The SEASONing Tool:

A Spice Engine for Adding Soft-errors On Netlists

Kevin KleinOsowski AJ KleinOsowski* David J. Lilja

kevinko@ece.umn.edu ajko@ece.umn.edu lilja@umn.edu

Department of Electrical and Computer Engineering

University of Minnesota, Minneapolis, MN 55455

Abstract

In this project, we describe a technique and the associated software tool for injecting and modeling transient radiation events, so-called soft-errors, in Spice netlists. We create a parser written in C to read and analyze a Spice netlist text file. The parser chooses random nodes and adds current pulses to these nodes to mimic the charge generated from a high energy particle strike. The fault density and time windows of the faults are fully configurable. The resulting modified netlist is then simulated using Spice. With our netlist parser, called SEASONing, designers can evaluate the robustness of their circuits. This produces a new metric by which alternative designs may be evaluated for soft-error robustness.

*AJ KleinOsowski is now with the IBM Austin Research Laboratory in Austin Texas.

1 Introduction and Motivation

Single event, radiation-induced upsets, so called *soft-errors*, are becoming more prevalent as process technologies scale to nanometer-scale lithographic nodes [3, 7]. These *soft-errors* occur when a charged particle strikes a transistor's depletion region forming an electrical current which may lead to a change in the transistor's state. These upsets have long been recognized in memory elements, but are now also being observed in combinational logic [10]. Fortunately, this event does not damage the transistor. However, Future non-CMOS device technologies are expected to be even more susceptible to these types of transient faults [6].

Past work on fault modeling has been conducted at the logic level or system level [4, 9] using hardware description language (HDL) simulations. Other work includes analytical models for studying soft-errors in individual gates [8]. Errors can also be injected by physical means with lasers or high energy particles [2] and then observed with electrical testing equipment. The previous work has left a void which must be filled by a method to test circuits for robustness at a lower gate level. A tool to simulate faults at the gate level will offer the opportunity to test logic designs intended to circumvent the *soft-errors* problem.

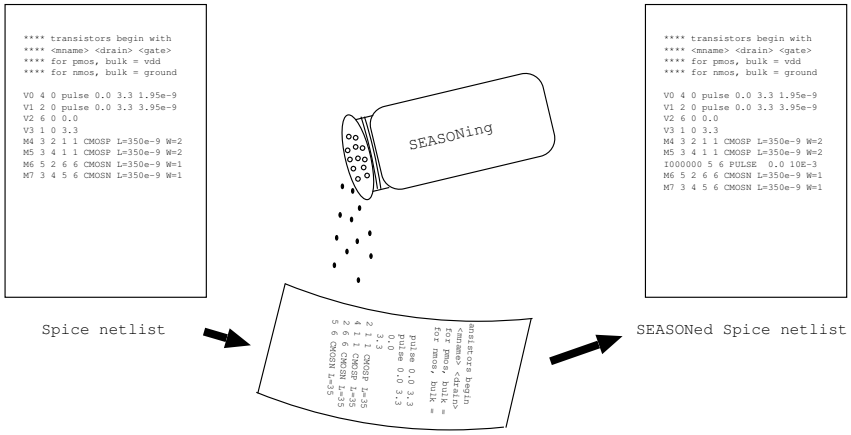

Figure 1: Faults injected in Spice netlist.

This work addresses the gate level fault simulation by producing an automated method for injecting multiple errors in small and medium sized circuit blocks. In this way, the robustness of circuit blocks may be quickly studied in the presence of a wide range of error densities and error time periods. This would produce a scale or metric by which a circuit block may be measured as being robust, thus producing a new dimension by which a circuit block is designed. The tool we designed for this purpose is named SEASONing, which is an acronym for a **S**pice **E**ngine for **A**dding **S**oft-errors **O**n **N**etlists. The process of inserting faults into a netlist is pictured in Figure 1.

2 Methodology and Results

The heart of the SEASONing tool is a parser written in C which reads in a flat Spice [1] netlist text file and then adds current pulses to randomly chosen nodes. To ensure the errors are randomly distributed, the netlist must be completely flattened, with no hierarchical elements. If the netlist uses hierarchical elements, all instances of the hierarchical element will have identical soft-errors, which is not representative of a physical system.

SEASONing is designed to be an intelligent tool that can understand Spice netlists. This intelligent design ensures that SEASONing is capable of determining what type of device it has found while reading through the netlist. It may then act accordingly to either inject a fault in the device or pass it through unaltered. SEASONing can determine the type of device it has found by reading the netlist, line-by-line, to tokenize the line into it's constituent parts. The parts of the line, from the netlist, are then compared with a known device. A known device is one in which SEASONing has been taught how to modify to insert a fault. Other tokens and lines of the netlist which SEASONing

```
.MODEL CMOSN NMOS ( LEVEL = 49
+VERSION = 3.1 TNOM = 27 TOX = 7.9E-9
+XJ = 1E-7 NCH = 2.2E17 VTH0 = 0.5296871
+K1 = 0.5553031 K2 = 0.0188221 K3 = 1E-3
+K3B = -6.633858 W0 = 1.508615E-5 NLX = 1.599745E-7
+DVT0W = 0 DVT1W = 0 DVT2W = 0
+DVT0 = 5.9211628 DVT1 = 0.9666357 DVT2 = -0.1012943
+U0 = 435.125839 UA = -6.20816E-12 UB = 1.65843E-18
+UC = 3.771653E-11 VSAT = 1.653881E5 A0 = 1.2133639
+AGS = 0.1816647 B0 = 1.111541E-6 B1 = 5E-6
+KETA = 7.498967E-3 A1 = 0 A2 = 0.4390819
+RDSW = 1.247094E3 PRWG = -0.0375101 PRWB = -0.0625955
+WR = 1 WINT = 7.318991E-8 LINT = 1.194415E-8
+XL = -2E-8 XW = 0 DWG = -7.467457E-9
+DWB = 4.74892E-9 VOFF = -0.0899316 NFACTOR = 1.2479012
+CIT = 0 CDSC = 2.4E-4 CDSCD = 0
+CDSCB = 0 ETA0 = 0.4774212 ETAB = 6.23259E-3
+DSUB = 0.76337 PCLM = 1.4589833 PDIBLC1 = 7.891827E-4
+PDIBLC2 = 4.549746E-3 PDIBLCB = -1E-3 DROUT = 0.0582836
+PSCBE1 = 5.907355E8 PSCBE2 = 8.560473E-5 PVAG = 0
+DELTA = 0.01 RSH = 3.3 MOBMOD = 1
+PRT = 0 UTE = -1.5 KT1 = -0.11
+KT1L = 0 KT2 = 0.022 UA1 = 4.31E-9
+UB1 = -7.61E-18 UC1 = -5.6E-11 AT = 3.3E4
+WL = 0 WLN = 1 WW = 0
+WWN = 1 WWL = 0 LL = 0
+LLN = 1 LW = 0 LWN = 1
+LWL = 0 CAPMOD = 2 XPART = 0.5
+CGDO = 2.65E-10 CGSO = 2.65E-10 CGBO = 1E-12
+CJ = 1.021865E-3 PB = 0.7353904 MJ = 0.3130423
+CJSW = 3.591995E-10 PBSW = 0.7380293 MJSW = 0.1257799
+CJSWG = 1.82E-10 PBSWG = 0.7380293 MJSWG = 0.1257799
+CF = 0 PVTH0 = -0.017651 PRDSW = -125.9988543
+PK2 = 1.679174E-3 WKETA = -6.913439E-3 LKETA = -3.052378E-3 )
```

Figure 2: Model name reference.

does not understand will simply pass through without modification. This decision to pass unknown statements through without modifications makes SEASONing robust in its ability to insert faults into a wide range of different netlists. This design also allows SEASONing to be extended for uses outside of transistor fault simulation. SEASONing may be extend to inject faults into netlists on a

```
#!/usr/local/bin/ksh
#usage:  season -f<filename> -t<totalTime> -c<change>
# -p<percent error> -d<delay>
# filename - name of file to SEASON, output will be to stdout
# totalTime - time in nanoseconds for the entire simulation run
# change - time in nanoseconde between faults changing
# percent error - the percentage of error for all the nodes in
# netlist, per change period
# delay - time in nanoseconds to delay before the first
# fault period begins
./season -f nand2.sp -t 8 -c 8 -p 25 -d 3 1> $1
```

Figure 3: SEASONing usage.

wide array of elements such as wires, connection points, voltage sources, or current sources.

SEASONing looks specifically for transistors when determining where and how to create a fault. Only *NMOS* and *PMOS* transistors are used for the fault simulation. Most tools that generate netlists do not explicitly use the *NMOS* and *PMOS* elements. Instead the tools generate model name references with the Spice *.MODEL* Statement. The *.MODEL* Statement uses a macro type replacement for the transistors to reduce the clutter of parameters that are needed with the *NMOS* and *PMOS* elements. An example of one of these model name references for a *NMOS* transistor is listed in Figure 2. As shown, the model name is given to be *CMOSN*. It is clear that this is a *NMOS* transistor in this example. However, it is possible that a name such as *MYTRAN* is used for the model name. This makes it difficult to determine the type of transistor that is present. The type of transistor is important to know for simulating a fault.

A fault is simulated differently depending on the transistor type selected for the fault. A fault on a *NMOS* transistor is simulated with a current sent into the drain of the transistor and out the body of the transistor. The fault on a *PMOS* transistor is simulated with a current entering the body of the device and exiting the drain. Because of the difference in how the faults are simulated,

```
**** transistors begin with 'M'
**** <mname> <drain> <gate> <source> <bulk>
**** voltage sources begin with 'V'
**** current sources begin with 'I'
* Vdd
V3 1 0 3.3
* Gnd
V2 6 0 0.0
* Input A
V0 4 0 pulse 0.0 3.3 1.95e-9 100e-12 100e-12 1.9e-9 4e-9
* Input B
V1 2 0 pulse 0.0 3.3 3.95e-9 100e-12 100e-12 3.9e-9 8e-9
* Nand2
M4 3 2 1 1 CMOSP L=350e-9 W=2.8e-6
M5 3 4 1 1 CMOSP L=350e-9 W=2.8e-6
M6 5 2 6 6 CMOSN L=350e-9 W=1.4e-6
M7 3 4 5 6 CMOSN L=350e-9 W=1.4e-6
```

Figure 4: Original netlist.

the type of a transistor must be know before any faults are injected. To determine all types of transistors that are possible in the netlist, a compiler technique employing multiple passes of the netlist is used to resolve forward references to these model names. This technique uses an initial pass through the netlist to find all model name references and the transistor types associated with the model names.

The first pass through the netlist finds all the *.MODEL* information. The transistor type is stored along with the model name and is used to resolve the transistor types on the second pass through the netlist. The first pass also counts the number of transistors in the netlist and gathers them into a list of transistors. Transistors start with the letter *M* and are numbered uniquely. The tally of transistors along with the transistor names are used on the second pass when determining options such as percentage of total transistors in error.

In the second pass through the netlist, SEASONing is armed with information to resolve the

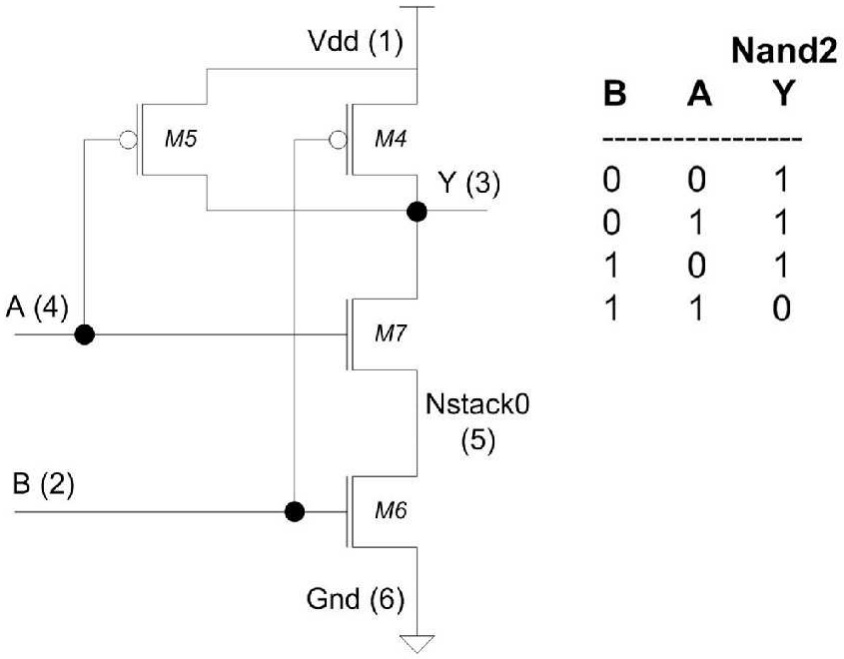

Figure 5: Schematic and truth table of the two input nand gate used in SEASONing detailed analysis and verification.

transistor types. SEASONing also knows the total number of transistors in the netlist and the names of the transistors. With this information SEASONing may then determine how many errors to inject each fault period and which transistor to be faulty. This is done by multiplying the total number of transistors by the fault density, which is passed to SEASONing as a command line parameter. The time between faults and the delay before the first fault can also be specified as command line parameters. During each fault period, a different collection of transistors is chosen to be at fault. All fault periods use the same fault density. Figure 3 shows the SEASONing usage statement with a description of each command line parameter. It is important to note that the fault periods are not the length of the entire circuit simulation. Rather, the fault periods are typically shorter. This produces the effect of a short period during the simulation when a transistor fault occurs. After this fault, the circuit behaves as it would have given there were no fault.

```
* SEASON: Notice, no change interval change:  8 totalTime:  8
* SEASON: parsed to End of file.
* SEASON: 1 time periods starting at 3
* SEASON: Number of transistors:  4
* SEASON: Maximum value of transistor ID: 7
* SEASON: Each fault period will have 1 nodes with faults
**** transistors begin with 'M'
**** <mname> <drain> <gate> <source> <bulk>
**** voltage sources begin with 'V'
**** current sources begin with 'I'
* Vdd
V3 1 0 3.3
* Gnd
V2 6 0 0.0
* Input A
V0 4 0 pulse 0.0 3.3 1.95e-9 100e-12 100e-12 1.9e-9 4e-9
* Input B
V1 2 0 pulse 0.0 3.3 3.95e-9 100e-12 100e-12 3.9e-9 8e-9
* Nand2
M4 3 2 1 1 CMOSP L=350e-9 W=2.8e-6
M5 3 4 1 1 CMOSP L=350e-9 W=2.8e-6
I000000 5 6 PULSE 0.0 10E-3 3.000000E-09 1E-12 1E-12 8E-12
M6 5 2 6 6 CMOSN L=350e-9 W=1.4e-6
M7 3 4 5 6 CMOSN L=350e-9 W=1.4e-6
```

Figure 6: SEASONed netlist.

Errors are injected by inserting a current pulse between the drain and body of a transistor. For simplicity, all current pulses contain 100 femtocoulombs of charge, shaped as a trapezoidal pulse with 1 picosecond rise and fall times. This pulse is representative of the charge delivered to a transistor when the transistor is impacted by a high energy neutron particle [5].

To illustrate SEASONing's usage, we show the netlist for a two input nand gate in Figure 4. The transistor schematic, along with the logical truth table, is shown in Figure 5. We add one fault, with one fault period, using the command line shown in Figure 3. The modified netlist is shown in Figure 6. This netlist shows the addition of a single current source attached to the transistor *M6*. This is an *NMOS* transistor so the fault is simulated with the current entering the drain and leaving

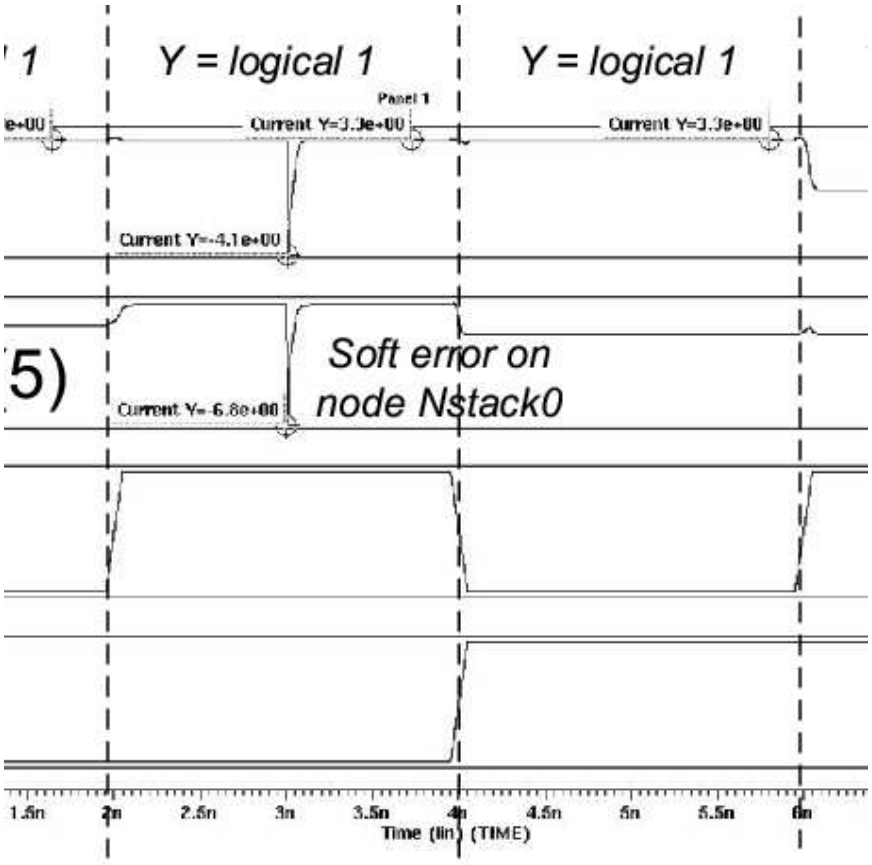

Figure 7: Simulated waveform showing a soft-error injected on node Nstack0 of the gate shown in Figure 5.

the body.

We simulate the SEASONed netlist using Hspice and show the waveforms in Figure 7. Figure 8 shows a magnified view of the fault disturbance waveform. In Figure 8 we see the expected linear drop in voltage as charge is delivered to the node, with an exponential recovery from the fault disturbance. In many of the simulations we ran, extra capacitance was added to all nodes to make the simulation converge. The was done with the command *.option cshunt = 1.00000e-15*.

The soft-error events added by the SEASONing tool incur minimal simulation time overhead, as listed in Figure 1. In addition to logic gates such as the two input nand, we ran SEASONing

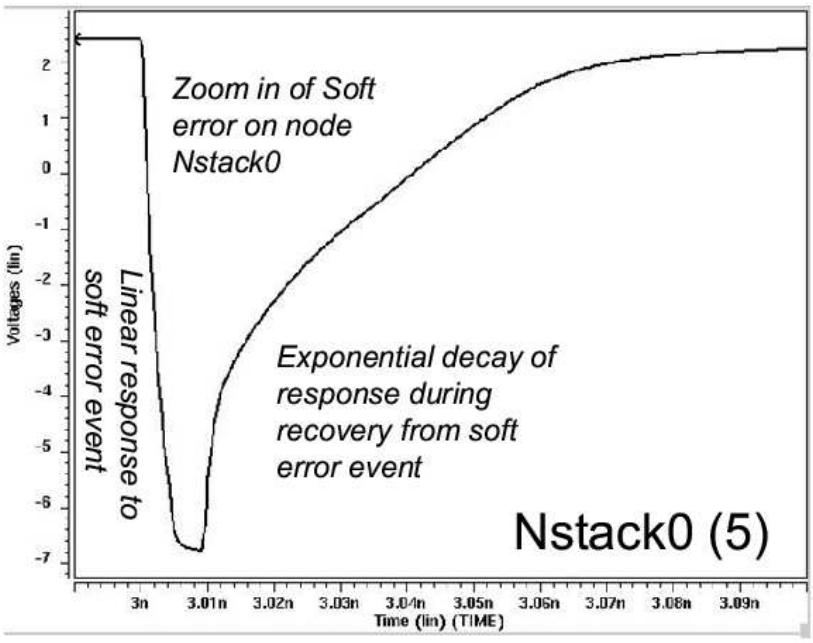

Figure 8: Magnified view of the soft-error event waveform shown in Figure 7.

with a bitslice of a four instruction arithmetic logic unit (ALU) containing 52 transistors. Our worst case run of the ALU netlist with 416 injected faults had a 15 percent simulation time overhead. These results demonstrate that SEASONing is a configurable and fast way to evaluate the effects of soft-errors on small and medium sized netlists.

Table 1: Simulation Runtime Overhead of SEASONed netlists.

netlist	errors	normalized simtime
nand2	1	1.0
nand2	2	1.0
nand2	4	1.06
alu	1	1.0
alu	2	1.02
alu	4	1.0
alu	416	1.15

3 Conclusion

In this project, we describe a technique and the associated software tool, called SEASONing, for injecting and modeling transient radiation events in Spice simulations. SEASONing is a parser written in C which reads and analyzes a Spice netlist text file. The parser chooses random nodes and then adds current pulses representative of radiation induced soft-errors. It is also extensible for testing other circuit error conditions. SEASONing creates a new metric for comparing alternative circuit designs for soft-error robustness. The SEASONing tool is open source, available for download at `http://www.arctic.umn.edu/seasoning`.

4 Acknowledgements

This work was supported in part by IBM, Semiconductor Research Corporation contract no. 2004-HJ-1190, National Science Foundation grant CCR-0210197, the University of Minnesota Digital Technology Center, and the Minnesota Supercomputing Institute.

References

[1] W. Banzhaf. *Computer-Aided Circuit Analysis Using SPICE*. Prentice Hall, 1989.

[2] S. Buchner, D. McMorrow, A. Sternberg, L. Massengill, R.L. Pease, and M. Maher. Single-Event Transient (SET) Characterization of an LM119 Voltage Comparator: An Approach to SET Model Validation Using a Pulsed Laser. *IEEE Transactions on Nuclear Science*, 49(3):1502–1508, June 2002.

[3] Paul E. Dodd, Marty R. Shaneyfelt, James R. Schwank, and Gerald L. Hash. Neutron-Induced Latchup in SRAMs at Ground Level. In *IEEE International Reliability Physics Symposium (IRPS)*, pages 51–55, 2003.

[4] P. Gawkowski and J. Sosnowski. Evaluation of Transient Fault Susceptibility in Microprocessor Systems. In *Proceedings of the EUROMICRO Systems on Digital System Design (DSD)*, pages 432–439, 2004.

[5] M. S. Gordon, P. Goldhagen, K. P. Rodbell, T. H. Zabel, H. H. K. Tang, J. M. Clem, and P. Bailey. Measurement of the Flux and Energy Spectrum of Cosmic-Ray Neutrons. *IEEE Transactions on Nuclear Science*, 51(6):3427–3434, December 2004.

[6] AJ KleinOsowski and David J. Lilja. The NanoBox Project: Exploring Fabrics of Self-Correcting Logic Blocks for High Defect Rate Molecular Device Technologies. In *IEEE Symposium on VLSI (ISVLSI)*, February 2004.

[7] Subhasish Mitra, Norbert Seifert, Ming Zhang, Quan Shi, and Kee Sup Kim. Robust System Design with Built-In Soft-Error Resilience. *IEEE Computer*, pages 43–52, February 2005.

[8] Kartik Mohanram. Closed-form simulation and robustness models for SEU-tolerant design. In *VLSI Test Symposium (VTS)*, 2005.

[9] Donald Shaw, Dhamin Al-Khalili, and Come Rozon. Fault security analysis of CMOS VLSI circuits using defect-injectable VHDL models. *The VLSI Journal of Integration*, 32:77–97, 2002.

[10] Premkishore Shivakumar, Michael Kistler, Stephen W. Keckler, Doug Burger, and Lorenzo Alvisi. Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic. In *International Conference on Dependable Systems and Networks (DSN)*, 2002.