Scalable Calculation of Logical Masking Effects for Selective Hardening Against Soft Errors*

Ilia Polian* Sudhakar M. Reddy** Bernd Becker*

*Albert-Ludwigs-University
Georges-Köhler-Allee 51
79110 Freiburg im Breisgau, Germany
{polian|becker}@informatik.uni-freiburg.de

**ECE Department
University of Iowa
Iowa City, IA, USA
reddy@engineering.uiowa.edu

Abstract

Selective hardening aims at achieving maximal soft error rate reduction at reasonable cost by applying hardening techniques to most susceptible circuit nodes only. Logical, electrical and latching-window masking effects must all be considered when calculating the susceptibility of circuit nodes to soft errors. We introduce a scalable selective hardening method based on an approximate calculation of fault detection probabilities at the nodes. Error probability reduction comparable to that obtained by the exact BDD-based algorithm (which is not scalable) can be achieved by setting an over-ambitious optimization target. The run times are negligible even for industrial multiple-million-gates circuits. Existing approaches for calculating electrical and latching-window masking can be readily incorporated into the framework.

Keywords: Soft error protection, Selective hardening

1 Introduction

Soft errors caused by ionizing radiation have been identified as a challenge for nanoscale electronics [1–6]. Hardening of circuit's components can be used to reduce or eliminate occurrence of errors at circuit's outputs due to soft errors. Since hardening the complete circuit against soft errors may result in unacceptable area and power consumption cost, it has been suggested to perform *selective hardening*, i.e., to harden only a subset of the circuit nodes with the largest contribution to the soft error rate (SER) [7–13].

Determining the probability of occurrence of errors at circuit outputs due to soft errors requires the following information: the probability of occurrence and the probability density function of the energy of the cause (e.g. radiation); the effect of the incidence on a circuit node involved; and the probability p_{err} that the soft error at the affected circuit node causes an error at some circuit output(s). It is also necessary to know the probability distribution of circuit inputs.

The probability p_{err} is determined by the node's *masking properties*. Three masking mechanisms are known: logical masking, latching-window masking and electrical masking [5]. Logical masking occurs due to the lack of a sensitized path from the node to an output or flip-flop, latching-window masking is present when the soft error of a duration less than a clock cycle does arrive at an output or flip-flop but not during the strobe or latching window, and electrical masking refers to the gradual attenuation of the soft error as it propagates through the combinational logic.

Masking measures filtering of soft errors by circuits and preventing errors to occur at the circuit outputs. In this paper, we investigate methods to calculate effects of logical masking. If only logical masking is considered, one could expect that the masking provided by a circuit is underestimated. This can be seen by the following. In computing logical masking one computes the probability that errors (faults) caused by soft errors at circuit nodes are propagated to circuit outputs. Electrical and latching-window masking computations filter out some of these errors since they may not propagate to circuit outputs even though sensitized paths exist from affected nodes to circuit outputs and/or because the errors do not arrive during the time window of output observation or latching of next states in memory elements. In this work we use logical masking to guide selection of circuit nodes to be hardened to increase masking of soft errors by a given circuit.

We use what we call *derating* as a measure of logical masking by a circuit. Derating D of a circuit is defined as the probability that a soft error occurs somewhere in the circuit divided by the probability that a soft error has arrived at an output or flip-flop. i.e., the conditional probability that an error which has occurred has *not* been masked. Hardening a node improves the circuit's derating. We calculate a cost-optimal subset of a circuit's nodes such that hardening these nodes increases the derating of the circuit above a predefined target D^{target}.

---

*This work was supported in part by the Alexander von Humboldt Foundation and the DFG Project RealTest under Grant BE 1176/15-1. We are thankful to J. Schlöffel of NXP for providing the industrial benchmarks.

The method we propose in this paper uses signal probabilities which can be computed efficiently. Additionally it can be applied to sequential circuits. However, in order to compare with the earlier work [10], we consider only combinational circuits in this work. We benchmark the proposed method against the exact approach from [10], which achieves optimal results for logical masking but is not scalable.

The remainder of the paper is organized as follows. The node selection problem is formulated and the assumptions are detailed in Section 2. The approximate heuristic is introduced in Section 3. The evaluation flow is described and the experimental results are reported in Section 4. Section 5 concludes the paper.

2 Node Selection Problem

Selective hardening is done in three steps. First, the contribution of the individual nodes in the circuit to the overall SER is calculated. Second, a sub-set of the nodes is selected for hardening such that either a cost criterion (e.g., number of selected nodes) or a quality criterion (e.g., a maximal soft error rate or a minimal soft error rate reduction) is met. Finally, the hardening mechanism such as gate duplication is applied to the selected nodes.

A complete selective hardening strategy should take all three masking mechanisms as well as other information mentioned above into account. We consider single gates as entities which could be hardened or left unprotected. Probability p_{err} for a gate g is calculated as

$$p_{err}(g) = (1 - p_{lm}(g)) \cdot (1 - p_{lwm}(g)) \cdot (1 - p_{em}(g)), \quad (1)$$

where $p_{lm}(g)$, $p_{lwm}(g)$ and $p_{em}(g)$ are the probabilities of logical masking, latching-window masking and electrical masking of gate g, respectively. Strictly speaking, $p_{err}(g)$ can not be decomposed as given in the equation above, but one can use this equation to lower-bound $p_{err}(g)$ by including the appropriate conditions on the computation of $p_{lm}(g)$, $p_{lwm}(g)$ and $p_{em}(g)$. We introduce an approximate method to calculate probability p_{lm}. We do not calculate probabilities p_{lwm} and p_{em} and set them to 0. A significant body of literature is available on calculating these probabilities [7–9]. We emphasize that we are not aiming at calculating accurate soft error rates.

We model a soft error by a pair (g, d), where g is the gate and d is the direction of the flip (0 or 1). The number of possible soft errors is twice the number of gates. If soft error $(g, 0)$ takes place when g's output is 0, no error is observed.

Under the assumptions outlined above, the probability that soft error $(g, 1)$ will have an observable effect at an output is given by $DP1(o(g))$, where $o(g)$ is the gate g's output and $DP1(o(g))$ is the detection probability of stuck-at-1 fault at line $o(g)$. Symmetrically, the probability that soft error $(g, 0)$ will have an observable effect is $DP0(o(g))$, the detection probability of stuck-at-0 fault at line $o(g)$.

For a circuit C with N logic gates, let G denote the set of that gates. p_{err}, the conditional probability that any error in circuit C will have an observable effect, equals

$$p_{err}(C) = \frac{1}{2N} \cdot \sum_{g \in G} \left(DP0(o(g)) + DP1(o(g)) \right). \quad (2)$$

We assume that selective hardening of a node eliminates the probability of a soft error at this node altogether. Moreover, we assume that there is a hardening mechanism against the flip-to-0 fault (the soft error which sets the node to logic-0) and a separate mechanism against the flip-to-1 fault on the same node. Let $SH = \{(g_1, d_1), (g_2, d_2), \ldots\}$ be the set of soft errors which are selected for hardening. The probability p_{err} for hardened circuit C_{SH} becomes

$$p_{err}(C_{SH}) = \frac{1}{2N} \cdot \sum_{\substack{g \in G, \\ (g,0) \notin SH}} DP0(o(g)) + \frac{1}{2N} \cdot \sum_{\substack{g \in G, \\ (g,1) \notin SH}} DP1(o(g)). \quad (3)$$

From the above definition of derating,

$$D(C) = \frac{1}{p_{err}(C)}, \qquad D(C_{SH}) = \frac{1}{p_{err}(C_{SH})} \quad (4)$$

holds. The probability that a soft error when it occurs produces an error at a circuit output is reduced by factor D. Smaller values of p_{err} and thus larger values of D correspond to a larger probability that a soft error is masked. The node selection problem is formalized as follows: given a derating target D^{target}, find a minimal set SH of soft errors such that $D(C_{SH}) > D^{target}$ according to Eq. (4).

3 Scalable Solution

To transform a circuit C which violates the derating target D^{target} into the circuit C_{SH} which satisfies the target, the set SH of soft errors to be addressed by hardening is calculated based on approximate stuck-at-1 and stuck-at-0 fault detection probabilities $DP1$ and $DP0$. We use a simple heuristic to compute the probabilities. The heuristic ignores signal correlations but it requires only linear run time in the size of the circuit.

Once the detection probabilities are available, the set SH is constructed as follows. Initially, SH is empty. Let (g, d) be the soft error such that $DPd(o(g))$ is maximal and thus has the largest impact in Eq. (2). (g, d) is included into SH and the soft error with the largest associated detection probability is selected. This is iterated as long as $D(C_{SH})$ is below the target D^{target}. To facilitate this process, the list of possible soft errors $(o(g_1), 0), (o(g_1), 1), \ldots, (o(g_N), 0), (o(g_N), 1)$ is sorted according to the detection probabilities.

3.1 Approximate calculation of detection probabilities

Scalable calculation of detection probabilities is done in two passes. First, the signal probability $SP(i)$ is determined for every line i in the circuit. The calculation of values $SP(i)$ processes the circuit's gates in the topological order. It starts at the inputs by setting $SP(i)$ to 0.5 for all inputs i. If i is the output of a gate with one input j, $SP(i)$ is set to $SP(j)$ for a buffer and $(1 - SP(j))$ for an inverter.

If i is the output of a two-input gate with inputs j and k, $SP(i)$ is set to $SP(j) \cdot SP(k)$ for an AND gate, $SP(j) + SP(k) - SP(j) \cdot SP(k)$ for an OR gate and $SP(j) \cdot (1 - SP(k)) + (1 - SP(j)) \cdot SP(k)$ for an XOR gate. Gates with more than two inputs are treated as cascades of two-input gates; NAND, NOR and XNOR gates are treated as an AND, OR or XOR gate followed by an inverter. All fanout branches are assigned identical signal probability values.

Once all signal probabilities are available, detection probabilities $DP0(i)$ and $DP1(i)$ are calculated in the reverse topological order. For an output i, $DP0(i)$ is set to $SP(i)$ and $DP1(i)$ is set to $(1 - SP(i))$. For input j of a buffer the values $DP0(j)$ and $DP1(j)$ are assigned the same values as its output i. In case of an inverter, $DP0(j)$ is set to $DP1(i)$, and vice versa. $DP1(j)$ and $DPO(j)$ of an input j of an AND gate with output i and l side-inputs $k_1, \dots, k_l$ are set to $DP1$ and $DP0$ of its output, respectively, both multiplied by the product $SP(k_1) \cdots SP(k_l)$ to account for the probability of fault effect propagation through the gate.

For a NAND gate, $DP0(i)$ of its input i is set to $DP1(j)$ of its output j multiplied by $SP(k_1) \cdots SP(k_l)$ of its l side-inputs $k_1, \dots, k_l$ while $DP1(i)$ is set to $DP0(j) \cdot SP(k_1) \cdots SP(k_l)$. For OR and NOR gates, $SP(k_1) \cdots SP(k_l)$ is replaced by $(1 - SP(k_1)) \cdots (1 - SP(k_l))$. All inputs j of an XOR gate are assigned the same $DP0(j)$ and $DP1(j)$ as its output i.

$DP0$ and $DP1$ of a fanout stem are set to the maximal value among all the branches, respectively (this is similar to STAFAN [14] with parameter α set to 0).

3.2 Relation to exact method

Calculation of defect probabilities outlined above introduces inaccuracies due to the handling of reconvergencies. Already early fault detection probability estimators such as STAFAN [14] and PROTEST [15] suffered from similar difficulties [16]. It is possible to calculate signal probabilities exactly (e.g., using binary decision diagrams), yet such approaches do not scale well. In the following, we discuss the implications of using defect probabilities calculated using an approximate method (we denote them $DP0_{ap}$ and $DP1_{ap}$) instead of exact signal probabilities $DP0_{ex}$ and $DP1_{ex}$ on the quality of the solution.

In general, these implications appear to be less severe for the considered node selection problem compared with other tasks such as fault coverage estimation. First, the approximated detection probabilities are not used as absolute numbers but rather for comparison with each other and subsequent sorting. If the detection probabilities are systematically over- or underapproximated for a circuit, all values are affected and the effect on the relative order may not be large.

Second, even if the order is affected the impact on the derating may not be grave. Suppose that a soft error (g', d') has been selected for hardening instead of the "optimal" soft error (g, d). This means that $DPd'_{ap}(o(g')) \geq DPd_{ap}(o(g))$ while $DPd'_{ex}(o(g')) < DPd_{ex}(o(g))$. The hardening against the "wrong" soft error does improve the derating (although not as strong as would have been possible), in particular when $(DPd_{ex}(o(g)) - DPd'_{ex}(o(g')))$ is small.

A further difficulty associated with the usage of approximate detection probability is the calculation of error probability p_{err} and derating D (Eqs. (2) and (4)). We call derating calculated using approximate detection probabilities $DP0_{ap}$ and $DP1_{ap}$ *approximate derating* D_{ap} and derating calculated using exact detection probabilities $DP0_{ex}$ and $DP1_{ex}$ *exact derating* D_{ex}. D_{ap} is an approximation of D_{ex}.

It is obvious that the derating target D^{target} is defined for the exact derating. A solution of the node selection problem, i.e., a set of soft errors SH, should satisfy condition

$$D_{ex}(C_{SH}) > D^{target}. \tag{5}$$

However, D_{ex} cannot be obtained if exact detection probabilities are not available. In this case, the condition

$$D_{ap}(C_{SH}) > D^{target} \tag{6}$$

must be used instead. A solution satisfying (6) does not have to satisfy (5). To compensate for this possible overestimation of the solution quality, we set an over-ambitious derating target for the approximate method by multiplying D^{target} with a *safety margin* $SM \geq 1$. We discuss the influence of SM on the robustness of the results in Section 4.

4 Experimental Results

4.1 Approximate vs. exact solution

For combinational circuits of limited size, the approximate algorithm proposed in this work and the exact algorithm from [10] can both be run. Let SH_{ap} and SH_{ex} be the solutions found by that algorithms, respectively. Solution SH_{ex} guarantees that $D_{ex}(C_{SH_{ex}}) > D^{target}$ holds. Solution SH_{ap} ensures condition $D_{ap}(C_{SH_{ap}}) > D^{target} \cdot SM$, where SM is the safety margin. As discussed above, $D_{ap}(C_{SH_{ap}})$ may overestimate the actual quality of solution $D_{ex}(C_{SH_{ap}})$. We study experimentally how large the safety margin must be such that $D_{ex}(C_{SH_{ap}}) > D^{target}$ holds.

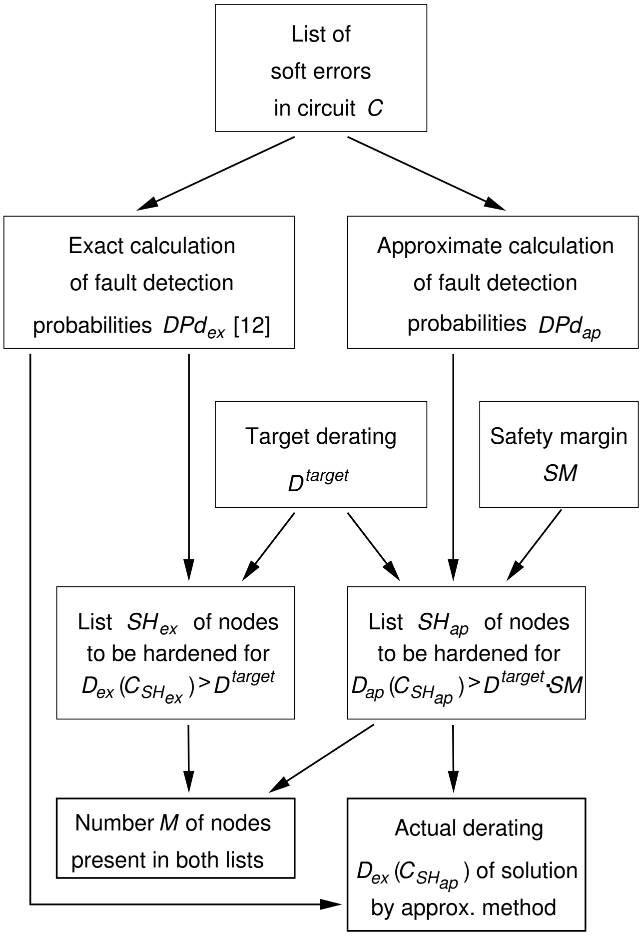

Figure 1: Flow of the evaluation of the approximate method

Figure 1 details the flow of the evaluation. Given a list of soft errors, approximate and exact detection probabilities and solutions SH_{ex} and SH_{ap} (for different values of SM) are calculated. The actual quality of an approximate solution SH_{ap} is determined as $D_{ex}(C_{SH_{ap}})$. Note that exact detection probabilities are required to evaluate the approximate solution SH_{ap} but not to obtain this solution. Furthermore, we compare the selections of the nodes found by the exact and the approximate algorithm and determine their overlap, i.e., the number of nodes M present in both selections.

Table 1 summarizes the results for combinational cores of the ISCAS 89 circuits for derating target $D^{target} = 10$. The number of soft errors in column 2 is twice the number of gates in the circuit, as we assume two soft errors per gate output. Column 3 (D_{ex}) contains the derating of the circuit when no soft error has been selected for hardening ($SH = \emptyset$) according to Eq. (3). Column 4 (H_{ex}) quotes the number of soft errors which the exact method of [10] selected for hardening in set SH_{ex}.

The next four columns report the quality of the solution determined by the approximate method for safety margin $SM = 1.0$, i.e., the calculation of SH_{ap} was terminated when $D_{ap}(C_{SH_{ap}})$ exceeded D^{target}. Column H_{ap} contains the number of soft errors selected for hardening and included in set SH_{ap}, column D_{ap} quotes the approximate derating $D_{ap}(C_{SH_{ap}})$ achieved by the approximate solution and column D_{ex} contains the exact derating $D_{ex}(C_{SH_{ap}})$ achieved by the found solution. This number is marked bold if it exceeds the target D^{target}. In column M (matching) the number of soft errors present in both solutions (exact and approximate) is reported.

Columns 9 through 12 and 13 through 16 present the same data on the solutions obtained for a higher safety margin. The final row of the table contains the sums of the selected soft errors. Table 2 gives the same data for derating target of 20. The run times for the approximate method were all negligible (below one second) and are not reported.

It can be seen that the approximate method typically over-approximates the derating, as D_{ap} is typically larger than D_{ex}. As a consequence, the approximate method selects fewer soft errors for hardening than the exact method if no safety margin is used. The exact derating of the resulting solution often does not meet the target. However, a large fraction of the selected soft errors are also present in the exact solution as indicated by the numbers in column M. For higher values of SM, more soft errors are selected and the target is met in most cases. In the remaining cases, the actual exact derating is not far away from the target. It appears that for large circuits for which no exact solution is available, using SM of 2.0 is a reasonable option to achieve the derating target with a high confidence, as the target is met for all but one circuit in Tables 1 and 2.

In some situations the cost of hardening is limited by area or power consumption constraints. To reflect these situations, we consider the problem to select the soft errors for hardening such that their number does not exceed a cost limit L and the achieved derating is maximized. For this purpose, we sort the soft errors according to the detection probabilities and select the first L errors from the sorted list.

We applied both the exact and the approximate algorithm to the ISCAS 89 circuits with the cost limit L set to 10, 20 and 50 per cent of the total number of soft errors in the respective circuit. The results are summarized in Table 3. Columns H contain the number of soft errors available for hardening. Columns $D_{ex}(C_{SH_{ex}})$ give the exact derating achieved by the solution SH_{ex} found by the exact algorithm. Column $D_{ex}(C_{SH_{ap}})$ quotes the exact derating achieved by the solution SH_{ap} found by the approximate algorithm. We do not include the approximate derating D_{ap} into the table in order to ensure the comparability of the solutions obtained by the exact and the approximate method. Column M contains the number of soft errors which show up in both the exact and the approximate solution.

Table 1: Selective hardening with derating target $D^{target} = 10$

Circuit	Soft	D_{ex}	H_{ex}	Approx. method, $SM = 1.0$				Approx. method, $SM = 1.33$				Approx. method, $SM = 2.0$			
	errors	(no hard.)	(exact)	H_{ap}	D_{ap}	D_{ex}	M	H_{ap}	D_{ap}	D_{ex}	M	H_{ap}	D_{ap}	D_{ex}	M
s510	1020	6.1	109	60	10.0	7.9	49	146	13.3	9.3	63	266	20.1	**13.2**	88
s526	1052	7.0	78	46	10.0	8.7	42	102	13.3	**11.1**	77	185	20.1	**14.7**	78
s641	1280	3.8	315	289	10.0	8.7	263	385	13.4	**10.9**	279	519	20.1	**15.5**	310
s713	1426	4.0	323	306	10.0	8.3	238	424	13.3	**10.8**	268	566	20.0	**16.5**	314
s820	1640	18.7	0	0	20.8	**18.7**	0	0	20.8	**18.7**	0	0	20.8	**18.7**	0
s832	1664	19.0	0	0	21.1	**19.0**	0	0	21.1	**19.0**	0	0	21.1	**19.0**	0
s953	1906	4.9	211	195	10.0	9.3	180	262	13.3	**12.7**	209	361	20.1	**19.9**	211
s1196	2392	9.5	12	0	10.8	9.5	0	46	13.4	**11.5**	12	153	20.0	**16.5**	12
s1488	2976	12.9	0	0	14.6	**12.9**	0	0	14.6	**12.9**	0	62	20.0	**16.9**	0
s1494	2988	13.0	0	0	14.7	**13.0**	0	0	14.7	**13.0**	0	60	20.0	**16.9**	0
s1238	2476	10.2	0	0	11.5	**10.2**	0	31	13.4	**11.6**	0	131	20.0	**16.6**	0
s1423	2846	5.8	504	268	10.0	7.8	225	476	13.3	9.1	266	777	20.0	**11.4**	354
s5378	10590	4.5	2166	831	10.0	6.1	819	1361	13.3	7.2	1348	2017	20.0	8.9	1690
s9234	18468	5.4	2610	1997	10.0	8.4	1644	2920	13.3	**10.6**	2112	3909	20.0	**14.6**	2379
s13207	26358	3.6	8426	7274	10.0	7.8	6392	8638	13.3	9.8	7019	10428	20.0	**12.8**	7406
s15850	31694	4.1	8011	6635	10.0	8.0	6196	8405	13.3	**10.2**	7181	10744	20.0	**13.9**	7563
Sum	117290		23619	18567			16685	24240			19602	31789			21219

Table 2: Selective hardening with derating target $D^{target} = 20$

Circuit	Soft	D_{ex}	H_{ex}	Approx. method, $SM = 1.0$				Approx. method, $SM = 1.33$				Approx. method, $SM = 2.0$			
	errors	(no hard.)	(exact)	H_{ap}	D_{ap}	D_{ex}	M	H_{ap}	D_{ap}	D_{ex}	M	H_{ap}	D_{ap}	D_{ex}	M
s510	1020	6.1	262	266	20.1	13.2	162	347	26.7	18.0	206	452	40.1	**25.6**	231
s526	1052	7.0	244	185	20.1	14.7	149	246	26.7	16.5	162	332	40.0	**22.1**	198
s641	1280	3.8	549	519	20.1	15.5	441	596	26.7	18.4	458	690	40.2	**23.0**	479
s713	1426	4.0	577	566	20.0	16.5	468	655	26.7	19.2	483	771	40.1	**26.8**	527
s820	1640	18.7	6	0	20.8	18.7	0	18	26.7	**23.4**	6	58	40.2	**33.7**	6
s832	1664	19.0	5	0	21.1	19.0	0	17	26.7	**23.5**	5	56	40.2	**33.8**	5
s953	1906	4.9	354	361	20.1	19.9	329	426	26.8	**28.5**	346	530	40.1	**46.6**	353
s1196	2392	9.5	205	153	20.0	16.5	143	253	26.7	**20.9**	173	401	40.1	**31.0**	201
s1488	2976	12.9	113	62	20.0	16.9	62	145	26.7	**21.4**	102	305	40.1	**29.6**	110
s1494	2988	13.0	111	60	20.0	16.9	60	142	26.7	**21.4**	100	299	40.1	**29.5**	108
s1238	2476	10.2	182	131	20.0	16.6	127	227	26.7	**21.1**	154	373	40.1	**31.0**	179
s1423	2846	5.8	1080	777	20.0	11.4	581	1000	26.7	14.4	710	1275	40.0	**20.2**	871
s5378	10590	4.5	3983	2017	20.0	8.9	1929	2541	26.7	10.0	2151	3270	40.0	11.7	2354
s9234	18468	5.4	4502	3909	20.0	14.6	3710	4560	26.7	17.9	3878	5470	40.0	**22.2**	4134
s13207	26358	3.6	11225	10428	20.0	12.8	9477	11632	26.7	15.5	9987	13096	40.0	**21.3**	10501
s15850	31694	4.1	11854	10744	20.0	13.9	9886	12314	26.7	16.0	10256	14116	40.0	**20.8**	10881
Sum	117290		37134	31789			28801	37125			30600	44026			32797

Table 3: Selective hardening with cost limit

Circuit	Soft	D_{ex}	10 per cent soft errors hardened				20 per cent soft errors hardened				50 per cent soft errors hardened			
	errors		H	D_{ex} ($C_{SH_{ex}}$)	D_{ex} ($C_{SH_{ap}}$)	M	H	D_{ex} ($C_{SH_{ex}}$)	D_{ex} ($C_{SH_{ap}}$)	M	H	D_{ex} ($C_{SH_{ex}}$)	D_{ex} ($C_{SH_{ap}}$)	M
s510	1020	6.1	102	9.7	8.7	60	204	15.5	10.8	117	510	60.3	32.6	365
s526	1052	7.0	105	11.4	11.2	88	210	17.3	15.5	139	526	113.7	63.0	443
s641	1280	3.8	128	5.6	5.5	92	256	8.5	7.9	206	640	27.9	19.8	514
s713	1426	4.0	142	5.9	5.8	110	285	9.1	8.0	201	713	32.3	21.7	551
s820	1640	18.7	164	69.1	59.6	135	328	118.4	105.8	262	820	522.8	248.8	612
s832	1664	19.0	166	71.1	61.3	137	332	122.5	109.4	266	832	549.6	254.9	616
s953	1906	4.9	190	9.1	9.1	167	381	23.2	22.3	358	953	413.6	187.4	796
s1196	2392	9.5	239	22.3	20.2	191	478	42.5	36.6	390	1196	323.1	205.3	1058
s1488	2976	12.9	297	32.4	29.2	239	595	56.9	49.6	495	1488	240.4	169.6	1260
s1494	2988	13.0	298	32.6	29.4	240	597	57.6	50.4	493	1494	244.7	173.7	1272
s1238	2476	10.2	247	24.5	22.1	194	495	48.1	40.6	403	1238	409.1	252.5	1089
s1423	2846	5.8	284	8.3	7.9	175	569	10.6	9.8	317	1423	39.5	25.5	1131
s5378	10590	4.5	1059	6.6	6.5	789	2118	9.8	9.1	1678	5295	46.7	19.2	3706
s9234	18468	5.4	1846	8.3	8.1	1376	3693	14.2	13.5	3216	9234	743.3	168.1	8400
s13207	26358	3.6	2635	4.8	4.6	1488	5271	6.3	6.0	4022	13179	34.2	21.6	11496
s15850	31694	4.1	3169	5.7	5.6	2093	6338	7.9	7.7	5125	15847	50.3	30.1	14030

Table 4: Selective hardening of industrial circuits with derating target

Circuit	Soft errors	$D^{target} = 10$ H	%	Time	$D^{target} = 20$ H	%	Time
p35k	93456	6965	7.45	0.05	16605	17.77	0.05
p45k	87442	10713	12.25	0.05	20301	23.22	0.06
p77k	143266	7706	5.38	0.10	17394	12.14	0.09
p78k	154782	8218	5.31	0.10	28270	18.26	0.10
p81k	185674	22545	12.14	0.13	51191	27.57	0.13
p100k	193338	27683	14.32	0.12	53254	27.54	0.13
p267k	559650	116334	20.79	0.41	188020	33.6	0.41
p330k	696048	54346	7.81	0.53	132966	19.1	0.54
p378k	773894	41082	5.31	0.56	141335	18.26	0.57
p2927k	4887944	502711	10.28	4.20	1105949	22.63	4.31

The quality of the approximate algorithm comes quite close to the optimum in most cases. Notable exceptions are solutions with very high absolute values of derating (over 100). The majority of soft errors selected by the approximate method is also part of the optimal solution.

4.2 Scalability

To evaluate the scalability of the approximate method we applied it to the large industrial circuits provided by NXP. The results for node selection with derating target $D^{target} = 10$ and $D^{target} = 20$ are reported in Table 4. Column % contain the percentage of the selected soft errors (given in column H) among all soft errors. The results for node selection with a cost limit C set to 10, 20 and 50 per cent of all soft errors in the circuit are presented in Table 5. The exact method could not be run for these circuits because binary decision diagrams (BDDs) grow exponentially with the circuit size, resulting in memory requirements exceeding the capabilities of state-of-the-art computers. The total number of soft errors again equals twice the number of gates in the circuit. The run times are in CPU seconds and do not include the time to load the circuit (which took around 15 seconds for the largest circuit p2927k with approximately 2.5 million gates).

The results suggest that high derating values can be obtained for large circuits with a reasonable overhead and that the method is scalable.

5 Conclusions

We presented a simple and fast heuristic to select circuit nodes for selective hardening such that the derating is maximized, i.e., the circuit-level soft error rate is minimized. We demonstrated that the algorithm can achieve the derating levels obtained by the optimal algorithm if a safety margin was employed. In a scenario with a cost limit, the heuristic produced results similar to the exact algorithm. Application to large industrial circuits was demonstrated.

The algorithm makes a number of assumptions on the electrical behavior of a circuit node struck by a soft error. Integrating more accurate electrical data into the framework is a focus of the future work.

Table 5: Selective hardening of industrial circuits with cost limits C

Circuit	Soft errors	$L = 10\%$ D_{ap}	Time	$L = 20\%$ D_{ap}	Time	$L = 50\%$ D_{ap}	Time
p35k	93456	12.1	0.06	24.0	0.06	29577.1	0.05
p45k	87442	8.9	0.06	16.1	0.05	211.2	0.05
p77k	143266	15.7	0.10	38.3	0.09	1229.2	0.09
p78k	154782	12.6	0.10	21.6	0.10	115.7	0.10
p81k	185674	9.1	0.13	14.5	0.13	100	0.14
p100k	193338	8.2	0.13	13.4	0.14	119.1	0.14
p267k	559650	6.5	0.41	9.6	0.41	87.5	0.39
p330k	696048	11.2	0.55	21.1	0.55	257.1	0.54
p378k	773894	12.7	0.57	21.6	0.57	115.7	0.55
p2927k	4887944	9.9	4.21	17.3	4.28	181.2	4.32

6 References

[1] R.C. Baumann and E.B. Smith. Neutron-induced boron fission as a major source of soft errors in deep submicron SRAM devices. In *Int'l Reliability Physics Symp.*, pages 152–157, 2000.

[2] N. Seifert, X. Zhu, and L.W. Massengill. Impact of scaling on soft-error rates in commercial microprocessors. *IEEE Trans. on Nuclear Science*, 49(6):3100–3106, 2002.

[3] P.E. Dodd and L.W. Massengill. Basic mechanisms and modeling of single-event upset in digital microelectronics. *IEEE Trans. on Nuclear Science*, 50(3):583–602, 2003.

[4] H.T. Nguyen and Y. Yagil. A systematic approach to SER estimation and solutions. In *Int'l Reiliability Physics Symp.*, pages 60–70, 2003.

[5] P. Shivakumar, M. Kistler, W. Keckler, D. Burger, and L. Alvisi. Modeling the effect of technology trends on the soft error rate of combinational logic. In *Int'l Conf. on Dependable Systems and Networks*, pages 389–398, 2002.

[6] J. Benedetto, P. Eaton, K. Avery, D. Mavis, M. Gadlage, T. Turflinger, P.E. Dodd, and G. Vizkelethyd. Heavy ion-induced digital single-event transients in deep submicron processes. *IEEE Trans. on Nuclear Science*, 51(6):3480–3485, 2004.

[7] K. Mohanram and N.A. Touba. Cost-effective approach for reducing soft error failure rate in logic circuits. In *Int'l Test Conf.*, pages 893–901, 2003.

[8] C. Zhao, S. Dey, and X. Bai. Soft-spot analysis: Targeting compund noise effects in nanometer circuits. *IEEE Design & Test of Comp.*, 22(4):362–375, 2005.

[9] T. Heijmen and A. Nieuwland. Soft-error rate testing of deep-submicron integrated circuits. In *European Test Symp.*, pages 247–252, 2006.

[10] J.P. Hayes, I. Polian, and B. Becker. An analysis framework for transient-error tolerance. In *VLSI Test Symp.*, pages 249–255, 2007.

[11] R. Garg, N. Jayakumar, S.P. Khatri, and G. Choi. A design approach for radiation-hard digital electronics. In *Design Automation Conf.*, pages 773–778, 2006.

[12] M. Zhang, S. Mitra, T.M. Mak, N. Seifert, N.J. Wang, Q. Shi, K.S. Kim, N.R. Shanbhag, and S.J. Patel. Sequential element design with built-in soft error resilience. *IEEE Trans. on VLSI*, 14(12):1368–1378, 2006.

[13] S.A. Seshia, W. Li, and S. Mitra. Verification-guided soft error resilience. In *Design, Automation and Test in Europe Conf.*, 2007.

[14] S.K. Jain and V.D. Agrawal. Statistical fault analysis. *IEEE Design & Test*, 2(1):38–44, 1985.

[15] H.-J. Wunderlich. PROTEST: A tool for probabilistic testability analysis. In *Design Automation Conf.*, 1985.

[16] L.M. Huisman. The reliability of approximate testability measures. *IEEE Design & Test*, 5(6):57–67, 1988.