PAPER Special Section on VLSI Design and CAD Algorithms

Crosstalk Noise Optimization by Post-Layout Transistor Sizing

Masanori HASHIMOTO[†a)] ***and*** **Hidetoshi ONODERA**[†*], ***Members***

SUMMARY This paper proposes a post-layout transistor sizing method for crosstalk noise reduction. The proposed method downsizes the drivers of aggressor wires for noise reduction, utilizing the precise interconnect information extracted from the detail-routed layouts. We develop a transistor sizing algorithm for crosstalk noise reduction under delay constraints, and construct a crosstalk noise optimization method utilizing an analytic crosstalk noise model and a transistor sizing framework that have been developed. Our method exploits the transistor sizing framework that can vary transistor widths inside cells with interconnects unchanged. Our optimization method therefore never causes a new crosstalk noise problem, and does not need iterative layout optimization. The effectiveness of the proposed method is experimentally examined using 2 circuits. The maximum noise voltage is reduced by more than 50% without delay violation. These results show that the risk of crosstalk noise problems can be considerably reduced after detail-routing.

key words: *crosstalk noise, capacitive coupling noise, transistor sizing, gate sizing, post-layout optimization*

1. Introduction

Crosstalk noise problem heavily depends on interconnect structure, i.e. coupling length, spacing between adjacent wires, and coupling position, and hence many techniques of routing and interconnect optimization for crosstalk noise reduction are proposed [1]–[3]. Buffer insertion is also effective for noise reduction, and some methods are proposed [4], [5]. References [6]–[8] discuss the effectiveness of transistor sizing for crosstalk noise reduction, but practical implementations are not shown. Recently, Refs. [9]–[11] propose transistor sizing methods for crosstalk noise reduction. Reference [9] expresses the influence of crosstalk noise as the amount of coupling capacitance, and optimizes noise as well as area, delay and power by gate and wire sizing. Reference [10] proposes a driver sizing algorithm for noise reduction using a crosstalk noise estimation tool [12]. Reference [11] estimates crosstalk noise by Ref. [8], and circuit area is minimized under delay and crosstalk noise constraints. The authors do not mention layout modification after optimization. When optimization results are applied to layout, a certain number of interconnects are changed, which may spoil the optimization result, or may cause a new crosstalk noise problem. More recently, a gate sizing method to reduce crosstalk induced delay is proposed [13]. This method is based on a crosstalk noise aware static timing analysis. Although this method changes circuits considerably, layout modifications are not taken into consideration. Reference [14] can not solve the problem that iterative gate sizing does not necessarily converge due to layout modification, either.

This paper proposes a post-layout transistor sizing method for crosstalk noise reduction. The proposed method optimizes detail-routed circuits and reduces peak noise voltage as much as possible while preserving interconnects entirely. The interconnect information required for crosstalk noise estimation can be accurately obtained after detail-routing. The optimization result of transistor sizing can be completely applied to the layout because the proposed method utilizes the transistor sizing framework that can downsize the transistors inside cells preserving interconnects [15], [16]. This framework is originally developed for power reduction. In this paper, we use this framework for crosstalk noise reduction. In this framework, various driving-strength cells are generated on the fly according to the optimization result, and hence transistor-level optimization can be executed in cell-base design. Cell layouts are generated by a layout generation system called VARDS [17], [18]. VARDS is based on a symbolic layout method and is enhanced to produce cell layouts with variable driving strength. Exploiting this framework, the proposed method reduces crosstalk noise efficiently after detail-routing. As for crosstalk noise estimation, our method utilizes a 2-π noise model with improved aggressor modeling [19]. This model can consider the location of coupling, the effect of distributed RC networks, and the slew of input signal. Reference [19] also mentions a transformation method that can apply all types of RC trees to the 2-π noise model, which enables crosstalk noise optimization of practical circuits. In this paper, we develop an optimization algorithm for crosstalk noise reduction that explores solution space effectively under delay constraints, and construct a crosstalk noise reduction method using the noise estimation method [19] and the transistor sizing framework [15], [16]. Thanks to the framework, the estimation method and the algorithm, our method can estimate and optimize crosstalk noise with interconnects unchanged. The proposed method can be applied to the circuits optimized by other methods, such as interconnect optimization and buffer insertion, and it further reduces the risk of crosstalk noise.

Upsizing and downsizing drivers have different mer-

Manuscript received March 18, 2004.
Manuscript revised June 19, 2004.
Final manuscript received August 9, 2004.

[†]The authors are with the Department of Communications and Computer Engineering, Kyoto University, Kyoto-shi, 606-8501 Japan.
[*]Presently, with the Department of Information Systems Engineering, Osaka University.
a) E-mail: hasimoto@ist.osaka-u.ac.jp

its and demerits although both approaches can optimize crosstalk noise. The upsizing approach does not degrade the robustness against random manufacturing variation but increases power dissipation and temperature, where it is known that random manufacturing variation becomes smaller as transistor size becomes larger. On the other hand, our approach decreases power dissipation and temperature, although it may degrade robustness against random manufacturing variability.

This paper is organized as follows. Section 2 explains an estimation method of crosstalk noise. Section 3 shows an optimization algorithm for crosstalk noise reduction. Section 4 demonstrates some experimental results. Finally, Sect. 5 concludes the discussion.

2. Crosstalk Noise Estimation

This section discusses crosstalk noise estimation. The proposed method utilizes the 2-π noise model for crosstalk estimation [19], and we explain it briefly. We next discuss a noise estimation method for a net with multiple aggressors based on superposition considering timing window.

2.1 Overview of Crosstalk Noise Estimation

In practical circuits, many interconnects couple with multiple interconnects, i.e. with multiple aggressors. We estimate the peak noise voltage caused by each aggressor respectively, and calculates the maximum noise voltage at the sink by superposition. The superposition of the noise voltage is discussed in Sect. 2.2.

The victim net with one aggressor is represented as two partially-coupled interconnects (Fig. 1). The partially-coupled interconnects in Fig. 1 are modeled as an equivalent circuit shown in Fig. 2. R_{v1} is the effective driver resistance of the victim net. The node n_{v2} corresponds to the center of the coupling portion of the victim interconnect, and n_{a2} similarly corresponds to the center of the coupling portion of the aggressor interconnect. R_{v2} is the resistance between the source and n_{v2}, and R_{v3} is the resistance between n_{v2} and the sink. C_c is the coupling capacitance between the victim and the aggressor. The capacitances C_{v1}, C_{v2} and C_{v3} are represented as $C_1/2$, $(C_1+C_2)/2$, and $C_2/2+C_l$ respectively, where C_1 is the wire capacitance from the source to n_{v2}, C_2 is the wire capacitance from n_{v2} to the sink, and C_l is the capacitance of the receiver. The parameters of the aggressor wire, R_{a1}, R_{a2}, R_{a3}, C_{a1}, C_{a2}, C_{a3}, are determined similarly. Reference [19] also develops a method that can apply interconnects with branches into the model circuit of Fig. 2. We here omit the explanation of this application method.

In the circuit of Fig. 2, the peak voltage V_{peak} is expressed as follows [19].

$$V_{peak} = \frac{(R_{v1}+R_{v2})C_cV_{dd}}{\tau_v}\left(\frac{\tau_v}{\tau_a}\right)^{-\frac{\tau_a}{\tau_v-\tau_a}}, \quad (1)$$

where

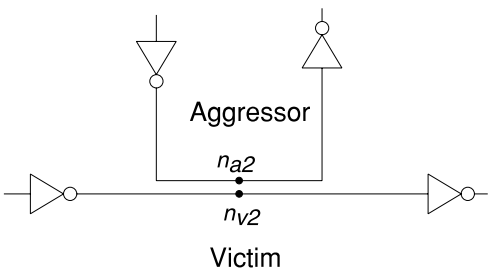

Fig. 1 Two coupled interconnects.

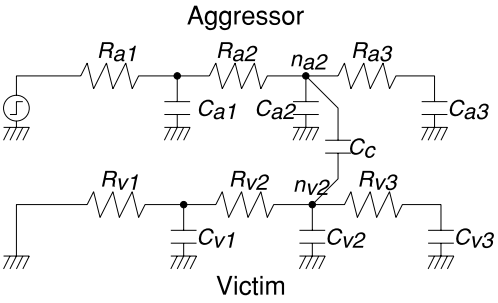

Fig. 2 An equivalent circuit of two partially-coupled interconnects for crosstalk estimation.

$$\tau_v = R_{v1}(C_{v1}+C_{v2}+C_c+C_{v3}) + R_{v2}(C_{v2}+C_c+C_{v3}) + R_{v3}C_{v3}, \quad (2)$$

$$\tau_a = R_{a1}(C_{a1}+C_{a2}+C_c+C_{a3eff}) + R_{a2}(C_{a2}+C_c+C_{a3eff}), \quad (3)$$

$$C_{a3eff} = C_{a3}\left(1-e^{-T/R_{a3}C_{a3}}\right), \quad (4)$$

$$T = R_{a1}(C_{a1}+C_{a2}+C_c+C_{a3}) + R_{a2}(C_{a2}+C_c+C_{a3}). \quad (5)$$

Peak noise V_{peak} in the case of $\tau_a = \tau_v$ is discussed in Appendix.

2.2 Noise Superposition Based on Timing Window Calculation

The proposed method evaluates the peak noise voltage caused by each aggressor separately, and calculates the maximum noise voltage at the sink by superposition. In linear systems, the principle of superposition holds. When the noise amplitude is not large, i.e. as long as CMOS gates can be treated as a linear resistance, the noise waveform at the sink of the victim can be represented as the superposition of the noise waveform from each aggressor wire. Reference [19] experimentally demonstrates that this assumption of noise superposition is reasonable in practical interconnects.

In the noise superposition, the relative timing of the transitions at the aggressor wires is an important factor. If we superpose two noises that never happen simultaneously, crosstalk noise is overestimated, and the estimated value is too pessimistic. In order to eliminate the overestimation, we calculate the timing window that is the timing range when

a transition may occur and is defined as the range between EAT_l and LAT_l. EAT_l is the earliest time of signal arrival at the output of cell l, and LAT_l is the latest arrival time. EAT_l and LAT_l are calculated as follows.

$$EAT_l = \min_{m \in FI(l)} \{EAT_m + d^{min}_{m,l}\}, \tag{6}$$

$$LAT_l = \max_{m \in FI(l)} \{LAT_m + d^{max}_{m,l}\}, \tag{7}$$

where $FI(l)$ is the set of the fan-in cells of cell l. $d^{min}_{m,l}$ represents the minimum delay between the output of cell m to the output of cell l in the case that the aggressors and the victim change in the same transition direction simultaneously. Similarly, $d^{max}_{m,l}$ is the maximum delay in the case that the transition direction of the victim is opposite to those of the aggressors.

The maximum noise voltage at the i-th sink of the victim net at time t is represented as follows.

$$V_{max,i}(t) = \sum_{j}^{n} k(t) \cdot V_{peak,j \to i}, \tag{8}$$

$$k(t) = \begin{cases} 1 & EAT_i \le t \le LAT_i \\ 0 & otherwise \end{cases} \tag{9}$$

where n is the number of the aggressors, and $V_{peak,j \to i}$ is the noise voltage at the i-th sink caused by the j-th aggressor. We sweep time t, and find the maximum noise voltage at each victim net.

We here use a simple method [20] for estimating the maximum delay $d^{max}_{m,l}$ and the minimum delay $d^{min}_{m,l}$. Other methods, such as Ref. [21], can be also used. When we have to estimate timing window more tightly considering the dependence of $d^{max}_{m,l}$ and $d^{min}_{m,l}$ on the timing window, we should calculate timing window iteratively [22]. In this paper, this iterative calculation is not executed simply because of an implementation matter. There are no technical limitations. Reference [20] indicates that the upper bound of $d^{max}_{m,l}$ can be estimated as follows; all coupling capacitances are converted into the 3X capacitances to the ground, and then cell delay and wire delay are calculated. As for $d^{min}_{m,l}$, coupling capacitances are replaced with the -1X capacitances to the ground. We utilize those upper and lower bound for $d^{max}_{m,l}$ and $d^{min}_{m,l}$ in this paper. Other sophisticated gate delay model [23], [24] also can be used, as long as the computational cost permits.

3. Optimization Algorithm

In this section, an optimization algorithm for crosstalk noise reduction is discussed. The proposed algorithm reduces crosstalk noise under delay and transition time constraints. First, an optimization algorithm for a localized problem that includes one victim net and its adjacent nets is explained. This section then shows the overall algorithm that builds and solves the local optimization problems, considering the global optimality under delay constraints.

3.1 Optimization Algorithm in Each Victim Net

First, the noise reduction algorithm for each victim net is explained. The proposed method downsizes the drivers of the adjacent aggressor wires in order to reduce the amount of crosstalk noise at the victim wire. When the driving strength of the aggressor wire becomes weak, i.e. the driver resistance R_{a1} becomes large, the time constant of the aggressor voltage source τ_a increases (Eq. (4)). Then the maximum noise voltage V_{peak} (Eq. (1)) at the victim net consequently decreases. This relationship can be revealed from the partial derivative of V_{peak} respect to R_{a1} as follows.

$$\frac{\partial V_{peak}}{\partial R_{a1}} = \frac{\tau_v (C_a + C_c)}{(\tau_v - \tau_a)^2} \left(\log \frac{\tau_a}{\tau_v} - \frac{\tau_a}{\tau_v} + 1 \right) \cdot V_{peak} \le 0. \tag{10}$$

In order to choose the driver of the aggressor wire to be downsized efficiently, a measure *priority* is devised.

$$priority_i = slack_i \cdot \sum_{j}^{m} V_{peak,i \to j}, \tag{11}$$

where $V_{peak,i \to j}$ is the peak noise voltage at the j-th sink caused by the i-th aggressor net, and m is the number of sinks. The value $slack_i$ represents the timing margin at the i-th aggressor net, and is defined as the time difference between the required time and the arrival time [25]. The measure $priority_i$ becomes large in the case that the i-th adjacent net causes a large amount of noise and the timing constraint at the i-th aggressor net is not tight. Using this measure, the proposed algorithm can find the aggressor net efficiently that has strong influence on the crosstalk noise at the victim net yet has little influence on the circuit delay.

One of the difficulties in crosstalk noise optimization is that each victim net also becomes an aggressor from the opposite standpoint. When the driver of an aggressor is downsized for reducing noise at the victim net, the noise at the aggressor may increase intolerably. We therefore calculate the peak noise voltages at both the victim and the aggressor wires, and find a proper driver size of the aggressor. For this purpose, we here minimize the sum of the squared noise voltage at the aggressor and the squared noise voltage at the victim.

Step 1: Calculate *priority* (Eq. (11)) for each adjacent aggressor net, and put all the aggressor nets into list L_l.

Step 2: Choose the aggressor net with the maximum *priority* from list L_l.

Step 3: Downsize the driver of the chosen aggressor net within the limit that the delay constraints and the transition time constraints are satisfied. The best size of the driver is decided such that the value of $(V_v^2 + V_a^2)$ becomes the smallest, where V_v is the noise voltage at the victim net, and V_a is the noise voltage at the aggressor net. In the practical implementation, we try several driver sizes and calculate the value of $(V_v^2 + V_a^2)$ and the

circuit delay. We then choose the best size from those sizes without delay violation. Remove the aggressor net from L_l.

Step 4: If the noise voltage becomes smaller than the target value V_{target}, or if the list L_l becomes empty, finish the optimization procedure. Otherwise go back to **Step 2**. The value V_{target} is explained in the following section.

3.2 Overall Optimization Algorithm

Section 3.1 discusses the optimization algorithm for the localized problem that contains one victim net and its adjacent aggressor nets. Next, the overall algorithm is discussed. This algorithm aims to reduce both the maximum noise voltage in a circuit and the number of nets whose noise is large.

The optimization iterates the following procedure from **Stage 1** to **Stage 4** for several times, as parameter *threshold* gradually decreases. The parameter *threshold* is used for selecting the nets to be optimized, and it ranges from 0 to 1. The nets whose noise voltages are larger than V_{target}, which is the product of *threshold* and the maximum noise voltage in the circuit V_{max}, are chosen as the optimization candidates. In the beginning, *threshold* is set close to 1 in order to reduce the maximum noise voltage intensively. In the end, *threshold* is set close to 0, and the most of the nets in the circuit are optimized.

Stage 1: Calculate the crosstalk noise at each net in the circuit.

Stage 2: Find the maximum voltage of crosstalk noise V_{max} in the circuit, and put the nets whose noise voltages are larger than $V_{target} = V_{max} \times threshold$ into the candidate list L_o.

Stage 3: Choose the net with the maximum noise voltage in the list L_o, and execute the optimization explained in Sect. 3.1. The value of V_{target} is given to the optimization as the target value. Remove the net from the list L_o, and update the information of timing window.

Stage 4: If the list L_o becomes empty, finish the optimization procedure. Otherwise go back to **Stage 3**.

When the timing constraints are given, the timing margin at each net should be utilized efficiently for reducing the crosstalk noise. Therefore the sequence of the nets to be optimized is critical and essential to obtain high-quality circuits. In order to reduce the maximum noise voltage, the proposed algorithm gives priority to the nets with large noise. **Stage 2** excludes the nets whose noise voltages are smaller than V_{target} from the the optimization candidates. In **Stage 3**, the nets are optimized in order of the amount of noise voltage.

In **Stage 3**, the target noise value V_{target} is given to the localized optimization problem, in order to control the local optimization from the viewpoint of global optimality. The optimization result that the noise voltage is minimized in the localized problem may incur a bad local-minimum solution globally. This is because the timing margins, which may be utilized for reducing the noise at other nets, are wasted. The proposed algorithm hence stops the local optimization when the noise voltage becomes smaller than the target value in Step 4. Thanks to the good sequence of the net to be optimized and setting the target noise value, the proposed method can reach a good solution under the delay constraints.

4. Experimental Results

This section shows the optimization results for crosstalk noise reduction. The circuits used for the experiments are an ALU in a DSP for mobile phone [26] (dsp_alu) and the circuits included LGSynth93 benchmark sets (des). These circuits are synthesized by a commercial logic synthesis tool. The circuit scale of dsp_alu is 12547 cells, and the number of cells in des is 3414. The placement and routing are performed by a commercial tool with an option that minimizes interconnect length and without any options for crosstalk noise reduction. The layout area of dsp_alu is 5.3 $(2.3 \times 2.3)\,\mathrm{mm}^2$, and the area of des is 0.64 $(0.8 \times 0.8)\,\mathrm{mm}^2$. RC trees of interconnects are extracted from the layouts by a quasi-3D RC extract tool [28]. The coupling capacitances below 10 fF are extracted as the capacitance to the ground, where the coupling capacitance of 10 fF corresponds to the length of 230 μm. The supply voltage is 3.3 V.

Cell layouts are generated using VARDS [17], [18] in a 0.35 μm process with three metal layers. The layout generation system VARDS can vary transistor widths in a cell while keeping the location of each pin. Exploiting this feature, the proposed method optimizes a detail-routed circuit without any wire modifications [15], [16]. The height of the generated cells is 13 interconnect-pitches. In transistor sizing, MOSFETs are downsized within the range that VARDS can generate cell layouts. The maximum transistor width of standard driving-strength ($\times$1) cells is 6.2 μm. The transistor width can be reduced to 0.9 μm. We characterize the resistance of a CMOS gate as 4 values; R_{Dp} and R_{Dn} are the driving resistances of the pull-up PMOS part and the pull-down NMOS part respectively, and R_{Hp} and R_{Hn} are the holding resistances. R_{Dp} and R_{Dn} are decided such that the propagation delay becomes the same with circuit simulation results [29]. R_{Hp} and R_{Hn} are evaluated by the operating condition analysis of circuit simulation.

The initial circuits used for the experiments are designed for minimizing the circuit delay in a usual cell-base design style. The circuits are optimized such that the circuit delay does not increase. The given constraint of the transition time is 1.0 ns and it is the same with the constraint given for the initial circuit design. The *threshold* value used in the optimization algorithm of Sect. 3.2 is varied from 0.98 at the beginning to 0.5 at the end of the optimization. The iteration number of the optimization of Sect. 3.2 is 6 for both circuits. In the current implementation, when we execute **Step 3** of the proposed algorithm explained in Sect. 3.1, we try five transistor widths between the current transistor width and the minimum transistor width.

Table 1 Noise optimization results.

Circuit	Maximum Noise(V)		CPU Time (s)	#Pairs†	# Cells
	Initial	Optimized			
des	0.40	0.19	12	1018	3414
dsp_alu	1.00	0.50	604	82368	12547

#Pairs†: number of pairs of an aggressor and a victim in a circuit.

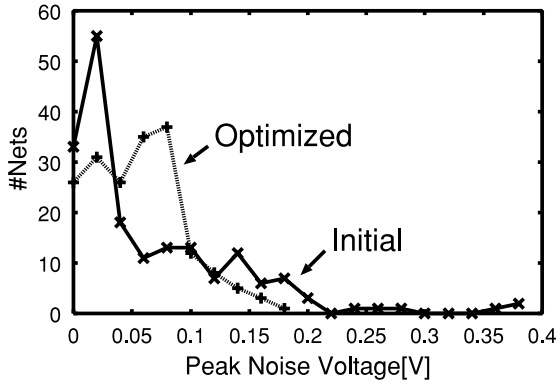

Fig. 3 Optimization results for crosstalk noise reduction (des).

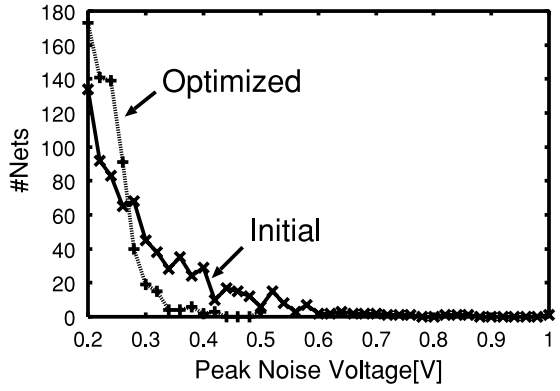

Fig. 4 Optimization results for crosstalk noise reduction (dsp_alu).

Table 1 demonstrates the crosstalk noise optimization results. Figures 3 and 4 show the distributions of the maximum noise voltage before and after the optimization. In des circuit, the maximum noise voltage is reduced from 0.40 V to 0.19 V by 53%. In dsp_alu circuit, the maximum noise is reduced from 1.00 V to 0.50 V by 50%. The distribution is also shifted in the direction that the noise voltage decreases. The number of nets whose noise voltages are over 0.5 V is decreased from 59 to 3. After the detailed-routing, the crosstalk noise can be reduced considerably by downsizing the transistors inside cells while keeping the interconnects without delay violation. The proposed method reduces crosstalk noise as much as possible after detail-routing, and reduces the risk of crosstalk noise problems at the final design stage. However when further noise reduction is necessary, we need to use other techniques in routing that can reduce coupling capacitance itself.

Figure 5 shows the slack distribution before and after the optimization in dsp_alu circuit. The proposed method

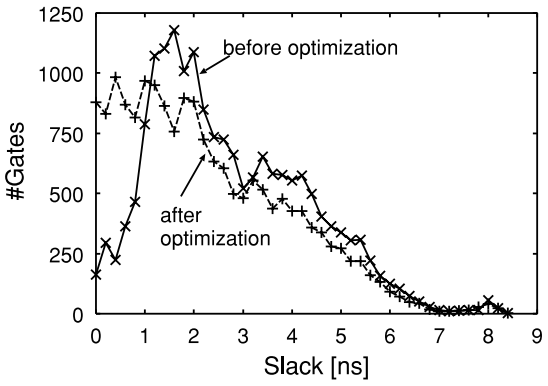

Fig. 5 Slack distributions before and after optimization (dsp_alu).

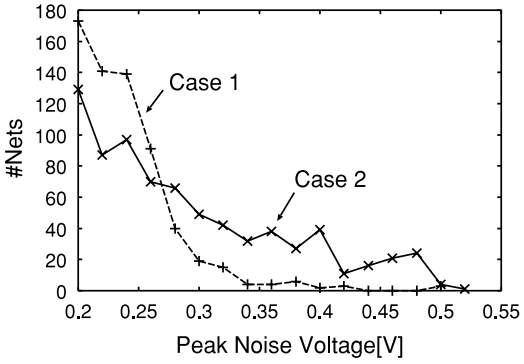

Fig. 6 Comparison of optimization results when iteration and *threshold* are changed (dsp_alu).

downsizes drivers whose timing constraint is not tight for crosstalk noise reduction, and hence the number of gates whose slack is small increases. Reference [30] indicates that increasing the number of gates whose slack is small may cause a delay violation when delay variation exists. If this problem is significant, we should use statistical static timing analysis for timing verification inside the optimization like Ref. [31].

The CPU times required for the optimization on an Alpha Station (600 MHz) are 12 seconds in des (3.4k cells), and 604 seconds in dsp_alu (13k cells). Here the CPU time for reading circuit and interconnect information files is excluded. The basic optimization of the proposed algorithm in Sect. 3.1 is executed for each pair of an aggressor and a victim, and hence the required computational time is basically proportional to the number of pairs of an aggressor and a victim, although it also depends on circuit structure, timing constraints and so on.

Figure 6 shows two peak noise distributions after the optimizations with different setup. In Case 1, the number of iterative optimization of Sect. 3.2 is 6 and *threshold* at each iteration is set as shown in Table 2. As for Case 2, the number of iteration is 1, and *threshold* is 0.5. In Case 2, the number of nets whose peak voltage is above 0.4 V is 116, whereas it is 8 in Case 1. The iteration with gradual decrease of *threshold* is important to obtain a good opti-

Table 2 Noise optimization progress at each iteration count.

Iteration Count	*threshold*	Maximum Noise (V)	
		des	dsp_alu
Initial	-	1.00	0.40
1	0.98	0.93	0.38
2	0.95	0.76	0.29
3	0.90	0.68	0.23
4	0.80	0.54	0.19
5	0.70	0.50	0.19
6	0.50	0.50	0.19

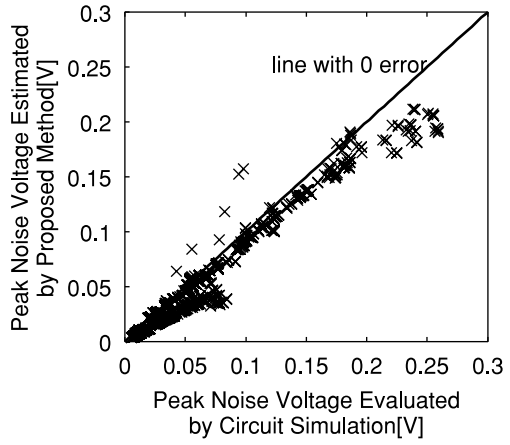

Fig. 7 Accuracy of peak noise estimation for each pair of an aggressor and a victim (des).

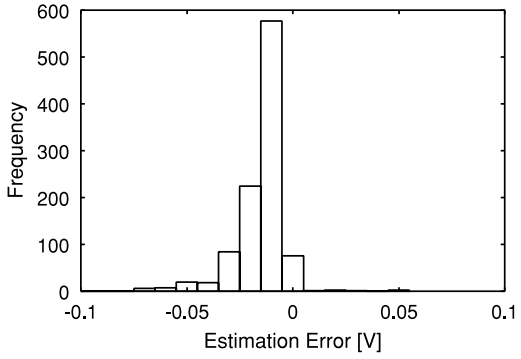

Fig. 8 Histogram of estimation error (des).

mization result. However, we experimentally observe that the values themselves of *threshold* in Table 2 are not so important, and the optimization results do not change so much even if we slightly change *threshold* values.

We finally demonstrate the accuracy of crosstalk noise estimation. The estimated noise voltage is compared with the circuit simulation results of actual circuits, i.e. interconnect with branches driven by CMOS gates. Figure 7 shows the accuracy of the peak noise estimation. Each dot corresponds to one victim net with one aggressor, i.e. this evaluation is before noise superposition. Figure 8 shows the histogram of the estimation error. The average error is 10 mV. The peak noise for 97% pairs of an aggressor and a victim is estimated within ± 40 mV error. In the case of dsp_alu circuit, the average error is 6 mV and the peak noise for 99% pairs is estimated within ± 40 mV error. As discussed in Ref. [19], driver modeling is one of error sources, and improved driver modeling such as Ref. [23] is necessary when more accurate estimation is required. We also observe that quiet aggressor drivers cause some amount of error, because our analytic noise model treats the coupling capacitance to a quiet aggressor as a capacitance to ground.

5. Conclusion

This paper proposes a method to reduce crosstalk noise as much as possible by transistor sizing after detail-routing. The proposed method optimizes the detail-routed circuits such that MOSFETs inside cells are downsized with interconnects unchanged. The effectiveness of the proposed method is experimentally verified using 2 benchmark circuits. The maximum noise voltage is reduced by more than 50% without delay increase after detail-routing, which reduces the failure risk of crosstalk noise and contributes to high-reliability LSI design.

Acknowledgement

This work is supported in part by the 21st Century COE Program (Grant No. 14213201).

References

[1] H. Zhou and D.F. Wong, "Global routing with crosstalk constraints," Proc. DAC, pp.374–377, 1998.
[2] P. Saxena and C.L. Liu, "Crosstalk minimization using wire perturbations," Proc. DAC, pp.100–103, 1999.
[3] T. Xue, E.S. Kuh, and D. Wang, "Post global routing crosstalk risk estimation and reduction," Proc. ICCAD, pp.302–309, 1996.
[4] C.-P. Chen and N. Menezes, "Noise-aware repeater insertion and wire sizing for on-chip interconnect using hierarchical moment-matching," Proc. DAC, pp.502–506, 1999.
[5] C.J. Alpert, A. Devgan, and S.T. Quay, "Buffer insertion for noise and delay optimization," Proc. DAC, pp.362–367, 1998.
[6] A. Vittal, L.H. Chen, M. Marek-Sadowska, K.-P. Wang, and S. Yang, "Modeling crosstalk in resistive VLSI interconnections," Proc. Int'l Conf. on VLSI Design, pp.470–475, 1999.
[7] J. Cong, D.Z. Pan, and P.V. Srinivas, "Improved crosstalk modeling for noise constrained interconnect optimization," Proc. ASP-DAC, pp.373–378, 2001.
[8] A. Vittal and M. Marek-Sadowska, "Crosstalk reduction for VLSI," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.16, no.3, pp.290–298, March 1997.
[9] I.H.-R. Jiang, Y.-W. Chang, and J.-Y. Jou, "Crosstalk-driven interconnect optimization by simultaneous gate and wire sizing," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.19, no.9, pp.999–1010, Sept. 2000.
[10] M.R. Becer, D. Blaauw, S. Sirichotiyakul, R. Levy, C. Oh, V. Zolotov, J. Zuo, and I.N. Hajj, "A global driver sizing tool for functional crosstalk noise avoidance," Proc. ISQED, pp.158–163, 2001.
[11] T. Xiao and M. Marek-Sadowska, "Crosstalk reduction by transistor sizing," Proc. ASP-DAC, pp.137–140, 1999.
[12] S. Alwar, D. Blaauw, A. Dasgupta, A. Grinshpon, R. Levy, C. Oh, B. Orshav, S. Sirichotiyakul, and V. Zolotov, "Clarinet: A noise analysis tool for deep submicron design," Proc. DAC, pp.233–238, 2000.
[13] T. Xiao and M. Marek-Sadowska, "Gate sizing to eliminate crosstalk induced timing violation," Proc. ICCD, pp.186–191, 2001.

[14] M.R. Becer, D. Blaauw, I. Algor, R. Panda, C. Oh, V. Zolotov, and I.N. Hajj, "Post-route gate sizing for crosstalk noise reduction," Proc. DAC, pp.954–957, 2003.
[15] M. Hashimoto and H. Onodera, "Post-layout transistor sizing for power reduction in cell-based design," IEICE Trans. Fundamentals, vol.E84-A, no.11, pp.2769–2777, Nov. 2001.
[16] H. Onodera, M. Hashimoto, and T. Hashimoto, "ASIC design methodology with on-demand library generation," Proc. Symposium on VLSI Circuits, pp.57–60, 2001.
[17] T. Hashimoto and H. Onodera, "Layout generation of primitive cells with variable driving strength," Proc. SASIMI, pp.122–129, 2000.
[18] M. Hashimoto, K. Fujimori, and H. Onodera, "Automatic generation of standard cell library in VDSM technologies," Proc. ISQED, pp.36–41, 2004.
[19] M. Hashimoto, M. Takahashi, and H. Onodera, "Crosstalk noise estimation for generic RC trees," IEICE Trans. Fundamentals, vol.E86-A, no.12, pp.2965–2973, Dec. 2003.
[20] A.B. Kahng, S. Muddu, and E. Sarto, "On switch factor based analysis of coupled RC interconnect," Proc. DAC, pp.79–84, 2000.
[21] P. Chen, D.A. Kirkpatrick, and K. Keutzer, "Miller factor for gate-level coupling delay calculation," Proc. ICCAD, pp.68–74, 2000.
[22] R. Arunachalam, K. Rajagopal, and L.T. Pileggi, "TACO: Timing analysis with coupling," Proc. DAC, pp.266–269, 2000.
[23] S. Sirichotiyakul, D. Blaauw, C. Oh, R. Levy, V. Zolotov, and J. Zuo, "Driver modeling and alignment for worst-case delay noise," Proc. DAC, pp.720–725, 2001.
[24] M. Hashimoto, Y. Yamada, and H. Onodera, "Equivalent waveform propagation for static timing analysis," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.23, no.4, pp.498–508, April 2004.
[25] R.B. Hitchcock, G.L. Smith, and D.D. Cheng, "Timing analysis of computer hardware," IBM J. Res. Dev., vol.26, no.1, pp.100–105, Jan. 1982.
[26] T. Iwahashi, T. Shibayama, M. Hashimoto, K. Kobayashi, and H. Onodera, "Vector quantization processor for mobile video communication," Proc. ASIC/SOC Conf., pp.75–79, 2000.
[27] Synopsys Inc., Design Compiler Reference Manual, 1998.
[28] Synopsys, Inc., Arcadia Reference Manual, 1999.
[29] M.J.S. Smith, Application-Specific Integrated Circuits, Addison Wesley Longman, 1997.
[30] M. Hashimoto and H. Onodera, "Increase in delay uncertainty by performance optimization," IEICE Trans. Fundamentals, vol.E85-A, no.12, pp.2799–2802, Dec. 2002.
[31] M. Hashimoto and H. Onodera, "A performance optimization method by gate resizing based on statistical static timing analysis," IEICE Trans. Fundamentals, vol.E83-A, no.12, pp.2558–2568, Dec. 2000.

Appendix

We prove that Eq. (1) is valid when $\tau_a = \tau_v$. Equation (1) is rewritten as

$$V_{peak} = \frac{A}{\tau_v}\left(\frac{\tau_v}{\tau_a}\right)^{-\frac{1}{\frac{\tau_v}{\tau_a}-1}}, \quad (\text{A}\cdot 1)$$

where A is $(R_{v1}+R_{v2})C_cV_{dd}$. We replace $\frac{\tau_v}{\tau_a}$ with $1+\delta$, and then Eq. (A· 1) becomes

$$V_{peak} = \frac{A}{\tau_v}(1+\delta)^{-\frac{1}{\delta}}. \quad (\text{A}\cdot 2)$$

We discuss V_{peak} when τ_a becomes equal to τ_v, i.e. we examine the following equation.

$$V_{peak}|_{\tau_a=\tau_v} = \frac{A}{\tau_v}\cdot\lim_{\delta\to 0}(1+\delta)^{-\frac{1}{\delta}}, \quad (\text{A}\cdot 3)$$

$$= \frac{A}{\tau_v}\cdot\lim_{\delta\to 0}\frac{1}{(1+\delta)^{\frac{1}{\delta}}}. \quad (\text{A}\cdot 4)$$

Here, the base of natural logarithms e is defined as follows.

$$\mathrm{e} = \lim_{n\to\infty}(1+\frac{1}{n})^n, \quad (\text{A}\cdot 5)$$

$$= \lim_{\frac{1}{m}\to 0}(1+m)^{\frac{1}{m}}. \quad (\text{A}\cdot 6)$$

Therefore Eq. (A· 4) becomes

$$V_{peak}|_{\tau_a=\tau_v} = \frac{A}{\tau_v}\cdot\frac{1}{\mathrm{e}}. \quad (\text{A}\cdot 7)$$

Equation (1) can be calculated even when $\tau_a = \tau_v$.

Masanori Hashimoto received the B.E., M.E. and Ph.D. degrees in Communications and Computer Engineering from Kyoto University, Kyoto, Japan, in 1997, 1999, and 2001, respectively. Since 2001, he was an Instructor in Department of Communications and Computer Engineering, Kyoto University. Since 2004, he has been an Associate Professor in Department of Information Systems Engineering, Graduate School of Information Science and Technology, Osaka University. His research interest includes computer-aided-design for digital integrated circuits, and high-speed circuit design. He is a member of IEEE, ACM and IPSJ.

Hidetoshi Onodera received the B.E., and M.E., and Dr.Eng. degrees in Electronic Engineering from Kyoto University, Kyoto, Japan, in 1978, 1980, 1984, respectively. He joined the Department of Electronics, Kyoto University, in 1983, and currently a Professor in the Department of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University. His research interests include design technologies for Digital, Analog, and RF LSIs, with particular emphasis on high-speed and low-power design, design and analysis for manufacturability, and SoC architectures. Dr. Onodera has been the Program Chair and General Chair of the ACM/IEEE International Conference on Computer-Aided Design (ICCAD) in 2003 and 2004, respectively, and served on the technical program committees for international conferences including DAC, DATE, ASP-DAC, CICC, etc. He was the Chairman of the IEEE Kansai SSCS Chapter from 2001 to 2002, and the Chairman of the Technical Group on VLSI Design Technologies, IEICE, Japan, from 2000 to 2001.