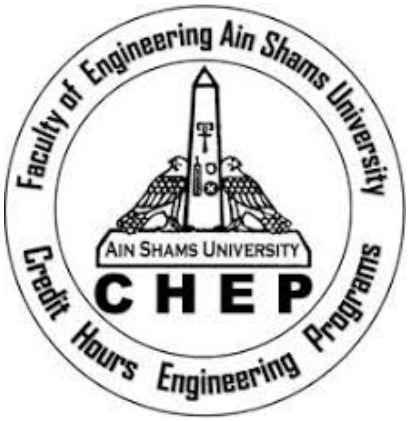
CSE215: EDA

Project

Phase (1)



Prepared by: Yara Hossam Mohamed

Email: youra_97@hotmail.com

Department: Computer Engineering

(CESS)

Group:2

ID: 16P3002

Introduction:

In this phase of the project, I was implementing a digital access control which has a correct entry code “26A05”. It operates in two modes: Daytime (in the morning) that can opens if the user press only “O” or one or more right digits then “O” for example “26” then “O”.

Although at the night the door only opens if the code is correct otherwise the alarm is triggered in case of any incorrect digit.

The code:

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


     );
end Digital_Access_Control;

architecture FSM of Digital_Access_Control is
  type STATE_TYPE is (start, s2, s6, sa, s0);
  signal NS, CS : STATE_TYPE;

begin
-- Process (1): Transition and Generation functions
  process (CS, daytime, code, reset)
  begin
-- Next state Transition function = f(inputs, current state)
-- output generation function =f(inputs, states)
    if (reset='1') then
      door<= '0';
      alarm<= '0';
```

```
      NS<=start;
    else
      case CS is
         when start =>
          if (code="0010") then -- if code = 2 and daytime don't care
              door<= '0';
              alarm <= '0';
           NS <= s2;
          elsif  (daytime = '1' and code="1101") then --open and daytime =
1
                door <= '1';
                alarm <= '0';
           NS <= start;
           else
                door<= '0';
                alarm<= '1';
           NS <= start;
       end if;

        when s2 =>
              if (code="0110") then -- if input is 6
                door<= '0';
                alarm<= '0';
           NS <= s6;
              elsif  ( daytime = '1' and code="1101") then --open and
daytime = 1
                    door<= '1';
                    alarm<= '0';
           NS <= start;
```

```
            else
              door<= '0';
              alarm<= '1';
          NS <= start;
        end if;
      when s6 =>
            if (code="1010") then -- if input is A
              door<= '0';
              alarm<= '0';
          NS <= sa;
            elsif  (code="1101" and daytime = '1')then --open and
daytime = 1
                door<= '1';
                alarm<= '0';
          NS <= start;
            else

              door<= '0';
              alarm<= '1';
          NS <= start;
        end if;
        when sa =>
            if (code="0000") then -- if input is 0
              door<= '0';
              alarm<= '0';
          NS <= s0;
            elsif  (code="1101" and daytime = '1') then--open and
daytime = 1
                door<= '1';
```

```
                    alarm<= '0';
            NS <= start;
                 else
                   door<= '0';
                   alarm<= '1';
            NS <= start;
               end if;

                 when s0 =>
                 if (code="0101") then -- if input is 5
                   door<= '1';
                   alarm<= '0';
            NS <= start;
                    elsif  (code="1101" and daytime = '1') then--open and
daytime = 1
                        door<= '1';
                        alarm<= '0';
                  NS <= start;
                  else
                        door<= '0';
                        alarm<= '1';
                                              NS <= start;
          end if;
        end case;
     end if;
  end process;

-- Process (2): State update (sequential)
  process(clk)
```

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

end FSM;
```

TestBench Code:

```
-- Entity declaration for your testbench. Don't declare any ports here
ENTITY testbench_digitalaccesscontrol IS
END ENTITY testbench_digitalaccesscontrol;

ARCHITECTURE testtestbench_digitalaccesscontrol OF
testbench_digitalaccesscontrol IS

-- Component Declaration for the Device Under Test (DUT)
component Digital_Access_Control is
port  (
        clk    : in     bit;
        vdd    : in     bit;
        vss    : in     bit;
        code       : in bit_vector (3 downto 0);
        daytime     : in     bit;
       reset : in bit;
        door     : out        bit;
        alarm     : out       bit
     );
END component Digital_Access_Control;

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

-- Declare input signals and initialize them
SIGNAL clk    : bit := '0';
SIGNAL vdd    : bit := '1';
```

```
SIGNAL vss   : bit := '0';
SIGNAL code     : bit_vector (3 downto 0);
SIGNAL daytime    : bit := '0';
SIGNAL reset    : bit := '0';
SIGNAL door : bit := '0';
SIGNAL alarm   : bit := '0';


-- Constants and Clock period definitions
constant clk_period : time := 50 ns;
BEGIN

-- Instantiate the Device Under Test (DUT)
dut: Digital_Access_Control PORT MAP (clk, vdd, vss, code, daytime,
reset, door, alarm);

-- Clock process definitions
  clk_process :process
  begin
       clk <= '1';
       wait for clk_period/2;
       clk <= '0';
       wait for clk_period/2;
  end process;

-- Stimulus process, refer to clock signal
proc: PROCESS IS
BEGIN
        --test0
```

```
 reset<='1';
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not 2"
 SEVERITY error;

 --test1
 reset <= '0'; code <= "0010" ; daytime <= '1';
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not 2"
 SEVERITY error;

code <= "0110" ;
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not 6"
 SEVERITY error;

code <= "1010" ;
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not A"
 SEVERITY error;

code <= "1101" ;
 WAIT FOR 50 ns;
 ASSERT door = '1' and alarm = '0'
```

```
 REPORT "Error , can't open!"
 SEVERITY error;


--test2

 reset <= '0'; code <= "0010" ; daytime <= '0';
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not 2"
 SEVERITY error;


code <= "0110" ;
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not 6"
 SEVERITY error;


code <= "1010" ;
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not A"
 SEVERITY error;


code <= "0000" ;
 WAIT FOR 50 ns;
```

```
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not zero"
 SEVERITY error;

code <= "0101" ;
 WAIT FOR 50 ns;
 ASSERT door = '1' and alarm = '0'
 REPORT "Error not 5"
 SEVERITY error;

 --test3


 reset <= '0'; code <= "0010" ; daytime <= '0';
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not 2"
 SEVERITY error;


code <= "0110" ;
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not 6"
 SEVERITY error;


code <= "1010" ;
 WAIT FOR 50 ns;
```

```
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not A"
 SEVERITY error;


code <= "0000" ;
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not Zero"
 SEVERITY error;


code <= "0111" ;
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '1'
 REPORT "Error can't read not 5"
 SEVERITY error;

--test4
 reset <= '0'; code <= "0010" ; daytime <= '1';
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '0'
 REPORT "Error not 2"
 SEVERITY error;


code <= "0111" ;
 WAIT FOR 50 ns;
 ASSERT door = '0' and alarm = '1'
```

```
        REPORT "Error can't read not 6"
        SEVERITY error;

       --test5
        reset <= '0'; code <= "1101" ; daytime <= '1';
        WAIT FOR 50 ns;
        ASSERT door = '1' and alarm = '0'
        REPORT "Error can't open"
        SEVERITY error;


WAIT; -- stop process simulation run

END PROCESS proc;
END ARCHITECTURE testtestbench_digitalaccesscontrol;
```

Screenshot of the wave:

Test Strategy:

Test Cases	reset	code	daytime	Delay	Expected door	Expected alarm
Test 0	1	X	X	50 ns	0	0
Test 1	0	2	1		0	0
	0	6	1		0	0
	0	A	1		0	0
	0	O	1		1	0
Test 2	0	2	0		0	0
	0	6	0		0	0
	0	A	0		0	0
	0	0	0		0	0
	0	5	0		1	0
Test 3	0	2	0		0	0
	0	6	0		0	0
	0	A	0		0	0
	0	0	0		0	0
	0	7	0		0	1
Test 4	0	2	1		0	0
	0	7	1		0	1
Test 5	0	O	1		1	0

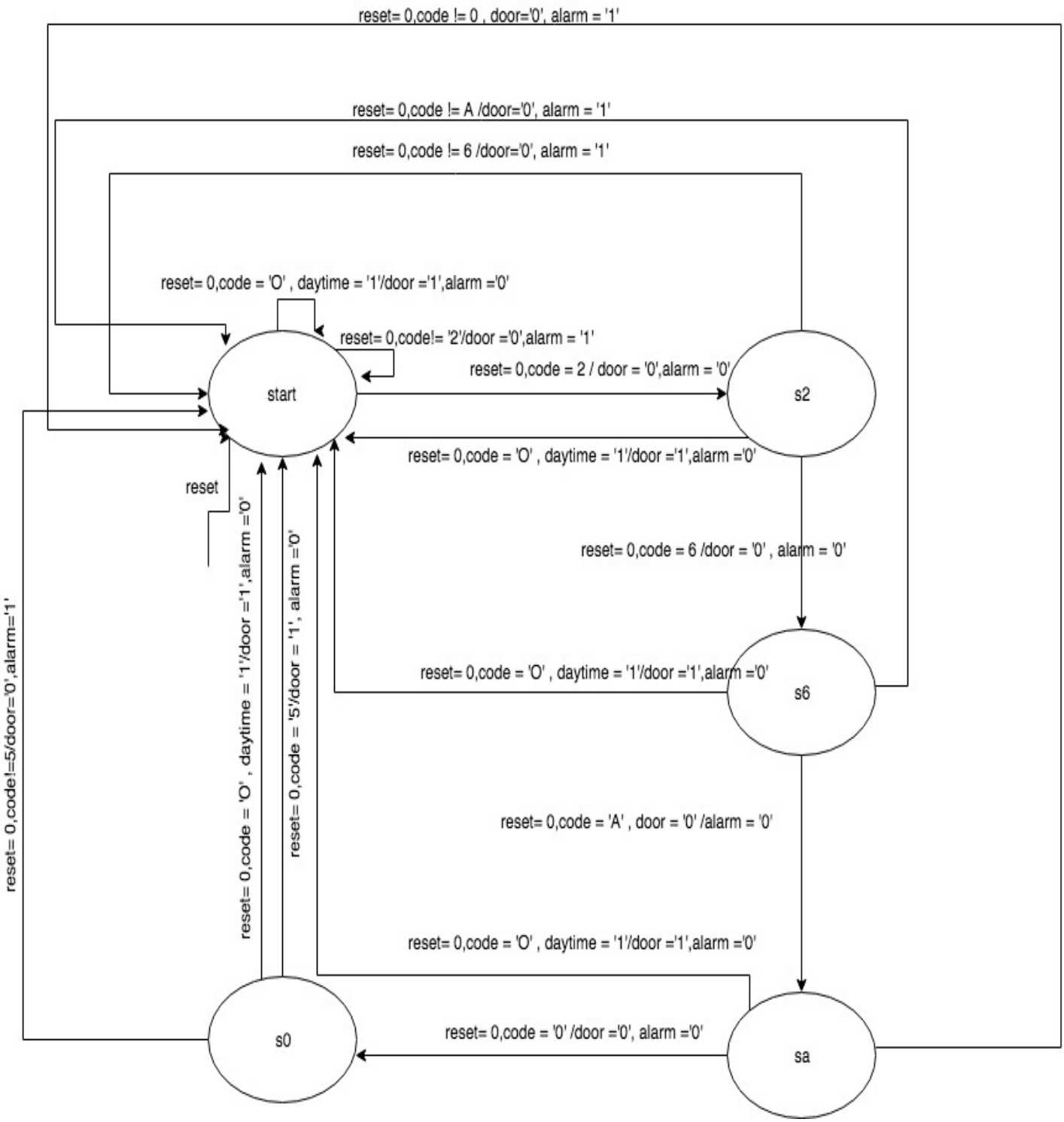
Note:

I have chosen those test cases to cover each one of those:

1. The reset input so that when it is on the door and the alarm are zeros
2. To test the daytime when it is set to one and the input is for example “26A” and then “O” (refers to open).
3. Also to test at the night and the code is correct

4. And test at the night but with wrong digit at the end of the code “26A07” which is wrong code.
5. Test a wrong code at the morning (with daytime =1) “27” so the alarm has to be 1
6. Test to only entering “O” at the morning (daytime=1) and the door has to open (Assuming that the mechanism implies that if the user enters “O” at the morning it has to open at this time).

State Diagram:



Note:

I have implement using mealy output because I preferred that when there is a transition between two states when the input change the output occurs. So it will have less states than while using the moore outputs.