AN ALGORITHM TO IDENTIFY CORNERSTONES OF DIGITAL CIRCUITS

Yanxiang Huang, Chunshu Li, Meng Li, Francky Catthoor, Liesbet Van der Perre, Wim Dehaene
Imec, Leuven, Belgium; KU Leuven, Leuven, Belgium
Yanxiang.huang@imec.be

Discuss with authors

Smart Identification Algorithm Needed

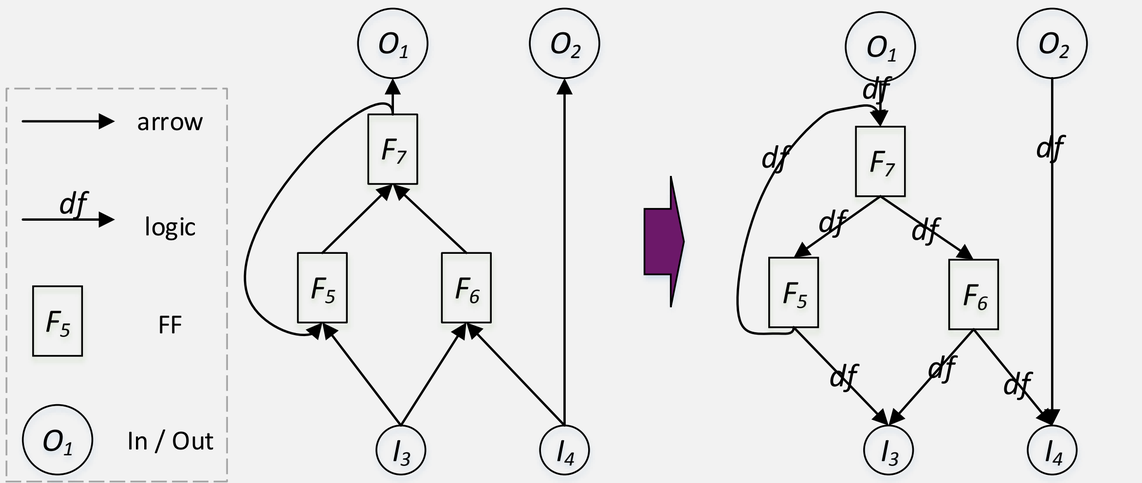
Reliability, the problem
- CMOS circuit impacted by particle strikes
- Technology Scaling increases the vulnerability

Hardware Hardening, the counteract
- Reduce the soft-error on Flip-flop
- However, at the cost of area & power overhead

Selective hardening algorithm, the solution
- Only hardening the most significant FF
- But which FFs are the most significant?

Significance Model at the Netlist Level

Take a circuit (Netlist), abstract it as a *graph of significance*



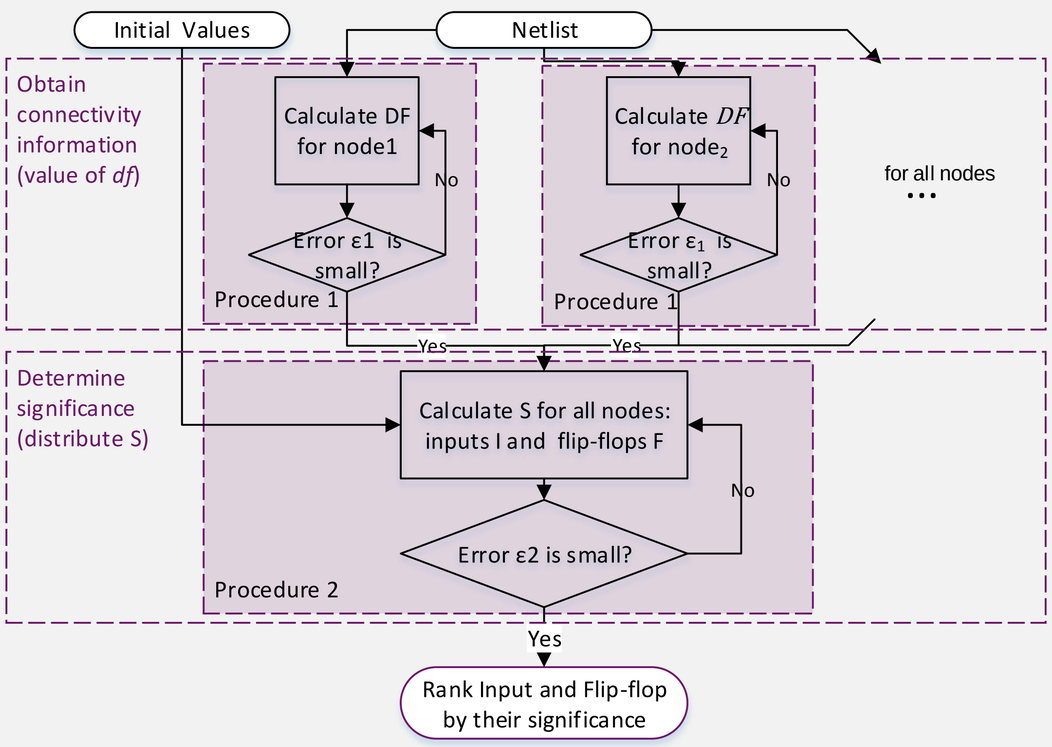
Netlist
inputs flow to the outputs

significance graph
S distributed from outputs to inputs

***Netlist Output**: user defined significance*
***Arrow**: the weight is the significance on the arrow*
***Node** (FF):*
- *significance = SUM(all inwards arrow weights)*
- *error **ε** = significance - SUM(all outwards arrow weights)*

Analytical Algorithm

*Graph travel to distribute the significance, to minimize the **ε***

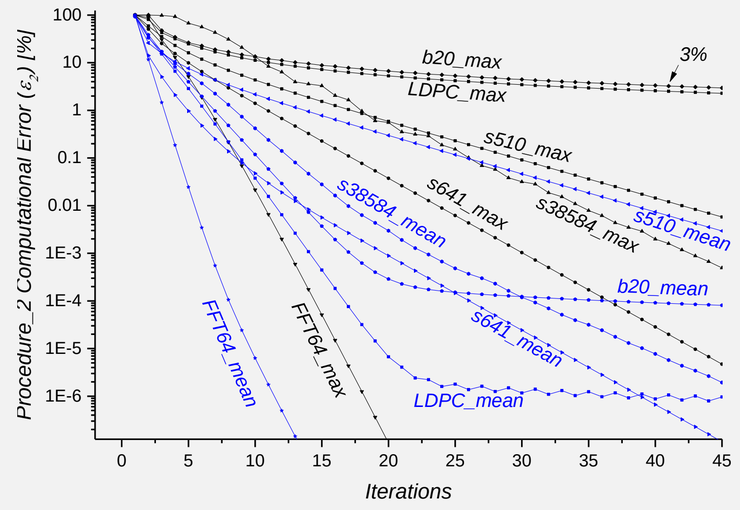


Proc. 1: non-cyclic directed graph with branches and conjunctions
Proc. 2: cyclic directed graph with branches and conjunctions

Algorithm Execution Time

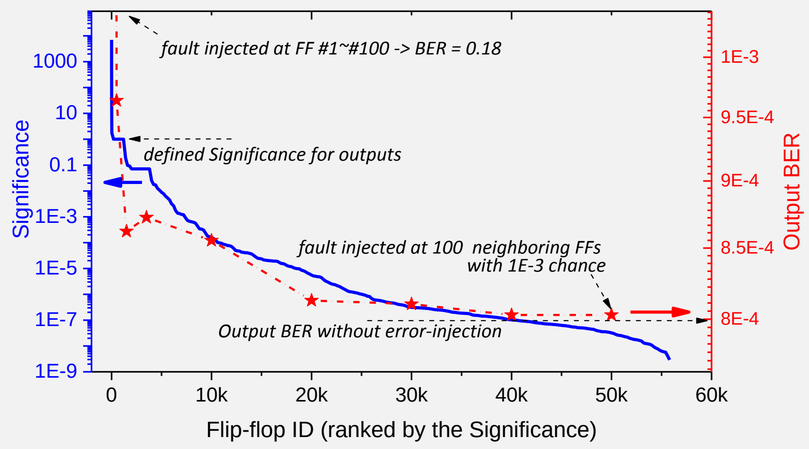
$Proc.1 \sim \#nodes * iter_1 * degree_{node}{}^2$
$Proc.2 \sim iter_2 * \#node * degree_{node}$
$^* degree_{node}$: *average FF a FF is connected to*

Circuits	Circuit complexity			Proce._1 (ε_1 = 0)		Proce._2
	#gate	#node (I/O & FF)	degree_node	mean $iter_1$	Run time	runtime per iter.
s27	28	8	2.6	2.25	0.2 ms	0.1 ms
s510	315	32	3.3	2.8	2 ms	0.8 ms
s641	392	78	6.6	3.5	9 ms	3.4 ms
s5378	3.3k	263	9.3	2.6	39 ms	18 ms
s13207	9.7k	852	6.3	2.6	97 ms	44 ms
s38584	28k	1.8k	12.5	2.8	292 ms	160 ms
b14	9.6k	233	90.7	14.4	5.7 s	158 ms
b20	21k	546	115	14.0	10 s	325 ms
b21	22k	546	129	14.6	12 s	369 ms
FFT64	52k	8.6k	12.6	2.9	6.1 s	965 ms
LDPC	778k	59k	136.9	4.2	1424 s	40 s



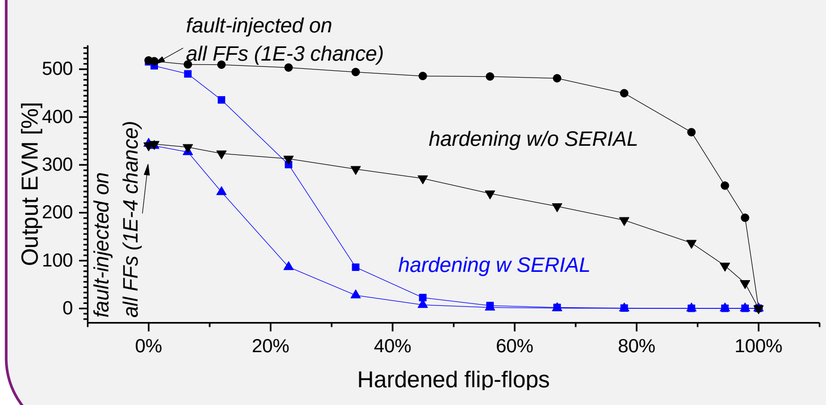
For procedure 2, error reduction when iteration increases

Verification (Significance VS. output quality)



LDPC:
significance of the FFs; & BER (lower is better), when random soft-errors are injected into different FFs

Soft-error on high significance nodes leads to more serious degradation



FFT64:
EVM (lower the better) under soft-error, for selective hardening with SERIAL vs. randomly hardening

Proposed algorithm helps hardening more smartly

Conclusion

An analytical (non-Monte-Carlo) algorithm to **automatically** identify the most important FFs to the output (hence corner-stones).
The cornerstones can be reset, clock-gating controller, mode settings, arithmetic MSBs, etc..
This enables smart hardening, by starting from the cornerstones.

KU LEUVEN imec