Output remapping technique for critical paths soft-error rate reduction

Q. Ding   Y. Wang   H. Wang   R. Luo   H. Yang

Department of Electrical Engineering, TNList, Tsinghua University, Beijing 100084, People's Republic of China
E-mail: yu-wang@mail.tsinghua.edu.cn

Abstract: As technology scales, soft errors in deep submicron circuits have become a major reliability concern due to smaller node capacitances and lower supply voltages. It is expected that the soft error rate (SER) of combinational logic will increase significantly. Previous solutions to mitigate soft errors in combinational logic suffer from delay penalty or area/power overhead. The authors proposed here an output remapping technique to reduce SER of critical paths. The SER reduction of our method ranges from 59.2 to 89.8%. This method does not introduce any delay penalty in most cases. The area/power overhead is limited as well. The output remapping method is based on the trade-off between SER and gate delay. The analysis shows that the width of the particle strike induced glitch scales down with technology scaling, which guarantees that output remapping technique works well along with technology scaling.

1 Introduction

Soft error is a transient failure of a circuit caused by alpha particles, fast neutrons and thermal neutrons. This kind of error changes the computation result of a circuit. It usually does not destroy the device. Along with the technology scaling, smaller node capacitances and lower supply voltages make the soft error a big concern [1−3].

When a particle hits a circuit node, it will deposit charge along the path and result in voltage glitch on the affected node. For memory elements, if deposited charge is more than a minimum value, the stored data will be damaged and a soft error occurs. This minimum value is called critical charge (or $Q_{critical}$), which is well accepted as the measure of the vulnerability of a circuit to soft errors by many researchers.

Combinational elements were considered much less vulnerable to soft errors due to three phenomena: logical masking, electrical masking and latching-window masking. However, because of technology scaling, soft error rate (SER) in combinational circuits is expected to increase significantly [2]. However, previous mitigation methods suffer from area and delay penalty.

In this work, based on the soft error generation and propagation model introduced in Sections 3 and 4, a trade-off between propagation delay and SER is presented and the importance of output nodes is addressed when considering SER reduction. We find out that a logic gate can control all the glitches that propagate through it. Furthermore, we propose the output remapping technique by controlling the output gate delay. Some output gates are selected and replaced with other complex logic gates that have longer propagation delay, and then narrow glitches will be filtered out at the outputs. The SER reduction of output remapping technique ranges from 59.2 to 89.8%, and delay/area penalty is minor for most circuits.

The paper is organised as follows. Section 2 introduces soft error problem, reviews previous mitigation techniques and then proposes the advantages of output remapping technique. Sections 3 and 4 introduce our soft error analysis model including the generation and propagation model. Section 5 presents the proposed output remapping technique to reduce SER based on our SER analysis platform, and the experimental results are also proposed and compared with related works. We summarise the paper in Section 6.

2 Soft error of combinational circuits

2.1 Technology trends

Technology scaling, which makes soft error a big concern, affects both memory and combinational circuits. Several studies have indicated that SER of an SRAM bit has been constant or even decreased. SER for some latches is constant or even decreasing slightly from 130 to 65 nm technologies [4]. Combinational circuits are considered to be less vulnerable due to three masking effects [2]: logical masking, electrical masking and latching-window masking. However, those masking effect is tempered for new technology generations. Owing to the decreasing pipeline stage, the effect of logical masking and electrical masking has been decreasing. The reduction in node capacitances and supply voltages decreases electrical masking. Increasing clock frequencies have reduced the latching-window masking. Because of these reasons, SER of combinational logic is expected to rise significantly [2].

2.2 Soft error mitigation

A common method to mitigate soft error is redundancy. Triple modular redundancy (TMR) consists of three copies of the original circuit and a majority voter. Apparently, TMR results in 200% overhead of area and power, whereas the voter introduces additional delay [5]. Time redundancy and partial duplication induce less overhead than TMR, but still adds additional delay [6, 7]. Such methods are applied when reliability is the most important design goal for the applications (such as space or banking application).

The radiation hardening techniques try to increase Q_{critical} of nodes by adding capacitance or resizing gates. For example, gate sizing was proposed to reduce SER [8]. This method altered the W/L ratio of transistors to improve the drive strength. Certainly, increased gate size results in area and power overhead, and many gates need to be resized to achieve prominent SER reduction. Additional load capacitance and optimal assignment of supply voltage, threshold voltage and gate size enhance the electrical masking effect and reduce SER [9]. It induces delay, area and power overhead as well. Latching-window masking through flip-flop selection [10] was also used to reduce SER, which introduced delay penalty as well.

In [11], two methods (gate cloning and cell resizing) were proposed to mitigate soft error. A robust compiler was designed to integrate those methods into existing design flow. A gate multiplication method was introduced in [12]. Researchers in [13] took advantage of selective hardening to mitigate soft error in combinational circuits. Although the results of these techniques [11–13] are relatively good, we compared them with our output remapping in Section 5 and showed that our output remapping technique can achieve even better results with less penalty. An alternative CMOS design style for soft error mitigation is proposed in [14], in which a static logic has two output ports. This method also needs more design effort than the output remapping method proposed in this paper.

2.3 Advantages of our technique

Previous methods cannot be used in critical paths. Some of those methods introduce significant area/power penalty. Comparing with those methods, the delay penalty of our method is negligible in most cases. Only one case suffers a delay penalty of 4%. Because only critical output nodes are modified in our method, area/power overhead can be ignored. As our model revealed later in Section 5.4, glitch width scales down with technology scaling, so we expect that this output remapping method scales well.

3 Soft error generation

For a single logic cell, there are two characteristics in terms of SER: glitch generation and glitch propagation. The generation characteristic of a logic cell determines the strength of voltage glitch caused by a particle strike. The propagation characteristic determines the electrical masking effect of the cell, that is, the glitch degradation effect. The particle strike-induced current pulse model is introduced in Section 3.1. Section 3.2 is regarding the property of the glitch width of generated pulses.

3.1 Current pulse

Particles that strike the silicon bulk will deposit a track of carriers. The carriers may recombine and form a very short current pulse at the circuit node. A double exponential current pulse (1) can be used to estimate this effect [15]

$$I(t) = I_{\text{peak}} \times (e^{-t/\tau_\alpha} - e^{-t/\tau_\beta}) \tag{1}$$

where I_{peak} is the amplitude of the pulse, τ_α is the collection time constant and τ_β is the ion-track establishment time constant.

3.2 Transient pulse generation

The current injection will cause voltage pulse at the output node of the gate. The strength of the voltage pulse is described by the pulse width (t_{w}) measured at $0.5V_{\text{dd}}$. If the amplitude of the current pulse (I_{peak}) induced by the particle strike at the storage node is more than a minimum value, t_{w} will be larger than 0. t_{w} will increase with I_{peak}. The strength of the injected current can be described by the charge of the current pulse, $Q_{\text{collected}}$.

The $Q_{\text{collected}} - t_{\text{w}}$ characteristic of each logic cell can be estimated by HSPICE simulation. The current source is injected to the circuit node that is hit by particles. At each run of simulation, I_{peak} is changed, and then $Q_{\text{collected}}$ and
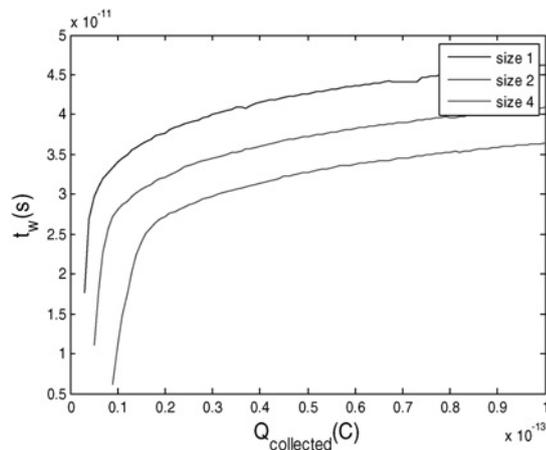
Figure 1 *Current injection will cause voltage pulse at the output node of the gate*

Current injections with different charge ($Q_{collected}$) will result in pulses with different strength (t_w). $Q_{collected}$–t_w characteristic of inverters with different sizes is illustrated in this figure

t_w are measured. This relation of an inverter with input state low is illustrated in Fig. 1. Notice that t_w increases significantly with $Q_{collected}$.

3.3 Glitch width

The simulation result of a real circuit is illustrated in Fig. 2. At each run of simulation, the current source is connected to the output of an inverter, and the I_{peak} of (1) is increased. When the peak voltage comes to be larger than a certain value (close to V_{dd} plus diode threshold voltage u), the drain diode is forward biased (Fig. 2) and the node voltage will not continue rising. After this point, the $Q_{collected}$–t_w curve shows an inflexion [16]. An analytical model
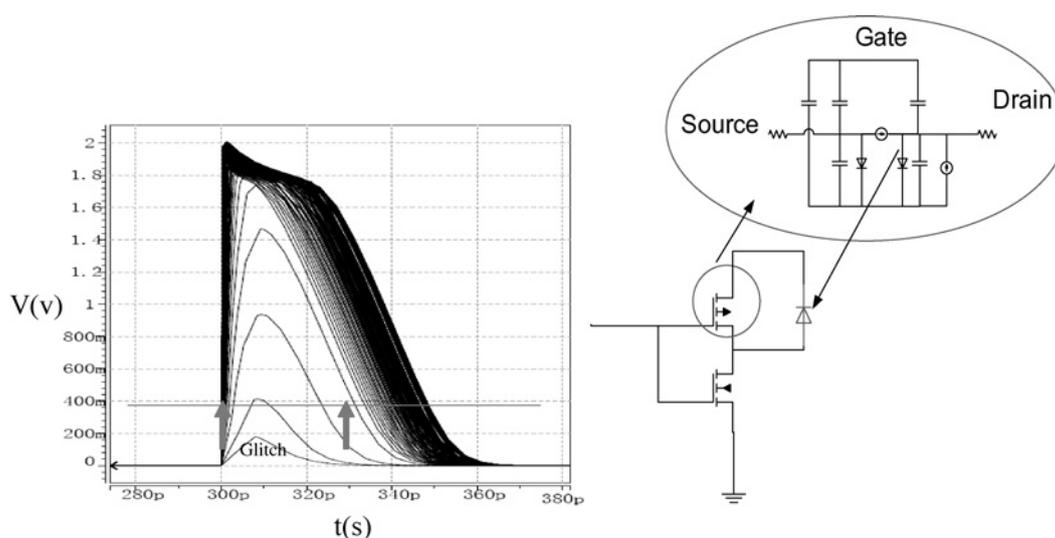
presented in [17] recently can be used to verify this conclusion.

4 Soft error propagation

In this section, the glitch propagation is first shown with an example, and follows the trade-off between propagation delay and SER. The importance of output nodes is demonstrated as well.

4.1 Transient pulse propagation example

The research of transient pulse propagation shows that glitch degradation is mainly determined by propagation delay (t_p) of a logic cell [18]. The example presented later is obtained by HSPICE simulation and Matlab curve fitting. In this example, the logic cell has propagation delay t_p and the duration of the input voltage pulse is τ_n. The glitch duration τ_{n+1} at the output of the cell can be depicted roughly by (2). When the input voltage pulse duration is smaller than t_p, the glitch cannot propagate to the next stage. The glitch degradation is faster when τ_{n+1} is closer to t_p as illustrated in Fig. 3. τ_{n+1} is always smaller than τ_n, but very wide glitch experiences little degradation.

$$\tau_{n+1} = 0.8469 \times t_p \times \left(\frac{\tau_n}{t_p} - e^{3.026 \times (1 - \tau_n/t_p)} \right) \quad (2)$$

4.2 Motivation example on logic chain

We have conducted experiments on inverter chains which are implemented with predictive technology model (PTM) 45 nm models [19]. The structure of our inverter chain is illustrated in Fig. 4. The last inverter is the gate with varying propagation delay. The inverter with the name '1' is
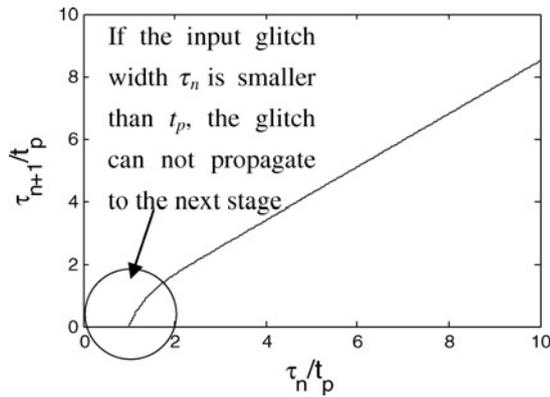

Figure 2 *At each run of simulation, the I_{peak} of (1) is increased*

When the peak voltage seems to be larger than a certain value (close to V_{dd} ($V_{dd} = 1$ V) plus diode threshold voltage u), the drain diode is forward biased and the node voltage rises much more with difficulty. The diode equivalent circuit is illustrated as well in the right-hand side, where the diode is reverse biased when the peak voltage is larger than ($V_{dd} + u$)

Figure 3 *Logic cell propagation characteristic example: when input voltage pulse duration is smaller than t_p, the glitch cannot propagate to the next stage*

The glitch degradation is faster when τ_{n+1} is closer to t_p
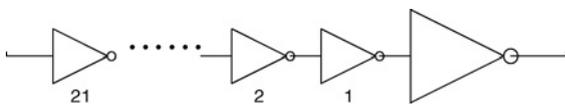

Figure 4 *Inverter chain used in our experiments: 21 levels of inverter is used in the experiment*

After the propagation delay is changed, the critical charge of every inverter is calculated. The result of the simulation is listed in Table 1

the first stage from the last gate. The inverter '2' is the second. After the propagation delay is changed, the critical charge of every inverter is calculated.

The results are listed in Table 1. Take the second row of Table 1 for example. Before we increase the delay, the output node will be flipped if the collected charge on the output node of inverter '2' is more than 2.23 fC. The collected charge has to be larger than 14 fC (72 fC) when the propagation delay of the output gate is increased to 32.5 ps (44.4 ps).

According to [2], Q_s is approximately 8 fC. We can obtain (3) as follows [1]

$$\frac{\text{SER}_{\text{before}}}{\text{SER}_{\text{delay1}}} = \frac{e^{-2.23/8}}{e^{-14/8}} = 4.4$$

$$\frac{\text{SER}_{\text{before}}}{\text{SER}_{\text{delay2}}} = \frac{e^{-2.23/8}}{e^{-72/8}} = 6.1 \times 10^3 \qquad (3)$$

where $\text{SER}_{\text{before}}$ is the SER before we increase the delay, $\text{SER}_{\text{delay1}}$ is the SER after the propagation gate delay is increased to 32.5 ps and $\text{SER}_{\text{delay2}}$ is the SER after the propagation gate delay is increased to 44.4 ps.

The critical charge appears to depend more upon the propagation delay than upon the position of the inverter, except inverter '1'. The result of inverter '1' is different,

Table 1 Effect of propagation delay over logic chain

Stage	Before, fC	32.5 ps, fC	44.4 ps, fC
1	2.01	4	8
2	2.23	14	72
3	2.39	14	72
4	2.47	15	75
5	2.56	15	71
7	2.69	15	71
9	2.81	15	71
11	2.93	15	72
13	3.06	16	72
15	3.21	16	72
17	3.36	16	72

because this node is strong coupled with logic chain output node. The output node of inverter '1' is a channel connected with the logic chain output [20]. The coupling effect comparatively weakens the electrical masking effect. However, the impact caused by this gate is considerably limited owing to the small number of the first stage gates. The delay variance of 11.9 ps ($44.4 - 32.5$ ps) results in about $4\times$ increase in critical charge, which implies that there is an exponential relationship between SER and propagation delay approximately, which is also our topic later.

4.3 Propagation delay and SER trade-off

We have conducted tests on the trade-off between propagation delay and SER. During each run of HSPICE simulation the propagation delay is changed and the Q_{critical} is obtained. The simulation result is illustrated in Fig. 5. It is shown in the figure as well that the simulation data can
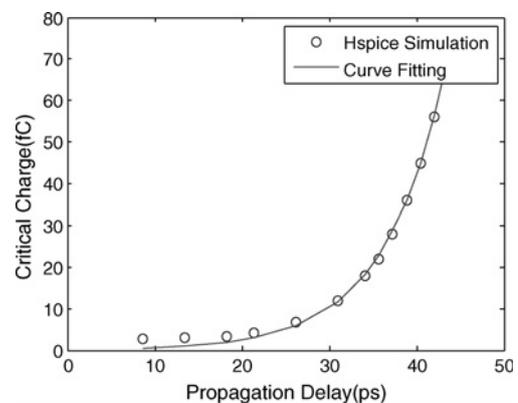

Figure 5 *Propagation delay impact on critical charge: in each run of HSPICE simulation the propagation delay is changed and the $Q_{critical}$ is calculated*

An exponential curve is used to fit the data

be estimated with an exponential equation. This relation is demonstrated in (4), where t_p is the propagation delay and $Q_{critical}$ is the critical charge; the larger the propagation delay, the better the (4) curve fitting accuracy

$$Q_{critical} = 0.148\,\text{fC} \times e^{t_p/7.1\,\text{ps}} \qquad (4)$$

The influence of propagation delay on critical charge is similar as that of critical charge on SER which is depicted in [1]. Following (4), the impact of propagation delay on SER is demonstrated approximately in (5)

$$\text{SER} \propto N_{flux} \times \text{CS} \times \exp\left(\frac{-0.148\,\text{fC} \times e^{t_p/7.1\,\text{ps}}}{Q_s}\right) \qquad (5)$$

4.4 Importance of output nodes

From the above analysis, an important conclusion can be made: a logic gate can control all the glitches that propagate through it. The reasons are listed in the following: First, as we mentioned before, glitches need to be propagated to the combinational output to cause a soft error. Second, according to Section 4.3, propagation delay has an exponential impact on critical charge. If we increase the propagation delay of a gate, all the critical charges of its fan-in cone will increase significantly. Third, SER has an exponential relation with critical charge [1].

We try to find out the most important gates when considering SER for every ISCAS85 benchmark circuit [21]. In the experiments, we calculate the shape of all particle-induced glitches that can propagate to the output and cause a soft error. The propagation path of each glitch is also recorded. Then the importance of each gate when considering SER is evaluated. The importance of a gate is defined as the accumulation of SER of each glitch that propagates through this gate. The experimental results show that the output nodes of ISCAS85 benchmark circuits are most important gates in the circuits when considering SER. The type of each benchmark circuit is unveiled in [22]. The results of [22] are listed in Table 2 as well. It shows that our experiments cover a vast variety of circuits.

5 Output remapping

In this section, our output remapping method is introduced [16] based on the importance analysis of all the gates in a circuit. We only focus on critical outputs. Critical output is defined as the output with delay close to the circuit delay. Circuit delay is defined as the maximum delay across all outputs. Non-critical outputs can be processed conveniently, because no delay penalty will be introduced.

5.1 Output remapping

We focus on the combinational outputs and replace some gates with other gates that have longer delay, so that SER

Table 2 ISCAS85 benchmark circuits

Benchmark	Circuit type
C432	27-channel interrupt controller
C499	32-bit SEC circuit
C880	8-bit arithmetic logic unit (ALU)
C1355	32-bit SEC circuit
C1908	16-bit SEC/DED circuit
C2670	12-bit ALU and controller
C3540	8-bit ALU
C5315	9-bit ALU
C6288	16 × 16 multiplier
C7552	32-bit adder/comparator

SEC/DED means 'single-error-correcting/double-error-detecting'

on the outputs will be reduced significantly. In this section, an example is first presented to explain the output remapping method.

Example: Fig. 6 shows an example from ISCAS85 benchmark circuits (C2670) [21]. Gate 1 is connected to an output which belongs to the critical path. Gates 1–4 are used to drive the output as a result of logic synthesis. Glitches are only filtered by the propagation delay of one NAND. If we replace gates 1–4 with a complex logic gate 5 that have longer delay, then glitches are filtered by the propagation delay of gate 5.

We use logical effort [23] to analyse, which is described as follows. The intrinsic delay of gate 5 is $(3 + 2 + 4)/3 = 3$. This comes from a negative channel metal-oxide semiconductor gate of size 3, one positive channel metal-oxide semiconductor (PMOS) gate of size 2 and one PMOS gate of size 4. The logical effort is $(4 + 3)/3 = 7/3$. So, the propagation delay of gate 5 is $t_{inv}(3 + 7/3F)$, where F is the path effective fan-out, which is defined as the load capacitance over the input capacitance.

The logic structure of gate 5 is illustrated in Fig. 6 as well. We can also calculate the delay for multi-stage version. The result is listed in Table 3 and illustrated in Fig. 7. As you can see, the single-stage version is faster than multi-stage version while F is smaller than 4. In Fig. 6, propagation delay from gate 3 input to gate 1 output is larger than the propagation delay of gate 5. We can adjust the load capacitance of gate 5 until its propagation delay approaches this value. Then all glitches narrower than the propagation delay of gate 5 cannot propagate to the latch input or the output node. According to Section 4.3, SER is reduced significantly. Notice that all the other gates belonging to
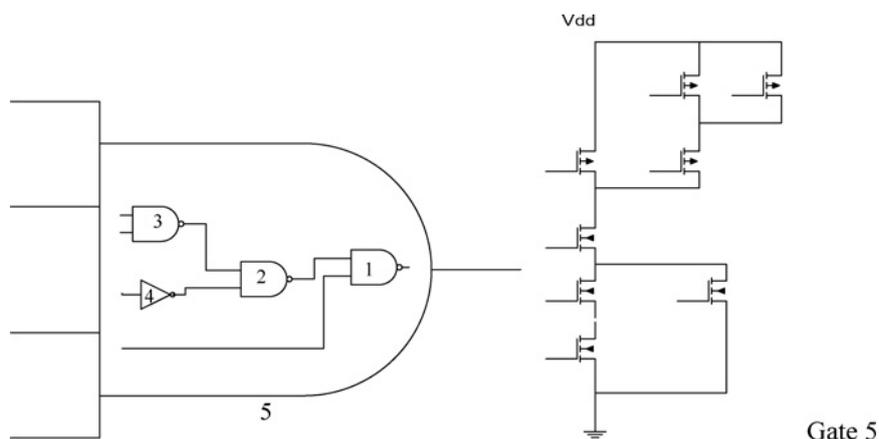
Figure 6 *C2670 example: Gate 1 is connected to an output which belongs to the critical path*

Gates 1–4 are used to drive the output as a result of logic synthesis. Gate 5 is used to replace gates 1–4. The structure of gate 5 is illustrated on the right

Table 3 Logical effort analysis

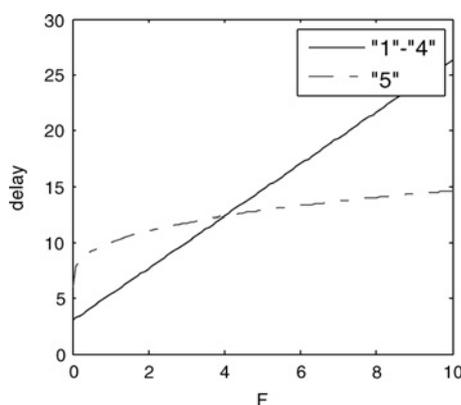| Single-stage	Multi-stage
$(3 + 7/3)F$	$6 + 4(F^{(1/3)})$

Figure 7 *Delay comparison of single-stage and multi-stage logic*

the fan-in cone of this output remain the same and no delay penalty is introduced.

The problem is that if we use single stage complex logic, it is sometimes too slow compared with the multi-stage version. The logic effort calculation of 8NAND and 8NOR is illustrated in Fig. 8 respectively. The delay of 8NAND and 8NOR static gates is much larger than the multi-stage logic when F increases. This introduces delay penalty. But this paper will show later that the delay penalty is minor for most circuits.

In those figures, the 'static' curve represents the single-stage propagation delay; the 'dynamic' curve represents the single-stage propagation delay when the gate is implemented with dynamic logic and the 'multi-stage' curve represents the original propagation delay.

5.2 Soft error analysis tool

We developed a C++-based SER analysis tool to calculate SER. The program flowchart is illustrated in Fig. 9. Researches of [24–28] can be referred to obtain a better understanding of the combinational circuit soft analysis tool, especially [28], which proposed a static, block-based and linear algorithm.

A standard cell library, the circuit netlist and the glitch generation/propagation model for each cell in the library are used as the input of this SER analysis tool. The glitch generation and propagation characteristics are obtained by HSPICE simulation. The inputs are parsed into the program and represented by a directed acyclic graph (DAG). Two hundred runs of simulation are performed after parsing. At the beginning of each run of the simulation, a logic input vector is randomly generated. Then a logic simulator is used to calculate the logic value of each node. After the logic simulation, the state of each gate is determined and so are the glitch generation and propagation characteristics of each gate. Then for each gate, if the particle strikes on it can cause a glitch at one of the output nodes, the relation between the injected charge and the glitch width of the output node is calculated. Based on those data, the circuit SER is calculated at last [2].

For example, there is a logic path 'a'–'b'–'c'–'d'–'o' in the circuit, and the path is not logically masked. 'a', 'b', 'c' and 'd' are inner gates of the circuit. 'o' is an output gate of the circuit. If a particle hits gate 'a', soft error generation characteristics of 'a' can be used to calculate the generated glitch shape at gate 'a'. The generated glitch will propagate from 'a' to 'b', then the propagation characteristics of gate 'b' is used to calculate the glitch shape at gate 'b'. At last, the corresponding glitch shape at gate 'o' is obtained. If a particle with different injected charge hits gate 'a', or if a particle hits another gate, the corresponding glitch shape can be obtained as well.
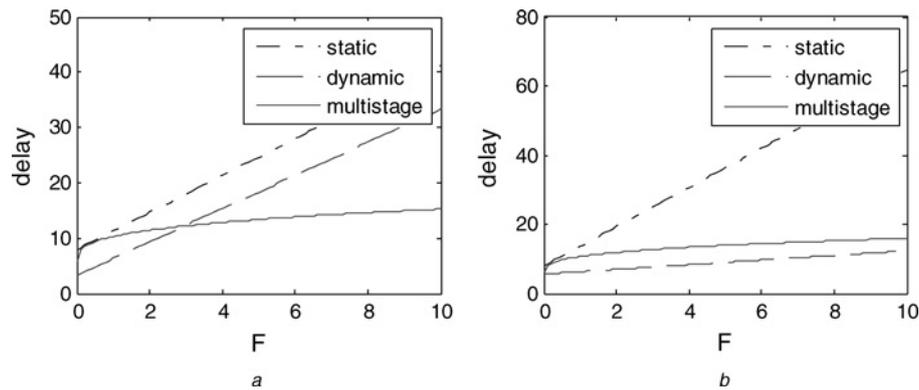
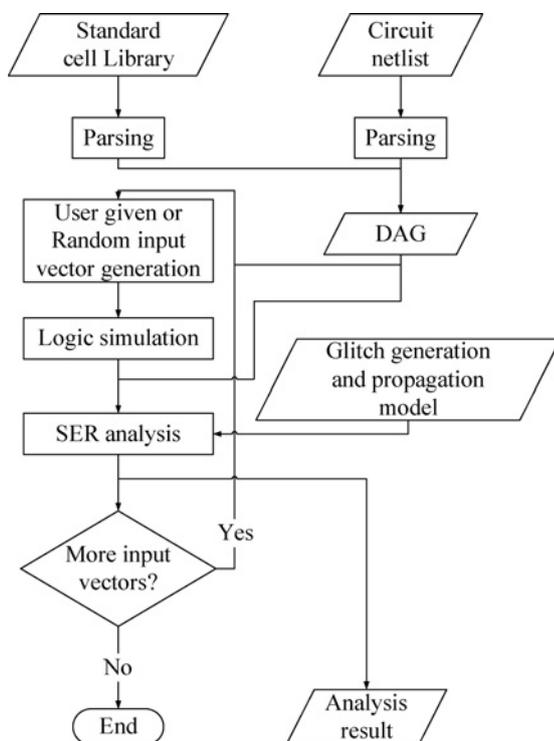**Figure 8** *Logical effort analysis*
a For 8NAND
b 8NOR

Figure 9 *Program flowchart of our SER analysis tool*

5.3 Experimental results

With the help of our SER analysis tool, we have conducted tests on benchmark circuits from ISCAS85. The logic cells used in our experiments is based on PTM 45 nm models [19] and Nangate Open Cell Library [29]. The results are demonstrated in Table 4. The result of [9] is listed in the table for comparison as well, because electrical masking effect is the focus of [9] and so is this paper. In [9], optimal assignment of supply voltage, threshold voltage, gate size and additional load capacitance enhance the electrical masking effect.

The SER reduction of our method ranges from 59.2 to 89.8%. The SER reduction of [9] ranges from 53.1 to 95.2%. The SER reduction results of two methods are close to each other. The delay penalty of our method is negligible in most cases. Only one case suffers a delay penalty of 4%. The C432 circuit has a delay penalty of 4%, because its size is the smallest and it tends to have more critical paths passing through the most critical output than other circuits. Although the delay penalty of [9] is small, ranging from 4.9 to 7.2%, the average delay penalty of our method is much smaller than the penalty of [9].

In ISCAS85 benchmark circuits, 1% of the gates belong to critical output nodes. Because only critical output nodes are modified in our method, power overhead can be ignored. The data of power penalty of our method is an approximation based on critical output number, circuit power consumption and output load capacitance. In [9] the power overhead ranges from 12.2 to 42.5%. Because the area of the remapped gates is small compared with the circuit, and the output node capacitance is large, SER caused by those gates can be ignored.

Comparing with our method, [11] can reduce error rate significantly (from 41 to 61%), but the reduction is smaller than ours; the area penalty of [12] is higher and its SER reduction is smaller; reference [13] has higher area penalty, while its SER reduction is similar to ours. As our model revealed later, glitch width scales down with technology scaling, so we expect that this output remapping method scales well.
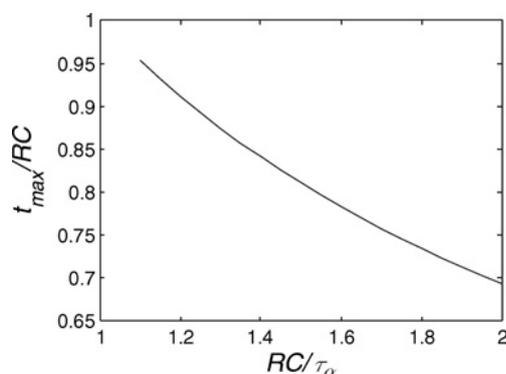
5.4 Glitch width scaling

As we mentioned in Section 3, the inflexion point reaches when the peak voltage approaches $(V_{dd} + u)$. The relation between t_{max}, the time required for the node voltage to rise from 0 to the peak value and RC can be estimated using (6) [30]

$$\frac{t_{max}}{RC} = \frac{1}{1 - (RC/\tau_\alpha)} \ln\left(\frac{\tau_\alpha}{RC}\right) \qquad (6)$$

It is also illustrated in Fig. 10, where you can see that t_{max}/RC changes a little when RC/τ_α rises from 1.1 to 2. According to

Table 4 SER reduction results of output remapping

Benchmark	Number of gates	Number of outputs	Number of remapped outputs	SER reduction, %	Delay increase, %	Energy increase, %	SER reduction of [9], %	Delay increase of [9], %	Energy increase of [9], %
C432	319	7	4	82.4	4	2.2	71.3	6.8	42.5
C499	796	32	32	81.3	0	7.1	53.1	6.2	12.2
C1908	646	25	7	72.5	0	1.1	95.2	6.3	41.2
C2670	968	140	6	89.8	0	0.5	83.5	6.3	45
C3540	1375	22	4	59.2	0	0.5	78.6	5.5	23.2
C5315	1941	123	19	79.6	0	2.0	88.6	4.9	23.5
C7552	2532	108	17	74.7	0	0.5	84.6	7.2	28.6
average	1225.3	65.3	12.7	77.1	0.57	2.0	79.3	6.2	30.9

Figure 10 t_{max}/RC changes a little when RC/τ_α rises from 1.1 to 2

This observation leads to the conclusion that t_{max} scales down with gate delay

Table 5 Glitch width scales down with gate delay

Technology, nm	t_w/t_p
180	4.2
90	6.2
65	5.3
45	4.9

[1], τ_α scales down with technology approximately linearly, which implies that RC/τ_α is approximately a constant. So, we could assume that t_{max}/RC does not change much which leads to the conclusion that t_{max} scales down with gate delay.

In addition, the time for the voltage to fall from t_{max} to ($V_{dd}/2$) is determined by RC or gate delay. Therefore the glitch duration corresponding to the inflexion point scales down with gate delay. The simulation results of relation between glitch width and gate delay at different technology

nodes are presented in Table 5. In the table, t_w/t_p is the glitch width over propagation delay. Both are simulation results of inverters. The current source is connected to the inverter output. The load is two inverters of the same size.

6 Conclusions

In this paper, based on a trade-off between propagation delay and SER, an output remapping technique is proposed to reduce SER. After the remapping, the multi-stage logic connected to the critical output is replaced with complex logic gate. This method takes advantage of electrical masking effect by increasing the last stage propagation delay to filter out the particle strike-induced glitches.

Our analysis also shows that glitch width scales down with technology scaling, so we expect that this method scales well. Our method only needs a little change at the combinational output and all the gates belonging to the fan-in cone of the output remains the same, so the power and area penalty are limited. Our output remapping technique can be used in critical paths. In most cases no delay penalty is introduced.

7 Acknowledgments

8 References

[1] HAZUCHA P., SVENSSON C.: 'Impact of CMOS technology scaling on the atmospheric neutron soft error rate', *IEEE Trans. Nucl. Sci.*, 2000, **47**, (6), pp. 2586–2594

[2] SHIVAKUMAR P., KISTLER M., KECKLER S., BURGER D., ALVISI L.: 'Modeling the effect of technology trends on the soft

332

IET Comput. Digit. Tech., 2010, Vol. 4, Iss. 4, pp. 325–333

error rate of combinational logic'. ICDSN, 2002, pp. 389–398

[3] SEIFERT N., ZHU X., MASSENGILL L.W.: 'Impact of scaling on soft-error rates in commercial microprocessors', *IEEE Trans. Nucl. Sci.*, 2002, **49**, pp. 3100–3106

[4] MITRA S., SEIFERT N., ZHANG M., SHI Q., KIM K.: 'Robust system design with built-in soft-error resilience', *IEEE Comput.*, 2005, **38**, (2), pp. 43–52

[5] SIEWIOREK D.P., SWARZ R.S.: 'Reliable computer systems: design and evaluation' (A.K. Peters, 1998, 3rd edn.)

[6] NICOLAIDIS M.: 'Time redundancy based soft-error tolerance to rescue nanometer technologies'. Proc. VTS, 1999, pp. 86–94

[7] MOHANRAM K., TOUBA N.A.: 'Cost-effective approach for reducing soft error failure rate in logic circuits'. Proc. Int. Test Conf. (ITC), September 2003, pp. 893–901

[8] ZHOU Q., MOHANRAM K.: 'Gate sizing to radiation harden combinational logic', *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst.*, 2006, **25**, (1), pp. 155–166

[9] DHILLON Y.S., DIRIL A.U., CHATTERJEE A., SINGH A.D.: 'Analysis and optimization of nanometer CMOS circuits for soft-error tolerance', *IEEE Trans. Very Large Scale Integ. (VLSI) Syst.*, 2006, **14**, pp. 514–524

[10] JOSHI V., RAO R.R., BLAAUW D., SYLVESTER D.: 'Logic SER reduction through flipflop redesign'. Proc. Int. Symp. Quality Electronic Design (ISQED), 2006, pp. 611–616

[11] ZHAO C., DEY S.: 'Improving transient error tolerance of digital VLSI circuits using robustness compiler'. ISQED, 2006, pp. 133–140

[12] NIEUWLAND A.K., JASAREVIC S., JERIN G.: 'Combinational logic soft error analysis and protection'. IOLTS, 2006

[13] SRINIVASAN V., STERNBERG A.L., DUNCAN A.R., ROBINSON W.H., BHUVA B.L., MASSENGILL L.W.: 'Single-event mitigation in combinational logic using targeted data path hardening', *IEEE Trans. Nucl. Sci.*, 2005, **52**, pp. 2516–2523

[14] ZHANG M., SHANBHAG N.: 'A CMOS design style for logic circuit hardening'. Proc. IEEE Int. Reliability Physics Symp., April 2005, pp. 223–229

[15] ZHOU Q., MOHANRAM K.: 'Cost-effective radiation hardening technique for logic circuits'. ICCAD, 2004, pp. 100–106

[16] DING Q., WANG Y., WANG H., LUO R., YANG H.: 'Output remapping technique for soft-error rate reduction in critical paths'. ISQED, 2008, pp. 74–77

[17] GARG R., NAGPAL C., KHATRI S.P.: 'A fast, analytical estimator for the SEU-induced pulse width in combinational designs'. DAC, 2008, pp. 918–923

[18] DHILLON Y.S., DIRIL A.U., CHATTERJEE A.: 'Soft-error tolerance analysis and optimization of nanometer circuits'. DATE, 2005, pp. 288–293

[19] CAO Y., SATO T., SYLVESTER D., ORSHANSKY M., HU C.: 'New paradigm of predictive MOSFET and interconnect modeling for early circuit design'. CICC, 2000, pp. 201–204

[20] WANG F., XIE Y., RAJARAMAN R., VAIDYANATHAN B.: 'Soft error rate analysis for combinational logic using an accurate electrical masking model'. VLSI Design, 2007, pp. 165–170

[21] BRGLEZ F., FUJIWARA H.: 'A neural netlist of ten combinational benchmark circuits and translator in Fortran'. Int. Symp. Circuits and Systems (ISCAS), June 1985, pp. 663–698

[22] HANSEN M.C., YALCIN H., HAYES J.P.: 'Unveiling the ISCAS-85 benchmarks: a case study in reverse engineering'. DATE, 1999, pp. 72–80

[23] HU B., WATANABE Y., MAREK-SADOWSKA M.: 'Gain-based technology mapping for discrete-size cell libraries'. DAC, 2003, pp. 574–579

[24] ZHANG M., SHANBHAG N.: 'A soft error rate analysis (SERA) methodology'. Int. Conf. Computer-Aided Design (ICCAD), November 2004, pp. 111–118

[25] RAJARAMAN R., KIM J., VIJAYKRISHNAN N., XIE Y., IRWIN M.: 'SEAT-LA: a soft error analysis tool for combinational logic'. Int. Conf. VLSI Design (VLSID), January 2006, pp. 499–502

[26] ZHANG B., ORSHANSKY M.: 'FASER: fast analysis of soft error susceptibility for cell based designs'. Int. Symp. Quality Electronic Design (ISQED), March 2006, pp. 755–760

[27] DHILLON Y., DIRIL A., CHATTERJEE A.: 'Soft-error tolerance analysis and optimization of nanometer circuits'. Design Automation and Test in Europe (DATE), 2005, pp. 288–293

[28] RAO R.R., CHOPRA K., BLAAUW D., SYLVESTER D.: 'An efficient static algorithm for computing the soft error rates of combinational circuits'. Design Automation and Test in Europe (DATE), 2006, pp. 164–169

[29] Available at http://www.opencelllibrary.org

[30] MOHANRAM K.: 'Closed-form simulation and robustness models for SEU-tolerant design'. VLSI Test Symp., 2005, pp. 327–333