J Electron Test (2012) 28:201–214
DOI 10.1007/s10836-011-5268-x

A Layout-Aware Pattern Grading Procedure for Critical Paths Considering Power Supply Noise and Crosstalk

Junxia Ma · Mohammad Tehranipoor · Patrick Girard

Received: 10 November 2010 / Accepted: 2 November 2011 / Published online: 17 December 2011

Abstract Power supply noise and crosstalk are the two major noise sources that are pattern dependent and negatively impact signal integrity in digital integrated circuits. These noise sources play a greater role in sub-65nm technologies and may cause timing failures and reliability problems in a design; thus must be carefully taken into consideration during test pattern generation and validation. In this paper, we propose a novel method to evaluate path-delay fault test patterns in terms of their ability to cause excess delay on targeted critical paths. It quantifies the noises with a pattern quality value (Q) using the activated aggressor gates and nets information. The proposed method offers design engineers a quick approach to evaluate the critical paths in static timing analysis (STA) and silicon to improve timing margin strategies. By evaluating the failed test pattern, the proposed method can be used to help identify the root cause during failure analysis. Simulation results demonstrate the efficiency and effectiveness of the pattern grading procedure.

Keywords Signal integrity · Crosstalk · Power supply noise · Pattern grading · Path delay test

Responsible Editor: A.D. Singh

This work was supported in part by National Science Foundation grant CCF-0811632.

J. Ma (✉) · M. Tehranipoor
ECE Department, University of Connecticut, Storrs, CT 06269-2157, USA
e-mail: junxia@engr.uconn.edu

M. Tehranipoor
e-mail: tehrani@engr.uconn.edu

P. Girard
LIRMM, Université de Montpellier II/CNRS, 34095 Montpellier Cedex 5, France
e-mail: girard@lirmm.fr

1 Introduction

As feature size of devices and interconnects further shrinks, silicon chip behavior becomes more sensitive to on-chip noise, process and environmental variations, and uncertainties. The reduced power supply voltage impacts the circuit noise immunity, further worsening the signal integrity (SI) issues, which may cause (1) in-field reliability problem and (2) reduced yield due to functional failure and timing-related failure. As is known, the power supply noise and crosstalk effects significantly increase as technology scales to 65 nm and below. The number of silicon failures and escapes caused by signal integrity is on the rise, because existing design tools and test methodologies cannot fully address these issues effectively. Therefore, it is important to generate patterns that can be used to evaluate the noise impact on signal integrity before tape-out, and to excite the noise events during silicon validation and production test to capture the potential chip failures. However, it is not possible to take into account these noise sources/uncertainties by running current ATPG tools because of the pattern dependent characteristics of these noise sources. Considering the interactions

and inter-dependencies between these noise effects, it is extremely difficult to develop an efficient SI-aware ATPG flow.

On the other hand, during failure analysis, if the failing mechanism is identified as noise-related reasons, it is beneficial to know (1) how much impact each noise source has on path delay and (2) the noise level of each effect. This information is important for identifying the root cause of the failure and determining whether it fails due to bad design or bad test pattern. A pattern grading procedure is needed to evaluate the failed pattern and report the impact of each noise on path delay.

Motivated by the reasons presented above, we propose a fast, layout-aware pattern grading procedure for critical path testing and diagnosis considering noises in the circuit. In this work we focus on path delay faults (PDFs) for efficient critical path delay testing and reducing test escapes; however it can be extended to grade transition delay fault (TDF) patterns as well. The 0/1/random/adjacent-fill methods commonly used in practice may not necessarily cause sufficient power supply noise and crosstalk on paths under test, which would allow the chip to pass manufacturing test but fail in the field. In general, it is easy to randomly fill the unspecified bits in a PDF pattern first, and then by grading the resulting patterns based on their noise effects the most effective ones can be selected. In this way, we would be able to generate a pattern that increases "multiple" noise effects. This is the main motivation of this work. For the application to diagnosis, the proposed pattern grading technique is applicable to both TDF and PDF tests. In this paper, we target power supply noise (PSN) and crosstalk effects during pattern grading.

1.1 Related Prior Work

Several works have addressed power supply noise during pattern generation. Genetic-algorithm-based methods for finding patterns that induce maximum supply noise are proposed in [7, 25]. In [7], randomly filled patterns are generated, then waveform simulations are performed to find the pattern that yields the largest supply noise. A combination of Monte Carlo and genetic algorithm search for identifying the worst-case input vector pairs that induce maximum switching noise is implemented in [25]. A method for measuring power called switching cycle average power (SCAP) was proposed in [1]. Ma et al. [12] proposes a method to generate path-delay fault patterns which maximize the supply noise effects on critical paths by taking into account the switching location in the circuit layout.

There are also some approaches proposed to deal with crosstalk issues during verification and test. Authors in [15] propose test generation for crosstalk-induced delay and noise for system-on-chip (SOC) interconnects. Closed form equations are derived in [5] that quantify the severity of crosstalk effects and describe qualitatively the dependence of crosstalk effects on the values of circuit parameters, the rise/fall times of the input transitions, and the skew between the transitions. The analysis was used as a basis for pattern generation to induce delay due to crosstalk in [3]. However, this approach only generates patterns for a single aggressor affecting a target path. The procedure proposed in [8] considers a genetic-algorithm based approach when inducing crosstalk into delay test patterns. A pattern generation procedure that magnifies crosstalk effects on critical paths by maximizing switching on nearby nets is presented in [9]. Pattern generation methodologies considering crosstalk and transition arrival time were proposed in [4, 16], but they lack the immediate use in practice since they are computationally intensive.

Pattern grading techniques to select best TDF pattern set for small delay defects testing (SDDs) are proposed in [22] and [14]. In these works, authors performed N-detect TDF ATPG first and then select the patterns that activates the longest paths that have less slack and thus have a larger chance to detect the SDD defects. They are not applicable to PDF test.

1.2 Contributions and Paper Organization

The test techniques mentioned above may be efficient for power supply noise or crosstalk individually, however, no solution takes both effects into account during PDF test. For crosstalk, the direction of transitions on aggressor and victim nets as well as the transition arrival time of aggressor with respect to the victim are important. However, for power supply noise, the location of transitions on aggressor gates in the physical layout, and direction of transition on aggressor gates are important. It is very hard to develop a method that can fill the don't-care bits deterministically to maximize both noise effects. Due to the different way these noises impact path delay, a pattern that maximizes one noise induced delay may not necessarily increase the other one. For example, a path can speed up due to crosstalk noise caused by the same direction transitions in neighboring nets. While more switching increases power supply noise and slows down the gates, it may also speed up the path due to the crosstalk noise it creates. Besides, power supply noise can change the arrival time on the crosstalk aggressor nets, and this

makes the pattern filling effect very hard to predict for crosstalk noise. In fact, our analysis have shown that in some cases maximizing one effect may minimize the other, so it is not trivial to find the best pattern when targeting both effects.

A path delay can be between d_{min} and d_{max} when tested by a PDF pattern generated by ATPG tools. The reason for the difference between path delay is pattern-induce noise in the circuit. The path delay would be d_{min} when the pattern (1) does poorly on increasing noise-induced delay, and (2) activates the power supply noise and/or crosstalk effect such that it speeds up the path. On the other hand, the path delay would be d_{max} when the pattern maximizes the two effects and increases the path delay. The path delay range (d_{max}–d_{min}) can be large for lower technology nodes.

In this paper, we propose a novel and efficient pattern grading procedure, which can be used to:

- Fast evaluate the patterns in terms of their ability to excite power supply noise and crosstalk induced delay on critical paths. By taking into account the power distribution network structure, considering coupling and parasitic information, and collecting switching information (location, direction, and arrival time) on the physical layout, a novel pattern quality metric is developed to estimate the power supply noise and crosstalk joint effects on the path under test for each pattern.
- Select the best patterns that magnify the power supply noise and crosstalk effects during PDF testing. It enables the test engineers to apply the most effective patterns early during first silicon validation and later deploy them in production test as effective screens.

And the best patterns selected by this method will help to:

- Address the issue of mis-binning. Nowadays, ASICs are also performed speed binning as what has been done for microprocessors [2]. The authors in [23] show that speed binning results obtained from structural testing of selected critical paths correlate closely with results from functional testing. Patterns could result in different path delay due to their own noise levels. Since they generate the worst-case supply noise and crosstalk effects on target paths, the patterns selected by this procedure can be used to characterize the speed path and address the mis-binning problem.
- Set proper margining. It offers the design engineers a quick approach to evaluate the critical paths in static timing analysis (STA) and silicon to improve timing margin strategies. If proper margins are set during design phase, the yield could also be potentially improved.
- Minimize test escapes of timing marginalities due to noise conditions. If the test patterns during production test do not have the worst-case noise level, there may be a pattern in field that has larger noise and fails the chip. In this case, the chip would be under-tested (tested optimistically). To avoid such an escape, the patterns selected by this procedure can be used for the production test.

The proposed grading procedure is applicable to both Launch-off-Shift (LOS) and Launch-off-Capture (LOC) patterns. In this paper, we use LOC method to generate PDF patterns. Since the second vector of LOC patterns is a functional response of the first vector, there is a good chance that the pattern is functionally valid [11, 24]. Note that in this work we do not address the issue of critical path selection. We assume that critical paths are selected using static or statistical static timing analysis tools.

This paper is organized as follows. Section 2 describes our layout-aware analysis of power supply noise and crosstalk effects and discusses the victims and aggressors for each noise. Section 3 elaborates the proposed integrated methodology for pattern grading and selection considering both noises. The applications of the proposed pattern quality value Q are also discussed in this section. In Section 4, experimental results and analysis are presented. Finally, the concluding remarks are given in Section 5.

2 Analyzing Pattern-Dependent Noise Sources: Power Supply Noise and Crosstalk

A path delay consists of two parts: gates delay and interconnects delay. The gate delay is mainly impacted by voltage drop and the interconnect delay is mostly impacted by both power supply noise and crosstalk noise. The increase in gate and interconnect delay has a direct impact on the path delay. In this section we present layout-aware analysis for these two noise sources. Based on such analysis, the identification of victims and aggressors for both noises is also discussed. Since both noise sources have pattern dependent characteristics, the activated aggressor gates and nets are used to quantify the power supply noise and crosstalk noise, respectively, in the proposed pattern grading method. The knowledge of aggressors is important to understand the proposed method.

2.1 Power Supply Noise Analysis and Aggressor Gates Identification

Power supply noise (PSN) includes two major components: inductive ($L\frac{dI}{dt}$) and resistive power/ground voltage noise (IR). Since the inductance, L, and resistance, R, of the power/ground distribution network can be considered fixed for a given layout and package, large changes in average current and instantaneous current would be the significant issues which contribute to high power supply noise. In typical CMOS integrated circuits, instantaneous current is mostly caused by gate switching. When the switching activity in a circuit increases, the current it draws from the power distribution network (PDN) will also increase, resulting in increased voltage drop. The PSN decreases the effective supply voltage reaching a gate which will reduce its driving strength and thus increases gate delay. In this paper, we only focus on the resistive noise increase, i.e. total voltage drop on both power and ground networks. However, it can be extended to include inductance noise impact too.

To illustrate the relation between path delay and its associated voltage drop, we target a functionally testable critical path in s38417 benchmark. A PDF pattern is generated for the targeted path with don't-care bits unfilled using commercial ATPG tool, then the don't-care bits in the pattern are filled randomly with bit '0' probability increasing from 0 to 100% 1000 times to generate 1000 PDF patterns. For example, pattern 1 is all 1-fill pattern, pattern 1000 is all 0-fill pattern, and pattern 500 is filled with 50% bit 0/1. In this way, we create a pattern set with patterns having different power supply noise levels. During simulation, all the capacitors are connected to ground. Thus it eliminates the impact of crosstalk noise and focus on only the power supply noise's impact. We extract a post-layout SPICE netlist for the path under test using timing analysis tool. The new supply voltages for power and ground pin of cells on the targeted path are reported by rail-analysis tool during power rail analysis when applying the 1000 patterns as stimuli. We then back-annotate the new supply voltages into the SPICE netlist and perform SPICE simulation for the 1000 test patterns. Figure 1a shows the path delay distribution collected by *hspice* simulation. We can see that for the same path, patterns filled with different 0/1 ratios causes different path delay. The patterns in the middle part present longer path delays compared with 0-fill or 1-fill at both ends. For patterns filled randomly (50% bit 0/1 probabilities), the path delays still vary in a wide range. This figure also illustrates the need to select best test pattern for path delay test. In Fig. 1b, the correlation of voltage drop and path delay is found to be 0.99. This implies the voltage drop has direct impact on the path delay.

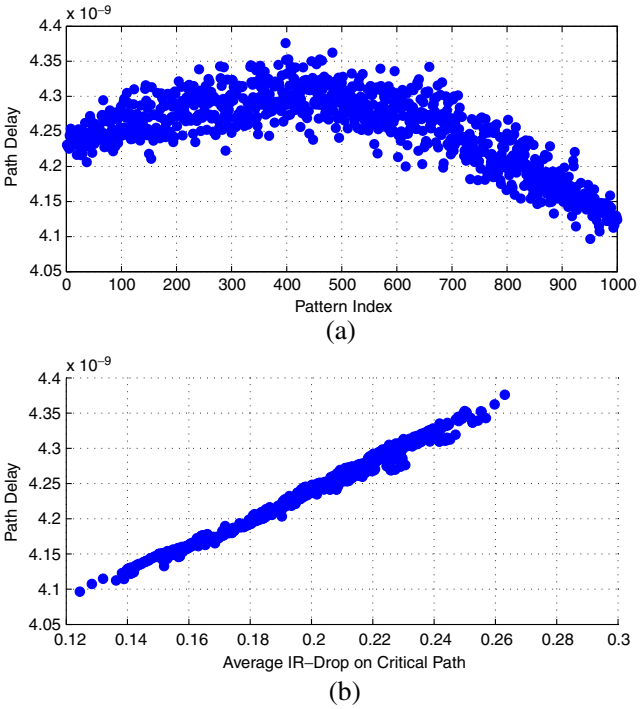

Fig. 1 Power supply noise impact on path delay (**a**) path delay distribution (**b**) correlation between IR-drop and path delay for one path in *s38417* benchmark for 1000 random PDF patterns; correlation coefficient is 0.99

Figure 2 illustrates a typical standard cell layout consisting of cells placed in rows and power/ground rails tied horizontally to the cells. Rows are placed back-to-back and share a common power/ground rail. Figure 2 also shows part of a critical path (Path X) going through this region. Cell G of this critical path is placed in between the power and ground rails as shown in the figure. Simulations have been performed to analyze localized IR-drop effects caused by switching cells that are in close proximity to one another in [12], which shows the closer the neighboring cell is to the switching cell, the larger the voltage drop created by the switching cell and experienced by the neighboring cell.

As shown in the figure, cells E, F, and K are placed in the same row as cell G. After applying a PDF pattern a transition on cell E will cause more voltage drop on G than that on cell K. We refer to cell G as "victim" cell and the neighboring cells as "aggressors" because their switching activity can impact the voltage drop and performance of the victim cell. If there are m gates on a critical path, we consider all of them to be victim cells; some may be in the same row and others in different rows.

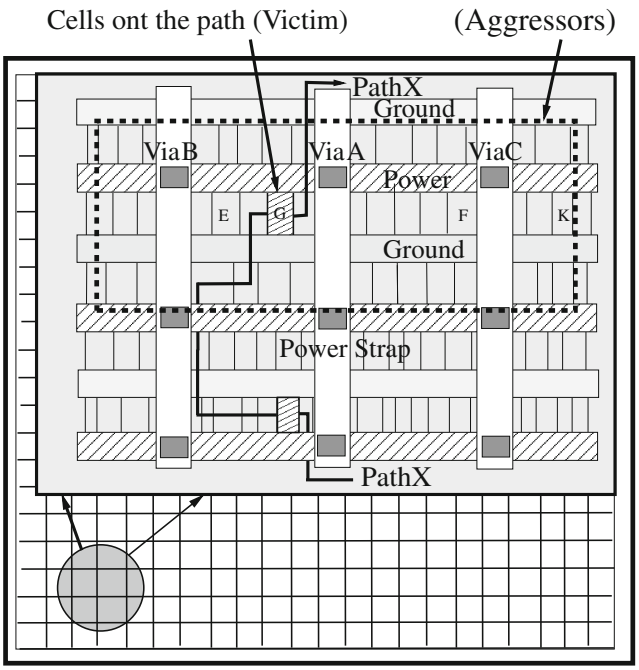

Fig. 2 Critical path cell (victim) and neighboring cells (aggressors) identification

As shown in the dashed box in Fig. 2, cells directly adjacent to *G* and those that extend beyond the right and left of *ViaC* and *ViaB*, respectively, are classified as aggressor cells. The range beyond the right of *ViaA* and the left of *ViaB* is determined by the amount of current drawn by cell *G* from those power vias when it switches. Based on our simulation results and analysis, the switching current from farther cells contributes negligibly to the voltage drop of the victim cell. As can be seen from Fig. 2, neighboring cells in three rows are included in the range for aggressor cells: cells in the same row, cells sharing same power and ground line with the victim cells. The switching of cells in these three rows has direct impact on the voltage drop on the victim cell. Their switching increases the current (I) flowing in the same power and ground line with the victim cell. Since R is a fixed value for the victim cell, the increased current (I) increases its IR-drop. We define the aggressor cell range using such analysis. The block with dashed lines in Fig. 2 shows all the aggressor cells for victim cell *G* used by the proposed pattern grading procedure.

Identifying the effective aggressor cells, whose switching will impact the delay of victim cell (e.g. cell *G*), would help evaluate the IR-drop impact on path delay more accurately and efficiently. It also reduces the complexity of our analysis as we will focus on much fewer aggressors rather than all neighboring cells which could be extremely large.

2.2 Crosstalk Analysis and Aggressor Nets Identification

As technology feature size scales, interconnect spacing and width are also being reduced. However, in order to keep the resistance low, the thickness of the wires is not scaled at the same rate. This produces tall sidewalls between long parallel interconnects separated by very little space, which creates parasitic coupling capacitance between wires. Due to this fact, crosstalk has become a significant contributor to signal integrity problems in modern designs.

As shown in Fig. 3, the delay from output of gate *G1* to the input of gate *G3* depends on how much time is needed to charge the grounded *C1* and the coupling *C2*. The time to charge grounded *C1* is known. However, the charge time for coupling *C2* depends on behavior of its neighboring net, which is referred to as 'aggressor' for the 'victim' net between gate *G1* and gate *G3*. If the aggressor net is rising at the same time as victim net, then the amount of charging needed for the coupling capacitor is 0, and the victim signal will transition faster than having a quiescent neighboring net. While, if the aggressor net changes in the opposite direction, the victim net needs to charge an almost $2C$ capacitance ($C1+C2$) and its transition time becomes longer. So the directions of the aggressor net transitions with respect to victim net can have a significant impact on the victim net's propagation delay. For multiple aggressors, varying combinations of rising and falling transitions around the victim net are possible to cause varying results [9].

The transitions on aggressor and victim lines cannot be assumed to always arrive simultaneously. Previous crosstalk analysis on buses have made this assumption to simplify the problem [6, 18, 19]. Although this is appropriate for buses due to the uniform structure and approximately simultaneous launching from the driving cells, this cannot be assumed for on-chip interconnects. On-chip detailed routing and the variety of gates that lie in the victim and aggressor paths are not nearly as uniform as a bus. For two parallel nets of two different

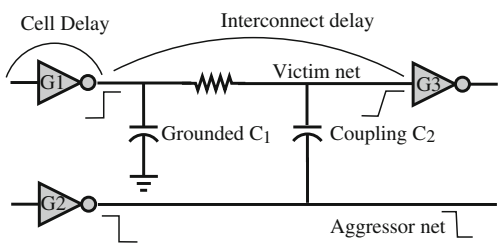

Fig. 3 Victim and aggressor nets for crosstalk

paths, a transition may have to travel through many different gates and several different metal layers delaying it hundreds of picoseconds behind a transition that only has to travel through a few gates and metal layers.

We ran SPICE simulations for two parallel interconnects at 180 nm technology node. The transitions on the victim net were staggered with respect to the aggressor net. These simulations were run with the victim rising and the aggressor either rising (Fig. 4a) or falling (Fig. 4b) to evaluate the effect on propagation delay of the victim. As the transition on the victim net occurs with a greater difference in time, which can be seen in *Region 1* of Fig. 4a and *Region 1* and *Region 4* of Fig. 4b, the victim net incurs almost no propagation delay due to the aggressor switching. However, as the transitions begin to coincide, there is either significant speed-up or slow-down depending on the direction of the aggressor transition. Even though the direction of the aggressor net transitions with respect to the victim net transition is important, it will have a significantly different impact if the transitions occur with as little as a couple hundred picoseconds apart. Similar conclusion has been derived for experiments using 0.35 μm technology in [5].

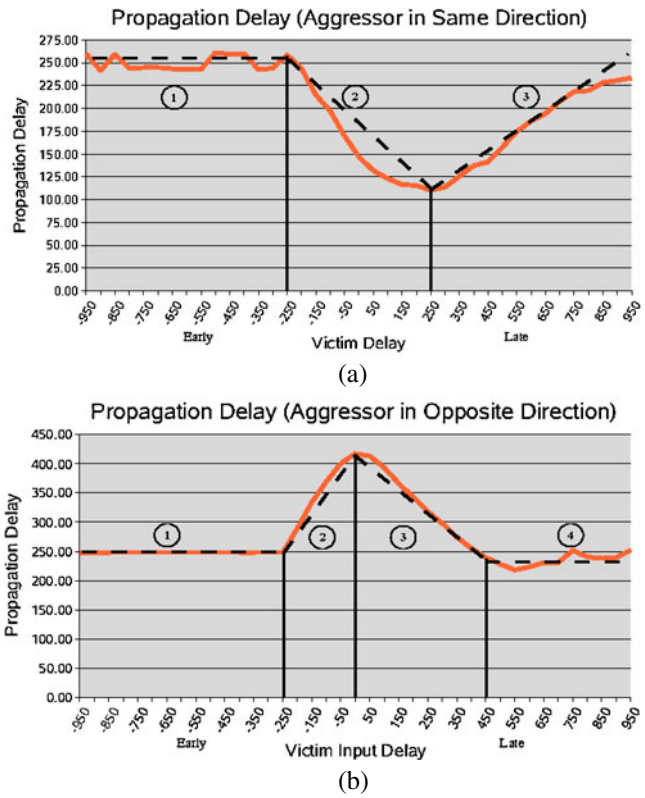

Fig. 4 Propagation delays of a rising transition on a victim net when (**a**) a rising transition and (**b**) a falling transition on an aggressor occurs for delays of the victim in the window of −950 ps to 950 ps

Although these timing simulations have only been performed for two minimally spaced interconnects, the slow-down or speed-up will scale in proportion to the coupling capacitance if the spacing is different. However, incorporating additional aggressors will not only affect the victim, but also affect the timing of the other aggressors that are affecting the victim. Even though these additional effects exist, these second-order effects are not included in our analysis. Our analysis considers each aggressor independent of each other and the crosstalk incurred by the victim is the superposition of crosstalk from all nearby aggressors. Although this simulation was done for a 180 nm technology node, the same trend is expected for lower technology nodes as well.

To evaluate the impact of crosstalk effects from the aggressors, we need to identify the aggressor nets for the critical paths, which requires knowledge of the physical design. Since the coupling capacitance is determined by both the space and the parallel distance of these nets, simply looking at all nets within a particular area of the targeted path will not necessarily identify nearby nets that have a significant crosstalk effect. For example, two nets that are routed closely together but for a very short distance will not necessarily create enough crosstalk to impact the timing of a delay-sensitive path.

As shown in Fig. 5, the relative coupling of the aggressors with the victim is shown by the size of the capacitor. The coupling value between the Victim and Aggressor *C* reflects the pitch and parallel distance between them. This is larger than the coupling capacitance between the Victim and Aggressor *A*, but less than the coupling between the Victim and Aggressor

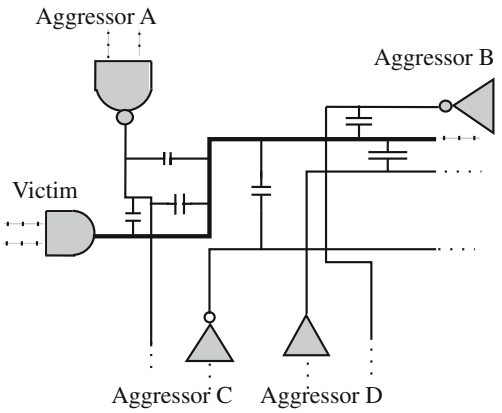

Fig. 5 The coupling capacitance effects on a victim line from four aggressors. The amount of coupling is represented in the size of the capacitor. For simplicity, coupling between aggressors are not shown

D. So, even though *A* is closer to the Victim than *C*, it is parallel for a very short distance, but since the pitch between *D* and the Victim is smaller, the coupling is larger although the parallel distance is shorter. Even though Aggressor *A* is physically closer than either Aggressors *C* or *D*, it may not be considered a neighboring net if the coupling capacitance value does not reach a *minimum coupling threshold*.

We use the extracted coupling capacitance of each of the net to identify those that will have a significant effect. A distributed RC model is used as the interconnect model during parasitic extraction. We use a *minimum coupling threshold* during 3D extraction of the layout to prune aggressors with coupling capacitance smaller than the threshold. Using the coupling threshold will reduce the complexity of our analysis by filtering some of the neighboring nets that have almost no effect on the victim path. This will eliminate those nets that may be near each other but are routed perpendicularly. The coupling threshold is defined as the minimum amount of capacitance between aggressor and victim that can impact the delay of transition traveling along the victim net.

3 PDF Pattern Grading Procedure

The common method to evaluate a pattern in terms of its noise effect is to run post-layout SPICE simulation. However, such simulation is usually extremely time-consuming and is not practical for today's designs. In this paper, we model the power supply noise and crosstalk noise for each pattern with a quality metric according to the way they impact the path delay. The metric represents the amount of noise the pattern introduces and can be calculated quickly and efficiently. Using this metric the method we proposed eliminates the extremely time-consuming SPICE simulation process and can grade the pattern efficiently. In this section, we present (1) the procedure to calculate the pattern quality value considering the noises it induces to the path under test; (2) the application of the pattern quality value, including selecting the best PDF pattern that causes the largest noises from a large number of patterns and grading patterns in terms of their abilities to excite noises on the path under test for diagnostic purpose.

3.1 Pattern Quality Value Calculation Procedure

The layout-aware quality value calculation procedure involves three major steps together with a preprocessing step as shown in the flow diagram in Fig. 6.

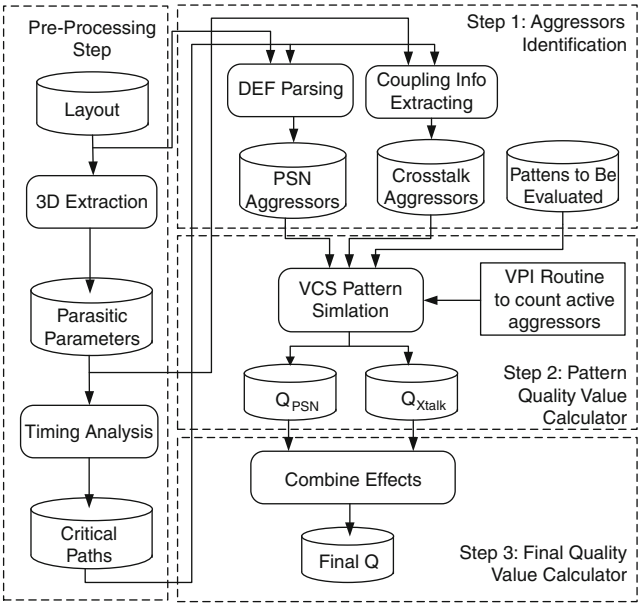

Fig. 6 Flow diagram of the layout-aware pattern quality value calculation procedure

In the pre-processing step, we extract the physical layout using a commercial tool and acquire wire resistance, capacitance to ground, and coupling capacitance. Then, the critical paths are identified using static timing analysis considering these parameters. The three steps for pattern quality value calculation consist of (1) Effective Aggressor Identification, (2) Pattern Quality Value (Q) Calculation for each effect, and (3) Final *Q* Value Calculation for each pattern. Each of these steps is described in the following.

Step 1 **PSN and Crosstalk Aggressors Identification:** Design exchange format (DEF) file is used to identify the cells of the path under consideration on the layout. The DEF file is generated from the layout design. It contains the physical placement information of the elements in the circuit. The library exchange format (LEF) file contains physical layout information of the standard cells, such as cell's width and height. We develop an in-house tool *DEFParser* in *C* programming language. The inputs to the *DEFParser* is DEF file, LEF File, and path file. By parsing the DEF file, the physical location of the cells on the specified critical path is identified. The *DEFParser* also gains information of the power distribution network information of the design through the DEF file. It collects information of the physical location (coordinates) for power ring, power straps, power vias and power rails in the layout. The *DEF-*

Parser first searches for victim cells in the DEF file and acquires their coordinates information. Then according to the range defined in Fig. 2, it reports all the aggressor cells to a text file. In this way, we identify the power supply noise aggressor cells for the path under test.

As for crosstalk noise, coupling report and timing analysis tool are exploited to identify the aggressor nets. In the pre-processing step, the coupling and parasitic information is extracted from the layout file. Then we use the timing analysis tool to parse the extracted file and process the coupling and parasitic capacitance. Custom scripts are developed in *TCL* language to be executed in timing analysis tool. For each interconnect net of the path under consideration, it reports the name of the coupling nets and its corresponding coupling capacitances between each pair of neighboring net segments. The resulting report is used to assist in identifying the effective neighboring nets to identify the crosstalk aggressor nets. If a net is coupled to a neighboring net in several locations, the tool will add the coupling capacitance into a single value. Only those segments with a coupling capacitance above a user-defined threshold will be identified as an aggressor net.

Step 2 **Pattern Quality Value Calculator:** After identifying the aggressor nets and gates, we will quantify the noise-induced delay based on the way the activated aggressors impact the path delay. When applying the patterns as stimuli to the circuit under test, aggressor gates and nets are activated by the input vectors and their switchings induce power supply noise and crosstalk noise on the victim cells. To monitor the transition of all the aggressors and detect the active aggressors, we use a verilog programming language interface (VPI) routine during gate-level verilog simulation. In this work, the verilog pattern simulation is done by the Synopsys VCS [17]. The VPI routine can also be used with other gate-level simulation tools. It provides a standard interface to the internal data of the design during simulation and can be used with any other industry standard verilog simulator. Through VPI, we acquire the internal signal switching information.

The metric named layout-aware weighted switching activity (*WSA*) [10] is used to calculate the pattern quality value in the presence of power supply noise. For gate k, the WSA_k will be dependent on the gate weight, τ_k, the number of fan-out of this gate, f_k, and the fan-out load weight, ϕ_k. We assign different weight to each gate based on its area size and cell type. The metric *WSA* for gate k in a circuit is calculated by:

$$WSA_k = d_k(\tau_k + \phi_k f_k), \text{ where}$$
$$d_k = \begin{cases} 1, \text{ Transition in required direction occurs} \\ 0, \text{ No transition} \end{cases} \quad (1)$$

The pattern quality value (Q) for power supply noise is the sum of the *WSA* value of all $N_{\text{aggr\_cell}}$ aggressor cells, which can be expressed as:

$$Q_{\text{PSN}} = \sum_{k=0}^{N_{\text{aggr\_cell}}} WSA_k \quad (2)$$

If IR-drop analysis is available using commercial rail analysis tool, average IR-drop of the path can be used instead of the *WSA* metric and acts as a more accurate quality value of the pattern. In this case, the quality value of power supply noise can be expressed as:

$$Q_{\text{PSN}} = \sum_{j=0}^{N_{\text{cell\_crit}}} IR_j / N_{\text{cell\_crit}} \quad (3)$$

where $N_{\text{cell\_crit}}$ is the total number of cells on the critical path and IR_j is the IR-drop value of cell j on the path. In the next section, we will present results from both metrics, i.e., average *IR-drop* and *WSA*.

In a similar way, we calculate Q for crosstalk noise as:

$$Q_{\text{Xtalk}} = \sum_{i=0}^{N_{\text{aggr\_net}}} d_i \cdot C_i \cdot f(\Delta t), \text{ where}$$

$$d_i = \begin{cases} 1, \text{ Opposite transition} \\ 0, \text{ No transition} \\ -1, \text{ Same transition} \end{cases} \quad (4)$$

$$f(\Delta t) = \begin{cases} 1, \ t_1 < \Delta t < t_2 \text{ and } \Delta t = t_a - t_v \\ 0, \text{ otherwise} \end{cases}$$

For all $N_{\text{aggr\_net}}$ aggressor nets the equation considers the direction of the transition with respect to associated nets of the targeted critical path, d, the amount of coupling between the two nets, C_i, and a timing window $f(\Delta t)$ to take into account the arrival time difference (Δt) of the transitions on aggressors (t_a) and victims (t_v). We employ a rectangular timing window as shown in Fig. 7 during Q_{Xtalk} calculation. This will simplify the calculation procedure and has shown to be effective from analyzing the results we have collected. The size of the timing window $[t_1,t_2]$ (as shown in Fig. 7) varies for different technologies. In this way, if the aggressor net's signal arrival time is within the window, it will be fully taken into account to measure its impact on victim

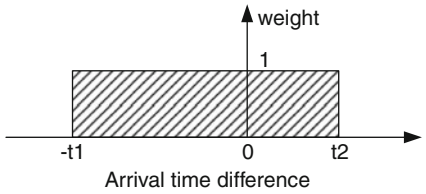

Fig. 7 Timing window for calculating the Q for crosstalk considering the arrival time difference of victim and aggressor nets

net delay. If it is outside the window, it will not have any impact on victim net delay. In other words, only the aggressor signals arriving within this window are considered as active aggressor. This will simplify our crosstalk Q calculation procedure and has shown to be effective from analyzing the results we have collected. $t1$ and $t2$ are calculated using the simulation results shown in Fig. 4a and b. A positive value of Q_{Xtalk} indicates that the targeted path will experience slow-down due to induced crosstalk effects from the switching aggressors. Similarly, a negative Q_{Xtalk} indicates that the targeted path will experience speed-up.

Step 3 **Final Q Calculation:** The total Q can be calculated using Eq. 5.

$$Q = W_P * Q_{\text{PSN}}^{\text{nom}} + W_X * Q_{\text{Xtalk}}^{\text{nom}} \tag{5}$$

where $Q_{\text{PSN}}^{\text{nom}}$ and $Q_{\text{Xtalk}}^{\text{nom}}$ are normalized pattern quality values for power supply noise and crosstalk. Weights W_P and W_X are used to account for the different impacts that power supply noise and crosstalk have on path delay. Different technologies may have different combinations of weights. This will be further elaborated in the following subsection.

3.2 Applications

There are two main applications for the pattern quality value Q: (1) to select the best pattern for PDF test; (2) to grade a pattern in terms of noise-induced delay for diagnostic purpose.

To perform pattern selection, we rank the patterns according to their *Final* Q values and then select the one with the largest Q value as the best pattern. W_P:W_X is required to calculate *Final* Q using Eq. 5, which reflects the relative impact of power supply noise and crosstalk on path delay. Empirical W_P:W_X value is suggested to be used for each technology. If no such value exists, a learning process is needed to generate the W_P:W_X values. A small number of fast-SPICE simulations can be done on small circuits to simulate the path delay for each pattern, and after calculating the pattern quality value Q, we could try different W_P:W_X ratios to calculate the correlation coefficients between Q and path delay and pick the one with best correlation. We have performed such analysis for s38417 benchmark designed in 180 nm technology and W_P:$W_X = 8$ provided the best correlation result between the *Final* Q values and the path delay simulated by fast-SPICE. This step needs to be done only once for each technology node.

Figure 8 illustrates the distribution curve for path delays and the corresponding Q values for a large number of patterns. This conceptual distribution shape is deduced from the SPICE simulation results in Fig. 1a. The lower bound of the noise distribution is given by a low-switching pattern. A 0-filled path delay fault pattern can be used as the lower bound pattern *P1*. As for the upper bound pattern *P2*, the pattern is either selected by the proposed flow or generated using a future pattern generation method.

For a single pattern, by comparing its Q value with the lower bound and upper bound (if available), we obtain an estimate about how effective the pattern is in activating power supply noise and crosstalk to increase the path delay. The higher the Q value of a pattern, the better the pattern is in exciting the noise.

The proposed method can also be used for diagnostic purpose to assist in identifying the root cause during failure analysis. For a pattern failed on tester, pattern quality metrics $Q_{\text{PSN}}^{\text{nom}}$ and $Q_{\text{Xtalk}}^{\text{nom}}$ can be calculated for the failed pattern over the suspect paths. These Q values can be compared with their upper bound and lower bound to estimate the noise levels. On the other side, when comparing these Q values with that of the functional patterns, we can determine whether the failure is caused by excessive pattern-induced noise or low noise immunity of the chip under test. If the failure is

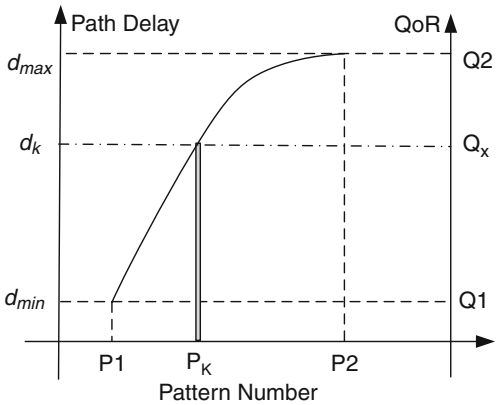

Fig. 8 Path delay and Q density distribution. Pattern P1 and P2 provide the lower (d_{min}) and upper (d_{max}) bounds

caused by low noise immunity of the design, the pattern evaluation results can help locate the weak point/area in the design. We can calculate the relative noise-induced delay using weighted pattern quality value $W_P * Q_{\mathrm{PSN}}^{\mathrm{nom}}$ and $W_X * Q_{\mathrm{Xtalk}}^{\mathrm{nom}}$ to evaluate each noise's impact on path delay. If excessive IR-drop or crosstalk is the failure reason, the victim gates' IR-drop or nets' coupling report generated by the VPI routine can be used to further locate the problematic area on the layout. The proposed pattern grading method has been used as a comparative and diagnostic tool in the flow we proposed in [13]. The simulation results presented in it demonstrate the usefulness of pattern quality value Q in diagnostic application.

4 Experimental Results

We implemented the pattern grading and pattern selection procedure on ISCAS benchmark *s38417*. The physical layout was designed using the 180 nm Cadence Generic Standard Cell Library with 1.8 V as its typical supply voltage. The program for parsing the DEF file to extract the victim and aggressor cells/nets list was developed in *C* language. In this work critical paths are selected using static timing analysis (STA) tools based on the extracted parasitic information. However, statistical static timing analysis (SSTA) [20, 21] considering process variations can also be used to select critical paths. To validate and analyze the accuracy and complexity of our proposed method, we perform our method over a number of patterns and compare its pattern grading results with that from the full-circuit fast-SPICE simulation.

First, we generated an unfilled PDF pattern for a target critical path using a commercial ATPG tool. Launch-off-Capture (LOC) method is used to generate PDF patterns. Since the second vector is a functional response of the first vector, there is a good chance that our pattern is functionally valid [11, 24]. We assume that DFT engineers generate PDF pattern using commercial ATPG tools and fill the don't-care bits in different ways. However, in this work to create a pattern set with random switching activity to induce noise effects at different levels, we fill the pattern in the following way. We use our program to fill the don't-care bits 950 times randomly with bit '0' probabilities increasing by 5% every 50 patterns from 5 to 95% to generate 950 PDF patterns for the path under test. For example, patterns 100 to 150 are filled with around 15% bits as '0' and the rest bits as '1'. Filling don't-care bit this way would provide a complete list of patterns with very high, medium, and very low switching activity in the circuit. It helps identify $d_{\min}$ and $d_{\max}$ (as shown in Fig. 8) for our target path in our experiment.

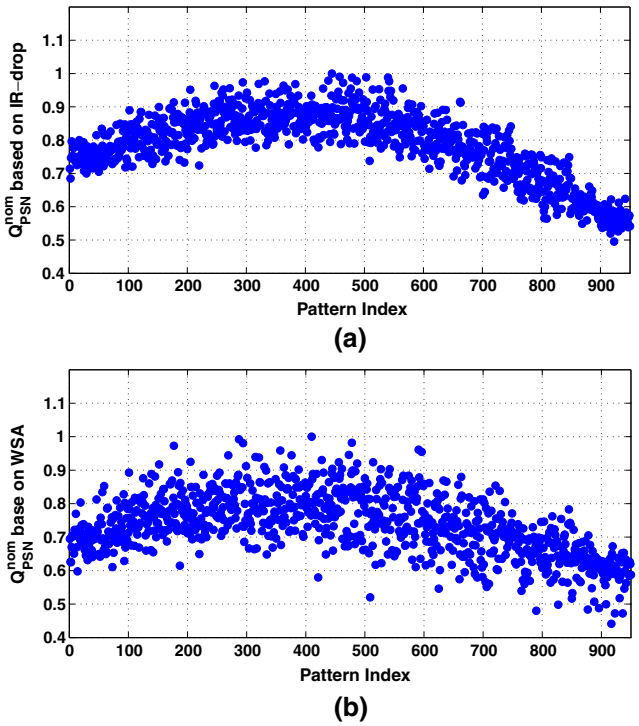

Fig. 9 $Q_{\mathrm{PSN}}^{\mathrm{nom}}$ based on (**a**) path average IR-drop; (**b**) *WSA* values

A full-circuit SPICE netlist is extracted using commercial extraction tool for the layout design of benchmark circuit s38417. Using this SPICE netlist and with test vectors as stimuli, transistor level fast-SPICE simulations are performed to verify the accuracy of the proposed pattern grading and selection procedure. *Note that the 950 pattern-filling method and the fast-SPICE simulation process are performed only for validation purpose in this paper. They are not required in real application.*

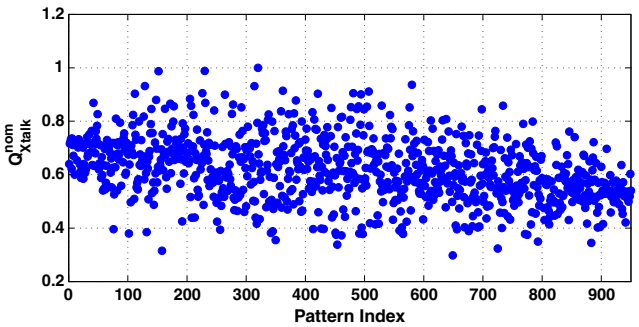

Fig. 10 $Q_{\mathrm{Xtalk}}^{\mathrm{nom}}$ for our pattern set

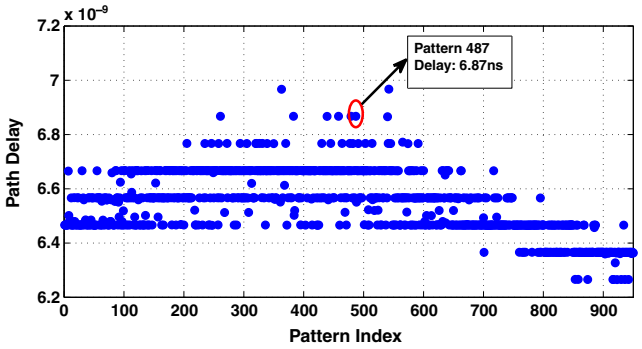

Fig. 11 Path delay based on fast-SPICE simulation for 950 random-filled patterns (used only for validation)

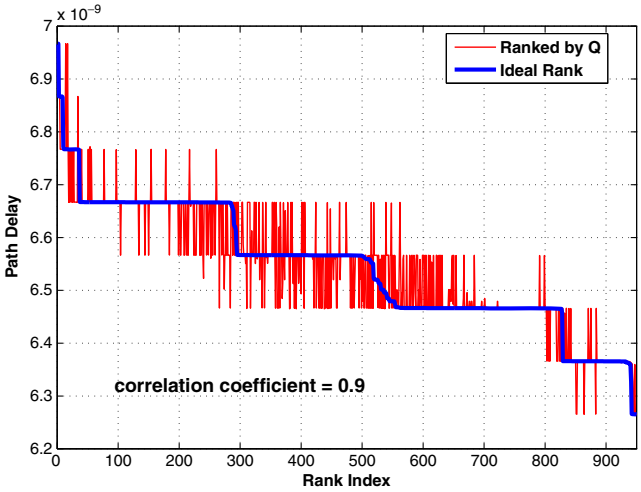

Fig. 12 Comparison of path delay of patterns ranked by Q values (*IR-Xtalk* method) with ideal (*Golden*) rank

In Section 3, we proposed two ways to calculate the pattern quality value Q_{PSN} for power supply noise, using *WSA* values and through IR-drop analysis as shown in Eqs. 2 and 3, respectively. Figure 9 presents the pattern quality values $Q_{\text{PSN}}^{\text{nom}}$ for both metrics which have been normalized to their largest quality value, respectively. The x-axis represents the pattern index which also implies increasing bit '0' probability for patterns from 0 to 950. From both results, we can see the $Q_{\text{PSN}}^{\text{nom}}$ values have smaller values at two ends and larger values in the middle. As expected, patterns filled with "near" 50% bit 0/1 probability cause more switching activity thus more power supply noise (although this may not always be the case). The pattern quality values $Q_{\text{Xtalk}}^{\text{nom}}$ for crosstalk effects normalized to the largest value are shown in Fig. 10. Compared with the $Q_{\text{PSN}}^{\text{nom}}$ values, the $Q_{\text{Xtalk}}^{\text{nom}}$ values are more random, which is expected since crosstalk effect depends on many parameters such as transition direction, transition arrival time and coupling capacitance size.

To analyze the accuracy of the proposed method, we ran full-circuit fast-SPICE simulation and show the path delay results in Fig. 11. The pattern index in x-axis is same as that in Figs. 9 and 10. As can be seen from the figure, the path delay values are mainly distributed on discrete numbers due to the limited fast-SPICE simulation resolution. The ranking of patterns based on fast-SPICE simulation results are treated as "golden" values in the following analysis.

Here we implement two methods: (1) *IR-Xtalk*: using IR-drop based $Q_{\text{PSN}}^{\text{nom}}$ metric for PSN and $Q_{\text{Xtalk}}^{\text{nom}}$ for crosstalk noise; and (2) *WSA-Xtalk*: using *WSA* based $Q_{\text{PSN}}^{\text{nom}}$ metric for PSN and $Q_{\text{Xtalk}}^{\text{nom}}$ for crosstalk noise. According to Eq. 5 and using $W_P : W_X = 8$, we calculate the final Q for each pattern and sort the 950 patterns based on their Q values. Table 1 summarizes the best and worst pattern reported by both methods and compares their path delays with the golden ones obtained from fast-SPICE. For example, the *IR-Xtalk* reports pattern 487 as the best pattern which has a 6.87 *ns* path delay (as circled in Fig. 11) and pattern 923 as the worst one. Only 2 patterns have path delay longer than pattern 487 and the error with respect to the golden one is 1.4%. The worst pattern hits the golden worst. To verify the effectiveness of our pattern quality metric in terms of representing the path delay caused by power supply and crosstalk noises, we calculate the correlation coefficient between *Final Q* values and path delays of these 950 patterns for each method. As can be seen from Table 1, the *IR-Xtalk* method shows a higher correlation value and provides a best pattern with longer path delay. This is because IR-drop is a more accurate model for power supply noise than the *WSA* values. For this reason, if IR-drop analysis is available we recommend using the *IR-Xtalk* method;

Table 1 Analysis of *IR-Xtalk*/*WSA-Xtalk* methods for *s38417* benchmark

Method	Best pattern			Worst pattern			Corr. coef.
	Pattern index	Path delay	Error (%)	Pattern idex	Path delay	Error (%)	
Golden	542	6.97*ns*	N/A	923	6.26*ns*	N/A	N/A
IR-Xtalk	487	6.87*ns*	1.4	923	6.26*ns*	0.0	0.9
WSA-Xtalk	294	6.77*ns*	2.8	917	6.26*ns*	0.0	0.7

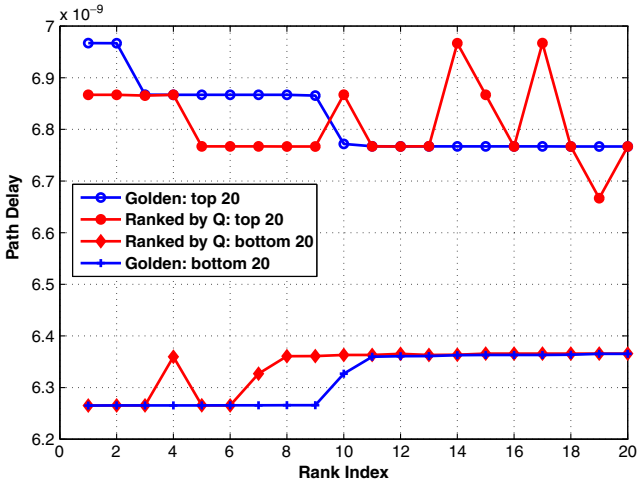

Fig. 13 Comparison of path delay of top 20 and bottom 20 patterns ranked by Q values (*IR-Xtalk* method) with that from *Golden* rank

while if only a rough estimation provides enough accuracy and speed is preferred, the *WSA-Xtalk* method provides a satisfactory solution. The performance of the *WSA-Xtalk* method could be improved if we adopt a more elaborate *WSA* model in the future.

Figure 12 presents the whole view of path delay comparison between patterns ranked by *IR-Xtalk* method and the *Golden* rank. The blue line shows the path delay of patterns sorted according to their path delays calculated by fast-SPICE simulation, i.e., *Golden* rank. The red line plots the path delay for the patterns sorted according to their the Q values calculated by *IR-Xtalk* method. Due to the limited resolution of the fast-SPICE simulation on path delay, the path delay of patterns in the center part of the plot falls into discrete numbers. Still, the results show pretty good matching, especially at the two ends. The correlation coefficient between these two sets of path delays is calculated as 0.9, which shows good correlation between the *IR-Xtalk* method with fast-SPICE simulation.

Figure 13 magnifies the details for both ends of the plot in Fig. 12. It demonstrates the path delay of 20 best patterns and 20 worst patterns selected by *IR-Xtalk* and fast-SPICE method, respectively, to test the critical path considering joint power supply noise and crosstalk effects. When comparing the path delay of patterns selected by the proposed *IR-Xtalk* method with the golden rank (those selected by fast-SPICE), the average error for the 20 best patterns is 1.02% and that for the worst 20 patterns is 0.32%. From this figure we can see, the patterns selected by the proposed pattern grading method is very close to the ones selected by fast-SPICE method.

As can be seen from Fig. 11, the most efficient pattern is usually found among patterns filled with bit '0' probability around 50%. Filling a lower number of patterns with bit '0' probability varying from 40 to 60% would be sufficient and faster to select the best pattern. Note that it does not mean 50% random-fill would *always* generate the most efficient pattern, because the path delay for those patterns usually varies in a large range and some ones have much shorter path delay than others. Based on this assumption, we fill PDF patterns 100 times with bit '0' probability increasing from 40 to 60% to generate 100 patterns and perform our *IR-Xtalk* pattern grading method to find the best pattern for 7 paths to demonstrate the consistency of our method. The path delay results for the *IR-Xtalk* method and the golden results are presented in Table 2. The number of victims/aggressors for power supply and crosstalk noise are shown in Columns 2–5. For all 7 paths, our *IR-Xtalk* method can select the best pattern with path delay difference less than 3% compared to the golden results according to fast-SPICE simulation. These results verify the effectiveness and accuracy of our proposed method.

Note that for the pattern evaluation, the proposed VPI procedure is run with parallel mode gate-level simulation. Since parallel simulation initializes all the scan cells in one cycle, it eliminates the time-consuming

Table 2 Path delay analysis for pattern grading method and fast-SPICE simulation results in *s38417* benchmark

Path	PSN		Crosstalk		Path delay		
No.	# of victim cells	# of aggr. cells	# of victim nets	# of aggr. nets	*IR-Xtalk* (*ns*)	Golden (*ns*)	Error (%)
1	25	1,575	25	387	6.666	6.767	1.49
2	25	1,546	25	392	5.803	5.945	2.40
3	25	2,062	25	366	5.515	5.515	0.00
4	25	2,026	25	363	5.314	5.316	0.04
5	25	2,029	25	360	5.212	5.314	1.92
6	17	839	17	298	3.720	3.723	0.08
7	24	1,420	24	383	3.471	3.473	0.06

Table 3 CPU Run-Time Comparison for *s38417* benchmark

Number of patterns	Our method		Fast-SPICE simulation
	WSA-Xtalk	*IR-Xtalk*	
1	0.43 s	0.8 s	18 min
100	50 s	60 s	29.5 hr
1000	13 min	20 min	≃300 hr

shift mode simulation process. This makes the proposed method practical and affordable for even large industry designs. Besides, large industry designs usually employs hierarchical testing configurations. A small number of blocks are tested once at a time. Those blocks that are not related to the paths under test are usually black-boxed during gate-level simulation, which further reduces the VCS-VPI simulation time. In addition, the proposed flow is intended to be performed before chip tape-out. Considering the long failure analysis and diagnosis time that is needed for potential noise-induced failure in post-silicon, running the pattern selection and pattern grading before chip tape-out is more cost-effective.

The CPU run-time comparison between our pattern grading method and selecting patterns by running the fast-SPICE simulation is shown in Table 3. The simulations were performed on an x86 server architecture, running a Linux OS, 8 CPU cores clocked at 2.826 GHz, and 32GB of RAM. The time for *WSA-Xtalk* represents the time consumed for running VPI procedure in parallel VCS simulation for both crosstalk and power supply noise evaluation. From this result, we can see that the proposed method has great advantage, which is around 1000 times faster than fast-SPICE simulation in terms of CPU run-time.

5 Conclusion

In this paper, we proposed a novel pattern grading and pattern selection procedure to quickly and efficiently evaluate the patterns in terms of their ability to increase path delay. The procedure is based on a novel quality metric for power supply and crosstalk noises. By evaluating the patterns with a quality metric to represent the impact of the active aggressors on path delay, the proposed procedure characterizes the patterns very quickly. The simulation results demonstrate (1) the effectiveness of the pattern evaluation and pattern selection process and (2) significant reduction (about 1000X) on CPU run-time.

References

1. Ahmed N, Tehranipoor M, Jayaram V (2007) Supply voltage noise aware ATPG for transition delay faults. In: Proc. VLSI test symposium (VTS'07), pp 179–186
2. Belete D, Razdan A, Schwarz W, Raina R, Hawkins C, Morehead J (2002) Use of DFT techniques in speed grading a 1GHz+ microprocessor. In: Proc. of int. test conf. (ITC'02), pp 1111–1119
3. Chen W, Gupta S, Breuer M (1999) Test generation for crosstalk-induced delay in integrated circuits. In: Proc. int. test conf. (ITC'99), pp 191–200
4. Chen W, Gupta S, Breuer M (2000) Test generation for crosstalk-induced faults: framework and computational results. In: Proc. Asian test conf. (ATS'00), pp 305–310
5. Chen W, Gupta SK, Breuer MA (2002) Analytical models for crosstalk excitation and propagation in VLSI circuits. In: IEEE Trans CAD 21(10):1117–1131
6. Cuviello M, Dey S, Bai X, Zhao Y (1999) Fault modeling and simulation for crosstalk in system-on-chip interconnects. In: Proc. int. conf. computer-aided design (ICCD'99), pp 297–303
7. Krstic A, Jiang YM, Cheng KT (2001) pattern generation for delay testing and dynamic timing analysis considering power-supply noise effects. IEEE Trans CAD 20(3):416–425
8. Krstic A, Liou J, Jiang Y, Cheng KT (2001) Delay testing considering crosstalk induced effects. In: Proc. int. test conf. (ITC'01), pp 558–567
9. Lee J, Tehranipoor M (2008) A novel pattern generation framework for inducing maximum crosstalk effects on delay-sentive paths. In: Proc. int. test conf. (ITC'08)
10. Lee J, Narayan S, Tehranipoor M (2008) Layout-aware, IR-drop tolerant transition fault pattern generation. In: Proc. design, automation, and test in Europe (DATE'08), pp 1172–1177
11. Liu X, Hsiao MS (2004) On identifying functionally untestable transition faults. In: IEEE intl. high-level design validation and test workshop, pp 121–126
12. Ma J, Lee J, Tehranipoor M (2009) Layout-aware pattern generation for maximizing supply noise effects on critical paths. In: Proc. IEEE VLSI test symposium (VTS'09), pp 221–226
13. Ma J, Ahmed N, Tehranipoor M (2011) Low-cost diagnostic pattern generation and evaluation procedures for noise-related failures. In: IEEE VLSI test symposium (VTS), pp 309–314
14. Peng K, Thibodeau J, Yilmaz M, Chakrabarty K, Tehranipoor M (2010) A novel hybrid method for SDD pattern grading and selection. In: Proc. IEEE VLSI test symposium (VTS'10), pp 45–50
15. Sinha A, Gupta SK, Breuer MA (2002) Validation and test issues related to noise induced by parasitic inductances of VLSI interconnects. IEEE Trans Adv Packaging 25(3):329–339
16. Sinha A, Gupta S, Breuer M (2003) An enchanced test generator for capacitance induced crosstalk delay faults. In: Proc. Asian test conf. (ATS'03), pp 174–177
17. Synopsys Inc. (2005) User manuals for synopsys toolset version 2005.09. Synopsys Inc.
18. Tehranipour M, Ahmed N, Nourani M (2003) Multiple transition model and enhanced boundaryscan architecture to test interconnects for signal integrity. In: Proc. int. conf. computer design (ICCD'03)
19. Tehranipour M, Ahmed N, Nourani M (2003) Testing SoC interconnects for signal integrity using boundary scan.

In: proc. IEEE VLSI test symposium (VTS'03), Napa, CA, pp 158–163
20. Visweswariah C et al (2006) First-order incremental block-based statistical timing analysis. IEEE Trans CAD 25(10):2170–2180
21. Wang L-C, Liou J-J, Cheng K-T (2004) Critical path selection for delay fault testing based upon a statistical timing model. IEEE Trans CAD 23(11):1550–1565
22. Yilmaz M, Chakrabarty K, Tehranipoor M (2008) Test-pattern grading and pattern selection for small-delay defects. In: Proc. IEEE VLSI test symposium (VTS'08), pp 233–239
23. Zeng J, Abadir MS, Vandling G, Wang L-C, Karako S, Abraham JA (2004) On correlating structural tests with functional tests for speed binning of high performance design. In: Proc. of int. test conf. (ITC'04), pp 31–37
24. Zhang Z, Reddy SM, Pomeranz I (2005) On generating pseudo-functional delay fault tests for scan designs. In: IEEE intl. symposium on defect and fault tolerance in VLSI systems, pp 398–405
25. Zhao S, Roy K (2000) Estimation of switching noise on power supply lines in deep sub-micro CMOS circuits. In: Proc. thirteenth int. conf. on VLSI design, pp 168–173

Junxia Ma received the Ph.D. degree in Electrical and Computer Engineering from University of Connecticut, Storrs, CT in 2010 with research topic on layout-aware delay fault testing techniques considering signal and power integrity issues. She received the M.E. degree in Electrical Engineering from University of Macau, Macao, in 2006 focusing on high speed Analog-to-Digital Converter (ADC) design and the B.Sc. Degree in Electrical Engineering from Beihang University, Beijing, China in 2003. She was a member of technical staff in Cadence Design Systems, Inc., Beijing, from 2006 to 2007. She currently works in LSI Corporation in Milpitas, California, as a senior methodology design engineer in test methodology group. Her research interests include: delay test, low-power test, power and signal integrity issues in nanometer VLSI testing.

Dr. Ma was selected as a finalist for Connecticut "Women of Innovation" Award in 2011 by Connecticut Technology Council. Her presentation on "Layout-Aware Test Pattern Generation to Maximize Power Supply Noise" received the "Best in Session" Award in Semiconductor Research Corporation (SRC) TECHCON 2009, Austin, TX.

Mohammad Tehranipoor is currently an Associate Professor of Electrical and Computer Engineering at the University of Connecticut. His current research projects include computer-aided design and test, reliability analysis, and hardware security and Trust. Dr. Tehranipoor has published over 150 journal articles and refereed conference papers, four books, and seven book chapters. He is a recipient of a best paper award at the 2005 VLSI Test Symposium (VTS), best paper award at the 2008 North Atlantic Test Workshop (NATW), best paper award at NATW-2009, honorable mention for best paper award at NATW-2008, best paper candidate at the 2006 Design Automation Conference (DAC), and best panel award at VTS-2006. Dr. Tehranipoor is also a recipient of the 2008 IEEE Computer Society Meritorious Service Award, the 2009 NSF CAREER award, and the 2009 UConn ECE Research excellence award.

He serves on the program committee of several leading conferences and workshops. He served as Program Chair of the 2007 IEEE Defect-Based Testing (DBT) workshop, Program Chair of the 2008 IEEE Defect and Data Driven Testing (D3T), Co-program Chair of the 2008 International Defect and Fault Tolerance in VLSI Systems (DFTS), and General Chair for D3T-2009 and DFT-2009. He co-founded a new workshop called IEEE International Workshop on Hardware-Oriented Security and Trust (HOST) and served as HOST-2008 and HOST-2009 General Chair as well as HOST's Chair of Steering Committee. He is currently serving as an Associate Editor for JETTA, IEEE Design and Test of Computers, Journal of Low Power Electronics. Dr. Tehranipoor is a Senior Member of the IEEE and Member of ACM and ACM SIGDA.

Patrick Girard received a M.S. degree in Electrical Engineering and a Ph.D. degree in Microelectronics from the University of Montpellier, France, in 1988 and 1992 respectively. He is currently Research Director at CNRS (French National Center for Scientific Research), and Chair of the Microelectronics Department of the Laboratory of Informatics, Robotics and Microelectronics of Montpellier (LIRMM) - France. His research interests include all aspects of digital testing and memory testing, with emphasis on critical constraints such as timing and power. Reliability and fault tolerance, as well as design and test of 3D ICs are also part of his new research activities.

He is Technical Activities Chair of the Test Technology Technical Council (TTTC) of the IEEE Computer Society. From 2006 to 2010, he was Vice-Chair of the European TTTC (ETTTC) of the IEEE Computer Society. He has served on numerous conference committees and is the founder and Editor-in-Chief of the ASP Journal of Low Power Electronics (JOLPE). He is also an Associate Editor of the IEEE Transactions on VLSI Systems and the Journal of Electronic Testing - Theory and Applications (JETTA - Springer).

He has supervised 29 PhD dissertations and has published 6 books or book chapters, 38 journal papers, and more than 150 conference and symposium papers on these fields. Patrick Girard is a Golden Core Member of the IEEE Computer Society.