Power Compiler : A Gate-Level Power Optimization and Synthesis System

Benjamin Chen Ivailo Nedelchev
bchen@synopsys.com ivailo@synopsys.com
Synopsys, Inc
700 E. Middlefield Road
Mountain View, CA. 94043-4033

Abstract

Commercial synthesis tools traditionally perform timing and area optimization, however power reduction is rapidly becoming an equally important design goal. Recent research in power optimization has produced several algorithms, however, each algorithm is focused on one aspect of the whole power equation.

This paper describes a commercial tool capable of optimizing power at the gate-level in addition to performing area and timing optimization. A power analysis engine that models all aspects of power consumption is integrated into the optimization tool so that all aspects of power are considered. Experimental results show an average 11.46% reduction on industrial circuits with a peak reduction of 66.62%. All delay constraints are met and an average 9.41% increase in area is observed.

1. Introduction

Reducing power consumption is both an imperative and a daunting challenge for today's ASIC and IC designers. Silicon technology advances have made it possible to pack hundreds of thousands of transistors on a single chip. In addition, performance goals require high clock speeds posing difficult power dissipation and distribution problems. Further complicating the mix is the rapidly increasing demand for power-sensitive applications like high-performance computer systems; portable, battery-operated computers, medical devices and telecommunications equipment.

In response, designers are moving to incorporate power considerations into all phases of their design flow. In methodologies that dictate the use of logic synthesis [3] tools on some portion of the design, such tools must also consider power consumption when performing design trade-offs. Until recently, commercial synthesis tools have mainly focused on timing and area optimization. Power reduction of a circuit, however, can come at the expense of these design goals, hence must operate within these design constraints and perform tradeoffs between them.

1.1 Components of Power

Average power dissipation in digital CMOS circuits can be roughly described by the following equation [1], [10]:

$$P_{average} = \frac{1}{2} \cdot C \cdot V^2 \cdot f + I_{short} \cdot V \cdot f + I_{leakge} \cdot V + I_{static} \cdot V$$

The first term represents power due to switching of the circuit (*switching power*) where C is the load being switched, V is the voltage swing (usually the supply voltage), and f is the number of 0->1 and 1->0 transitions. The second term represents power dissipated when both NMOS and PMOS transistors are active (*short power*) creating a short circuit current I_{short}. The amount of this power is proportional to the number of times the circuit switches f and depends on the capacitive loading at the output of the CMOS gate. The third term represents power due to leakage current $I_{leakage}$ (*leakage power*) that can exist when gate voltages are just below their threshold values. The last term represents static power dissipated when a circuit is inactive where static currents I_{static} (*static power*) occur due to sub-supply voltages at gates and the reversed-biased p-n junction between diffusion and substrate.

Switching power usually accounts for 70 to 90 percent of the power dissipation. Internal power typically accounts for 10 to 30 percent, while leakage and static power represent less than 1 percent of power dissipated. Leakage and static power, however, can easily become the dominant source of power dissipation in applications where the circuit is predominantly inactive and will become more of an issue as supply voltages are scaled down.

1063-6404/97 $10.00 © 1997 IEEE

1.2 Gate-Level Power Optimization

Circuit Characteristic	Power Component
Gate dimensions	$C, I_{short}, I_{leakage}, I_{static}$
Net switching	F
Net transition time	I_{short}
Net capacitance	C, I_{short}

Table 1. Gate-level properties related to power.

From a gate-level perspective, the above components of power are determined by the following circuit characteristics: gate dimensions, net switching, net transition time, net capacitance.

Table 1 associates the circuit characteristics with the component of power. Gate-level transformations that focus on the above circuit characteristics can reduce the power of the design. Such transformations have been successfully implemented in the first commercially available power optimization synthesis tool Power Compiler™. Power Compiler is built on the synthesis framework of Design Compiler™. This feature allows power optimization to be performed within other optimization goals such as timing and area.

2. Power Model

Accurate gate-level power estimation is required for gate-level power optimization. All components of power as described in section 1.2 must be modeled, since reducing one power factor can increase another. Power Compiler obtains its power estimates from Design Power™[8]:

- *Switching power*: Pin and wire capacitance information ae available in the synthesis technology libraries. Transition count information is expressed by *toggle rates* which is defined to be the number of transitions per unit time. Toggle rates can come either from Design Power's BDD-based probabilistic estimation algorithm [6] or from a gate-level simulation.
- *Internal or short circuit power*: The definition of internal and short-circuit power in section 1.2 easily applies to CMOS inverters. For gates with multiple stages of CMOS transistors, the definition is extended to be all the power consumed within a gate. The internal power model is a nonlinear model based on technology provided by the ASIC vendor as an internal energy look-up table derived from SPICE characterization runs for each gate in the technology library. This energy table is indexed by the cell's input edge rates and output loads to produce an energy value which is then multiplied by the toggle rate of the output of the gate to obtain an estimate of power consumed within the gate.
- *Static or leakage power*: The static/leakage power model is a single constant value for the cell specified by the ASIC vendor.

3. Power Cost Function

Power Compiler optimizes a design according to a set of constraints that are defined by the user and the technology library. These constraints are used to determine the cost of a given design which guides the optimization algorithms. The cost measures the extent to which a constraint has been met. If a constraint has been satisfied, the corresponding cost will be zero.

The application of optimization algorithms is divided into two phases. The two phases are distinguished by the cost functions being optimized. The first phase uses an optimization only cost function, while the second phase adds a design rule cost.

The optimization cost function is shown below in order of importance. Some components might not be active on a given design:

1. Maximum delay
2. Minimum delay
3. Maximum dynamic power
4. Maximum leakage power
5. Maximum area

A prioritized cost function means that timing constraints will not be violated to save power, but available timing slack can be consumed if power can be reduced.

In the design rule phase, the cost function is identical to the optimization cost function except that a design rule cost is added as the most important cost. Design rule constraints reflect technology-specific restrictions that must be met for a functional design, such as the maximum signal transition time for nets.

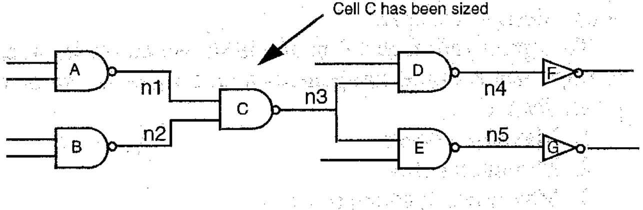
Cost function components are evaluated independently in order of importance. A transformation is accepted if it decreases the cost of one component without increasing more important costs. Optimization stops when all costs are zero, or no further improvement can be made to the cost function [7].

Note that if design rule violations are present after the first phase of optimization, there may be multiple ways to fix them. Ties in moves that fix violations are broken by subsequent cost components such as power. Consequently, although design rule (and delay) optimization for violated constraints can increase the power of a design, such increases will be minimized.

Keeping the cost function current after every attempted transformation is essential. Each move can have an impact across multiple costs. A move can also have different effects on components within a particular cost. For example, a move that improves the internal power of a cell may increase the pin capacitance of a cell. Internal power is reduced but switching power increases, possibly creating a total increase in dynamic power.

Completely recomputing the cost function is prohibitively expensive. Incremental updating of the cost function enables more powerful algorithms to explore a greater part of the solution space. For power, switching, internal, and leakage values are incrementally updated during power optimization. Leakage power updates are simple, as it is a simple cell value. Cells that are added or deleted affect the leakage power of the design in a corresponding way.

Cell internal power updates occur not only when a cell is added, deleted, or sized but also when input transition times or the output load of a cell changes. The latter situations can be easily detected such that internal power updates only occur on cells that have changed their transition times or capacitance (Figure 1).

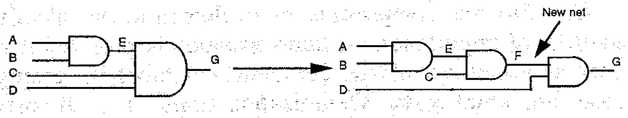


Capacitive load affects cells: A, B

Driven net transition time affects cells: D, E, F, G

Figure 1 — Incremental Internal Power Updates

For switching power, activities of only the new or changed nets are recomputed by the probabilistic simulation engine when the connectivity or local functionality of a design changes. The switching activity of the unchanged nets serve as startpoints for propagation. This method is referred to as local propagation and is very fast since the BDD needed to calculate the switching activity is small (Figure 2).



Only net F requires incremental switching activity calculation which is a function of the switching activity of signals E and C

Figure 2 — Local Propagation

Since the BDDs are not built with respect to the primary inputs, some accuracy is lost due to less spatial correlation information. However, absolute accuracy loss is acceptable as long as the relative accuracy of the switching activities between nets is maintained.

4. Power Optimization Techniques

Many transformations are used during power optimization in order to reduce the overall power consumption of the design. Below are descriptions of transformations that are evaluated that try to reduce one of the four principle factors mentioned earlier: gate transistor dimensions, net switching activities, net transition times and net capacitive loading.

4.1 Gate transistor dimensions

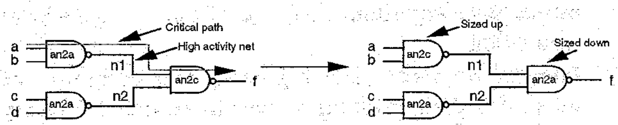
As earlier described in Table 1, the dimensions of the transistors that compose a CMOS gate can influence a number of factors that determine the power consumption of a design. Of these factors, there are two that have the largest direct impact on power: parasitic capacitance and current drain.

For MOS transistors, the main sources of capacitance (to a first order) are the gate-to-channel, gate-to-bulk, and the diffusion-to-bulk capacitance [10]. In addition, the wiring that is required to connect the transistors to form the logic function create capacitance internal to a logic gate. Current drain impacts the short circuit and leakage power of a CMOS gate and is partially dependent on the length and width of the transistor channel.

As described earlier, Design Power models the above components of power in its gate-level power model. Since Power Compiler works at the gate-level, it uses cells from a pre-characterized set of discrete cell sizes defined in the technology library. Sizing of a cell is performed by choosing different implementations of the same function from the technology library [2],[5]. These implementations vary in their parasitic capacitance as well as internal and leakage power characteristics.

Power Compiler will choose an optimal set of sizes for cells that will minimize the power in a design while considering other design objectives according to the prioritized cost function. The success of sizing depends on the technology library and the design being optimized [4]. Libraries that have many different implementations of each logic function will give the tool the best opportunities for power reduction.

Designs that have many critical signals are difficult to optimize for power, since timing is a higher priority cost. For the most part, moves that improve power will worsen delay. By first consuming timing slack on the areas of greater power, Power Compiler achieves the greatest power reduction with minimal impact to timing (Figure 3).



Delay (a,f) : reqd = 4, actual = 3.3
Cload: f = 4; n1, n2 = 2
TR: a, b = .25, c,d = .5
=> n1 = .125, n2 = .25, f = .56
Power = 4.125

Delay (a,f) : reqd = 4, actual = 3.5
Cload: f = 3; n1 = 2.5, n2 = 1.5
TR: a, b = .25, c,d = .5
=> n1 = .125, n2 = .25, f = .56
Power = 3.69

Note: Internal power effects (i.e. edge rate) also considered

Figure 3 — Cell Sizing Example

4.2 Net switching activities and capacitive loading

As described in section 1.2, switching power dissipation is directly proportional to the sum of the product of transition count and capacitance for each net in the design. If this total can be reduced, then the total power can be reduced.

One technique that Power Compiler uses to reduce switching power is composition [10]. Sets of cells are merged or composed into a single complex cell. The nets connecting the cells are enclosed within a single cell (Figure 4).

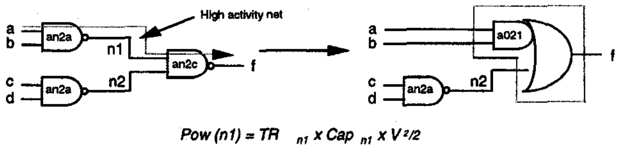


Figure 4 — Composition Transformation Example

The power reduction comes from the fact that the switching power of the enclosed net is completely eliminated. The cell internal power may increase due to the increase in gate size. This transformation is focused on nets with high switching power.

Another technique used to reduce the switching power is pin swapping. Some cells can have input pins that are symmetric with respect to the function of the gate, but have different capacitance values. Power can be reduced by assigning a high switching net to a pin with a lower capacitance and a low switching net to a logically symmetric pin with higher capacitance (Figure 5).

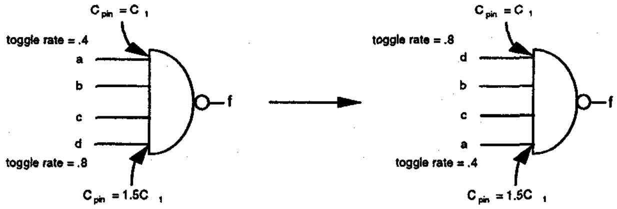


Figure 5 — Pin Swapping Example

4.3 Net transition times

The internal power of a cell is a function of input transition time (and output load). The slower the rise or fall of an input signal, the longer the transistors stay in their linear mode of operation, prolonging the duration of the direct path between V_{DD} and GND. This leads to increased short circuit power dissipation.

The power due to net transition time can be minimized by decreasing (sharpening) the transition times at the inputs. Power Compiler will reduce net transition times by two ways: sizing and buffering.

Substituting the driver of a net with a higher drive cell can sharpen the transition time. However, this type of move will generally increase the internal power of the driver as well as the total net capacitance, so the move is only accepted if it improves the overall power of the design.

Adding buffers can also improve transition times, but at the expense of additional internal power and capacitance introduced by the new cell. Note that improving transition times of one net can improve transition times in its transitive fanout which may lead to further power reductions (Figure 6).

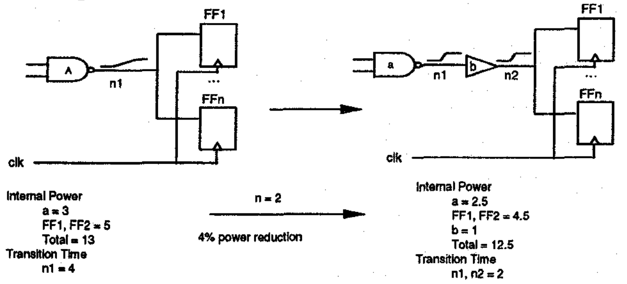


Figure 6 — Buffer Insertion

5. Experimental Results

			Power (mW)		
Design	library	# cells	initial	final	reduction
Alarm	lib1	628.00	19.86	18.67	5.99%
Cpu	lib1	1390.00	111.27	104.03	6.51%
Barcode	lib1	266.00	5.85	5.77	1.44%
Cla	lib1	36.00	20.75	18.68	9.96%
dsp1	lib1	228.00	9.19	9.05	1.50%
sort	lib1	808.00	5.51	5.43	1.47%
dsp2	lib1	1155.00	23.73	22.15	6.67%
wall	lib1	329.00	232.21	222.32	4.26%
alarm	lib2	633.00	33.16	11.07	66.62%
cpu	lib2	1193.00	153.32	100.49	34.46%
barcode	lib2	257.00	24.21	19.15	20.91%
cla	lib2	58.00	15.86	15.69	1.07%
dsp1	lib2	224.00	39.79	29.54	25.75%
sort	lib2	831.00	128.02	117.99	7.83%
dsp2	lib2	1162.00	26.45	22.35	15.50%
wall	lib2	325.00	191.13	168.96	11.60%
alarm	lib3	564.00	14.92	13.88	6.97%
cpu	lib3	1108.00	63.49	58.03	8.60%
barcode	lib3	226.00	8.88	8.54	3.86%
cla	lib3	26.00	5.24	4.84	7.59%
dsp1	lib3	170.00	15.34	15.13	1.43%
sort	lib3	803.00	77.65	71.47	7.96%
dsp2	lib3	761.00	11.11	9.64	13.29%
wall	lib3	265.00	53.82	51.81	3.74%
Totals					**11.46%**

Table 2. Experimental Results

Design	library	Area			CPU
		initial	final	% larger	(s)
Alarm	lib1	2392.00	2420.00	1.17%	481.64
Cpu	lib1	4838.00	5558.00	14.88%	2509.76
Barcode	lib1	1173.00	1386.00	18.16%	296.83
Cla	lib1	132.00	149.00	12.88%	25.65
dsp1	lib1	905.00	1069.00	18.12%	265.09
sort	lib1	3032.00	3476.00	14.64%	1450.88
dsp2	lib1	4162.00	4377.00	5.17%	2226.96
wall	lib1	1065.00	1093.00	2.63%	541.98
alarm	lib2	1644.00	1776.00	8.03%	455.95
cpu	lib2	3360.00	3994.50	18.88%	1785.28
barcode	lib2	900.00	927.00	3.00%	221.26
cla	lib2	96.00	121.50	26.56%	29.79
dsp1	lib2	682.50	784.50	14.95%	320.24
sort	lib2	2379.00	2476.50	4.10%	1006.42
dsp2	lib2	3193.50	3379.50	5.82%	1741.34
wall	lib2	934.50	979.50	4.82%	457.05
alarm	lib3	988.50	1104.50	11.73%	523.59
cpu	lib3	2259.25	2467.75	9.23%	1931.40
barcode	lib3	592.50	601.50	1.52%	255.52
cla	lib3	62.00	63.25	2.02%	29.47
dsp1	lib3	398.00	432.75	8.73%	110.23
sort	lib3	1703.25	1687.25	-0.94%	1104.42
dsp2	lib3	2001.25	2418.25	20.84%	1442.16
wall	lib3	560.75	555.25	-0.98%	329.07
Totals				**9.41%**	

Table 2, contd. Experimental Results

Table 2 shows the experimental results from Power Compiler The initial mapping was obtained from Design Compiler with delay and area constraints applied to each design. Synopsys VHDL System Simulator (VSS) was run with full gate delays to obtain accurate toggle rates for all nets which were then used to compute the power dissipation of the mapped design. Power Compiler was then invoked to perform technology-dependent power optimizations based on the simulation-generated toggle rates with delay and area constraints still present. Final power values were calculated again using VSS with full gate delays. All experiments were performed on a SPARCstation 20.

The experiment was run with eight industrial designs with three different libraries. The libraries have been characterized for internal power. The large variation of power optimization success between *lib2* and the other two libraries is due mainly to the presence of a low-power sequential cell in *lib2*, which clearly demonstrates the dependence of results on library quality.

Also worth noting is that the area of a design increased on average with power optimization. Design Compiler had performed good area optimization in the initial mapping. Some area increase was due to the presence of lower power cells that had larger area than their higher power equivalents in a given library. Buffer insertion and up-sizing for transition time reduction also accounted for area increases.

6. Conclusion

This paper presented the power optimization techniques used by Power Compiler™, the first commercially available power optimization synthesis tool. These techniques are able to address the major elements of power consumption in a CMOS digital design due to the accurate power models of the Design Power™ analysis tool. Power optimization operates within other specified design constraints such as delay.

Experimental results show an average 11.46% power reduction across a set of industrial designs and show a strong dependency on the quality of the technology library. Power optimization did not sacrifice delay, but came at the expense of some area, which shows that area optimization is not sufficient to realize the best gains in power.

7. Acknowledgments

The authors would like to thank Ashutosh Mauskar for providing the benchmark designs and design methodology to run the experiments. The authors would also like to thank Olivier Coudert, Ramsey Haddad, Shankar Krishnamoorthy, and Rick Rudell for their insights into the gate-level power optimization problem.

8. References

[1] A.P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R.W. Brodersen, "Optimizing power using transformations", *IEEE Transactions on CAD,* vol. 14, pp. 12-31, January 1995

[2] O. Coudert, R. Haddad, S. Manne, "New Algorithms for Gate Sizing: A Comparative Study", *Proceedings of the Design Automation Conference June 1996,* pp. 734-739

[3] S. Devadas. A.Ghosh, K. Keutzer, *Logic Synthesis,* McGraw-Hill, 1994

[4] K. Keutzer, K. Scott, "Improving Cell Libraries for Synthesis", *Proceedings of the International Workshop on Layout Synthesis 1993,* pp 8b1-8b5 .

[5] D.P. Marple, "Transistor Size Optimization in the Tailor Layout System", *Proceedings of the Design Automation Conference,* pp. 43-48, June 1989

[6] F. Najm. "Transition Density: A Stochastic Measure of Activity in Digital Circuits", *Proceedings of the Design Automation Conference,* pp. 644-649, June 1991

[7] *Synopsys Design Compiler Family Reference Manual*

[8] *Synopsys Design Power Reference Manual*

[9] C.Y. Tsui, M. Pedram, and A.M. Despain, “Power Efficient Technology Decomposition and Mapping Under an Extended Power Consumption Model,” *IEEE Transactions on CAD,* vol. 13, pp. 1110-1122, Sept. 1994

[10] N. Weste and K. Eshragian, *Principles of CMOS VLSI Design,* 2nd Edition, Addison-Wesley, 1993