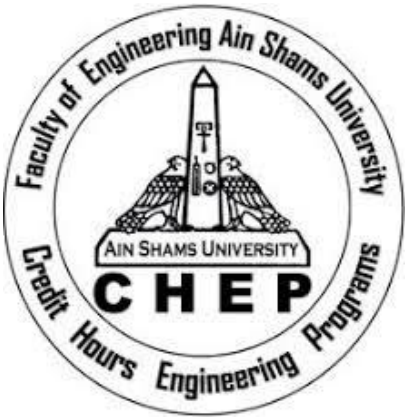
CSE215: EDA

Phase (1)

Project



Prepared by: Ayman Hesham Mohamed

ID: 16P3037

Email: ayman19982016@outlook.com

Department: Computer Engineering (CESS)

Group:2

Introduction:

The code is about to implement a digital access control with a correct code "26A05" and it operates at 2 modes which are daytime and night. At daytime the door opens when pressing "O" or entering the complete correct code or pressing "O" in the middle between any digit of the correct code and at the night it only opens if the code is correct otherwise the alarm has to be triggered.

FSM Code:

```
entity Digital_Access_Control is
port  (
          ck    : in bit;
          vdd   : in          bit;
          vss   : in bit;
          inp     : in          bit_vector (3 downto 0);
          daytime    : in    bit;
          reset : in          bit;
          alarm     : out     bit;
          door    : out       bit
     );
end Digital_Access_Control;

architecture FSM of Digital_Access_Control is
type STATE_TYPE is (S0, S1, S2, S3, S4);
 signal NS, CS : STATE_TYPE;
 begin
  process (CS, inp, daytime, reset)
  begin
   if (reset='1') then
       alarm <='0';
                    door <='0';
       NS<=S0;
   else
     case CS is
                    when S0 =>
                    if(inp="1101" and daytime='1') then --sobh w gali O
```

```
            alarm <='0';
            door <='1';
elsif (inp="0010") then -- gali 2
            alarm <='0';
            door <='0';
            NS<=S1;

else -- ay haga tanya alarm terouh state error
            alarm <='1';
            door <='0';

end if ;

when S1 =>
if(inp="1101" and daytime='1') then --sobh w gali O
            alarm <='0';
            door <='1';
            NS<=S0;


elsif (inp="0110") then -- gali 6
            alarm <='0';
            door <='0';
            NS<=S2;

else -- ay haga tanya alarm terouh state error
            alarm <='1';
            door <='0';
            NS<=S0;
```

```
end if ;
when S2 =>
if(inp="1101" and daytime='1') then --sobh w gali O
        alarm <='0';
        door <='1';
        NS<=S0;

elsif (inp="1010") then -- gali A
        alarm <='0';
        door <='0';
        NS<=S3;

else -- ay haga tanya alarm terouh state error
        alarm <='1';
        door <='0';
        NS<=S0;

end if ;

when S3 =>
if(inp="1101" and daytime='1') then --sobh w gali O
        alarm <='0';
        door <='1';
        NS<=S0;

elsif (inp="0000") then -- gali 0
        alarm <='0';
        door <='0';
```

```
                NS<=S4;
            else -- ay haga tanya alarm terouh state error
                alarm <='1';
                door <='0';
                NS<=S0;

            end if ;

            when S4 =>
            if(inp="1101" and daytime='1') then --sobh w gali O
                alarm <='0';
                door <='1';
                NS<=S0;

            elsif (inp="0101") then -- gali 5
                alarm <='0';
                door <='1';
                NS<=S0;

            else -- ay haga tanya alarm terouh state error
                alarm <='1';
                door <='0';
                NS<=S0;

            end if ;

        end case;
  end if;
end process;
```

```
  process(ck)
 begin
          if(ck = '1' and  ck'event)then
    CS <= NS;
   end if;
 end process;

end FSM;
```

TestBench :

```
ENTITY testbench_Digital_Access_Control IS
END ENTITY testbench_Digital_Access_Control;

ARCHITECTURE test_Digital_Access_Control OF testbench_Digital_Access_Control IS
Component Digital_Access_Control is
port  (
        ck    : in bit;
        vdd   : in          bit;
        vss   : in bit;
        inp     : in        bit_vector (3 downto 0);
        daytime    : in     bit;
        reset : in          bit;
        alarm     : out     bit;
        door    : out       bit
    );
end Component Digital_Access_Control;

FOR dut: Digital_Access_Control USE ENTITY WORK.Digital_Access_Control (FSM);

SIGNAL ck    : bit := '0';
SIGNAL vdd   : bit := '1';
SIGNAL vss   : bit := '0';
SIGNAL inp     : bit_vector (3 downto 0);
SIGNAL daytime    : bit;
SIGNAL reset : bit ;
SIGNAL door   : bit := '0';
```

```
SIGNAL alarm   : bit := '0';

--constant clk_period : time := 50 ns;

BEGIN

dut: Digital_Access_Control PORT MAP (ck, vdd, vss, inp, daytime, reset, alarm, door);
 clk_process :process
   begin
       ck <= '1';
       wait for 25 ns;
       ck <= '0';
       wait for 25 ns;
   end process;

p1: process

begin



-- test 0 ( i use the reset button )
reset<='1';
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 0 at state 0"
severity error;

--test 1 (we use 'O' at the begin )
reset<='0';
```

```
daytime<='1';
inp<="1101";
wait for 50 ns ;
Assert (door='1') and (alarm='0')
report "ERROR in test 1 at state 0"
severity error;

--test 2 (a wrong input )

--correct input '2':

inp<="0010";
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 2 at state 0"
severity error;


inp<="0000";
wait for 50 ns ;
Assert (door='0') and (alarm='1')
report "ERROR in test 2 at state 1"
severity error;

--test 3 (the daytime at morining and the normal sequence "26A05" )

--correct input '2' :
reset<='0';
daytime<='1';
```

```
inp<="0010";
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 3 at state 0" severity error;
--correct input '6':

inp<="0110";
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 3 at state 1" severity error;


--correct input'A':

inp<="1010";
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 3 at state 2" severity error;


--correct input'0':

inp<="0000";
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 3 at state 3 " severity error;


--correct input '5':
```

```
inp<="0101";
wait for 50 ns ;
Assert (door='1') and (alarm='0')
report "ERROR in test 3 at state 4 " severity error;


--test 4 (we use button 'O' at night)

daytime<='0';
inp<="1101";
wait for 50 ns ;
Assert (door='0') and (alarm='1')
report "ERROR in test 4 at state 1" severity error;



--test 5 (the daytime at night and the normal sequence "26A05" )

--correct input '2' :

daytime<='0';
inp<="0010";
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 5 at state 0" severity error;


--correct input '6':
```

```
inp<="0110";
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 5 at state 1" severity error;


--correct input'A':

inp<="1010";
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 5 at state 2" severity error;


--correct input'0':

inp<="0000";
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 5 at state 3 " severity error;


--correct input '5':

inp<="0101";
wait for 50 ns ;
Assert (door='1') and (alarm='0')
report "ERROR in test 5 at state 4 " severity error;
```

```
-- test 6 (a wrong input at night )

--correct input '2':
daytime<='0';
inp<="0010";
wait for 50 ns ;
Assert (door='0') and (alarm='0')
report "ERROR in test 6 at state 0" severity error;


--wrong input '3'

inp<="0011";
wait for 50 ns ;
Assert (door='0') and (alarm='1')
report "ERROR in test 6 at state 0" severity error;


WAIT;
END PROCESS p1;
END ARCHITECTURE test_Digital_Access_Control;
```


Wave - Default
Msgs
/testbench_digital_...
1
1
0
4'h3
0
0
0
1
4'h0 4'hD 4'h2 4'h0 4'h2 4'h6 4'hA 4'h0 4'h5 4'hD 4'h2 4'h6 4'hA 4'h0 4'h5 4'h2 4'h3
ck
vdd
vss
inp
daytime
reset
door
alarm
Name
Now
1200 ns
Cursor 1
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns

Test Strategy :

Input	Daytime	Reset	Door	Alarm
X	X	1	0	0
2	1	0	0	0
6	1	0	0	0
A	1	0	0	0
0	1	0	0	0
5	1	0	1	0
2	1	0	0	0
0	1	0	0	1
2	0	0	0	0
8	0	0	0	1
2	0	0	0	0
6	0	0	0	0
A	0	0	0	0
0	0	0	0	0
5	0	0	1	0
2	0	0	0	0
2	0	0	0	1
O	0	0	0	1
O	1	0	1	0

Note:

I choose those testcases to test the button ‘O’ at morning and night

I choose a test case of the successful way (26A05) at morning and night

I choose a fail way at the morning and night.

Note:

I used the mealy outputs because we wait for change in inputs to have output and it make less states than moore output.

State Diagram: