Functional Design Bugs Detected With Low Power Verification

Alexander Kryvoshaev
Qualcomm

October, 8th, 2014
SNUG Canada

Agenda

Discussion On Power Consumption

Introduction To Power Aware Flow

Bugs Found With Power Aware Flow

Tips On Using VCS NLP Flow

Discussion On Power Consumption

Why Save Power?

Limited Battery Capacity

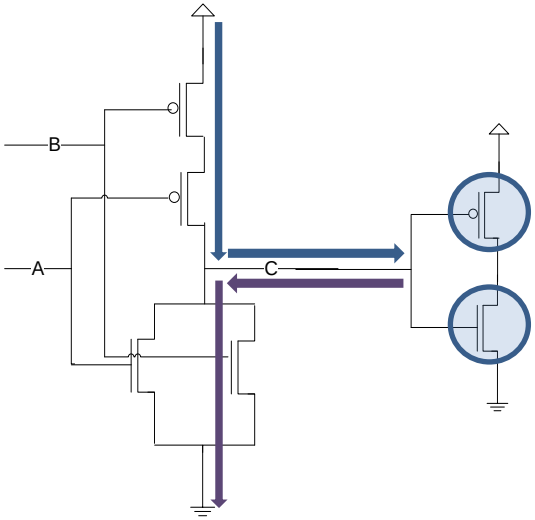
- Batteries are not sufficient for contemporary needs
- The existing improvements are expensive

Heat Problem

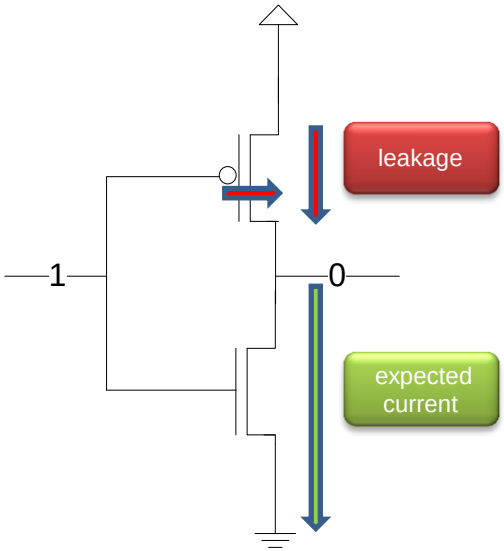
- Increasing the number of transistors per area unit leads to heat problems
- Cooling systems are not sophisticated enough

Area increase → Noise → More repeaters → Lower frequency

Dynamic And Static Power Consumption

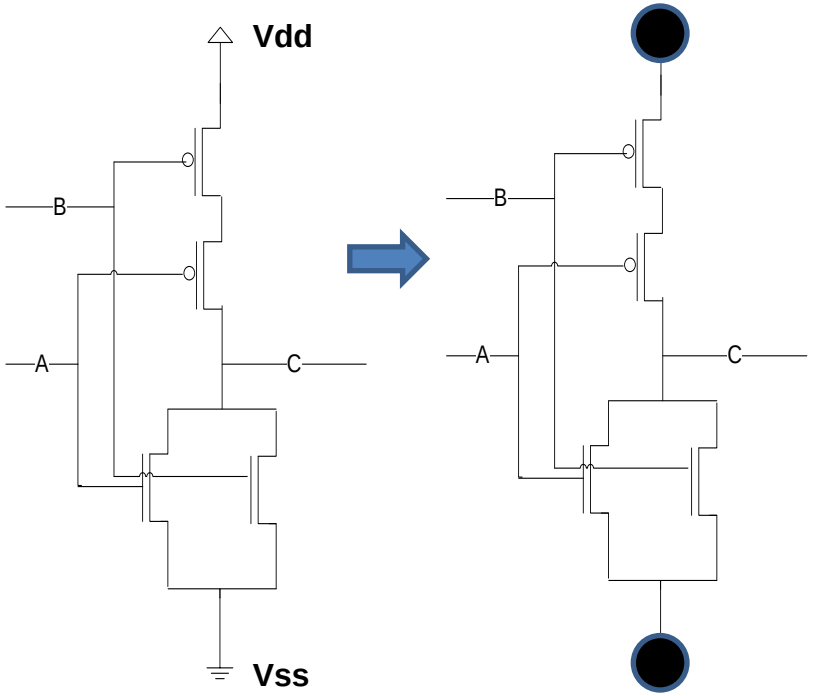


- The current, which charges the capacitor of the next gate, consumes power
- The bigger the gate, the higher current required. This causes higher power consumption
- The width of transistor is proportional to the current going through it



- When a transistor is closed, there is still current going from source to drain and through the gate
- The only way to eliminate the static power consumption is to shut down power supplies

Shutting Power Down

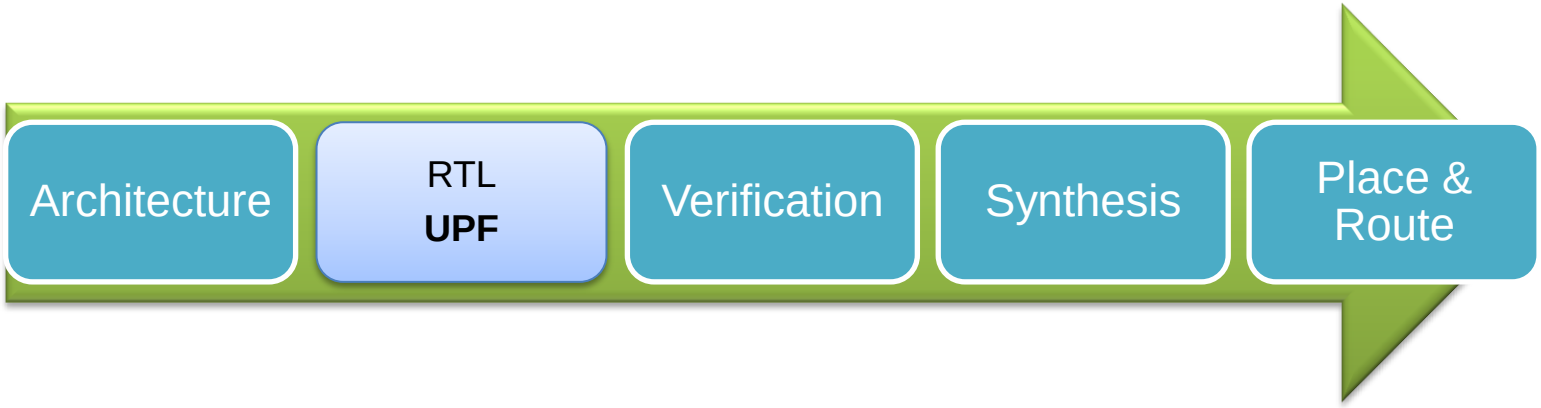


- Power supply, of transistors in the collapsed area, is turned off
- After shutting down, the next gate is not charged anymore
- The gate charges are subject to noise
- As a result, the state of the next gate is unknown. In simulations this unknown state is denoted by ‘X’

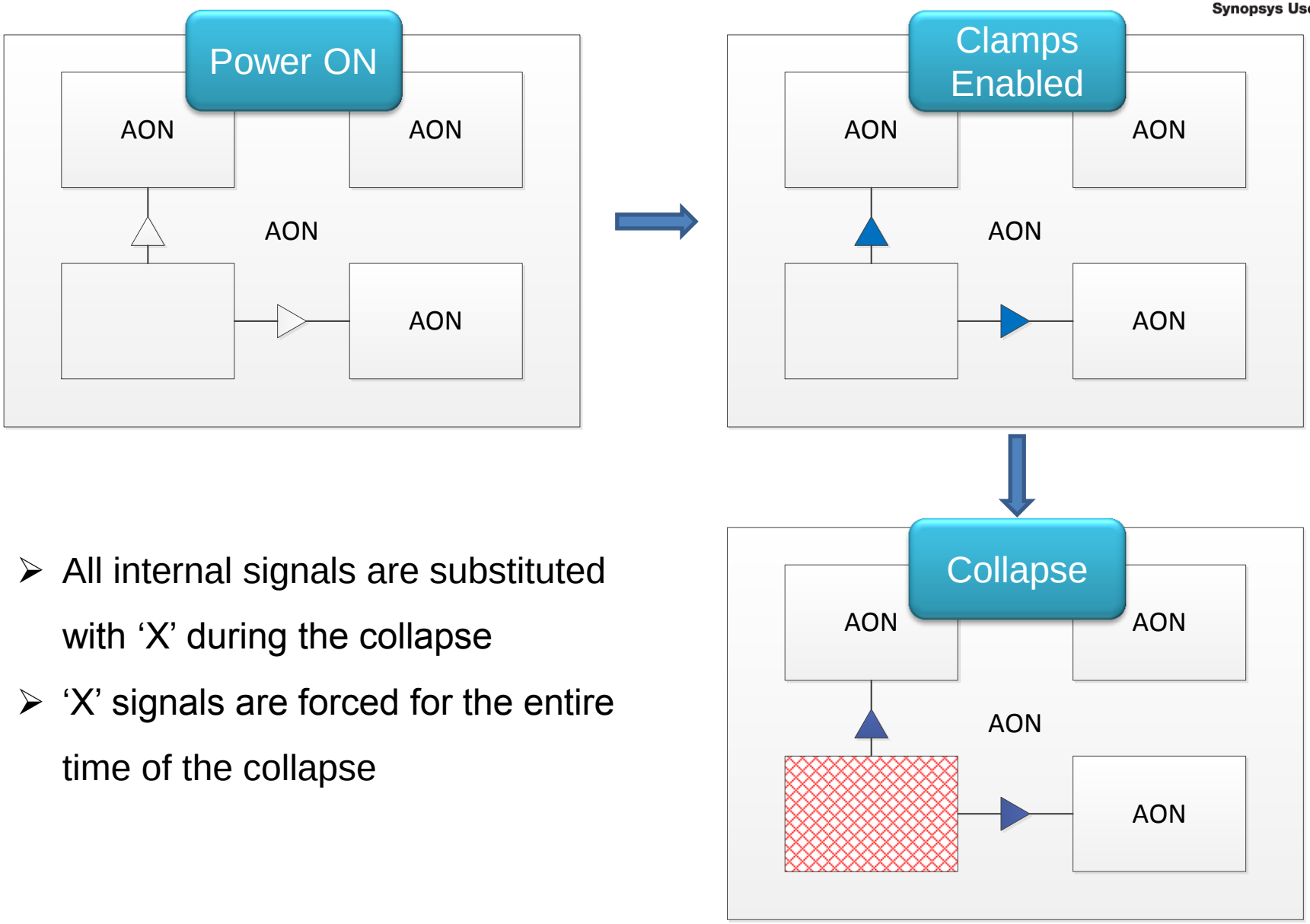
Introduction To Power Aware Flow

Chip Development Flow

- Static power can be saved by shutting down power supplies
- Power map has to be developed along with RTL design
- Power supply shutdown is initiated by triggering the RTL logic condition
- UPF files describe the power intent in a chip development flow
- An updated chip development flow can be found below

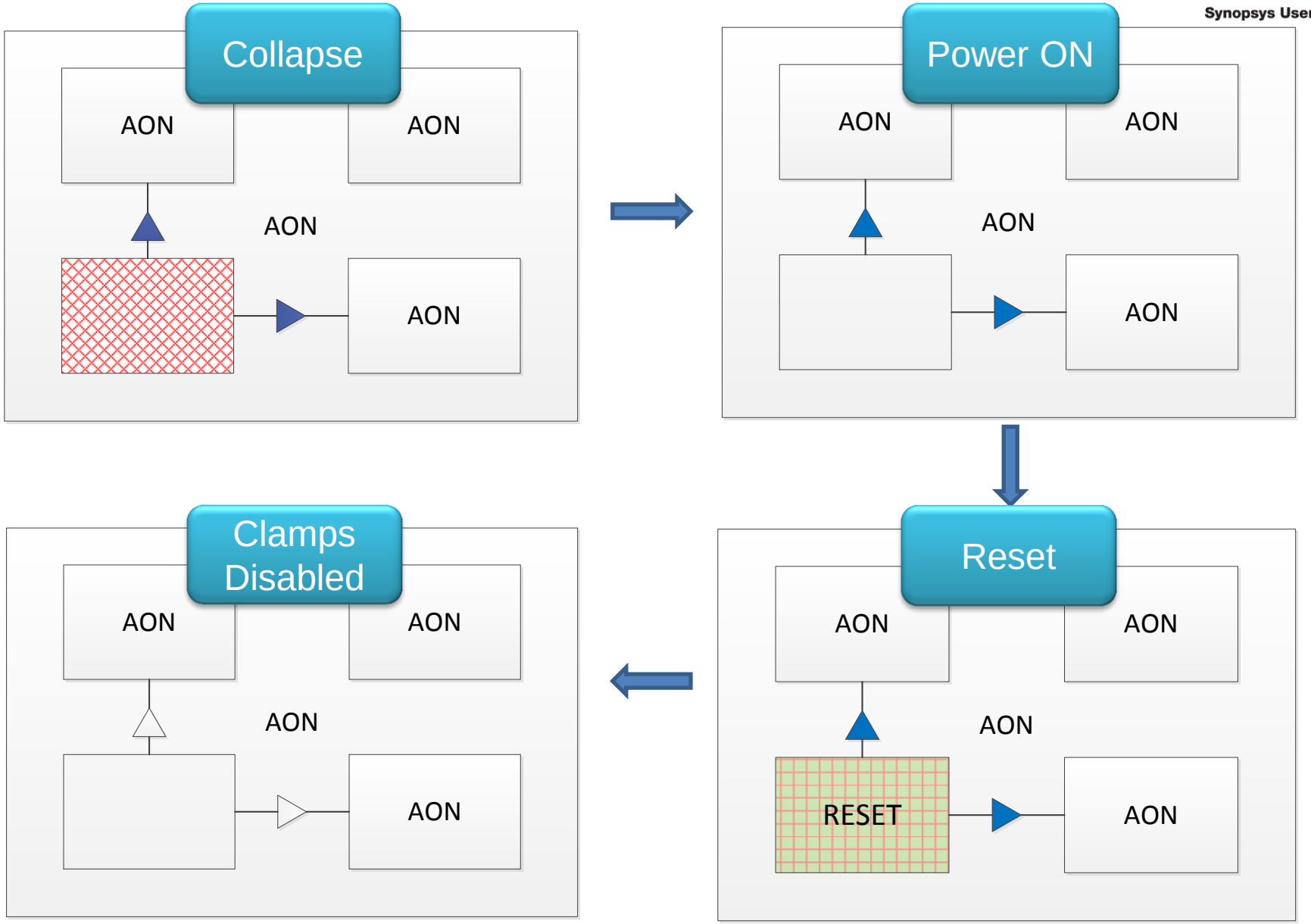


PA flow simulation – collapsing

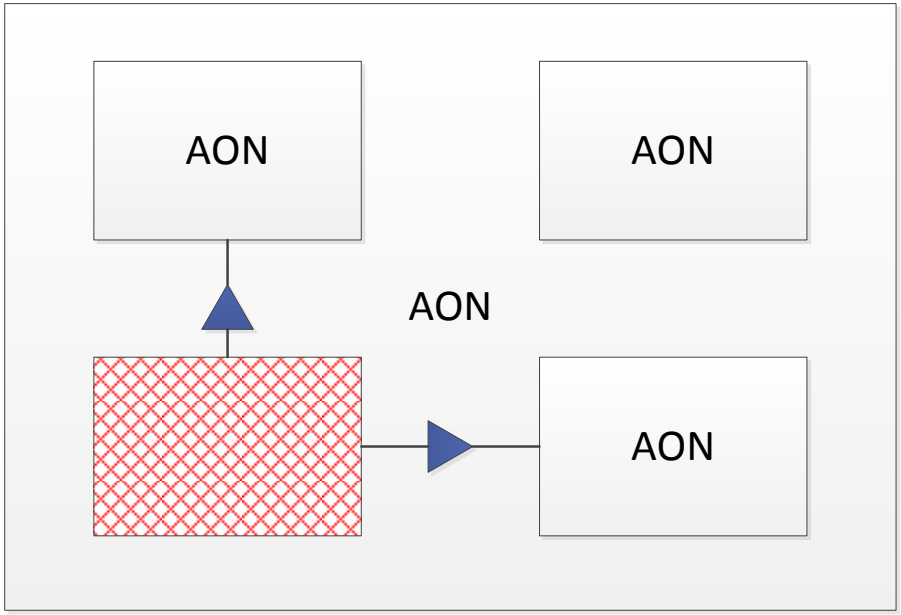


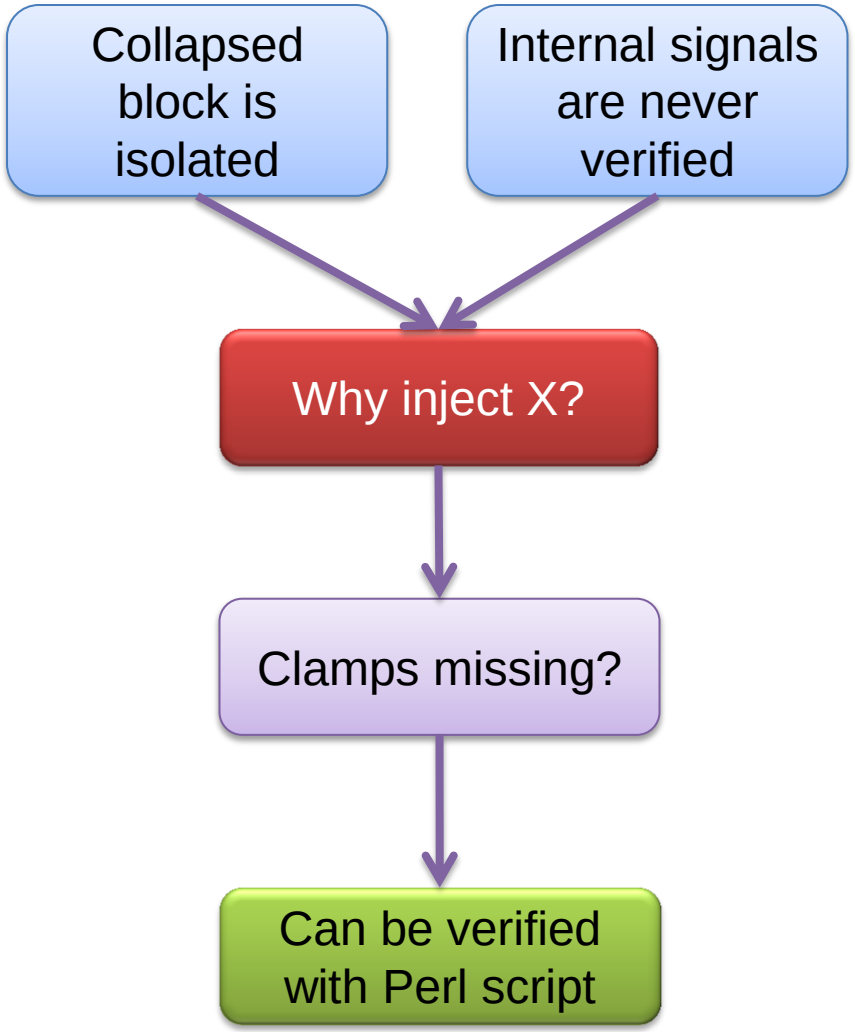
- All internal signals are substituted with ‘X’ during the collapse
- ‘X’ signals are forced for the entire time of the collapse

PA flow simulation – recovery

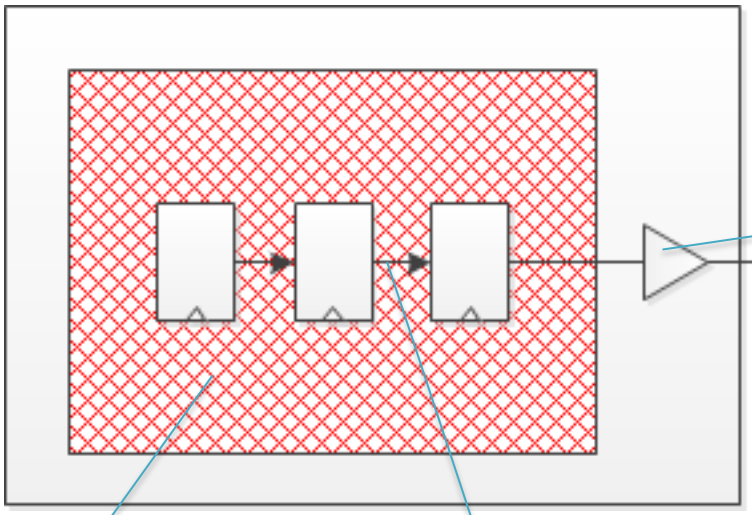


PA flow simulation – X insertion





Reset And Clamps



Step 3: Clamps

Isolation is removed

Step 1: Power ON

Power is back

Step 2: Reset

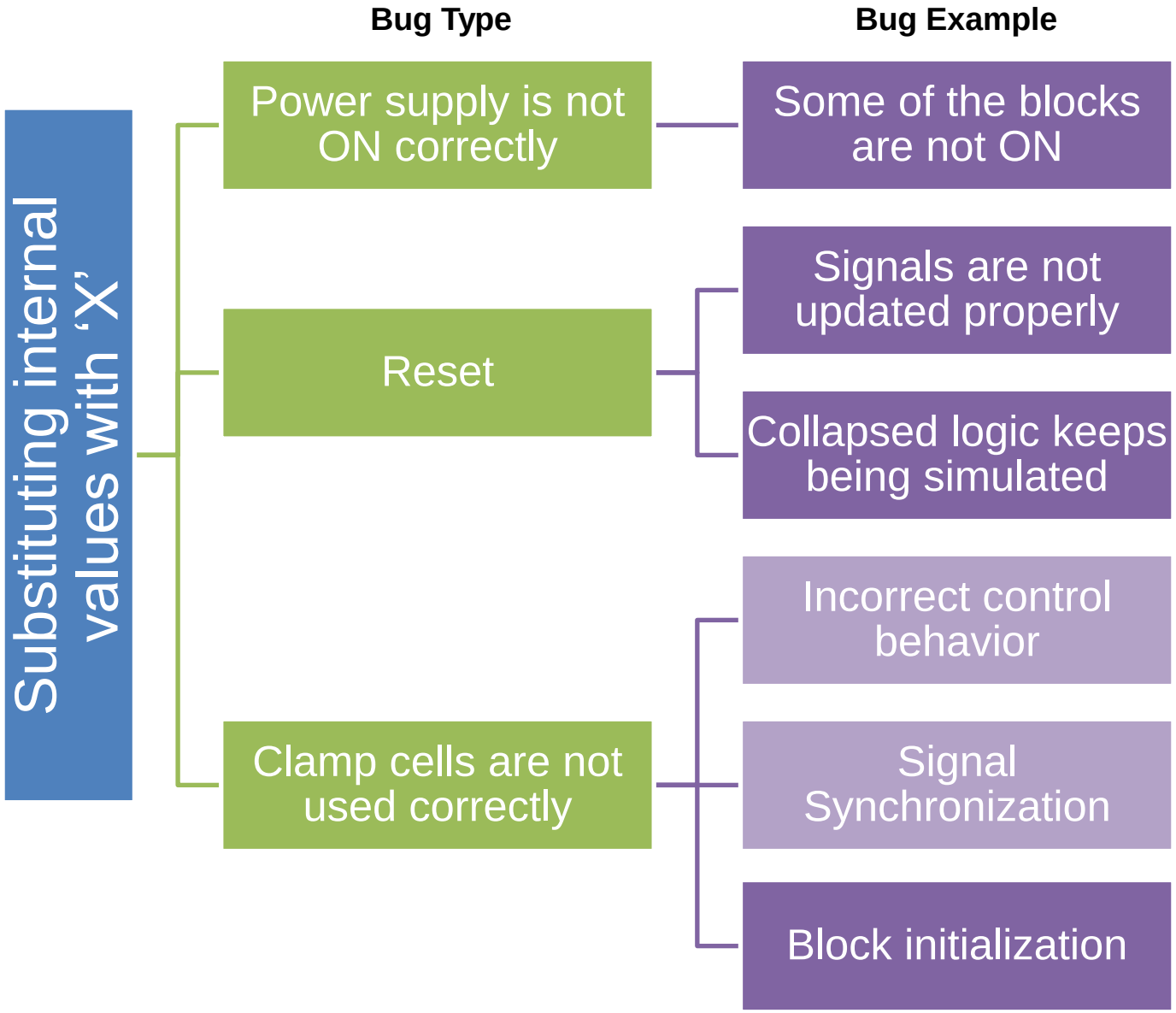
FFs are reset

If these 3 steps are done correctly the injection of X is meaningless

In an ideal world we wouldn't need PA simulations

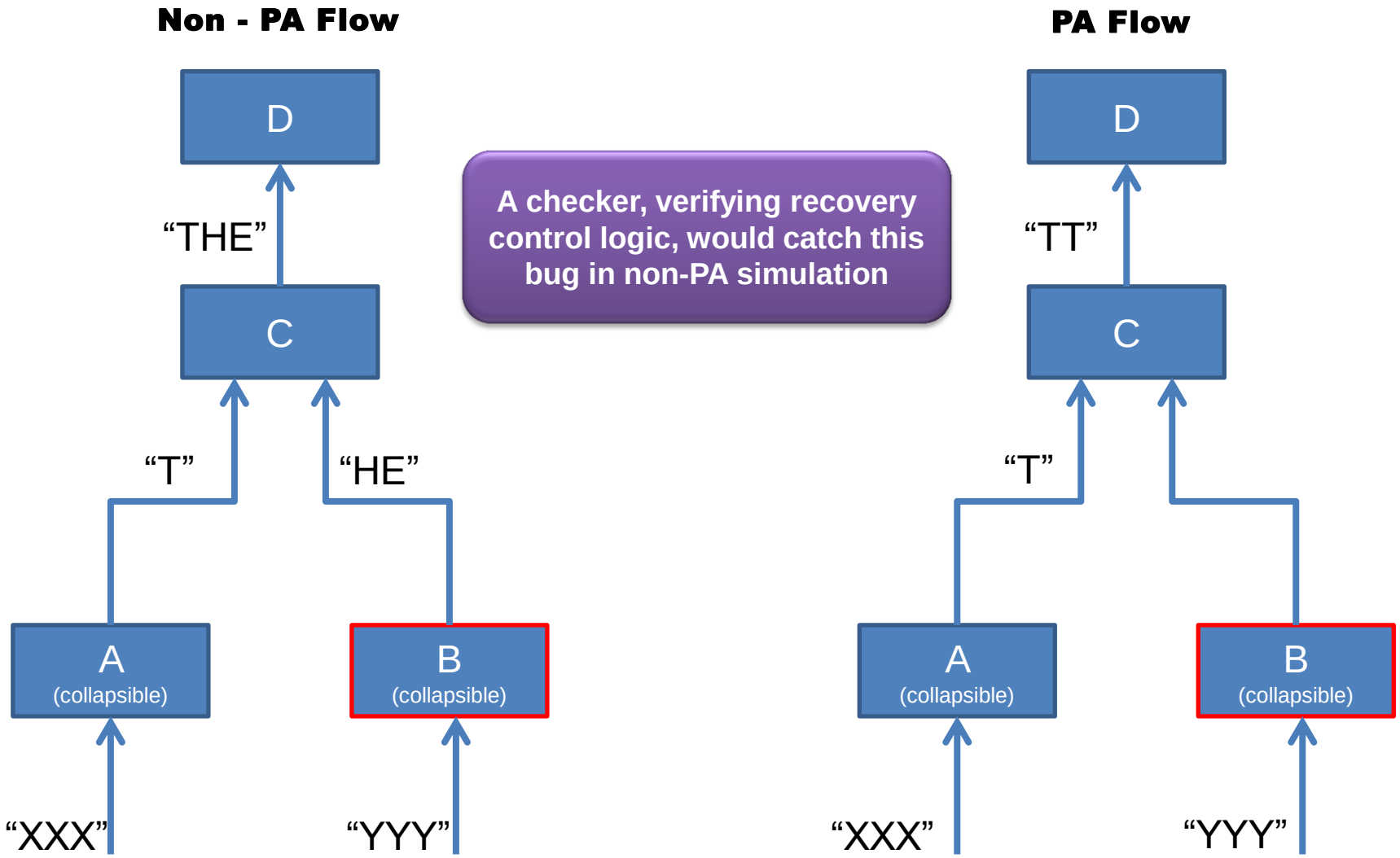
Every bug, found with PA is a verification scenario inaccuracy

Bugs Classification

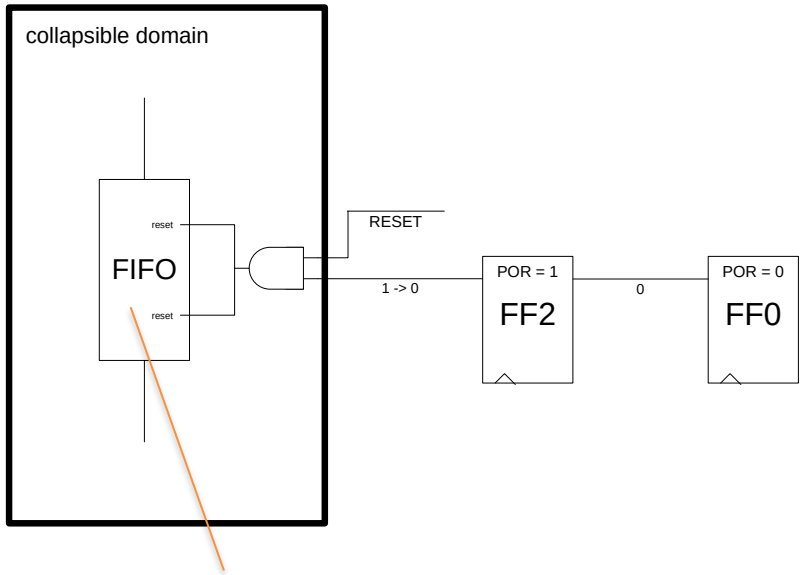


Bugs found with Power Aware Flow

Example 1: Power doesn't come ON



Example 2: Reset Bug



FIFO returns acknowledgement on abort request only when it is empty.

Counters on both sides are always synchronized.

A checker, verifying reset propagation, would catch this bug in non-PA simulation

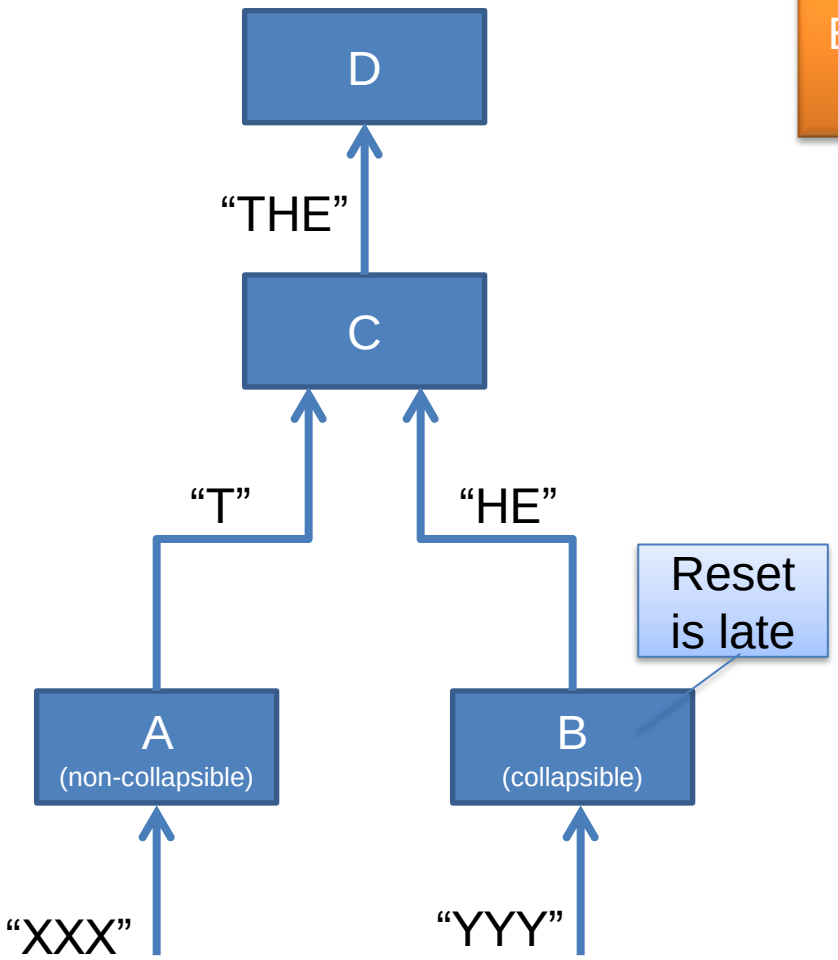
Non - PA Flow

Cycle	Operation	Write Pointer	Read Pointer	Power State
1		0	0	ON
2	Push 1	1	0	ON
3	Push 2	2	0	ON
4	Pop 1	2	1	ON
5	Pop 2	2	2	ON
6		2	2	Collapsed
7	Reset is not propagating	2	2	ON
8	Push 3	3	2	ON
9	Pop 3	3	3	ON

Real World / PA Flow

Cycle	Operation	Write Pointer	Read Pointer	Power State
1		0	0	ON
2	Push 1	1	0	ON
3	Push 2	2	0	ON
4	Pop 1	2	1	ON
5	Pop 2	2	2	ON
6		X	X	Collapsed
7	Reset is not propagating	X	X	ON
8	Push 3	X	X	ON
9	Pop - getting error	X	X	ON

Example 3: Collapsed Logic Simulated



BUG: Collapsing before the word is completed is not allowed (architectural bug)

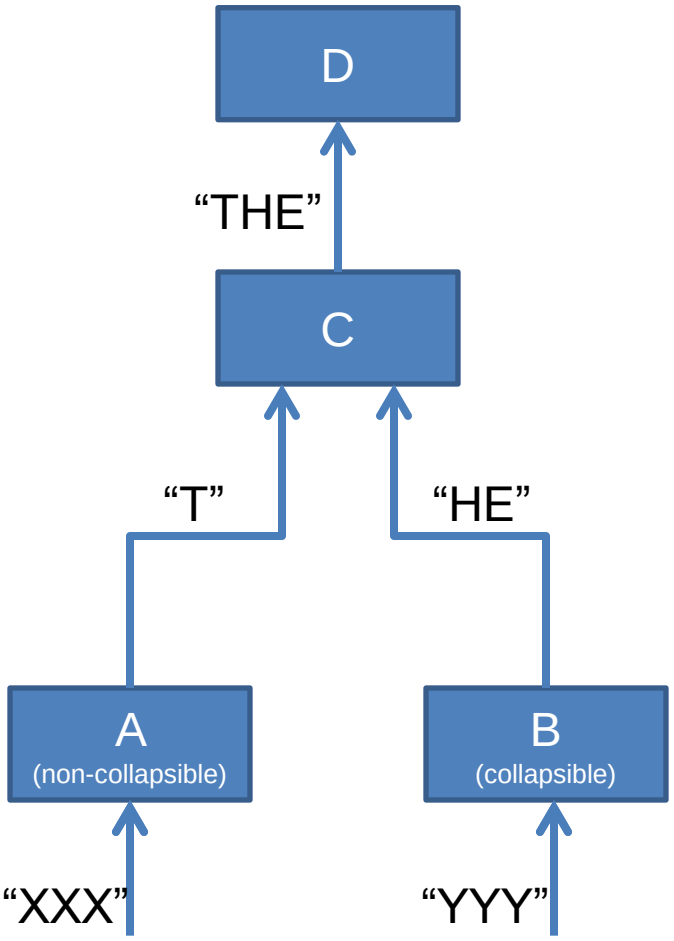
Non-PA simulation processes inputs during the collapse

'Illegal' conversion is not reset before power recovery

Block B gives output on time and is synchronized with A

Collapsed Logic Simulated

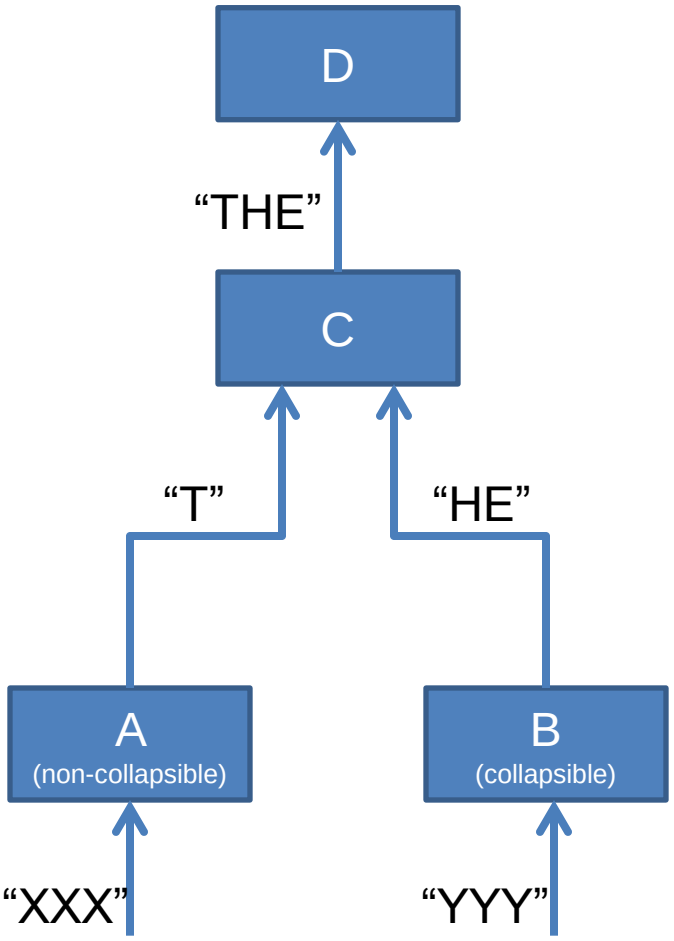
Synopsys Users Group



Real World / PA Flow

Cycle	A	B	C	PWR status
1	Works on "T"		Waiting	
2	Works on "T"	Works on “HE"	Waiting	
3	Transmits "T"	Works on “HE"	Receives “T", ACK to A	
4	Works on "T"	Transmits “HE"	Receives “HE", ACK to B	
5	Works on "T"	Works on “HE"	Waiting	
6	Transmits "T"	Works on “HE"	Receives “T"	
7	Works on "T"	Transmits “HE"	Receives “HE"	
8	Works on "T"	Works on “HE"	Waiting	
9	Transmits "T"	Works on “HE"	Receives “T"	
10	Works on "T"	Collapsed	Waiting	Collapse
11	Works on "T"	Collapsed	Waiting	Collapse
12	Transmits "T"	Collapsed	NACK to A	Collapse
13	Transmits "T"	Collapsed	NACK to A	Collapse
14	Transmits "T"	Collapsed	NACK to A	Collapse
15	Transmits "T"	Collapsed	NACK to A	Collapse
16	Transmits "T"	Before reset/NOP	ACK to A	
17	Works on "T"	“Late reset” values	Waiting	
18	Works on "T"	Works on “HE"	Waiting	

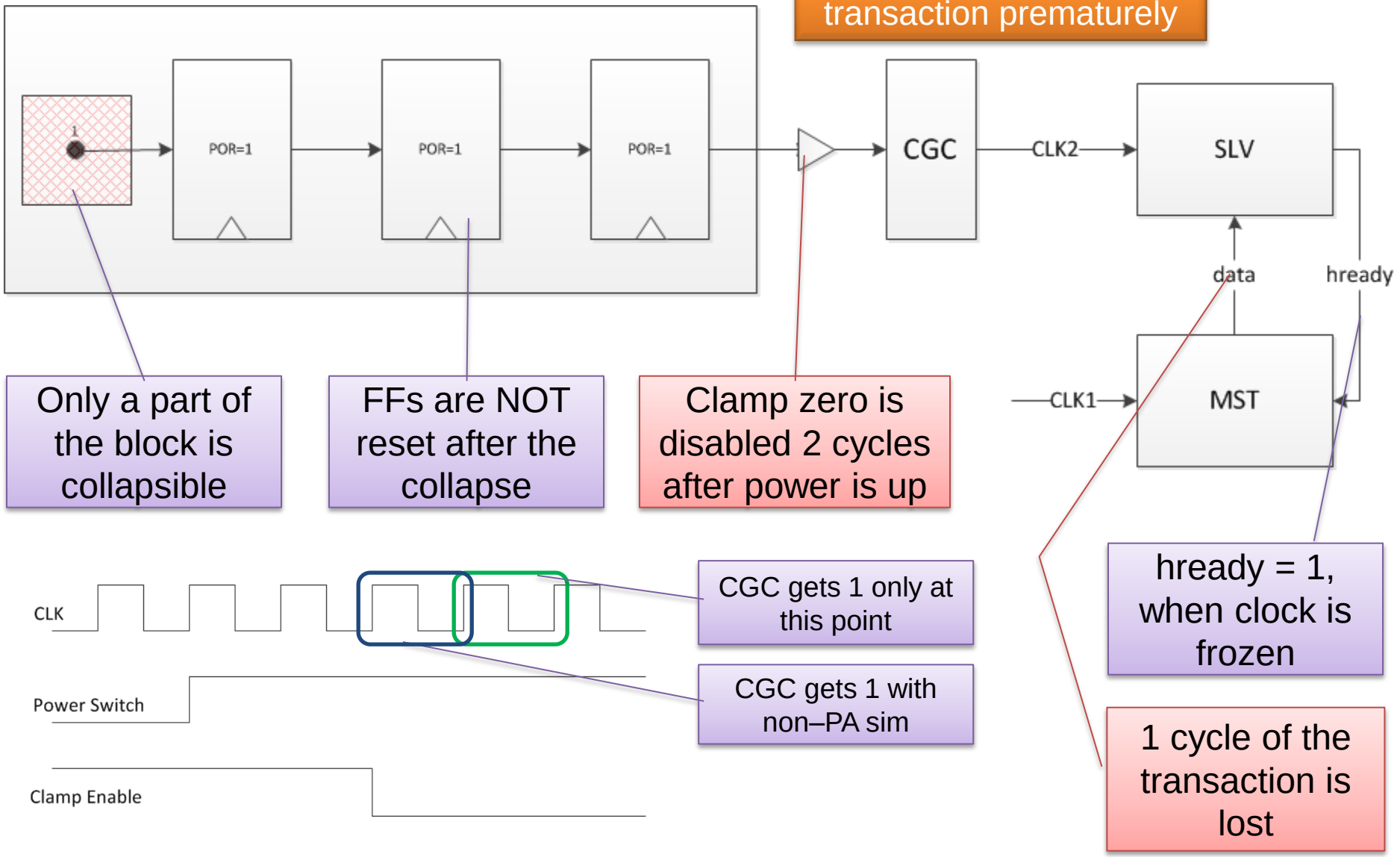
Collapsed Logic Simulated



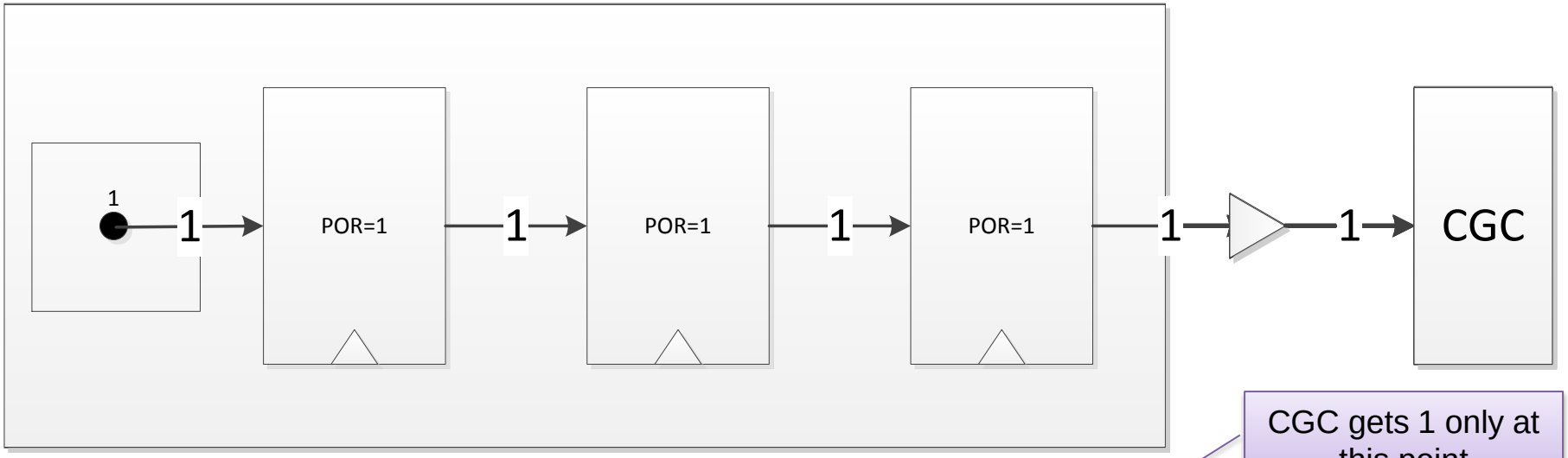
Non - PA Flow

Cycle	A	B	C	PWR status
1	Works on "T"		Waiting	
2	Works on "T"	Works on "HE"	Waiting	
3	Transmits "T"	Works on "HE"	Receives "T", ACK to A	
4	Works on "T"	Transmits "HE"	Receives "HE", ACK to B	
5	Works on "T"	Works on "HE"	Waiting	
6	Transmits "T"	Works on "HE"	Receives "T"	
7	Works on "T"	Transmits "HE"	Receives "HE"	
8	Works on "T"	Works on "HE"	Waiting	
9	Transmits "T"	Works on "HE"	Receives "T"	
10	Works on "T"	Transmits "HE"	NACK to B	Collapse
11	Works on "T"	Transmits "HE"	NACK to B	Collapse
12	Transmits "T"	Transmits "HE"	NACK to A and B	Collapse
13	Transmits "T"	Transmits "HE"	NACK to A and B	Collapse
14	Transmits "T"	Transmits "HE"	NACK to A and B	Collapse
15	Transmits "T"	Transmits "HE"	NACK to A and B	Collapse
16	Transmits "T"	Transmits "HE"	ACK to B, NACK to A	
17	Transmits "T"	“Late Reset” values	ACK to A	
18	Works on "T"	Works on "HE"	Waiting	

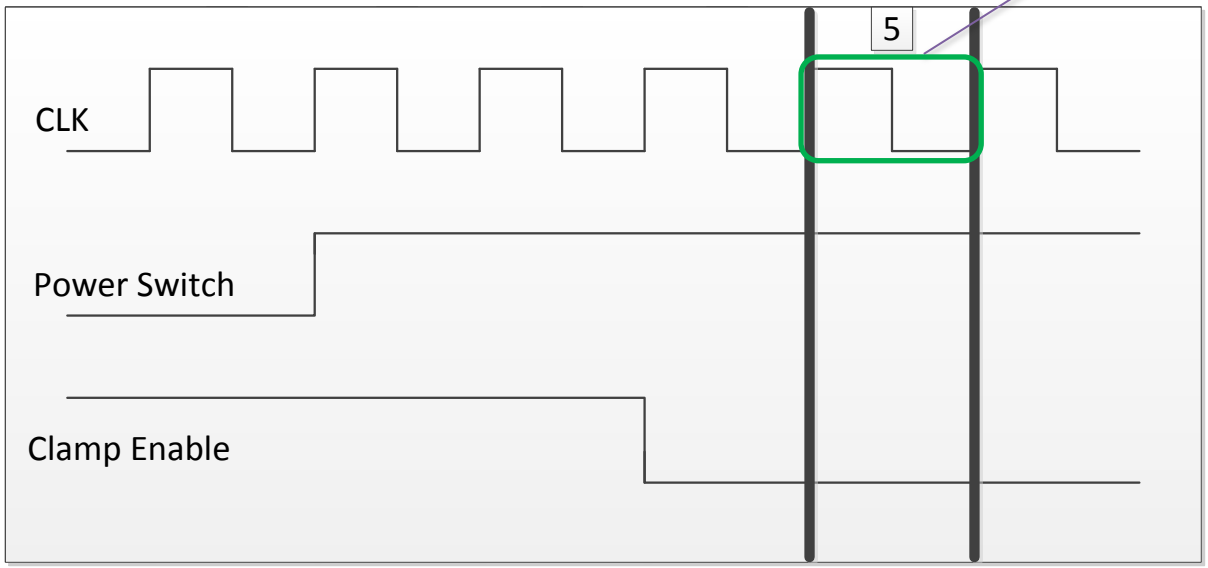
Example 4: Long Initialization



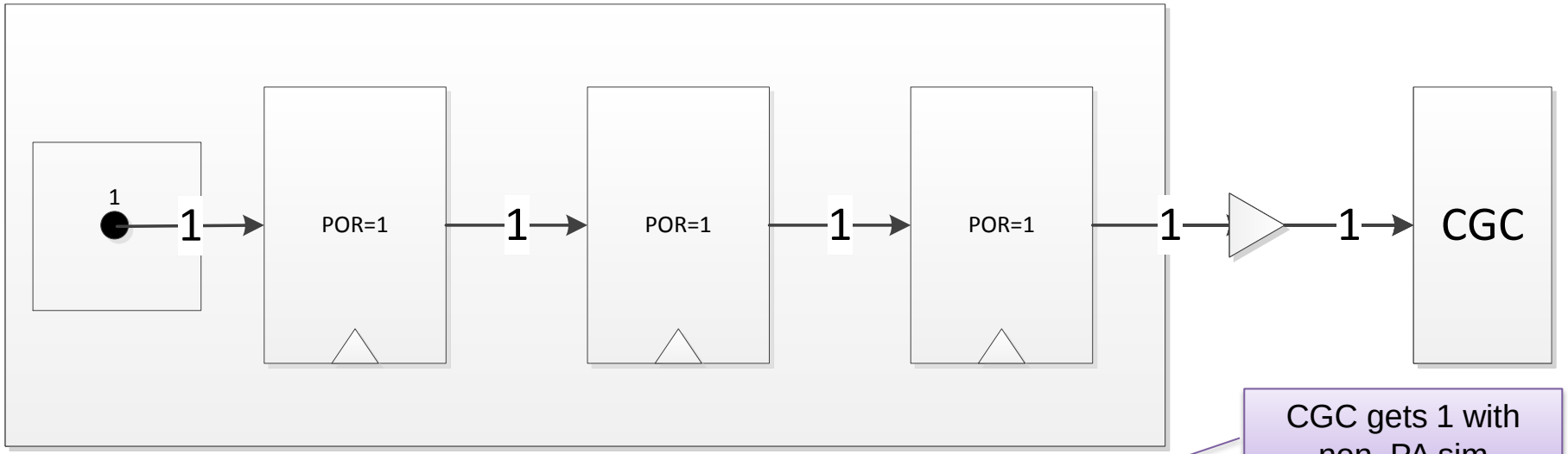
Long Initialization – PA flow



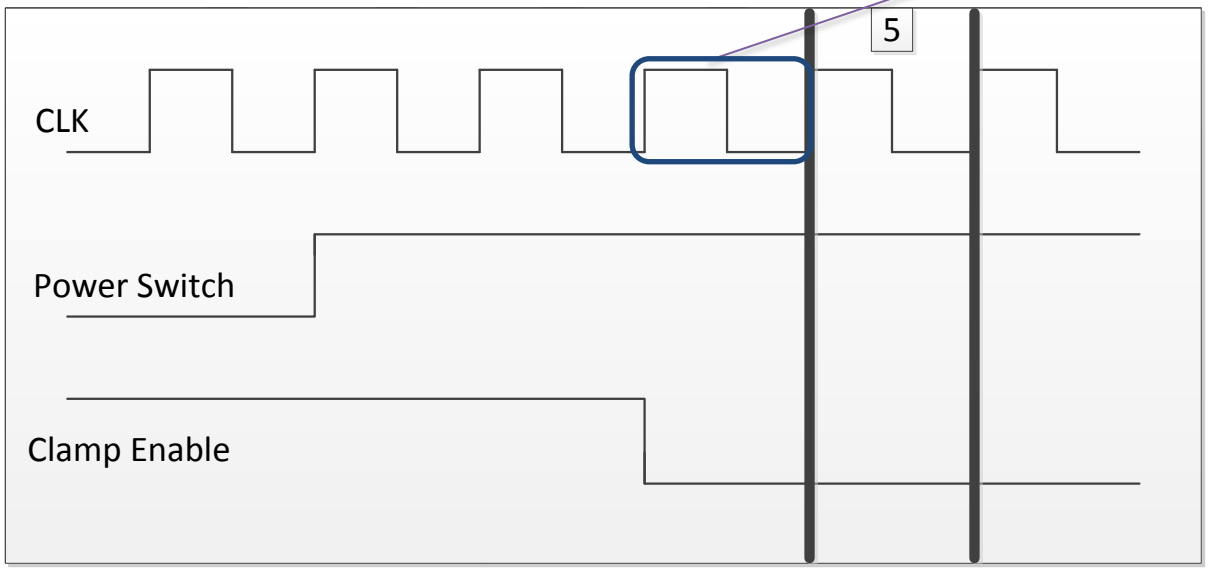
CGC gets 1 only at this point



Long Initialization – non-PA flow

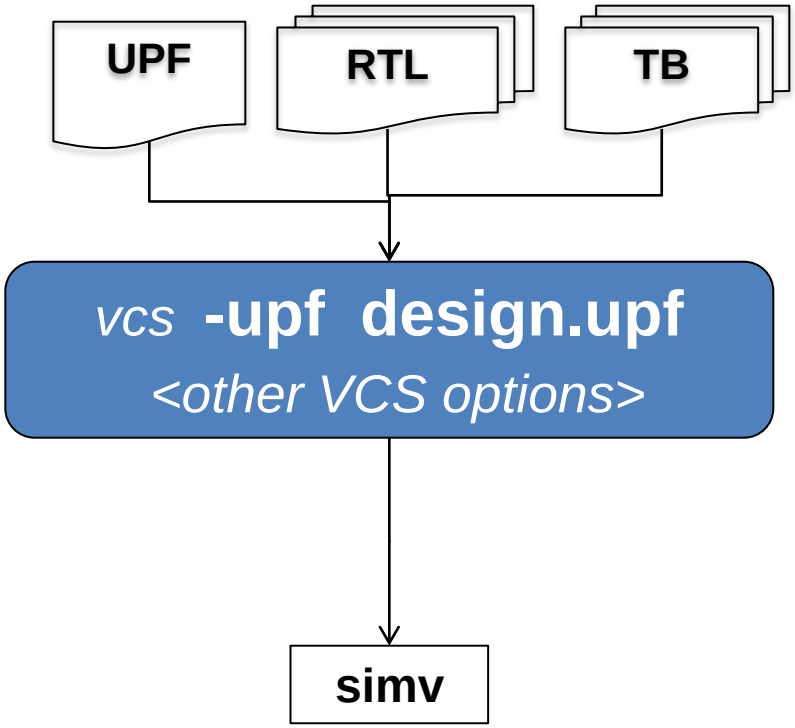


CGC gets 1 with non–PA sim



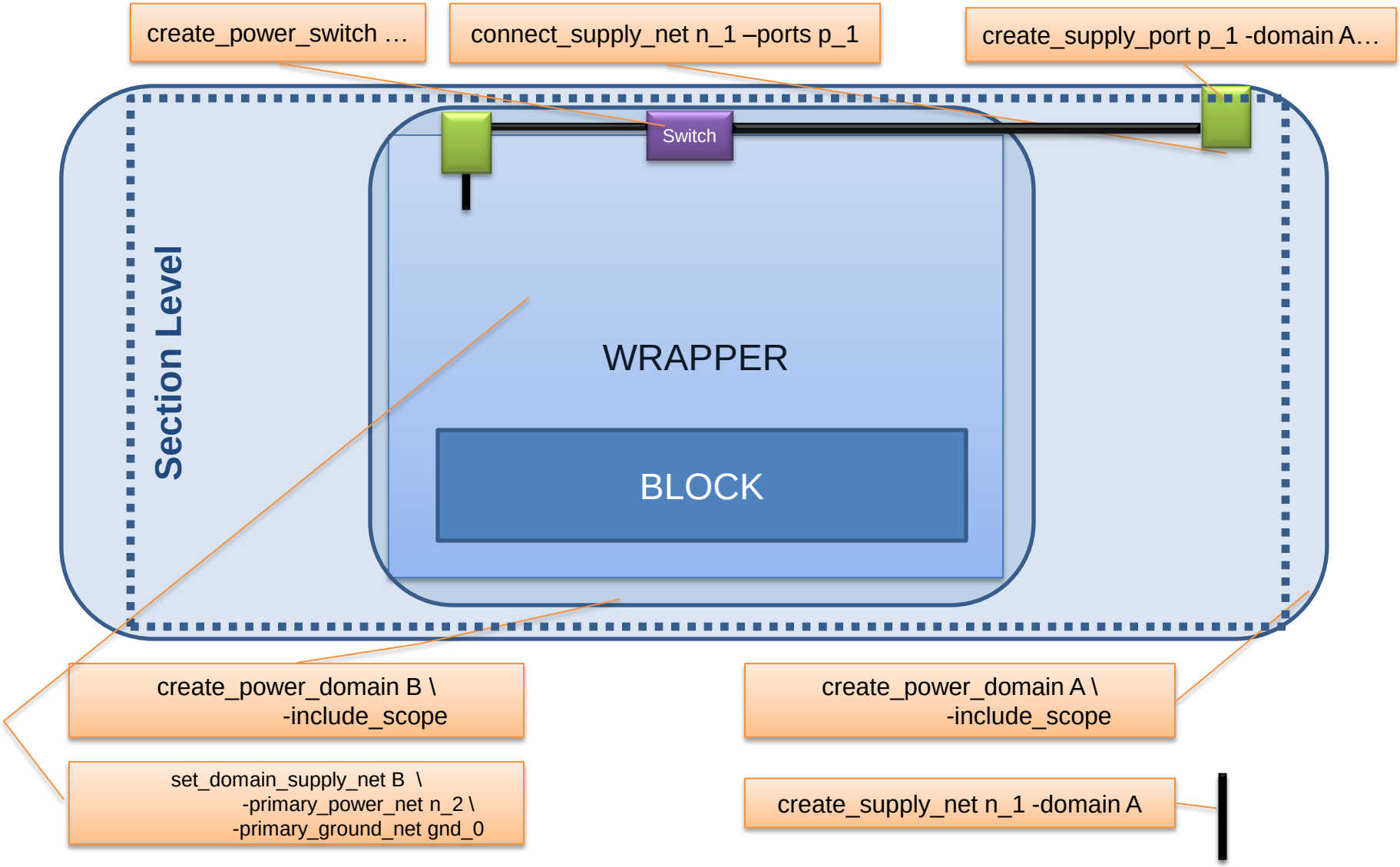
Tips On Using VCS NLP Flow

Debugging With NLP



- UPF and RTL describe the design
- TB is a layer between design and verification
- Adding UPF power content is easy by adding a switch

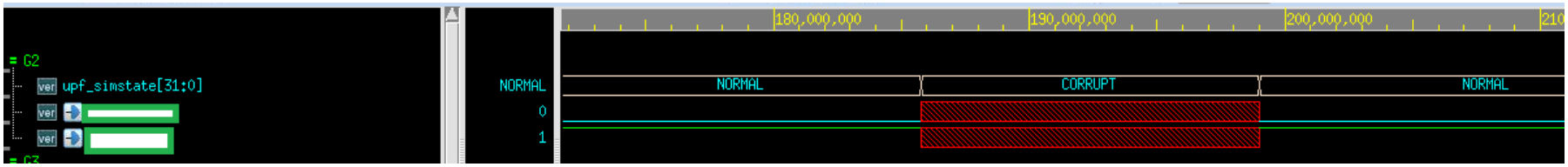
UPF Introduction



Debugging With NLP

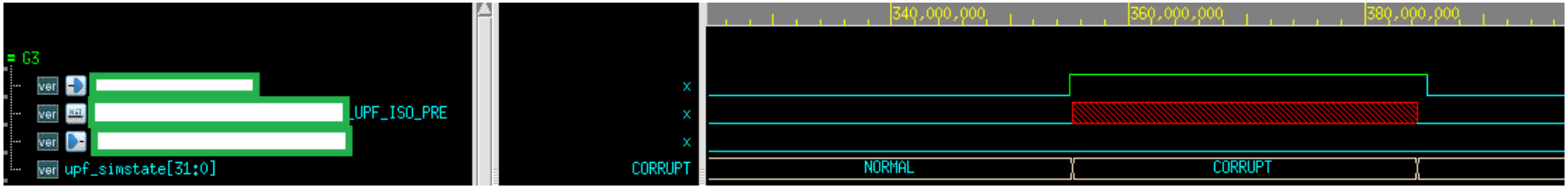
Power Shutdown

upf_simstate virtual signal keeps the state of the power domain.
The second and the third signals are collapsed, when the block is collapsed.



Isolation cells

The first signal is clamp control. Once it is enabled the isolation signal (the third one), gets value ‘1’. The pre-clamp value of the signal is kept in “_UPF_ISO_PRE” virtual signal.



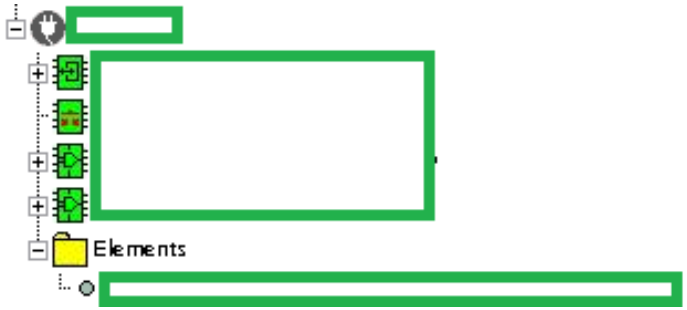
The signal names are grayed for security purposes.

Debugging With NLP

Synopsys Users Group

UPF interpretation

DVE/Verdi provide a convenient UPF interpretation:



CORRUPT	Power Domain
	Retention Strategy
INVALID	Power Switch
	Isolation Strategy
	Isolation Strategy
CORRUPT	Element

We can observe power domain, isolation cells and element, included in this power domain.

The signal names are grayed for security purposes.

Debugging With NLP

Synopsys Users Group

Messages

Corrupt state message:

[1285825103 ps] [INFO] [LP_PPN_STATE_CHANGE] State of the primary power net 'vdd_int' of power domain 'SOME_PATH/VDD_INT' changed from FULL_ON to OFF.
[1285825103 ps] [INFO] [LP_PPN_VALUE_CHANGE] Voltage of the primary power net 'vdd_int' of power domain 'SOME_PATH/VDD_INT' changed from 1 V to 0 V.

Power recovery message:

[2656232 ps] [INFO] [LP_PPN_STATE_CHANGE] State of the primary power net 'vdd_int' of power domain 'SOME_PATH/VDD_INT' changed from OFF to FULL_ON.

[2656232 ps] [INFO] [LP_PPN_VALUE_CHANGE] Voltage of the primary power net 'vdd_int' of power domain 'SOME_PATH/VDD_INT' changed from 0 V to 1 V.

Debugging With NLP

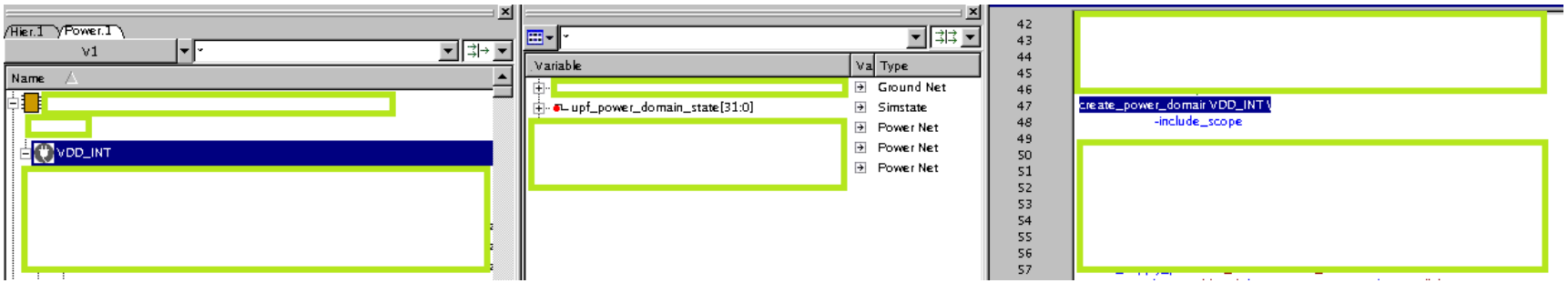
LPA VCS built in assertions

Built in LPA assertions are a useful tool for catching low power issues.LPA provide warning messages when inputs are toggled during the collapse.

Disbabling of LPA built in assertions

LPA VCS built in assertions are very useful, but sometimes the assertions trigger an alert on custom design. This is an example on how to disable LPA assertions:
set_lp_msg_onoff off -msg {LP_PSW_CTRL_INIT_INVALID}

DVE has an option to show UPF source code



The signal names are grayed for security purposes.

Summary

- PA flow makes a difference only when there is reset, clamp control or wake up logic issues
- Reset, clamp control or wake up logic issues are difficult to detect
- These issues lead to missing a wide range of initialization, performance and reset issues by non-PA simulations
- These gaps are successfully detected by PA flow

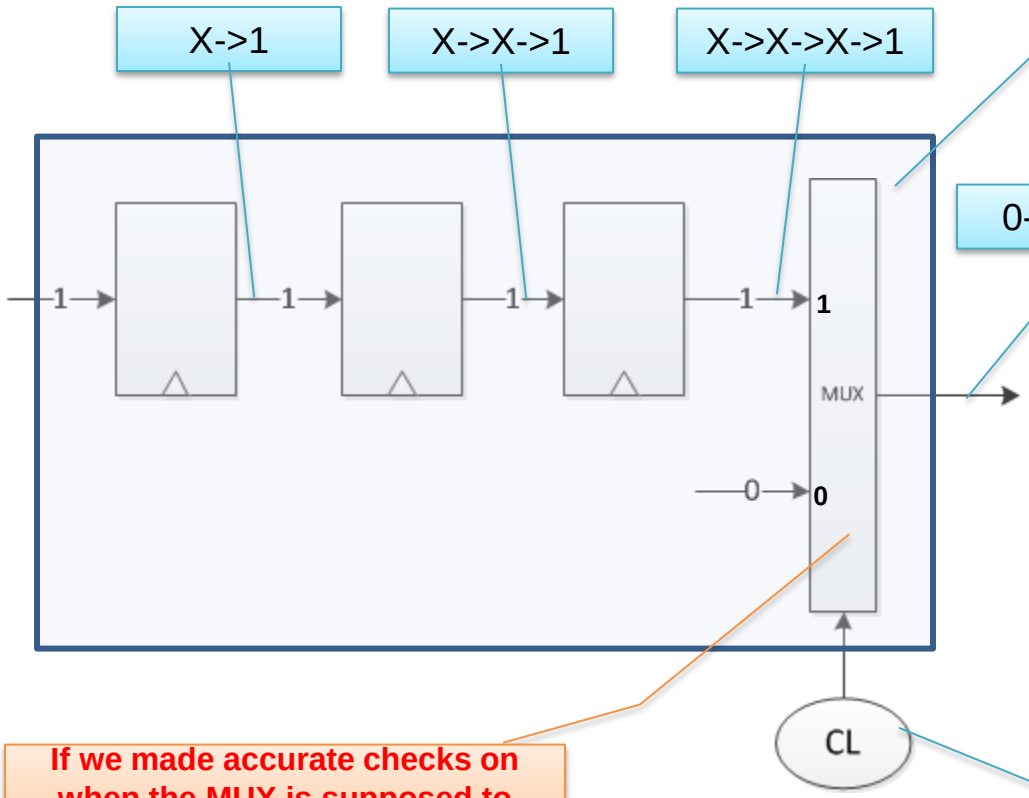

Thank You

Appendix

Incorrect Control

If we had a scenario of propagating 1 before the collapse and 0 after the collapse, we would catch this bug without PA.

By using of PA flow we increase the number of scenarios being covered without using of engineering time



Reset is not done as a part of the architecture

0->0->X->1

Without PA flow the input of the mux would be always 1. The bug wouldn't be caught.

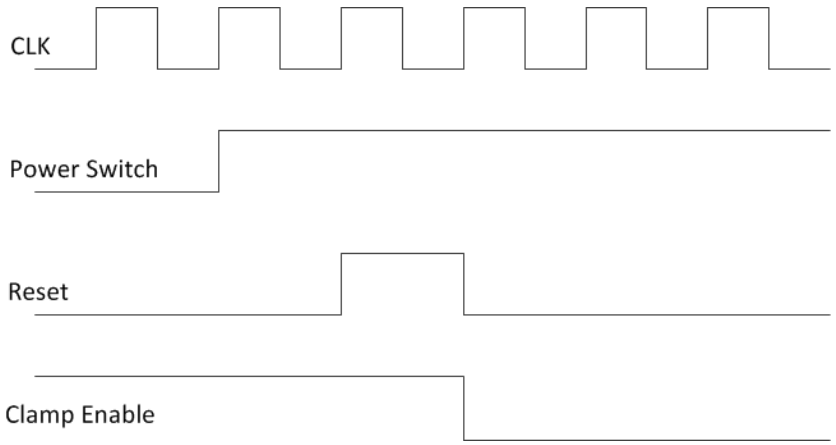
CL selects input 1 prematurely

If we made accurate checks on when the MUX is supposed to switch the control, then we would catch this bug

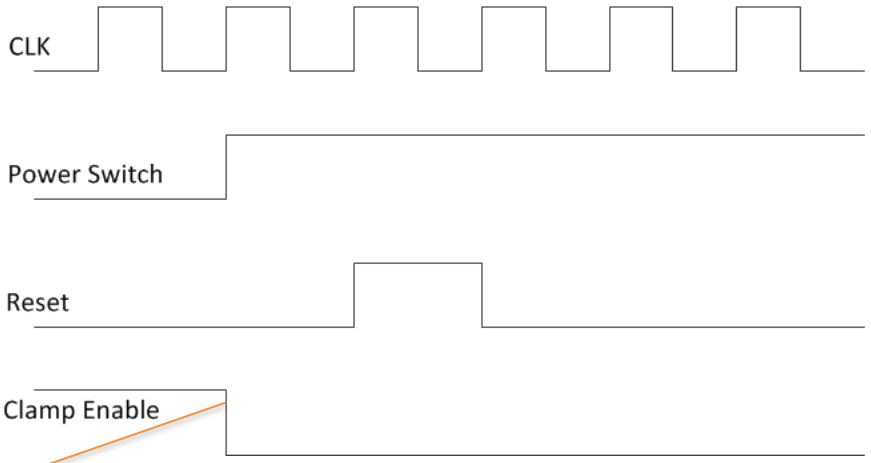
Block Synchronization

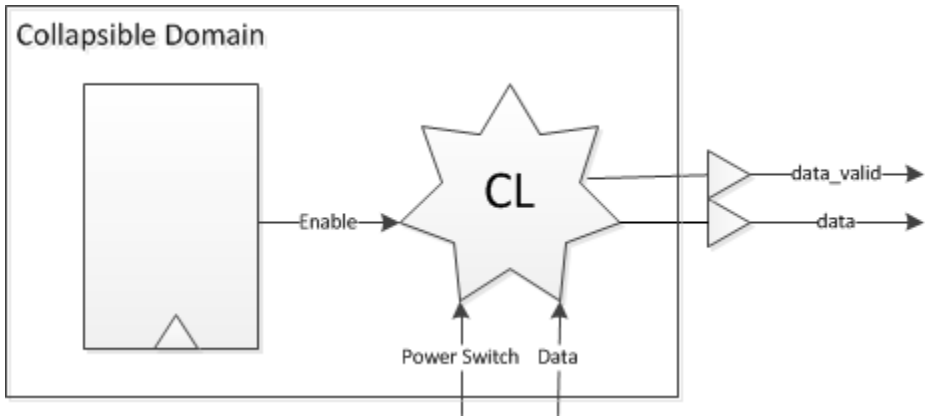
Synopsys Users Group

Correct timing



Current Timing

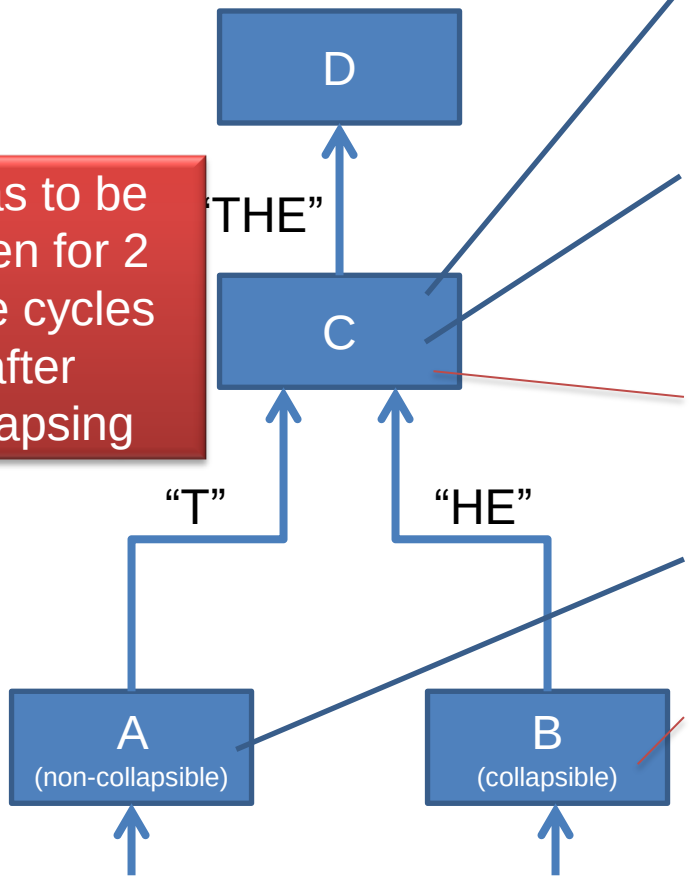




If we verified that clamp enable occurs on time then we would catch this bug

- Enable was kept 1 from before the collapse.
- Enable was supposed to get 1 only after the reset.
- When the power switch is up, CL proceeds with the data coming in.
- Without PA the output data comes out 2 cycles prematurely

Block Synchronization



A has to be frozen for 2 more cycles after collapsing

C merges every 2 consecutive transactions

The arbitration is given to the first transaction, which is ready.
C gives ack, when it reads from a block.
It doesn't read during the collapse.

C merges 2 words from A and instead of "THE" we obtain "TT"

Initially, A starts one cycle before B

B produces output 2 cycles later than in non-PA simulation.

Blocks A and B convert between interfaces.
If A and B provide input in the same cycle A is the first word in concatenation.
B has collapsible power domain and A doesn't.

Without PA flow, B is up 2 cycles earlier than it is supposed to be. This ensures synchronization and the bug are being missed.