Constraint-Aware Robustness Insertion for Optimal Noise-Tolerance Enhancement in VLSI Circuits

Chong Zhao
Department of ECE
Univ. of California, San Diego
La Jolla, CA 92093
(858) 534-7883
chong@ece.ucsd.edu

Yi Zhao
Department of ECE
Univ. of California, San Diego
La Jolla, CA 92093
(858) 435-7883
yizhao@ece.ucsd.edu

Sujit Dey
Department of ECE
Univ. of California, San Diego
La Jolla, CA 92093
(858) 534-0750
dey@ece.ucsd.edu

ABSTRACT

Reliability of nanometer circuits is becoming a major concern in today's VLSI chip design due to interferences from multiple noise sources as well as radiation-induced soft errors. Traditional noise analysis/avoidance and manufacturing testing are no longer sufficient to handle the dynamic interactions between various noise sources and unpredictable operational variations. Therefore, "robustness insertion" has been adopted as the supplementary approach to ensure high circuit reliability through on-line protections. However, the related design overhead is not always acceptable, especially for cost/timing-sensitive designs. In this paper, we present a novel "**constraint-aware robustness insertion**" methodology protect the sequential elements in digital circuits against various noise effects. Based on a configurable hardening sequential cell design and an efficient sequential cell robustness estimation technique, an optimization algorithm is developed to search for the optimal protection scheme under given timing and area constraints. Experiment results demonstrate that the proposed methodology is able to achieve a high degree of noise-tolerance while keeping the protection cost within limit.

Categories and Subject Descriptors

B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-tolerance.

General Terms

Algorithms, Design, Reliability.

Keywords

Nanometer circuits, Robustness calibration, Circuit hardening, Robustness insertion.

1. INTRODUCTION

With feature size shrinking to nano-meter scale, clock frequency reaching multi-GHz and supply voltage approaching sub-voltage range, various noise effects in VLSI circuits are becoming much stronger than ever and the noise margin of semiconductor devices is significantly reduced at the same time. As a result, according to the International Technology Roadmap for Semiconductor (ITRS) [1], reliability of nano-meter circuits is being greatly threatened.

The circuit reliability might be degraded by factors in every phase during chip development. Signal integrity issues such as crosstalk and power-grid noise that were considered second-order effects in the previous technology generations have become primary design concerns; their effects might be intensified by manufacturing defects and process variations, resulting in significant yield loss and performance degradation; environmental factors such as temperature fluctuations and particle strikes impose severe menace to on-line reliability. Although tremendous research have been done in analyzing and mitigating each of these noise effects [2][3][4][5][6], none considers that multiple noise sources and operational variations can dynamically interact with each other to aggravate the error effects [10]. This so called "compound noise effects" makes stand-alone pre-manufacturing noise analysis/avoidance and deterministic manufacturing testing overly optimistic and insufficient, mandating on-line protection techniques to provide extra assurance [7].

Circuit-hardening [8] [9], or "robustness insertion", has been recognized as a promising solution to improve on-line reliability. By inserting spatial and/or temporal redundancies, a circuit-hardening technique detects and corrects errors occurred during chip operation. All such techniques are associated with certain design overhead and the protection might not be efficient or economical without guidelines. On one hand, excessive design penalty will be paid if blindly applying these techniques to the entire circuit. On the other hand, the vulnerable circuit elements may not receive proper protection while other circuit elements may be over-protected if the hardening technique can only be applied to limited locations due to design constraints. It is desirable to insert the redundancies only to judiciously selected locations that are most likely to be affected to achieve optimal protection without unacceptable design overhead.

In CMOS digital circuits, sequential cells such as D-type flip-flops (DFFs) play a crucial role in circuit reliability because noise originated in the combinational circuit can affect the circuit functionality only if it is captured by a DFF. Therefore, hardening DFFs is essential in designing highly robust digital circuit. As we will see, the probability for noise to attack different DFFs and the capability of a DFF to resist these attacks greatly vary, and the tight design constraints may prevent sufficient hardening from being applied to all DFFs. Hence, the quest for an optimal protection scheme requires careful consideration of several components: (1) Hardening Cell: a low-cost noise-tolerant DFF design must be used; (2) Robustness Calibration: an efficient

This work was supported partially by the MARCO/DARPA Gigascale Systems Research Center (GSRC)

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

technique for evaluating the noise-immunity of the DFFs and the entire circuit must be adopted as a gauging metric; and (3) Robustness Optimization: a fast algorithm must be developed to search for the optimal protection schemes under constraints.

In this paper, we present a novel "*constraint-aware robustness insertion*" methodology to maximally increase the reliability of a circuit facing various noise sources under given design constraints. It is capable of simultaneously considering the three factors mentioned above. For the hardening cell, we use a cost-effective "Separate Dual Transistor" DFF design (SDT-DFF) [9] with configurable error-tolerant capability. For robustness calibration, we evaluate the individual DFF robustness based on the result of the noise impact analysis technique introduced in [11]; the overall circuit robustness is calculated as a weighted sum over all DFFs. For robustness optimization, we formulate it as a multi-constrained optimization problem and develop a dynamic-programming-based algorithm. A highly integrated framework is constructed to implement the proposed methodology. As we will show, our methodology can greatly reduce the number of errors caused by various noise sources with very low design overhead.

The major contribution of this work is two-fold. First, to the best of our knowledge, this is the first effort of selective robustness insertion under explicit guidelines of both robustness evaluation and design constraints. Second, the optimization algorithm is the crucial link that ties the robustness analysis and noise-tolerant cell design together in designing reliable digital circuit. Currently, we are focused on glitch-type noise in CMOS digital circuit, where noise is modeled as a glitch with certain duration and amplitude. With reasonable efforts, delay-type noise can be handled as well.

The rest of the paper is organized as follows. Section 2 introduces the hardening sequential cell design; Section 3 discusses the DFF robustness calibration technique; Section 4 describes the robustness optimization algorithm and the integrated implementation framework; Section 5 includes experimental results and discussions; and Section 6 concludes the paper.

2. ROBUST SEQUENTIAL CELL DESIGN

Various noise-tolerant sequential cells have been developed. Among them, [9] presented a novel Separate-Dual-Transistor (SDT) latch design (**Figure 1(a)**), based on which a noise-tolerant DFF (**Figure 1(b)**) has been designed. Compared with traditional latch design, each transistor in the feedback loop is duplicated to form a pair of transistors of the same type and the input signal D is differentiated by a preset delay *dt* before it is fed into the feedback loop. The transistor pairs (M1 and M2, M3 and M4, M5 and M6, M7 and M8) concurrently compare the values of the differentiated signals D1 and D2. If input D is stable, D1 and D2 carry the same logic value, the duplicated transistors are in the same states and the SDT-DFF functions as a normal DFF. If a glitch with duration w ($w \leq dt$) is present at D, D1 and D2 will carry different logic values (**Figure 1(c)**) so the two transistors in a pair will be in opposite states, keeping the output unchanged. Hence, any glitch whose width w is smaller than *dt* will not be able to erroneously change the state of the DFF.

We chose SDT-DFF as the noise-tolerant cell in our work because of its two major advantages: (1) Low spatial overhead: the area overhead is about 60%, much lower as compared with other hardening cell designs such as the Triple Modular Redundancy designs [8]. (2) Configurable temporal overhead: since the error-tolerance of an SDT-DFF increases with the inserted delay *dt*, we can choose to use SDT-DFFs with different *dt* values for different level of protection at the different cost of area and delay.

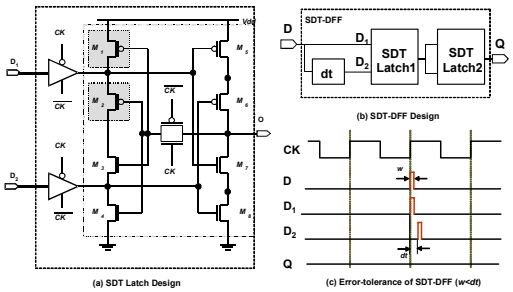


Figure 1 Separate-Dual-Transistor Design

3. ROBUSTNESSS CALIBRATION

The robustness of a DFF depends on three factors. The first is the intrinsic noise-tolerance of the DFF itself. Depending on the loading conditions, a specific type of DFF can only capture incoming noise whose width and height are larger than certain value. This can be depicted by a "Noise Capture Zone" (*NCZ*), as shown in **Figure 2**. The higher noise-tolerance of an SDT-DFF can be reflected in the shrinkage of its *NCZ* (**Figure 2(b)**) because the SDT-DFF is designed to be immune to noise with a width smaller than *dt*. The longer the insertion delay, the smaller the *NCZ* and the higher the noise-tolerance.

The second factor is the inherent characteristics of the circuit, which affects how likely noise from the internal nodes can propagate to the DFF with proper timing and enough strength. Noise from the combinational logic can become an observable error only if it is captured by the sequential elements. There exists three "masking effects" that all noise occurrences have to overcome in order to cause observable errors: logic masking, timing masking and electrical masking [10]. These masking effects exist regardless of the external noise disturbances but closely related to the logic/timing/electrical structures that can be analyzed in the early design phase to some degree of accuracy.

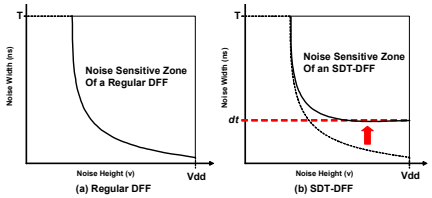


Figure 2 Noise Capture Zone (NCZ) of DFFs

The third factor is the distribution of noise occurrences inside the circuit: noise caused by different mechanisms is likely to occur at certain regions. Therefore, the endpoint DFFs of the paths on which noise are more likely to occur will potentially experience a higher rate of noise attack. This factor can be best described probabilistically because all noise occurrences are random and transient in nature so definitive description is unrealistic.

In [11], a technique of estimating the noise impact on DFFs was developed. It considered all the three factors by combining deterministic analysis of the inherent circuit noise-immunity and probabilistic description of noise occurrences. As shown in **Figure 3**, both the logic gates and noise occurrences were represented in matrices, and the noise-circuit interaction is

modeled as matrix transformation, where the three masking effects are modeled as matrix operations (such as "mapping", "reshaping"). This transformation process is repeatedly executed as the circuit netlist is searched. As the matrices are propagated a DFF, a "Noise Probability Density Function" (*NPDF*) is obtained to represent the distribution of noise that has survived all three masking effects at all internal nodes to reach the DFF. As the example shown in **Figure 4(a)**, an *NPDF* is represented in an 8x8 table where P_{ik} ($0 \le i, k \le 7$) represents the possibilities (normalized to 1000) of incoming noise with the height of between *[i/8, (i+1)/8]* of the supply voltage (V_{dd}) and the width of between *[k/16, (k+1)/16]* of the clock period (T).

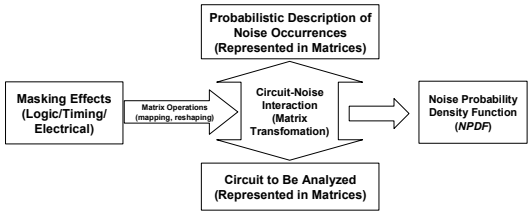
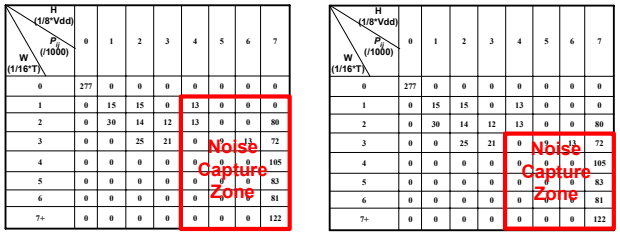


Figure 3 Robustness Calibration using Matrix Transformation



(a) Final NPDF and Noise Capture Zone *R(DFF) = 1000/591=1.69*

(b) Noise Capture Zone of an SDT-DFF *R(SDT-DFF) = 1000/485=2.06*

Figure 4 Using *NPDF* to Evaluate DFF *Robustness*

The destination DFF can only capture noise located in its *NCZ*. If we project the NCZ onto the *NPDF*, the sum of all P_{ik}'s covered by the *NCZ* indicates the possibility for the DFF to capture noise propagating from the combinational logics. We use the *NPDF* and the *NCZ* to measure the noise-tolerant capability of a DFF using its "*Robustness*", defined as:

$$R(DFF) = \frac{\sum_{[i,k] \in NPDF} P_{ik}}{\sum_{[i,k] \in NCZ} P_{ik}} \qquad (1)$$

An SDT-DFF has higher *Robustness* than a regular DFF because of its smaller *NCZ*. For example, in **Figure 4(b)**, the *Robustness* of the SDT-DFF with $dt = 3/16 * T$ increases to 2.06 from 1.69 of the regular DFF. The *Robustness* of an SDT-DFF increases with insertion delay *dt*.

In order to measure the noise-immunity of the entire circuit, we define a "*Robustness Function*" ***RF*** of a circuit with *M* DFFs as:

$$RF = \sum_{j=1}^{M} w_j R(DFF_j) \qquad (2)$$

where w_j is an optional weighting factor for the j^{th} DFF, heuristically introduced to represent its functional significance.

The *Robustness Function* of a circuit can be improved by increasing *Robustness* of the individual DFFs, which can be realized through replacing regular DFFs by SDT-DFFs. Unfortunately, this is at the cost of longer timing path and larger area. In high-performance circuit design, it is not always acceptable to introduce excessive area overhead or to insert a large amount of delay on critical timing paths. Therefore, some algorithm is needed to find the insertion scheme that maximizes ***RF*** without violating design constraints. In the next section, we will develop such an algorithm.

4. ROBUSTNESS INSERTION

The "constraint-aware robustness insertion" methodology is to replace a judiciously-chosen set of DFFs by their SDT-DFF counterparts with proper insertion delay so that the overall robustness is optimally increased while the design constraints are not violated. We will consider two of the most important constraints in digital circuit design – timing and area. Other type of constraints (such as power) can be considered similarly. Since the related design change is limited and localized, we assume it will not drastically change the global place & route and the change in wire delay and routing area is negligible.

4.1 Problem Formulation

In this sub-section, we derive mathematical expression of the area/timing constraints and formulate the optimization problem.

The total area and delay cost of an SDT-DFF as compared with a regular DFF consists of two parts: the first part is due to the transistor duplication, with fixed area overhead δA_{dup} and timing overhead δT_{dup}; the other part is due to delay insertion. In reality, the different insertion delay can be realized by using different number of identical delay units, each having an area of δA_{buf} and a delay of δT_{buf}. Replacing DFF_j by an SDT-DFF with n_j delay units increase the total cell area $\Delta A(n_j)$ and delay $\Delta T(n_j)$ by:

$$\Delta A(n_j) = \begin{cases} 0, n_j = 0 \\ \delta A_{dup} + n_j * \delta A_{buf}, nj \neq 0 \end{cases} \qquad (3)$$

$$\Delta T(n_j) = \begin{cases} 0, n_j = 0 \\ \delta T_{dup} + n_j * \delta T_{buf}, n_j \neq 0 \end{cases} \qquad (4)$$

Considering a design with *M* DFFs, let ΔA_{total} be the total allowed area overhead and ΔT_{oh} be the allowed extra delay (as a result, the clock period increases from T to $T+\Delta T_{oh}$) that the designer is willing to sacrifice in order to increase circuit reliability. The timing and area constraint can be expressed as:

$n_j \le (n_j)_{MAX}$, with $(n_j)_{MAX}$ determined by :

$$\delta T_{dup} + (n_j)_{MAX} * \delta T_{buf} \le t_{avail}(j) + \Delta T_{oh}, \quad 1 \le j \le M \qquad (5)$$

$$\sum_{j=1}^{M} \Delta A(n_j) \le \Delta A_{total} \qquad (6)$$

where $t_{avail}(j)$ is the minimum timing slack among all timing paths with DFF_j as the endpoint, and $(n_j)_{MAX}$ is the maximum number of delay units allowed at DFF_j.

These design constraints should be obtained from application-specific requirement and design analysis. Area overhead can not be totally avoided so $\Delta A_{total} > 0$. For designs with tight speed requirement, ΔT_{oh} can be set as zero, indicating that SDT-DFFs can be only used to replace DFFs with enough timing slack. With these design constraints, we formulate the constraint-aware robustness optimization problem as:

Given a circuit with M DFFs, find a assignment $\{n_j\}$ ($1 \le j \le M$) to replace the j^{th} DFF by an SDT-DFF with n_j delay units so that total Robustness Function defined in equation (2) is maximized, subject to timing constraint as defined in equation (5) and area constraint as defined in equation (6).

Since ***RF*** monotonically increases with the number of inserted delay units, the problem possesses the feature that the final optimal solution contains optimal solutions to its sub-problems,

we may consider this optimization process as a multi-constrained version of money allocation problem [12] and employ a dynamic programming technique [13] to solve it.

4.2 Dynamic Programming Solution

It is obvious that the optimality of $RF_j(n)$ requires its sub-function $RF_{j-1}(n-n_j)$ to be optimal. To break it down to sub-problems, we construct the recursive function as:

$$RF_j(n) = \max_{\substack{0 \le n_j \le (n_j)_{MAX} \\ \sum_{i=1}^{n} \Delta A(n_i) \le \Delta A_{total}}} \{RF_{j-1}(n-n_j) + w_j R(n_j)\} \qquad (7)$$

where $RF_j(n)$ is the *Robustness Function* when n delay units are inserted into the first j DFFs;
n_j is the number of delay units inserted at the j^{th} DFF;
$R(n_j)$ is the robustness of the j^{th} DFF with n_j delay units;
w_j is the robustness weighting factor of the j^{th} DFF;
$(nj)_{MAX}$ is the maximum allowed delay units at the j^{th} DFF;
$\Delta A(n_j)$ is area overhead of the j^{th} DFF with n_j delay units;
ΔA_{total} is the total allowed area overhead;

To solve this optimization problem, the first step is to search for the optimal ***RF***, described by ***Calculate_Maximum_Robustness*** in **Figure 5**. First, the available area that can be consumed (a_{avail}) is initialized to the total area constraint ΔA_{total} (line 1) and the total number of inserted delay units ***n*** is set to zero (line 2). In each iteration of the main loop (line 3-18), one DFF is being considered for optimization. Line 4-5 calculates the number of delay units (n_{limit}) allowed for the current DFF based on timing ($(n_j)_{MAX}$) and area constraint (a_{avail}). If no more delay unit is allowed, the *Robustness Function* and the delay assignment remain unchanged (line 6-8). Otherwise, as the inner loop (10-18) iterates through each allowed value of delay unit number, the new *Robustness Function* is calculated. If it produces a better result, the *Robustness Function*, the total number of delay units and available area are updated accordingly (line 15-18). A two-dimensional array ***TB_DIR[n,jj]*** is maintained to remember the decision. If no more delay unit is to be inserted to the current DFF, the value is set to be "***Left***", otherwise it is set to be "***Up***". This array will be used to retrieve the optimal insertion scheme.

```
Calculate_Maximum_Robustness(M, ΔA_total, {(n_j)_MAX})
1.  a_avail <= ΔA_total
2.  n<-0
3.  for j<-1 to M
      /* Calculate the available number of delay units allowed */
4.    n_avail = int((a_avail - δa_dup)/δa_buf)
5.    n_limit = min {n_avail, (n_j)_MAX}
6.    if (n_limit == 0) then
        /* If no more delay is allowed, the total robustness function remains unchanged */
7.      RF_j(n) <- RF_j-1(n)
8.      TB_DIR[n,j] <- "left"
9.    else
10.     for n_j <-0 to n_limit
11.       if (RF_j-1(n-n_j)+w_jR(n_j)) = RF_k-1(n)) then
            /* If the insertion of n_j delay units does not increase the robustness function,
               nothing is changed */
12.         RF_j(n) <- RF_j-1(n)
13.         TB_DIR[n,j] <- "left"
14.       else
            /* Otherwise, adopt this insertion and update the robustness function and remaining
               area constraint. The traceback path is updated as well */
15.         RF_j(n) <- RF_j-1(n-n_j) + w_jR(n_j)
16.         TB_DIR[n,j] <- "up"
17.         a_avail <- a_avail - Δa(n_j)
18.         n <- n + n_j
```

Figure 5 Pseudo-Code: Calculate_Maximum_Robustness

The second step, as described by ***Find_Delay_Assignment*** in **Figure 6**, is to find the final assignment of delay units for all DFFs by back-tracing the TB_DIR array. A one-dimensional array ***DU[jj]*** is used to store the number of delay units inserted at the j^{th} DFF. Starting from executing ***Find_Delay_Assignment(K,M)***, with ***K*** being the total number of inserted delay units and ***M*** being the total number of DFFs, whenever an "***Up***" (line 3) is encountered in ***TB_DIR[n, j]***, ***DU[j]*** is incremented by 1 (line 5), indicating one more delay unit is to be inserted to the j^{th} DFF, and the trace-back continues for the same DFF (line 4). Otherwise, the trace-back proceeds to the $(j-1)^{th}$ DFF (line 7). The recursion ends when either all DFFs are considered or the allowed number of delay units is reached (line 1-2), when ***DU[j]*** gives the number of delay units to be inserted to the j^{th} DFF.

```
Find_Delay_Assignment (n, j)
1.    if n=0 or j = 0
        /* End of back tracing */
2.      return
3.    if TB_DIR[n,j] == "Up"
4.      Find_Delay_Assignment(n-1,j)
      /* Insert one more delay unit to the j^th DFF */
      /* Note: SX[j] should be initialized to zero before back-tracing. */
5.      DU[j] <-DU[j]+1
6.    else
      /* Do not insert more delay unit to the j^th DFF */
7.      Find_Delay_Assignment(n,j-1)
```

Figure 6 Pseudo-Code: Find_Delay_Assignment

4.3 Implementation Framework

We developed a highly integrated "Constraint-Aware Robustness Insertion" framework, as shown in **Figure 7**. We first fabricated SDT-DFF cells with 1, 2 or 3 delay units of δT_{buf} =100ps for every regular DFF in a 0.18µm standard cell library and saved them in the "Noise-Tolerant Sequential Cell" database. The "Noise Capture Zone" of all DFFs was also pre-calibrated using HSPICE simulation so that they can be quickly referenced during analysis. Given a "Circuit to be hardened", the "Robustness Calibration Engine" derives the *NPDF*s, calibrates the *Robustness* of each DFF as a function of the inserted delay and calculates the *Robustness Function*. The area and timing constraints are derived by the "Constraint Generation Engine" based on circuit analysis (static timing analysis using Synopsys PrimeTime™) and the "Design Specification". The "Robustness Optimization Engine", implemented in C, uses the dynamic programming algorithm described in 4.2 to find the optimal delay units assignment for all DFFs. The selected DFFs are replaced by the "Noise-tolerant Sequential Cells" (SDT-DFFs) with proper delay values.

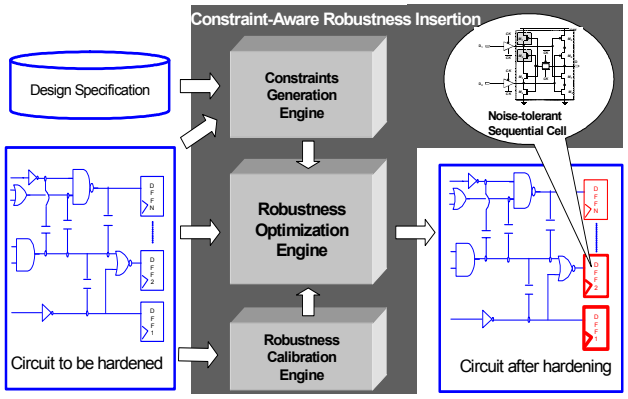


Figure 7 Constraint-Aware Robustness Insertion Framework

Although we chose to use SDT-DFF and *NPDF* transformation, it is worth emphasizing that the optimization methodology does not depend on particular choices of the hardening cell design and the robustness calibration technique. Other choices of hardening cells

are widely available, such as the Code Word State Preserving (CWSP) [14] and an alternative robustness calibration choice is fault simulation. We chose to use SDT-DFF and the *NPDF* transformation due to their advantages discussed above. However, both have limitations. For example, the temporal redundancy required by the SDT-DFF is still relatively high and accuracy of the *NPDF* transformation depends on the availability of an accurate statistical noise distribution. Nevertheless, they both greatly facilitate the execution of the optimization algorithm.

5. EXPERIMETAL RESULTS

5.1 Accuracy: Robustness Calibration Engine

We first verified the accuracy of the robustness calibration engine based on *NPDF* transformation by comparing with HSPICE simulation result. Due to the long simulation time of HSPICE, we could only use a small circuit CUT0 with 4 primary inputs (PIs), 12 internal nodes, 21 combinational gates and 8 DFFs. The clock period was set to be 1.6ns. During the simulation, all 2^4=16 possible input vectors were used; a large amount of random glitches were injected on each circuit node; errors observed in each DFF were counted. The *Robustness* of the DFF was obtained as the ratio of the total number of injected noise to the number of errors captured in the DFF and was compared with the *Robustness* calculated using the "Robustness Calibration Engine". Without much application-specific information about the circuit, the weighting factor w_i's in equation (2) were all set to 1. The result is shown in **Figure 8**, from where we can see that the *NPDF* transformation technique can indeed evaluate the robustness of DFFs to a high degree of accuracy.

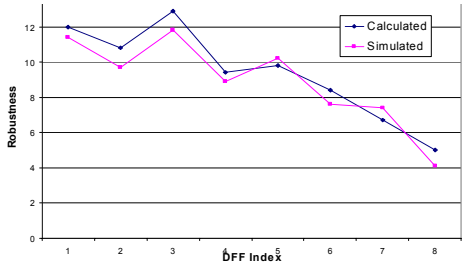


Figure 8 Comparison of DFF *Robustness*

5.2 Performance of the SDT-DFF Design

Next, we implemented the SDT-DFF cell designs and examined its area and timing cost. For the convenience of description, we will focus on only one type of DFF, for which we have developed 3 SDT-DFF counterparts with delay of 100ps, 200ps and 300ps, using 1, 2 and 3 delay units with δT_{buf}=100ps, respectively. **Table 1** lists the results: The column labeled "Ordinary DFF" shows the actual area (in library unit) and timing overhead (zero) of the original DFF. The next three columns shows the actual area and timing overhead for SDT-DFF design with 1, 2 and 3 delay units, respectively. The values of δA_{dup}, δA_{buf}, δT_{dup} defined in section 4.1 are all derived and listed above the table.

Table 1 Area and Timing Overhead of SDT-DFFs

δA_{dup}= 29, δA_{buf} = 13, δT_{dup} = 100ps, dt = 100ps

	Ordinary DFF	SDT-DFF 1	SDT-DFF 2	SDT-DFF 3
Area	70	112	125	138
ΔT	0	200ps	300ps	400ps

5.3 Optimized Robustness Insertion

We applied our methodology to a variety of circuits. The first circuit we studied was CUT0 introduced in 5.1. We first generated design constraints and calibrated *Robustness* of all 8 DFFs, as listed in **Table 2**. We arbitrarily set ΔA_{total} to 10% of the total area (118 library units) and ΔT_{oh} to 0, allowing no speed penalty. The second column shows the available timing slack, and the third column is the number of allowed delay units for each DFF fitting into the available timing slack. The *Robustness* of the original DFFs is in the fourth column. In column 5-7, $R_k(i)$ (k=1,2,3 and $1 \le i \le 8$) represents the *Robustness* of the i^{th} DFF if replaced by an SDT-DFF with k delay units, respectively.

Table 2 Robustness Calibration of CUT0

A_{total} = 1180, $\Delta A_{total} = A_{total}$*10%=118, ΔT_{oh}=0

DFF	Timing Slack	Allowed DU	$R_0(i)$	$R_1(i)$	$R_2(i)$	$R_3(i)$
1	227ps	1	12.0	15.7	36.9	57.2
2	379ps	2	10.8	13.5	32.2	59.4
3	318ps	2	12.9	15.2	27.4	43.1
4	304ps	2	9.4	14.2	32.9	50.0
5	83ps	0	9.8	11.9	28.4	49.2
6	48ps	0	8.4	11.0	27.1	52.3
7	254ps	1	6.7	8.6	19.7	31.1
8	122ps	0	5.0	7.7	18.1	29.9

We then applied the optimization algorithm and recorded the execution detail in Error! Not a valid bookmark self-reference.: column labeled as $\boldsymbol{RF_j(n)}$ ($1 \le j \le 8$) shows the *Robustness Function* after the first j DFFs have been considered and n delay units have been inserted during the execution of the process ***Calculate_Maximum_Robustness***. The "↑"s and "←"s indicate the traceback path when executing the ***Find_Delay_Assignment*** process. The final insertion scheme is listed in the last row: SDT-DFFs with 2 delay units are used to replace the 2nd and 4th DFF. The maximum value of the *Robustness Function* achieved under this insertion scheme is located at location $\boldsymbol{RF_8(4)}$ = 119.9. The area overhead is 110 library units, meeting the 10% constraint.

Table 3 Growth of the *Robustness Function*: CUT0

n	$RF_1(n)$	$RF_2(n)$	$RF_3(n)$	$RF_4(n)$	$RF_5(n)$	$RF_6(n)$	$RF_7(n)$	$RF_8(n)$
0	12.0	22.8(←)	35.7	45.1	54.9	63.3	70.0	75.0
1		25.5(↑)	38.4	47.8	57.6	66.0	72.7	77.7
2		44.2(↑)	57.1(←)	66.5(←)	76.3	84.7	91.4	96.4
3				71.3(↑)	81.1	89.5	96.2	101.2
4				90.0(↑)	99.8(←)	108.2(←)	114.9(←)	**119.9(←)**
DU	0	2	0	2	0	0	0	0

For the small circuit with 8 DFFs, we were able to enumerate all possibilities allowed by the area and timing constraints to verify that the insertion scheme found by the Robustness Optimization Engine resulted in the most *Robustness Function* improvement. We repeated the HSPICE simulation described in 5.1 using the hardened circuit in which the 2nd and 4th DFFs are replaced by SDT-DFFs with 2 delay units. The number of errors detected in the 2nd and 4th DFFs reduced by **83%** and **85%**, respectively. The total number of errors in all DFFs is reduced by **46%**.

We then applied our methodology to other five circuits with various sizes. Among them, CUT1-CUT4 are logic units widely used in digital circuits and CUT5, with 97 DFFs, 338 input ports and 3156 internal nodes, is a key functional block extracted from

the commercial state-of-the-art configurable and extensible Xtensa™ processor [15]. Since speed is usually considered a more crucial requirement than area, we set ΔT_{oh} to 0 and released the area constraints in all experiments. The area overhead was then calculated to measure the protection cost. In all experiments, the clock period was set to 1.6ns.

The results are listed in **Table 4**: Column 2-5 show the number of DFFs, area, timing constraints and total number of allowed delay units, respectively. Columns 6-8 show the original value, optimized values and improvement of the *Robustness Function*, respectively. The last column shows the associated area overhead. As we can see, ***RF***s of these circuits were improved significantly (from 25% for CUT2 to 96% for CUT4) with no timing penalty and reasonable area overhead. The difference in improvement is because these circuits have different timing conditions and the more positive timing slack, the more improvement can be achieved. It also showed that the area overhead decreases as the circuit becomes larger because usually the larger the circuit, the less portion of the non-combinational (DFFs) area (in the brackets in column labeled "Total Area", with the exception of CUT5).

Table 4 Robustness Optimization Results

CUT	No. DFF	Total Area (Lib. Unit)	ΔT_{oh}	No. DU	Orig. RF	Opt. RF	ΔRF	ΔA
CUT1	8	1396(40.1%)	0	8	62	103	66%	19%
CUT2	8	2490(22.5%)	0	9	103	129	25%	10%
CUT3	5	2971(11.8%)	0	5	136	203	49%	5.6%
CUT4	11	7479(10.3%)	0	10	280	549	96%	3.8%
CUT5	97	39894(17.0%)	0	27	2561	3559	39%	2.7%

5.4 Robustness-Cost Trade-off

In order to further understand the trade-off between circuit robustness and protection cost, we repeated the optimization process in the largest circuit CUT5 with different design constraints. In this experiment, we gradually increased ΔT_{oh} from 0 to 300ps without setting the area constraints to see how much the *Robustness Function* could be improved. The achieved ***RF*** improvement (ΔRF) and the associated area overhead (ΔA) were plotted against the allowed timing overhead (ΔT_{oh}) in **Figure 9**.

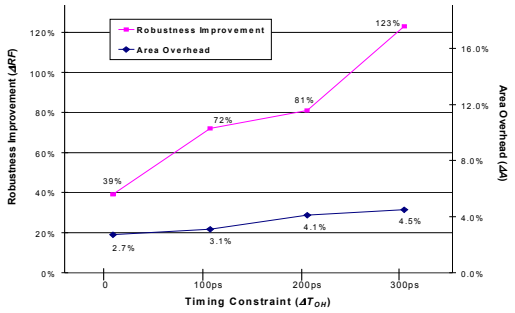


Figure 9 Robustness-Cost Trade-off in CUT5

As we expected, the *Robustness Function* increases as the timing constraint is becoming less strict because more DFFs will have timing slack for more delay units. However, the robustness enhancement is not linearly increasing with ΔT_{oh}. As an example, increasing ΔT_{oh} from 100ps to 200ps only results in 9% increase in the *Robustness Function*, whereas increasing ΔT_{oh} from 200ps to 300ps causes a 42% improvement. Therefore, identifying the most economical scheme requires careful investigation of the trade-off between the robustness improvement and related protection cost. Finally, we noticed that the area penalty does not increase quickly and remains at a low level (2.7%-4.5%) because sequential elements occupy a small percentage of the chip area. This proves that robustness insertion to protect DFFs is an area-efficient approach to achieve high reliability.

6. CONCLUSION

In this paper, we proposed an efficient "Constraint-Aware Robustness Insertion" methodology for optimal robustness insertion. Using cost-effective hardening sequential cell design and circuit robustness calibration, the robustness optimization algorithm is able to find the optimal scheme to increase the circuit noise-tolerance while keeping related cost within area and timing constraints. The proposed methodology will greatly facilitate the quest for cost-effective solution to reliable VLSI circuit design.

REFERENCES

[1] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 2001

[2] J. Cong, D. Z. Pan, and P. V. Srinivas, "Improved crosstalk modeling for noise constrained interconnect optimization," *Proc. ASP-DAC*, *pp*. 373-378, 2001.

[3] Shen Lin, Chang N., "Challenges in power-ground integrity," *Proc. ICCAD'01*, *pp*. 651-644, 2001.

[4] J.-J. Liou, A. Krstic, Y.-M. Jiang and K.-T. Cheng, "Modeling, Testing, and Analysis for Delay Defects and Noise Effects in Deep Submicron Devices," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 6, pp. 756-769, Jun. 2003.

[5] Hess C, Stine BE, Weiland LH, Sawada K., "Logic characterization vehicle to determine process variation impact on yield and performance of digital circuits," *Intl. Conf. Microelectronic Test Structures, pp*. 189-196, 2002.

[6] Peter Hazucha, Christer Svensson, "Cosmic-Ray Soft Error Rate Characterization of a Standard 0.6-µm CMOS Process," IEEE *Jnl. Solid-State Circuits*, vol. 35, no. 10, Oct. 2000.

[7] Anghel, L., Nicolaidis, M., "Cost Reduction and Evaluation of a Temporary Faults Detecting Technique," *DATE'00*, *pp*. 591-598, 2000.

[8] E. Dupont, M. Nicolaidis, and P. Rohr, "Embedded Robustness IPs for Transient-Error-Free ICs," *IEEE Design & Test of Computers*, *pp*.56-70, May-June, 2002.

[9] Y. Zhao, S. Dey, "Separate Dual Transistor Register-an Circuit Solution for on-line Testing of Transient Errors in UDSM-IC," *Proc.IOLTS, 2003, pp.*7-11, 2003.

[10] C.Zhao, X. Bai, S.Dey, "A scalable soft spot analysis methodology for compound noise effects in nano-meter circuits," in *Proc. DAC'04, pp. 894-899,* 2004.

[11] C.Zhao, X. Bai, S.Dey, "A Static Noise-Impact-Analysis Methodology for Evaluating Transient Error Effects in Digital VLSI Circuits", *Research Report,* http://esdat.ucsd.edu/~chong/nia.pdf.

[12] T. C. Hu and M. T. Shing, *Combinatorial Algorithms, Dover Publications, Inc., pp*.111-113, 2002.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithms", Ch 15, McGraw-Hill, 1990.

[14] Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," *Proc. VTS*, *pp.* 86-94, 1999.

[15] http://www.tensilica.com/xtensa_overview_handbook.pdf, *Xtensa™ Microprocessor Overview Handbook*, Tensilica Inc, August 2001