Crosstalk Noise Reduction in Synthesized Digital Logic Circuits

Oleg Milter and Avinoam Kolodny

***Abstract*—As CMOS technology scales into the deep submicrometer regime, digital noise is becoming a metric of importance comparable to area, timing, and power, for analysis and design of CMOS VLSI systems. Noise has two detrimental effects in digital circuits: First, it can destroy logical information carried by a circuit net. Second, it causes delay uncertainty: Noncritical paths might become critical because of noise. As a result, circuit speed becomes limited by noise, primarily because of capacitive coupling between wires. Most design approaches address the crosstalk noise problem at the layout generation stage, or via postlayout corrections. With continued scaling, too many circuit nets require corrections for noise, causing a design convergence problem. This work suggests to consider noise at the gate-level netlist generation stage. The paper presents a simplified analysis of on-chip crosstalk models, and demonstrates the significance of crosstalk between local wires within synthesized circuit blocks. A design flow is proposed for automatically synthesizing CMOS circuits that have improved robustness to noise effects, using standard tools, by limiting the range of gate strengths available in the cell library. The synthesized circuits incur a penalty in area/power, which can be partially recovered in a single postlayout corrective iteration. Results of design experiments indicate that delay uncertainty is the most important noise-related concern in synthesized static CMOS logic. Using a standard synthesis methodology, critical path delay differences up to 18% of the clock cycle time have been observed in functional blocks of microprocessor circuits. By using the proposed design flow, timing uncertainty was reduced to below 3%, with area and power penalties below 20%.**

***Index Terms*—Crosstalk, delay uncertainty, noise, noise management, signal integrity, synthesis, timing verification.**

I. Introduction

Capacitive crosstalk noise has become a major concern in design of high-performance VLSI digital circuits as a result of interconnect and device scaling [1]–[3]. In deep submicrometer technology, cross-coupling capacitance between neighboring signal nets is the dominant component of total net capacitance[1], hence any signal net making a logic transition may act as an *aggressor*, injecting charge into adjacent nets, considered as *victims*. The coupled noise voltage waveform might induce a *logic hazard* on a "quiet" victim net, leading to eventual logic failure. If noise is injected into a victim net during logic transition it can modify the victim's waveform, causing *delay uncertainty* [4]–[6] depending on the detailed behavior of aggressor signals. Consequently, a noncritical path might become a critical speed path, and clocking frequency may be limited by noise. Also, a short minimum-delay path may become faster and cause a fatal race, which cannot be corrected by adjusting clock frequency [7].

The engineering approach to crosstalk noise has been initially focused on postlayout verification, using extracted interconnect models and various simplification methods to calculate the peak noise voltage. Methodologies using classical dc noise margins [8] have become too conservative for deep submicrometer circuits, hence, the theory has been extended to analyze propagation of noise transients in logic circuits. Metrics such as noise sensitivity and noise stability were defined based on the peak values of noise waveforms [9]–[11]. Static noise analysis was developed for propagating worst case noise peaks through logic in large circuits. Results are checked against noise design rules to ensure immunity of circuits to noise peaks. In recent years, progress was made in modeling noise-induced delay uncertainty in static timing analysis [5], [12]–[14]. With technology progress, a large number of circuit nets are flagged as potential noise problems, requiring new pruning methods to reduce pessimism and manage uncertainty [15], [16].

Noise violations can be fixed by wire spacing, shield insertion, incremental rerouting, repeater insertion, or device sizing [14]–[25]. Effectiveness of noise avoidance techniques has been analyzed in [24]. Although postlayout changes in the circuit are effectively applied in the industry to reduce noise, they might cause costly iterations and engineering convergence problems. Our main interest is crosstalk within logic-synthesized, automatically-routed large blocks of random logic, where postlayout corrective changes should be avoided as much as possible. At the logic synthesis stage, crosstalk noise considerations are typically ignored because cross-coupling information is still unknown. A probabilistic approach has been introduced in [26] to estimate crosstalk before routing is performed, which may be employed in a placement-aware synthesizer. In the future, integration of logic synthesis with routing tools is conceivable, but a more practical approach would be to incorporate noise-related heuristics into logic synthesis, applied before detailed layout information is available.

In this paper, we address the issue of considering noise effects in synthesis of static CMOS logic, using standard tools. Our goal is to increase circuit robustness to noise and to reduce timing uncertainty by modifying the prelayout design flow, in order to minimize postlayout corrections. We analyze the dependence of noise effects on basic circuit parameters, and observe that uniform-strength drivers can efficiently limit crosstalk noise and are most appropriate for resistive interconnect. Next, we propose a noise-aware design flow to control the ratio of driver strengths in the synthesized circuit. Results of design experiments applying the proposed flow to circuit blocks from microprocessors in 0.13 μm and 0.09 μm technology are presented. We conclude the paper with a short discussion.

II. Noise Effects and Their Dependence on Circuit Parameters

Fig. 1(a) shows a classical aggressor/victim model. Associated circuit-simulation waveforms, demonstrating noise glitches with amplitude V_p and delay uncertainty Δt are shown in Fig. 1(b). The simulation assumes a large-size inverter as the aggressor and a small-size inverter as the victim. Both drivers were selected from an industrial cell library. The inverters drive 50 μm of parallel metal 3 wires at minimum pitch, loaded by minimum-size gates. These simulation results exemplify the significance of crosstalk noise even for relatively short local wires within functional blocks [27]. We assume a single aggressor as a typical situation, although a victim net might be attacked simultaneously from both sides by two aggressors. The following analysis can be modified to account for such a pessimistic scenario without changing the nature of the problem.

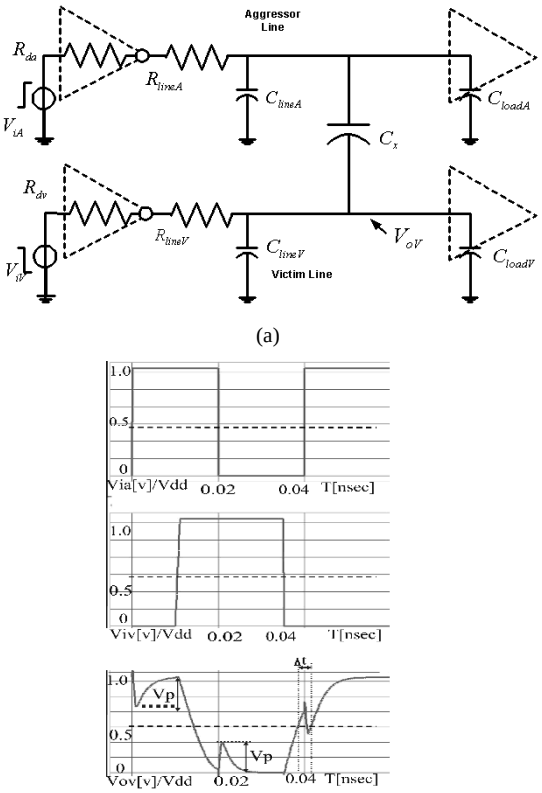
Several models have been presented in the literature for calculating crosstalk noise effects analytically, instead of using nonlinear circuit simulation [4], [6], [22], [28]–[31]. These models use linearizations to approximate the behavior of a logic gate by a Thevenin voltage source in series with an effective linear output resistance, and a lumped *RC* network to approximate the distributed coupled wires. In order to gain insight by analyzing a simple expression, we model the peak noise V_p for step inputs by (1) as shown at the bottom of the next page, [6],

Manuscript received July 22, 2002; revised December 21, 2002.

O. Milter is with Intel Corporation, Israel Development Center, Haifa 32000, Israel (e-mail: oleg.milter@intel.com).

A. Kolodny is with theVLSI Systems Research Center, Electrical Engineering Department, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: kolodny@ee.technion.ac.il).

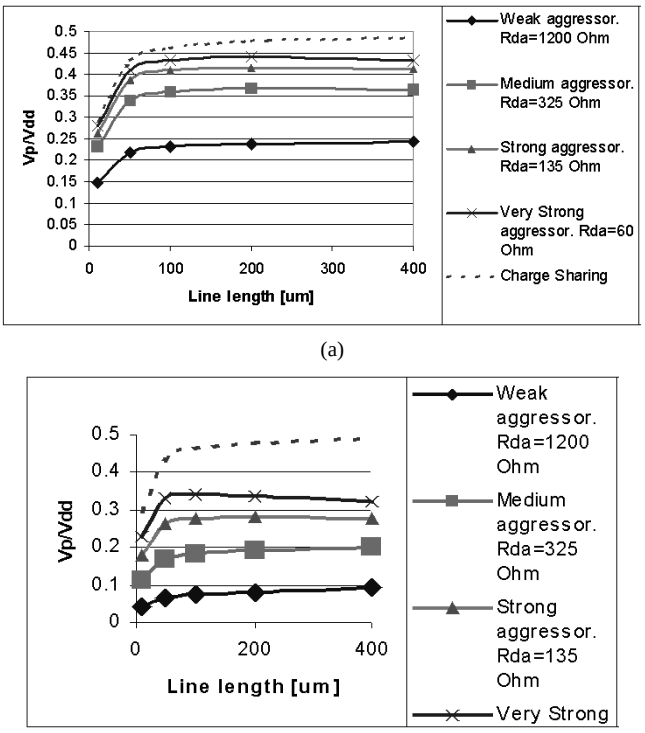
Digital Object Identifier 10.1109/TVLSI.2003.817551



(a)

(b)

Fig. 1. (a) Classical aggressor/victim model. (b) Circuit simulation waveforms for 50 micronmeters of M3 wires with a very strong aggressor driver and a weak inverter as victim driver, corresponding to $\boldsymbol{R_{dA} = 60\ \Omega, R_{dV} = 1200\ \Omega}$ in the linearized model of (a). Both wires are loaded by weak inverters.



(a)

(b)

Fig. 2. Normalized peak noise for various aggressors as a function of line length; Metal 3 at minimum pitch. (a) The victim is weak ($\boldsymbol{R_{dV} = 1200\ \Omega}$). (b) The victim is medium strength ($\boldsymbol{R_{dV} = 325\ \Omega}$); Resistive shielding of the aggressor causes a decrease in peak noise for the very strong aggressor at wirelengths above 100 $\boldsymbol{\mu}$m.

[30] where V_p is the peak noise voltage, V_{dd} is the supply voltage, R_{dA}, R_{dV} are effective resistances of aggressor and victim driver, R_{lineA}, R_{lineV} are lumped wire resistances of aggressor and victim line, C_{lineA}, C_{lineV} are lumped ground capacitances for aggressor and victim line, C_{loadA}, C_{loadV} are capacitances of aggressor and victim fanout gates, and C_x is lumped cross-coupling capacitance.

If wire resistances are neglected (R_{lineA}, $R_{\mathrm{lineV}} = 0$), and a very strong aggressor is assumed ($R_{dA} = 0$), then (1) becomes equivalent to the well-known Charge Sharing Model [22], [29] which is an upper bound for the coupled noise

$$\frac{V_p}{V_{dd}} = \frac{C_x}{C_x + C_{\mathrm{lineV}} + C_{\mathrm{loadV}}}. \tag{2}$$

The resistance terms in (1) have a suppressing effect on the noise peak. In particular, the aggressor wire resistance R_{lineA} provides "resistive shielding" as distance from the aggressor grows. However, a similar resistive shielding effect on the victim wire might worsen the noise situation when multiple aggressors are considered [19], or when an aggressor is coupled to the victim line at the receiver side [31], while the victim's driver ability to supply charge is limited by R_{lineV}. In practical synthesized functional blocks it is reasonable to assume that the driver resistances dominate the wire resistance terms in (1). By further assuming that the wire capacitances C_{lineV} and C_{lineA} are identical and the fanout loads are relatively small, we obtain

$$\frac{V_p}{V_{dd}} \approx \frac{R_{dV}}{R_{dV} + R_{dA}} \frac{C_x}{C_x + C_{\mathrm{lineV}}}. \tag{3}$$

Hence, the noise peak under these assumptions is determined by the ratio of driver strengths, since the capacitance ratio at a given metal pitch is a fixed process parameter.

The normalized noise peak voltage V_p/V_{dd} in the configuration depicted in Fig. 1, as determined by (1), is plotted in Fig. 2(a) versus line length L, for various aggressor driver strengths (victim driver is constant and is weak). Fig. 2(b) shows similar data for a medium strength victim driver. The figures demonstrate the importance of driver-strength ratio on crosstalk between local wires in functional blocks. Maximal noise occurs when a strong aggressor injects charge into a medium-length wire. Short wires incur less noise because load capacitance is dominant. Long wires demonstrate resistive shielding of the aggressor, causing a decrease in peak noise. Note that if the strength of an aggressor is reduced too much, the aggressor itself may become a victim in another voltage transition.

In Fig. 3, V_p/V_{dd} is plotted versus L again, but here the victim driver sizes were selected from a cell library according to L, so as to keep the stage delay approximately constant at 25 P/s. The results show that noise is significant for short and medium wirelengths. It is also evident

$$\frac{V_p}{V_{dd}} = \frac{(R_{dV} + R_{\mathrm{lineV}})C_x}{(R_{dA} + R_{\mathrm{lineA}})(C_{\mathrm{lineA}} + C_{\mathrm{loadA}} + C_x) + (R_{dV} + R_{\mathrm{lineV}})(C_{\mathrm{lineV}} + C_{\mathrm{loadV}} + C_x)} \tag{1}$$

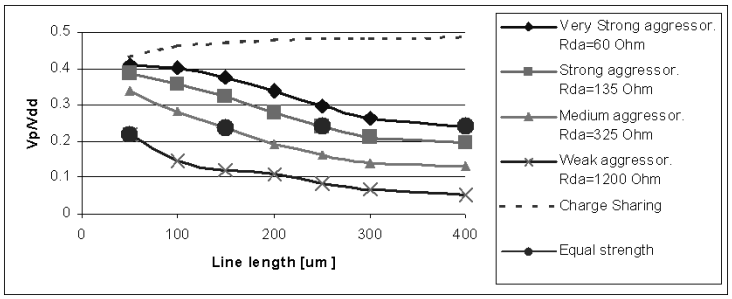


Fig. 3. Normalized peak noise versus wirelength with victim driver size selected to obtain a total stage delay of approximately 25 picoseconds (including wire delay). The dots represent cases when aggressor and victim drivers are of the same strength.

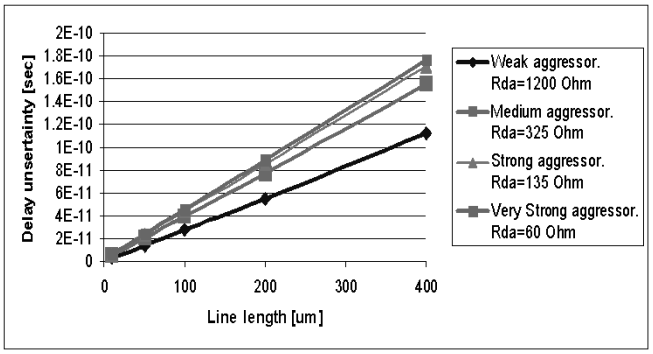


Fig. 4. Delay uncertainty $\Delta t_{\max}$ for various aggressors as a function of line length, with a weak victim driver ($\mathbf{Rdv = 1200}\ \Omega$).

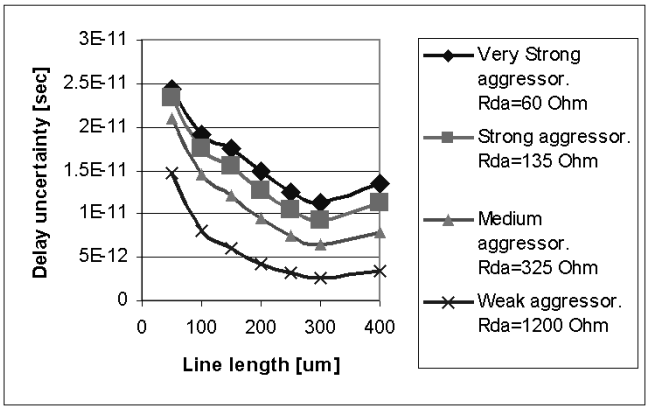


Fig. 5. Delay uncertainty for networks with equal stage delay.

in Fig. 3 that when both aggressor and victim are of equal strength (having equal effective output resistances), the peak noise is below approximately 25%, as can be seen from (3) assuming $R_{da} = R_{dv}$ and $C_{\text{lineV}} \approx C_x$.

For analytically modeling the delay uncertainty caused by effects of crosstalk noise on circuit timing, a common practice is to use a decoupled circuit model for each net, in which the cross-capacitance is multiplied by a Miller factor and connected to ground [1], [29]. Another approach is to use superposition of "quiet" waveform with the noise waveform and compute a delay change [1], [5]. We use the superposition-based approximations developed and accuracy-verified in [32]. Assuming worst-case aggressor alignment, an upper bound for the delay change Δt is expressed as

$$\Delta t_{\max} = \tau_r \ln(2V_p/V_{dd} + 1) \tag{4}$$

where

$$\tau_r = (R_{dV} + R_{\text{lineV}})(C_{\text{loadV}} + C_{\text{lineV}} + C_x). \tag{5}$$

Because of the logarithmic dependence on V_p, R_{dV} is the single most important parameter affecting delay uncertainty $\Delta t_{\max}$.

Fig. 4 shows calculated delay uncertainty as a function of wirelength for a weak victim, which can become impractically large as L grows. Delay uncertainty for a fixed stage delay is plotted in Fig. 5 versus wirelength.

High uncertainty exists in relatively short wires, because their drivers are weak. For longer wires with stronger drivers V_p and τ_r are reduced, delay uncertainty goes through a minimum and then starts growing because of wire resistance, demonstrating the need to insert repeaters for noise on long lines. Selecting gates with equal drive-strengths is useful when either net may be considered as the victim. It is useful also before routing, when adjacency of nets has not been determined yet. This observation has been made by [22], who emphasized the role of driver strength in noise calculations, and suggested to perform track assignments in channel routing by sorting nets according to the strength of their drivers, such that adjacent wires will have roughly equal drivers. We extend this idea by suggesting that noise-aware logic synthesis should use a cell library with uniform cell-strengths, such that the crosstalk peak between any pair of nets will be bounded by approximately $0.25V_{dd}$. Since threshold voltage in scaled processes is around $0.3V_{dd}$, the induced noise will not propagate through the gates. Cell libraries containing a variety of gate sizes for each logic function have been developed for timing optimization while driving purely capacitive loads. However, in state-of-the-art processes the interconnect resistance is significant, such that oversizing a driver may not be very beneficial to reduce delay, while repeater insertion is the better way to drive wires [33]. In future processes, even local wires may be repeater intensive, with all gates and repeaters having a uniform strength matched to the characteristic interconnect segments.

To enable the uniform-driver-size methodology, we must employ a capacitance-management policy, such that all high-fanout nets will be broken into buffered trees [3], [34], [35] and repeaters will be inserted in all wires as appropriate for delay optimization in resistive interconnect [33], [36]. Thus, interconnect should be divided into roughly uniform-capacitance chunks matching the uniform-strength drivers, such that both Vp and τ_r are bounded.

Section III describes a heuristic method for picking a uniform driver strength and a typical capacitance for a given logic block, within a complete experimental circuit-synthesis flow, and presents results of design experiments.

III. Experimental Design Flow and Results

Our experimental noise driven synthesis flow includes the steps illustrated in Fig. 6. Driver strength selection is performed for each circuit by running a preliminary regular synthesis and generating a wire load

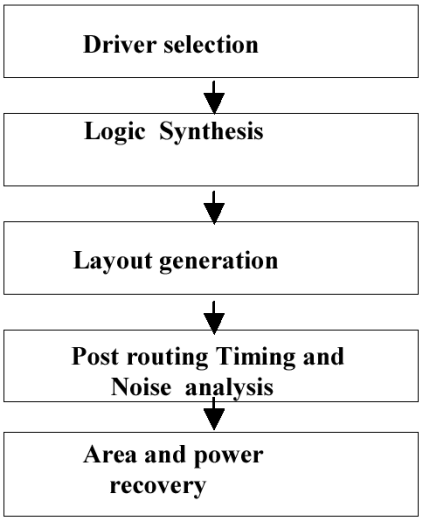


Fig. 6. Steps of noise-driven flow.

model, which correlates expected nodal capacitance with logic fanout. Using the wire load model, we calculate the capacitance corresponding to fanout 3 as a "typical load." Our experience shows that a fanout of 3 represents most nets in typical circuits (95% of the nets have fanout 3 or less). Next, we select a driver size from the library to drive such a typical load at acceptable stage delay and transition times.

The second stage is logic synthesis. In our flow we limit the cell library to use only cells having the selected strength, and restrict the maximum fanout of a net, such that the synthesizer must create buffer trees on high-fanout nodes. Placement and routing are done in a standard fashion, without any timing optimizations or crosstalk optimizations. After layout *RC* extraction, static noise analysis is performed in order to monitor noise peaks, and noise-aware timing analysis [13] is performed in order to monitor delay uncertainties. At the last stage, we use real routing data for postlayout optimizations. We use the timing analyzer report to identify nets which pick up negligible crosstalk noise. The drivers of such nets can be safely downsized at this stage for area and power recovery. The downsizing step maintains cell placement and routing. There is practically no wasted area associated with keeping the layout essentially unchanged, because synthesized random logic circuits are typically routing limited anyway. Final timing verification is used to check that these cells have not become victims.

We applied the flow to eight circuits from microprocessor designs, using 0.13-μm and 0.09-μm silicon technologies [37], [38]. Although, these are small/medium circuits they represent a challenging crosstalk design problem because of their high operating frequency (more than 1.5-GHz clock rate). The two technologies were found to be similar in terms of noise behavior, because of improved process characteristics such as low-*k* dielectric in the 0.09-μm generation. Table I shows basic data on our test circuits.

Table II presents peak noise results. In all of our test cases, peak noise has not been excessive, even when using a standard synthesis flow. The proposed flow reduced average noise peak and maximal nodal peak. In general, functional failure because of noise peaks is not a critical issue in synthesized static design. The main benefit of peak noise reduction is its influence on reduction of delay uncertainty.

Table III shows timing results after synthesizing the circuits using a standard design flow with unrestricted gate sizes and fanout. The delays are normalized and presented in percents of the clock cycle. Postlayout analysis shows that delay uncertainty effect can significantly increase critical path delay (up to $\sim$18% of cycle time in our test cases).

TABLE I
TEST CIRCUITS DESCRIPTION

Name	Process	# Cells	Layout Area utilization
Circuit 1	0.13 um	1000	75%
Circuit 2	0.13 um	1200	65%
Circuit 3	0.13 um	7000	55%
Circuit 4	0.13 um	10000	75%
Circuit 5	0.09 um	1900	50%
Circuit 6	0.09 um	5500	50%
Circuit 7	0.09 um	6300	55%
Circuit 8	0.09 um	12000	60%

TABLE II
NORMALIZED PEAK NOISE RESULTS V_p/V_{dd}

Test case	Standard flow			Proposed flow		
	Average Peak Noise	Maximal Nodal Peak Noise	# of Peaks > 0.1 Vdd	Average Peak Noise	Maximal Nodal Peak Noise	# of Peaks > 0.1 Vdd
Circuit 1	0.017	0.252	6	0.012	0.109	1
Circuit 2	0.021	0.166	33	0.017	0.101	2
Circuit 3	0.017	0.173	40	0.016	0.149	17
Circuit 4	0.019	0.181	71	0.013	0.103	2
Circuit 5	0.017	0.225	10	0.012	0.195	6
Circuit 6	0.019	0.221	25	0.015	0.185	14
Circuit 7	0.023	0.209	30	0.020	0.163	8
Circuit 8	0.025	0.145	21	0.016	0.116	9

TABLE III
STANDARD SYNTHESIS TIMING RESULTS (% OF CYCLE TIME)

Test case	Worst path slack w/o uncertainty	Worst path slack with uncertainty	Noise-induced timing difference	Worst nodal uncertainty
Circuit 1	0.0%	-7.1%	7.1%	3.5%
Circuit 2	-11.2%	-25.7%	14.6%	35.4%
Circuit 3	-7.8%	-10.0%	2.2%	8.3%
Circuit 4	-31.5%	-49.3%	17.9%	16.3%
Circuit 5	-16.3%	-19.0%	2.8%	6.8%
Circuit 6	-15.3%	-26.5%	11.3%	12.5%
Circuit 7	-25.3%	-32.8%	7.5%	9.4%
Circuit 8	0.0%	-8.0%	8.0%	7.9%

Noncritical circuits can become timing critical because of noise, as in circuits 1 and 8.

Table IV shows improvements in delay uncertainty for these circuits when synthesized using the proposed noise-driven flow, before repeater insertion for wire timing optimization, and before area/power recovery. For all test cases maximum noise-induced delay degradation on paths is less than 2% of cycle time. Worst uncertainty on a single net has been reduced from 35% (in Table III) to $\sim$3%. Note that the circuits in Table IV are logically equivalent to those in Table III, but they were synthesized, placed, and routed independently; hence, the worst-path slack in circuit 3 came out even worse than in Table III because of a long wire that will need repeater insertion for speed at the next stage.

A clear effect of our methodology is speed improvement even without considering noise, manifested in reduced negative slacks, due to usage of stronger cells. The average cell strength is larger at this point (compared with standard flow), and this is reflected in power dissipation of the circuits. Buffers are added and the total cell area grows too at this point. Note from (3)–(5), that peak noise improvement is obtained from uniformity of driver strengths, while timing uncertainty is also directly reduced by lower-resistance victim

TABLE IV
RESULTS AFTER "NOISE DRIVEN" SYNTHESIS

test case	worst path slack w/o uncertainty	worst path slack with uncertainty	noise-induced timing difference	worst nodal uncertainty
Circuit 1	-1.6%	-1.6%	0.0%	1.5%
Circuit 2	-10.4%	-12.1%	1.6%	3.0%
Circuit 3	-10.4%	-12.1%	1.6%	3.0%
Circuit 4	-3.1%	-4.8%	1.7%	1.6%
Circuit 5	-16.5%	-17.5%	1.0%	1.1%
Circuit 6	-14.8%	-15.8%	1.0%	1.2%
Circuit 7	-24.0%	-24.8%	0.8%	4.8%
Circuit 8	0.0%	-2.8%	2.8%	5.1%

TABLE V
FINAL RESULTS OF "NOISE DRIVEN FLOW" AFTER AREA/POWER RECOVERY

test case	worst path slack w/o noise	worst path slack with uncertainty	noise-induced timing difference	worst nodal uncertainty	cell area change	power change
Circuit 1	0.0%	-2.6%	2.6%	2.1%	20.6%	-7.4%
Circuit 2	-10.9%	-12.5%	1.6%	3.3%	19.9%	18.0%
Circuit 3	-1.0%	-1.3%	0.3%	4.5%	-0.8%	13.7%
Circuit 4	0.0%	0.0%	0.0%	2.6%	8.1%	6.3%
Circuit 5	-17.5%	-18.5%	1.0%	2.0%	12.3%	13.41%
Circuit 6	-12.3%	-13.5%	1.3%	1.4%	17.0%	7.82%
Circuit 7	-24.2%	-25.8%	1.6%	5.3%	19.2%	2.37%
Circuit 8	-12.3%	-13.5%	1.3%	1.4%	9.0%	17.1%

drivers. At the last stage of the design flow, we recover most of these extra area and power by selectively downsizing some gates without modifying the placement and routing. Since this step does not adhere to the uniform strength rule, it is performed only for nets which do not have significant noise peaks and delay changes. The downsizing is performed automatically by the logic synthesizer in post layout mode, given the performance requirements and "don't touch" directives for all nodes that have significant noise effects. In our test cases, 40% to 50% of the cells were downsized at this step.

Table V shows final results of the complete flow, after the downsizing step. Area and power penalties compared with standard synthesis have been reduced to below 20%, while noise effects are kept within acceptable margins. The area here is the sum of cell areas (not just active transistor areas). The total floor-plan area in all the circuits remained unchanged because it was limited by routing resources in all cases (Table I). Circuit 1 is an interesting case where total cell area increased because of additional cells and buffers compared with the standard flow, but power went down with weaker drivers in this synthesis. A corrective design process starting from standard synthesis of the circuit would be much harder to implement and slower, because numerous noise violations would have to be handled, and iterative convergence would be required in such a process. While our results were obtained with standard tools to perform simple downsizing, specialized postlayout crosstalk-aware optimizations [14] could be applied instead at the last stage.

IV. DISCUSSION

Circuits with improved noise robustness have been synthesized by using uniform output resistance gates from a restricted cell library, and by avoiding high-fanout nets. Buffers and repeaters having the same output strength as the gates are added to the circuit accordingly, such that no strong aggressors exist in the circuit. This approach guarantees reduced peak noise voltage while using standard tools, compared with unconstrained synthesis which attempts to use weak drivers whenever possible, considering only timing and area while ignoring noise effects. Timing uncertainty is significantly reduced in these circuits, because of the reduced noise peaks and because of the reduced time constant at victim nets, due to avoidance of weak victim gates.

Timing uncertainty appears to be the most important noise-related concern in synthesized static CMOS logic blocks. Noise stability is of lesser concern, because the restoring properties of static gates in the current technology generation easily prevent propagation of noise when design-rules are followed. However, there is no restoring mechanism for delay uncertainties. Delay uncertainty accumulates statistically along logic paths. If ignored, delay uncertainty can easily transform noncritical paths to timing critical. We have observed critical-path delay differences up to 18% of the clock-cycle time in our test cases. Beyond noise-induced performance degradation, delay uncertainty also affects short-path delays and requires additional delay padding. Checking the magnitude and likelihood of path delay uncertainty is a very challenging task [16] when each net can have a significant delay change. Therefore, it is recommended to limit these noise-effects *a priori* by using the proposed approach. A totally different approach is crosstalk prevention by shielding of all signal nets using additional wires. Such a methodology is considered impractical for our test circuits in the current technology, since their density is already limited by routing resources. The main penalty of our methodology is an increase in area and power dissipation. We have seen that a single pass of postlayout gate resizing, directed by a noise-aware timing verification run, can reduce these penalties while maintaining improved noise performance.

REFERENCES

[1] D. Sylvester and C. Hu, "Analytical modeling and characterization of deep-submicrometer interconnect," *Proc. IEEE*, vol. 89, pp. 634–664, May 2001.
[2] R. H. Ho, K. W. Mai, and M. H. Horowitz, "The future of wires," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 490–504, April 2001.
[3] J. Cong, "An interconnect-centric design flow for nanometer technologies," *Proc. IEEE*, vol. 89, pp. 505–528, Apr. 2001.
[4] K. T. Tang and E. G. Friedman, "Delay and noise estimation of CMOS logic gates driving coupled resistive-capacitive interconnections," *Integration, VLSI J.*, vol. 29, pp. 131–165, 2000.
[5] F. Dartu and L. Pileggi, "Calculating worst-case gate delays due to dominant capacitive coupling," in *Proc. Design Automation Conf. (DAC)*, 1997, pp. 46–51.
[6] A. B. Kahng, S. Muddu, and D. Vidhani, "Noise and delay uncertainty studies for coupled RC interconnects," in *IEEE Int. ASIC/SOC Conf.*, Sept. 1999, pp. 3–8.
[7] E. G. Friedman, "Clock distribution networks in synchronous digital integrated circuits," *Proc. IEEE*, vol. 89, pp. 665–692, May 2001.
[8] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*. Fort Worth: Texas Saunders College, 1991.
[9] K. Shepard and V. Narayanan, "Noise in deep submicron digital design," in *Proc. ICCAD*, 1996, pp. 524–531.
[10] ——, "Conquering noise in deep-submicron digital ICs," *IEEE Des. Test Comput.*, vol. 15, pp. 51–62, Jan.–Mar. 1998.
[11] K. L. Shepard, "Design methodology for noise in digital integrated circuits," in *Proc. Design Automation Conf. (DAC)*, San Francisco, CA, 1998, pp. 94–99.
[12] Y. Cao, T. Sato, X. Huang, C. Hu, and D. Sylvester, "New approaches to noise-aware static timing analysis," in *Proc. ACM/IEEE Int. Workshop Timing Issues (TAU)*, Austin, TX, Dec. 2000.

[13] B. Franzini and C. Forzan, "Crosstalk aware static timing analysis environment," in *SNUG Europe 2001*, [Online]. Available: http://www.synopsys.com/products/primetime_si/crosstalk_aware.pdf.
[14] T. Xiao and M. Marek-Sadowska, "Gate sizing to eliminate crosstalk induced timing violation," in *Proc. ICCD 2001*, pp. 186–191.
[15] P. Chen and K. Keutzer, "Toward true crosstalk noise analysis," in *Proc. ICCAD*, 1999, pp. 132–137.
[16] S. B. K. Vrudhula, D. Blaauw, and S. Sirichotiyakul, "Estimation of the likelihood of capacitive coupling noise," in *Proc. DAC 2002*, pp. 653–658.
[17] D. Kirkpatrick and A. Sangiovanni-Vincentelli, "Techniques for crosstalk avoidance in the physical design of high-performance digital systems," in *Proc. ICCAD'94*, 1994, pp. 616–620.
[18] H. Zhou and D. F. Wong, "Global routing with crosstalk constraints," in *Proc. Design Automation Conf. (DAC)*, 1998, pp. 374–378.
[19] C. Alpert and A. Devgan, "Buffer insertion for noise and delay optimization," *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 1633–1645, 1999.
[20] C. Chen and N. Menezes, "Noise-aware repeater insertion and wire sizing for on-chip interconnect using hierarchical moment-matching," in *Proc. '99 Design Automation Conf. (DAC)*, pp. 502–506.
[21] D. Li *et al.*, "A repeater optimization methodology for deep sub-micron, high-performance processors," in *Proc. Int. Conf. Computer Design*, 1997, pp. 726–731.
[22] A. Vittal and M. Marek-Sadowska, "Crosstalk reduction for VLSI," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 290–298, 1997.
[23] T. Xiao and M. Marek-Sadowska, "Transistor sizing for crosstalk reduction," in *Proc. ASP-DAC*, 1999, pp. 137–141.
[24] M. Becer, D. Blaauw, V. Zolotov, R. Panda, and I. Hajj, "Analysis of noise avoidance techniques in DSM interconnects using a complete crosstalk noise model," in *Proc. DATE 2002*, pp. 456–463.
[25] M. Becer, D. Blaauw, O. Chanhee, I. Hajj, Z. Jingyan, R. Levy, S. Sirichotiyakul, and V. Zolotov, "A global driver sizing tool for functional crosstalk noise avoidance," in *Proc. ISQED 2001*, pp. 158–163.
[26] M. Becer, R. Panda, D. Blaau, and I. Hajj, "Pre-route noise estimation in deep submicron integrated circuits," in *Proc. ISQED 2002*, pp. 413–418.
[27] D. Sylvester and K. Keutzer, "Getting to the bottom of deep submicron," in *Proc. ICCAD*, 1998, pp. 203–211.
[28] A. Rubio, N. Itzaki, X. Xu, and K. Kinoshita, "An approach to the analysis and detection of crosstalk faults in digital VLSI circuits," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 387–395, 1994.
[29] T. Sakurai, "Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSI's," *IEEE Trans. Electron Devices*, vol. 40, pp. 118–124, Jan. 1993.
[30] A. Vittal, A. L Chen, M. Marek-Sadowska, K.-P. Wang, and S. Yang, "Crosstalk in VLSI interconnections," *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 1817–1824, Dec. 1999.
[31] J. Cong, D. Pan, and P. Srinivas, "Improved crosstalk modeling for noise constrained interconnect optimization," in *Proc. ACM/IEEE Int. Workshop Timing Issues (TAU)*, Austin, Texas, 2000.
[32] T. Sato, Y. Cao, D. Sylvester, and C. Hu, "Characterization of interconnect coupling noise using in-situ delay change curve measurements," in *Proc. IEEE ASIC/SoC Conf.*, 2000, pp. 321–325.
[33] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.
[34] L. P. P. van Ginneken, "Buffer placement in distributed RC-tree networks for minimal Elmore delay," in *Proc. Int. Symp. Circuits and Syst.*, 1990, pp. 865–868.
[35] V. Adler and E. Friedman, "Repeater design to reduce delay and power in resistive interconnect," *IEEE Trans. Circuits Syst.*, vol. 45, pp. 607–616, May 1998.
[36] R. Otten, R. Robert, and R. Brayton, "Performance planning," *Integration, VLSI J.*, vol. 29, pp. 1–24, 2000.
[37] S. Tyagi *et al.*, "A 130 nm generation logic technology featuring 70nm transistors, dual Vt transistors and 6 layers of Cu interconnects," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, 2000, pp. 567–570.
[38] S. Thompson *et al.*, "A 90 nm logic technology featuring 50 nm strained silicon channel transistors, 7 layers of Cu Interconnects, Low k ILD, and 1 μm^2 6-T SRAM cell," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, 2002, pp. 61–64.