CSE:215 EDA

Digital Access Control
Project 2

Name: Ahmed Bahaa Ibrahim
ID: 16P6057
Email:ahmedbahaa.6251@gmail.com
Department: CESS

Introduction:

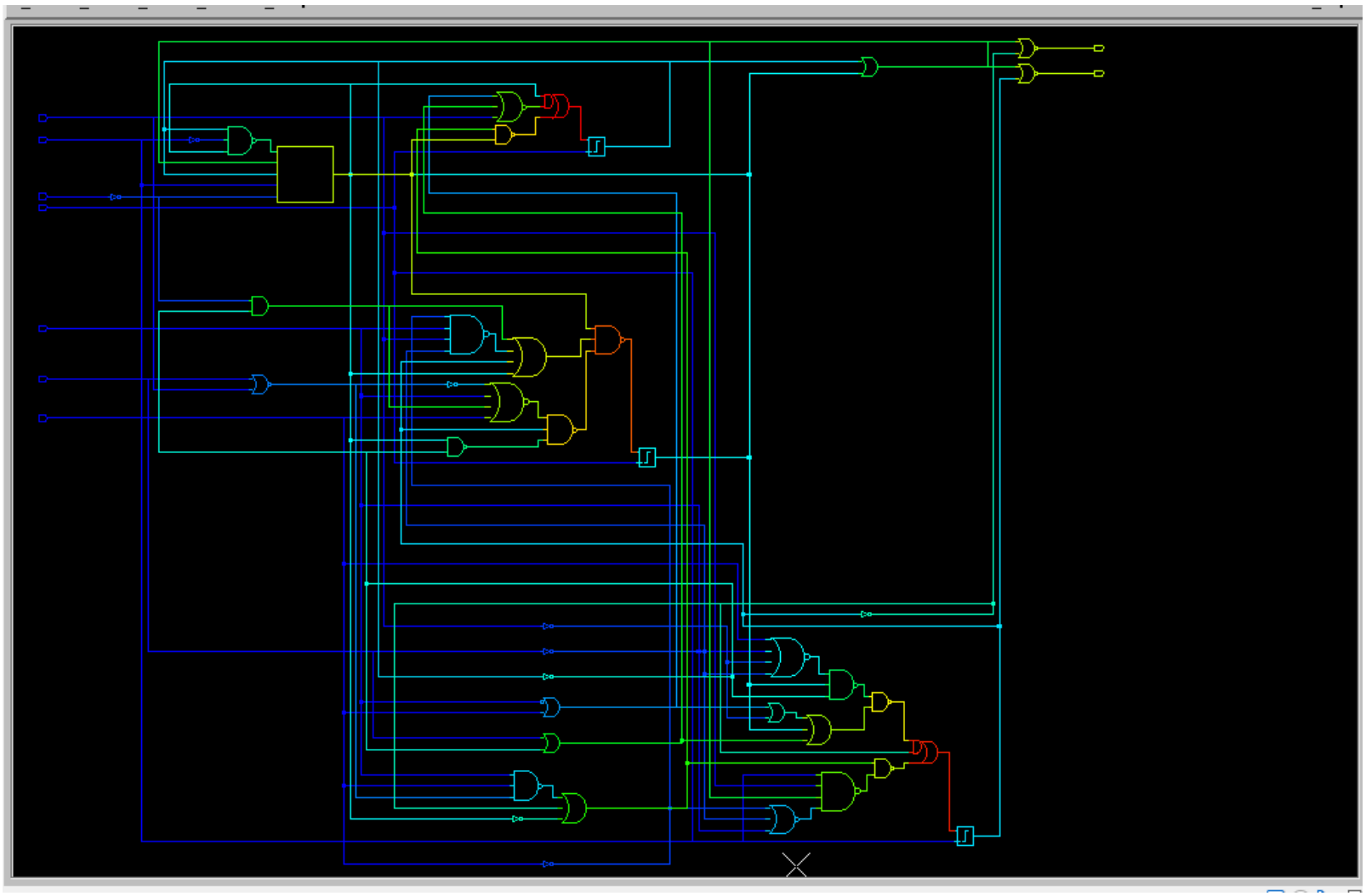
This is a report for Part 2 of the Project of Electronic Design Automation which is about a finite state machine system which is Digital Access Control.in this part I passed through different Alliance Tools to Synthesis the VHDL Code Developed in Phase 1.

This Report consist of :

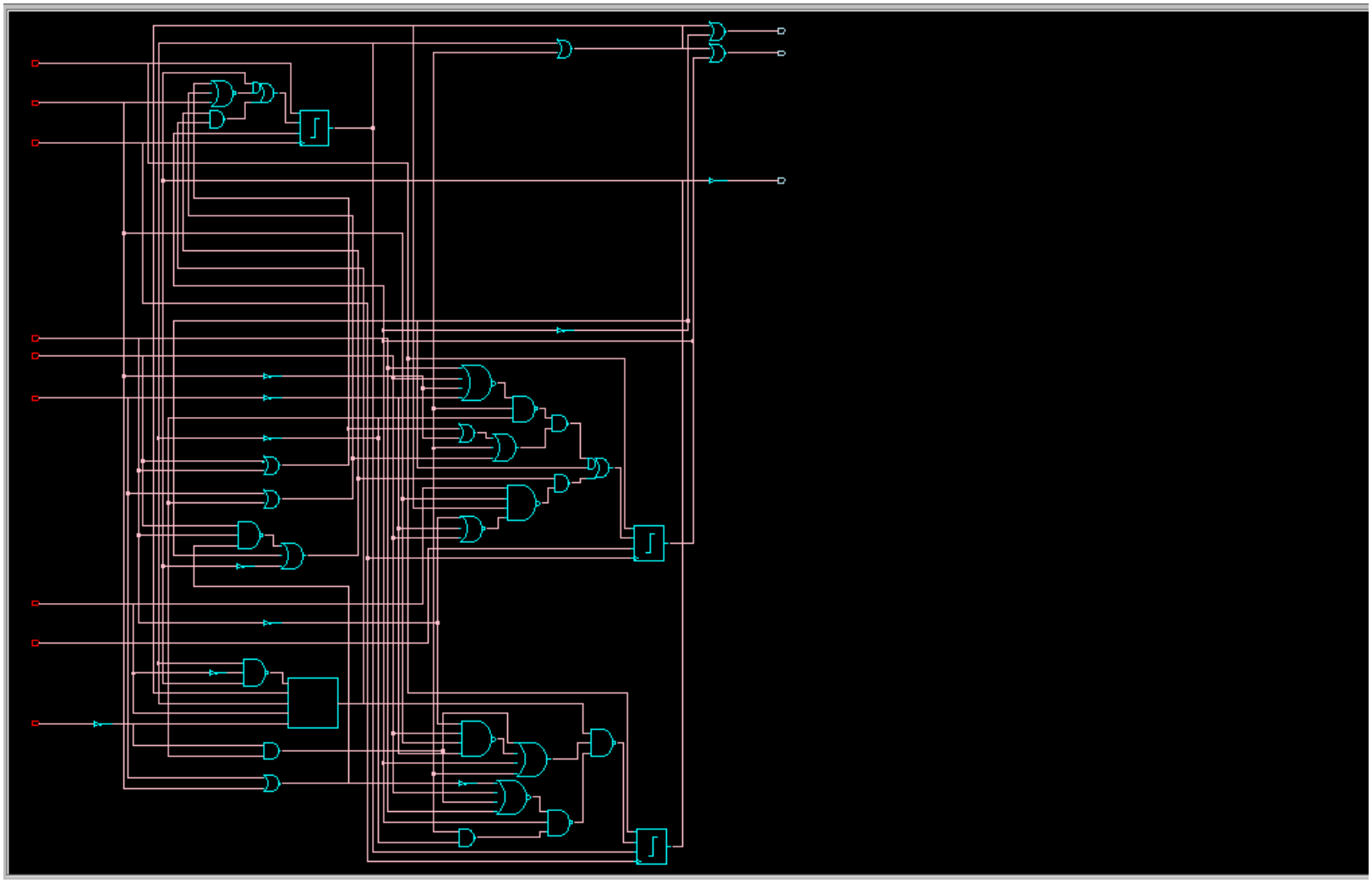
1. obtained circuit netlists (xsch)
2. table comparison of all state encodings
3. chosen state encoding, and reasons.
4. Delay Simulation
5. Scan-Path Simulation
6. Appendix (Make File + codej_b_l.vhd + code_j_b_l_scan.vhd + testbench.vhd)

1.Obtained Circuit Netlists (xsch)

Codej_b_l.vst:



Codej_b_l_scan.vst :



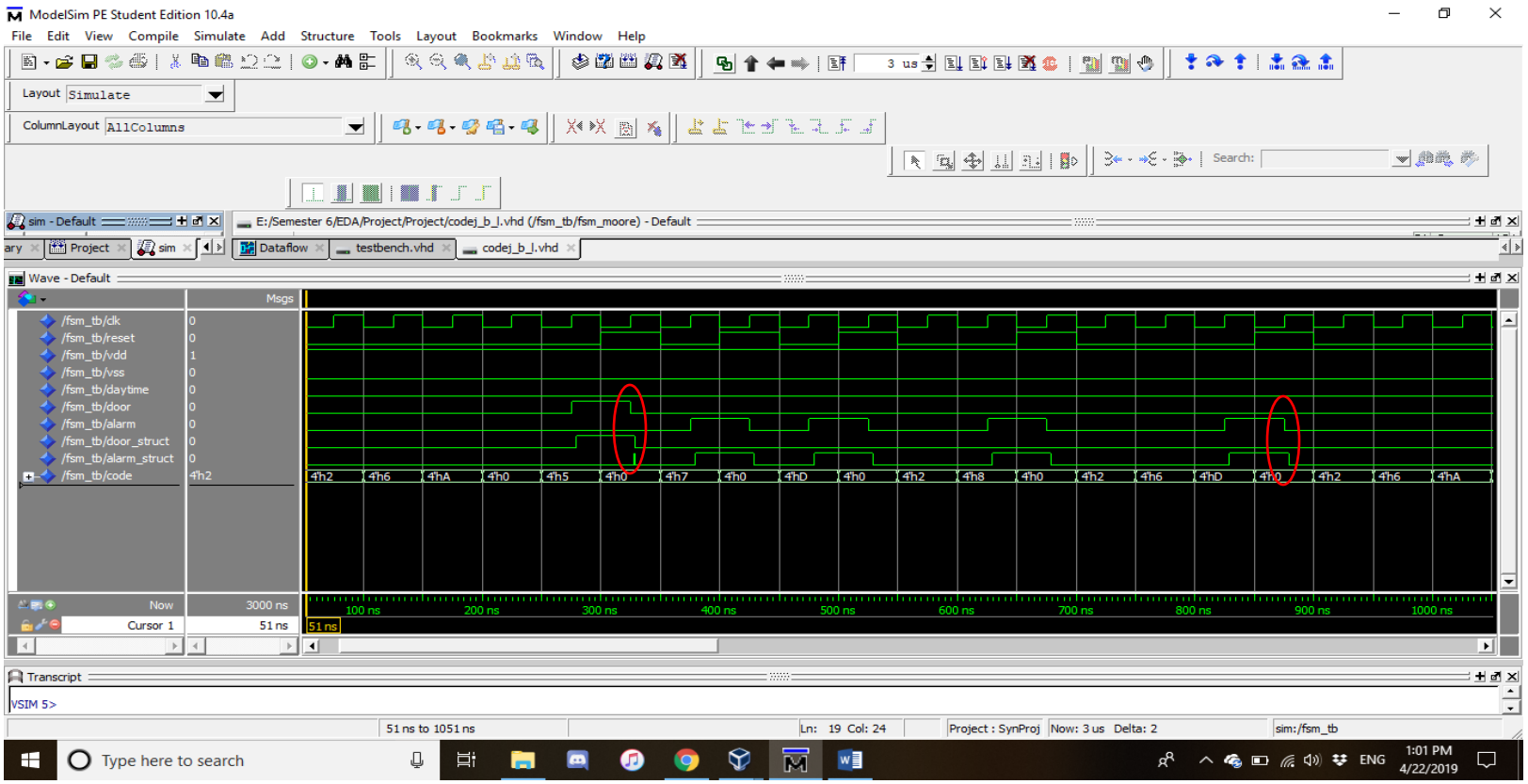
2.Table Comparison of All State Encodings

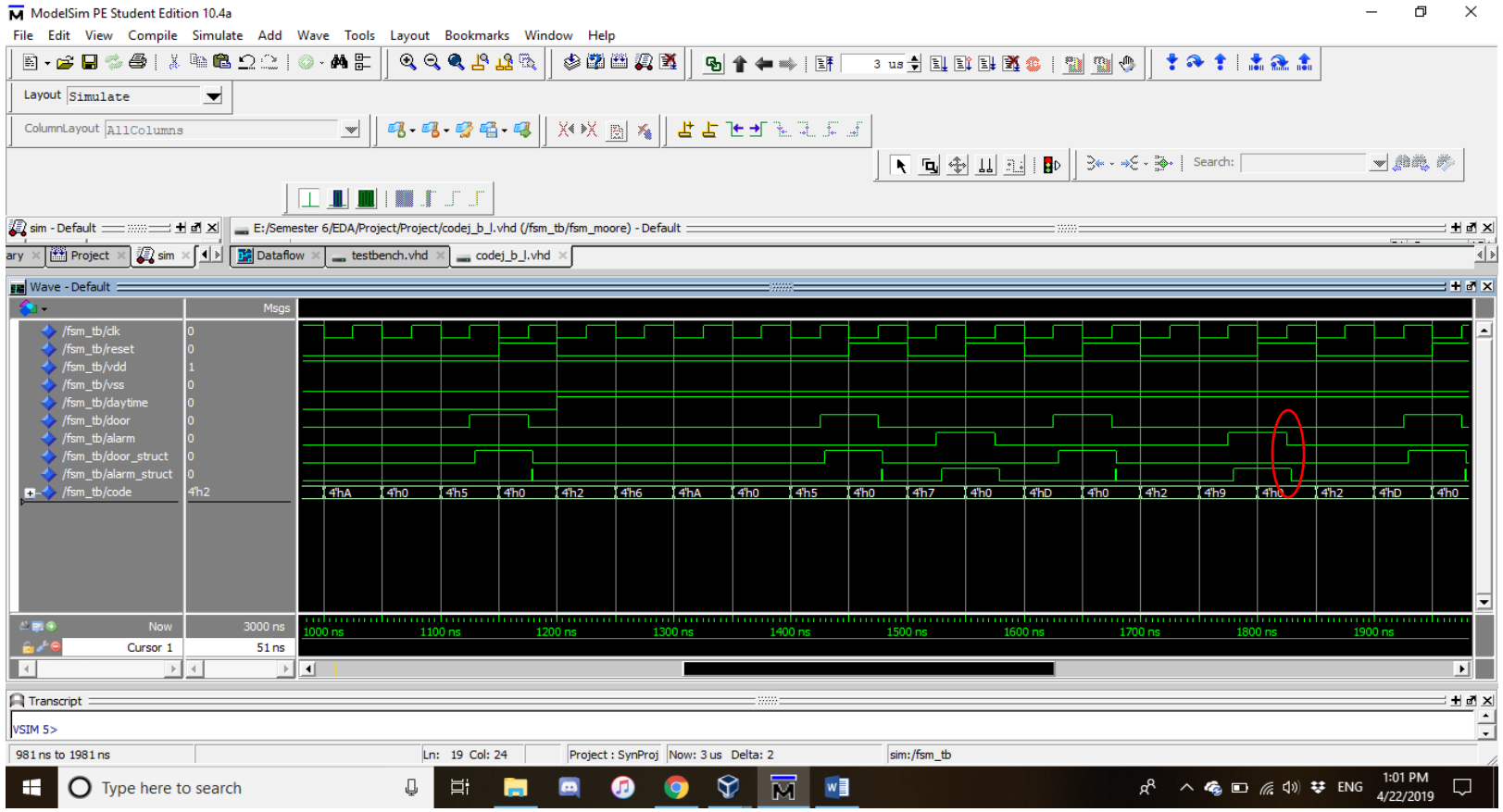
file	a	j	m	o	r
Initial Cost:					
Surface	217500	217500	218750	197500	221250
Depth	12	11	12	11	12
Literals	154	154	155	131	157
Bdd nodes	77	70	78	97	129
Final Cost:					
Surface	82250	71000	75500	74500	127500
Depth	8	6	7	7	9
Literals	67	68	66	72	107
Area (lamda^2)	61750	58750	60500	84000	105750
Delay (PS)	1998	2234	2476	2250	3324
Encoding States:					
sAlarm	0	0	0	0	5
sDoor	4	1	1	1	2
sChar4	3	4	4	2	4
sChar3	5	3	7	3	1
sChar2	1	5	5	4	7
sChar1	6	2	2	5	6
sBegin	2	6	6	6	3

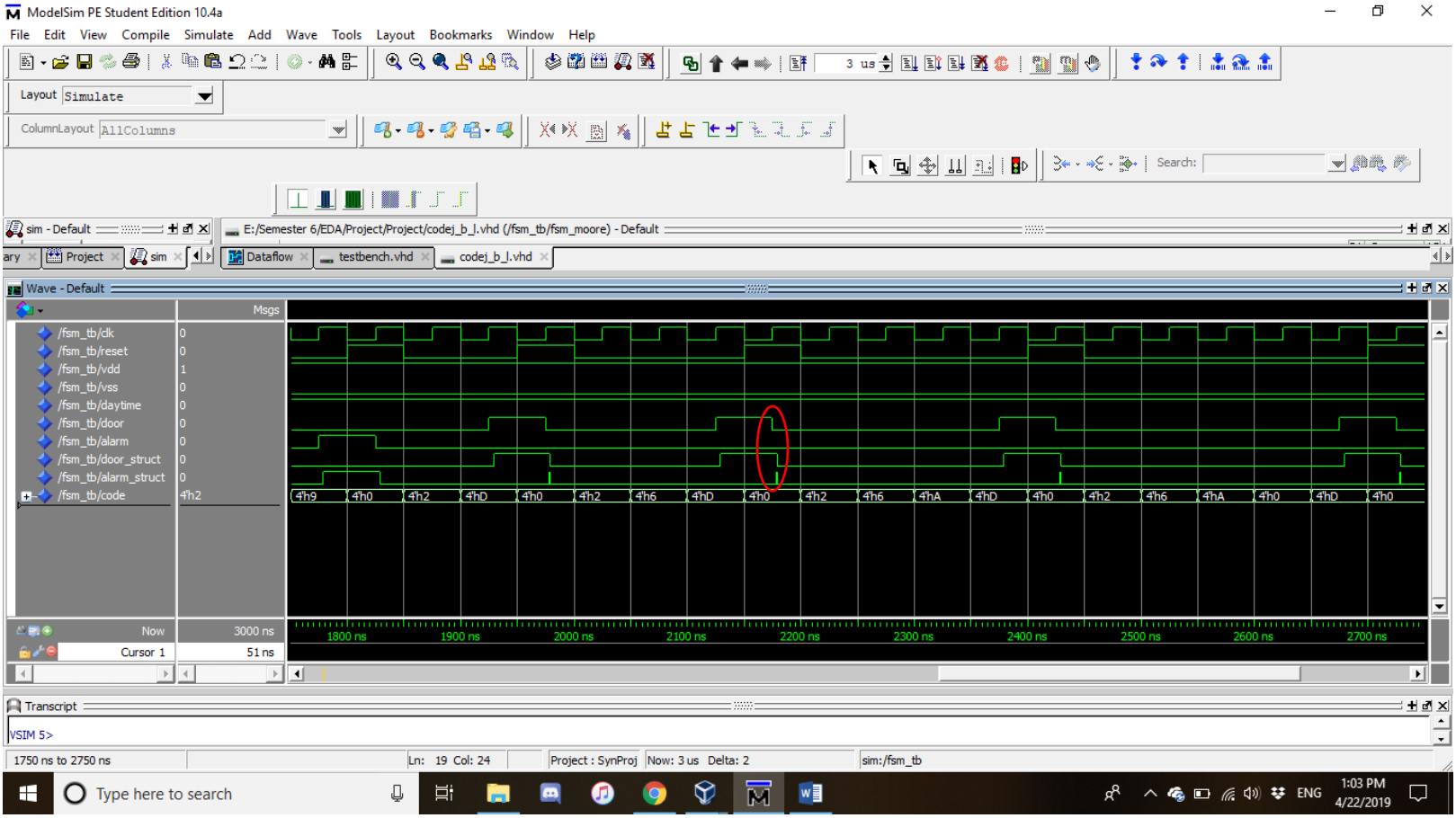
3.Chosen State Encoding

I chose file (j) for State Encoding , as it has the Minimum Area so it will have the least power consumbtion as the area is directly proportional to power consumbtion.and it also has the second least delay , so I guess it will be the best implementation.

4. Delay Simulation

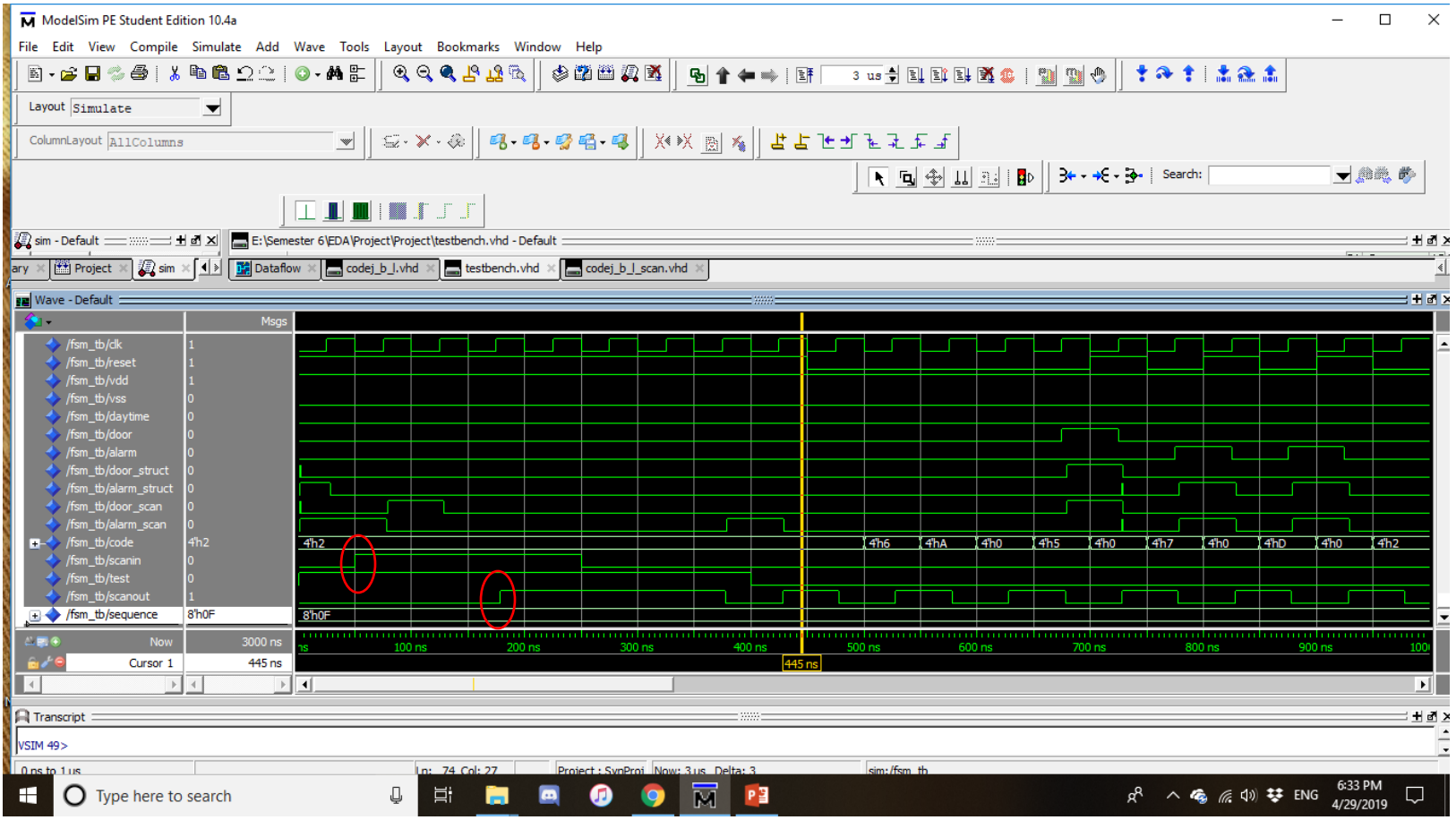




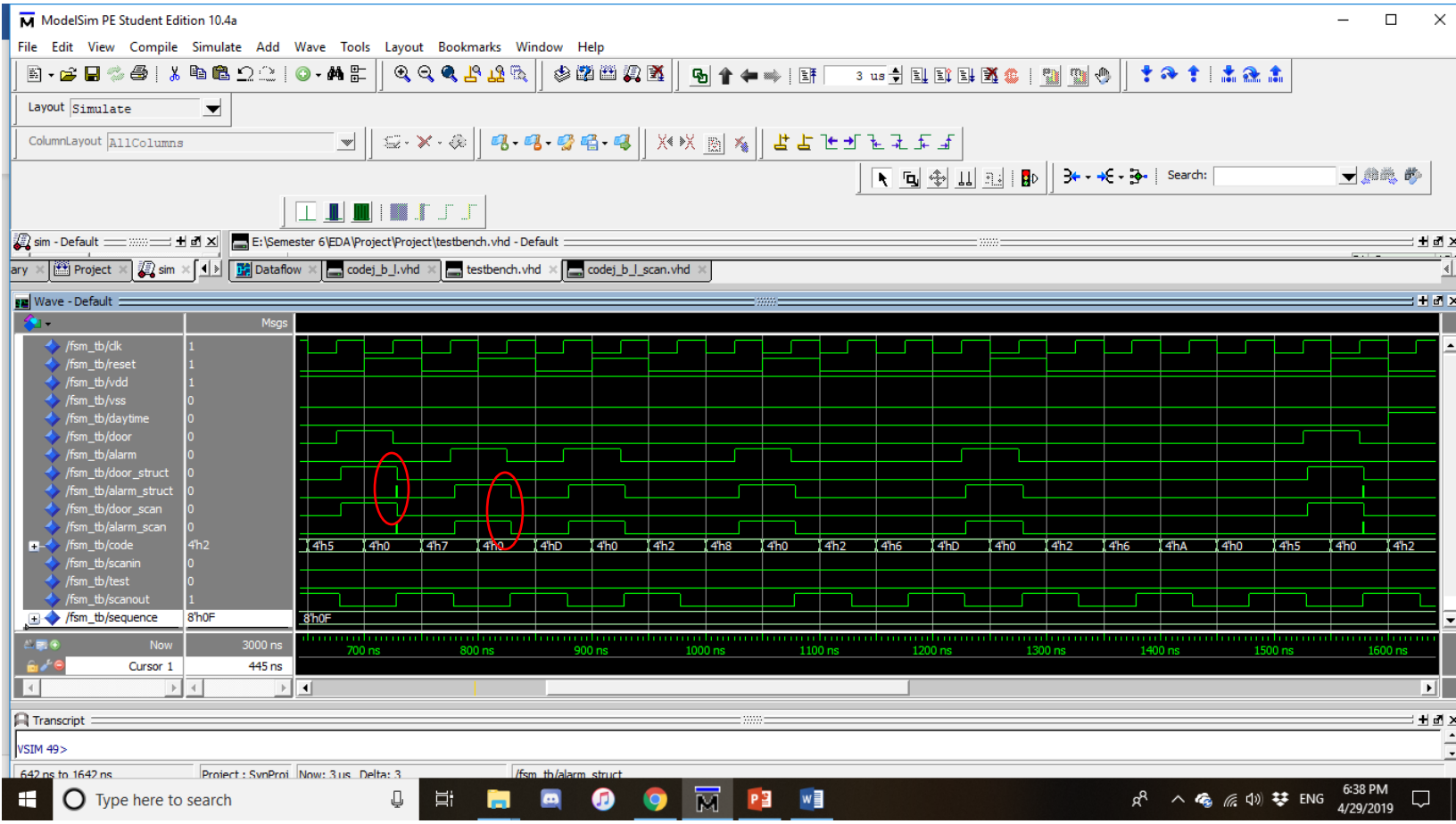


Comments: door and door_struct have the same Values , and also alarm and alarm_struct have the same Value, but door_struct and alarm_struct have a very small delay after door and alarm , as shown in the red ovals in the screenshots. And this delay is due to the delay of the sxlib components used in the structural Behavior.

5. Scan-Path Simulation



Comments: Scanout followed Scanin after 3 clock cycles as My Design Contains 3 Shift Registers.



Comments: when test = 0 , door_struct and door_scan are identical , also alarm_struct and alarm_scan are identical.

6.Appendix

1.MakeFile:

```
#-------Sdet-----------------------------------------#

all: codea.vbe \
    codej.vbe \
    codem.vbe \
    codeo.vbe \
    coder.vbe \
    code_boom \
    code_loon
                @echo "<-- Generated"

#-------Finite State Machine Synthesis-----------------#

vhd_to_fsm:
            rename .vhd .fsm *.vhd

codea.vbe: code.fsm
            @echo "   Encoding Synthesis -> codea.vbe"
            syf -CEV -a code

codej.vbe: code.fsm
            @echo "   Encoding Synthesis  -> codej.vbe"
            syf -CEV -j code

codem.vbe: code.fsm
            @echo "   Encoding Synthesis  -> codem.vbe"
            syf -CEV -m code

codeo.vbe: code.fsm
            @echo "   Encoding Synthesis  -> codeo.vbe"
```

```
	syf -CEV -o code

coder.vbe: code.fsm
	@echo "    Encoding Synthesis  -> coder.vbe"
	syf -CEV -r code

code_boom: codea_b.vbe codej_b.vbe codem_b.vbe \
	  codeo_b.vbe coder_b.vbe

%_b.vbe: %.vbe
	 @echo "   Boolean Optimiztaion -> $@  "
	 boom -V -d 50 $* $*_b > $*_boom.out

code_boog: codea_b.vst codej_b.vst codem_b.vst \
	  codeo_b.vst coder_b.vst

code_loon: codea_b_l.vst codej_b_l.vst codem_b_l.vst \
	  codeo_b_l.vst coder_b_l.vst

%.vst : %.vbe paramfile.lax
	@echo "  Logical Synsthesis -> $@ "
	boog -x 1 -l paramfile $* > $*_boog.out

%_l.vst : %.vst paramfile.lax
	@echo "  Netlist OPtimization -> $@ "
	loon -x 1 -l paramfile $* $*_l > $*_loon.out

%_b_l_net.vbe : %_b_l.vst %.vbe
	@echo " Formal checking -> $@ "
	flatbeh $*_b_l $*_b_l_net > $*_flatbeh.out
	proof -d $* $*_b_l_net > $*_proof.out

ac_scapin_registers:
	cat codej_b_l.vst | grep sff

%_scan.vst : %.vst scan.path
	@echo " scan-path insertion -> $@ "
```

```
		scapin -VRB $* scan $*_scan > scapin.out

#-------Clean Up-------------------------------------#

clean :
		rm -f  *.vbe *.enc *~
		@echo "Erase all the files generated by the
makefile"
```

2. code_j_b_l.vhd

```
LIBRARY sxlib_ModelSim;

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


architecture structural of codej_b_l is
Component a2_x2
  port (
    i0  : in     bit;
    i1  : in     bit;
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
 );
end component;
```

```
Component oa2ao222_x2
  port (
    i1  : in     bit;
    i0  : in     bit;
    i2  : in     bit;
    i3  : in     bit;
    i4  : in     bit;
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
 );
end component;

Component on12_x1
  port (
    i0  : in     bit;
    i1  : in     bit;
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
 );
end component;

Component inv_x4
```

```
  port (
    i   : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component o2_x2
  port (
    i0  : in     bit;
    i1  : in     bit;
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component o3_x2
  port (
    i0  : in     bit;
    i2  : in     bit;
    i1  : in     bit;
    q   : out    bit;
    vdd : in     bit;
```

```
    vss : in      bit
);
end component;

Component no3_x1
  port (
    i0  : in      bit;
    i1  : in      bit;
    i2  : in      bit;
    nq  : out     bit;
    vdd : in      bit;
    vss : in      bit
);
end component;

Component oa22_x2
  port (
    i0  : in      bit;
    i1  : in      bit;
    i2  : in      bit;
    q   : out     bit;
    vdd : in      bit;
    vss : in      bit
);
end component;
```

```
Component na2_x1
  port (
    i0  : in     bit;
    i1  : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component inv_x2
  port (
    i   : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component no4_x1
  port (
    i0  : in     bit;
    i1  : in     bit;
    i2  : in     bit;
```

```
    i3  : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
 );
end component;

Component na4_x1
  port (
    i0  : in     bit;
    i1  : in     bit;
    i2  : in     bit;
    i3  : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
 );
end component;

Component o4_x2
  port (
    i0  : in     bit;
    i1  : in     bit;
    i2  : in     bit;
    i3  : in     bit;
```

```
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component na3_x1
  port (
    i0  : in     bit;
    i1  : in     bit;
    i2  : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component sff1_x4
  port (
    ck  : in     bit;
    i   : in     bit;
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
);
```

```
end component;

Component no2_x1
  port (
    i0  : in     bit;
    i1  : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
 );
end component;

signal fsm_current_state     : bit_vector( 2 downto 0);
signal not_code              : bit_vector( 3 downto 0);
signal not_fsm_current_state : bit_vector( 1 downto 0);
signal oa22_x2_sig           : bit;
signal oa22_x2_2_sig         : bit;
signal o4_x2_sig             : bit;
signal o3_x2_sig             : bit;
signal o2_x2_sig             : bit;
signal not_reset             : bit;
signal not_aux6              : bit;
signal not_aux14             : bit;
signal not_aux13             : bit;
signal not_aux12             : bit;
```

```
signal not_aux1          : bit;
signal not_aux0          : bit;
signal no4_x1_sig        : bit;
signal no4_x1_2_sig      : bit;
signal no3_x1_sig        : bit;
signal no3_x1_2_sig      : bit;
signal na4_x1_sig        : bit;
signal na4_x1_2_sig      : bit;
signal na3_x1_sig        : bit;
signal na3_x1_5_sig      : bit;
signal na3_x1_4_sig      : bit;
signal na3_x1_3_sig      : bit;
signal na3_x1_2_sig      : bit;
signal na2_x1_sig        : bit;
signal na2_x1_4_sig      : bit;
signal na2_x1_3_sig      : bit;
signal na2_x1_2_sig      : bit;
signal inv_x2_sig        : bit;
signal inv_x2_3_sig      : bit;
signal inv_x2_2_sig      : bit;
signal aux2              : bit;

begin

not_aux13_ins : a2_x2
```

```
  port map (
    i0  => not_reset,
    i1  => not_fsm_current_state(1),
    q   => not_aux13,
    vdd => vdd,
    vss => vss
  );

inv_x2_ins : inv_x2
  port map (
    i   => daytime,
    nq  => inv_x2_sig,
    vdd => vdd,
    vss => vss
  );

na3_x1_ins : na3_x1
  port map (
    i0  => fsm_current_state(1),
    i1  => inv_x2_sig,
    i2  => fsm_current_state(2),
    nq  => na3_x1_sig,
    vdd => vdd,
    vss => vss
  );
```

```
not_aux12_ins : oa2ao222_x2
  port map (
    i1  => na3_x1_sig,
    i0  => not_aux0,
    i2  => fsm_current_state(1),
    i3  => daytime,
    i4  => not_reset,
    q   => not_aux12,
    vdd => vdd,
    vss => vss
  );

not_aux14_ins : on12_x1
  port map (
    i0  => code(1),
    i1  => code(3),
    q   => not_aux14,
    vdd => vdd,
    vss => vss
  );

not_aux1_ins : o2_x2
  port map (
    i0  => code(0),
```

```
      i1  => not_fsm_current_state(1),
      q   => not_aux1,
      vdd => vdd,
      vss => vss
    );

inv_x2_2_ins : inv_x2
    port map (
      i   => fsm_current_state(2),
      nq  => inv_x2_2_sig,
      vdd => vdd,
      vss => vss
    );

na3_x1_2_ins : na3_x1
    port map (
      i0  => code(1),
      i1  => code(3),
      i2  => aux2,
      nq  => na3_x1_2_sig,
      vdd => vdd,
      vss => vss
    );

not_aux6_ins : o3_x2
```

```
  port map (
    i0  => na3_x1_2_sig,
    i1  => not_fsm_current_state(0),
    i2  => inv_x2_2_sig,
    q   => not_aux6,
    vdd => vdd,
    vss => vss
  );

not_fsm_current_state_0_ins : inv_x2
  port map (
    i   => fsm_current_state(0),
    nq  => not_fsm_current_state(0),
    vdd => vdd,
    vss => vss
  );

not_aux0_ins : o2_x2
  port map (
    i0  => fsm_current_state(1),
    i1  => fsm_current_state(2),
    q   => not_aux0,
    vdd => vdd,
    vss => vss
  );
```

```
not_fsm_current_state_1_ins : inv_x4
  port map (
    i   => fsm_current_state(1),
    nq  => not_fsm_current_state(1),
    vdd => vdd,
    vss => vss
  );

not_reset_ins : inv_x2
  port map (
    i   => reset,
    nq  => not_reset,
    vdd => vdd,
    vss => vss
  );

not_code_3_ins : inv_x2
  port map (
    i   => code(3),
    nq  => not_code(3),
    vdd => vdd,
    vss => vss
  );
```

```
not_code_2_ins : inv_x2
  port map (
    i   => code(2),
    nq  => not_code(2),
    vdd => vdd,
    vss => vss
  );

not_code_0_ins : inv_x2
  port map (
    i   => code(0),
    nq  => not_code(0),
    vdd => vdd,
    vss => vss
  );

aux2_ins : no2_x1
  port map (
    i0  => code(0),
    i1  => code(2),
    nq  => aux2,
    vdd => vdd,
    vss => vss
  );
```

```
no3_x1_ins : no3_x1
  port map (
    i0  => not_code(3),
    i1  => not_code(0),
    i2  => code(1),
    nq  => no3_x1_sig,
    vdd => vdd,
    vss => vss
  );

na4_x1_ins : na4_x1
  port map (
    i0  => daytime,
    i1  => code(2),
    i2  => not_aux0,
    i3  => no3_x1_sig,
    nq  => na4_x1_sig,
    vdd => vdd,
    vss => vss
  );

na2_x1_ins : na2_x1
  port map (
    i0  => not_aux6,
    i1  => na4_x1_sig,
```

```
    nq  => na2_x1_sig,
    vdd => vdd,
    vss => vss
  );

o2_x2_ins : o2_x2
  port map (
    i0  => not_aux14,
    i1  => not_code(2),
    q   => o2_x2_sig,
    vdd => vdd,
    vss => vss
  );

o3_x2_ins : o3_x2
  port map (
    i0  => o2_x2_sig,
    i2  => not_aux1,
    i1  => fsm_current_state(2),
    q   => o3_x2_sig,
    vdd => vdd,
    vss => vss
  );

no4_x1_ins : no4_x1
```

```
  port map (
    i0  => code(3),
    i1  => code(1),
    i2  => not_code(2),
    i3  => not_code(0),
    nq  => no4_x1_sig,
    vdd => vdd,
    vss => vss
  );

na3_x1_3_ins : na3_x1
  port map (
    i0  => no4_x1_sig,
    i1  => fsm_current_state(2),
    i2  => not_fsm_current_state(1),
    nq  => na3_x1_3_sig,
    vdd => vdd,
    vss => vss
  );

na2_x1_2_ins : na2_x1
  port map (
    i0  => na3_x1_3_sig,
    i1  => o3_x2_sig,
    nq  => na2_x1_2_sig,
```

```
    vdd => vdd,
    vss => vss
  );

oa22_x2_ins : oa22_x2
  port map (
    i0  => na2_x1_2_sig,
    i1  => not_fsm_current_state(0),
    i2  => na2_x1_sig,
    q   => oa22_x2_sig,
    vdd => vdd,
    vss => vss
  );

fsm_current_state_0_ins : sff1_x4
  port map (
    ck  => clk,
    i   => oa22_x2_sig,
    q   => fsm_current_state(0),
    vdd => vdd,
    vss => vss
  );

na2_x1_3_ins : na2_x1
  port map (
```

```
      i0  => not_aux6,
      i1  => not_aux12,
      nq  => na2_x1_3_sig,
      vdd => vdd,
      vss => vss
    );

no3_x1_2_ins : no3_x1
    port map (
      i0  => not_aux14,
      i1  => not_aux1,
      i2  => code(2),
      nq  => no3_x1_2_sig,
      vdd => vdd,
      vss => vss
    );

oa22_x2_2_ins : oa22_x2
    port map (
      i0  => fsm_current_state(2),
      i1  => no3_x1_2_sig,
      i2  => na2_x1_3_sig,
      q   => oa22_x2_2_sig,
      vdd => vdd,
      vss => vss
```

```
  );

fsm_current_state_1_ins : sff1_x4
  port map (
    ck  => clk,
    i   => oa22_x2_2_sig,
    q   => fsm_current_state(1),
    vdd => vdd,
    vss => vss
  );

na2_x1_4_ins : na2_x1
  port map (
    i0  => fsm_current_state(2),
    i1  => not_fsm_current_state(1),
    nq  => na2_x1_4_sig,
    vdd => vdd,
    vss => vss
  );

inv_x2_3_ins : inv_x2
  port map (
    i   => aux2,
    nq  => inv_x2_3_sig,
    vdd => vdd,
```

```
    vss => vss
  );

no4_x1_2_ins : no4_x1
  port map (
    i0  => inv_x2_3_sig,
    i1  => code(1),
    i2  => not_aux13,
    i3  => code(3),
    nq  => no4_x1_2_sig,
    vdd => vdd,
    vss => vss
  );

na3_x1_5_ins : na3_x1
  port map (
    i0  => no4_x1_2_sig,
    i1  => fsm_current_state(0),
    i2  => na2_x1_4_sig,
    nq  => na3_x1_5_sig,
    vdd => vdd,
    vss => vss
  );

na4_x1_2_ins : na4_x1
```

```
  port map (
    i0  => not_code(3),
    i1  => code(1),
    i2  => code(2),
    i3  => not_code(0),
    nq  => na4_x1_2_sig,
    vdd => vdd,
    vss => vss
  );

o4_x2_ins : o4_x2
  port map (
    i0  => not_aux13,
    i1  => na4_x1_2_sig,
    i2  => fsm_current_state(0),
    i3  => fsm_current_state(2),
    q   => o4_x2_sig,
    vdd => vdd,
    vss => vss
  );

na3_x1_4_ins : na3_x1
  port map (
    i0  => not_aux12,
    i1  => o4_x2_sig,
```

```
      i2  => na3_x1_5_sig,
      nq  => na3_x1_4_sig,
      vdd => vdd,
      vss => vss
    );

fsm_current_state_2_ins : sff1_x4
    port map (
      ck  => clk,
      i   => na3_x1_4_sig,
      q   => fsm_current_state(2),
      vdd => vdd,
      vss => vss
    );

alarm_ins : no2_x1
    port map (
      i0  => not_aux0,
      i1  => fsm_current_state(0),
      nq  => alarm,
      vdd => vdd,
      vss => vss
    );

door_ins : no2_x1
```

```
  port map (
    i0  => not_aux0,
    i1  => not_fsm_current_state(0),
    nq  => door,
    vdd => vdd,
    vss => vss
  );



end structural;

architecture moore of codej_b_l is
   type state_type is(sBegin,sChar1,sChar2,sChar3,sChar4,sDoor,sAlarm);
   signal current_state: state_type;
   signal next_state: state_type;
-- Synthesis directives :
-- pragma current_state current_state
-- pragma next_state next_state
-- pragma clock clk
   begin

      process(clk)
      begin
         if clk='1' and clk'event then
            current_state <=next_state;
```

```
        end if;
        end process;



      process(current_state,code,reset)
      begin
        -- if reset ='1' then
       --      next_state <= sBegin;
        -- else
         --Night Conditions
if daytime ='0' then
        case current_state is

           when sBegin =>
              door <= '0';
              alarm <= '0';
                                  if reset ='1' then
                    next_state <= sBegin;
                                 else
           if code= "0010" then
              next_state <= sChar1;
           else
               next_state <= sAlarm;
             end if;
end if;
```

```
when sChar1 =>
     door<= '0';
    alarm <= '0';
if code= "0110" then
    next_state <= sChar2;
else
    next_state <= sAlarm;
   end if;

   when sChar2 =>
   door<= '0';
   alarm <= '0';
   if code= "1010" then
       next_state <= sChar3;
   else
       next_state <= sAlarm;
      end if;

      when sChar3 =>
      door<= '0';
      alarm <= '0';
      if code= "0000" then
          next_state <= sChar4;
      else
```

```
                    next_state <= sAlarm;
                    end if;

                    when sChar4 =>
                    door<= '0';
                    alarm <= '0';
                    if code= "0101" then
                       next_state <= sDoor;
                    else
                       next_state <= sAlarm;
                     end if;

                     when sDoor =>
                     door<= '1';
                     alarm <= '0';
                     next_state <= sBegin;

                     when sAlarm =>
                     door<= '0';
                     alarm <= '1';
                     next_state <= sBegin;

        end case;
else
                        case current_state is
```

```
when sBegin =>
    door<= '0';
    alarm <= '0';
if code= "0010" then
    next_state <= sChar1;
                        elsif code= "1101" then
                            next_state <= sDoor;
else
     next_state <= sAlarm;
   end if;

   when sChar1 =>
        door<= '0';
       alarm <= '0';
   if code= "0110" then
       next_state <= sChar2;
 elsif code= "1101" then
                                next_state <= sDoor;
   else
       next_state <= sAlarm;
      end if;

      when sChar2 =>
      door<= '0';
```

```
alarm <= '0';
if code= "1010" then
    next_state <= sChar3;
  elsif code= "1101" then
                                    next_state <= sDoor;
else
    next_state <= sAlarm;
   end if;

   when sChar3 =>
   door<= '0';
   alarm <= '0';
   if code= "0000" then
       next_state <= sChar4;
  elsif code= "1101" then
                                    next_state <= sDoor;
   else
       next_state <= sAlarm;
      end if;

      when sChar4 =>
      door<= '0';
      alarm <= '0';
      if code= "0101" then
          next_state <= sDoor;
```

```
                elsif code= "1101" then
                                        next_state <= sDoor;
                else
                   next_state <= sAlarm;
                  end if;

                  when sDoor =>
                  door<= '1';
                  alarm <= '0';
                                    if reset ='1' then
                  next_state <= sBegin;
                  end if;
                  when sAlarm =>
                  door<= '0';
                  alarm <= '1';
                                    if reset ='1' then
                  next_state <= sBegin;
                                        end if;

        end case;
end if;
        end process;


   end moore;
```

3. codej_b_l_scan.vhd

```
LIBRARY sxlib_ModelSim;

entity codej_b_l_scan is
  port (
    clk     : in      bit;
    vdd     : in      bit;
    vss     : in      bit;
    daytime : in      bit;
    reset   : in      bit;
    code    : in      bit_vector(3 downto 0);
    door    : out     bit;
    alarm   : out     bit;
    scanin  : in      bit;
    test    : in      bit;
    scanout : out     bit
);
end codej_b_l_scan;


architecture structural of codej_b_l_scan is
Component a2_x2
  port (
    i0  : in      bit;
    i1  : in      bit;
    q   : out     bit;
    vdd : in      bit;
```

```
    vss : in      bit
);
end component;

Component oa2ao222_x2
  port (
    i1  : in      bit;
    i0  : in      bit;
    i2  : in      bit;
    i3  : in      bit;
    i4  : in      bit;
    q   : out     bit;
    vdd : in      bit;
    vss : in      bit
);
end component;

Component on12_x1
  port (
    i0  : in      bit;
    i1  : in      bit;
    q   : out     bit;
    vdd : in      bit;
    vss : in      bit
);
```

```
end component;

Component inv_x4
  port (
    i   : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component o2_x2
  port (
    i0  : in     bit;
    i1  : in     bit;
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component o3_x2
  port (
    i0  : in     bit;
    i2  : in     bit;
```

```
    i1  : in     bit;
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component no3_x1
  port (
    i0  : in     bit;
    i1  : in     bit;
    i2  : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component oa22_x2
  port (
    i0  : in     bit;
    i1  : in     bit;
    i2  : in     bit;
    q   : out    bit;
    vdd : in     bit;
```

```
    vss : in      bit
);
end component;

Component na2_x1
  port (
    i0  : in      bit;
    i1  : in      bit;
    nq  : out     bit;
    vdd : in      bit;
    vss : in      bit
);
end component;

Component inv_x2
  port (
    i   : in      bit;
    nq  : out     bit;
    vdd : in      bit;
    vss : in      bit
);
end component;

Component no4_x1
  port (
```

```
    i0  : in     bit;
    i1  : in     bit;
    i2  : in     bit;
    i3  : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
 );
end component;

Component na4_x1
  port (
    i0  : in     bit;
    i1  : in     bit;
    i2  : in     bit;
    i3  : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
 );
end component;

Component o4_x2
  port (
    i0  : in     bit;
```

```
    i1  : in     bit;
    i2  : in     bit;
    i3  : in     bit;
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
 );
end component;

Component na3_x1
  port (
    i0  : in     bit;
    i1  : in     bit;
    i2  : in     bit;
    nq  : out    bit;
    vdd : in     bit;
    vss : in     bit
 );
end component;

Component no2_x1
  port (
    i0  : in     bit;
    i1  : in     bit;
    nq  : out    bit;
```

```
    vdd : in     bit;
    vss : in     bit
);
end component;

Component sff2_x4
  port (
    ck  : in     bit;
    cmd : in     bit;
    i0  : in     bit;
    i1  : in     bit;
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;

Component buf_x2
  port (
    i   : in     bit;
    q   : out    bit;
    vdd : in     bit;
    vss : in     bit
);
end component;
```

```
signal fsm_current_state     : bit_vector( 2 downto 0);
signal not_code              : bit_vector( 3 downto 0);
signal not_fsm_current_state : bit_vector( 1 downto 0);
signal oa22_x2_sig           : bit;
signal oa22_x2_2_sig         : bit;
signal o4_x2_sig             : bit;
signal o3_x2_sig             : bit;
signal o2_x2_sig             : bit;
signal not_reset             : bit;
signal not_aux6              : bit;
signal not_aux14             : bit;
signal not_aux13             : bit;
signal not_aux12             : bit;
signal not_aux1              : bit;
signal not_aux0              : bit;
signal no4_x1_sig            : bit;
signal no4_x1_2_sig          : bit;
signal no3_x1_sig            : bit;
signal no3_x1_2_sig          : bit;
signal na4_x1_sig            : bit;
signal na4_x1_2_sig          : bit;
signal na3_x1_sig            : bit;
signal na3_x1_5_sig          : bit;
signal na3_x1_4_sig          : bit;
```

```
signal na3_x1_3_sig         : bit;
signal na3_x1_2_sig         : bit;
signal na2_x1_sig           : bit;
signal na2_x1_4_sig         : bit;
signal na2_x1_3_sig         : bit;
signal na2_x1_2_sig         : bit;
signal inv_x2_sig           : bit;
signal inv_x2_3_sig         : bit;
signal inv_x2_2_sig         : bit;
signal aux2                 : bit;

begin

not_aux13_ins : a2_x2
  port map (
    i0  => not_reset,
    i1  => not_fsm_current_state(1),
    q   => not_aux13,
    vdd => vdd,
    vss => vss
  );

inv_x2_ins : inv_x2
  port map (
    i   => daytime,
```

```
    nq  => inv_x2_sig,
    vdd => vdd,
    vss => vss
  );

na3_x1_ins : na3_x1
  port map (
    i0  => fsm_current_state(1),
    i1  => inv_x2_sig,
    i2  => fsm_current_state(2),
    nq  => na3_x1_sig,
    vdd => vdd,
    vss => vss
  );

not_aux12_ins : oa2ao222_x2
  port map (
    i1  => na3_x1_sig,
    i0  => not_aux0,
    i2  => fsm_current_state(1),
    i3  => daytime,
    i4  => not_reset,
    q   => not_aux12,
    vdd => vdd,
    vss => vss
```

```
  );

not_aux14_ins : on12_x1
  port map (
    i0  => code(1),
    i1  => code(3),
    q   => not_aux14,
    vdd => vdd,
    vss => vss
  );

not_aux1_ins : o2_x2
  port map (
    i0  => code(0),
    i1  => not_fsm_current_state(1),
    q   => not_aux1,
    vdd => vdd,
    vss => vss
  );

inv_x2_2_ins : inv_x2
  port map (
    i   => fsm_current_state(2),
    nq  => inv_x2_2_sig,
    vdd => vdd,
```

```
    vss => vss
  );

na3_x1_2_ins : na3_x1
  port map (
    i0  => code(1),
    i1  => code(3),
    i2  => aux2,
    nq  => na3_x1_2_sig,
    vdd => vdd,
    vss => vss
  );

not_aux6_ins : o3_x2
  port map (
    i0  => na3_x1_2_sig,
    i2  => inv_x2_2_sig,
    i1  => not_fsm_current_state(0),
    q   => not_aux6,
    vdd => vdd,
    vss => vss
  );

not_fsm_current_state_0_ins : inv_x2
  port map (
```

```
      i   => fsm_current_state(0),
      nq  => not_fsm_current_state(0),
      vdd => vdd,
      vss => vss
    );

not_aux0_ins : o2_x2
    port map (
      i0  => fsm_current_state(1),
      i1  => fsm_current_state(2),
      q   => not_aux0,
      vdd => vdd,
      vss => vss
    );

not_fsm_current_state_1_ins : inv_x4
    port map (
      i   => fsm_current_state(1),
      nq  => not_fsm_current_state(1),
      vdd => vdd,
      vss => vss
    );

not_reset_ins : inv_x2
    port map (
```

```
      i   => reset,
      nq  => not_reset,
      vdd => vdd,
      vss => vss
    );

not_code_3_ins : inv_x2
    port map (
      i   => code(3),
      nq  => not_code(3),
      vdd => vdd,
      vss => vss
    );

not_code_2_ins : inv_x2
    port map (
      i   => code(2),
      nq  => not_code(2),
      vdd => vdd,
      vss => vss
    );

not_code_0_ins : inv_x2
    port map (
      i   => code(0),
```

```
      nq  => not_code(0),
      vdd => vdd,
      vss => vss
    );

aux2_ins : no2_x1
    port map (
      i0  => code(0),
      i1  => code(2),
      nq  => aux2,
      vdd => vdd,
      vss => vss
    );

no3_x1_ins : no3_x1
    port map (
      i0  => not_code(3),
      i1  => not_code(0),
      i2  => code(1),
      nq  => no3_x1_sig,
      vdd => vdd,
      vss => vss
    );

na4_x1_ins : na4_x1
```

```
  port map (
    i0  => daytime,
    i1  => code(2),
    i2  => not_aux0,
    i3  => no3_x1_sig,
    nq  => na4_x1_sig,
    vdd => vdd,
    vss => vss
  );

na2_x1_ins : na2_x1
  port map (
    i0  => not_aux6,
    i1  => na4_x1_sig,
    nq  => na2_x1_sig,
    vdd => vdd,
    vss => vss
  );

o2_x2_ins : o2_x2
  port map (
    i0  => not_aux14,
    i1  => not_code(2),
    q   => o2_x2_sig,
    vdd => vdd,
```

```
    vss => vss
  );

o3_x2_ins : o3_x2
  port map (
    i0  => o2_x2_sig,
    i2  => not_aux1,
    i1  => fsm_current_state(2),
    q   => o3_x2_sig,
    vdd => vdd,
    vss => vss
  );

no4_x1_ins : no4_x1
  port map (
    i0  => code(3),
    i1  => code(1),
    i2  => not_code(2),
    i3  => not_code(0),
    nq  => no4_x1_sig,
    vdd => vdd,
    vss => vss
  );

na3_x1_3_ins : na3_x1
```

```
  port map (
    i0  => no4_x1_sig,
    i1  => fsm_current_state(2),
    i2  => not_fsm_current_state(1),
    nq  => na3_x1_3_sig,
    vdd => vdd,
    vss => vss
  );

na2_x1_2_ins : na2_x1
  port map (
    i0  => na3_x1_3_sig,
    i1  => o3_x2_sig,
    nq  => na2_x1_2_sig,
    vdd => vdd,
    vss => vss
  );

oa22_x2_ins : oa22_x2
  port map (
    i0  => na2_x1_2_sig,
    i1  => not_fsm_current_state(0),
    i2  => na2_x1_sig,
    q   => oa22_x2_sig,
    vdd => vdd,
```

```
    vss => vss
  );

na2_x1_3_ins : na2_x1
  port map (
    i0  => not_aux6,
    i1  => not_aux12,
    nq  => na2_x1_3_sig,
    vdd => vdd,
    vss => vss
  );

no3_x1_2_ins : no3_x1
  port map (
    i0  => not_aux14,
    i1  => not_aux1,
    i2  => code(2),
    nq  => no3_x1_2_sig,
    vdd => vdd,
    vss => vss
  );

oa22_x2_2_ins : oa22_x2
  port map (
    i0  => fsm_current_state(2),
```

```
      i1  => no3_x1_2_sig,
      i2  => na2_x1_3_sig,
      q   => oa22_x2_2_sig,
      vdd => vdd,
      vss => vss
    );

na2_x1_4_ins : na2_x1
    port map (
      i0  => fsm_current_state(2),
      i1  => not_fsm_current_state(1),
      nq  => na2_x1_4_sig,
      vdd => vdd,
      vss => vss
    );

inv_x2_3_ins : inv_x2
    port map (
      i   => aux2,
      nq  => inv_x2_3_sig,
      vdd => vdd,
      vss => vss
    );

no4_x1_2_ins : no4_x1
```

```
  port map (
    i0  => inv_x2_3_sig,
    i1  => code(1),
    i2  => not_aux13,
    i3  => code(3),
    nq  => no4_x1_2_sig,
    vdd => vdd,
    vss => vss
  );

na3_x1_5_ins : na3_x1
  port map (
    i0  => no4_x1_2_sig,
    i1  => fsm_current_state(0),
    i2  => na2_x1_4_sig,
    nq  => na3_x1_5_sig,
    vdd => vdd,
    vss => vss
  );

na4_x1_2_ins : na4_x1
  port map (
    i0  => not_code(3),
    i1  => code(1),
    i2  => code(2),
```

```
      i3  => not_code(0),
      nq  => na4_x1_2_sig,
      vdd => vdd,
      vss => vss
   );

o4_x2_ins : o4_x2
   port map (
      i0  => not_aux13,
      i1  => na4_x1_2_sig,
      i2  => fsm_current_state(0),
      i3  => fsm_current_state(2),
      q   => o4_x2_sig,
      vdd => vdd,
      vss => vss
   );

na3_x1_4_ins : na3_x1
   port map (
      i0  => not_aux12,
      i1  => o4_x2_sig,
      i2  => na3_x1_5_sig,
      nq  => na3_x1_4_sig,
      vdd => vdd,
      vss => vss
```

```
  );

alarm_ins : no2_x1
  port map (
    i0  => not_aux0,
    i1  => fsm_current_state(0),
    nq  => alarm,
    vdd => vdd,
    vss => vss
  );

door_ins : no2_x1
  port map (
    i0  => not_aux0,
    i1  => not_fsm_current_state(0),
    nq  => door,
    vdd => vdd,
    vss => vss
  );

fsm_current_state_0_ins_scan_0 : sff2_x4
  port map (
    ck  => clk,
    cmd => test,
    i0  => oa22_x2_sig,
```

```
      i1  => scanin,
      q   => fsm_current_state(0),
      vdd => vdd,
      vss => vss
    );

fsm_current_state_1_ins_scan_1 : sff2_x4
    port map (
      ck  => clk,
      cmd => test,
      i0  => oa22_x2_2_sig,
      i1  => fsm_current_state(0),
      q   => fsm_current_state(1),
      vdd => vdd,
      vss => vss
    );

fsm_current_state_2_ins_scan_2 : sff2_x4
    port map (
      ck  => clk,
      cmd => test,
      i0  => na3_x1_4_sig,
      i1  => fsm_current_state(1),
      q   => fsm_current_state(2),
      vdd => vdd,
```

```
    vss => vss
  );

buf_scan_3 : buf_x2
  port map (
    i   => fsm_current_state(2),
    q   => scanout,
    vdd => vdd,
    vss => vss
  );


end structural;
```

4. testbench.vhd

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

entity fsm_tb is
   end entity fsm_tb;

   architecture tb_arch of fsm_tb is
      component codej_b_l is
         port (
     clk     : in      bit;
     vdd     : in      bit;
     vss     : in      bit;
     daytime : in      bit;
     reset   : in      bit;
     code    : in      bit_vector(3 downto 0);
     door    : out     bit;
     alarm   : out     bit
 );
 end component codej_b_l;
         component codej_b_l_scan is
  port (
     clk     : in      bit;
```

```
    vdd     : in      bit;
    vss     : in      bit;
    daytime : in      bit;
    reset   : in      bit;
    code    : in      bit_vector(3 downto 0);
    door    : out     bit;
    alarm   : out     bit;
    scanin  : in      bit;
    test    : in      bit;
    scanout : out     bit
 );
end component codej_b_l_scan;

        signal clk:bit :='0';
        signal reset:bit := '1';
        signal vdd:bit := '1';
        signal vss:bit := '0';
        signal daytime:bit := '0';
        signal door:bit := '0';
        signal alarm: bit := '0';
           signal door_struct:bit := '0';
        signal alarm_struct: bit := '0';
           signal door_scan:bit := '0';
        signal alarm_scan: bit := '0';
        signal code:bit_vector(3 downto 0) := "0010";
```

```
        signal scanin:bit := '0';
    signal test:bit := '0';
    signal scanout: bit := '0';
        signal sequence:bit_vector(7 downto 0) :="00001111";
    constant clk_period: time := 50 ns;
    for fsm_struct: codej_b_l use entity work.codej_b_l(structural);
                    for fsm_moore: codej_b_l use entity work.codej_b_l(moore);
                    for fsm_scan: codej_b_l_scan use entity
work.codej_b_l_scan(structural);
    begin
        process is
        begin
            clk <= '0';
            wait for clk_period/2;
            clk <= '1';
            wait for clk_period/2;
            end process;

        process is
        begin
reset <= '1';
test <= '1';
for i In 0 to sequence'length-1 loop
wait for clk_period; -- Leave time for the output to stabilize
if i>=3 then -- Assert condition
Assert scanout=sequence(i-3)
```

```
Report "scanout does not follow scan in"
Severity error;
end if;
scanin <= sequence(i); -- scanin changes on the next wait statement
end loop;
test<='0';
daytime<='0';
  --Night
        --code<="0010";
  wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
  reset<='0';
              code<="0010";
              wait for 50 ns;
              assert door='0' and alarm='0'
              report "there is error"
              severity warning;
```

```
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
                code<="0110";
                wait for 50 ns;
                assert door='0' and alarm='0'
                report "there is error"
                severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
                code<="1010";
                wait for 50 ns;
                assert door='0' and alarm='0'
                report "there is error"
                severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
```

```
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
                code<="0000";
                wait for 50 ns;
                assert door='0' and alarm='0'
                report "there is error"
                severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
                code<="0101";
                wait for 50 ns;
                assert door='1' and alarm='0'
                report "Door Must Open"
                severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
```

```
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
reset<='0';
                code<="0111";
                wait for 50 ns;
                assert door='0' and alarm='1'
                report "Alarm Must Ring"
                severity warning;

reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
```

```
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
reset<='0';
                code<="1101";
                wait for 50 ns;
                assert door='0' and alarm='1'
                report "Alarm Must Ring"
                severity warning;
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
reset<='0';
```

```
                code<="0010";
                wait for 50 ns;
                assert door='0' and alarm='0'
                report "there is error"
                severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
                code<="1000";
                wait for 50 ns;
                assert door='0' and alarm='1'
                report "Alarm Must Ring"
                severity warning;
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
reset<='0';
                code<="0010";
                wait for 50 ns;
```

```
                assert door='0' and alarm='0'
                report "there is error"
                severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
                code<="0110";
                wait for 50 ns;
                assert door='0' and alarm='0'
                report "there is error"
                severity warning;
                code<="1101";
                wait for 50 ns;
                assert door='0' and alarm='1'
                report "Alarm Must Ring"
                severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
```

```
assert door=door_struct and alarm=alarm_struct
   report "Behavioral and Structural Behavior are not the same"
   severity warning;
reset<='0';
                code<="0010";
                wait for 50 ns;
                assert door='0' and alarm='0'
                report "there is error"
                severity warning;
                code<="0110";
                wait for 50 ns;
                assert door='0' and alarm='0'
                report "there is error"
                severity warning;
                code<="1010";
                wait for 50 ns;
                assert door='0' and alarm='0'
                report "there is error"
                severity warning;
                code<="0000";
                wait for 50 ns;
                assert door='0' and alarm='0'
                report "there is error"
                severity warning;
                code<="0101";
```

```
                wait for 50 ns;
                assert door='1' and alarm='0'
                report "Door Must Open"
                severity warning;
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
--reset<='0';

  daytime<='1';
 --  reset<='1';
--code<=X"0";
--Morning
--wait for 50 ns;
--assert door='0' and alarm='0'
--report "Reset error"
--severity warning;
reset<='0';
            code<="0010";
```

```
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
            code<="0110";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
            code<="1010";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
```

```
            severity warning;
            code<="0000";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
            code<="0101";
            wait for 50 ns;
            assert door='1' and alarm='0'
            report "Door Must Open"
            severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;

reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
reset<='0';
```

```
            code<="0111";
            wait for 50 ns;
            assert door='0' and alarm='1'
            report "Alarm Must Ring"
            severity warning;

reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
reset<='0';
            code<="1101";
            wait for 50 ns;
            assert door='1' and alarm='0'
            report "Door Must Open"
            severity warning;
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
```

```
reset<='0';
            code<="0010";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
            code<="1001";
            wait for 50 ns;
            assert door='0' and alarm='1'
            report "Alarm Must Ring"
            severity warning;
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
reset<='0';
code<="0010";
wait for 50 ns;
assert door='0' and alarm='0'
report "there is error"
severity warning;
code<="1101";
wait for 50 ns;
```

```
assert door='1' and alarm='0'
report "Door Must Open"
severity warning;
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
reset<='0';
            code<="0010";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
            code<="0110";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
            code<="1101";
wait for 50 ns;
```

```
assert door='1' and alarm='0'
report "Door Must Open"
severity warning;
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
reset<='0';
            code<="0010";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
            code<="0110";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
            code<="1010";
            wait for 50 ns;
```

```
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
            code<="1101";
            wait for 50 ns;
            assert door='1' and alarm='0'
            report "Door Must Open"
            severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
reset<='0';
           code<="0010";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
```

```
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
            code<="0110";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
            code<="1010";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
                  code<="0000";
            wait for 50 ns;
            assert door='0' and alarm='0'
            report "there is error"
            severity warning;
            code<="1101";
            wait for 50 ns;
            assert door='1' and alarm='0'
            report "Door Must Open"
            severity warning;
assert door=door_struct and alarm=alarm_struct
```

```
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
reset<='1';
code<=X"0";
wait for 50 ns;
  assert door='0' and alarm='0'
  report "Reset error"
  severity warning;
assert door=door_struct and alarm=alarm_struct
  report "Behavioral and Structural Behavior are not the same"
  severity warning;
assert door_struct=door_scan and alarm_struct=alarm_scan
  report "Structural and Scan Behaviors are not the same"
  severity warning;
reset<='0';


            wait;
            end process;
      fsm_struct: codej_b_l port
map(clk,vdd,vss,daytime,reset,code,door_struct,alarm_struct);
                fsm_moore: codej_b_l port
map(clk,vdd,vss,daytime,reset,code,door,alarm);
                fsm_scan: codej_b_l_scan port
map(clk,vdd,vss,daytime,reset,code,door_scan,alarm_scan,scanin,test,scanout);
      end architecture tb_arch;
```