



# **Automated Gas Station**

Presented by:

Mariam Abdelazim 221007852

Marina Bassem 221004949

Omar Khalifa 221017776

Yassine Ahmed 221027716

Yasmin Ahmed 221005064

Malak Mohamed 221007696

Hannah Mahmoud 221017546

Presented to: Dr Mohamed ElBably

## ❑ Table of contents:

1) Introduction of automated gas station.	Page 3
2) ASM Chart.	Page 4 to 6
3) Steps of ASM chart.	Page 7 to 9
4) VHDL code of the process.	Page 10 to 15
5) Conclusion.	Page 16
6) Picture of chip	Page 17 and 18
7) References	Page 19

## **Introduction:**

With the speed at which technology is developing and the growing need for convenience and efficiency, traditional gas stations are becoming automated engines of innovation. Automated gas stations use modern technologies to improve customer satisfaction and speed operations, bringing about a revolutionary change in the way cars are fueled.

By lowering pollutants, reducing down on idle time, and assisting in converting to cleaner fuels, automated gas stations also significantly contribute to environmental sustainability. We can create a transportation ecology that is more efficient and sustainable by adopting these creative solutions.

## **Steps of ASM chart:**

1. Initial state:  
Initialize ramp and counter to 0. Initialize maximum variable. (maximum number of failed tries)
2. Check if amount of fuel 95 or fuel 92 in the gas station is less than sufficient:  
If it is, display a message to the user then proceed to fuel car
3. Fuel car:  
If 0 then return to initial state  
If 1 then proceed to dispense gas station state and display message.
4. Dispense gas station state
5. Reach gas station:  
If 0 then return to dispense gas station state.  
If 1 then proceed to next statement
6. Statement:  
Amount inside gas station of 92 = amount 92  
Amount inside gas station of 95 = amount 95  
Fuel taken 92 = 0  
Fuel taken 95 = 0
7. Check if amount of fuel 95 or fuel 92 in the gas station is less than sufficient:  
If it is more ,then proceed to car sensor
8. Check if a car is entering the station:  
If the sensor sensed a car entering, proceed to dimension state.  
If it doesn't return to initial state.
9. Dimension state
10. Check if the car is parked in the right dimensions:  
If it is in the right dimensions proceed to car stop.

If it doesn't, display a message that tells if to move forward or backward to be in the right dimension.

11. Car stops:

If the car stops then proceed to open the ramp.

If it doesn't goes back to dimension state.

12. Opens ramp

13. Choice between dispense tank until full or with amount N:

If 1 then dispense tank until full

If 0 then enter the amount of fuel needed to be dispensed

14. Fuel Type:

If 1 then the fuel chosen is fuel 92.

If 0 then the fuel chosen id fuel 95.

15. Dispense fuel 92 state

16. Reach 92:

If 0 then return to Dispense 92 state.

If 1 then add amount to fuel taken 92 then proceed to payment method

17. Dispense fuel 95 state

18. Reach 95:

If 0 then return to Dispense 95 state.

If 1 then add amount to fuel taken 95 then proceed to Payment method.

15.Payment method state

16. Payment method decision:

If 1 then mobile app.

If 0 then credit.

17. Mobile Cash (payment method) state

18. OTP:

If 0 then return to Mobile cash state.

If 1 then proceed to Failed decision.

19. Visa (Payment method) state

20. Pin:

If 0 then return to Visa state.

If 1 then proceed to Failed decision.

21. Failed

If 1 then start counting how many times the transaction has failed.

If 0 then payment transaction is successful then return to initial state.

22. Flag decision:

If 1 then proceed to Alarm state.

If 0 then return to payment method state.

23. Alarm state

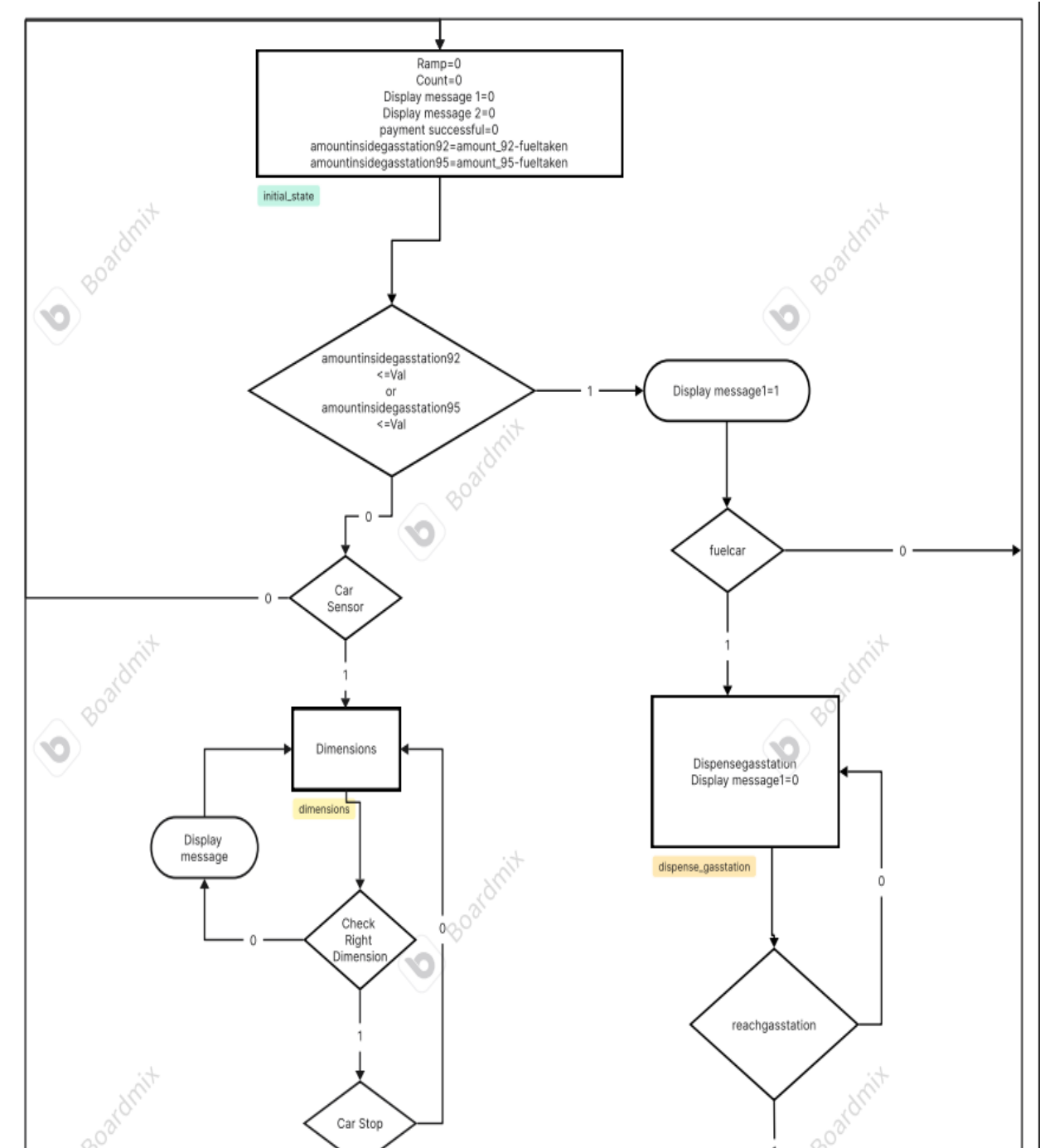
24. Activate Key decision:

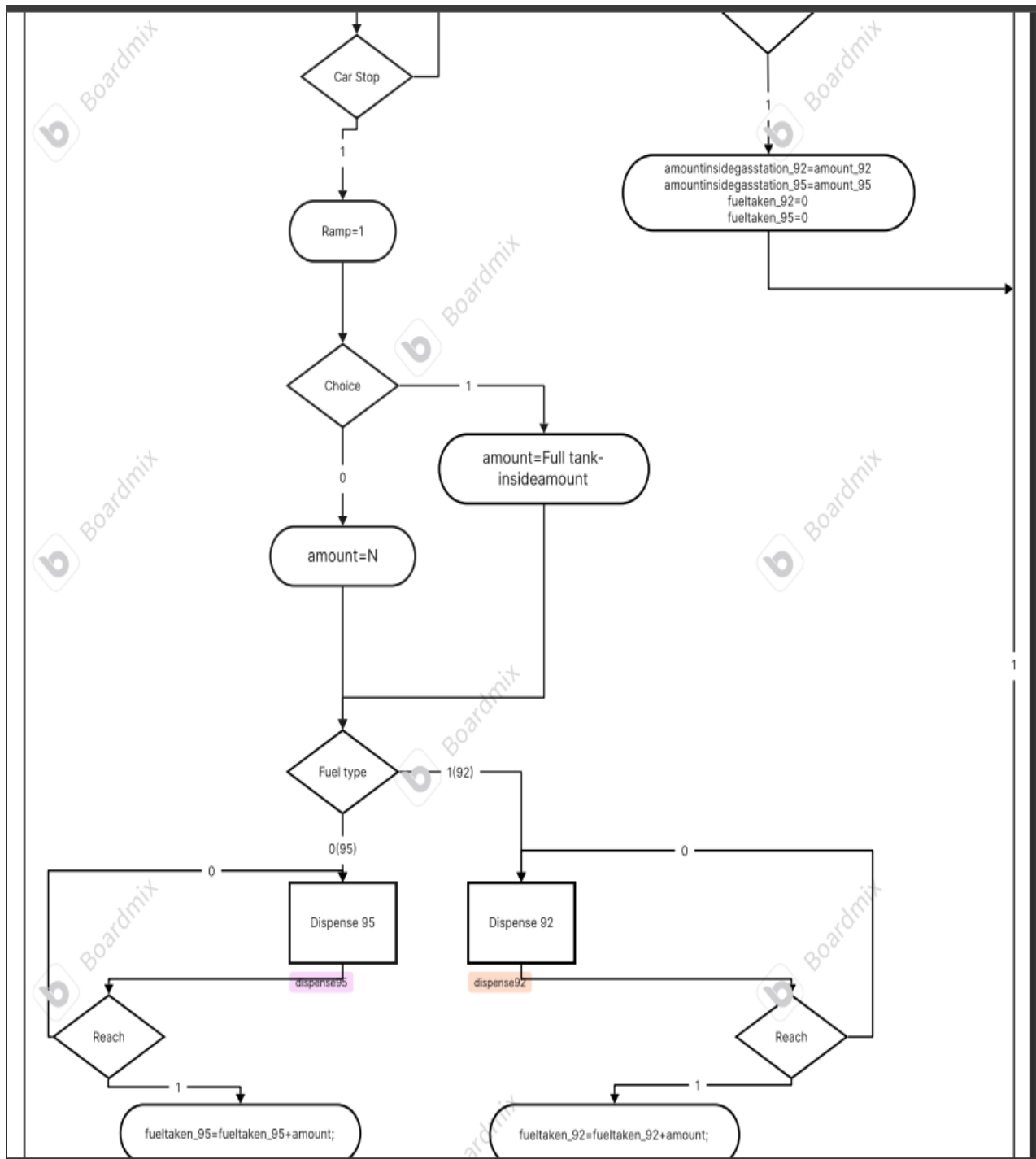
If 0 then return to Alarm state.

If 1 then return to initial state.

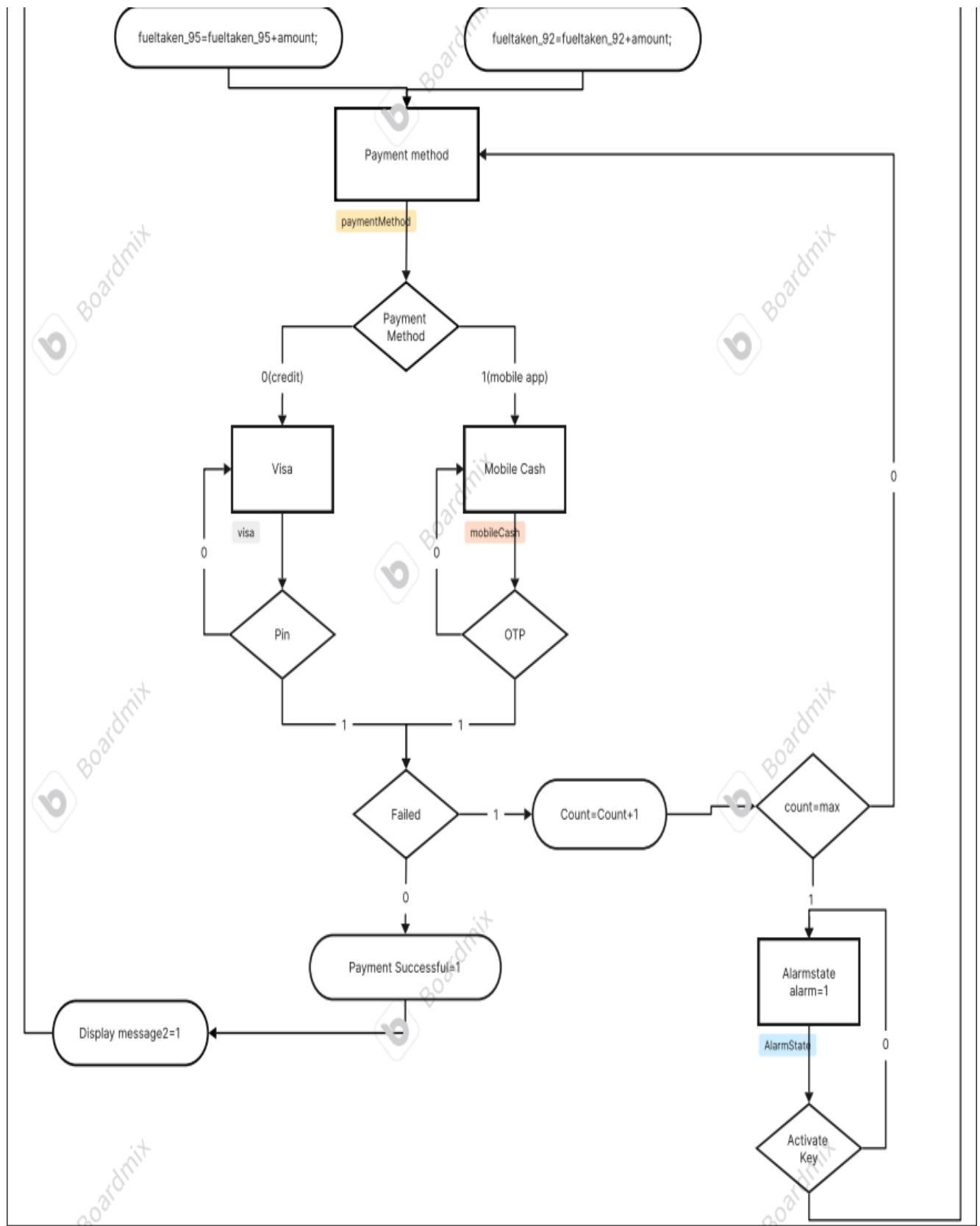
25. Back to Initial state again

**ASM Chart:**









## **VHDL code of the process:**

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.numeric_std.all;
entity gas_station is
  Generic(
    max : integer := 3; -- Maximum value constant
    v: integer := 45; -- Average liter constant
    amount_92:integer:=1000; --maximum amount in the gas station for 92
    amount_95:integer:=1000); --maximum amount in the gas station for 95
  port(
    clk,reset:in std_logic;

    car_sensor,check_dimension,car_stop,choice,fuel_type,reach95,reach92,payment,pin,OTP,reach
    gasstation,failed,activate_key,fuelcar: in std_logic;

    N, FullTank, inside_amount : in integer ;

    ramp,alarm,paymentSuccessful,DisplayMessage1,DisplayMessage2: out std_logic );
  end gas_station;

  architecture behavioural of gas_station is

    type state is(initial_state, dimensions, dispense95, dispense92,paymentMethod,visa,
    mobileCash,AlarmState,dispense_gasstation);

    signal present_state, next_state:state;

    begin

    seq: process(clk, reset)

    begin

    if reset='1' then

    present_state<=initial_state;

    elsif rising_edge(clk) then

    present_state<=next_state;
```

```

end if;

end process seq;

com: process(present_state, car_sensor, check_dimension, car_stop, choice,
fuel_type, reach95, reach92, payment, pin, OTP, failed, flag, activate_key, reachgasstation, fuelcar, N,
FullTank, inside_amount)

variable amountinsidegasstation_92: integer range 0 to amount_92;
variable amountinsidegasstation_95: integer range 0 to amount_95;
variable count: integer range 0 to 3:=0;
variable fueltaken_92: integer range 0 to amount_92:=0;
variable fueltaken_95: integer range 0 to amount_95:=0;
variable amount : integer ;
begin
case present_state is
when initial_state=>
ramp<='0';
alarm<='0';
DisplayMessage1<='0';
DisplayMessage2<='0';
paymentSuccessful<='0';
amountinsidegasstation_92:=amount_92-fueltaken_92;
amountinsidegasstation_95:=amount_95-fueltaken_95;
if (amountinsidegasstation_92<=v) or ( amountinsidegasstation_95<=v) then
DisplayMessage1<='1';
if fuelcar='1' then
next_state <= dispense_gasstation;
else
next_state<=initial_state;
end if;
elsif car_sensor ='1' then

```

```

next_state<=dimensions;
else
next_state<=initial_state;
end if ;
when dimensions =>
if(check_dimension='0')then
report"Please Move" severity note;
next_state<=dimensions;
elsif(car_stop ='1') then
ramp<='1';
else
next_state<=dimensions;
end if;
if(choice='1') then
amount:=FullTank-inside_amount;
else
amount:=N;
end if;
if(fuel_type='1')then
next_state<=dispense92;
else
next_state<=dispense95;
end if;
when dispense95=>
if(reach95='1')then
fueltaken_95:=fueltaken_95+amount;

next_state<=paymentMethod;
else

```

```

next_state<=dispense95;
end if;
when dispense92 =>
if(reach92='1') then
fueltaken_92:=fueltaken_92+amount;
next_state<=paymentMethod;
else
next_state<=dispense92;
end if;
when paymentMethod =>
if(payment='1') then
next_state<=mobileCash;
else
next_state<=visa;
end if;
when visa=>
if(pin='0')then
next_state<=visa;
elsif(failed='1')then
count:=count+1;
if(count=max)then
next_state<=Alarmstate;
else
next_state<=paymentMethod;
end if;
else
paymentSuccessful='1';

DisplayMessage2='1';

```

```

next_state<=initial_state;
end if;

when mobileCash =>
if(OTP='0')then
next_state<=mobileCash;
elsif(failed='1')then
count:=count+1;
if(count=max)then
next_state<=Alarmstate;
else
next_state<=paymentMethod;
end if;
else
paymentSuccessful<='1';
DisplayMessage2<='1';
next_state<=initial_state;
end if;

when alarmState=>
alarm<='1';
if (activate_key ='1') then
next_state<=initial_state;
else
next_state<=alarmstate;
end if;

when dispense_gasstation =>
DisplayMessage1<='0';
if(reachgasstation='1')then
amountinsidegasstation_92:=amount_92;

```

```
amountinsidegasstation_95:=amount_95;
fueltaken_92:=0;
fueltaken_95:=0;
next_state<=initial_state;
else
next_state<=dispense_gasstation;
end if;
end case;
end process com;
end behavioural;
```

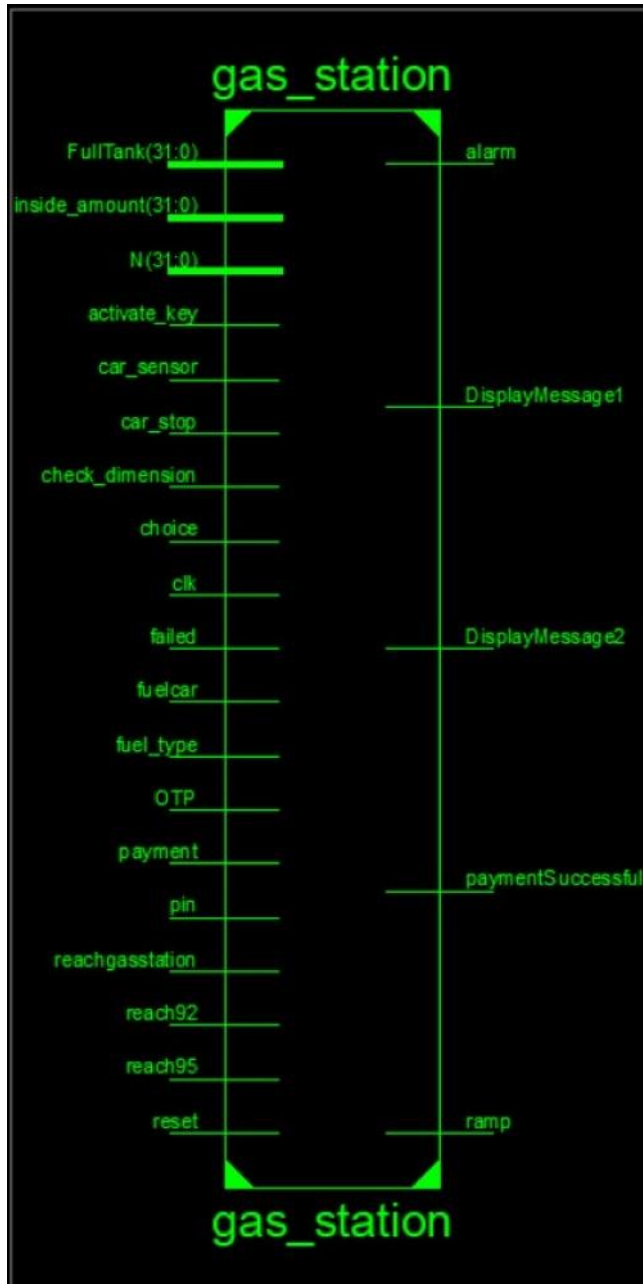
## **Conclusion:**

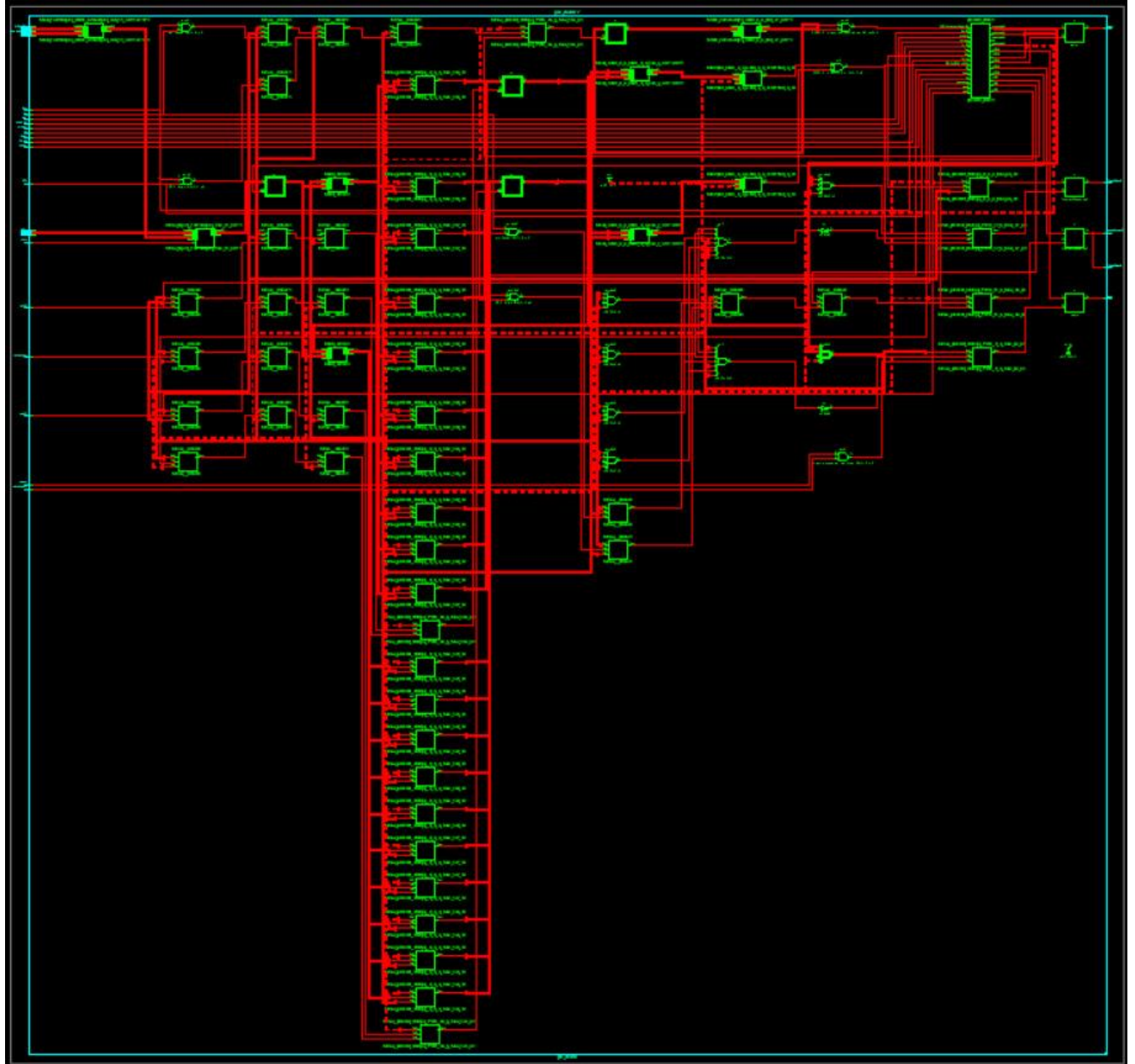
In conclusion, automated gas stations mark a significant advancement in the fueling industry. These creative focuses have completely reimagined the conventional gas station experience by utilizing the latest innovations to provide unmatched sustainability, efficiency, and convenience.

The increasing use of automated gas stations has a lot of potential for the future, both for consumers and station operators. The self-service capabilities, continuously accessibility, and advanced features like remote monitoring and alternative fuel choices make these stations well-suited to adapt to the evolving needs of a world that is changing quickly.



## Chip:





## **References**

*The Finnish station using a robot to fuel up cars.* (2022, November 8). Retrieved from youtube:  
<https://www.youtube.com/watch?v=3NVhDmHxVGs>