The challenge of high-performance, deep-submicron design in a turnkey ASIC environment

K. L. Shepard
Department of Electrical Engineering
Columbia University
New York, NY 10027

Abstract

In this paper, we review the trends and techniques that are shaping high-performance ASIC design in deep-submicron technology. The importance of interconnect in determining performance is breaking the clean division between logical and physical design, while noise analysis is becoming as important as timing analysis in ensuring correct functionality. The growing complexity of rules in precharacterized libraries is making analysis tools that understand transistors more attractive. In addition, new layout techniques allow libraries to be created dynamically, creating the opportunity for truly transistor-level synthesis. Domino logic and pass-transistor circuits promise to find their way into synthesis with increasing performance demands at the high end.

1 Interconnect upsets the traditional ASIC flow

The ASIC industry has traditionally been driven by the design flow shown in Figure 1(a), which has a strict partitioning between the physical design and the logical design. (Static timing analysis is shown since it has largely displaced simulation-based timing verification.) The fabless design shop is responsible for the HDL design and synthesis, passing a netlist to the foundry for physical design. The success of this approach is largely based on the *predictability* of timing after placement and routing; that is, the relatively small effect that interconnect loading has on timing results so that crude interconnect capacitance estimate can be used in synthesis and few, if any, timing problems would be evident after physical design. The effect of interconnect resistance is small and is traditionally not even included in the postplacement extraction.

With deep submicron technologies, the growing importance of interconnect capacitance, resistance, and coupling breaks the clean division between logical and physical design, creating a design flow more like Figure 1(b), which probably reflects the state-of-the-art design methodology. Creating close interaction between synthesis and placement enables more accurate capacitance information based on Steiner-tree route estimates to be used in the synthesis process, allowing better repowering and fanout correction[1, 2]. Timing-driven placement approaches can sometimes be employed to limit the capacitance along critical paths. After placement and routing, detailed RC extraction is performed and the information fed back to the timer in the form of SDF[3] or, better, SPEF[4]. The clock network usually requires more detailed analysis to verify that acceptable skew targets have been achieved. If coupling information is available, then attempts are made to reflect the effect of coupling on delay and slew through the SDF or SPEF. A coupled RC extraction also forms the basis for static noise analysis described in the following section.

The difficulty with the flow of Figure 1(b) is that it does not include automated techniques to control RC delays and interconnect coupling, such as repeater insertion[5], wide-wire routing[6, 7], increased spacing of interconnect lines, or the use of ground or power shielding. As a result, the methodology of Figure 1(b) works best in a hierarchical environment in which the global routes are done with more user involvement. To enable a more turnkey approach to the design of the global interconnect, RC extraction must be incorporated into the synthesis process and directives applied to both placement and routing as a result of static timing and static noise analysis. Coupled multiport interconnect macromodels are an important part of this analysis[8].

2 Noise as a new metric for design

Noise figures prominently in the methodologies of Figure 1(b) and 1(c). This recognizes the fact that noise is a design metric in deep submicron designs of comparable importance to area and timing. Increasing interconnect densities, faster clock rates (which mean comparably faster slew rates), and scaling threshold voltages act to degrade the signal-to-noise ratio for CMOS digital designs.

Noise has two deleterious effects on digital design. When noise acts against a normally static signal, it can transiently destroy the logical information carried by the static node in the circuit. If this ultimately results in incorrect machine state stored in a latch, functional failure will result. When noise acts simultaneously with a *switching* node, this is manifest as a change in the timing (delay and slew) of the transition (a noise-on-delay effect). Static noise analysis addresses the former effect, while static timing analysis must consider the latter.

Noise occurs because of the use of large-signal voltage changes to switch logic levels. As shown in Figure 2, these switching events interfere with static signals because of coupling through the interconnect (coupling noise), through the transistors (charge-sharing noise), through the substrate or nwell (substrate noise), or through the power supply (power-supply noise). Static noise analysis[9, 10] when used along with a power-supply integrity analysis[2, 11] ensures that the chip will function in the presence of all possible noise sources.

Proactive measures almost always guide the design of the power and ground distribution to ensure adequate rigidity. Similar proactive measures are also used in the design of the ASIC library to build in extra noise margin, particularly for the latch circuits, but this comes at a distinct cost in performance. Performing noise analysis, as well as timing

0-7803-4980-6/98/$10.00

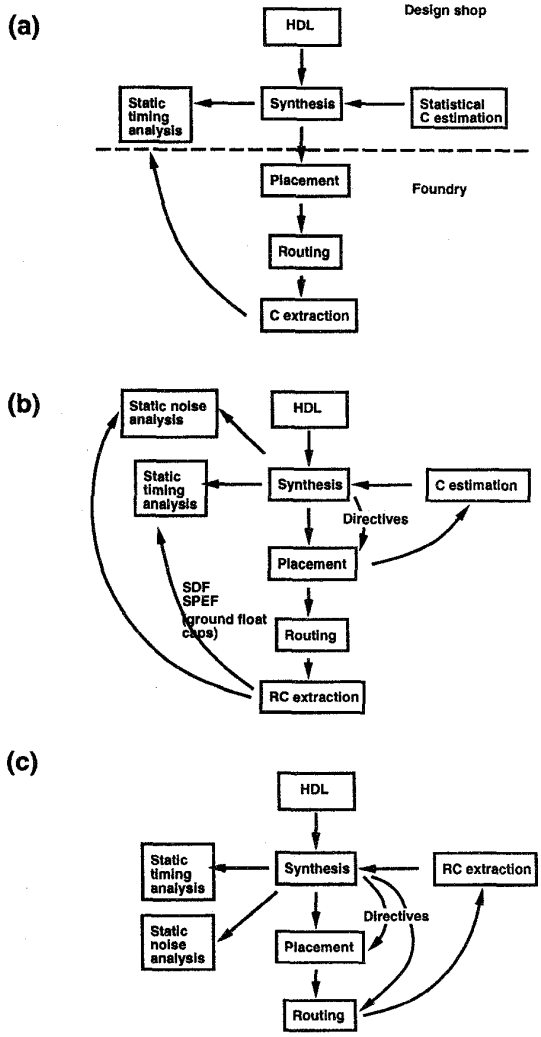

Figure 1: ASIC design flows: (a) Traditional flow with a strict partitioning of logical and physical. (b) State-of-the-art flow which includes static noise analysis and the effects of coupling capacitance and resistance in post-placement verification but does not automate RC delay optimization. Timing-driven placement may work to control capacitance along critical paths. (c) Future design flow in which synthesis, interacting with both static timing and static noise analysis, directs both placement and routing.

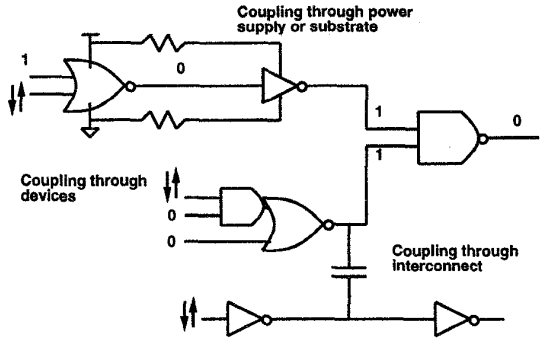

Figure 2: Switching events interfere with static signals because of coupling through the interconnect, through the transistors, through the substrate or nwell, or through the power supply.

analysis, as part of the design process allows the use of more aggressive circuit designs in the library (e. g. pass-gate input latches). In the design flow of Figure 1(b), coupling capacitance can be estimated based on a fraction of the total loading in the early phases of the design. In the more advanced flow of Figure 1(c), more detailed RC extraction information and more sophisticated interconnect models will be available to guide noise analysis.

3 Challenging fixed library approaches

The increasing complexity of timing[12], power, and noise rules to handle deep submicron effects and the rapid pace of technology migration are challenging the precharacterized ASIC library approach to design. Analysis tools which function at the transistor-level but still preserve the cell level of abstraction have a unique opportunity in timing, power, and noise analysis. These tools, in effect, perform very accurate rule characterization "on the fly" and can instantly adjust to technology changes.

In the absence of precharacterized library rules, the opportunity also exists to move away from fixed libraries entirely, allowing even greater flexibility in performance tuning and optimization. Cell layouts can be generated[13] automatically before placement and routing. These automated layouts also allow rapid technology migration or adaptation to new design rules. The availability of a continuously-parameterizable static CMOS library, for example, creates new possibilities in how one thinks about sizing. In an approach that closely follows the theory of logical effort[14], a gate can be parameterized by quantities directly related to delay rather than physical size[2]. This allows "automated" retuning to changes in interconnect load and quick identification of optimal sizing.

4 Beyond static CMOS

The more transistor-level approach to turnkey ASIC design that includes a tighter integration between synthesis and static timing and static noise analysis creates several new opportunities. Even within the context of conventional static CMOS logic, the possibility of transistor-level synthesis offers new options for restructuring[15]. More importantly,

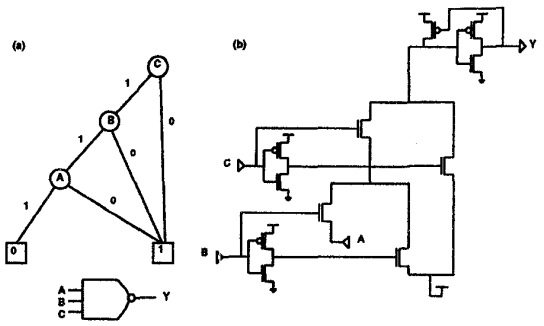

Figure 3: Pass-gate implementation of a three-input NAND gate: (a) binary-decision diagram representation of the logic, and (b) one possible pass-gate implementation.

more complex circuit styles such as transparent latch design, domino logic, or pass-transistor logic become possible in a synthesis context.

Pass-transistor logic synthesis is an attractive alternative for building certain logic functions, such as selectors, multiplexors, and XORs. Pass transistor mapping[16, 17] can follow immediately from a binary-decision diagram (BDD)[18] representation of the logic as shown in Figure 3 for the case of a three-input NAND. Restoring logic gates, such as the inverter with half-latch in Figure 3(b), are required every two or three level of pass gates for performance and to restore full-rail logic values. Some partitioning schemes are based on analyzing a monolithic BDD[16] or partitioning before BDD analysis[17]. Promising new approaches find analogy with LUT FPGA mapping[19]. The use of smart-body contacts in partially-depleted SOI[20] gives an even bigger performance advantage to pass-transistor logic in this technology[21].

Domino logic synthesis[22] is another alternative for high-performance design. The success of this technique relies on accurate timing and noise analysis and on combining domino logic and static CMOS successfully. Domino logic comes with a considerable increase in noise sensitivity[23]. In addition, there are a significant number of additional timing checks required to ensure correct functioning. For the domino gate of Figure 4(a), for example, there are four additional timing checks which must be performed as shown in Figure 4(b):

- The dynamic node must fall before the falling edge of the clock (setup).
- The data node must fall before the rising edge of the clock (setup).
- The dynamic node must rise before the rising edge of the clock (setup).
- The falling edge of the data node must be held until after the dynamic node falls (hold).

Domino logic must also be noninverting, requiring that logic be restructured[24] to push inverters forward or back, so that the inversions can be incorporated into latches or static logic. Because of this complexity in timing and logic structure, there have been efforts to use self-timed techniques to work around these[25], unfortunately at a considerable cost in performance.

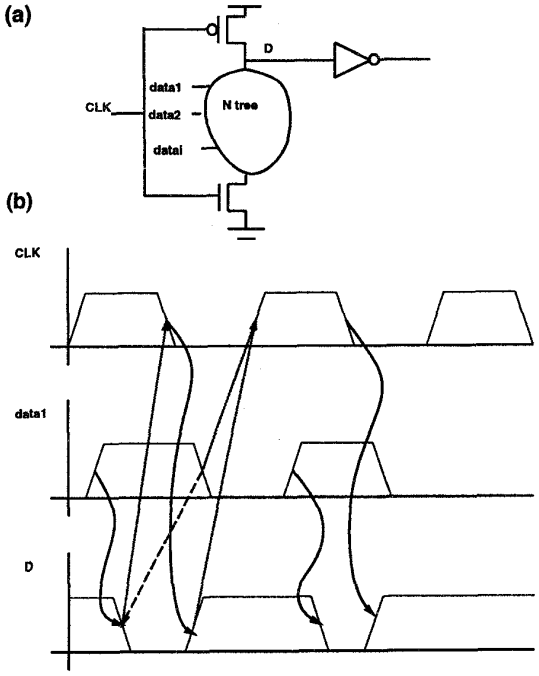

Figure 4: Timing checks for a standard "footed" domino stage: (a) Circuit topology for the stage. D denotes the dynamic node of the gate, and CLK is the clock. (b) Example waveforms for the data inputs, clock, and dynamic nodes. Dotted arrows denote setup test. Dashed arrow denotes hold test. Solid arrows denote delays.

5 Conclusions

Technology trends are inevitably forcing closer interaction between synthesis and physical design. At the same time, both static timing and static noise analysis are essential engines for optimization and verification. The increasing complexity of rule-based libraries is going to create a push for analysis tools which understand transistors. This will create opportunities at the high end for more transistor-level ASIC design.

References

[1] Shervin Hojat and Paul Villarrubia. An integrated placement and synthesis approach for timing closure of Power-PC microprocessors. In *Proceedings of the International Conference on Computer Design*, pages 206 –210, 1997.

[2] K. L. Shepard, S. Carey, E. Cho, B. Curran, R. Hatch, D. Hoffman, S. McCabe, G. Northrop, and R. Seigler. Design Methodology for the G4 S/390 Microprocessors. *IBM Journal of Research and Development*, 21(4/5):515 – 548, 1997.

[3] Standard Delay Format. Technical report, Open Verilog International, 1998. Version 3.0.

[4] IEEE DASC P1481 Draft Standard for Integrated Circuit Delay and Power Calculation. Technical report, 1998.

[5] L. P. P. P. van Ginneken. Buffer placement in distributed RC-tree networks for minimal Elmore delay. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 865 – 868, 1990.

[6] S. S. Sapatnekar. Wiring sizing as a convex optimization problem: exploring the area-delay tradeoff. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15:1001 – 1011, 1996.

[7] J. J. Cong and Kwok-Shing Leung. Optimal wiresizing under Elmore delay model. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14:321 – 336, 1995.

[8] K. L. Shepard, V. Narayanan, P. C. Elmendorf, and Gutuan Zheng. Global Harmony: Coupled Noise Analysis for Full-Chip RC Interconnect Networks. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, 1997.

[9] K. L. Shepard and V. Narayanan. Noise in deep submicron digital design. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 524–531, San Jose, CA, November 1996.

[10] K. L. Shepard and V. Narayanan. Conquering noise in deep submicron digital design. *IEEE Design and Test of Computers*, pages 51 – 62, January/March 1998.

[11] H. H. Chen. Minimizing chip-level simultaneous switching noise for high-performance microprocessor design. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 4, pages 544 – 547, 1996.

[12] F. Dartu, N. Menezes, J. Qian, and L. T. Pillage. A gate-delay model for high-speed cmos circuits. In *Proceedings of the 31st Design Automation Conference*, pages 576–580, 1994.

[13] Jeffrey L. Burns and Jack A. Feldman. C5M – a control-logic layout synthesis system for high-performance microprocessors. *IEEE Trans. CAD*, 17:14 – 23, January 1998.

[14] Ivan Sutherland and Robert Sproull. The theory of logical effort: Designing for speed on the back of an envelope. In *Advanced Research in VLSI*, UC Santa Cruz, 1991.

[15] S. Gavrilov, A. Glebov, S. Pullela, S. C. Moore, A. Dharchoudhury, R. Panda, G. Vijayan, and D. T. Blaauw. Library-less synthesis for static CMOS combinational logic circuits. In *Proceedings of the International Conference on Computer-Aided Design*, pages 658 – 662, 1997.

[16] Kazuo Yano, Yasuhiko Sasaki, Kunihito Rikino, and Koichi Seki. Top-down pass-transistor logic design. *IEEE Journal of Solid-State Circuits*, 31:792 – 803, 1996.

[17] Premal Buch, Amit Narayan, A. Richard Newton, and A. Sangiovanni-Vincentelli. Logic synthesis for large pass transistor circuits. In *Proceedings of the International Conference on Computer-Aided Design*, pages 663 – 670, 1997.

[18] Randal E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Trans. CAD*, 35:677–691, 1986.

[19] Jason Cong and Yuzheng Ding. An optimal technology mapping algorithm for delay optimization in lookup-table based fpga designs. In *Proceedings of the International Conference on Computer-Aided Design*, pages 48 – 53, 1992.

[20] C. T. Chuang, P.-F. Lu, and C. J. Anderson. SOI for digital CMOS VLSI: Design considerations and advances. *Proceedings of the IEEE*, 86:689 – 720, 1998.

[21] T. Fuse, Y. Oowaki, M. Terauchi, S. Watanabe, M. Yoshimi, K. Ohuchi, and J. Matsunaga. 0.5 V SOI CMOS pass-gate logic. In *Proceedings of the IEEE International Solid-State Circuits Conference*, pages 88 – 89, 1996.

[22] M. Hofmann and A. R. Newton. A domino CMOS logic synthesis system. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 411 – 414, 1985.

[23] Jacobus A. Pretorius, Alex S. Shubat, and C. Andre T. Salama. Charge redistribution and noise margins in domino CMOS logic. *IEEE Transactions on Circuits and Systems*, CAS-33(8):786 – 793, 1986.

[24] Ruchir Puri, Andrew Bjorksten, and Thomas E. Rosser. Logic optimization by output phase assignment in dynamic logic synthesis. In *Proceedings of the International Conference on Computer-Aided Design*, pages 2 – 8, 1996.

[25] Gin Yee and Carl Sechen. Dynamic logic synthesis. In *Proceedings of the IEEE Custom Integrated Circuits Conference*, pages 15.3.1 – 15.3.4, 1997.