Electronic Design Automation

Project 2

Presented to:

Prof. Mohamed Dessouky

By:

Muhammad Tamer Bassiony
muhammadbassiony@gmail.com
17P6044
CESS

Contents

Introduction

In this project we continue on our progress from project 1 and explore the next step in IC design; synthesis of our design.

FSM Synthesis

SYF transforms an abstract FSM description in VHDL (.fsm) to a dataflow VHDL description (.vbe), while choosing a state encoding algorithm.

Encodings used:

-a	Asp
-j	Jedi
-m	Mustang
-o	One-hot encoding
-r	Random encoding

Boolean network optimization

- Logic Optimization (Technology Independent)
- Starting point: Transform equations to Boolean equations.
- Two-level logic reduction
- Multi-level logic optimization
- Area estimated by number of literals
- Delay estimated by path length.

Encoding	-a	-j	-m	-o	-r
Literals Before	59	61	61	47	61
Literals After	34	34	38	29	38

Library Mapping

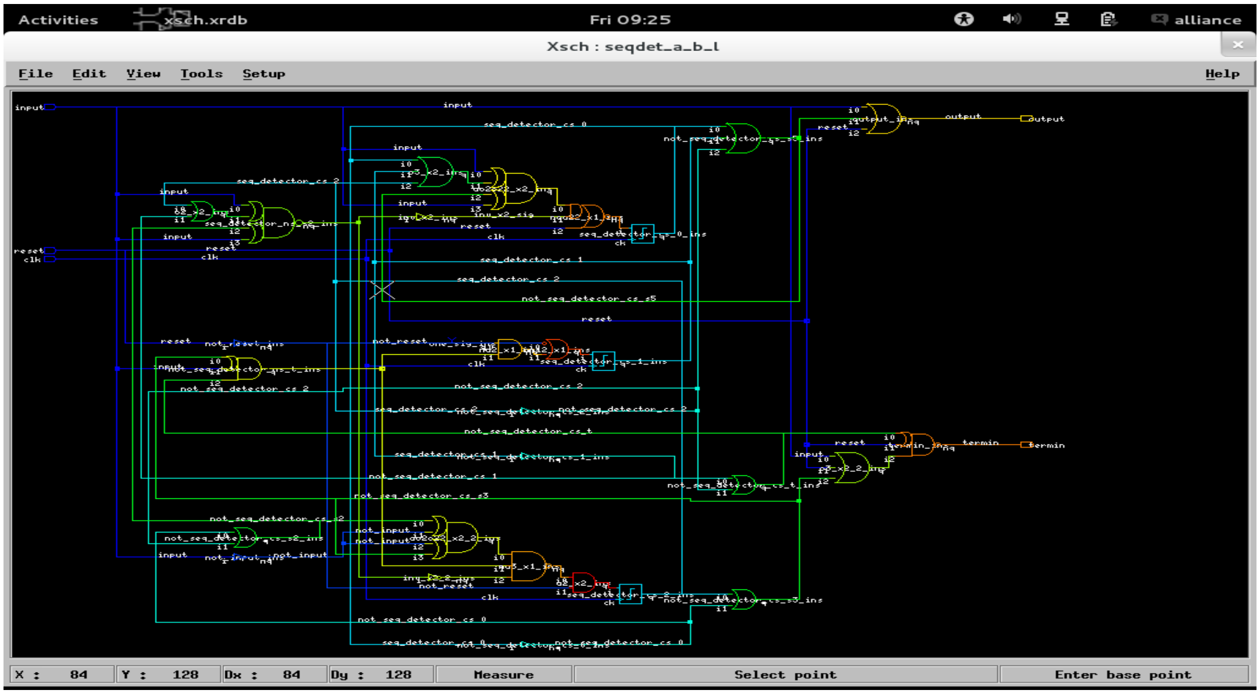
- Produces a gate-level netlistfrom an input behavioral VHDL.
- Handles only logic functions generated by boom.
- Uses only low-driving capability gates
- Recognizes cell characteristics from the .vbegenerics.

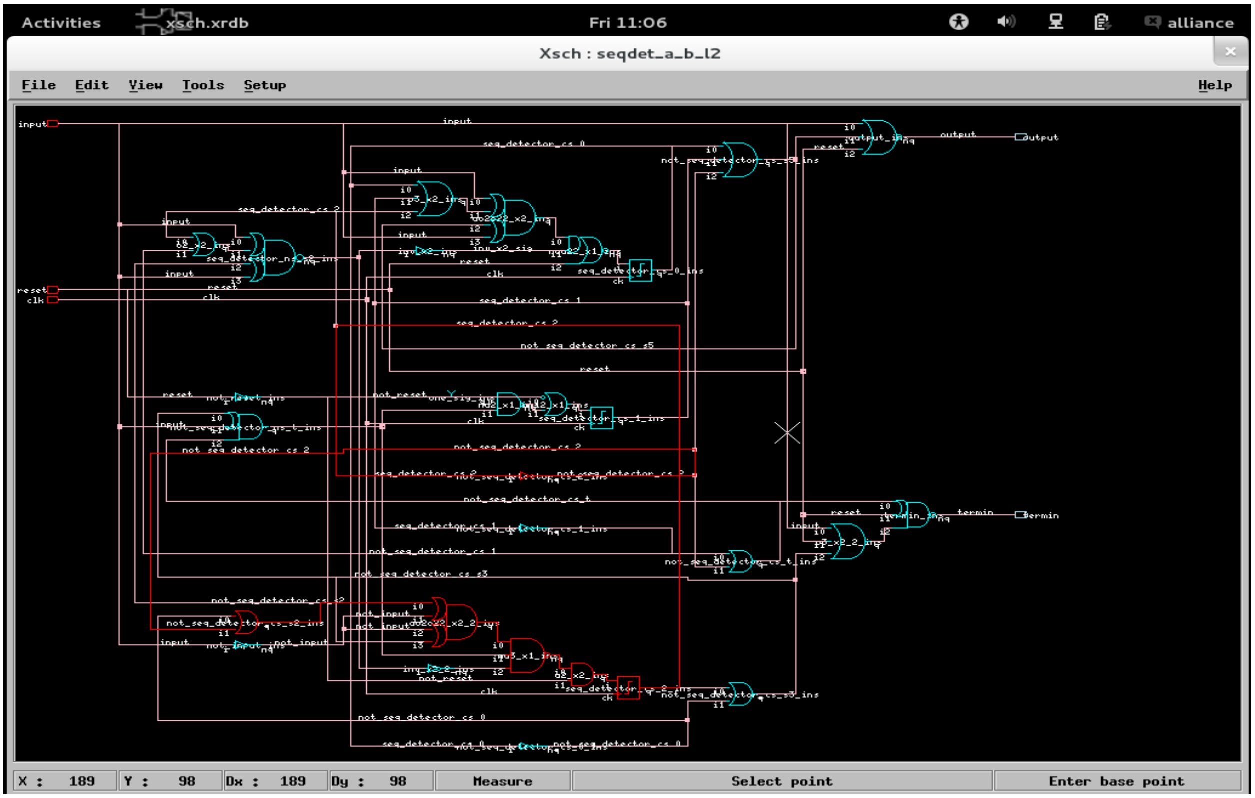
Encoding	-a	-j	-m	-o	-r
Delay Before	2256 ps	2453 ps	2460 ps	1253 ps	2444 ps
Delay After	2213 ps	2358 ps	2379 ps	1251 ps	2398 ps
Area Before	45750 lamda²	44250 lamda²	44250 lamda²	53500 lamda²	47500 lamda²
Area After	46250 lamda²	47500 lamda²	46500 lamda²	53500 lamda²	48750 lamda²

The chosen implementation is the -o one hot encoding implementation, due to its much lower delay, almost by half and while it has a larger area than the rest, it is not much greater.

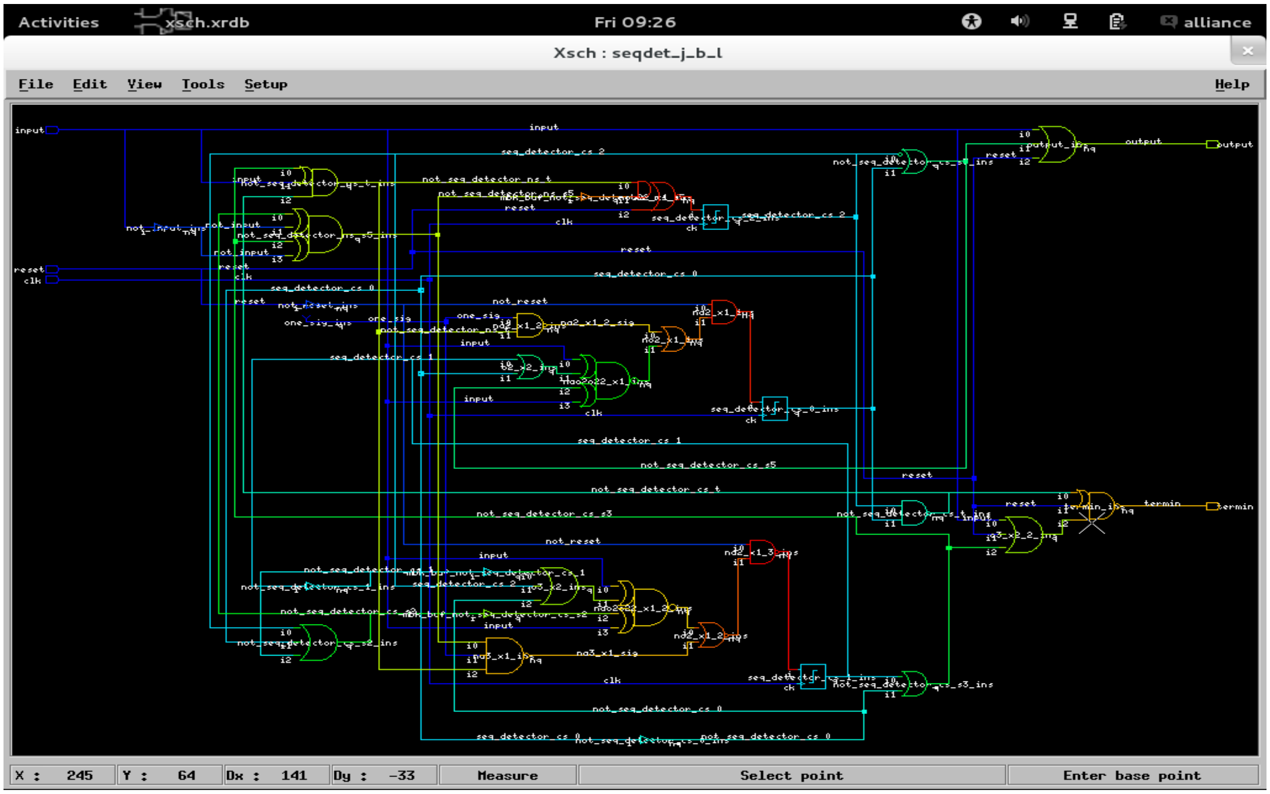
Netlist visualization

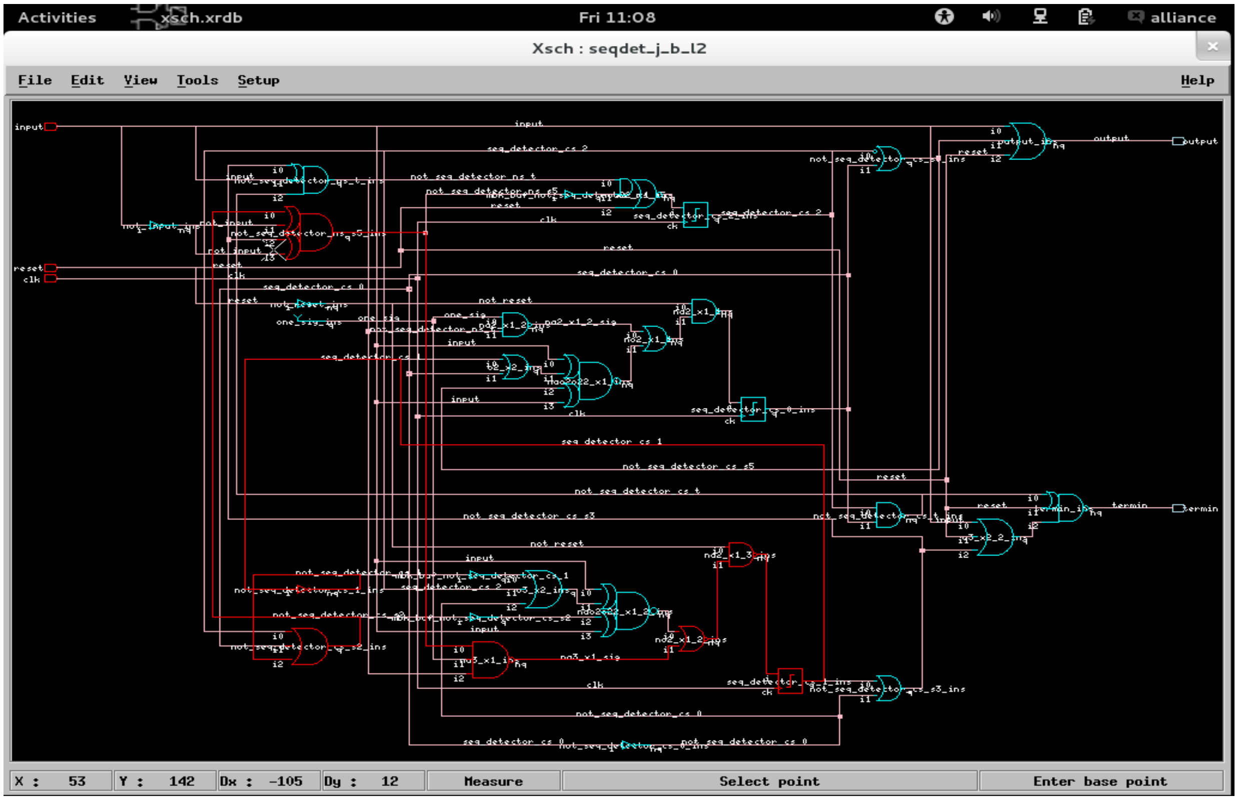
-a



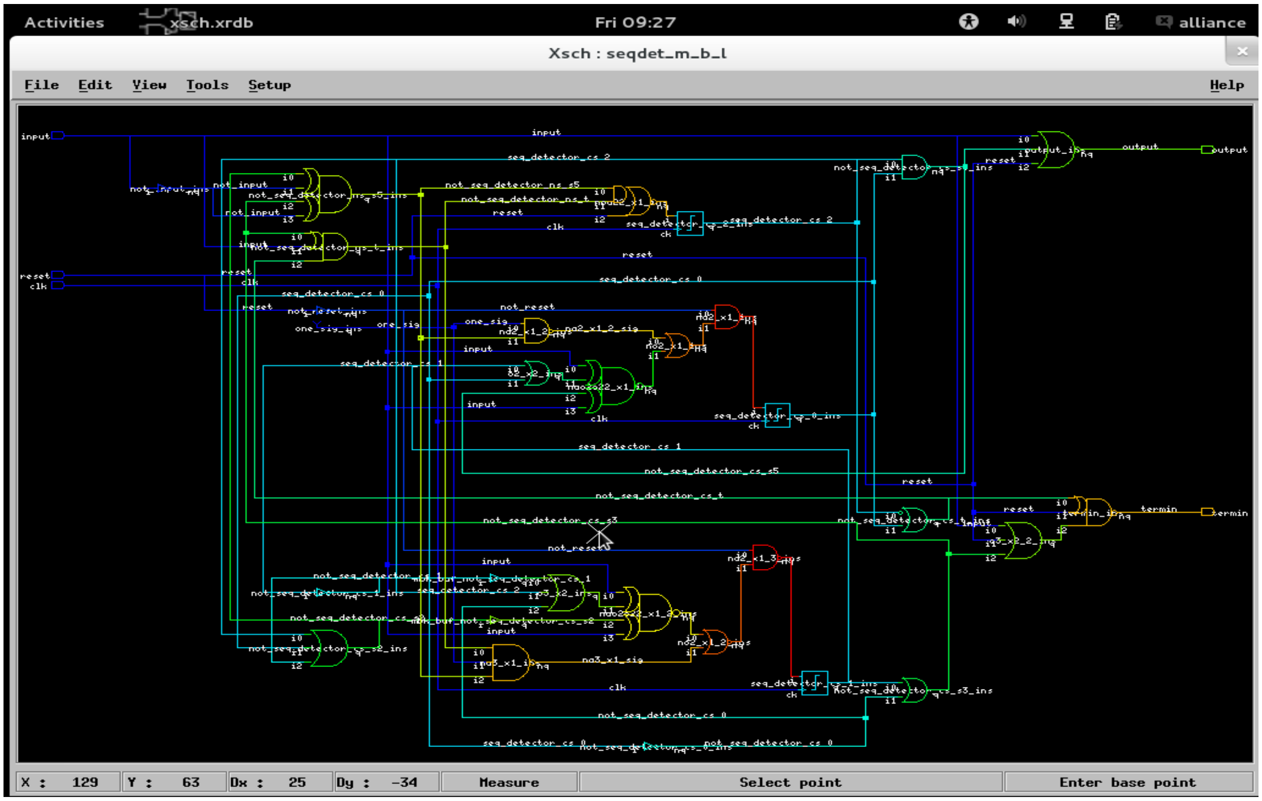


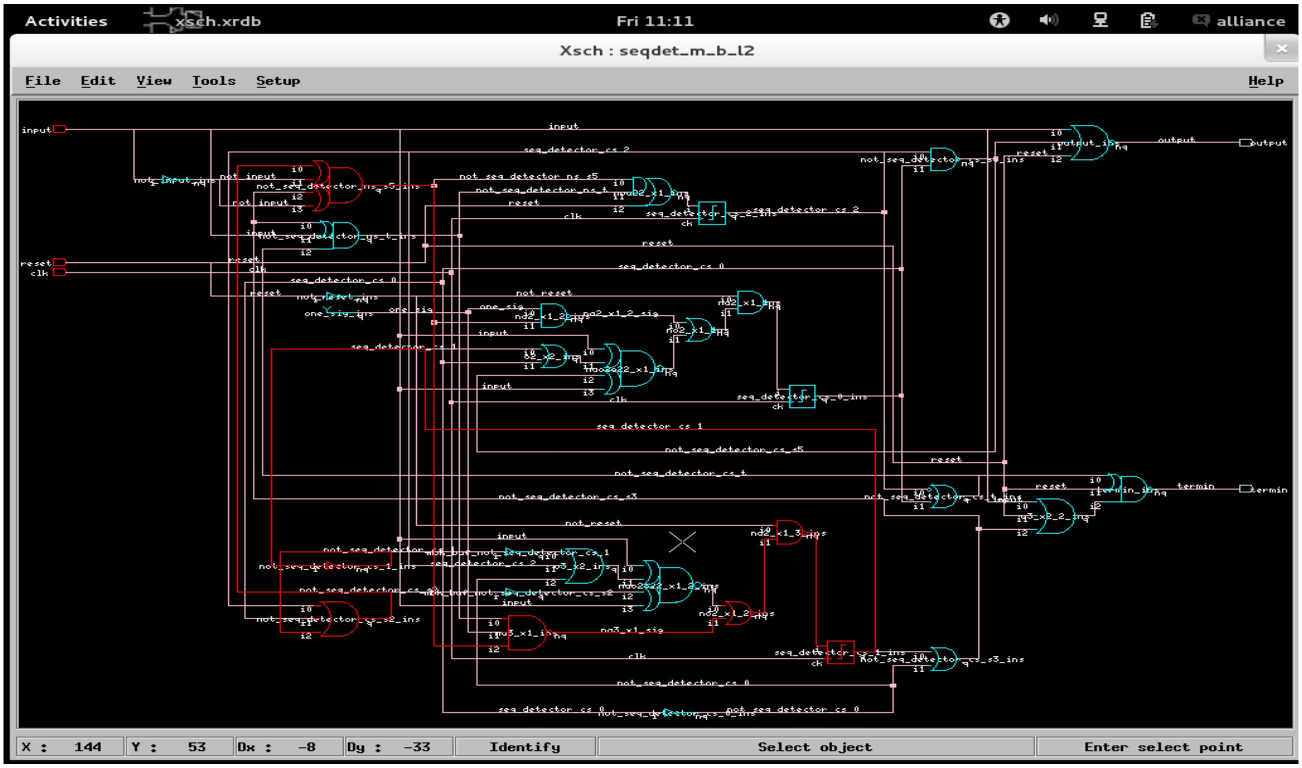
-j


Activities
xsch.xrdb
Fri 09:26
alliance
Xsch : seqdet_j_b_l
File Edit View Tools Setup
Help
input
reset
clk
output
termin
seq_detector_cs_2
seq_detector_cs_0
seq_detector_cs_1
not_seq_detector_cs_s5
not_seq_detector_cs_t
not_seq_detector_cs_s3
not_seq_detector_cs_0
not_reset
one_sig
X : 245
Y : 64
Dx : 141
Dy : -33
Measure
Select point
Enter base point

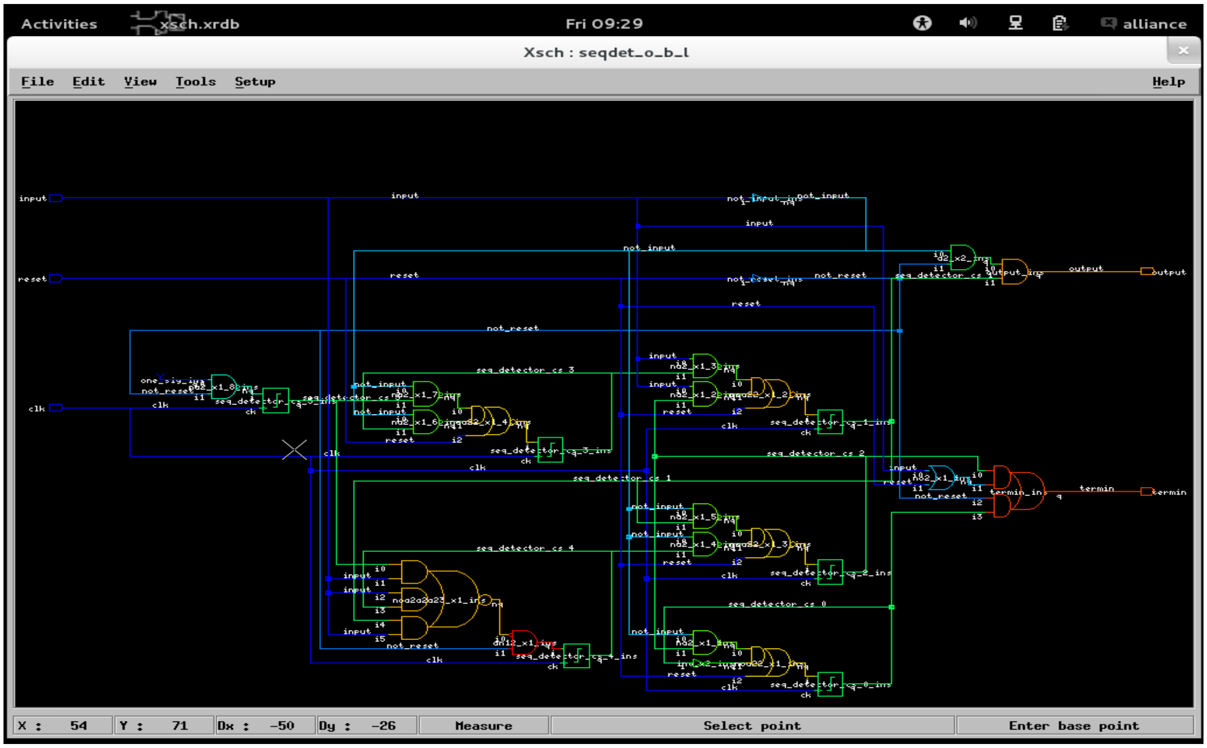

Activities
xsch.xrdb
Fri 11:08
alliance
Xsch : seqdet_j_b_l2
File Edit View Tools Setup
Help
input
reset
clk
output
termin
seq_detector_cs_2
seq_detector_cs_0
seq_detector_cs_1
not_seq_detector_cs_s5
not_seq_detector_cs_t
not_seq_detector_cs_s3
not_seq_detector_cs_0
not_reset
one_sig
X : 53
Y : 142
Dx : -105
Dy : 12
Measure
Select point
Enter base point

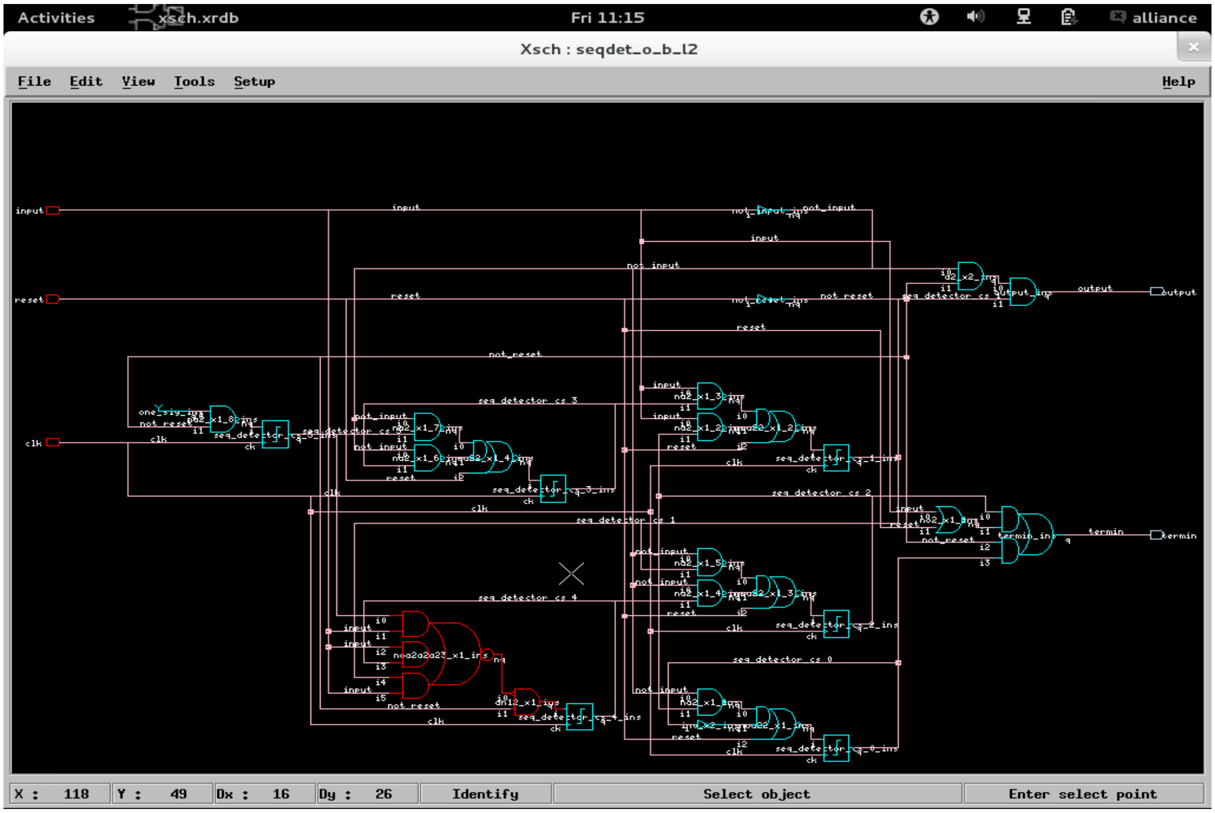
-m


Activities
xsch.xrdb
Fri 09:27
alliance
Xsch : seqdet_m_b_l
File
Edit
View
Tools
Setup
Help
input
output
reset
clk
termin
seq_detector_cs_2
not_seq_detector_ns_s5
not_seq_detector_ns_t
seq_detector_cs_0
seq_detector_cs_1
not_seq_detector_cs_s5
not_seq_detector_cs_t
not_seq_detector_cs_s3
not_reset
not_seq_detector_cs_0
X : 129
Y : 63
Dx : 25
Dy : -34
Measure
Select point
Enter base point

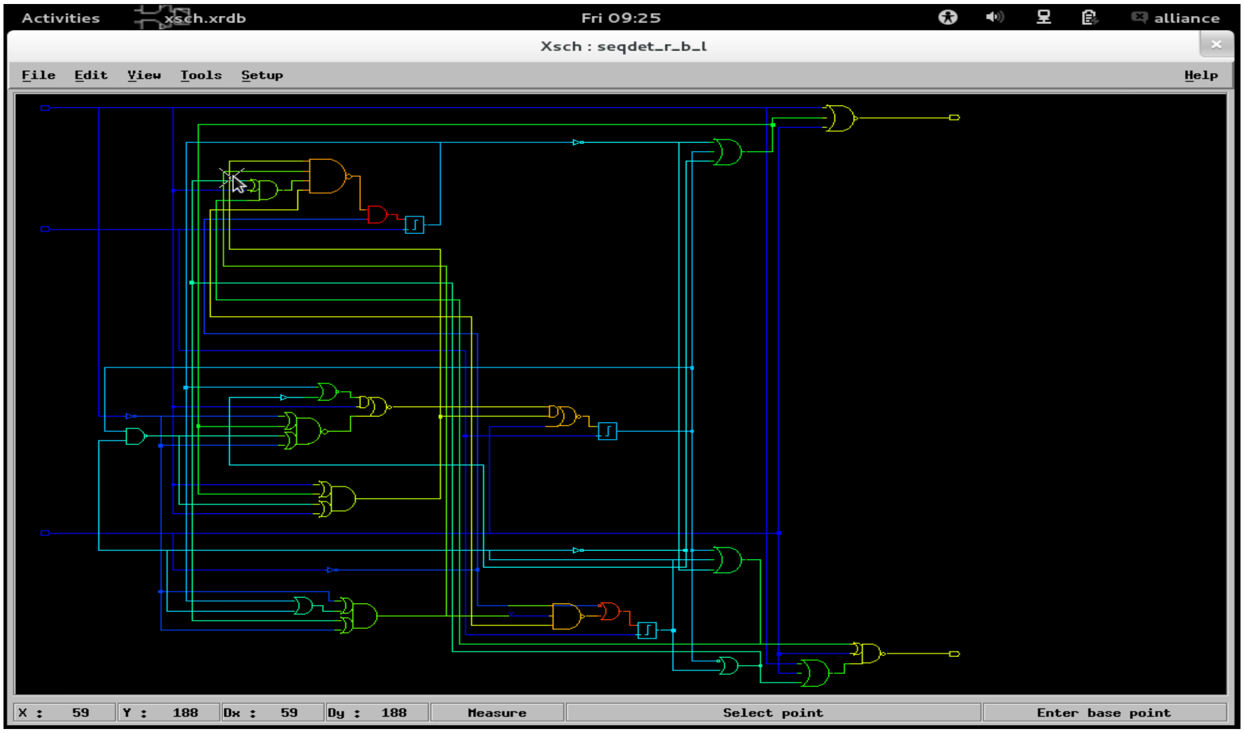

Activities
xsch.xrdb
Fri 11:11
alliance
Xsch : seqdet_m_b_l2
File
Edit
View
Tools
Setup
Help
input
output
reset
clk
termin
seq_detector_cs_2
not_seq_detector_ns_s5
not_seq_detector_ns_t
seq_detector_cs_0
seq_detector_cs_1
not_seq_detector_cs_s5
not_seq_detector_cs_t
not_seq_detector_cs_s3
not_reset
not_seq_detector_cs_0
X : 144
Y : 53
Dx : -8
Dy : -33
Identify
Select object
Enter select point

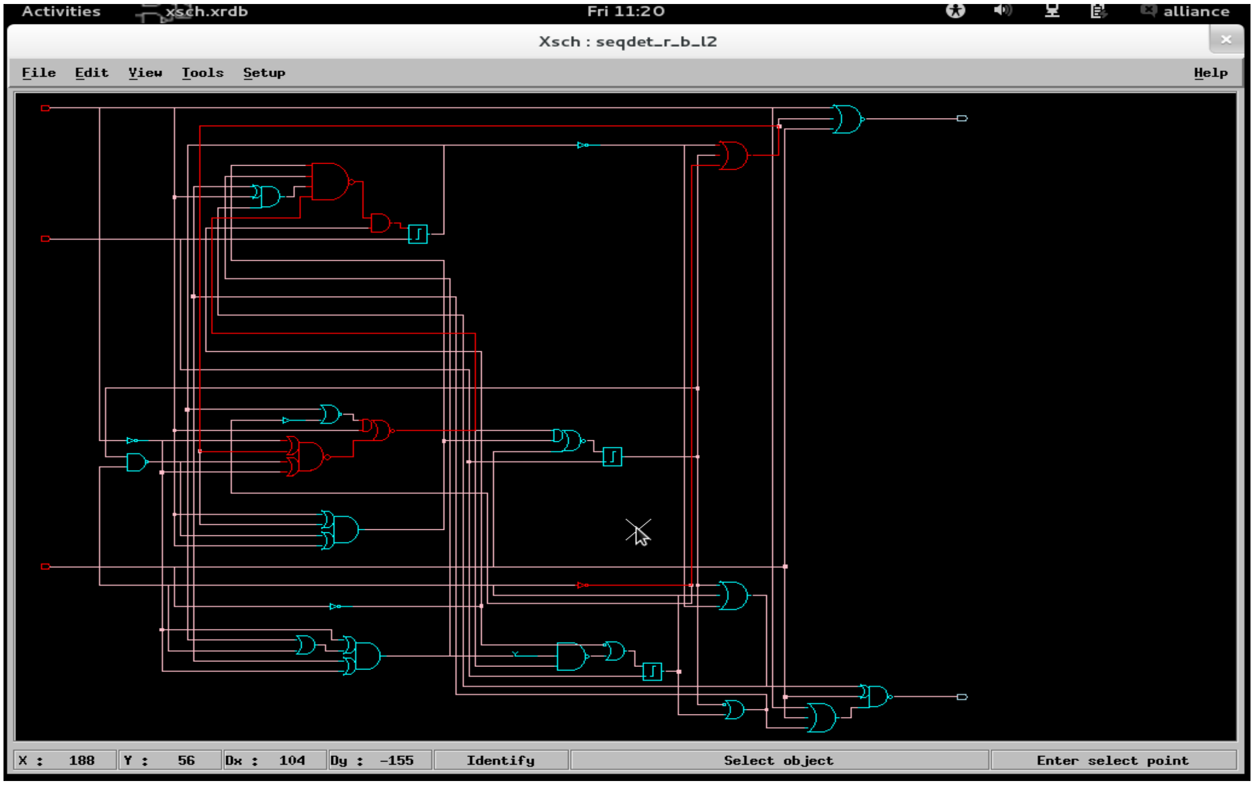
-o


Activities
xsch.xrdb
Fri 09:29
alliance
Xsch : seqdet_o_b_l
File
Edit
View
Tools
Setup
Help
input
reset
clk
not_input
not_reset
seq_detector_cs_0
seq_detector_cs_1
seq_detector_cs_2
seq_detector_cs_3
seq_detector_cs_4
output
termin
X : 54
Y : 71
Dx : -50
Dy : -26
Measure
Select point
Enter base point


Activities
xsch.xrdb
Fri 11:15
alliance
Xsch : seqdet_o_b_l2
File
Edit
View
Tools
Setup
Help
input
reset
clk
not_input
not_reset
seq_detector_cs_0
seq_detector_cs_1
seq_detector_cs_2
seq_detector_cs_3
seq_detector_cs_4
output
termin
X : 118
Y : 49
Dx : 16
Dy : 26
Identify
Select object
Enter select point

-r



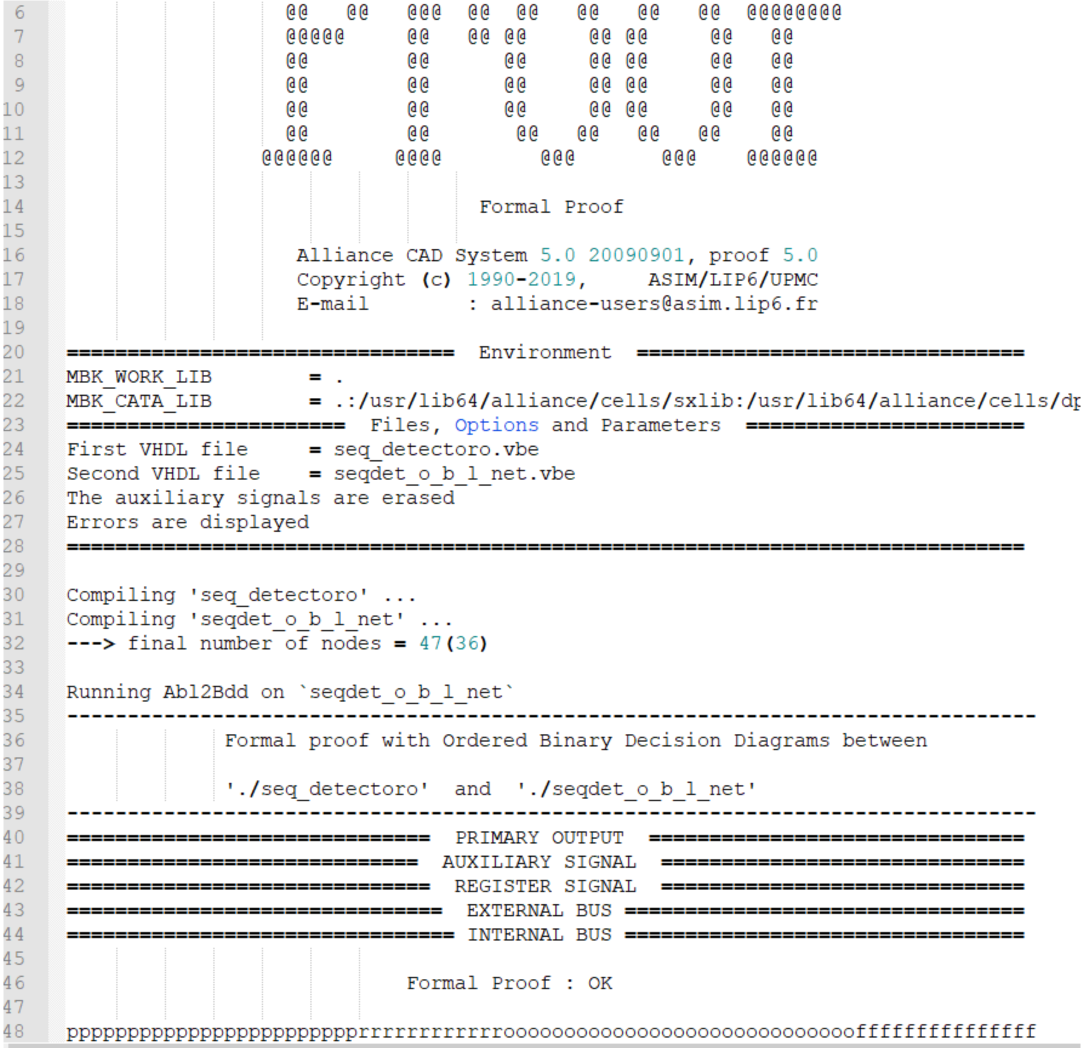


The critical path is always highlighted in red in the second visualization for all encodings.

Netlist checking

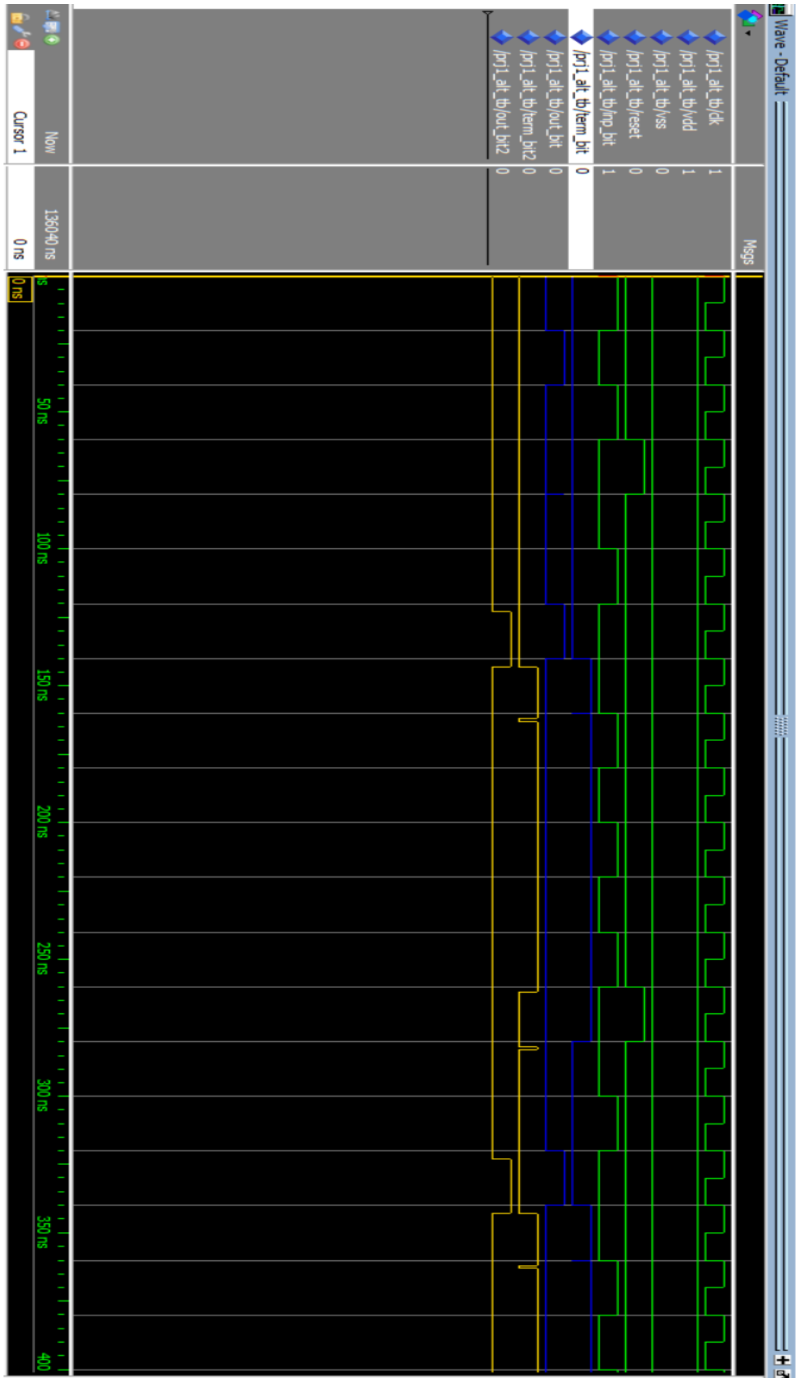
From this point we will only proceed with the chosen one hot encoding.

Proof output:



```
                  @@   @@   @@@  @@  @@   @@   @@   @@  @@@@@@@@
                  @@@@@     @@   @@ @@      @@ @@      @@   @@
                  @@        @@      @@      @@ @@      @@   @@
                  @@        @@      @@      @@ @@      @@   @@
                  @@        @@      @@      @@ @@      @@   @@
                  @@        @@       @@   @@   @@   @@     @@
                @@@@@@     @@@@        @@@       @@@    @@@@@@

                                  Formal Proof

                   Alliance CAD System 5.0 20090901, proof 5.0
                   Copyright (c) 1990-2019,     ASIM/LIP6/UPMC
                   E-mail        : alliance-users@asim.lip6.fr

================================  Environment  ================================
MBK_WORK_LIB        = .
MBK_CATA_LIB        = .:/usr/lib64/alliance/cells/sxlib:/usr/lib64/alliance/cells/dp
=======================  Files, Options and Parameters  =======================
First VHDL file     = seq_detectoro.vbe
Second VHDL file    = seqdet_o_b_l_net.vbe
The auxiliary signals are erased
Errors are displayed
================================================================================

Compiling 'seq_detectoro' ...
Compiling 'seqdet_o_b_l_net' ...
---> final number of nodes = 47(36)

Running Abl2Bdd on `seqdet_o_b_l_net`
---------------------------------------------------------------------------------
             Formal proof with Ordered Binary Decision Diagrams between

             './seq_detectoro'  and  './seqdet_o_b_l_net'
---------------------------------------------------------------------------------
==============================  PRIMARY OUTPUT  ===============================
=============================  AUXILIARY SIGNAL  ==============================
==============================  REGISTER SIGNAL  ==============================
===============================  EXTERNAL BUS =================================
================================ INTERNAL BUS =================================

                            Formal Proof : OK

pppppppppppppppppppppppprrrrrrrrrrrroooooooooooooooooooooooooooooffffffffffffff
```

Delay Simulation



Wave - Default

Msgs

/prj1_alt_tb/clk 1
/prj1_alt_tb/vdd 1
/prj1_alt_tb/vss 0
/prj1_alt_tb/reset 0
/prj1_alt_tb/inp_bit 1
/prj1_alt_tb/term_bit 0
/prj1_alt_tb/out_bit 0
/prj1_alt_tb/term_bit2 0
/prj1_alt_tb/out_bit2 0

Now 136040 ns
Cursor 1 0 ns

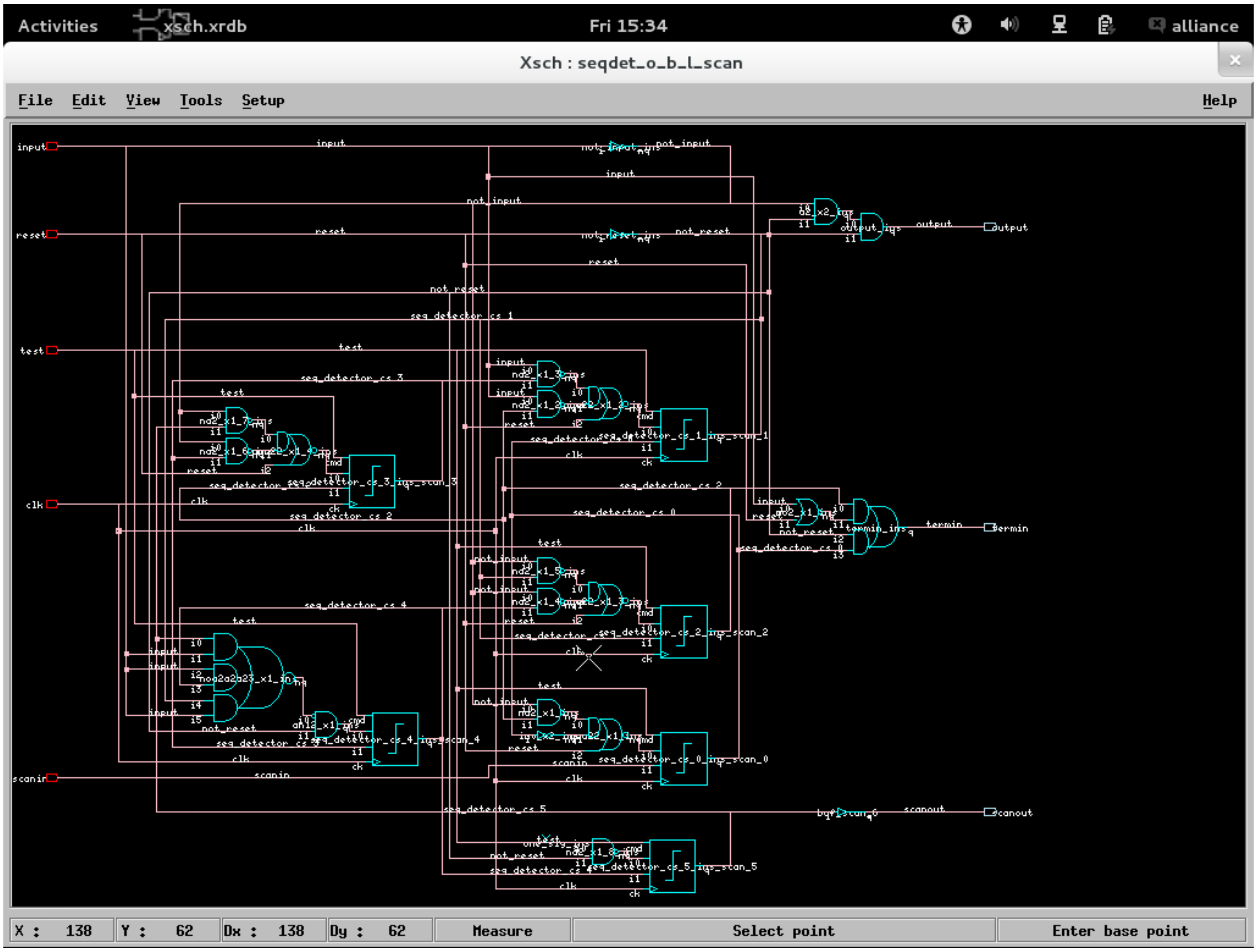
out_bit and term_bit correspond to the behavioral model (waveforms in blue) and out_bit2 and term_bit 2 correspond to the structural model (waveforms in yellow)

we can start seeing the delay starting at around 125ns as there is a noticeable difference between the 2 output bits. We can also notice the delay again at around 325ns.

Likewise, the termination bit signal from the structural model ends earlier than the behavioral model. Even the spikes are larger in the structural model.

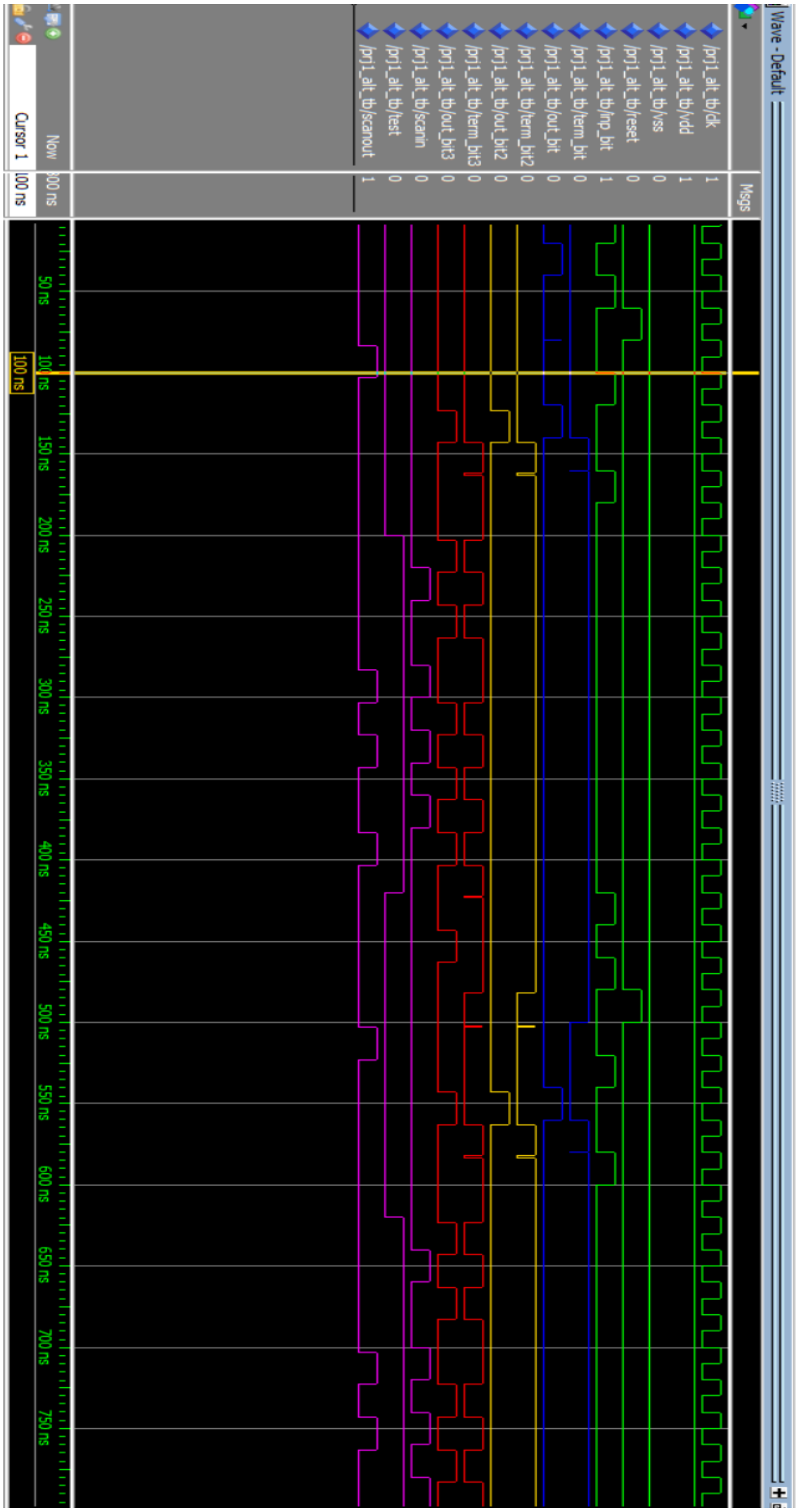
Scan-path insertion (DFT)

Netlist after Scan Path Insertion



X : 138 | Y : 62 | Dx : 138 | Dy : 62 | Measure | Select point | Enter base point

Scan-path Simulation



Wave - Default

	Msgs
/prj1_alt_tb/clk	1
/prj1_alt_tb/vdd	1
/prj1_alt_tb/vss	0
/prj1_alt_tb/reset	0
/prj1_alt_tb/inp_bit	1
/prj1_alt_tb/term_bit	0
/prj1_alt_tb/out_bit	0
/prj1_alt_tb/term_bit2	0
/prj1_alt_tb/out_bit2	0
/prj1_alt_tb/term_bit3	0
/prj1_alt_tb/out_bit3	0
/prj1_alt_tb/scanin	0
/prj1_alt_tb/test	0
/prj1_alt_tb/scanout	1
Now	00 ns
Cursor 1	100 ns

The output of the scan netlist corresponds with our expectations and aligns with the other outputs.

Appendix

MakeFile

```
#-------prj2-------------------------------------------#


all: seq_detectora.vbe \
    seq_detectorj.vbe \
    seq_detectorm.vbe \
    seq_detectoro.vbe \
    seq_detectorr.vbe
             @echo "<-- Generated"


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


vhd_to_fsm:
         rename .vhd .fsm *.vhd


seq_detectora.vbe: seq_detector.fsm
         @echo "    Encoding Synthesis -> prj2_a.vbe"
         syf -CEV -a seq_detector


seq_detectorj.vbe: seq_detector.fsm
         @echo "    Encoding Synthesis  -> sdetj.vbe"
         syf -CEV -j seq_detector
```

```
seq_detectorm.vbe: seq_detector.fsm
	@echo "    Encoding Synthesis  -> sdetm.vbe"
	syf -CEV -m seq_detector

seq_detectoro.vbe: seq_detector.fsm
	@echo "    Encoding Synthesis  -> sdeto.vbe"
	syf -CEV -o seq_detector

seq_detectorr.vbe: seq_detector.fsm
	@echo "    Encoding Synthesis  -> sdetr.vbe"
	syf -CEV -r seq_detector

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

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

#-------------------Boom--------------------------------#

boom_all: seqdet_a_b.vbe \
	seqdet_j_b.vbe \
	seqdet_m_b.vbe \
	seqdet_o_b.vbe \
	seqdet_r_b.vbe



seqdet_a_b.vbe: seq_detectora.vbe
	boom -V -d 50 seq_detectora seqdet_a_b > seqdet_a_boom.out

seqdet_j_b.vbe: seq_detectorj.vbe
	boom -V -d 50 seq_detectorj seqdet_j_b > seqdet_j_boom.out

seqdet_m_b.vbe: seq_detectorm.vbe
	boom -V -d 50 seq_detectorm seqdet_m_b > seqdet_m_boom.out

seqdet_o_b.vbe: seq_detectoro.vbe
	boom -V -d 50 seq_detectoro seqdet_o_b > seqdet_o_boom.out

seqdet_r_b.vbe: seq_detectorr.vbe
	boom -V -d 50 seq_detectorr seqdet_r_b > seqdet_r_boom.out

#-----------------------------boog--------------------------#
boog_all: seqdet_a_b.vst \
	seqdet_j_b.vst \
	seqdet_m_b.vst \
	seqdet_o_b.vst \
	seqdet_r_b.vst

seqdet_a_b.vst : seq_detectora.vbe paramfile.lax
	boog -x 1 -l paramfile seq_detectora > seqdet_a_boog.out

seqdet_j_b.vst : seq_detectorj.vbe paramfile.lax
	boog -x 1 -l paramfile seq_detectorj > seqdet_j_boog.out

seqdet_m_b.vst : seq_detectorm.vbe paramfile.lax
	boog -x 1 -l paramfile seq_detectorm > seqdet_m_boog.out

seqdet_o_b.vst : seq_detectoro.vbe paramfile.lax
	boog -x 1 -l paramfile seq_detectoro > seqdet_o_boog.out

seqdet_r_b.vst : seq_detectorr.vbe paramfile.lax
	boog -x 1 -l paramfile seq_detectorr > seqdet_r_boog.out

#---------------------loon---------------------------------#
loon_all: seqdet_a_b_l.vst \
	seqdet_j_b_l.vst \
	seqdet_m_b_l.vst \
	seqdet_o_b_l.vst \
	seqdet_r_b_l.vst
```

```
seqdet_a_b_l.vst: seq_detectora.vst paramfile.lax

        loon -x 1 seq_detectora seqdet_a_b_l paramfile >
seqdet_a_bl_loon.out


seqdet_j_b_l.vst: seq_detectorj.vst paramfile.lax

        loon -x 1 seq_detectorj seqdet_j_b_l paramfile >
seqdet_j_bl_loon.out


seqdet_m_b_l.vst: seq_detectorm.vst paramfile.lax

        loon -x 1 seq_detectorm seqdet_m_b_l paramfile >
seqdet_m_bl_loon.out


seqdet_o_b_l.vst: seq_detectoro.vst paramfile.lax

        loon -x 1 seq_detectoro seqdet_o_b_l paramfile >
seqdet_o_bl_loon.out


seqdet_r_b_l.vst: seq_detectorr.vst paramfile.lax

        loon -x 1 seq_detectorr seqdet_r_b_l paramfile >
seqdet_r_bl_loon.out




#---------------------loon2--------------------------------#

loon2_all: seqdet_a_b_l2.vst \
	seqdet_j_b_l2.vst \
	seqdet_m_b_l2.vst \
	seqdet_o_b_l2.vst \
	seqdet_r_b_l2.vst



seqdet_a_b_l2.vst: seq_detectora.vst paramfile.lax
	loon -x 0 seq_detectora seqdet_a_b_l2 paramfile > seqdet_a_bl2_loon.out

seqdet_j_b_l2.vst: seq_detectorj.vst paramfile.lax
	loon -x 0 seq_detectorj seqdet_j_b_l2 paramfile > seqdet_j_bl2_loon.out

seqdet_m_b_l2.vst: seq_detectorm.vst paramfile.lax
	loon -x 0 seq_detectorm seqdet_m_b_l2 paramfile > seqdet_m_bl2_loon.out

seqdet_o_b_l2.vst: seq_detectoro.vst paramfile.lax
	loon -x 0 seq_detectoro seqdet_o_b_l2 paramfile > seqdet_o_bl2_loon.out

seqdet_r_b_l2.vst: seq_detectorr.vst paramfile.lax
	loon -x 0 seq_detectorr seqdet_r_b_l2 paramfile > seqdet_r_bl2_loon.out
```

```
#-----------------flatbeh and proof---------------------------#

seqdet_o_b_l_net.vbe: seqdet_o_b_l.vst seq_detectoro.vbe
	flatbeh seqdet_o_b_l seqdet_o_b_l_net > seqdet_o_b_l_net_flatbeh.out
	proof -d seq_detectoro seqdet_o_b_l_net > seqdet_o_proof.out
```

Path file

```
BEGIN_PATH_REG
seq_detector_cs_0_ins
seq_detector_cs_1_ins
seq_detector_cs_2_ins
seq_detector_cs_3_ins
seq_detector_cs_4_ins
seq_detector_cs_5_ins
END_PATH_REG

BEGIN_CONNECTOR
SCAN_IN  scanin
SCAN_OUT scanout
SCAN_TEST test
END_CONNECTOR
```

References

- Lecture slides