VLSI Report
3st Year Computer Engineering

Pizza Machine

Team Members

SN	Student Name	Section Number
1	الاء رائد سعيد عبدالجابر	1
2	امال محمد السعيد ابراهيم	1
3	ايه محمد القدري عامر حسين	1
4	تسنيم عبدالرحمن حامد عبدالوهاب	1
5	رحاب محمد احمد لطفي محمد	2
6	شروق محمد محمود صبري محمد	2
7	فرح محسن سامي محمد	2
8	مريم ايهاب مصطفى ربيع	4
9	منه الله أشرف حنفي عبدالحميد	4
10	هدى محمد عبداللطيف احمد	4

1.Reference Video

https://youtube.com/shorts/lTEVX6Oo8OQ?si=LQ4uidwcRVJN9Qon

2.Sequence for Controller

1. The machine is inactive (initial state).
2. The Operator presses the start button to start the machine.
3. The machine will dough the pizza.
4. The machine will press the dough.
5. The machine makes sure that it’s pressured If it’s not pressured machine will active the pressing.
6. The machine will move the conveyor.
7. The machine will check the existence of pizza before do the next step, if it’s not existed the machine will move the conveyor again.
8. The machine will put the sauce on the pizza dough.
9. The machine will check the existence of the sauce, if it’s not existed the machine will active the sauce.
10. The machine will move the conveyor.
11. The machine will check the existence of pizza before do the next step, if it’s not existed the machine will move the conveyor again.
12. The machine will put cheese.
13. The machine will check the existence of the cheese before do the next step, if it’s not existed the machine will active the cheese.
14. The machine will move the conveyor.
15. The machine will check the existence of pizza before do the next step, if it’s not existed the machine will move the conveyor again.

16. The machine will turn on heater.

17. The machine will check if pizza ready, if not machine will active heater.

18. The machine will move the conveyor.

19. The machine will check the existence of pizza before do the next step, if it’s not existed the machine will move the conveyor again.

20. The machine will check if the customer need pepperoni as additional option.

21. If the Operator needs pepperoni the machine will add pepperoni.

22. The machine check if there is enough pepperoni, if not it will active the pepperoni.

23. The machine will move the conveyor.

24. If the operator press auto machine will return to the starting state to make new pizza.

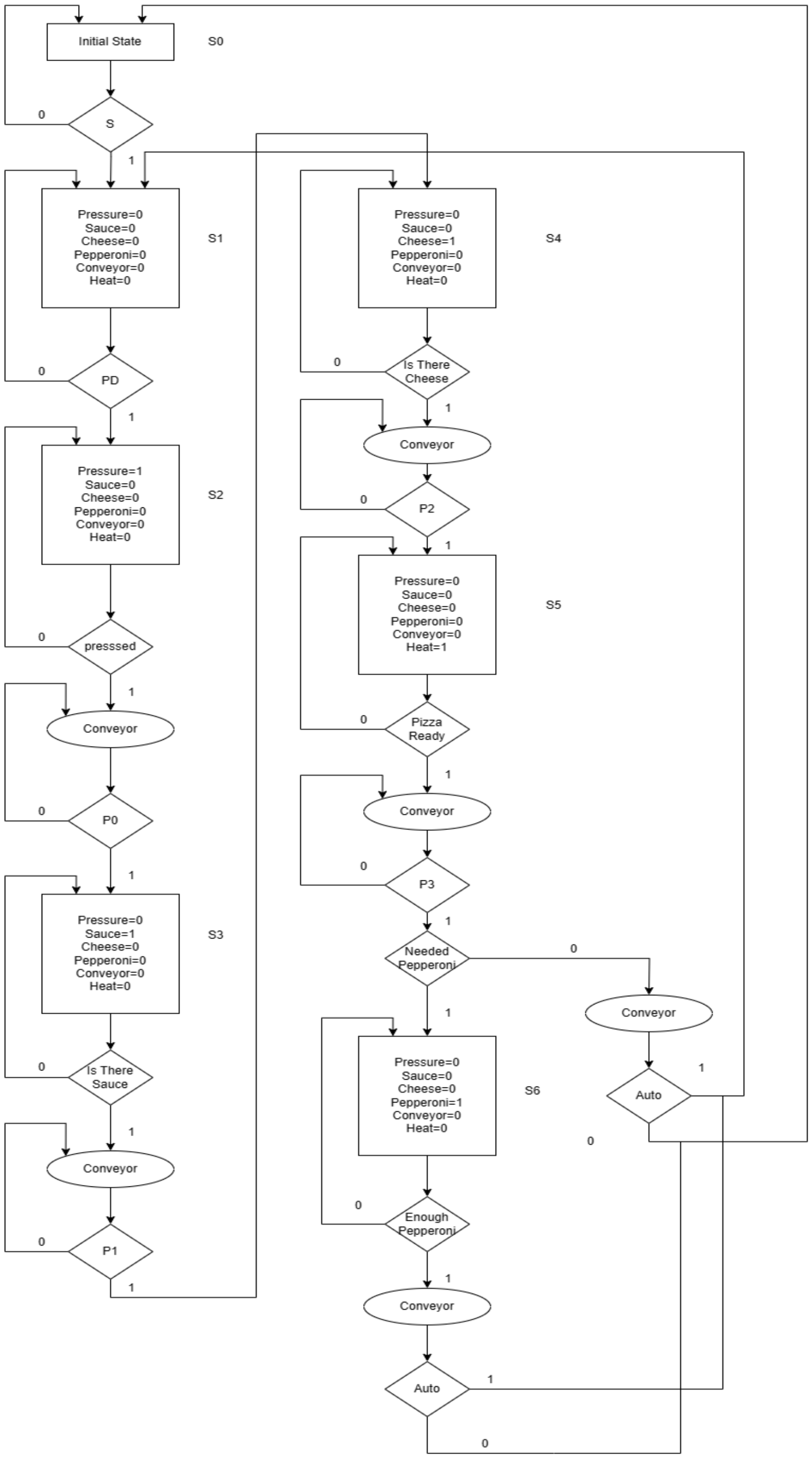
25. If the operator doesn’t press auto machine will stop the machine and wait until he presses start again.

26. If the customer doesn’t need pepperoni the machine moves the conveyor.

27. If the operator press auto machine will return to the pressing state to make new pizza.

28. If the operator doesn’t press auto machine will stop machine and wait until he presses start again.

3.ASM Chart


Initial State
S0
S
0
1
Pressure=0
Sauce=0
Cheese=0
Pepperoni=0
Conveyor=0
Heat=0
S1
PD
0
1
Pressure=1
Sauce=0
Cheese=0
Pepperoni=0
Conveyor=0
Heat=0
S2
presssed
0
1
Conveyor
P0
0
1
Pressure=0
Sauce=1
Cheese=0
Pepperoni=0
Conveyor=0
Heat=0
S3
Is There Sauce
0
1
Conveyor
P1
0
1
Pressure=0
Sauce=0
Cheese=1
Pepperoni=0
Conveyor=0
Heat=0
S4
Is There Cheese
0
1
Conveyor
P2
0
1
Pressure=0
Sauce=0
Cheese=0
Pepperoni=0
Conveyor=0
Heat=1
S5
Pizza Ready
0
1
Conveyor
P3
0
1
Needed Pepperoni
0
1
Conveyor
Auto
1
0
Pressure=0
Sauce=0
Cheese=0
Pepperoni=1
Conveyor=0
Heat=0
S6
Enough Pepperoni
0
1
Conveyor
Auto
1
0

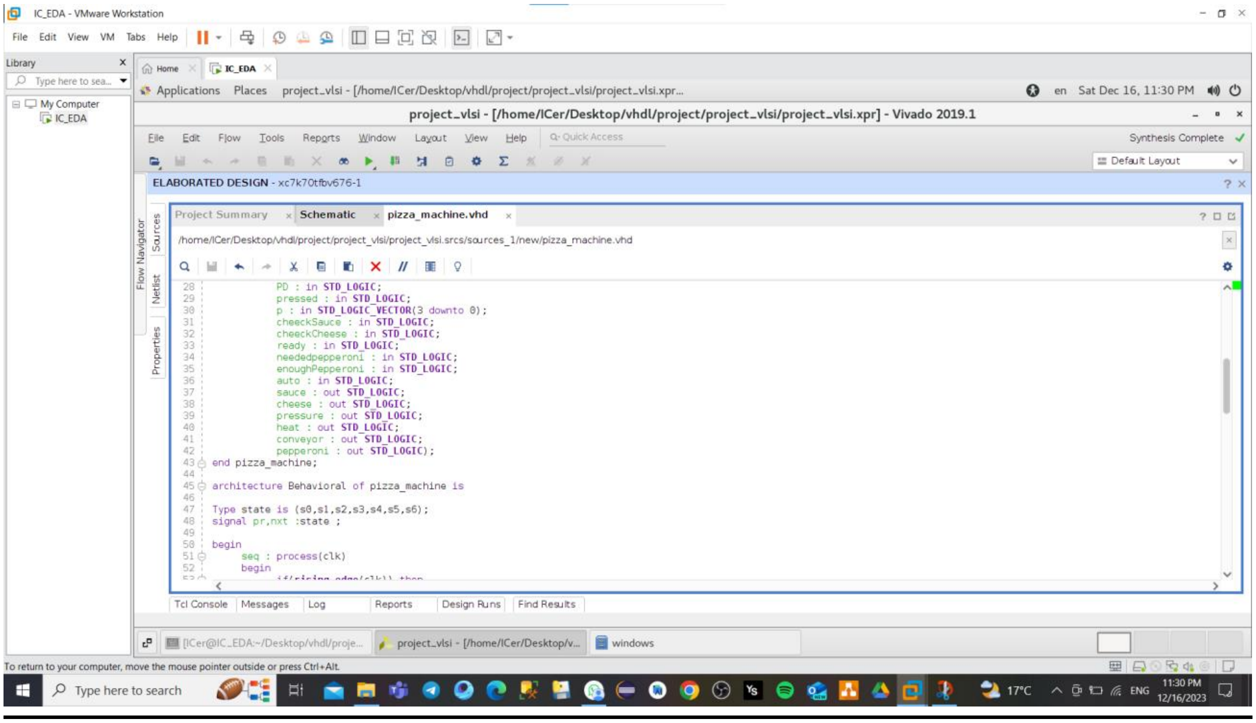
4.VHDL Code

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

entity pizza_machine is
   Port ( clk,rst: in STD_LOGIC;
          start : in STD_LOGIC;
          PD : in STD_LOGIC;
          pressed : in STD_LOGIC;
          p : in STD_LOGIC_VECTOR(3 downto 0);
          cheeckSauce : in STD_LOGIC;
          cheeckCheese : in STD_LOGIC;
          ready : in STD_LOGIC;
          neededpepperoni : in STD_LOGIC;
          enoughPepperoni : in STD_LOGIC;
          auto : in STD_LOGIC;
          sauce : out STD_LOGIC;
          cheese : out STD_LOGIC;
          pressure : out STD_LOGIC;
          heat : out STD_LOGIC;
          conveyor : out STD_LOGIC;
          pepperoni : out STD_LOGIC);
end pizza_machine;

architecture Behavioral of pizza_machine is

Type state is (s0,s1,s2,s3,s4,s5,s6);
signal pr,nxt :state ;

begin
    seq : process(clk)
    begin
          if(rising_edge(clk)) then
             if(rst='1') then  pr <= s0  ;
             else              pr <= nxt ;
             end if;
          end if;
    end process seq ;
```

```
      comb :
process(pr,start,PD,pressed,p,cheeckSauce,cheeckCheese,ready,neededpepperoni,enough
Pepperoni,auto)
    begin
        case pr is
           when s0 =>  sauce <='0'; cheese <='0'; pressure <= '0'; heat <= '0'; conveyor <=
'0'; pepperoni <= '0';
                    if(start = '1')  then  nxt <= s1 ;
                    else                   nxt <= s0 ;
                    end if;
           when s1 =>  sauce <='0'; cheese <='0'; pressure <= '0'; heat <= '0'; conveyor <=
'0'; pepperoni <= '0';
                    if(PD='1')       then  nxt <= s2 ;
                    else                   nxt <= s1  ;
                    end if ;
           when s2 =>  sauce <='0'; cheese <='0'; pressure <= '1'; heat <= '0'; conveyor <=
'0'; pepperoni <= '0';
                    if(pressed='1')       then  conveyor <= '1' ;
                     if(p(0)='1')        then  nxt <= s3 ;
                     else                      conveyor <= '1' ;
                     end if ;
                    else                       nxt <= s2  ;
                    end if ;
           when s3 =>  sauce <='1'; cheese <='0'; pressure <= '0'; heat <= '0'; conveyor <=
'0'; pepperoni <= '0';
                    if(cheeckSauce='1')       then  conveyor <= '1' ;
                     if(p(1)='1')        then  nxt <= s4 ;
                     else                      conveyor <= '1' ;
                     end if ;
                    else                       nxt <= s3  ;
                    end if ;
           when s4 =>  sauce <='0'; cheese <='1'; pressure <= '0'; heat <= '0'; conveyor <=
'0'; pepperoni <= '0';
                    if(cheeckCheese='1')       then  conveyor <= '1' ;
                     if(p(2)='1')        then  nxt <= s5 ;
                     else                      conveyor <= '1' ;
                     end if ;
                    else                       nxt <= s4  ;
                    end if ;
           when s5 =>  sauce <='0'; cheese <='0'; pressure <= '0'; heat <= '1'; conveyor <=
'0'; pepperoni <= '0';
```

```
                    if(ready='1')              then  conveyor <= '1' ;
                     if(p(3)='1') then
                      if(neededpepperoni='1')         then  nxt <= s6 ;
                      else                       conveyor <= '1' ;
                       if(auto<='1')           then  nxt <= s1 ;
                       else                      nxt <= s0 ;
                       end if ;
                      end if ;
                     else                 conveyor <= '1' ;
                     end if ;
                    else                   nxt <= s5  ;
                    end if ;
           when s6 =>  sauce <='0'; cheese <='0'; pressure <= '0'; heat <= '0'; conveyor <= '0'; pepperoni <= '1';
                    if(enoughPepperoni='1')            then  conveyor <= '1' ;
                     if(auto='1')        then  nxt <= s1 ;
                     else                  nxt <= s0 ;
                     end if ;
                    else                   nxt <= s6  ;
                    end if ;
        end case;

    end process comb ;

end Behavioral;
```

5.No Syntax Error



project_vlsi - [/home/ICer/Desktop/vhdl/project/project_vlsi/project_vlsi.xpr] - Vivado 2019.1

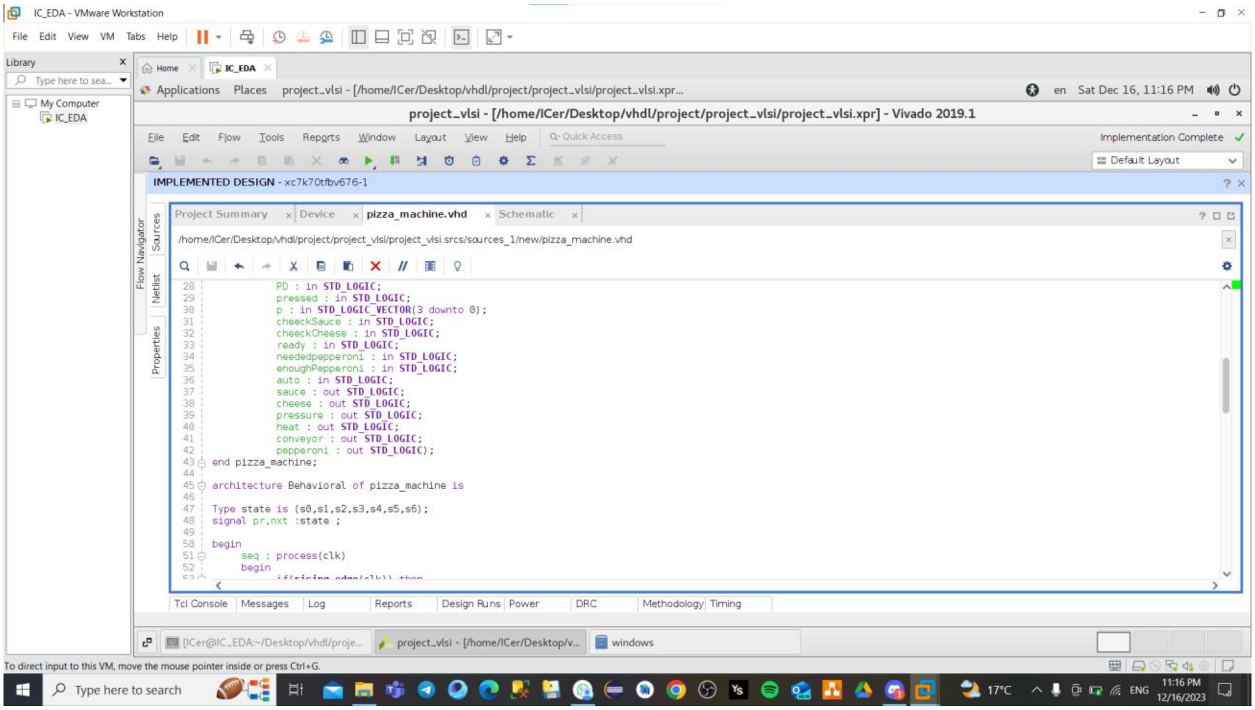
Synthesis Complete ✓

ELABORATED DESIGN - xc7k70tfbv676-1

/home/ICer/Desktop/vhdl/project/project_vlsi/project_vlsi.srcs/sources_1/new/pizza_machine.vhd

```
            PD : in STD_LOGIC;
            pressed : in STD_LOGIC;
            p : in STD_LOGIC_VECTOR(3 downto 0);
            cheeckSauce : in STD_LOGIC;
            cheeckCheese : in STD_LOGIC;
            ready : in STD_LOGIC;
            neededpepperoni : in STD_LOGIC;
            enoughPepperoni : in STD_LOGIC;
            auto : in STD_LOGIC;
            sauce : out STD_LOGIC;
            cheese : out STD_LOGIC;
            pressure : out STD_LOGIC;
            heat : out STD_LOGIC;
            conveyor : out STD_LOGIC;
            pepperoni : out STD_LOGIC);
end pizza_machine;

architecture Behavioral of pizza_machine is

Type state is (s0,s1,s2,s3,s4,s5,s6);
signal pr,nxt :state ;

begin
     seq : process(clk)
     begin
```

Synthesis complete



project_vlsi - [/home/ICer/Desktop/vhdl/project/project_vlsi/project_vlsi.xpr] - Vivado 2019.1

Implementation Complete ✓

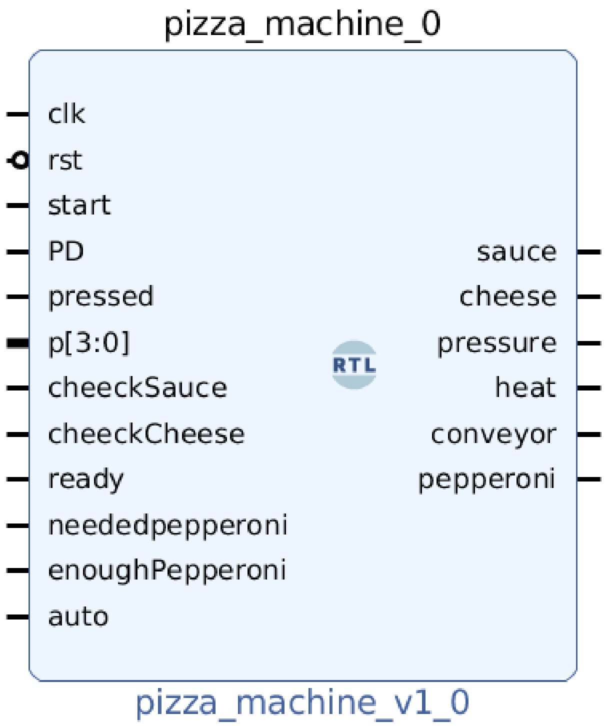
IMPLEMENTED DESIGN - xc7k70tfbv676-1

/home/ICer/Desktop/vhdl/project/project_vlsi/project_vlsi.srcs/sources_1/new/pizza_machine.vhd

```
            PD : in STD_LOGIC;
            pressed : in STD_LOGIC;
            p : in STD_LOGIC_VECTOR(3 downto 0);
            cheeckSauce : in STD_LOGIC;
            cheeckCheese : in STD_LOGIC;
            ready : in STD_LOGIC;
            neededpepperoni : in STD_LOGIC;
            enoughPepperoni : in STD_LOGIC;
            auto : in STD_LOGIC;
            sauce : out STD_LOGIC;
            cheese : out STD_LOGIC;
            pressure : out STD_LOGIC;
            heat : out STD_LOGIC;
            conveyor : out STD_LOGIC;
            pepperoni : out STD_LOGIC);
end pizza_machine;

architecture Behavioral of pizza_machine is

Type state is (s0,s1,s2,s3,s4,s5,s6);
signal pr,nxt :state ;

begin
     seq : process(clk)
     begin
```

Implementation complete

6.Schematic ASIC


pizza_machine_0
clk
rst
start
PD
pressed
p[3:0]
cheeckSauce
cheeckCheese
ready
neededpepperoni
enoughPepperoni
auto
sauce
cheese
pressure
heat
conveyor
pepperoni
RTL
pizza_machine_v1_0