Student Name: Omar Magdy Shaaban

<CESS>

<Project Phase 2>

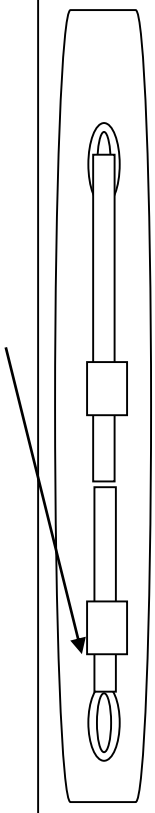
UEL Module Title: Engineering Systems

Module Code: EG7423

ASU Course Title: Electronic Design Automation

Course Code: CSE215

Using file fastener



Date of Submission: 1/5/2019

Lecturer: Dr. Mohamed Dessouky

UELID: u1927296

ASUID: 16P6034

Email: 16P6034@eng.asu.edu.eg

University of East London

Abstract

This report illustrates the conversion of the finite state machine produced in Phase 1 of the project, to behavioral and structural architectures in several state encodings. Each one is then logically optimized, converted to gate netlist and optimized concerning delay. At this point, the structural architecture should be ready to perform the exact behavior of the FSM produced in Phase 1.

Further verification can be done by selecting the final structural architecture, converting it again to behavioral and check its validity. Also, the structural file ".vst" can be used in a testbench to check correct functionality of the output file.

Table of Contents

Figures 3

1. Obtained Circuit Netlists 4

2. State Encodings 15

3. Simulation Output 16

Figures

Figure 1: sdeta_b netlist 4
Figure 2: sdetj_b netlist 5
Figure 3: sdetm_b netlist 6
Figure 4: sdeto_b netlist 7
Figure 5: sdetr_b netlist 8
Figure 6: sdeta_b_l netlist 9
Figure 7: sdetj_b_l netlist 10
Figure 8: sdetm_b_l netlist 11
Figure 9: sdeto_b_l netlist 12
Figure 10: sdetr_b_l netlist 13
Figure 11: sdetm_b_l_scan netlist 14
Figure 12: Simulation results (1) 16
Figure 13: Simulation results (2) 16
Figure 14: Transcript output 17

1. Obtained Circuit Netlists

Boog outputs:

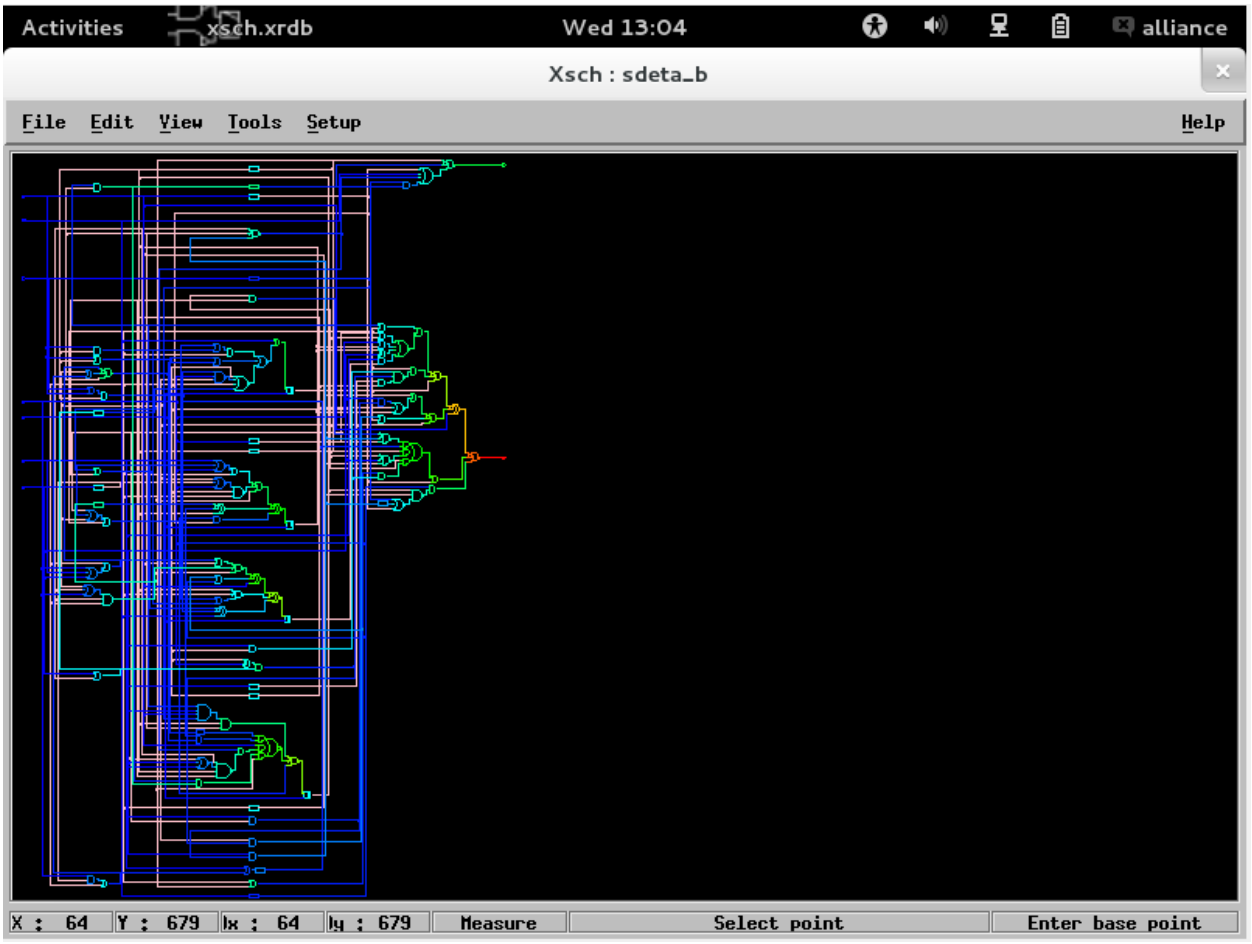


Figure 1: sdeta_b netlist

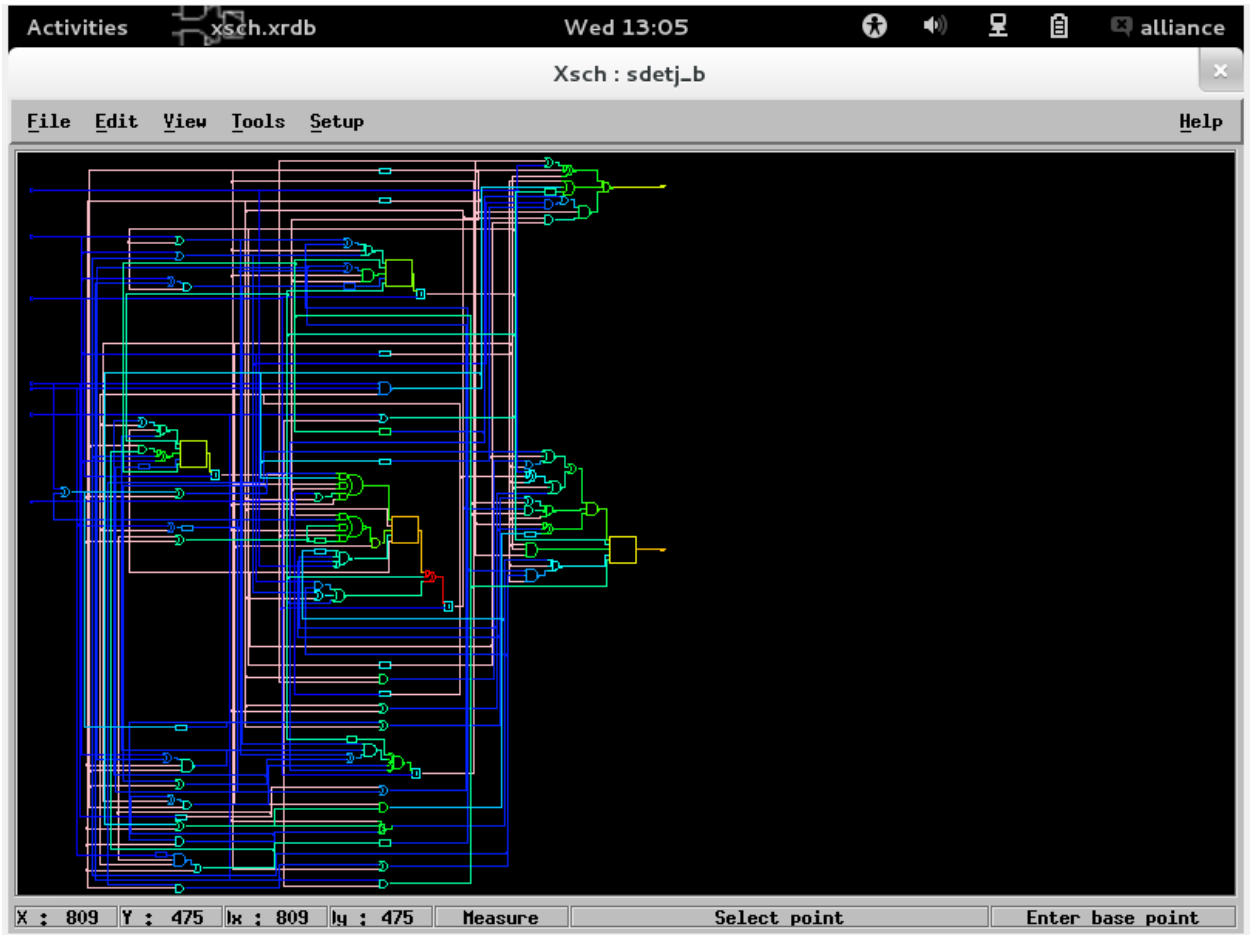


Figure 2: sdetj_b netlist

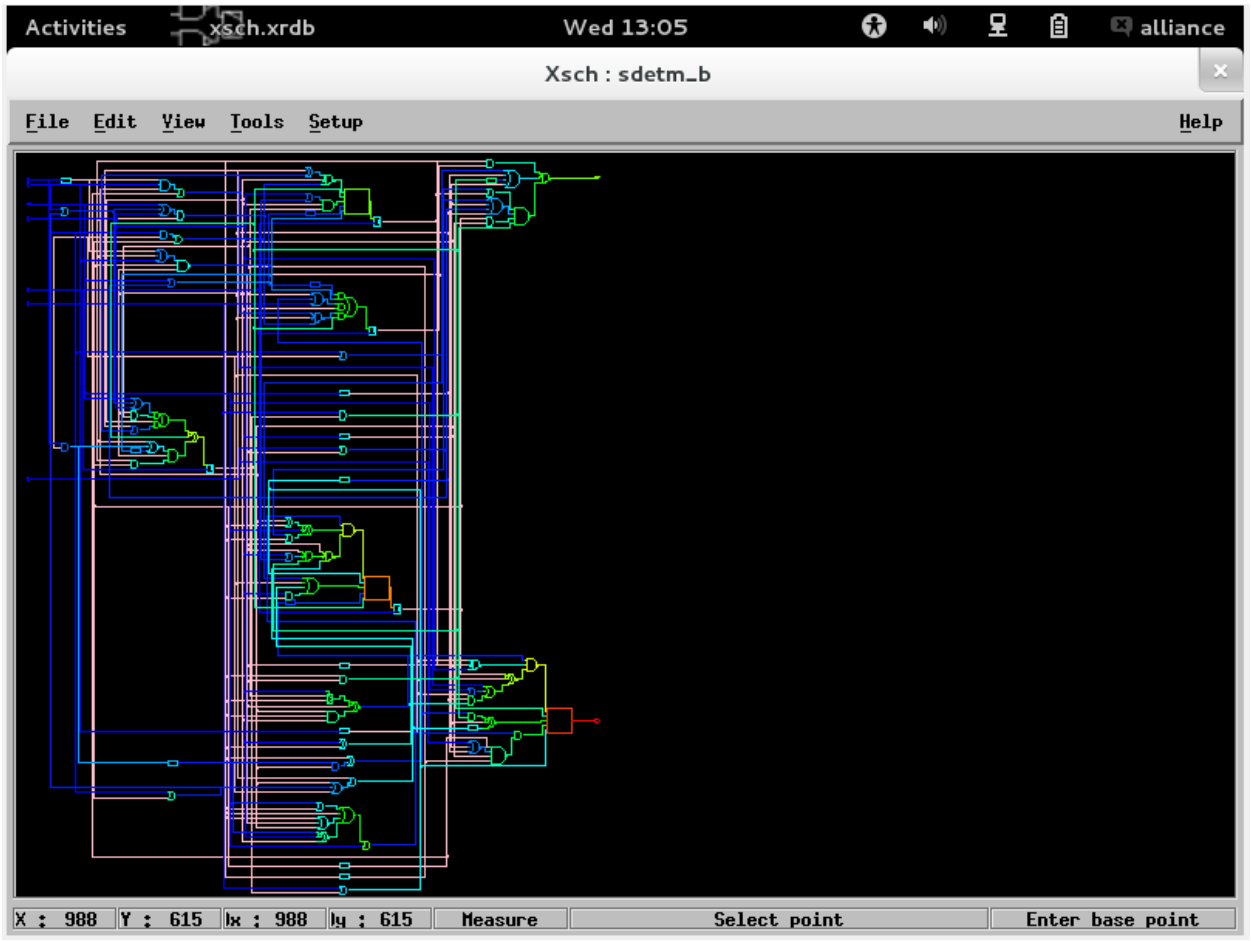


Figure 3: sdetm_b netlist

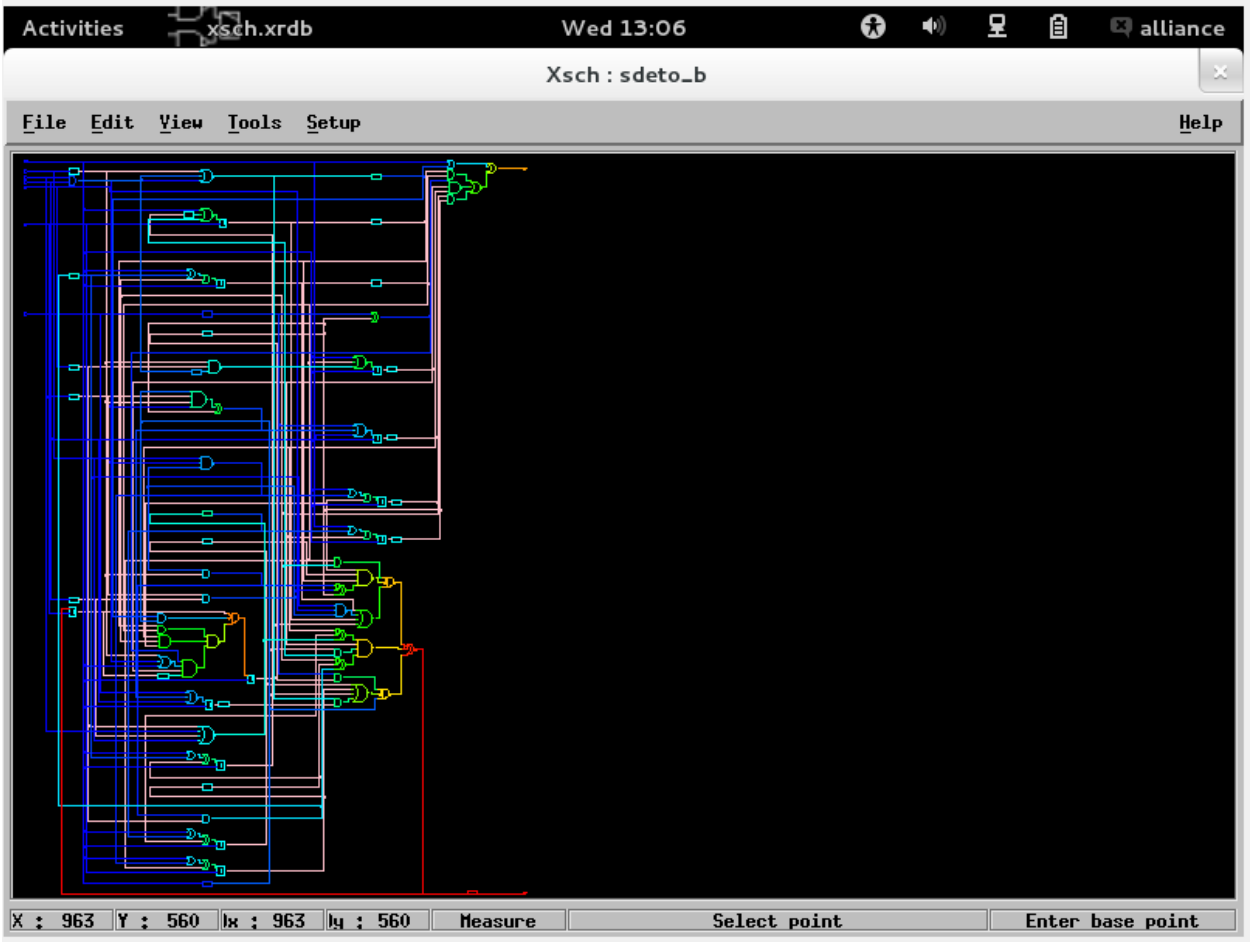


Figure 4: sdeto_b netlist

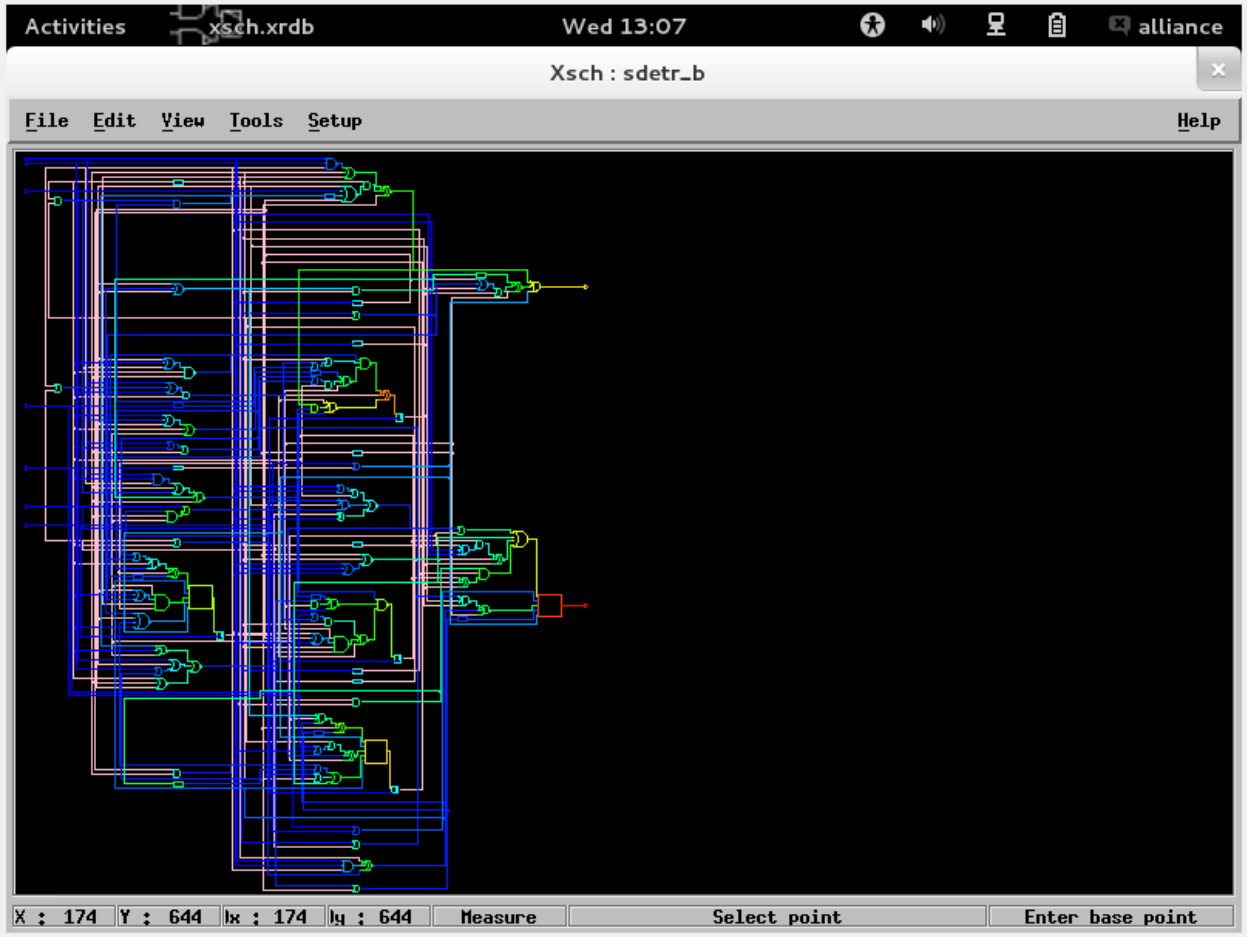


Figure 5: sdetr_b netlist

Loon outputs:

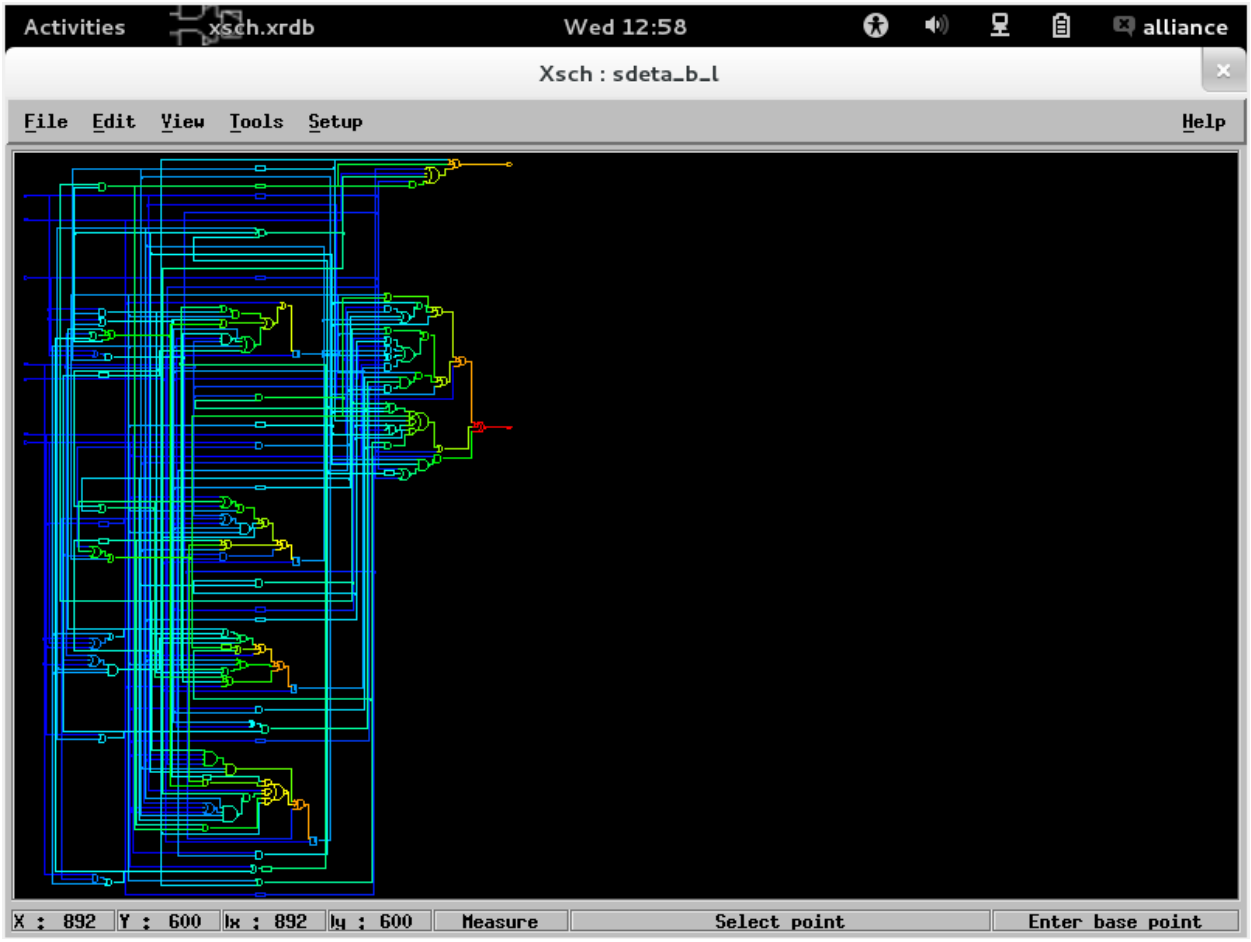


Figure 6: sdeta_b_l netlist

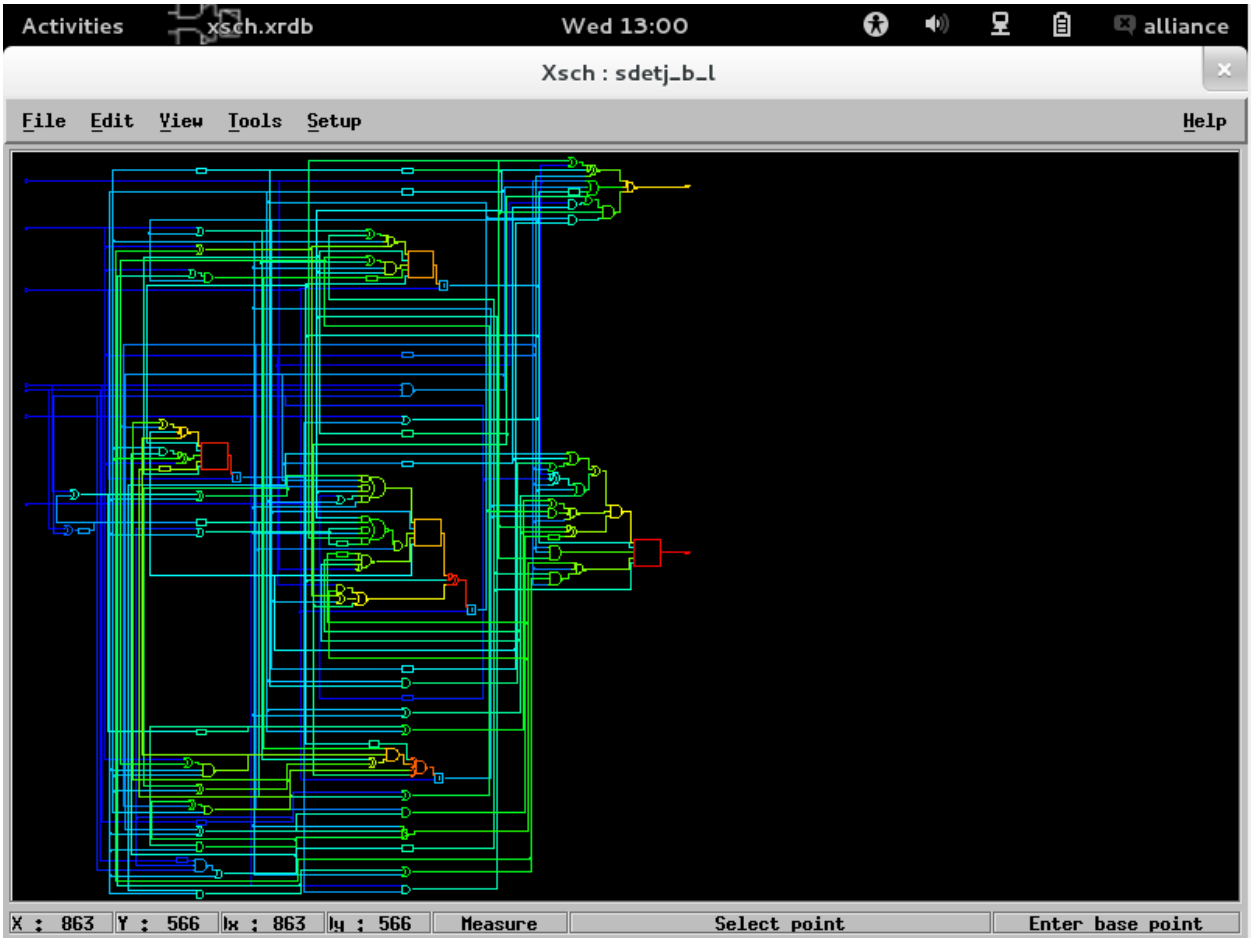


Figure 7: sdetj_b_l netlist

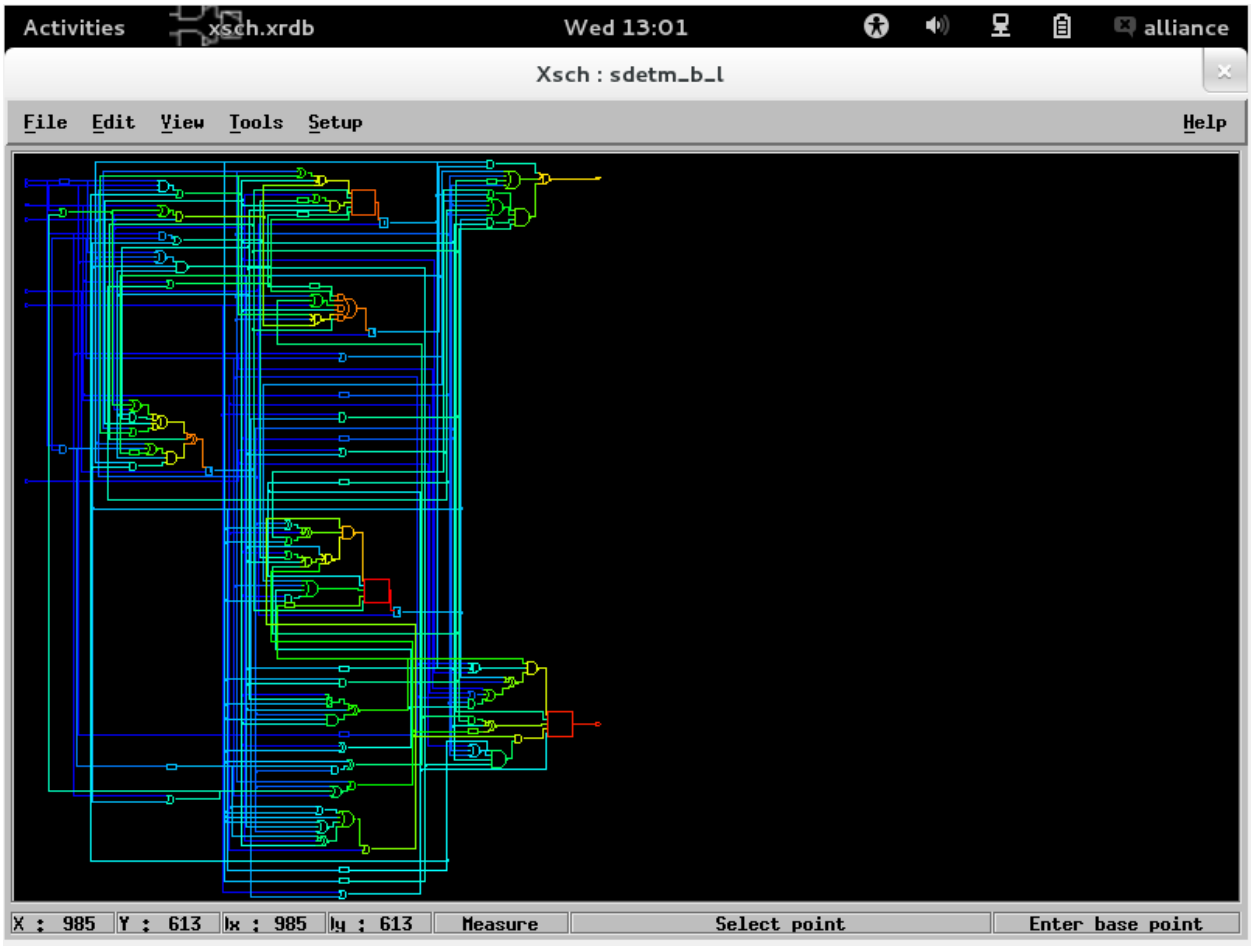


Figure 8: sdetm_b_l netlist

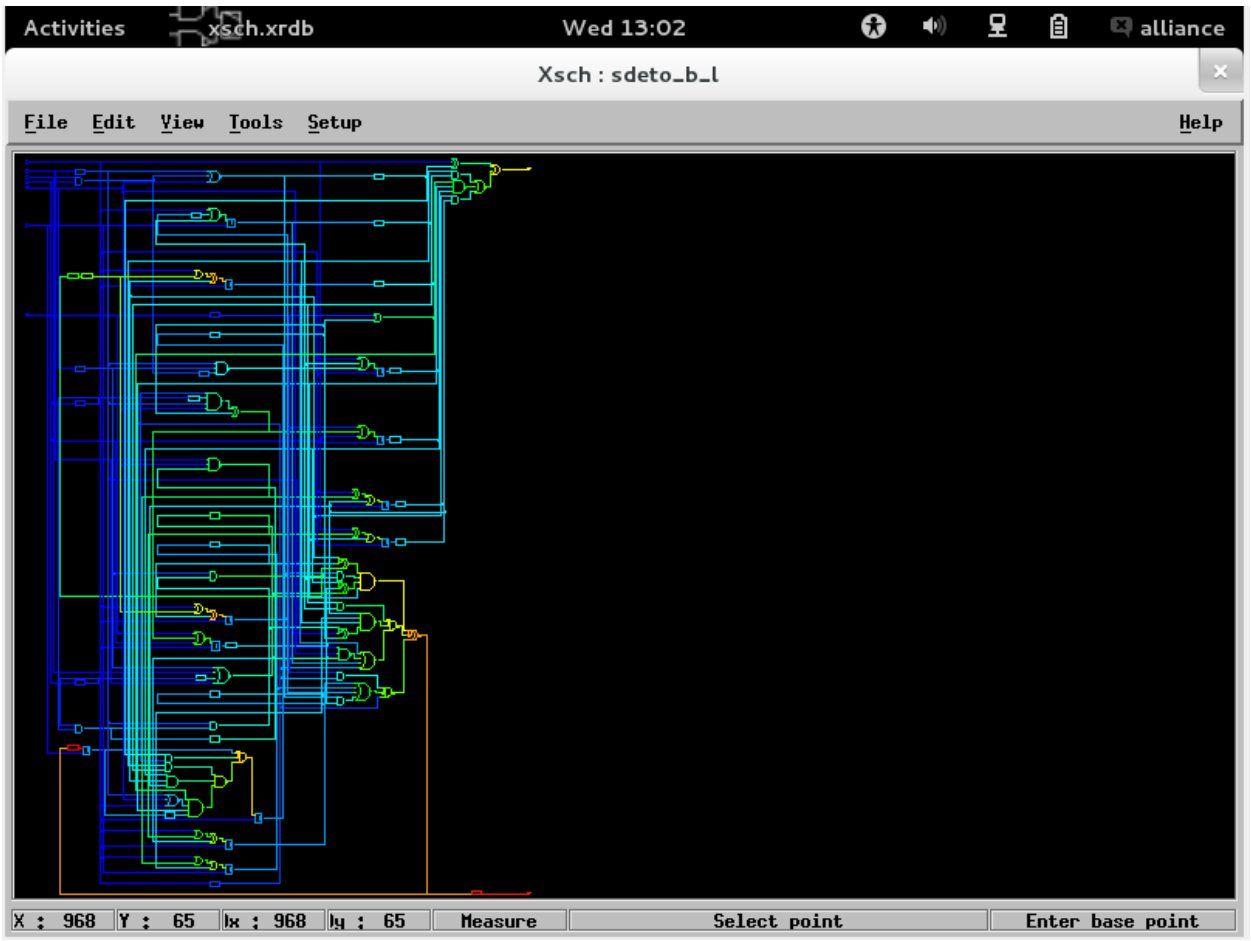


Figure 9: sdeto_b_l netlist

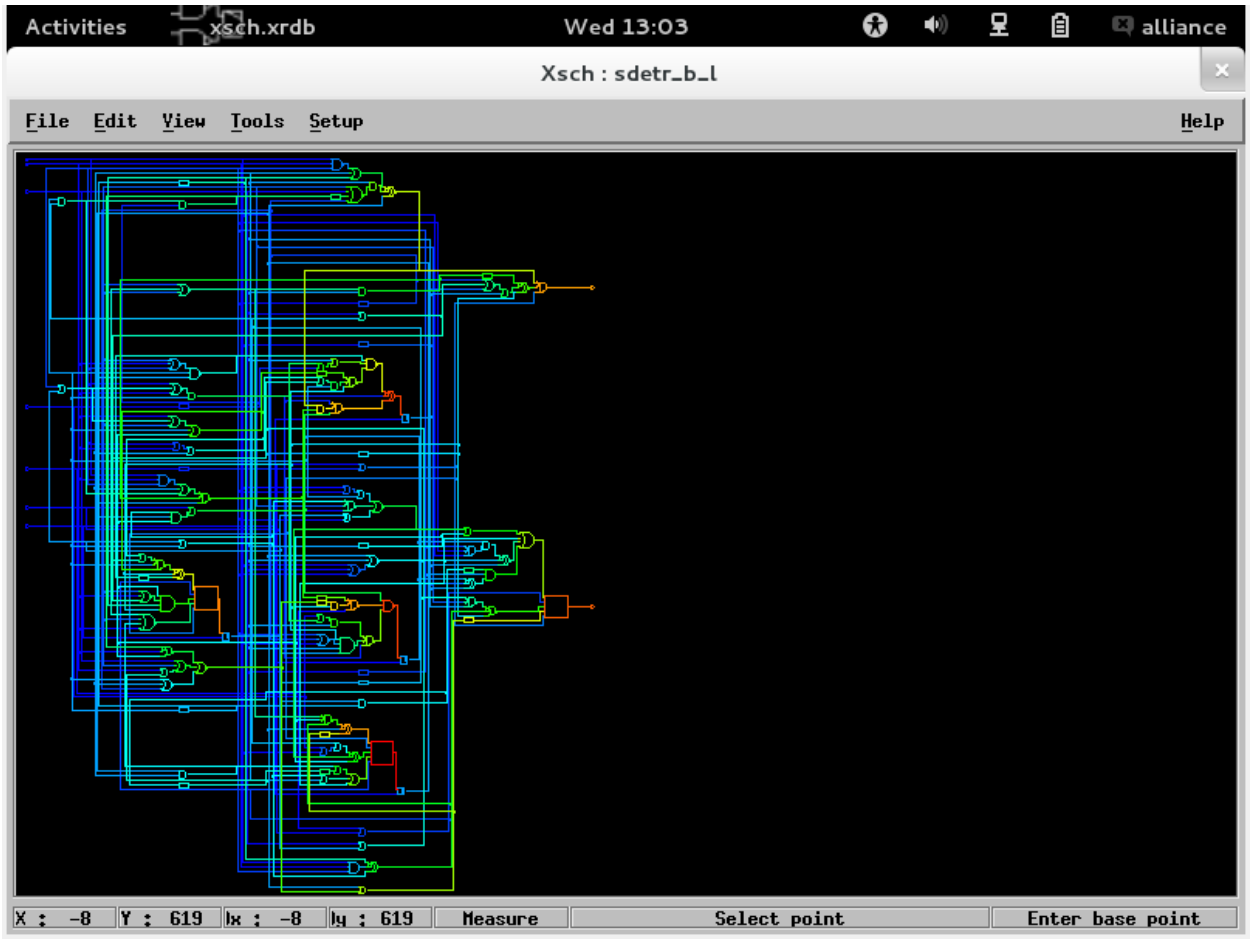


Figure 10: sdetr_b_l netlist

Scapin output for chosen file:

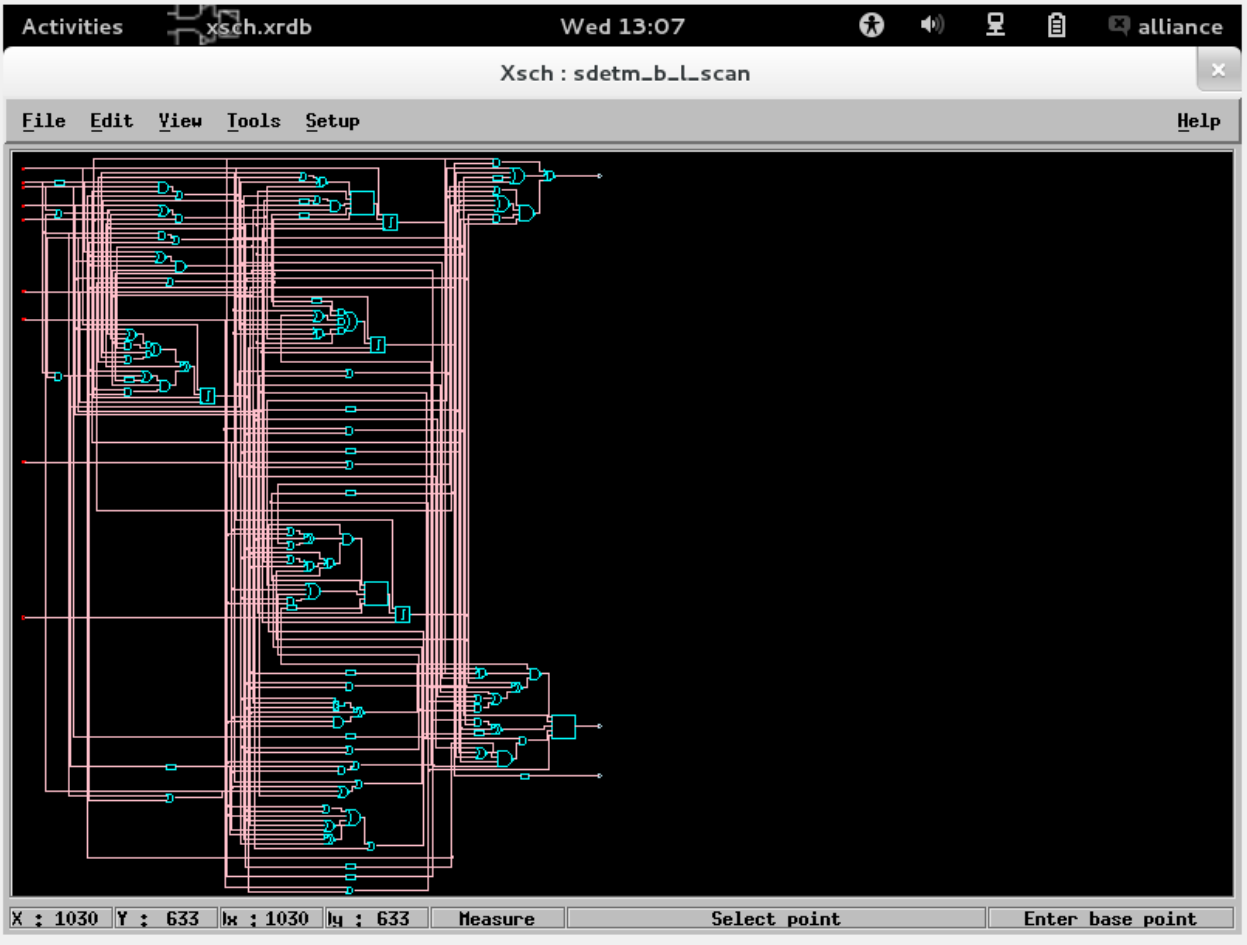


Figure 11: sdetm_b_l_scan netlist

2. State Encodings

State/ Algorithm	a	J	m	o	r
s11	E	6	C	0	6
s10	6	5	5	1	A
s9	C	8	4	2	1
s8	A	4	8	3	C
s7	4	2	2	4	8
s6	0	1	1	5	2
s5	2	C	A	6	5
s4	D	9	F	7	4
s3	5	A	E	8	0
s2	9	B	9	9	E
s1	1	3	3	A	3
s0	8	0	0	B	9

The chosen state encoding is the m encoding, as it produced the most reasonable compromise between area and delay among the rest of the algorithms. Here is another table showing the area and delay of each algorithm:

Algorithm/ Area and Delay	Area	Delay
A	147500	3548
J	135250	2979
M	134750	2966
O	148000	2873
R	159750	3629

3. Simulation Output

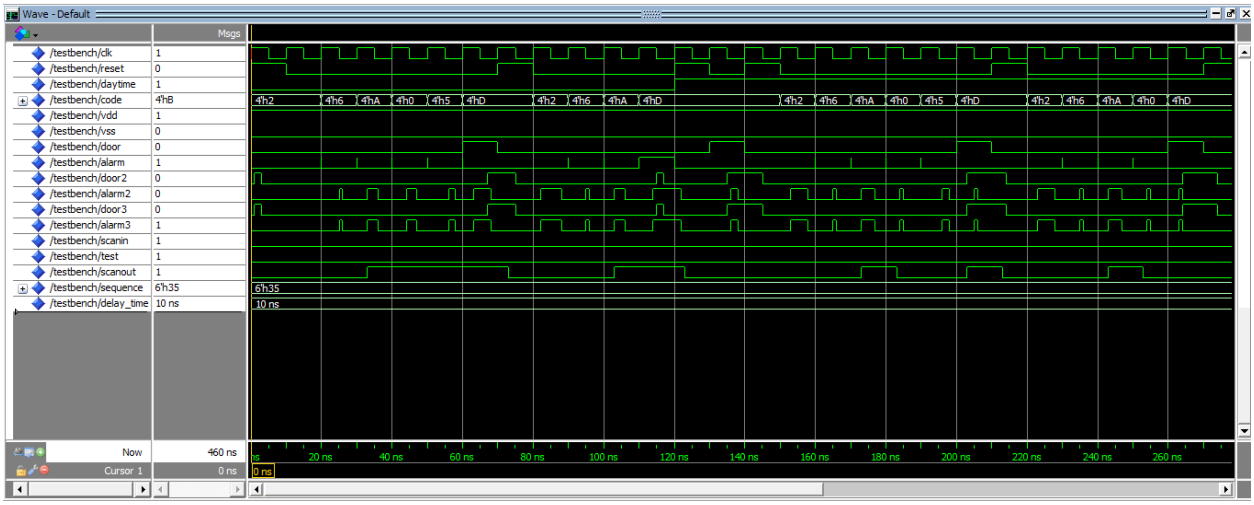


Figure 12: Simulation results (1)

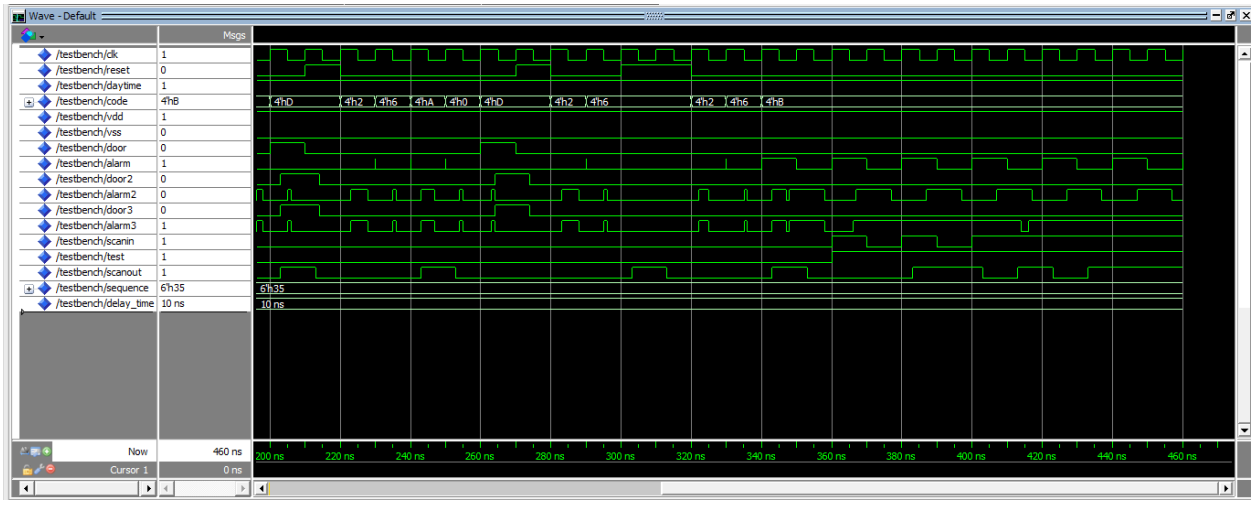


Figure 13: Simulation results (2)

And this is the transcript window accompanying the simulation process:

```
# Loading project Phase2b
ModelSim> vsim -gui work.testbench
# vsim -gui work.testbench
# Start time: 19:46:38 on May 01,2019
# ** Note: (vsim-8009) Loading existing optimized design _opt1
# Loading std.standard
# Loading work.testbench(mealy)#1
# Loading work.doorpass(fsm)#1
# Loading work.sdetm_b_1(structural)#1
# ** Warning: (vsim-3479) Time unit 'ps' is less than the simulator resolution (1ns).
#    Time: 0 ns  Iteration: 0  Instance: /testbench/dp2
# Loading work.sdetm_b_1_scan(structuralscan)#1
add wave -position insertpoint  \
sim:/testbench/clk \
sim:/testbench/reset \
sim:/testbench/daytime \
sim:/testbench/code \
sim:/testbench/vdd \
sim:/testbench/vss \
sim:/testbench/door \
sim:/testbench/alarm \
sim:/testbench/door2 \
sim:/testbench/alarm2 \
sim:/testbench/door3 \
sim:/testbench/alarm3 \
sim:/testbench/scanin \
sim:/testbench/test \
sim:/testbench/scanout \
sim:/testbench/sequence \
sim:/testbench/delay_time
VSIM 3> run

VSIM 4>
```

Figure 14: Transcript output