A Unified Online Fault Detection Scheme via Checking of Stability Violation

Guihai Yan[†‡], Yinhe Han[†], Xiaowei Li[†*]
[†]Key Laboratory of Computer System and Architecture,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P.R. China, 100190
[‡]Graduate University of Chinese Academy of Sciences, Beijing, P.R. China, 100039
{yan_guihai, yinhes, lxw}@ict.ac.cn

Abstract

In ultra-deep submicro technology, two of the paramount reliability concerns are soft errors and device aging. Although intensive studies have been done to face the two challenges, most take them separately so far, thereby failing to reach better performance-cost tradeoffs. To support a more efficient design tradeoff, we present a new fault model, Stability Violation, derived from analysis of signal behavior. Furthermore, we propose a unified fault detection scheme — Stability Violation based Fault Detection (SVFD), by which the soft errors (both Single Event Upset and Single Event Transient), aging delay, and delay faults can be uniformly handled. SVFD can greatly facilitate soft error-resistant and aging-aware designs. SVFD is validated by conducting a set of intensive Hspice simulations targeting 65nm CMOS technology. Experimental results show that SVFD has more robust capability for fault detection than previous schemes at comparable overhead in terms of area, power, and performance.

1. Introduction

The development of semiconductor technology in the following decade will bring a broad set of reliability challenges at a dramatic fast pace [1]. Two of the paramount challenges are soft errors and aging-driven lifetime reliability.

Many researchers focused on soft error modeling and mitigation within a wide design spectrum: device level, circuit level [2–4], microarchitecture level [5], and software level [6]. In addition, the industry and academic communities have done much work on understanding the semiconductor device reliability failure mechanisms and models, such as Electromigration [7], NBTI [8] [9] [10], TDDB, Hot Carrier Injection, Temperature cycling [11] etc.

To mitigate aging effects, a promising approach is by aging-failure prediction[12][13]. Unlike soft errors, device aging is a gradual process. Before the devices totally breakdown and thereby loss their functionalities, they always tend to exhibit performance degradation, e.g. increased threshold voltage instability, soaring leakage power, worse heat characteristics etc. Most of these negative effects can result in the degradation of switch performance of the transistors[14], and eventually excessive path delay. In some words, most of the aging failures can be predicted by sensing the gradually increased aging delay. Agarwal et al. designed an aging sensor for this purpose.

On the other hand, to alleviate the threats of soft errors, most approaches are redundancy-based, such as spatial redundancy by duplicating the flip-flops [2][15], or temporal redundancy by multiple-sampling [16]. However, those "redundancy" resources help little in mitigating aging effects, and even speed up the aging process due to the extra heat generated by those redundancy resources. This dilemma makes the goal of providing a not only aging-aware but also soft error-resistant scheme hard to achieve, unless a cumbersome combination of the previous aging-sensor and redundancy-based approaches is conducted.

*To whom correspondence should be addressed.

This work was supported in part by the High-Tech Research and Development Program of China under grant No.(2007AA01Z109, 2008AA01Z147) and National Natural Science Foundation of China under grant No.(60806014, 60633060, 60831160526, 60776031), and by the National Basic Research Program of China under grant No.(2005CB321604, 2005CB321605).

Rather than exploiting such a cumbersome combination, in this paper, we provide a unified mechanism to handle the two challenges. Based on the signal behavior analysis, we find that the soft errors and aging delay can converge into same signal behavior: *Stability Violation*. Even the conventional delay faults, which could result from transition hazard, crosstalk etc., can be brought into this behavior. Hence a unified fault model and detection mechanism can be exploited, thereby creating the chance of reaching a more optimum tradeoff between detection capability, design complexity, and implementation overhead. To our knowledge, this is the first work to handle the soft errors, aging delay, and delay faults under a unified fault detection mechanism.

The main contributions of this paper are:

1) We propose a new fault model, called Stability Violation. We conclude that, at signal behavior level, the soft errors, aging delay, and delay faults can be uniformly modeled as Stability Violation.

2) Based on the new fault model, we propose an efficient online fault detection scheme—Stability Violation based Fault Detection (SVFD). SVFD can not only facilitate soft error-resistant design, but also aging-failure prediction. Besides that, SVFD can handle the conventional delay faults.

The rest of this paper is organized as follows: Section 2 shows the related work and evaluates their limitations. Section 3 presents the SVFD mechanism. Section 4 addresses two major implementation concerns, which can greatly facilitate the circuit design presented in Section 5. Section 6 shows the experimental results. Section 7 discusses how to distinguish detection results, followed by the conclusion in Section 8.

2. Related Work

Mitra et al. proposed a self-checking flip-flop design (SCFF)[2]. In their scheme, the scan portion of the scannable flip-flop is reused as redundancy of the flip-flop working in functional mode to detect SEU. Furthermore, through some clock manipulation—skewing the clock of the redundancy flip-flop [15], the modified flip-flop, referred as "LOW-COST", can enable the SET detection capability. Although the "reuse" philosophy can reduce area overhead, it limit the applicability since not all circuit designs employ the redundancy-style flip-flop design, especially for some performance-critical pipelines where the sequential units are more timing-saving latches, rather than flip-flops. Moreover, this design can not be used for aging prediction.

Agarwal et al. proposed a sensor design dedicated for aging failure prediction, called Aging Resistant Stability Checker (ARSC) [12]. The fundamental principle of aging prediction is delay detection since the aging process tends to induce performance degradation. Figure 1 shows the only difference between the aging delay detection and traditional delay fault detection: the former takes place in a safe timing interval called "Guard Band" [12], while the latter takes place in the interval after the effective clock edge called "Detection Slack". The targeting "aging delay", strictly speaking, is not a delay fault since it never translates into error. In contrast, a general delay fault can induce a error. ARSC inherently possesses little SET detection capability: the maximum detectable glitch width is T_{GB}. In addition, it can not detect large SET, delay fault, and SEU.

Recently, Nagpal et al. presented a Code Word State Preserving based flip-flop design (CWSPFF) [17] dedicated for SET protection.

978-3-9810801-5-5/DATE09 © 2009 EDAA

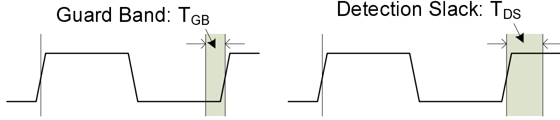

Figure 1. Guard Band and Detection Slack

The maximum detectable glitch width is $\min\{t_{cd}/2, (t_{pd}-\Delta)/2\}$ (t_{cd} and t_{pd} are the contamination delay and propagation delay of a combinational logic, respectively). This limitation is more stringent than our approach in which the maximum detectable glitch width can easily excess $t_{pd}/2$. Moreover, the CWSPFF can not be used to protect SEU and predict aging failures, and the area overhead is unattractive.

3. SVFD: Stability Violation based Fault Detection

First, we specify the target fault types. Then, we present the unified Stability Violation model and the SVFD mechanism.

3.1 Target Fault Types

For an on-line fault detection mechanism, three classes of faults are particularly significant:

Soft Error: *Single Event Upset* (SEU) and *Single Event Transient* (SET) [18]. If some high energy radioactive particles induce a storage cell to be flipped, this unintentional bit-flip is called SEU. If the particles cause a node of combinational logics to collect enough charge, a transient current pulse could be generated. This pulse can transform into a voltage pulse and propagate along logic paths [19]. This type of soft error is called SET. A soft error might not be captured by flip-flops due to three masking effects [19]: Logic Masking, Electrical Masking, and Latching-window Masking.

Aging Delay: the aging effects, such as NBTI, can cause aging delay which can be used for aging-failure prediction[12]. Usually, the aging delay increasing is a gradual process over time, but the abrupt delay increasing is possible when the devices suffer from breakdowns induced by mechanical stresses. This type of "abrupt" aging delay will not be covered in this paper.

Delay Fault: This type of faults refers to the conventional delay faults [20] which is caused by device defect, signal crosstalk, etc. We just cover the delay fault whose size is less than the width of the Detection Slack.

3.2 Modeling Faulty Signals

Mathematically, a signal S can be expressed as a function of time t, that is $S = f(t)$. Given the time interval of (t_i, t_t), in which S can get into a stable state before t_t. The interval (t_i, t_t) is divided into *variable period*, denoted as $T_{vp}^{S} = (t_i,\ t_s)$, and *stable period*, denoted as $T_{sp}^{S} = (t_s,\ t_t)$, where t_s is the complete time of the last transition of S. The initial value and the terminal value of the signal are denoted as $F_i^S = f(t_i)$, and $F_t^S = f(t_t)$, respectively.

According to the above definition, we define a faulty signal, S_f, that commits at least one of the three violations:

1) Initial Value Violation (IVV): The obtained value of $F_i^{S_f}$ at time t_i differs with $f(t_i)$.

2) Terminal Value Violation (TVV): The obtained value of $F_t^{S_f}$ at time t_t differs with $f(t_t)$.

3) Stability Violation (SV): One or multiple transitions happen in the stable period.

There are some correlations among the three types of Violations. For example, the TVV of a signal could cause the IVV of another signal; A SV of a signal can cause the TVV of the signal itself.

The above violation behaviors, strictly speaking, can not precisely capture all details of signal mismatch between a fault-free signal and its faulty counterpart. Practically, however, the above violation rules are robust enough to guide high efficient on-line fault detection. In fact, given the target fault types (Section 2.1), only the Stability Violation of a signal is needed to be verified. Rather than mathematically describing the application of the violation based fault model, the following will explain how to use this model in a practical way.

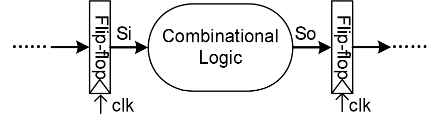

Figure 2. Generic Logic Circuit

First, the T_{vp}, and T_{sp} for specified signals need to be established respectively. Figure 2 models a general logic circuit. The input signal S_i comes from the upstream flip-flop, and the output S_o is captured by the downstream flip-flop. The both flip-flops are synchronized by the same clock clk with cycle period of T. Several timing parameters are denoted as:

t_{pd}: the propagation delay of the combinational logic;

t_{cd}: the contamination delay (a.k.a. short-path delay) of the combinational logic;

t_{cq}: the flip-flop's clock-to-q time.

The S_i is updated at every effective clock transition, and is held on for the whole time period, which means there is almost no *variable period* exist. Thus the stable value, the variable period, and the stable period of S_i in the nth clock cycle ($(n-1)T, nT$) can be expressed as:

$$T_{vp}^{S_i} = ((n-1)T,\ (n-1)T + t_{cq}) \quad (1)$$

$$T_{sp}^{S_i} = ((n-1)T + t_{cq},\ nT) \quad (2)$$

Unlike S_i, S_o has a wider variable period. The S_o's stable value, variable period, and stable period are expressed as:

$$T_{vp}^{S_o} = ((n-1)T + t_{cq} + t_{cd},\ (n-1)T + t_{cq} + t_{pd}) \quad (3)$$

$$T_{sp}^{S_o} = ((n-1)T + t_{cq} + t_{pd},\ nT + t_{cq} + t_{cd}) \quad (4)$$

Figure 3 illustrates the above time period in the nth cycle, where $t_1 = (n-1)T + t_{cq} + t_{cd}$, $t_2 = (n-1)T + t_{cq} + t_{pd}$.

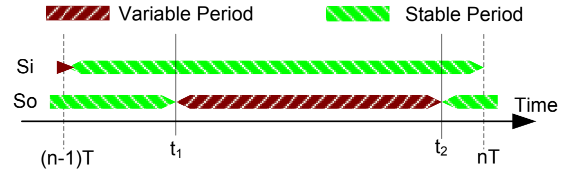

Figure 3. Variable Period VS. Stable Period

Then, we can explain how a target fault (delay fault, SEU, and SET) results in some of the above violations, and how these IVV, and TVV converge to SV.

1) Suppose that a delay fault occurs, the delayed S_o will cause SV in Detection Slack (T_{DS}) during which the S_o should keep stable. Equivalently, the delay fault will result in S_o's TVV since at the end of the cycle, S_o can not reach the expected value. This TVV then causes the IVV of the signal in the next stage of logic. So, SV, TVV, and IVV are equivalent to each other for the delay fault.

2) Suppose that an aging delay occurs, the delayed S_o will cause SV in Guard Band (T_{GB}). Unlike the delay fault, the aging delay will not cause TVV and IVV. So, an aging delay just represents as SV.

3) Suppose that an unmasked SEU takes place in the upstream flip-flop. Clearly, the S_i's SV is committed because, after a short clock-to-q time, S_i is supposed to keep stable during the whole cycle period. this S_i's SV could potentially cause the downstream flip-flop to capture faulty data, and thereby results in S_o's TVV, then IVV of input signals in the next stage logic. So, the SEU will represents as SV, and possible IVV and TVV.

4) Suppose that an unmasked SET happens in the combinational logic. If the duration of the SET is less than $T_{DS} + T_{GB}$, the SET fault behavior is similar with the delay fault: the unexpected signal transitions within the S_o's stable period. Therefore, the analysis result for the delay faults is held for SET faults. That is SV, TVV, and IVV are equivalently to each other for the SET.

From the above analysis we can conclude that, at the signal behavior-level, the target faults either induce equivalent SV, IVV, and TVV (for delay fault and SET), or only represent as SV (for aging delay), or SV and possible equivalent IVV and TVV (for SEU). In other words, the target faults can be uniformly modeled as SV. The implication is that we can employ a unified stability checker to handle the detection for all the target faults. This unification can support more efficient implementation of an online fault detection scheme than traditional redundancy-based approaches [2][3]. More attractively, the capability for aging failure prediction [12][13] can be readily exploited in place with the same scheme.

The following section presets the implementation of SVFD mechanism. In particular, two major practical design considerations are covered, which can greatly facilitate the circuit design (in Section 5).

4. Implementation of SVFD

A valid execution of a logic (shown in Figure 2) implies that the flip-flops capture "right data at right time". Generally, the right time is guaranteed by a set of optimized system clocks. In this study, we mainly concern about the integrity of the data at right time.

4.1 Manipulate Precharge Period

A general checker design is based on the dynamic circuit style. The first concern is how to schedule the precharge period because the traditional cycle-begin precharge or cycle-end precharge styles are unapplicable in our detection mechanism. As Section 3.2 explained, during the *Guard Band* and *Detection Slack*, the checker should be on duty, rather than stay in precharge state. For S_o the precharge can be scheduled for its variable period. However, the same schedule strategy is unallowable for S_i because there is almost no any variable period can be employed for precharge. If we "brutally" borrow some time from S_i's stable period for precharging the checker, the fault coverage is hard to be guaranteed.

To address this problem, we find if the precharge stage is scheduled according to some specific timing requirements, the fault coverage will not be sacrificed. The timing manipulation is based on the key observation, called as **Propagation of Stability Violation**.

Suppose that an unmasked SEU occurs in the upstream flip-flops at time t in the nth cycle, then the effects of the Stability Violation of S_i will be propagated to S_o within the time period of $(t + t_{cd},\ t + t_{pd})$. If the effects of S_i's stability violation can propagate into S_o's stable period, that is

$$(t + t_{cd},\ t + t_{pd}) \subseteq (nT - T_{GB}, nT + T_{DS}); \quad (5)$$

The SEU induced S_i's SV will be represented as S_o's SV since the S_o should keep stable during the *Guard Band* and *Detection Slack*. Thus, the checker for detecting S_o's SV now can naturally handle a part of S_i's SV within the particular time interval, referred as **Propagation Detectable Period (PDP)**. From (5), we can obtain the PDP as

$$nT - T_{GB} - t_{cd} < t < nT + T_{DS} - t_{pd}. \quad (6)$$

Clearly, not all unmasked SEUs taking place in the upstream flip-flop can translate into the S_o's SV; e.g. if a S_i's SV happened during the $((n-1)T,\ t_1)$ (Figure 3), then it could not be detected by S_o's checker because Eq.(5) can not be held any more.

To cover this period, one way is setting another stability checker for S_i, at the expense of almost doubled area, power and complexity. In contrast, we employ a simple but far more efficient way to cover this period, referred as *XOR Protection*, as Figure 4 shows. The effectiveness of this scheme based on the key observation: the $S_o^{(K-1)}$ is consistent with the S_i^K within the period of $((n-1)T,\ t_1)$; therefore one XOR gate is capable of capturing any S_i^K stability violation during this time. Clearly, the overhead imposed by a XOR gate is far more efficient than that imposed by another stability checker or other traditional redundancy flip-flop based schemes [2]. How to efficiently handle the output of XOR will be present in Section 5.

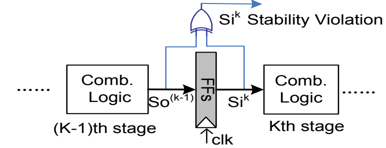

Figure 4. XOR Protection

Then, we can derive the start point of the available precharge time is $nT - t_{cd} - T_{GB}$. Additionally, to avoid Detection Slack violation, the actual start point of the precharge stage should be

$$\max\{(n-1)T + T_{DS},\ nT - t_{cd} - T_{GB}\} \quad (7)$$

If we schedule precharge stage within this period of

$$(\max\{(n-1)T + T_{DS},\ nT - t_{cd} - T_{GB}\},\ nT + T_{DS} - t_{pd}),$$

the fault coverage will not be sacrificed. The available precharge duration τ can be expressed as

$$\tau = \begin{cases} T_{DS} + T_{GB} + t_{cd} - t_{pd} & \text{if } t_{cd} < T - T_{GB} - T_{DS}, \\ T - t_{pd} & \text{otherwise.} \end{cases}$$

To sustain normal operations, there is a minimum precharge duration τ_0, which is determined by the intrinsic RC constant. Clearly, $\tau > \tau_0$ needs to be meet.

4.2 Eliminate SEU detection "Blind Zone"

Considering the propagation delay of a SEU, we can claim that the SEU must be benign if

$$t > nT - t_{cd}. \quad (8)$$

For S_i protection, besides the XOR protection period, the Propagation Detectable period, and the benign period, there might be the fourth region that has not be covered so far. Figure 5 shows that the whole Stable Period of S_i could be divided into four or three zones depending on different timing parameters.

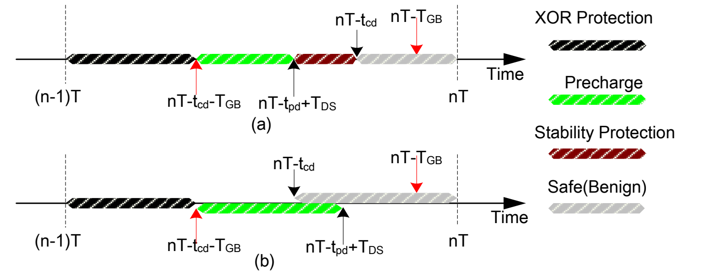

Figure 5. Variety of Timing Period

Figure 5(a) shows if $nT - t_{pd} + T_{DS} < nT - t_{cd}$, then a SEU taking place in the zone of $(nT - t_{pd} + T_{DS},\ nT - t_{cd})$ may not propagate into a detectable period, thereby resulting in detection "Blind Zone". Unlike the XOR protection period, this trouble can not be eliminated unless another dedicated stability checker is set for S_i, at considerable expense of implementation overhead. However, we propose a new approaches: Contamination Delay Optimization, by which the "Blind Zone" can be eliminated by some timing manipulation.

Contamination Delay Optimization: Clearly, the "Blind Zone" can be naturally eliminated if

$$nT - t_{cd} < nT - t_{pd} + T_{DS} \quad (9)$$

is satisfied, as Figure 5(b) shows. The SEU taking place in $(nT - t_{cd} - T_{GB},\ nT - t_{pd} + T_{DS})$ is either propagated into a Stability Violation detectable zone of S_o, or have nothing bad effect due to being benign period. From (9), we get

$$t_{cd} > t_{pd} - T_{DS} \quad (10)$$

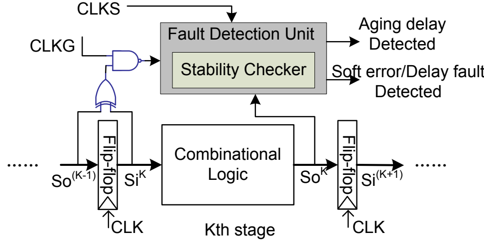

Figure 6. Top View of SVFD Scheme

In addition, the XOR protection zone should meet

$$nT - t_{cd} - T_{GB} < (n-1)T + t_{cd} + t_{cq}$$
$$\Rightarrow t_{cd} > \frac{1}{2}(T - T_{GB} - t_{cq}) \qquad (11)$$

From (10), and (11), t_{cd} should meet the requirement:

$$t_{cd} > \max\{t_{pd} - T_{DS},\ \frac{1}{2}(T - T_{GB} - t_{cq})\} \qquad (12)$$

Generally, (12) requires the contamination delay of the combinational logic reaches up to about a half cycle period. This same requirement is needed to be satisfied in some previous studies [15] to address "short path effects" [21]. Actually, It is consistent with the goal of variety timing optimization strategies [22][23], and therefore not a substantial limitation.

A byproduct coming with the Contamination Delay Optimization is that the terminal time of precharge period can be extended from

$$(nT - t_{pd} + T_{DS}) \text{ to } (nT - T_{GB})$$

since the checker does not need to be on duty from time $(nT - t_{pd} + T_{DS})$ to $(nT - T_{GB})$ any more. Combined with (7), we can derive the available precharge period after eliminating the "Blind Zone" as

$$(\max\{(n-1)T + T_{DS},\ nT - t_{cd} - T_{GB}\}, (nT - T_{GB})). \qquad (13)$$

Actually, (13) shows that the available τ is easy to excess τ_0. Experiment results indicate that for 65nm CMOS, 1GHz, τ_0 is merely 40ps, while τ is at least at the magnitude of hundreds of picoseconds.

To sum up, we can use only one stability checker, with the assistant XOR protection, for soft errors, aging delay, and delay faults detection. Figure 6 shows the top view of the whole fault detection scheme. Note that the XOR output needs to be gated beyond the XOR protection period. Figure 7 shows the overall timing relations. The CLKS is used to precharge-evaluation control, and the CLKG is the gating clock for XOR output.

5. Circuit Design

Figure 8 shows the transistor level design of SVFD scheme which consists of two key components: a stability checker(Figure 8(b)) and a output compressor (Figure 8(c)).

The basis stability checker can be derived from a sensing circuit for on-line delay fault detection [20], in which the signal integrity is verified by a pair of consistent charge/discharge nodes, a delay fault will trigger one of the nodes to be discharged/chargeed and thereby causes states inconsistent between them. The same fundamental detection principle is employed to design a sensor dedicated for aging prediction, referred as Aging Resistant Stability Checker (ARSC) [12]. Based on the same principle, we design a new stability checker. Compared with ARSC, the checker has several new features which can improve the robustness and reduce the overhead. The following explains 1) how does the circuit work, and then 2) presents the new features.

During precharge period, both nodes $S1$ and $S2$ in stability checker are charged up to HIGH. Then, the circuit starts evaluation, one of the two nodes is pulled down, while the other one floats at HIGH because

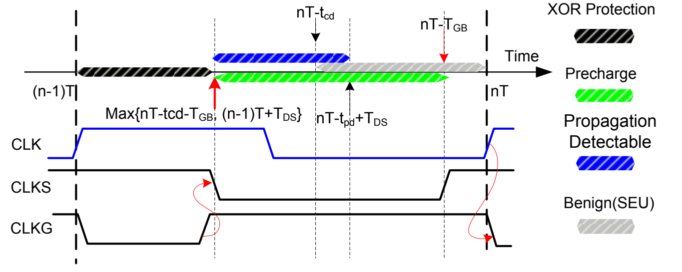

Figure 7. Precharge Clock and XOR Gating Clock Timing

the gate signal of M3 is always complemented with that of M4. A keeper can help the floated node stick to HIGH. The node $S1$ and $S2$ are always exclusive during fault-free time, which will make the node $S4$ stick to HIGH because the high-impedance path between S4 and GND always exist. When a Stability Violation is committed by S_i (out of the XOR protection period) or S_o, the violation will trigger the discharge of the node that has charged up to HIGH. Eventually both nodes are discharged, and thereby the node S4 is pull down to LOW. Then, the node X in output compressor will be discharged, which flags a fault being detected. The compressed result X needs to be latched twice: CLK-latched for indicating aging delay and CLKG-latched for indicating soft error or delay fault (Figure 8(d)). The reason will be explained in Section 7.

There are two new features in the detection unit.

1) The NOR logic for combining the states of S1 and S2 is realized with a dynamic logic (M6, M7, and M8), which can improve the robustness of the checker and reduce the area overhead and switch power dissipation. Unlike the stability checker in ARSC [12], where the checker output, a static NOR gate, is directly driven by a floated HIGH node during fault-free time, our checker's output is generated by a dynamic NAND gate. This change is based on the key observations: during precharge, both the node S1 and S2 are pulled up to HIGH, consequently, both M7 and M8 are turned off; thereby no short path exist when precharge. So the foot transistor for the dynamic NAND is eliminated. Note that due to the precharge RC delay of S1 and S2, the M6's precharge clock should be delayed by a precharge delay constant, though this delay is not necessary during the dominated fault-free time.

2) The outputs are compressed for reducing the number of output latches. Since such fine-grained detection results, which are for every signal, are not necessary for the most recovery or aging-aware designs. This relaxation creates the chance of compressing the detection results for reducing area overhead. We use a wide dynamic NOR to realize the compressor, in which the M11 and M12 serve as a level restorer for node X.

In addition, this detection unit can be easily disabled by pulling up the CLKS to HIGH. The PMOS transistor's aging process induced by NBTI effects is greatly slow down in disable mode. Unlink ARSC, where the Guard Band is confined by a clock and its delayed counterpart [12], SVFD unit provide Guard Band by the CLK and CLKS. So the clock skew needs to be controlled well and thus incur potential complexity overhead. This issue is beyond the scope of this paper.

6. Experiments

The experiments consist of two parts. The first is dedicated for detailed timing verification; the second presents the overhead in terms of area, power dissipation, and performance. The results are obtained by using the Hspice targeting the 65nm Predictive Technology Model [24]. Table 1 shows the related experiment parameters.

Table 1. Experiment Parameters (65nm CMOS)

V_{DD}	V_{th}	T	T_{GB}	T_{DS}	t_{cq}	t_{pd}	t_{cd}
1.25V	0.42V	1 ns	0.1ns	0.5ns	10ps	0.85ns	0.45ns

Using (13), we derive the precharge clock's duty cycle is: (0.5ns, 0.9ns). The duty cycle of the signal gating clock, CLKG, is: (0.5ns, 1ns) (Figure 7).

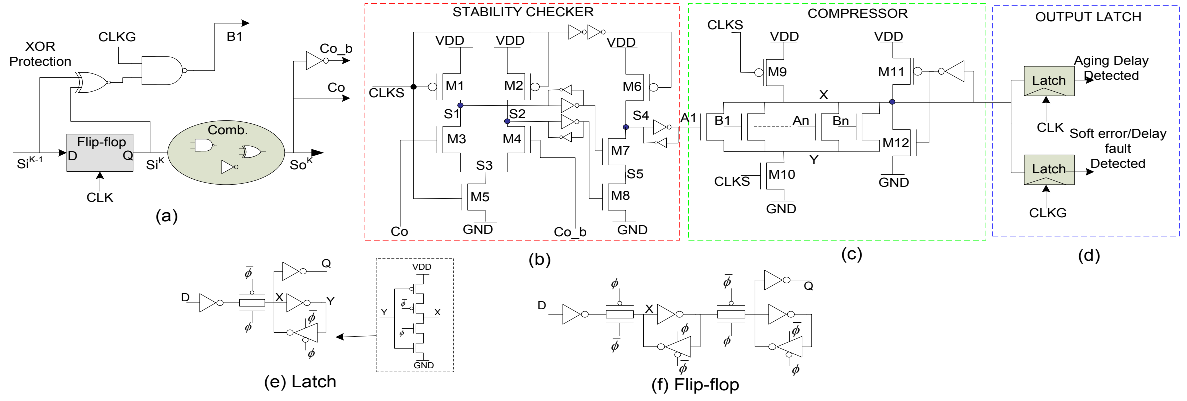

Figure 8. SVFD Implementation

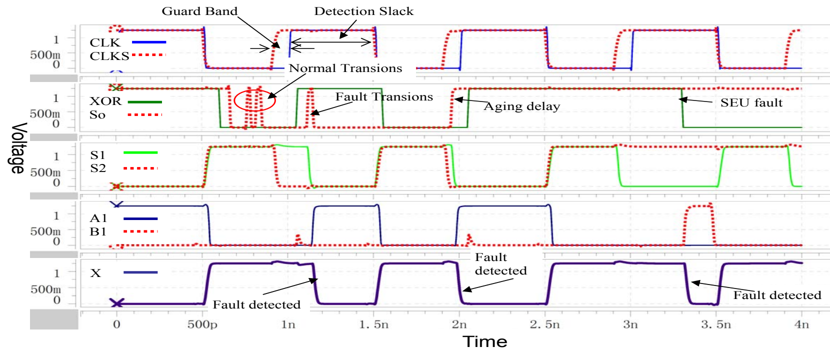

Figure 9. Hspice Simulated Signal State Transitions

Figure 9 shows the detail timing of a SVFD unit in 4 cycles. The topmost shows the system clock CLK, the precharge-evaluation clock CLKS. The second shows the monitored signals: XOR protection output and S_o. The third illustrates the state transitions of the two most important node S1 and S2. The forth shows the signals A1, the output of the stability checker, and B1, the gated output of XOR protection unit. Both are feeded to the compressor. The bottommost shows the detection result generated by the compressor.

During the first cycle, S_o presents some normal transitions. In the fist half of the second cycle, a unexpected glitch, which can simulate a SET fault, takes place; then in the guard band of the second cycle, a aging delay is simulated. The third cycle is fault-free. In the fourth cycle, a SEU fault is simulated by XOR signal.

From the bottom figure, we can see that all the Stability Violation shown in the second waveform are successfully detected.

We zoom in the figure to extract some useful timing information (the zoomed figures are omitted due to space limitation): 1)the critical precharge time τ_0 is about 40ps, while the available precharge time is 400ps, far larger than that. So the precharge time will not be a limitation when we manipulate the related timings (Section 4.1). 2)The detection delay is just about 45ps is just about 3 FO4 delay in 65nm technology. 3)The maximum undetectable glitch width is about 18ps, which is even less than most soft error induce glitch width in 65nm technology, so the robustness of SET detection should not be in question.

Table 2 shows the tradeoffs comparison among the SEFF [2], LOWCOST [15], ARSC [12], CSWPFF [17], and SVFD. we use the number of transistors as the area overhead metric, as many circuit-level studies take.

Area: To conduct comparisons between variety schemes, a baseline latch and flip-flop design needs to be determined. Figure 8(e) and (f) show the adopted baseline design. The similar latch design is used by Intel as a standard datapath latch [25]. The flip-flops is used in PowerPC603 processor [26]. In addition, a XOR gate consumes at least 12 transistors when computing the number of transistors. For fairness, only the checker and its input generating logics are considered; the subcomponents that can be shared among checkers (i.e. output compressor, and output latches) are not taken account though such amortization will make the area overhead of SVFD more attractive.

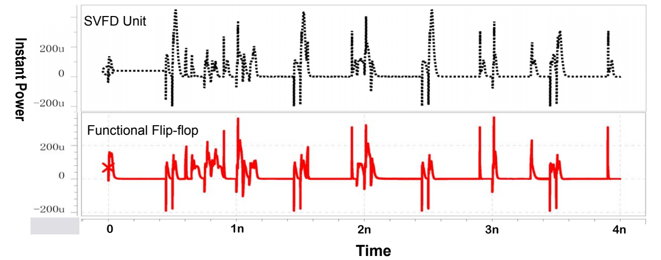

Figure 10. Instant Power Comparison Between a Standard Flip-Flop and SVFD Unit

Power: Another concern is about power overhead. We compare our SVFD unit's instant power against that of a standard flip-flop(as shown in Figure 8(e)) with the same input signal (Figure 9(b)) and frequency. Figure 10 shows that our SVFD's power consumption P_{SVFD} is comparable with a standard flip-flop's power consumption P_{FF}. We use Hspice to integrate the waveform over time and then get the average power relation: $P_{SVFDavg} \approx 0.96\, P_{FFavg}$.

SEFF's power is doubled, as [2] shows, is because the redundancy flip-flop is enabled. Similar modification is conducted in LOWCOST; moreover, a extra lath is employed. So the power of LOWCOST must be larger than that of SEFF.

Note that our checker seems more power-hungry than ARSC. That is because the power overhead metric in [12] is different with ours. In ARSC, the power overhead is calculated as the whole logic (include both the flip-flop and combinational logic) power increase. Because the combinational logic's power is relatively constant, so the actual sensor power consumption compared with flip-flops should be much higher.

Table 2. Comparing Tradeoffs with other schemes

	SEFF[2]	LOWCOST[15]	ARSC[12]	CWSPFF[17]	SVFD
Transistor	14	36	24	46	36
Power	2	>2	>1.1◁	N/A	1.96
performance	0	N/A	<1%	<1%	<1%
Clock	1	2	2	2	3
Applicability	Limited◇	Limited	General	General	General
Capability					
SEU detec.	Yes	Yes	No	No	Yes
SET detec.	No	Yes	No	No	Yes
Aging pred.	No	No	Yes	No	Yes
Delay fault detec.	No	Yes	No	Yes	Yes

N/A: Not applicable.
◇ The scheme needs support from a specific scannable flip-flop.
◁ ARSC uses a different metric of power overhead.

Performance: The performance mainly depends on the flip-flops time overhead and the critical path delay. In SVFD, there is not modification to the flip-flops and the critical path is not changed as well. The only timing penalty results from several extra gate capacitances drived by the S_i and S_o. Our experiment result shows this penalty is less than 1% for a special combinational logic: 8-inverter chain. In fact, the same situation is faced by SEFF, LOWCOST, ARSC and CWSPFF.

Clock: At last, we compare the number of clock used by these schemes. For example, SEFF needs one extra clock, while SVFD needs two extra clocks: CLKS and CLKG. This is a negative attribute of SVFD since the extra clocks could potentially increase the complexity. However, the SVFD's detection capability is the most powerful.

Applicability: The SEFF and LOWCOST need the support from a particular type of scannable flip-flop, but the other three schemes do not suffer from this limit.

7. Discussion

Distinguish Detection Results: It is useful to distinguish the aging delay caused detection positive from the rest of detection results, because the detected aging delay rate is used as the input for some aging-aware designs.

SVFD implicitly apply a rule for distinguish the detected results. That is: If a stability violation is detected in Guard Band, then this violation is viewed as aging delay induced; the stability violation detected in other region is viewed as soft error or delay fault induced. Figure 8(d) is used to implement this rule. However, this might degrade the confidence level of detected aging delay rate since if a stability violation takes place within the Guard Band, SVFD can not determine whether this violation is caused by a soft error or an aging delay.

Fortunately, this confidence degradation incurred by this implementation is negligible. To quantitatively evaluate the miss rate, we define the miss as: a soft error induced stability violation is misjudged as a aging-fault stability violation.

Suppose that the raw soft error rate (SER), $R_{softerror}$, is uniformly distributed over time. The detectable SER is $\alpha R_{softerror}$, where the α is a constant ($0 < \alpha < 1$) related to the three masking effects [19]. The aging fault rate is denoted as R_{aging}.

The misjudgment rate R_{miss} can be expressed as

$$R_{miss} = 1 - \frac{R_{aging}}{R_{aging} + \alpha R_{softerror} \times \frac{T_{GB}}{T_{DS}+T_{GB}}}$$

Practically, the Guard Band should not be larger than the timing margin to avoid extra timing penalty. A typical timing margin is 10%. Assume that $\alpha = 0.5$, and $R_{softerror} = 0.1 \times R_{aging}$ (actually, after some detectable aging effects of devices begin emerging, the assumptions of α and raw SER are heavily conservative), $T_{GB}/TDS = 0.2$ then R_{miss} is not large than 1%. Therefore, we can safely conclude that the unperfect distinguishing capability will not impose a substantial problem.

8. Conclusions and Future Work

In this study, we propose a unified online fault detection mechanism—SVFD, by which the soft error, aging delay, and delay fault are uniformly and efficiently handled. In particular, SVFD has the capability of aging-failure prediction. We present an efficient implementation of SVFD at circuit-level. Experimental results show SVFD can achieve an attractive tradeoff between capability and overhead.

A further implication of employing SVFD at chip-level has not been investigated, which is planed to be conducted in the future work.

9. Acknowledgments

We thank Dr. Lei Zhang for his helpful comments. We also thank Ms. Hui Liu and Mr. Jun Liu for their useful suggestions.

References

[1] International SEMATECH, Inc., "Critical Reliability Challenges for The International Technology Roadmap for Semiconductors (ITRS)," 2003.

[2] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K.S. Kim, "Robust System Design with Built-In Soft-Error Resilience," *IEEE Computer*, vol. 38, pp. 43–52, Feb. 2005.

[3] J. Han, J. Gao, Y. Qi, P. Jonker, and J.A.B. Fortes, "Toward Hardware-Redundant, Fault-Tolerant Logic for Nanoelectronics," *IEEE Design & Test of Computers*, vol. 22, pp. 328–339, July-Aug. 2005.

[4] 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 Transactions on VLSI*, vol. 14, pp. 1368–1378, Dec. 2006.

[5] T.N. Vijaykumar, I. Pomeranz, and K. Cheng, "Transient-Fault Recovery Using Simultaneous Multithreading," *ISCA*, pp. 87–98, May 2002.

[6] N. Oh, and E.J. McCluskey, "Error Detection by Selective Procedure Call Duplication for Low Energy Consumption," *IEEE Transactions on Reliability*, vol. 51, pp. 392–402, Dec. 2002.

[7] J.R. Black, "Electromigration—A brief survey and some recent results," *IEEE Transactions on Electron Devices*, vol. 16, no. 4, pp. 338–347, 1969.

[8] B.C. Paul, K. Kunhyuk, H. Kufluoglu, M.A. Alam, and K. Roy, "Impact of NBTI on the temporal performance degradation of digital circuits," *IEEE Electron Device Letters*, vol. 26, no. 8, pp. 560–562, 2005.

[9] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of PMOS NBTI effect for robust nanometer design," *DAC*, pp. 1047–1052, 2006.

[10] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao, "The impact of NBTI on the performance of combinational and sequential circuits," *DAC*, pp. 364–369, 2007.

[11] International SEMATECH, Inc., "Semiconductor Device Reliability Failure Models," *http://ismi.sematech.org/docubase/abstracts/3955axfr.htm*, 2000.

[12] M. Agarwal, B.C. Paul, M. Zhang, and S. Mitra, "Circuit Failure Prediction and Its Application to Transistor Aging," *VTS*, pp. 277–286, 2007.

[13] M. Agarwal, et al., "Optimized Circuit Failure Prediction for Aging: Practicality and Promise," *ITC*, 2008.

[14] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, no. 6, pp. 10–16, 2005.

[15] M. Mehrara, M. Attariyan, S. Shyam, K. Constantinides, V. Bertacco, and T. Austin, "Low-cost protection for SER upsets and silicon defects," *DATE*, pp. 1146–1157, 2007.

[16] M. Nicolaidis, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies," *VTS*, pp. 86–94, 1999.

[17] C. Nagpal, R. Garg, and S.P. Khatri, "A Delay-efficient Radiation-hard Digital Design Approach Using CWSP Elements," *DATE*, pp. 354–359, 2008.

[18] M. Nicolaidis, "Design for Soft Error Mitigation," *IEEE Transactions on Device and Materials Reliability*, vol. 5, pp. 405–418, Sep. 2005.

[19] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic," *DSN*, pp. 389–398, 2002.

[20] M. Favalli, and C. Metra , "Sensing circuit for on-line detection of delay faults," *IEEE Transactions on VLSI*, vol. 4, no. 1, pp. 130–133, 1996.

[21] M. Nicolaidis, "GRAAL: A New Fault Tolerant Design Paradigm for Mitigating the Flaws of Deep Nanometric Technologies," *ITC*, pp. 1–10, Oct. 2007.

[22] B. Taskin, and I.S. Kourtev,, "Delay insertion method in clock skew scheduling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 4, pp. 651–663, 2006.

[23] N.V. Shenoy, and R.K. Brayton, et al., "Minimum padding to satisfy short path constraints," *ICCAD*, pp. 156–161, 1993.

[24] W. Zhao, and Y. Cao, "New generation of Predictive Technology Model for sub-45nm design exploration," *ISQED*, pp. 585–590, 2006.

[25] T. Karnik, B. Bloechel, K. Soumyanath, V. De, and S. Borkar, "Scaling trends of cosmic ray induced soft errors in static latches beyond 0.18 μ," *Symposium on VLSI Circuits, Digest of Technical Papers.*, pp. 61–62, 2001.

[26] G. Gerosa et al., "A 2.2 W, 80 MHz superscalar RISC microprocessor," *JSSC*, vol. 29, pp. 1440–1452, Dec 1994.