CMOS Circuit Design for Minimum Dynamic Power and Highest Speed*

Tezaswi Raja
Rutgers University, Dept. of ECE
Piscataway, NJ 08854, USA
tezaswir@caip.rutgers.edu

Vishwani D. Agrawal†
Rutgers University, Dept. of ECE
Piscataway, NJ 08854, USA
vishwani02@yahoo.com

Michael L. Bushnell
Rutgers University, Dept. of ECE
Piscataway, NJ 08854, USA
bushnell@caip.rutgers.edu

Abstract–*A new low-power design method produces CMOS circuits that consume the least dynamic power at the highest speed permitted under the technology constraint. A gate is characterized by an inertial delay and separate delays between its inputs and output. The technology constraint, related to feasible ranges of lengths and widths of transistors, is specified by a parameter u_b. It is the upper bound on the difference between the input to output delays corresponding to any pair of inputs of a gate. We formulate a linear program (LP) whose size is proportional to the circuit size. This LP determines the inertial delay as well as input to output delays for each gate of the circuit with the given u_b, such that all glitches are eliminated and the overall delay of the circuit is minimized. Because of the additional flexibility in specifying gate delays, the glitch suppression is guaranteed without any delay buffers. Hence this design consumes less power than those designed by other methods. We designed the circuit c1355 with 46% of the original power dissipation compared to a reference design. A previously published method, that characterizes each gate with a single delay, produced a c1355 circuit consuming 58% of the original power. Both low-power circuits had the same overall delay. The previous design required 224 delay buffers, whereas the new design needed none.*

1. Introduction

The power dissipated in a CMOS circuit consists of *dynamic power, leakage power* and *short-circuit power* components. The topic of this paper is the reduction of *dynamic power*. When an input vector is applied to the primary inputs (PI), the minimum power requirement for each gate output is to produce either 0 or 1 transition. However, in reality there may be many more transitions due to *glitches* or *hazards*, caused by the *differential delays* of paths leading to the gate inputs.

Dynamic power of a circuit is reduced by eliminating some or all glitches. The principal idea of a glitch reduction technique is to find delay assignments for all gates in the circuit so as to reduce the differential path delays at gate inputs. Published techniques are *balanced delay method* [7, 11, 17, 18, 21], *hazard filtering method* [1, 25], *transistor sizing* [5, 6, 10, 12, 22, 23, 26], *gate sizing* [3, 4, 24], and *linear programming (LP) techniques* [2, 19, 20].

This paper falls under the category of LP techniques with three major contributions. The first is the realization that gates can be designed with different input-output delays along different IO paths through the gate, even though these delays are not independent variables. The second contribution is the formulation of a linear program that incorporates this information into the constraint set and comes up with a delay assignment. A third contribution is the elimination of the buffer insertion which is the main drawback of many previous LP techniques.

We outline prior LP techniques and state their drawbacks in Section 2. The new LP formulation is given in Section 3. Results are tabulated in Section 4 and Section 5 describes the transistor level realization of the delay assignment.

2. Prior Work

We examine the main ideas in the path enumeration technique and the linear constraint set method.

2.1. Path Enumeration Method

Agrawal *et al.* [2] show that for a correct operation with *minimum transient energy* (MTE) consumption,

*This research is supported in parts by the NSF grant no. CCR-9988239

†Presently with ECE Dept., Auburn University, Alabama 36849, USA.

every CMOS gate in the circuit must produce no more than one event (signal change) at its output during a transition interval. The *transition interval* is defined as the interval after the primary inputs change and during which all signals attain their steady state. They prove that if the new logic output is different from the old value then only a single transition can achieve the correct result. Assuming a single delay variable per gate, Agrawal *et al.* [2] eliminate all glitches by making the gate delay exceed the differential path delay at the gate inputs. They find that,

1. If the overall circuit delay is allowed to increase then an MTE design is always possible by adjusting the output delays of the gates. This MTE design does not require buffer insertion and hence is the lowest dynamic power design.
2. If the overall delay is bounded then an MTE design is not guaranteed without the insertion of delay buffers.

Agrawal *et al.* [2] describe an LP model to generate constraints for hazard filtering, keeping the overall delay within the specified limits. Their constraint set size is proportional to the number of paths in the circuit. Since the number of paths terminating at a gate increases exponentially, the constraint set also increases exponentially with circuit size. This high complexity prevents the model from optimizing large circuits. For example the circuit c880 needs 6.9 million path constraints, which cannot be tackled by many linear programming tools.

2.2. Linear Constraint Set Method

Raja *et al.* [19, 20] have described a way of reducing the complexity of the constraint set from exponential to linear in circuit size. In addition to the inertial delay, they introduce two new variables per gate in the LP, *viz.*, earliest time of arrival of the signal at a gate and the latest signal arrival time. This approach is similar to the timing verification algorithm described by Hitchcock [14, 15]. These two variables define the *timing window* in which the signal can change at the output of the gate. The LP constraint set then forces the inertial delay to be greater than the timing window at the output of the gate. The authors prove that their formulation is equivalent to the path enumeration model although constraint set is linear in circuit size. For example, the circuit c880, requires only 3,611 constraints by this method.

Chuang *et al.* use a similar set of delay variables to simultaneously optimize the area and timing of a standard-cell design [8, 9].

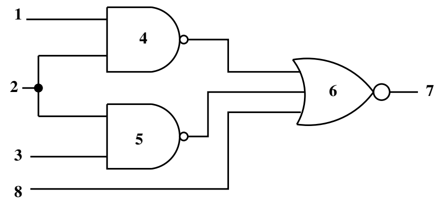

Figure 1: A combinational circuit.

2.3. Shortcomings of Above Methods

In both the methods described above, the problem is with the buffer insertion. These buffers, although they do not alter the signal value, consume switching and short-circuit power during operation and leakage power even when not in operation. When many buffers are inserted, the amount of power saved by buffer insertion is reduced by the power consumed by them. Thus, the power saving method should not introduce unnecessary elements into the circuit.

Agrawal *et al.* [2] prove a theorem which states that, in general, a circuit cannot be designed for minimum dynamic power, i.e., without inserting delay buffers, unless the overall delay of the circuit is allowed to increase. Thus, the lowest-power version of the circuit will be slower than the original circuit. This is undesirable since a designer would prefer the low-power version to be as fast as possible. The limitation of these methods stems from the conventional design of CMOS gates where only the output delay of a gate can be varied.

In this paper, we propose a method that will redesign the circuit with minimum dynamic power without the insertion of delay buffers and with least reduction in speed. This method will produce the minimum dynamic power circuit with the fastest speed possible for the given CMOS technology.

3. New Formulation

This section introduces the new formulation that we propose. We consider the example of a simple circuit to explain the formulation.

Consider the combinational circuit shown in Figure 1. Traditionally, for the purpose of gate sizing the circuit is generally viewed with each gate having a single *inertial delay* and all input-output (IO) paths through the gate are assumed to have the same delay. We redefine the gate delay. A gate can be viewed as having one *basic inertial delay* and a set of transport delays for the IO paths running through the gate. This is illustrated in Figure 2.

Each gate can be assumed to have a basic delay variable with input delay elements at the inputs of

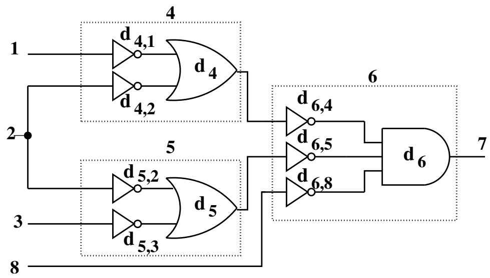

Figure 2: Delay model for the circuit of Figure 1.

the gate. We assert that *these input delay elements are only for analysis purposes and are not actual extra components in the circuit.* The inertial delay and input delays of a CMOS gate are not independent. In the LP we treat them independent variables, which are bounded by a feasibility constraint so that the delays can be realized in practice.

3.1. Linear Program

Consider the delay model of Figure 2. Now the linear program can be written as follows.

3.1.1 Variables

- Basic inertial delay of the gate: d_4, d_5, d_6.
- Gate Input Delay: $d_{i,j}$ which is the extra delay on the path from the fanin gate j to gate i. For instance $d_{4,1}$ is the extra delay of the path through gate 4 while arriving from PI 1. This models the differences in delays of various IO paths through the gate. Its minimum value is 0.
- T_i is the latest time of signal change at the output of gate i.
- t_i is the earliest time of signal change at the output of gate i.
- $T_{i,j}$ is the latest time of signal change at the output of delay element whose delay is $d_{i,j}$.
- $t_{i,j}$ is the earliest time of signal change at the output of delay element whose delay is $d_{i,j}$.

3.1.2 Constraints on Delays

Following constraints set the lower and upper bounds on the variables:

- Lower bound on gate inertial delays are set to 1. The actual value of this time unit will depend on the specific technology used.
- Lower bound on gate input delays are set to 0.
- We also set an upper bound u_b on the gate input delays (see Subsection 3.1.6).

3.1.3 Glitch Suppression Constraints

These constraints ensure that the timing window for signal transitions at every gate output does not exceed the inertial delay [19,20]. Consider gate 6 in Figure 2. The constraints for it are given as:

$$t_6 \leq t_{6,4} + d_6; \quad t_6 \leq t_{6,5} + d_6; \quad t_6 \leq t_{6,8} + d_6;$$
$$T_6 \geq T_{6,4} + d_6; \quad T_6 \geq T_{6,5} + d_6; \quad T_6 \geq T_{6,8} + d_6;$$
$$d_6 \geq T_6 - t_6$$

and the constraints for an IO delay element $d_{6,4}$ are

$$t_{6,4} \leq t_4 + d_{6,4}; \quad T_{6,4} \geq T_4 + d_{6,4}$$

3.1.4 Maxdelay Constraints

For every PO we have: $T_7 \leq maxdelay$.

3.1.5 Objective Function

The following objective function makes the circuit as fast as possible:

$$Minimize \quad maxdelay$$

3.1.6 Feasibility Constraints

The main issue here is the extra upper bound added to the delay of IO elements. The idea behind this formulation is to design a gate that can have different delays along different IO paths through it. This is possible but there are limitations (see Subsection 5). Given a CMOS technology transistor lengths and widths, that control the delay of a gate, can be varied within limited ranges. Thus the amount of difference in delay one can get from two paths through a gate is limited. Hence the extra delay added to the gates by way of IO delay elements must be within these feasibility ranges. We assume a certain feasibility range over which one can vary the different delays through a single gate. We call this the maximum differential delay upper bound u_b.

Definition: *Gate input differential delay upper bound u_b:* The gate input delay upper bound is a measure of the maximum difference in delay for any two IO paths through the gate, that can be designed in a particular technology. If unconstrained, the program may give gate input delays that differ by large amounts as its solution. However, every technology has a limit to the amount of flexibility that the designer is allowed. This limit of flexibility shows the feasibility of designing the gate input delays for the technology used at the transistor and layout levels. Hence we call this the *feasibility condition.* Now the feasibility constraints for gate 6 would become

$$d_{6,4} \leq u_b; \quad d_{6,5} \leq u_b; \quad d_{6,8} \leq u_b;$$

This allows the gate input delay to be varied up to a value of u_b by the program. This value is a design parameter and is specific to the design technology in which the circuit is being designed. As explained in Subsection 3.1.6, u_b can be determined by the delay analysis of actual gate layouts. Given that feasibility value we can use the linear program to design the *lowest power consuming yet fastest realizable circuit.*

4. Results

The LP was written for the ISCAS'85 benchmark circuits and solved using AMPL [13]. The resulting delay assignments are used in the delay simulator for the power estimation analysis as described by Hsiao *et al.* [16]. We present our results in this section.

4.1. Feasible Gate Differential Delay Upper Bound

The gate delay upper bound (u_b) is a measure of the flexibility we have in terms of designing a gate with different IO path delays through the gate. We have run the LP formulation on the ISCAS'85 benchmark circuits for different feasibility bounds and the results are shown in Figure 3. Each curve in the figure corresponds to a different circuit. We can see that as the feasibility upper bound is increased we have lower *maxdelay* and hence a faster circuit.

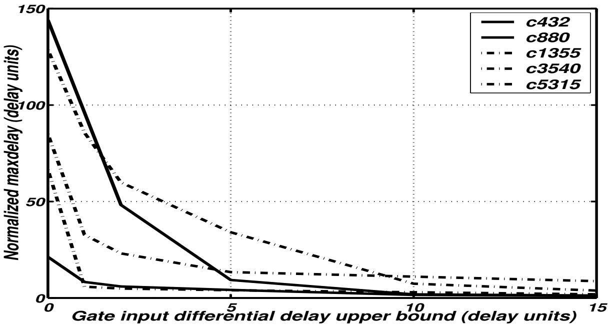

Figure 3: Normalized *maxdelay* versus u_b. The *maxdelay* is normalized to the fastest possible circuit design, i.e., without altering delays along the critical path.

4.2. Power Savings

The LP gives the optimal set of delays for the gates. The circuits were then simulated using a variable delay simulator. The results on some of the benchmark circuits are shown in Table 1.

The vectors used are the compacted ATPG test vectors for each circuit. All gates in the unoptimized circuit are assumed to have a delay of one unit. This is the smallest possible delay realizable at the physical level in that technology. The *maxdelay* and u_b are shown in the same delay units. The normalized delay is the critical path delay normalized to the *maxdelay* of the fastest possible implementation of the circuit. The power calculations are done only for certain u_b's for each circuit because of the limitation of the simulator to handle delays larger than 100. These are in cases where the u_b is too tight and the gates have very high delays. These cases can be ignored as a designer would not want to slow the circuit down 15-20 times, which is the case in these designs. We compare the preliminary results of this work with the *linear constraint set* results given by Raja *et al.* [19, 20] in Table 1.

The power savings are much better for the proposed method. For example, for the circuit c432 the power saving increased by 24% for the same *maxdelay.* This is because 95 buffers were inserted on non-critical paths by the previous method.

The technique described in this paper differs from the previous techniques [19, 20] in two significant ways. Consider Figure 4. The previous technique finds the lowest power consuming circuit for a given *maxdelay.* The solution curve at the point shown as $u_b = 0$ is the circuit with no buffers inserted in the circuit. Now to increase the speed of the circuit we need to insert buffers, but this increases the power also as the buffers consume power. This is shown by the increase in the power consumed by buffers in the curve. In the technique described here, if we use conventional gate design, i.e., $u_b = 0$, we get the same buffer-less design as shown. But if we increase the u_b the designs get progressively faster, but since we do not add any new buffers the power is still the minimum dynamic power possible for the circuit. As the upper bound u_b is increased to higher levels, there will be a design that will have both the highest possible speed and the lowest possible power. This is shown by the point $u_b = \infty$ in Figure 4. Thus, the new technique gives the lowest power consuming circuit with the fastest speed permitted by the technology.

Table 1: Dynamic power dissipation in ISCAS'85 benchmark circuits for proposed design method.

Circuit	Proposed Method						Raja *et al.* [19, 20]	
	Avg. Norm. Power		No. of	*maxdelay*	Norm.	u_b	Avg. Norm.	No. of
	Unoptimized	Optimized	Vectors	(delay)	Delay	(delay)	Power	Buffers
c432	1.0	0.52	56	71	4.17	5	-	-
	1.0	0.49	56	27	1.58	10	0.62	66
	1.0	0.48	56	17	1.00	15	0.72	95
c499	1.0	0.70	54	34	2.26	0	0.70	0
	1.0	0.74	54	20	1.33	2	-	-
	1.0	0.75	54	15	1.00	5	0.91	48
c880	1.0	0.48	78	45	1.50	10	0.68	34
	1.0	0.47	78	30	1.00	15	0.68	62
c1355	1.0	0.47	87	96	2.08	5	-	-
	1.0	0.46	87	71	1.54	10	0.57	192
	1.0	0.46	87	46	1.00	15	0.58	224

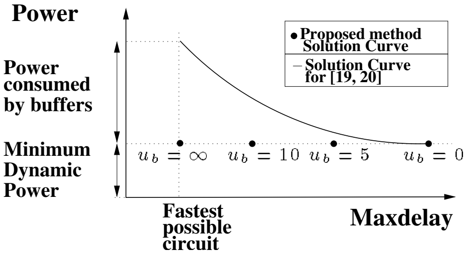

Figure 4: Power *vs.* Maxdelay curves

5. Transistor Level Design

The delays given by the linear program (LP) are implemented at the transistor level. This means that we need to design the gates with different IO path delays. We design a gate by appropriate sizing of transistors that affect the particular IO path according to the specified delay.

Consider a CMOS NAND gate with inputs 1 and 2 and output 3. The gate has two IO paths (1,3) and (2,3). The IO path delay from gate 1 to gate 3 can be varied by changing the transistors connected to input 1. This will result in changing the input capacitance associated with input 1 without *much* affecting the path delay through input 2. We have designed a NAND gate in $0.25\mu m$ technology using Cadence tools. We varied the width and length of a single transistor pair and measured the difference in delay through both IO paths of the gate using Spectre. The result is shown in Figure 5. The graph shows the delay difference of the two paths keeping one transistor pair constant and varying the width and lengths of the transistor pair corresponding to the other input. As shown, we have achieved a differential delay of up to 400 ps. In this technology the fastest gate design is about 50 ps. Hence we have achieved a design of $u_b = 8$ for the NAND gate.

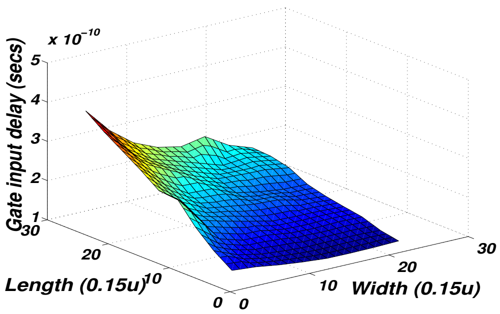

Figure 5: Delay plot of a CMOS NAND gate, by varying the sizes of transistor pair of input 1 and keeping the sizes of the pair in input 2 constant.

6. Conclusion

A given CMOS circuit is optimized for minimum dynamic power when its operation contains no glitches. In the design presented, all glitches are suppressed by adjusting the gate delays. However, to keep the total number of signal transitions at the lowest level, no gates or delay buffers are inserted. This restriction, in the previous designs, would have forced an increase in the overall circuit delay. In the present method, the overall circuit delay is minimized through a novel design of CMOS gates in which inputs of a gate can have different delays. The amount of differential input delays is restricted by the available range of transistor sizes in the CMOS technology. Thus, for a given technology the optimization procedure produces a circuit with no glitches and the minimum possible delay. Experiments show that, in comparison to a previous design, on an average an extra 20% reduction in dynamic power is possible without any speed reduction. Routing delays and manufacturing tolerances of gate delays are not accounted for in the current experiments. These and other practical aspects are currently under investigation.

References

[1] V. D. Agrawal, "Low Power Design by Hazard Filtering," in *Proc. of the International Conference on VLSI Design*, Jan. 1997, pp. 193–197.

[2] V. D. Agrawal, M. L. Bushnell, G. Parthasarathy, and R. Ramadoss, "Digital Circuit Design for Mimimum Transient Energy and Linear Programming Method," in *Proc. of the International Conference on VLSI Design*, Jan. 1999, pp. 434–439.

[3] M. Berkelaar, P. Buurman, and J. Jess, "Computing Entire Area/Power Consumption versus Delay Trade-off Curve for Gate Sizing Using a Piecewise Linear Simulator," *IEEE Transactions on Circuits and Systems*, vol. 15, no. 11, pp. 1424–1434, Nov. 1996.

[4] M. Berkelaar and E. Jacobs, "Using Gate Sizing to Reduce Glitch Power," in *Proc. of the ProRISC Workshop on Circuits, Systems and Signal Processing*, (Mierlo, The Netherlands), Nov. 1996, pp. 183–188.

[5] M. Berkelaar and J. A. G. Jess, "Transistor Sizing in MOS Digital Circuits with Linear Programming," in *Proc. of the European Design Automation Conference*, (Mierlo, The Netherlands), Mar. 1990, pp. 217–221.

[6] M. Borah, M. J. Irwin, and R. M. Owens, "Minimizing Power Consumption of Static CMOS Circuits by Transistor Sizing and Input Reordering," in *Proc. of the International Conference on VLSI Design*, Jan. 1995, pp. 294–298.

[7] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design.* Boston: Kluwer Academic Publishers, 1995.

[8] W. Chuang, S. S. Sapatnekar, and I. N. Hajj, "A Unified Algorithm for Gate Sizing and Clock Skew Optimization to Minimize Sequential Circuit Area," in *Proc. of the International Conference on Computer-Aided Design*, Nov. 1993, pp. 220–223.

[9] W. Chuang, S. S. Sapatnekar, and I. N. Hajj, "Timing and Area Optimization for Standard Cell VLSI Circuit Design," *IEEE Transactions on Computer-Aided Design*, vol. 14, no. 3, pp. 308–320, Mar. 1995.

[10] S. Datta, S. Nag, and K. Roy, "ASAP: A Transistor Sizing Tool for Area, Delay and Power Optimization of CMOS Circuits," in *Proc. of the IEEE International Symposium on Circuits and Systems*, May 1994, pp. 61–64.

[11] M. S. Elrabaa, I. S. Abu-Khater, and M. I. Elmasry, *Advanced Low-Power Digital Circuit Techniques.* Boston: Kluwer Academic Publishers, 1997.

[12] J. P. Fishburn and A. E. Dunlop, "TILOS: A Posynomial Programming Approach to Transistor Sizing," in *Proc. IEEE International Conf. Computer-Aided Design*, Nov. 1985, pp. 326–328.

[13] R. Fourer, D. M. Gay, and B. M. Kernighan, *AMPL: A Modeling Language for Mathematical Programming.* South San Francisco, California: The Scientific Press, 1993.

[14] R. B. Hitchcock Sr., "Timing Verification and the Timing Analysis Program," in *Proc. of the 19th Design Automation Conf.*, June 1982, pp. 594–604.

[15] R. B. Hitchcock Sr., G. L. Smith, and D. C. Cheng, "Timing Analysis of Computer Hardware," *IBM Journal of Research & Development*, vol. 26, no. 1, pp. 100–105, Jan. 1982.

[16] M. Hsiao, E. M. Rudnick, and J. H. Patel, "Effects of Delay Model in Peak Power Estimation of VLSI Circuits," in *Proc. of the International Conference on Computer-Aided Design*, Nov. 1997, pp. 45–51.

[17] J. Monteiro and S. Devadas, *Computer-Aided Design Techniques for Low Power Sequential Logic Circuits.* Boston: Kluwer Academic Publishers, 1997.

[18] J. M. Rabaey and M. Pedram, *Low Power Design Methodologies.* Boston: Kluwer Academic Publishers, 1995.

[19] T. Raja, "A Reduced Constraint Set Linear Program for Low Power Design of Digital Circuits," Master's thesis, Rutgers University, Dept. of ECE, Piscataway, New Jersey, May 2002.

[20] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Minimum Dynamic Power CMOS Circuit Design by a Reduced Constraint Set Linear Program," in *Proc. of the International Conference on VLSI Design*, Jan. 2003, pp. 527–532.

[21] K. Roy and S. C. Prasad, *Low-Power CMOS VLSI Circuit Design.* New York: Wiley Interscience Publication, 2000.

[22] C. V. Schimpfle, A. Wroblewski, and J. A. Nassek, "Transistor Sizing for Switching Activity Reduction in Digital Circuits," in *Proc. of the European Conference on Theory and Design*, Aug. 1999.

[23] J. M. Shyu, A. L. Sangiovanni-Vincntelli, J. P. Fishburn, and A. E. Dunlop, "Optimization-based Transistor Sizing ," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. "400–409", Apr. 1988.

[24] V. Sundararajan, S. Sapatnekar, and K. Parhi, "Fast and Exact Transistor Sizing Based on Iterative Relaxation," *IEEE Transactions on Computer Aided Design of Circuits and Systems*, vol. 21, 2002.

[25] S. H. Unger, *Asynchronous Sequential Switching Circuits.* New York: Wiley-Interscience, 1969.

[26] A. Wroblewski, C. V. Schimpfle, and J. A. Nassek, "Automated Transistor Sizing Algorithm for Minimizing Spurious Switching Activities in CMOS Circuits," in *Proc. of the IEEE International Symposium on Circuits and Systems*, May 2000.