Phase 2 EDA

Name: Mohand Mahmoud Farag

ID: 16p6042

Group: 1

Section: 2

Introduction:

In this phase we use fedora in Vm to implement different circuits to obtain the best in delay and area to be more useful. At first we change the original code to unix then make three files (makefile , paramfile, path.path) (in this zip file).

We use this makefile to generate some operation and then order them by some commands.

Commands:

Make vhd_to_fsm

Make syf

Make boom

Make boog

Make loon

Make flatbeh

Make dft

Then we use the output code from loon and dft in test bench to insure that code is right.

The Syf output

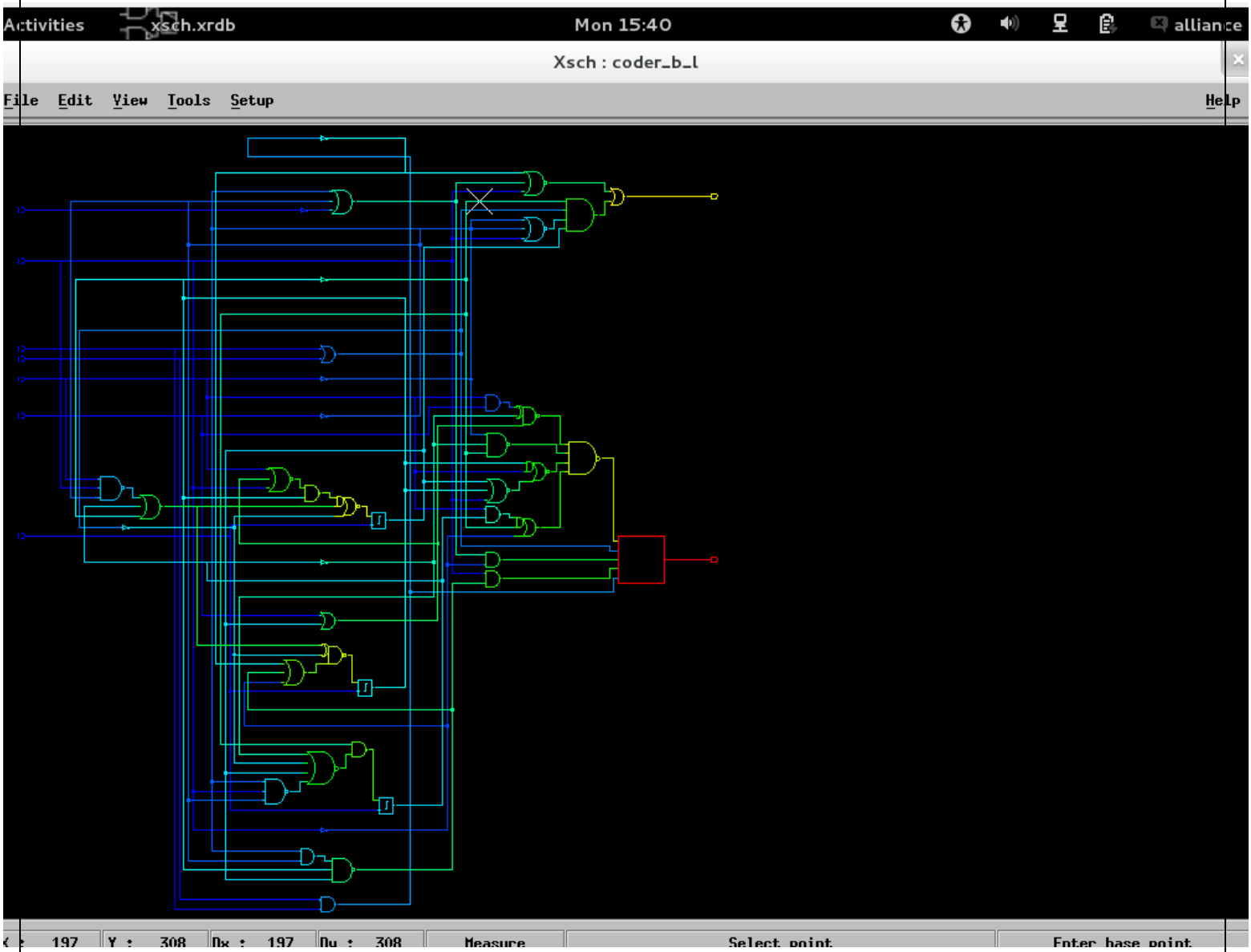
Type	delay	area	Multiplication
A	3221	65500	210975500
J	2687	64000	171968000
M	2305	61250	141181250
O	2458	93500	229823000
R	2271	61750	140234250

I choose the R type because it is the minimum multiplication

The test bench of loon:

Test number	Daytime	Code	Reset	Expected Door & Doors	Expected Alarm & Alarms	Door & Doors	Alarm & Alarms
Reset test	X	X	1	0	0	0	0
O with dt=1 (test case 1)	1	1101	0	1	0	1	0
Full input with dt=1 (test case 2)	1	0010	0	0	0	0	0
	1	0110	0	0	0	0	0
	1	1010	0	0	0	0	0
	1	0000	0	0	0	0	0
	1	0101	0	1	0	1	0
O with dt=1 (test case 3)	0	1101	0	0	1	0	1
Full input with dt=0 (test case 4)	0	0010	0	0	0	0	0
	0	0110	0	0	0	0	0
	0	1010	0	0	0	0	0
	0	0000	0	0	0	0	0
	0	0101	0	1	0	1	0
Incorrect input (test case 5)	0	0010	0	0	0	0	0
	0	0110	0	0	0	0	0
	0	1010	0	0	1	0	1

The circuit:


Activities
xsch.xrdb
Mon 15:40
alliance
Xsch : coder_b_l
File Edit View Tools Setup
Help
X : 197
Y : 308
Dx : 197
Dy : 308
Measure
Select point
Enter base point

The code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use ieee.NUMERIC_STD.all;
ENTITY projecttbb IS
END ENTITY projecttbb;
ARCHITECTURE fsmtbb OF projecttbb IS
component project1 is
port  (
   clk    : in      bit;
                    code   : in          bit_vector(3 downto 0) ;
                    daytime    : in    bit;
                    reset : inbit;
                    vdd   : in bit;
                    vss   : in  bit;
                    door    : out       bit;
                    alarm     : out     bit
    );
end component;
component coder_b_l is
  port (
    clk    : in     bit;
    code    : in     bit_vector(3 downto 0);
    daytime : in     bit;
    reset   : in     bit;
    vdd     : in     bit;
```

```
    vss    : in     bit;
    door   : out    bit;
    alarm  : out    bit
 );
end component;

FOR dut: project1 USE ENTITY WORK.project1 (FSM);
FOR duts: coder_b_l USE ENTITY WORK.coder_b_l (structural);
SIGNAL clk    : bit := '0';
SIGNAL code    : bit_vector(3 downto 0) := "0000";
SIGNAL daytime    : bit := '1';
SIGNAL reset : bit := '1';
SIGNAL vdd   : bit := '1';
SIGNAL vss   : bit := '0';
signal door : bit:= '0';
signal alarm : bit := '0';
--SIGNAL clk    : bit := '0';
SIGNAL codes    : bit_vector(3 downto 0) := "0000";
SIGNAL daytimes    : bit := '1';
SIGNAL resets : bit := '1';
--SIGNAL vdds   : bit := '1';
--SIGNAL vsss   : bit := '0';
signal doors : bit:= '0';
signal alarms : bit := '0';

begin
dut: project1 PORT MAP (clk, code, daytime, reset,vdd, vss, door, alarm);
duts: coder_b_l PORT MAP (clk, codes, daytimes, resets,vdd, vss, doors, alarms);
```

```
 clk_process :process
begin
      clk <= '1';
      wait for 10 ns;
      clk <= '0';
      wait for 10 ns;

  end process;
p1:process is begin
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT "1 error"
SEVERITY error;
---- first testbensh
daytime<='1';
code<="1101";
reset<='0';
daytimes<='1';
codes<="1101";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT "2 error"
SEVERITY error;
---------second testbench
daytime<='1';
code<="0010";
reset<='0';
daytimes<='1';
```

```
codes<="0010";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT " 3 error"
SEVERITY error;
------------------------
daytime<='1';
code<="0110";
reset<='0';
daytimes<='1';
codes<="0110";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT " 4 error"
SEVERITY error;
------------------------
daytime<='1';
code<="1010";
reset<='0';
daytimes<='1';
codes<="1010";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT " 5 error"
SEVERITY error;
-------------------------
```

```
daytime<='1';
code<="0000";
reset<='0';
daytimes<='1';
codes<="0000";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT " 6 error"
SEVERITY error;
-------------------------
daytime<='1';
code<="0101";
reset<='0';
daytimes<='1';
codes<="0101";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT " 7 error"
SEVERITY error;
----------third testbench
daytime<='0';
code<="1101";
reset<='0';
daytimes<='0';
codes<="1101";
resets<='0';
wait For 20 ns;
```

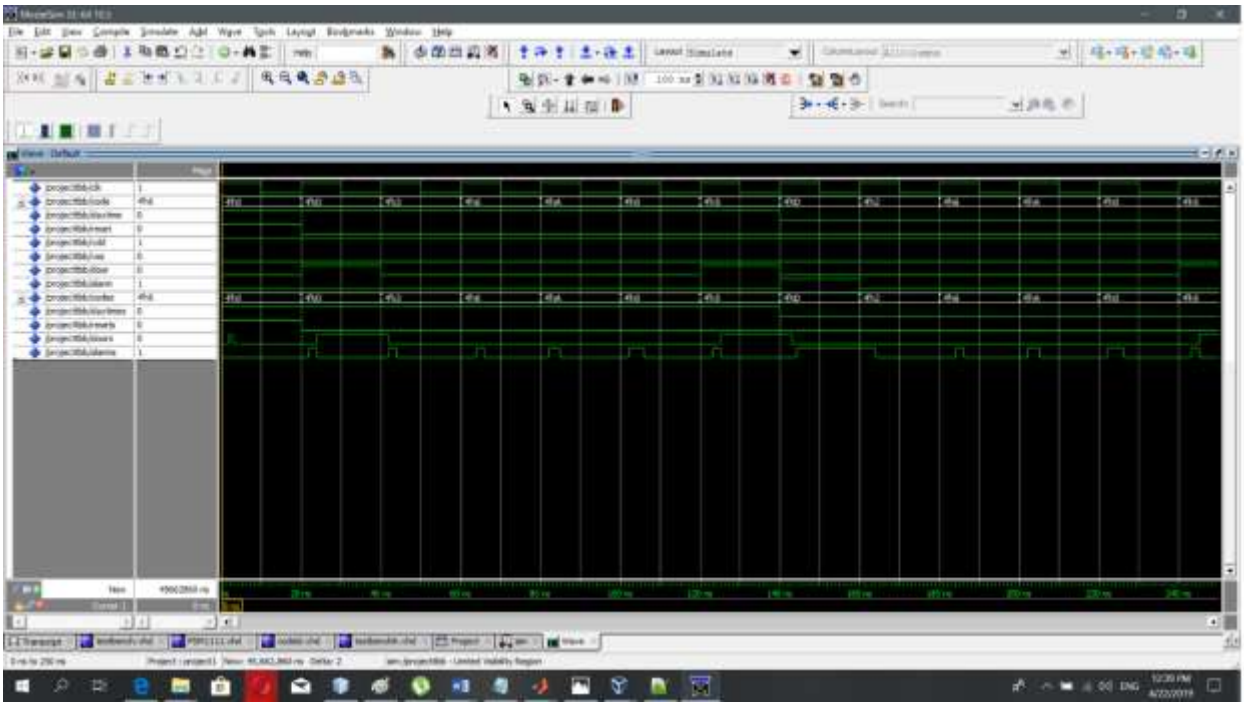
```
ASSERT door = doors and alarm = alarms
REPORT "8 error"
SEVERITY error;
----------fourth testnech
daytime<='0';
code<="0010";
reset<='0';
daytimes<='0';
codes<="0010";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT "9 error"
SEVERITY error;
---------------------------
daytime<='0';
code<="0110";
reset<='0';
daytimes<='0';
codes<="0110";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT "10 error"
SEVERITY error;
---------------------------
daytime<='0';
code<="1010";
reset<='0';
```

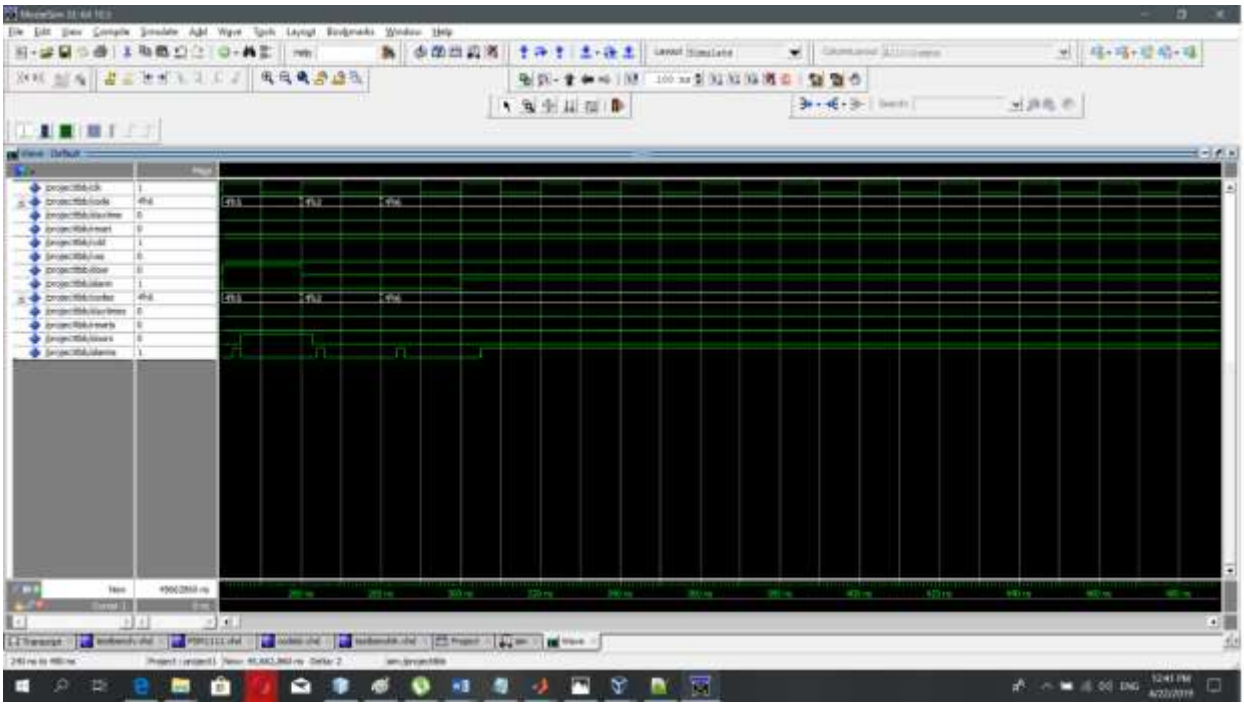
```
daytimes<='0';
codes<="1010";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT "11 error"
SEVERITY error;
----------------------------
daytime<='0';
code<="0000";
reset<='0';
daytimes<='0';
codes<="0000";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT "12 error"
SEVERITY error;
----------------------------
daytime<='0';
code<="0101";
reset<='0';
daytimes<='0';
codes<="0101";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT "13 error"
SEVERITY error;
```

```
----------fifth testbench
daytime<='0';
code<="0010";
reset<='0';
daytimes<='0';
codes<="0010";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT "14 error"
SEVERITY error;
----------
daytime<='0';
code<="0110";
reset<='0';
daytimes<='0';
codes<="0110";
resets<='0';
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT "15 error"
SEVERITY error;
----------
daytime<='0';
code<="0110";
reset<='0';
daytimes<='0';
codes<="0110";
resets<='0';
```

```
wait For 20 ns;
ASSERT door = doors and alarm = alarms
REPORT "16 error"
SEVERITY error;
---------------------------
WAIT;
END PROCESS ;
END ARCHITECTURE ;
```

The output:

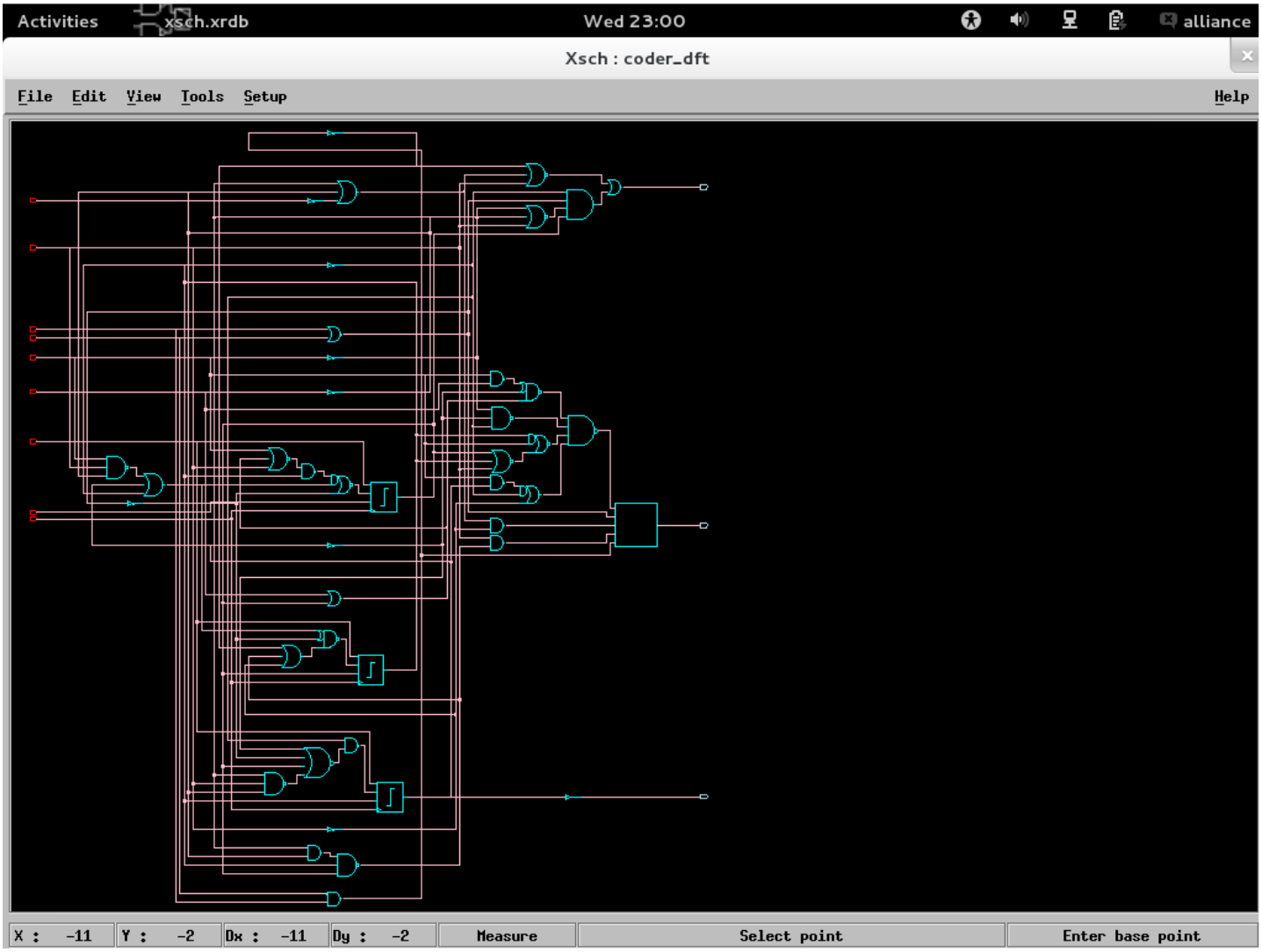




The test bench of DFT:

Test number	Daytime	Code	Reset	Expected Door & Doors & Doord	Expected Alarm & Alarms & Alarmd	Door & Doors & Doord	Alarm & Alarms & Alarmd
Reset test	X	X	1	0	0	0	0
O with dt=1 (test case 1)	1	1101	0	1	0	1	0
Full input with dt=1 (test case 2)	1	0010	0	0	0	0	0
	1	0110	0	0	0	0	0
	1	1010	0	0	0	0	0
	1	0000	0	0	0	0	0
	1	0101	0	1	0	1	0
O with dt=1 (test case 3)	0	1101	0	0	1	0	1
Full input with dt=0 (test case 4)	0	0010	0	0	0	0	0
	0	0110	0	0	0	0	0
	0	1010	0	0	0	0	0
	0	0000	0	0	0	0	0
	0	0101	0	1	0	1	0
Incorrect input (test case 5)	0	0010	0	0	0	0	0
	0	0110	0	0	0	0	0
	0	1010	0	0	1	0	1

The circuit:


Activities
xsch.xrdb
Wed 23:00
alliance
Xsch : coder_dft
File Edit View Tools Setup Help
X : -11
Y : -2
Dx : -11
Dy : -2
Measure
Select point
Enter base point

The Code:

```
library Sxlib_ModelSim;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use ieee.NUMERIC_STD.all;
ENTITY projecttbb IS
END ENTITY projecttbb;
ARCHITECTURE fsmtbb OF projecttbb IS
component project1 is
port  (
   clk    : in        bit;
          code   : in         bit_vector(3 downto 0) ;
          daytime    : in    bit;
          reset : in          bit;
          vdd   : in          bit;
          vss   : in bit;
          door    : out      bit;
          alarm    : out     bit
    );
end component;
component coder_b_l is
  port (
    clk    : in     bit;
    code    : in     bit_vector(3 downto 0);
    daytime : in     bit;
    reset   : in     bit;
    vdd     : in     bit;
```

```
    vss     : in      bit;
    door    : out     bit;
    alarm   : out     bit
 );
end component;
component coder_dft is
  port (
    clk     : in      bit;
    code    : in      bit_vector(3 downto 0);
    daytime : in      bit;
    reset   : in      bit;
    vdd     : in      bit;
    vss     : in      bit;
    door    : out     bit;
    alarm   : out     bit;
    scanin  : in      bit;
    test    : in      bit;
    scanout : out     bit
 );
end component;
FOR dut: project1 USE ENTITY WORK.project1 (FSM);
FOR duts: coder_b_l USE ENTITY WORK.coder_b_l (structural);
FOR dutd: coder_dft USE ENTITY WORK.coder_dft (structural11);
SIGNAL clk    : bit := '0';
SIGNAL code     : bit_vector(3 downto 0) := "0000";
SIGNAL daytime     : bit := '1';
SIGNAL reset : bit := '1';
SIGNAL vdd   : bit := '1';
SIGNAL vss   : bit := '0';
```

```
signal door : bit:= '0';
signal alarm : bit := '0';
--SIGNAL clk    : bit := '0';
SIGNAL codes     : bit_vector(3 downto 0) := "0000";
SIGNAL daytimes     : bit := '1';
SIGNAL resets : bit := '1';
--SIGNAL vdds   : bit := '1';
--SIGNAL vsss   : bit := '0';
signal doors : bit:= '0';
signal alarms : bit := '0';
SIGNAL coded     : bit_vector(3 downto 0) := "0000";
SIGNAL daytimed     : bit := '1';
SIGNAL resetd : bit := '1';
signal doord : bit:= '0';
signal alarmd : bit := '0';
SIGNAL scanin: bit := '0';
SIGNAL test: bit := '0' ;
SIGNAL scanout: bit := '0' ;
SIGNAL seqance: bit_vector (3 downto 0) := "1101" ;
begin
dut: project1 PORT MAP (clk, code, daytime, reset,vdd, vss, door, alarm);
duts: coder_b_l PORT MAP (clk, codes, daytimes, resets,vdd, vss, doors, alarms);
dutd: coder_dft PORT MAP (clk, coded, daytimed, resetd,vdd, vss, doord, alarmd,scanin,test,scanout);
 clk_process :process
begin
      clk <= '1';
      wait for 10 ns;
      clk <= '0';
      wait for 10 ns;
```

```
  end process;
p1:process is begin
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT "1 error"
SEVERITY error;
---- first testbensh
daytime<='1';
code<="1101";
reset<='0';
daytimes<='1';
codes<="1101";
resets<='0';
daytimed<='1';
coded<="1101";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT "2 error"
SEVERITY error;
---------second testbench
daytime<='1';
code<="0010";
reset<='0';
daytimes<='1';
codes<="0010";
resets<='0';
daytimed<='1';
```

```
coded<="0010";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT " 3 error"
SEVERITY error;
------------------------
daytime<='1';
code<="0110";
reset<='0';
daytimes<='1';
codes<="0110";
resets<='0';
daytimed<='1';
coded<="0110";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT " 4 error"
SEVERITY error;
------------------------
daytime<='1';
code<="1010";
reset<='0';
daytimes<='1';
codes<="1010";
resets<='0';
daytimed<='1';
coded<="1010";
```

```
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT " 5 error"
SEVERITY error;
-------------------------
daytime<='1';
code<="0000";
reset<='0';
daytimes<='1';
codes<="0000";
resets<='0';
daytimed<='1';
coded<="0000";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT " 6 error"
SEVERITY error;
-------------------------
daytime<='1';
code<="0101";
reset<='0';
daytimes<='1';
codes<="0101";
resets<='0';
daytimed<='1';
coded<="0101";
resetd<='0';
```

```
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT " 7 error"
SEVERITY error;
----------third testbench
daytime<='0';
code<="1101";
reset<='0';
daytimes<='0';
codes<="1101";
resets<='0';
daytimed<='0';
coded<="1101";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT "8 error"
SEVERITY error;
----------fourth testnech
daytime<='0';
code<="0010";
reset<='0';
daytimes<='0';
codes<="0010";
resets<='0';
daytimed<='0';
coded<="0010";
resetd<='0';
wait For 20 ns;
```

```
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT "9 error"
SEVERITY error;
----------------------------
daytime<='0';
code<="0110";
reset<='0';
daytimes<='0';
codes<="0110";
resets<='0';
daytimed<='0';
coded<="0110";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT "10 error"
SEVERITY error;
----------------------------
daytime<='0';
code<="1010";
reset<='0';
daytimes<='0';
codes<="1010";
resets<='0';
daytimed<='0';
coded<="1010";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
```

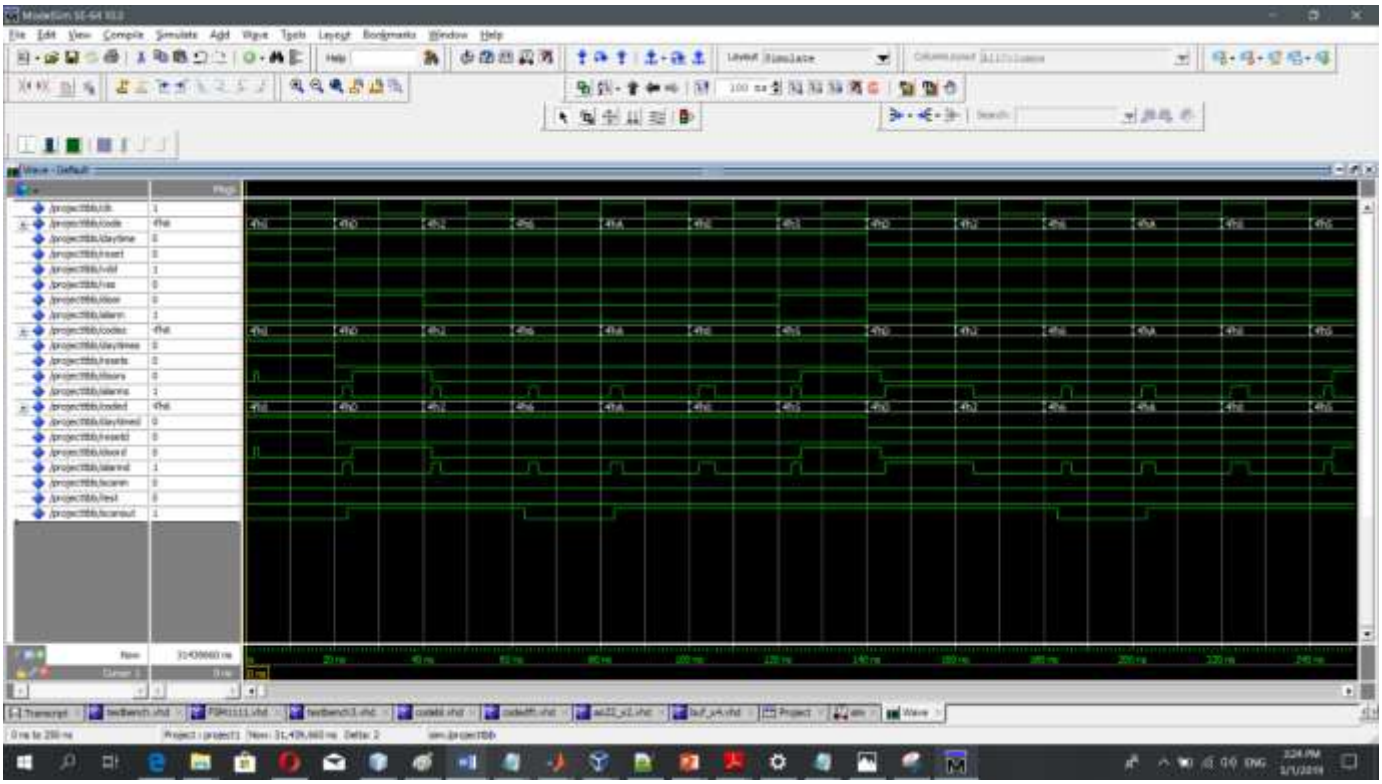
```
REPORT "11 error"
SEVERITY error;
----------------------------
daytime<='0';
code<="0000";
reset<='0';
daytimes<='0';
codes<="0000";
resets<='0';
daytimed<='0';
coded<="0000";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT "12 error"
SEVERITY error;
----------------------------
daytime<='0';
code<="0101";
reset<='0';
daytimes<='0';
codes<="0101";
resets<='0';
daytimed<='0';
coded<="0101";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT "13 error"
```

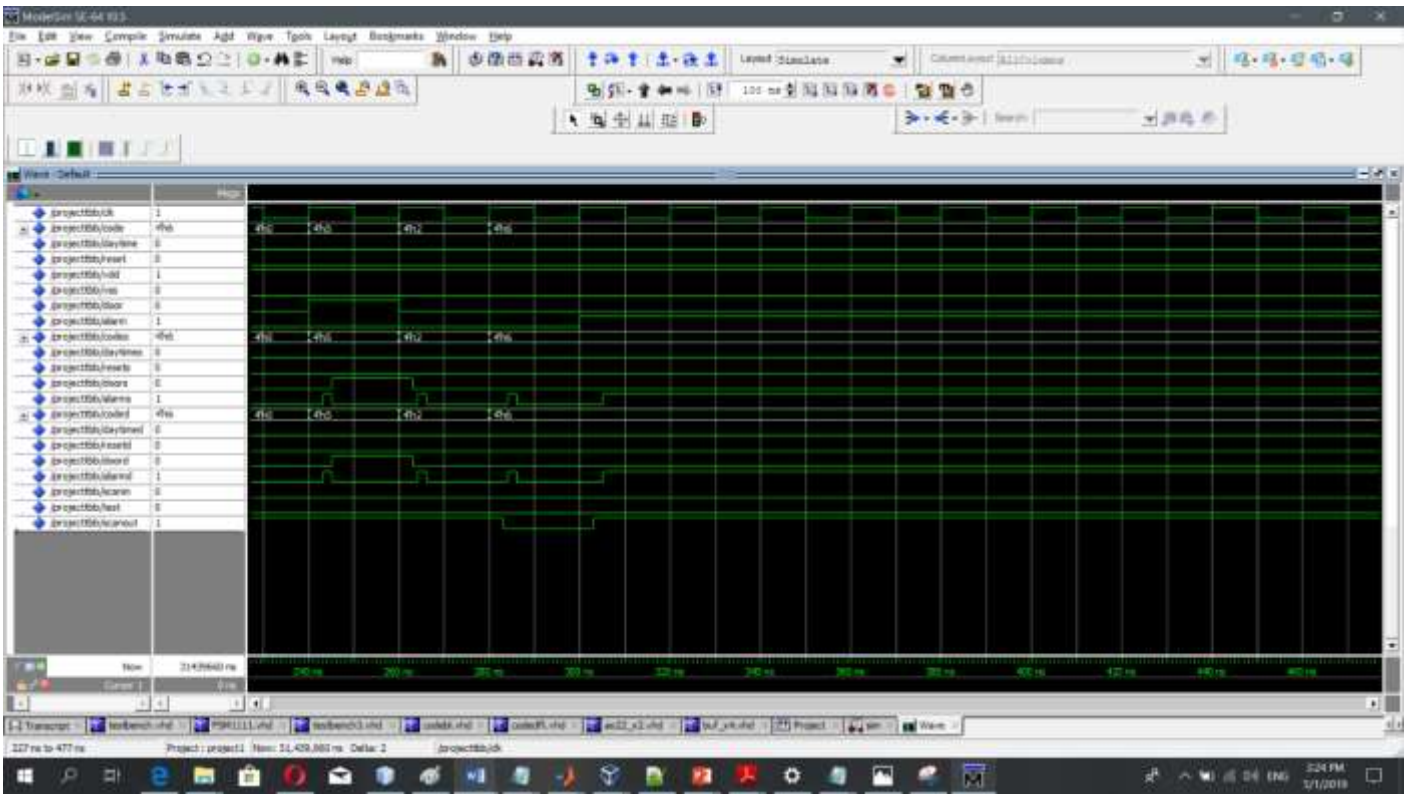
```
SEVERITY error;
----------fifth testbench
daytime<='0';
code<="0010";
reset<='0';
daytimes<='0';
codes<="0010";
resets<='0';
daytimed<='0';
coded<="0010";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT "14 error"
SEVERITY error;
---------------------------
daytime<='0';
code<="0110";
reset<='0';
daytimes<='0';
codes<="0110";
resets<='0';
daytimed<='0';
coded<="0110";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT "15 error"
SEVERITY error;
```

```
----------------------------
daytime<='0';
code<="0110";
reset<='0';
daytimes<='0';
codes<="0110";
resets<='0';
daytimed<='0';
coded<="0110";
resetd<='0';
wait For 20 ns;
ASSERT door = doors and door = doord and alarm =alarms and alarm = alarmd
REPORT "16 error"
SEVERITY error;
----------------------------

WAIT;
END PROCESS ;
END ARCHITECTURE ;
```

The output:





The Test bench of scan:

test	scanin	Expected scanout	scanout
1	1010	1010	1010

The Code:

```
library Sxlib_ModelSim;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use ieee.NUMERIC_STD.all;
ENTITY testbenchsc IS
END ENTITY testbenchsc;
ARCHITECTURE fsmtbsc OF testbenchsc IS
component coder_dft is
  port (
    clk    : in     bit;
    code   : in     bit_vector(3 downto 0);
    daytime : in    bit;
    reset  : in     bit;
    vdd    : in     bit;
    vss    : in     bit;
    door   : out    bit;
    alarm  : out    bit;
    scanin : in     bit;
    test   : in     bit;
    scanout : out   bit
```

```
 );
end component;
FOR dut: coder_dft USE ENTITY WORK.coder_dft (structural11);
SIGNAL clk    : bit := '0';
SIGNAL vdd   : bit := '1';
SIGNAL vss   : bit := '0';
SIGNAL coded     : bit_vector(3 downto 0) := "0000";
SIGNAL daytimed     : bit := '1';
SIGNAL resetd : bit := '1';
signal doord : bit:= '0';
signal alarmd : bit ;
SIGNAL scanin: bit ;
SIGNAL test: bit  ;
SIGNAL scanout: bit;
SIGNAL seqance: bit_vector (3 downto 0) := "1010" ;
begin
dut: coder_dft PORT MAP (clk, coded, daytimed, resetd,vdd, vss, doord, alarmd,scanin,test,scanout);
clk_process :process
begin
      clk <= '1';
      wait for 10 ns;
      clk <= '0';
      wait for 10 ns;

  end process;
p1:process is begin
test <= '1';
            for i In 0 to seqance'length-1
            loop scanin <= seqance(i);
```

```
        wait for 20 ns;
        if i>3 then
        Assert scanout=seqance(i-3)
        Report "scanout does not follow scan in"
        Severity error;
        end if;
        end loop;
WAIT;
END PROCESS ;
END ARCHITECTURE ;
```

The output: