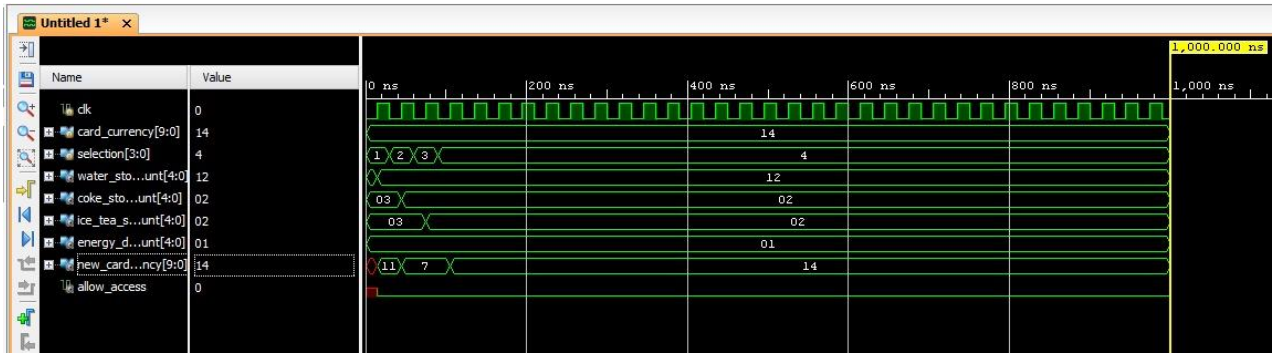
DRINK AUTOMAT

PURPOSE

These are the verilog codes of a 4-compartment vending machine with water, coke, ice tea and energy drinks.

DESCRIPTION



There are 3 inputs clk, card currency, and selection. Clk is a clock, card currency is card currency without any spending, and a selection of what you want to buy. If you select 1 it is water, 2 is coke, 3 is ice tea and 4 is an energy drink. There are 6 outputs water stock amount, coke stock amount, ice tea stock amount, energy drink stock amount, new card currency and allow access. The first 4 are the amounts of drinks in the machine, new card currency is card currency after buying a drink, and allow access is permission to buy. According to the script I wrote in testbench, we have 14 coins in our card. There are 13 glasses of water, 3 cokes, 3 ice teas, and an energy drink in the machine. Prices are 3,7,7 and 15 coins. First, we select 1 and the water stock amount decreases to 12, new card currency turns 11. Then we select 2 and the coke stock amount decreases to 2, new card currency turns 7. Because I reset card currency on testbench for each selection. When we select 3 ice tea stock amount decreases to 2 and new card currency stays at 7 because ice tea's price is the same as coke's. Finally, we select energy drink but we can't buy it. Because our coin is not enough for ice tea. So new card currency stays the same with card currency and energy drink stock amount stays same. Allow access can be 1 or 0. It is 0 because we couldn't buy an energy drink. The first parts of new card currency and allow access is red because there is not any values at first.

CODES

```
`timescale 1ns / 1ps
module drinkAutomat(
    input clk,
    input [9:0] card_currency,
    input [3:0] selection,
    output reg [4:0] water_storage_amount, coke_storage_amount,
ice_tea_storage_amount,energy_drink_storage_amount,
    output reg [9:0] new_card_currency,
    output reg allow_access
    );
    initial begin
    water_storage_amount=5'b01101;
    coke_storage_amount=5'b00011;
    ice_tea_storage_amount=5'b00011;
    energy_drink_storage_amount=5'b00001;
    end
    always@(posedge clk)
    begin
       if(selection == 1'd1)
       begin
          if(water_storage_amount>1'b0)
          begin
             if(card_currency>=3'd3)
             begin
             allow_access=1;
             water_storage_amount=water_storage_amount-1;
             new_card_currency=card_currency-2'd3;
             end
             else
             begin
             allow_access=0;
             end
          end
          else
          begin
          allow_access=0;
          end
       end
     if(selection == 2'd2)
        begin
           if(coke_storage_amount>1'b0)
           begin;
              if(card_currency>=3'd7)
              begin
              allow_access=1;
              coke_storage_amount=coke_storage_amount-1;
              new_card_currency=card_currency-3'd7;
              end
              else
```

```
            begin
            allow_access=0;
            end
        end
        else
        begin
        allow_access=0;
        end
      end
   if(selection == 2'd3)
   begin
      if(ice_tea_storage_amount>1'b0)
      begin
         if(card_currency>=3'd7)
         begin
         allow_access=1;
         ice_tea_storage_amount=ice_tea_storage_amount-1;
         new_card_currency=card_currency-3'd7;
         end
         else
         begin
         allow_access=0;
         end
      end
      else
      begin
      allow_access=0;
      end
   end
   if(selection == 3'd4)
   begin
      if(energy_drink_storage_amount>1'b0)
      begin
         if(card_currency>=4'd15)
         begin
         allow_access=1;
         energy_drink_storage_amount=energy_drink_storage_amount-1;
         new_card_currency=card_currency-4'd15;
         end
         else
         begin
         allow_access=0;
         new_card_currency=card_currency;
         end
      end
   end
   else
   begin
   allow_access=0;
   end
  end
endmodule
```

```
`timescale 1ns / 1ps
module drinkAutomattb;
reg clk;
reg [9:0] card_currency;
reg [3:0] selection;
wire [4:0] water_storage_amount, coke_storage_amount, ice_tea_storage_amount,
energy_drink_storage_amount;
wire [9:0] new_card_currency;
wire allow_access;
drinkAutomat UUT(clk, card_currency, selection, water_storage_amount, coke_storage_amount,
ice_tea_storage_amount, energy_drink_storage_amount, new_card_currency, allow_access);
initial begin
clk=0;
forever
#15 clk=~clk;
end
initial begin
card_currency=10'b0000001110;
selection = 1'd1;
#30;
card_currency=10'b0000001110;
selection = 2'd2;
#30;
card_currency=10'b0000001110;
selection = 2'd3;
#30;
card_currency=10'b0000001110;
selection = 3'd4;
#100;
end
endmodule
```


Erkan Tiryakioğlu

150720051