에 1신호를 주었지만 현재 자판기에 투입된 돈이 0원이기 때문에 예외 처리 되어서 출력이 없음. 이후 25ns부터 10ns 주기로 100원을 3번 투입하고 55ns에 cocoa_button에 1신호, 65ns에 coffee_button에 1신호를 주자 cocoa_out과 coffee_out에 각각 1(2) 신호가 출력됨을 확인할 수 있다.

이후, 75ns에 return_button에 1신호를 주자 coin50out에 1(2)가 출력되어 50원이 반환됨을 보여줌

Case 8) 커피와 코코아를 뽑은 상태에서 rst(reset)에 0 신호 줘보기.



5ns부터 10ns 주기로 100원을 투입 받아서 자판기에 300원을 충전 시킨 뒤

35ns, 45ns에 coffee_button과 cocoa_button에 1 신호를 주어서

coffee_out, cocoa_out의 출력을 각각 1(2), 1(2)로 만듦.

이후, rst에 0 신호를 주니 프로그램이 정상적으로 초기화 됨을 확인할 수 있음.

이때 Total의 값도 다시 300으로 돌아가기 때문에 위 과정을 다시 반복하자 똑같은 출력이 도출됨을 확인할 수 있음.

4. 코드

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity vhd1_2016114172 is
    port(coffee_button,cocoa_button,rst,coin50,coin100,return_button,ck : in std_logic;
          coffee_out, cocoa_out : out integer;
          coin50out,coin100out : out integer );
end vhd1_2016114172;

architecture Behavioral of vhd1_2016114172 is
    type stype is (s0, s50, s100, s150, s200, s250, s300);
    signal st : stype;

begin
    process (coffee_button,cocoa_button,rst,coin50,coin100,return_button)

        variable total : integer := 300; --total은 수중에 남은 돈
        variable coffee_value : integer := 0; --후에 output으로 뽑은 coffee 수를 전달해줄 변수
        variable cocoa_value : integer := 0; --후에 output으로 뽑은 cocoa 수를 전달해줄 변수

        --커피와 코코아는 손에 들고 있기 때문에, 출력 신호를 계속해서 유지하지만,
        --반환되는 동전의 수는 나오는 출력만 보여주면 되기 때문에 신호처리 함.

        begin

            if(rst ='0') then
                st <= s0;
                total := 300;
                coffee_value :=0;
                cocoa_value :=0;
                coin50out <=0;
                coin100out <=0;

            elsif(ck'event and ck='1') then

                case st is

                    when s0 =>
                        if(total >= 50 and coin50 = '1' and coin100 = '0') then
                            st <= s50;
                            coin50out <= 0;
                            coin100out <= 0;
                            coffee_value := coffee_value+0;
                            cocoa_value := cocoa_value +0;
                            total := total - 50;
                        elsif(total >= 100 and coin100 = '1' and coin50 = '0') then
                            st <= s100;
                            coin50out <= 0;
                            coin100out <= 0;
                            coffee_value := coffee_value;
                            cocoa_value := cocoa_value;
                            total := total - 100;
                        else -- 커피(150), 코코아(100), 동전 두 개 이상, 가진 돈이 없을
```

때

```
coin50out <= 0;
coin100out <= 0;
coffee_value := coffee_value;
cocoa_value := cocoa_value;
st <= s0;
end if;
```

```
when s50 =>
  if(return_button = '1') then --반환버튼을 먼저 따진다.
    st <= s0;
    coin50out <= 1;
    coin100out <= 0;
    coffee_value := coffee_value;
    cocoa_value := cocoa_value;
    total := total + 50;
  elsif(total >= 50 and coin50 = '1' and coin100 = '0') then
    st <= s100;
    coin50out <= 0;
    coin100out <= 0;
    coffee_value := coffee_value;
    cocoa_value := cocoa_value;
    total := total - 50;
  elsif(total >= 100 and coin100 = '1' and coin50 = '0') then
    st <= s150;
    coin50out <= 0;
    coin100out <= 0;
    coffee_value := coffee_value;
    cocoa_value := cocoa_value;
    total := total - 100;
  else -- 커피(150), 코코아(100), 동전 두 개 이상
    st <= s100;
    coin50out <= 0;
    coin100out <= 0;
    coffee_value := coffee_value;
    cocoa_value := cocoa_value;
  end if;
```

```
when s100 =>
  if(return_button = '1') then
    st <= s0;
    coin50out <= 0;
    coin100out <= 1;
    coffee_value := coffee_value;
    cocoa_value := cocoa_value;
    total := total + 100;
  elsif(cocoa_button = '1' and coffee_button = '0') then
    st <= s0;
    coin50out <= 0;
    coin100out <= 0;
    coffee_value := coffee_value;
    cocoa_value := cocoa_value + 1;
  elsif(total >= 50 and coin50 = '1' and coin100 = '0') then
    st <= s150;
```

```

        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
        total := total - 50;
    elsif(total >= 100 and coin100 = '1' and coin50 = '0') then
        st <= s200;
        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
        total := total - 100;
    else --커피(150)을 뽑거나, 동전을 2개 이상 넣는 상황
        st <= s100;
        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
    end if;

```

```

when s150 =>
    if(return_button = '1') then
        st <= s0;
        coin50out <= 1;
        coin100out <= 1;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
        total := total + 150;
    elsif(coffee_button = '1' and cocoa_button = '0') then
        st <= s0;
        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value + 1;
        cocoa_value := cocoa_value;
    elsif(cocoa_button = '1' and coffee_button = '0') then
        st <= s50;
        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value + 1;
    elsif(total >= 50 and coin50 = '1' and coin100 = '0') then
        st <= s200;
        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
        total := total - 50;
    elsif(total >= 100 and coin100 = '1' and coin50 = '0') then
        st <= s250;
        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
        total := total - 100;
    else --동전을 2개 이상 넣는 상황, 코코아 커피 둘 다 누른 상황
        st <= s150;
        coin50out <= 0;

```



```

        coin100out <=0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
    end if;

when s200 =>
    if(return_button = '1') then
        st <= s0;
        coin50out <=0;
        coin100out <= 2;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
        total := total + 200;
    elsif(coffee_button = '1' and cocoa_button = '0') then
        st <= s50;
        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value + 1;
        cocoa_value := cocoa_value;
    elsif(cocoa_button = '1' and coffee_button = '0') then
        st <= s100;
        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value + 1;
    elsif(total >= 50 and coin50 = '1' and coin100 = '0') then
        st <= s250;
        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
        total := total - 50;
    elsif(total >= 100 and coin100 = '1' and coin50 = '0') then
        st <= s300;
        coin50out <= 0;
        coin100out <= 0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
        total := total - 100;
    else --동전을 2개 이상 넣는 상황, 코코아 커피 둘 다 누른 상황
        st <= s200;
        coin50out <= 0;
        coin100out <=0;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
    end if;

```

```

when s250 =>
    if(return_button = '1') then
        st <= s0;
        coin50out <=1;
        coin100out <= 2;
        coffee_value := coffee_value;
        cocoa_value := cocoa_value;
        total := total + 250;
    elsif(coffee_button = '1' and cocoa_button = '0') then

```

피 둘 다 누른 상황

```
st <= s100;
coin50out <= 0;
coin100out <= 0;
coffee_value := coffee_value + 1;
cocoa_value := cocoa_value;
elsif(cocoa_button = '1' and coffee_button = '0') then
st <= s150;
coin50out <= 0;
coin100out <= 0;
coffee_value := coffee_value;
cocoa_value := cocoa_value + 1;
elsif(total >= 50 and coin50 = '1' and coin100 = '0') then
st <= s300;
coin50out <= 0;
coin100out <= 0;
coffee_value := coffee_value;
cocoa_value := cocoa_value;
total := total - 50;
else --100원을 넣는 경우, 동전을 2개 이상 넣는 상황, 코코아 커피
st <= s250;
coin50out <= 0;
coin100out <= 0;
coffee_value := coffee_value;
cocoa_value := cocoa_value;
end if;
```

```
when s300 =>
if(return_button = '1') then
st <= s0;
coin50out <= 0;
coin100out <= 3;
coffee_value := coffee_value;
cocoa_value := cocoa_value;
total := total + 300;
elsif(coffee_button = '1' and cocoa_button = '0') then
st <= s150;
coin50out <= 0;
coin100out <= 0;
coffee_value := coffee_value + 1;
cocoa_value := cocoa_value;
elsif(cocoa_button = '1' and coffee_button = '0') then
st <= s200;
coin50out <= 0;
coin100out <= 0;
coffee_value := coffee_value;
cocoa_value := cocoa_value + 1;
else --돈을 넣는 경우, 코코아 커피 둘 다 누른 상황
st <= s300;
coin50out <= 0;
coin100out <= 0;
coffee_value := coffee_value;
cocoa_value := cocoa_value;
end if;
```

end case;

```
end if; --rst if
```

```
coffee_out <= coffee_value;
```

```
cocoa_out <= cocoa_value;
```

```
end process;
```

```
end Behavioral;
```