

An STA CAD Tool for SET Generation and Propagation

**CAS (Circuits and Systems) Lab,
University of Thessaly, Greece.**

Who we are – CAS Lab @ UTH



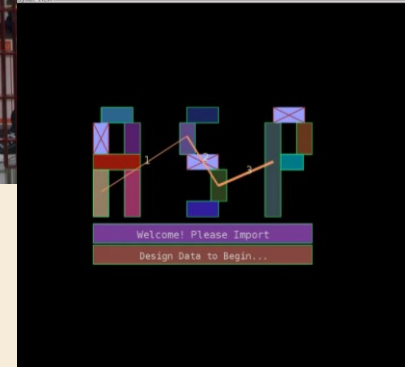
Volos, Thessaly,
Greece



University of
Thessaly



Our Lab



EDA R&D

CONFIDENTIAL

CAS Lab EDA Technologies

**Static Timing Analysis
(STA)**



**Software
Development and
Deployment**



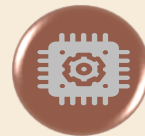
Detailed Placement



**RADiation
HARDening**



**Asynchronous Logic
Synthesis
(and Security)**



3D Integration



Legalisation



Global Placement



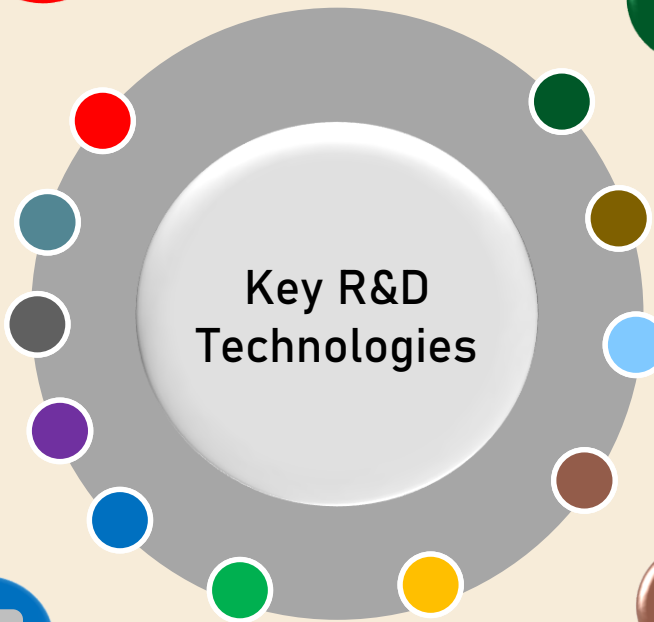
Floorplanning

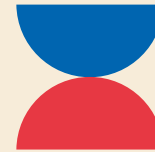
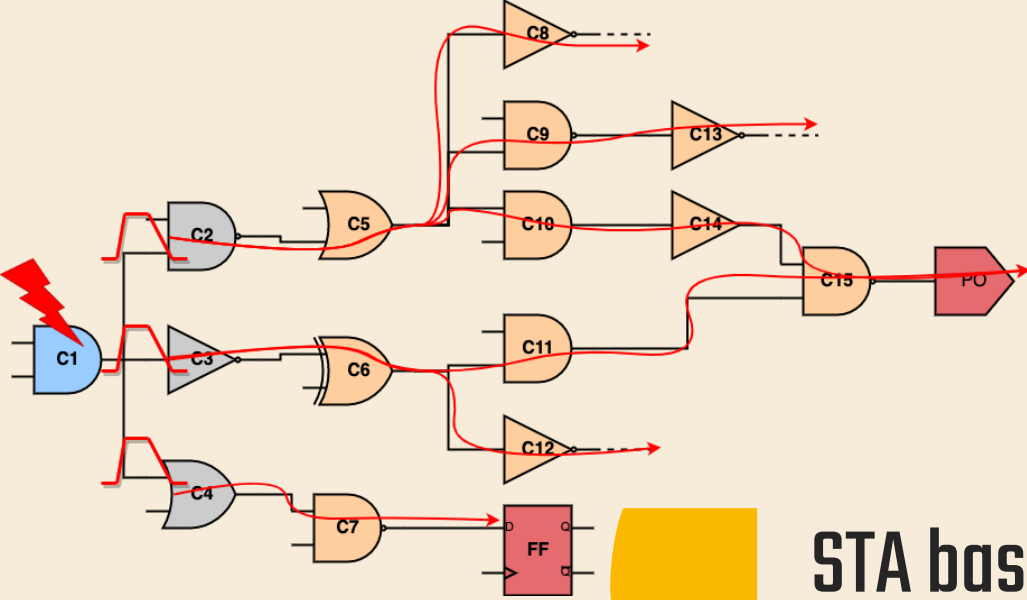


**Structured
Datapath
Placement**



Asynchronous STA





STA based SET GENERATION and PROPAGATION

Technology Overview

What we do

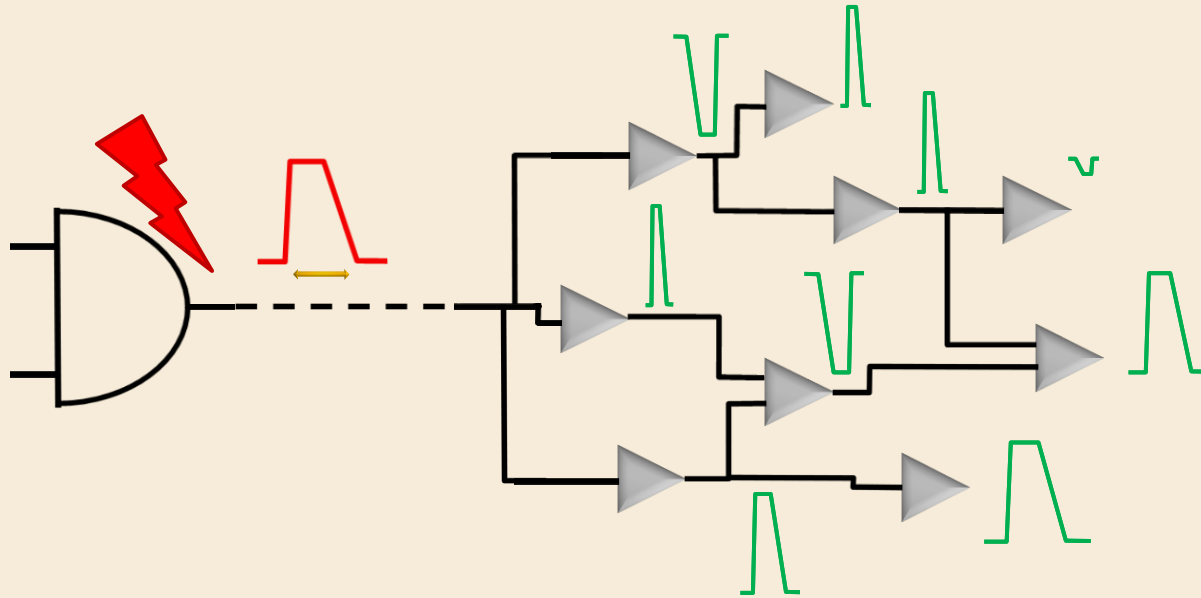
- ☐ SET Generation and Propagation
- ☐ Using State of the Art Static Timing Analysis (STA)
- ☐ SPICE accuracy, with interfaces to SPICE for error calculation at the path or logic cone level
- ☐ Our toolset includes an embedded STA Engine
- ☐ Accurate RC Wire Delay Modelling
- ☐ Can identify the worst FF, with respect to SET pulse width
- ☐ Supports multiple SET Generation current models, e.g. Double-Exp
- ☐ Supports internal Case Analysis conditions for SET Pulse Propagation

- ☐ Mitigation part → perform IPO operations to improve SET Propagation → WIP

What we don't do

- ☐ NOT transistor/electrical level
- ☐ NOT device level
- ☐ NO Special Library Characterization required, only user-provided Parameters

The Problem - SETs



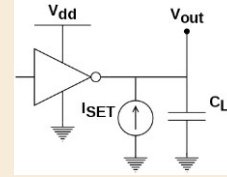
STA-based RADHARD Flow at a glance

.lib, .lef,
.SPEF,
.Verilog, .SDC
files

- Load Design Netlist and Physical Representation into Tool

Perform STA

- Analyse Design Timing

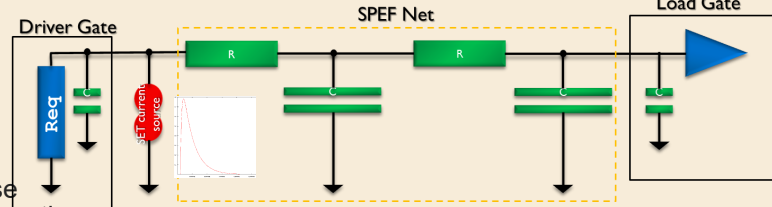


SET current model (DEXP), SET location, side inputs
→ Scenario File

- User-input for SET Generation

Run SET Generation

- Pulse Generation at SET location



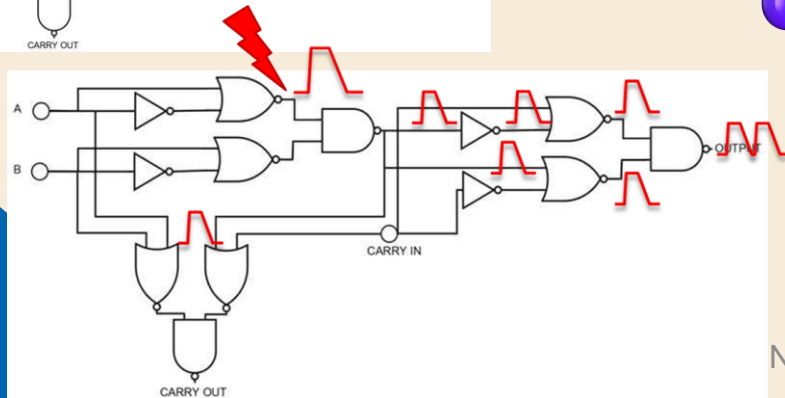
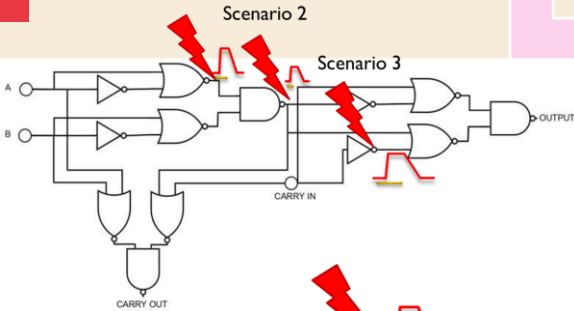
- Propagation at Endpoints (FFs, POs)

Run SET Propagation

Measure Error against SPICE/SPECTRE

SPICE out

MEASUREMENTS in

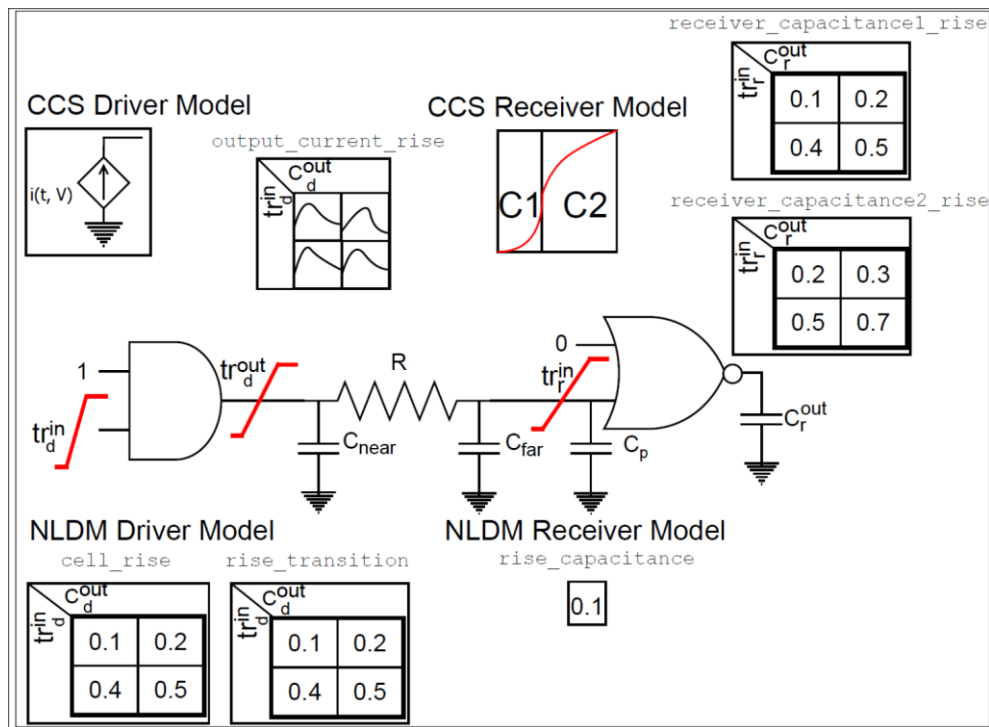


CONFIDENTIAL

CCS Timing Model

CCS Timing

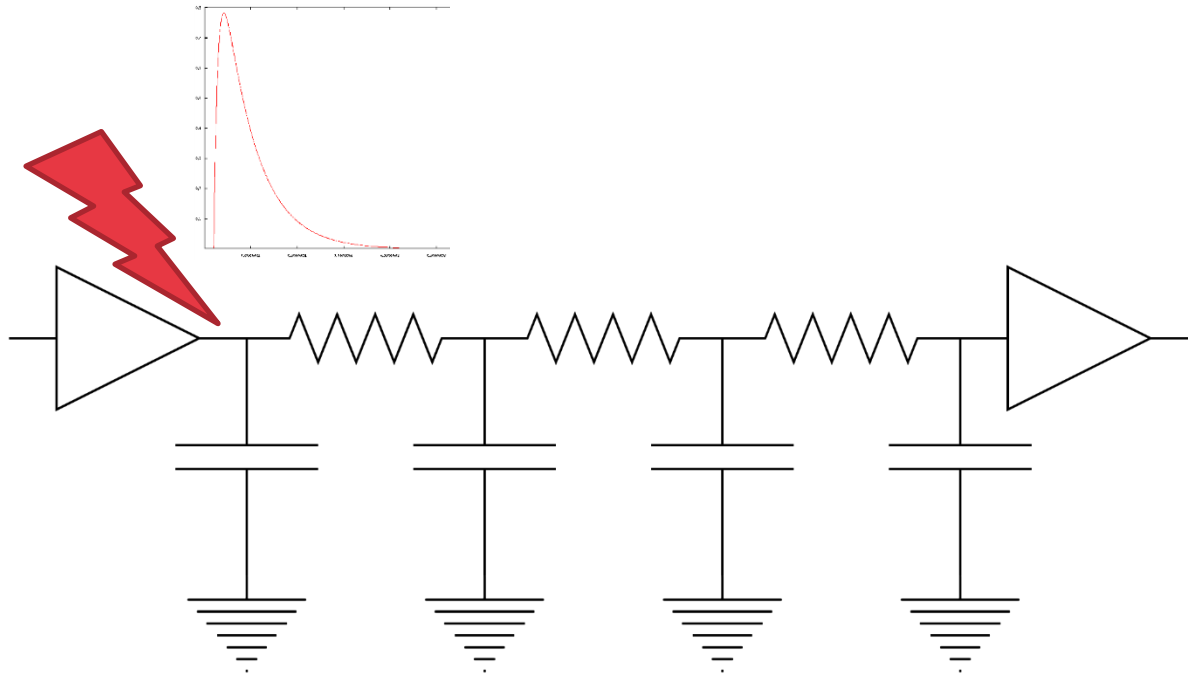
- **Highly Accurate RC Delays;**
key for accurate pulse propagation through RC Networks
- **Current profiles of CCS gate responses may be used to extract standard-cell information, e.g. R_{eff} , V_{out} , without SPICE or Characterisation**
- **Good for First/Second-Order SET Generation Approximations**



Set Generation Example

Set Generation Example

- ❑ User provides SET parameters, location, side inputs

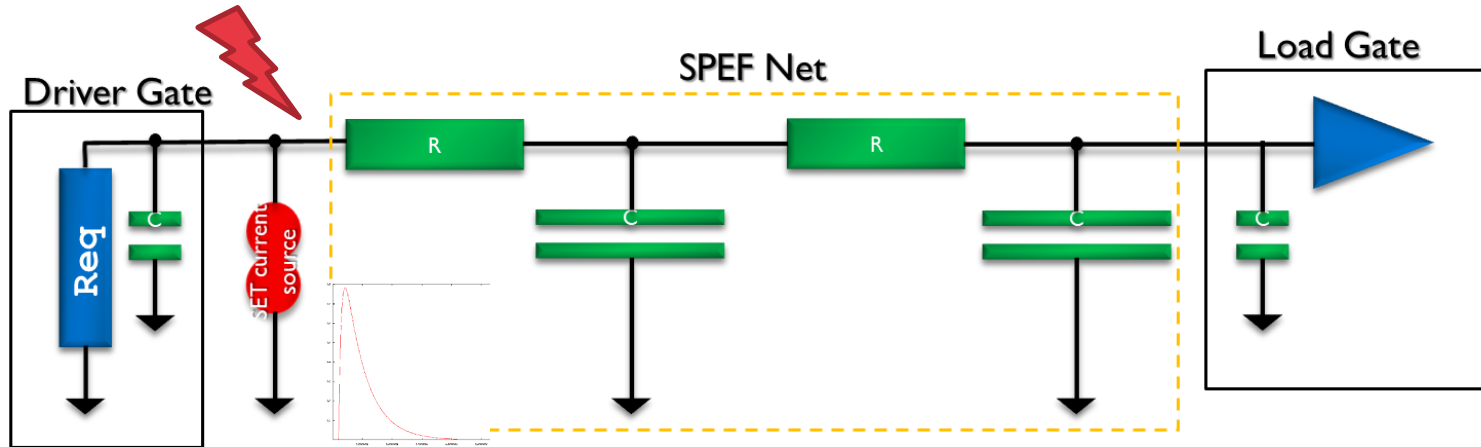


Set Generation Example

Set Generation Example

- ❑ User provides SET parameters, location, side inputs
- ❑ With an annotated SPEF – **this is a First/Second Order Approximation Model**
- ❑ Pulse has two parameters, **slew**, **delay (AT)**
- ❑ Delay is MORE critical as (i) **slew will be regenerated**, (ii) **width $\leftrightarrow$ delay**

SET Generation Model



Set Generation Example

Set Generation Example

❑ ASP Tool Screenshot

driver_pin

endpoint

Tdelay, tau1,
tau2, Q
current parameters

```
generate_SET_glitch -scenario ASU_2bufs/U1/Y ASU_2bufs/U2/A 0.1n 47p 347p 700f -corner 0
```

```
DEBUG: vector1Req 0.025548, vector1slew 0.005000, vector1outputcapacitance 0.005760
DEBUG: vector2Req 0.121528, vector2slew 0.005000, vector2outputcapacitance 0.011520
DEBUG: vector3Req 0.414434, vector3slew 0.010000, vector3outputcapacitance 0.005760
DEBUG: vector4Req 0.183335, vector4slew 0.010000, vector4outputcapacitance 0.011520
DEBUG: Maximum PullDown Resistance: 0.530368 KOHms
DEBUG: Probe Edge Lower Time 1.03884e-10
DEBUG: Probe Edge Delay Time 1.22646e-10
DEBUG: Probe Edge Upper Time 1.62254e-10
DEBUG: Stored Probe Rise Edge (0) Slew 5.83705e-11 and Arrival 1.22646e-10
DEBUG: Probe Edge Upper Time 2.84014e-10
DEBUG: Probe Edge Delay Time 4.9978e-10
DEBUG: Probe Edge Lower Time 1.05848e-09
DEBUG: Stored Probe Fall Edge (0) Slew -7.74464e-10 and Arrival 4.9978e-10
```

Rise Edges:

0. Slew: 0.052236, Arrival: 0.122646

Fall Edges:

0. Slew: 0.693077, Arrival: 0.499780

```
[73884ece3f7593b47b1ab3d8ab7ef250e93edf5e]%
```

CONFIDENTIAL

Set Generation Example

Set Generation Example

ASP Tool Screenshot

driver_pin

endpoint

Tdelay, tau1,
tau2, Q
current parameters

```
generate_SET_glitch -scenario ASU_2bufs/U1/Y ASU_2bufs/U2/A 0.1n 47p 347p 700f -corner 0
```

```
DEBUG: vector1Req 0.325548, vector1slew 0.005000, vector1outputcapacitance 0.005760
DEBUG: vector2Req 0.121528, vector2slew 0.005000, vector2outputcapacitance 0.011520
DEBUG: vector3Req 0.414434, vector3slew 0.010000, vector3outputcapacitance 0.005760
DEBUG: vector4Req 0.183335, vector4slew 0.010000, vector4outputcapacitance 0.011520
DEBUG: Maximum PullDown Resistance: 0.530368 Kohms
DEBUG: Probe Edge Lower Time 1.03884e-10
DEBUG: Probe Edge Delay Time 1.22646e-10
DEBUG: Probe Edge Upper Time 1.62254e-10
DEBUG: Stored Probe Rise Edge (0) Slew 5.83705e-11 and Arrival 1.22646e-10
DEBUG: Probe Edge Upper Time 2.84014e-10
DEBUG: Probe Edge Delay Time 4.9978e-10
DEBUG: Probe Edge Lower Time 1.05848e-09
DEBUG: Stored Probe Fall Edge (0) Slew -7.77777e-11 and Arrival 4.9978e-10
```

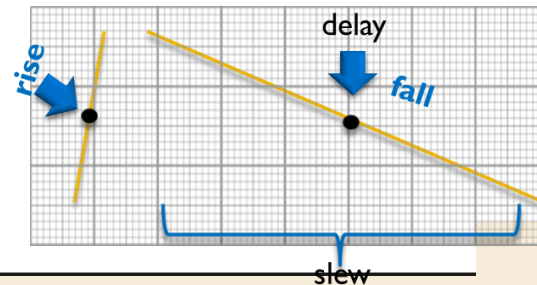
Rise Edges:

0. Slew: 0.052236, Arrival: 0.122646

Fall Edges:

0. Slew: 0.693077, Arrival: 0.499780

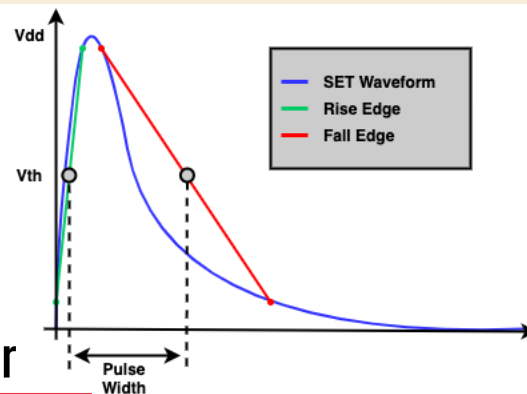
Endpoint
Measurements



```
[73884ece3f7593b47b1ab3d8ab7ef250e93edf5e]%>
```

CONFIDENTIAL

SET Generation Results VS. SPICE



Req Error Analysis

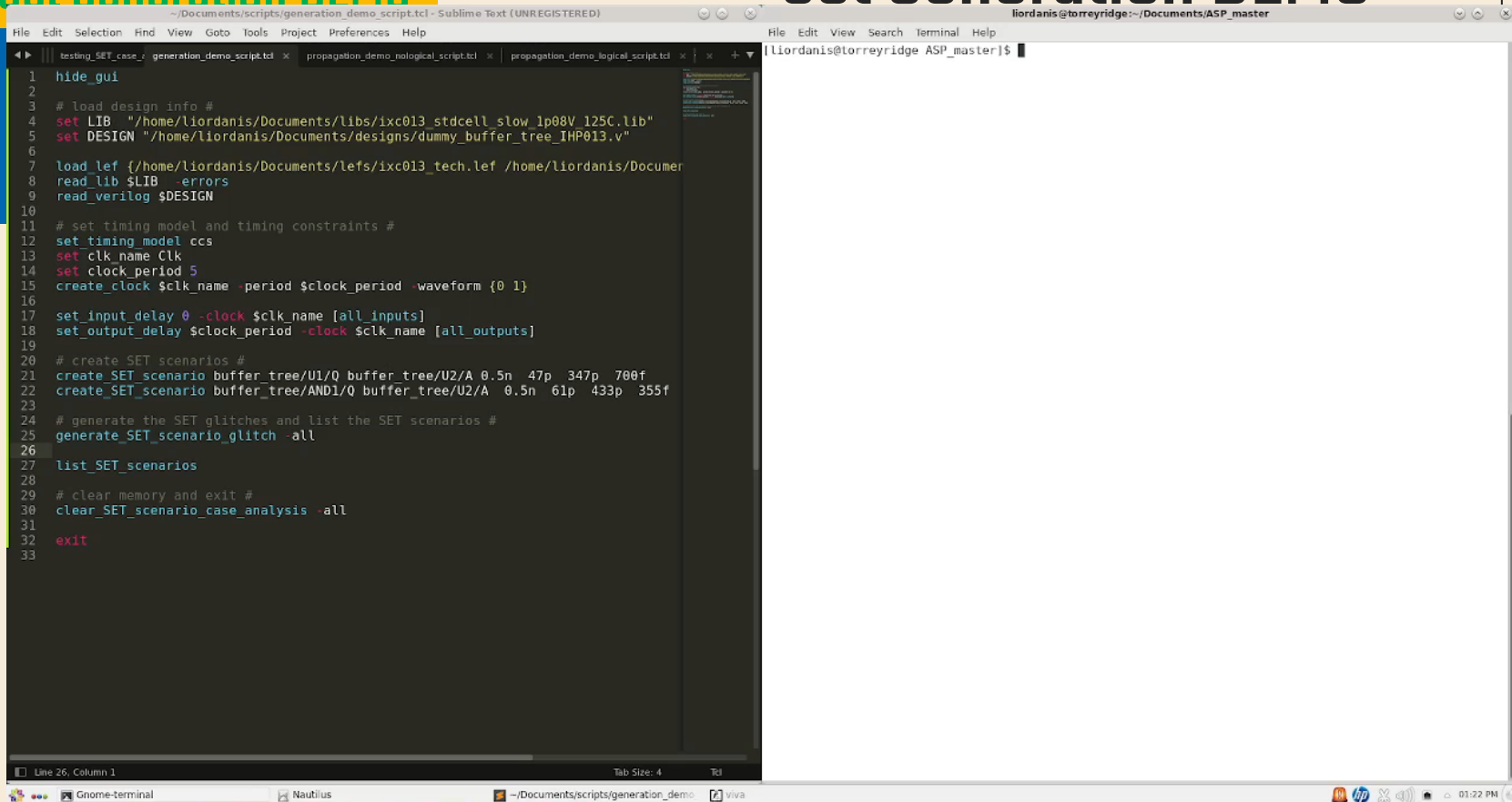
Vdd as Reference Signals (2394 Req Sweep Experiments)	Average error (- = pessimistic, + = optimistic)
Fall Arrival	-38.41%
Fall Slew	-94.63%
Rise Arrival	-5.7%
Rise Slew	33.5%

Best Req Error

Full Swing Signals	Error vs. SPICE (minimum)
Fall Arrival	9.28e-04%
Fall Slew	-0.144%
Rise Arrival	-7.24e-04%
Rise Slew	0.07%

Set Generation DEMO

Set Generation DEMO



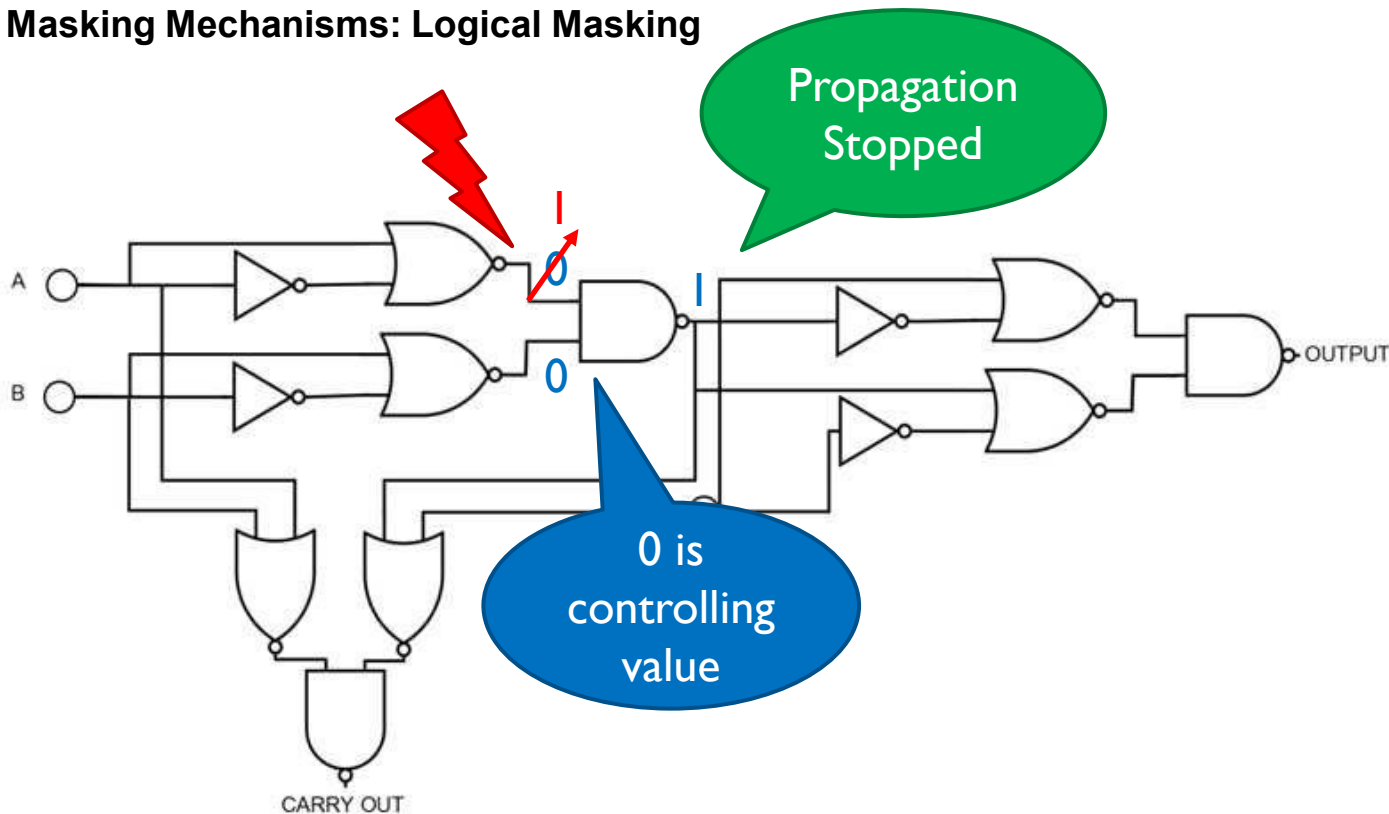
```
File Edit Selection Find View Goto Tools Project Preferences Help
testing_SET_case_ generation_demo_script.tcl x propagation_demo_logical_script.tcl x x +
1 hide_gui
2
3 # load design info #
4 set LIB "/home/liordanis/Documents/libs/ixc013_stdcell_slow_1p08V_125C.lib"
5 set DESIGN "/home/liordanis/Documents/designs/dummy_buffer_tree_IHP013.v"
6
7 load_lef {/home/liordanis/Documents/lefs/ixc013_tech.lef /home/liordanis/Documen
8 read_lib $LIB -errors
9 read_verilog $DESIGN
10
11 # set timing model and timing constraints #
12 set_timing_model ccs
13 set_clk_name Clk
14 set_clock_period 5
15 create_clock $clk_name period $clock_period waveform {0 1}
16
17 set_input_delay 0 -clock $clk name [all inputs]
18 set_output_delay $clock_period -clock $clk_name [all outputs]
19
20 # create SET scenarios #
21 create_SET_scenario buffer_tree/U1/Q buffer_tree/U2/A 0.5n 47p 347p 780f
22 create_SET_scenario buffer_tree/AND1/Q buffer_tree/U2/A 0.5n 61p 433p 355f
23
24 # generate the SET glitches and list the SET scenarios #
25 generate_SET_scenario_glitch -all
26
27 list_SET_scenarios
28
29 # clear memory and exit #
30 clear_SET_scenario_case_analysis -all
31
32 exit
33
```

liordanis@torreyridge ASP_master\$

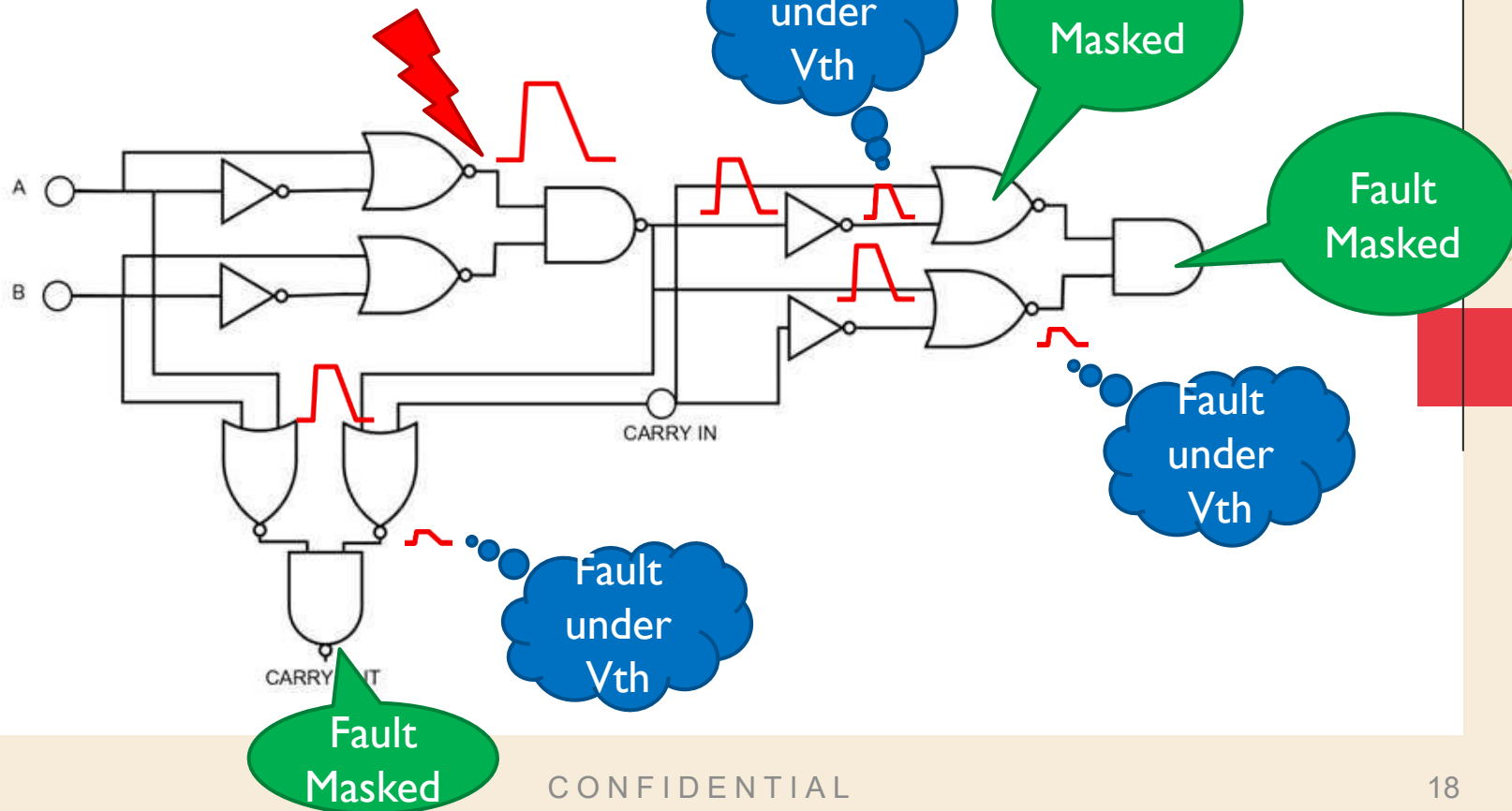
Line 26, Column 1 Tab Size: 4

Gnome-terminal Nautilus ~/Documents/scripts/generation_demo viva 01:22 PM

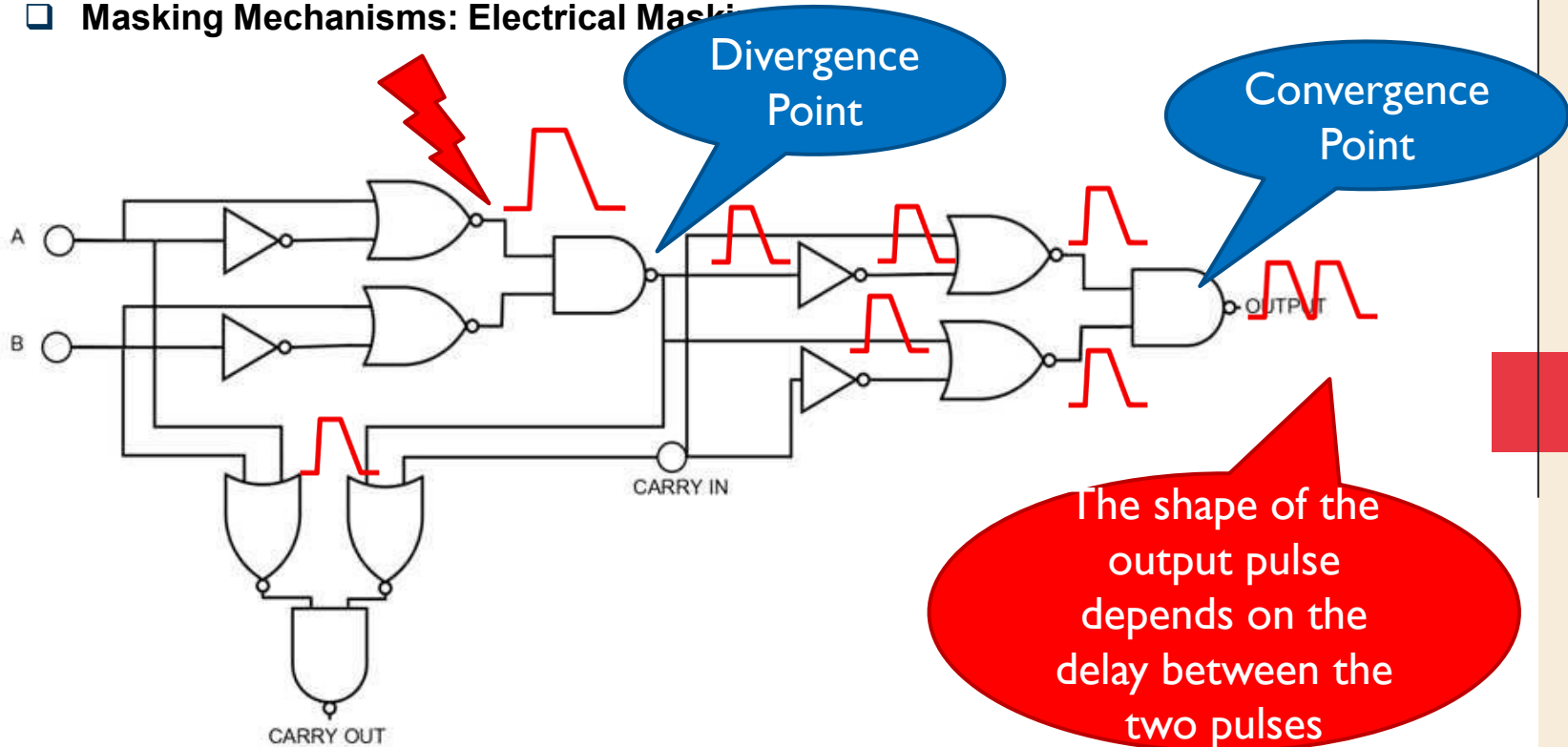
Masking Mechanisms: Logical Masking



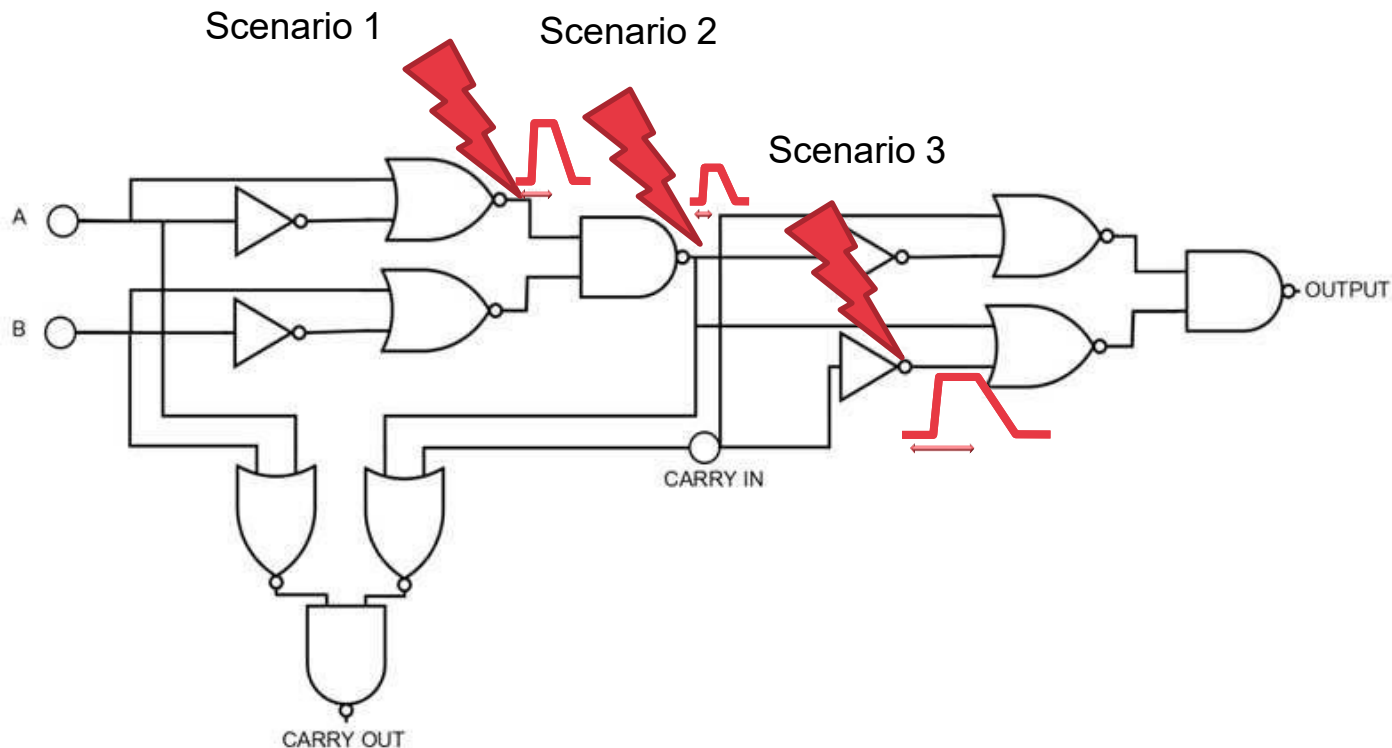
Masking Mechanisms: Electrical Masking



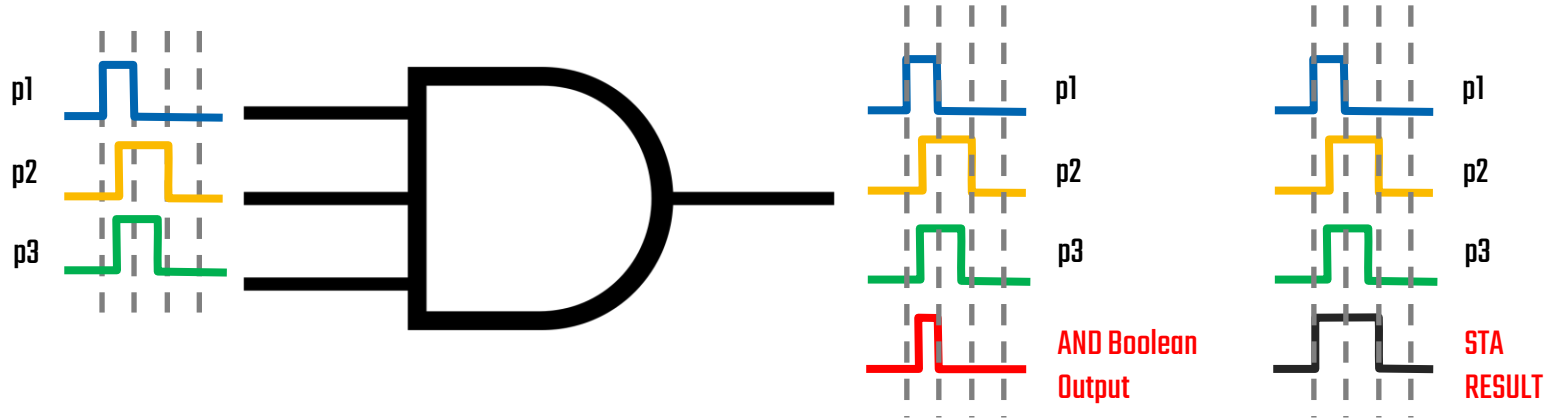
Masking Mechanisms: Electrical Masking



- ❑ Simulate multiple SETs in the same design
- ❑ Different Double Exponential current injection per SET



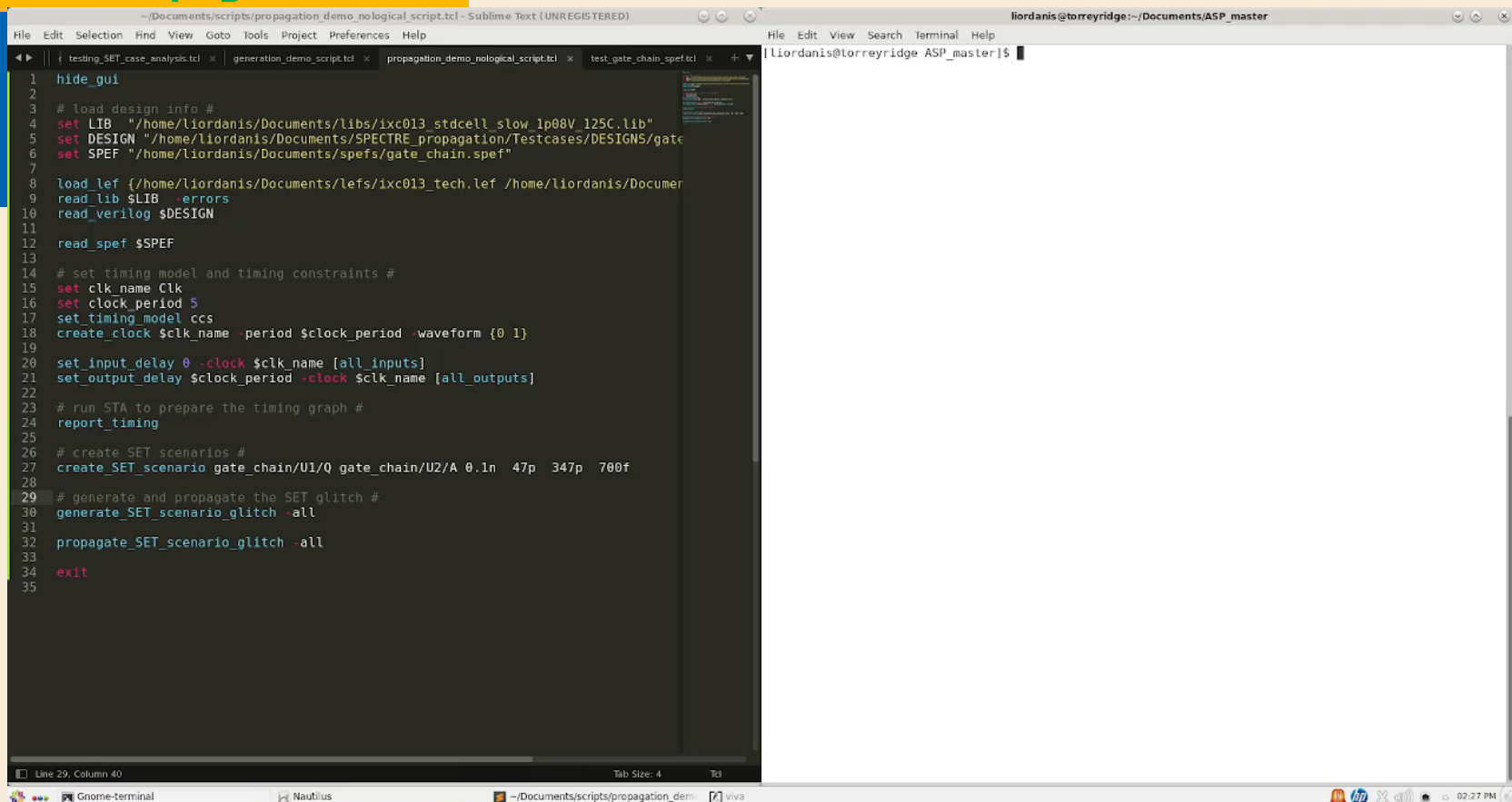
- ❑ STA Merges Pulses, based on Worst-case Timing!!!
- ❑ Here p1, p2, p3 arrive from SET Generation point @ AND inputs:



- ❑ STA does not take into account Boolean values for edge propagation
- ❑ Propagates $MAX(earliest\ rise, latest\ fall), MAX(earliest\ fall, latest\ rise)$
- ❑ Depending on Gate Unateness

Set Propagation DEMO

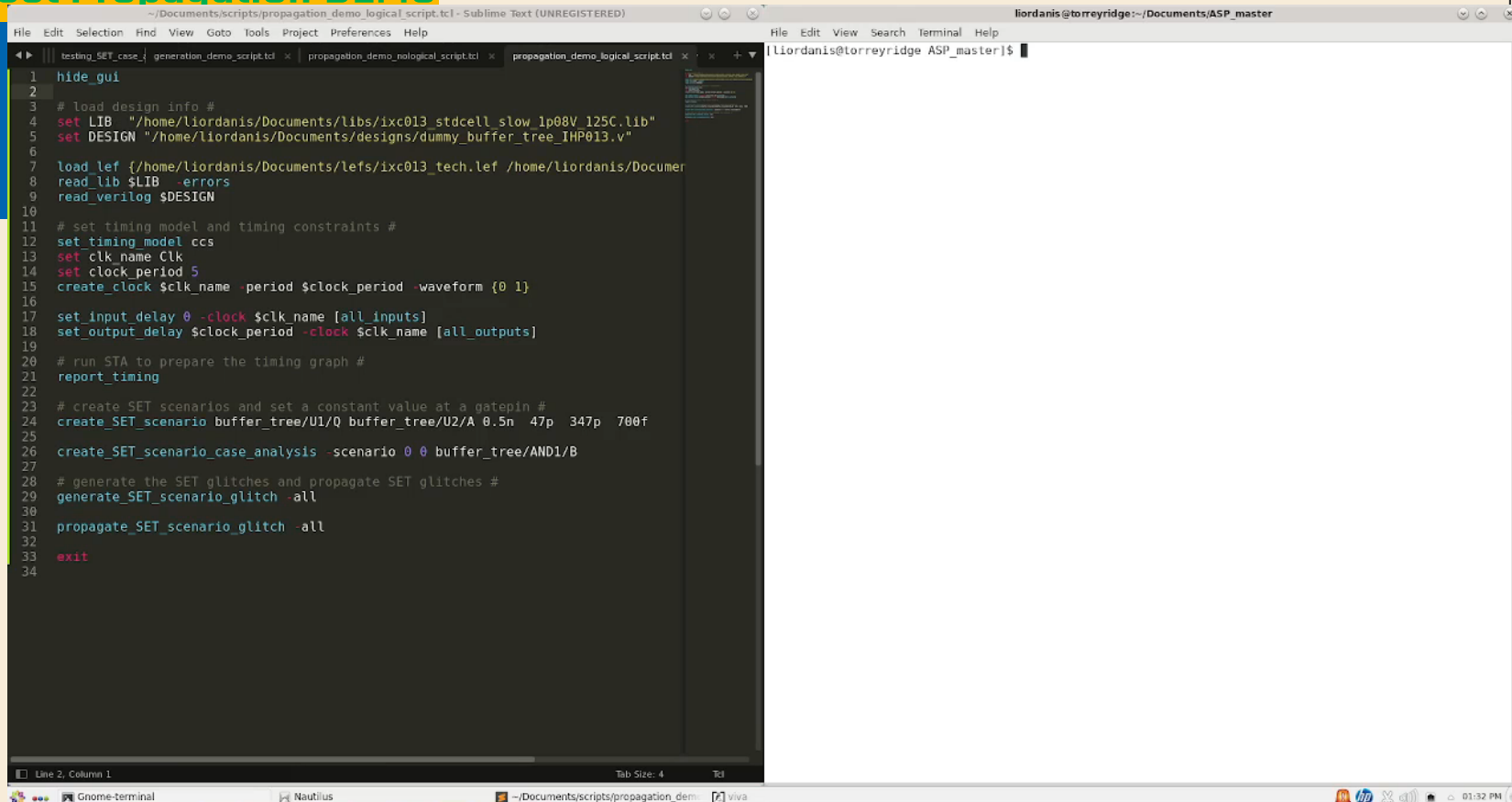
Set Propagation DEMO 1 – No Logical Masking



```
1 hide gui
2
3 # load design info #
4 set LIB "/home/liordanis/Documents/Libs/ixc013_stdcell_slow_1p08V_125C.lib"
5 set DESIGN "/home/liordanis/Documents/SPECTRE_propagation/Testcases/DESIGNS/gate_chain.spef"
6 set SPEF "/home/liordanis/Documents/spefs/gate_chain.spef"
7
8 load_lef {/home/liordanis/Documents/lefs/ixc013_tech.lef /home/liordanis/Documents/lefs/ixc013_tech.lef}
9 read_lib $LIB -errors
10 read_verilog $DESIGN
11
12 read_spef $SPEF
13
14 # set timing model and timing constraints #
15 set clk_name Clk
16 set clock_period 5
17 set timing_model ccs
18 create_clock $clk_name -period $clock_period -waveform {0 1}
19
20 set_input_delay 0 -clock $clk_name [all_inputs]
21 set_output_delay $clock_period -clock $clk_name [all_outputs]
22
23 # run STA to prepare the timing graph #
24 report_timing
25
26 # create SET scenarios #
27 create_SET_scenario gate_chain/U1/Q gate_chain/U2/A 0.1n 47p 347p 700f
28
29 # generate and propagate the SET glitch #
30 generate_SET_scenario_glitch -all
31
32 propagate_SET_scenario_glitch -all
33
34 exit
35
```

Set Propagation DEMO

Set Propagation DEMO 2 – With Logical Masking



The screenshot displays a Sublime Text editor window titled "liordanis@torreyridge: ~/Documents/ASP_master". The editor contains a Tcl script for set propagation. The script includes comments for loading design info, setting the library and design paths, loading the left file, reading the library and design, setting the timing model and constraints, creating a clock, setting input and output delays, running STA to prepare the timing graph, creating SET scenarios, and generating and propagating SET glitches. The script ends with an exit command.

```
1 hide_gui
2
3 # load design info #
4 set LIB "/home/liordanis/Documents/libs/ixc013_stdcell_slow_1p08V_125C.lib"
5 set DESIGN "/home/liordanis/Documents/designs/dummy_buffer_tree_IHP013.v"
6
7 load_lef {/home/liordanis/Documents/lefs/ixc013_tech.lef} /home/liordanis/Documen
8 read_lib $LIB -errors
9 read_verilog $DESIGN
10
11 # set timing model and timing constraints #
12 set_timing_model ccs
13 set_clk_name Clk
14 set_clock_period 5
15 create_clock $clk_name period $clock_period waveform {0 1}
16
17 set_input_delay 0 -clock $clk_name [all_inputs]
18 set_output_delay $clock_period -clock $clk_name [all_outputs]
19
20 # run STA to prepare the timing graph #
21 report_timing
22
23 # create SET scenarios and set a constant value at a gatepin #
24 create_SET_scenario buffer_tree/U1/Q buffer_tree/U2/A 0.5n 47p 347p 700f
25
26 create_SET_scenario_case_analysis -scenario 0 0 buffer_tree/AND1/B
27
28 # generate the SET glitches and propagate SET glitches #
29 generate_SET_scenario_glitch -all
30
31 propagate_SET_scenario_glitch -all
32
33 exit
34
```

The terminal window on the right shows the prompt "liordanis@torreyridge: ASP_master\$" and a cursor.

SET Propagation Results VS. SPECTRE - 1

SPECTRE Design	Rise Arrival (ns)	Rise Slew (ns)	Fall Arrival (ns)	Fall Slew (ns)	Pulse Width (ns)	Execution Time
800buf	58,24090	0,01700	72,55300	0,02120	14,31210	2h 17m
300buf	21,80190	0,01640	27,75580	0,01990	5,95390	18m 51s
200buf	14,50600	0,01470	18,79700	0,01630	4,29100	8m 30s
100buf	7,23025	0,01702	9,85192	0,02160	2,62167	2m 14s

ASP Req Design	Rise Arrival (ns)	Rise Slew (ns)	Fall Arrival (ns)	Fall Slew (ns)	Pulse Width (ns)	Execution Time
800buf	54,53018	0,01793	67,83548	0,02167	13,30530	1.730s
300buf	20,42540	0,01703	26,26638	0,02038	5,84098	0.43s
200buf	13,59925	0,01489	17,95487	0,01639	4,35562	0.35s
100buf	6,78687	0,01793	9,64884	0,02167	2,86197	0.21s

SET Propagation Results VS. SPECTRE - 2

SPECTRE Design	Rise Arrival (ns)	Rise Slew (ns)	Fall Arrival (ns)	Fall Slew (ns)	Pulse Width (ns)	Execution Time
800buf	58,24090	0,01700	72,55300	0,02120	14,31210	2h 17m
300buf	21,80190	0,01640	27,75580	0,01990	5,95390	18m 51s
200buf	14,50600	0,01470	18,79700	0,01630	4,29100	8m 30s
100buf	7,23025	0,01702	9,85192	0,02160	2,62167	2m 14s
ASP SPECTRE Pulse Design	Rise Arrival (ns)	Rise Slew (ns)	Fall Arrival (ns)	Fall Slew (ns)	Pulse Width (ns)	Execution Time
800buf	54,53580	0,01793	67,37930	0,02167	12,84350	1.640s
300buf	20,43102	0,01703	25,81022	0,02038	5,37920	0.290s
200buf	13,60487	0,01489	17,49871	0,01693	3,89384	0.180s
100buf	6,79249	0,01793	9,19268	0,02167	2,40019	0.04s

SET Propagation Results – AT & Slew Errors

AT Error	ASP rise AT Error	ASP fall AT Error	ASP SPECTRE Pulse rise AT Error	ASP SPECTRE Pulse fall AT Error
800buf	6,37%	6,50%	6,36%	7,13%
300buf	6,31%	5,37%	6,29%	7,01%
200buf	6,25%	4,48%	6,21%	6,91%
100buf	6,13%	2,06%	6,05%	6,69%
Averages	6,27%	4,60%	6,23%	6,93%

Slew Error	ASP rise Slew Error	ASP fall Slew Error	ASP SPECTRE Pulse rise slew Error	ASP SPECTRE Pulse fall slew Error
800buf	-5,44%	-2,24%	-5,44%	-2,24%
300buf	-3,82%	-2,41%	-3,82%	-2,41%
200buf	-1,29%	-0,57%	-1,29%	-3,87%
100buf	-5,32%	-0,34%	-5,32%	-0,32%
Averages	-3,97%	-1,39%	-3,97%	-2,21%

(- = pessimistic, + = optimistic)

CONFIDENTIAL

SET Propagation Results – PW Errors

Pulse Width Error vs SPETRE	ASP	ASP SPECTRE Pulse
800buf	7,03%	10,26%
300buf	1,90%	9,65%
200buf	-1,51%	9,26%
100buf	-9,17%	8,45%
Averages	-0,44%	9,40%

~5000 x

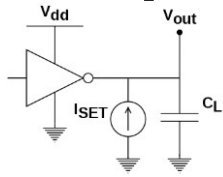
Speedup

ASP VS. PrimeTime *for SETs*

Features	Path-based SET Analysis using PrimeTime	ASP Tool
SET Generation	X	✓
Pulse Propagation	X	✓
Constant Node Support	X	✓
Logical Masking (Primary Inputs Case Analysis)	✓	✓
Electrical Masking	X	✓
Non Full-Swing Signals	X	✓
Path Support	✓	✓
Non-Path, Logic Cone Support	X	✓
Graph Based (High-Capacity)	X	✓

CONFIDENTIAL

SET Propagation, Simulation Interaction

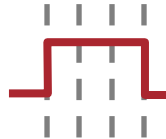


STA-based
SET Generation

Calibration vs.
SPICE/SPECTRE

STA-based
SET Propagation

STA Pulse (WC)

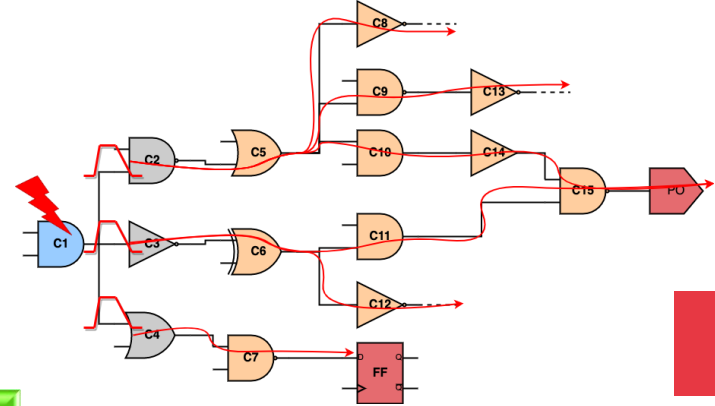


Simulation Pulse
Vector Based



Optimisations to
Improve SER; Gate
Upsizing, Buffer
Insertion, etc.

Simulation to
Reduce Pulse
Width Pessimism



csotiriou@gmail.com

(US): +1 858 247 9421

caslab.e-ce.uth.gr

CONFIDENTIAL

