Faster solution of nonlinear equations using logical effort method and curve fitting in low power design

Mohsen Radfar[a] **and Saadat Pourmozaffari**[b]

Department of Computer Engineering & IT, Amirkabir University of Technology

P. O. Box 15914, Hafez St., Tehran, Iran

a) *radfar@ce.aut.ac.ir*

b) *saadat@ce.aut.ac.ir*

Abstract: Delay optimization with considering effective parameters in the design, such as energy consumption, need an extra effort to achieve quick solutions. In this paper, by taking the requirements of current design processes into account for sake of simultaneously considering all of the effective parameters, in one hand, and also trying to use the privilege of Logical Effort method's extra speed, in the other hand, a new method for solving the equations optimization problems is represented. With providing an initial point based on Logical Effort method by Geometric Programming, the convergence rate of optimization algorithms can be greatly increased.

Keywords: low power design, VLSI, GGP[1], NLP[2]

Classification: Integrated circuits

References

[1] H. Chou, Y.-H. Wang, and C. C.-P. Chen, "Fast and Effective Gate-Sizing with Multiple-Vt Assignment using Generalized Lagrangian Relaxation," *Proc. Asia South Pacific – Design Autom. Conf.*, pp. 381–386, 2005.

[2] K. Agarwal, D. Sylvester, D. Blaauw, and A. Devgan, "Achieving Continuous Vt performance in a dual-Vt process," *Proc. Asia South Pacific – Design Autom. Conf.*, pp. 393–398, 2005.

[3] T. Karnik, Y. Ye, J. Tschanz, L. Wei, S. M. Burns, V. Govindarajulu, V. De, and S. Borkar, "Total Power Optimization by Simultaneous Dual-Vt Allocation and Device Sizing in High Performance Microprocessors," *Proc. Design Autom. Conf.*, pp. 486–491, 2002.

[4] S. Shah, A. Srivastava, D. Sharma, D. Sylvester, D. Blaauw, and V. Zolotov, "Discrete Vt Assignment and Gate Sizing Using a Self-Snapping Continuous Formulation," *Proc. Asia South Pacific – Design Autom. Conf.*, pp. 301–306, 2005.

[5] C. Chen, A. Srivastava, and M. Sarrafzadeh, "On gate level power optimization using dual-supply voltages," *IEEE Trans. VLSI Syst.*, vol. 9, pp. 616–29, 2001.

[1]General Geometric Programming
[2]Non-Linear Programming

DOI: 10.1587/elex.6.889
Received March 08, 2009
Accepted June 04, 2009
Published July 10, 2009

[6] Y. S. Dhillon, "Hierarchical Optimization of Digital CMOS Circuits for Power, Performance and Reliability," *A Dissertation Presented to Georgia Institute of Technology*, 2005.

[7] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optimization and Engineering*, Springer, vol. 8, no. 1, pp. 67–127, 2007.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[9] R. F. Sproull and I. E. Sutherland, "Logical Effort: Designing for Speed on the Back of an Envelop," *IEEE Advanced Research in VLSI*, C. Sequin (editor), MIT Press, 1991.

[10] I. Sutherland, B. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*, Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[11] T. F. Coleman and Y. Li, "On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds," *Mathematical Programming*, vol. 67, no. 2, pp. 189–224, 1994.

[12] T. F. Coleman and Y. Li, "An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds," *SIAM Journal on Optimization*, vol. 6, pp. 418–445, 1996.

[13] A. Mutapcic, K. Koh, S. Kim, and S. Boyd, *A MATLAB® Toolbox for Geometric Programming, February 2007*, ggplab version 1.0 RC2, [Online] www.stanford.edu/_boyd/ggplab.html

[14] S. Boyd, S.-J. Kim, D. Patil, and M. Horowitz, *Digital circuit optimization via geometric programming, 2005*, [Online] www.stanford.edu/˜boyd/˜gp_digital_ckt.html

[15] M. Grant, S. Boyd, and Y. Ye, *CVX Users' Guide: for cvx version 1.2 (build 667), June 2008*, [Online] www.stanford.edu/˜boyd/cvx/cvx_usrguide.pdf

1 Introduction

For much of the history of the CMOS design, power was a secondary consideration behind speed and area for many chips. As transistor counts and clock frequency increased, power consumption has increased and now is a primary constraint.

Local solutions currently exist in the low power design in subjects such as supply voltage [1, 2] and threshold voltage [3], exclusively, and some solutions have been made by discrete optimization algorithms [4] or heuristics [5]. Their accuracy and speed are suffering from the algorithms, such as [5], and most of these algorithms have been designed for special purposes [6].

But, Geometric Programming (GP), now, provides a general, efficient and fast method for VLSI optimization problems [7] (and Convex Problems [8], generally). Here, mathematical modeling of VLSI circuits in a GP problem format can also be easily done. Nevertheless, GP is still a nonlinear programming and so it needs some improvements which here are provided through initial point improvements. Curves that are fitted on different initial points patterns, help to achieve immediately initial points and lead the nonlinear solution toward the optimum solution.

The rest of the paper is organized as follows: section 2 introduces the

© IEICE 2009
DOI: 10.1587/elex.6.889
Received March 08, 2009
Accepted June 04, 2009
Published July 10, 2009

logical effort method to be used in modeling. Then, a curve fitting on solution space of this problem's format is made by the routine expressed in section 3. Based on the results in section 4, this method can be used as a quick and accurate solution for the problems. Finally this paper is concluded in section 5.

2 Gate delay modeling using the Logical Effort

In general the propagation delay of a gate is modeled as a first degree equation. The complexity is represented by Logical Effort [9]. Logical effort of a gate is defined as the ratio of the input capacitance of the gate to the input capacitance of an inverter.

Assume a gate size is x times to its smallest size (with equal rise and fall times) and its subsequent gate is y times, thus delay equation becomes $D = R/x(C_{out} + xC_p)$ where C_{out} and C_p are output and parasitic capacitances of the gate, respectively. If C is the gate size of a feature size transistor (in which $x = 1$), then, for an inverter as an output or load gate with $C_p = 0$, $D = 3RC$. So equation is normalized as (lower-case letters for capacitances indicate normalized values):

$$D = \frac{1}{3x}\left(yc_{out} + xc_p\right) = \frac{1}{x}\left(\frac{xc_{in}}{3}\frac{yc_{out}}{xc_{in}} + \frac{xc_p}{3}\right) = \frac{c_{in}}{3}\frac{yc_{out}}{xc_{in}} + \frac{c_p}{3} \qquad (1)$$

where C_{in} is input capacitance of the gate, $\frac{C_{in}}{C} = c_{in}$, $\frac{C_{out}}{C} = c_{out}$ and $\frac{C_p}{C} = c_p$.

yc_{out} is exactly the capacitance on the gate output, that is, $c_L = \frac{C_L}{C}$, and xc_{in} is the capacitance on the gate's input, that is, $c_I = \frac{C_I}{C}$.

Assuming a circuit with n sequential gates, arbitrary and distinct branches at each gate's output, given load, C_L, and input, C_I, capacitances, the main problem is obtaining x in these n gates such that delay is minimized, and then power consumption which has been applied as a constraint. It should be considered that here the critical path in a multiple path circuit is dealt with and the shared gates with other paths are involved automatically in the branches. On this problem, it is assumed that the size of 0-th and n-th gate is given and the objective is to obtain $x_1, x_2, x_3, \ldots, x_n$ at situation in which no gate has branches; and in general specifying x_{ij}, that is the size of j-th branch's gate on output of (i-1)-th gate for $0 \leq i \leq n+1$ and $1 \leq j \leq b_i$. b_i is the number of branches on i-th gate's output and x_{i1}, $0 \leq i \leq n+1$, is the size of i-th gate of the primary given n gates. By these definitions we have:

$$D = \sum_{i=0}^{n}\left(\frac{1}{3x_{i1}}\left(\sum_{j=1}^{b_{i+1}}\left(x_{i+1\,j}\,c_{in\,i+1\,j}\right) + x_{i1}c_{p\,i}\right)\right) \qquad (2)$$

where $c_{in\,i+1\,j}$, $0 \leq i < n+1$, $1 \leq j \leq b_{i+1}$, is also the size of j-th branch's input capacitance on i-th gate's output of the n gates. Indeed in this situation, the given values of problem are $c_I = x_{01}\,c_{p0}$, (so x_{01} is given),

© IEICE 2009
DOI: 10.1587/elex.6.889
Received March 08, 2009
Accepted June 04, 2009
Published July 10, 2009

$c_{L\,j} = x_{n+1\,j}\,c_{in\,n+1\,j}$, (so $x_{n+1\,j}$ is given), $c_{in\,i+1\,j}$ for $0 \le i < n$, and $c_{p\,i}$ for $0 < i \le n$. In minimizing D, because of independency assumption of $x_{i\,j}$ for $1 \le i \le n$ and $2 \le j \le b_{i+1}$ to $x_{i\,1}$ for $1 \le i \le n$, obviously reducing $x_{i\,j}$ results in D reduction and hence the optimal value is $x_{i\,j} = 0$. In [10], by simple assumption of equality $x_{i\,j} = x_{i\,1}$ and simplifying of (2), D has been minimized. Thus equation (2) becomes:

$$D = \sum_{i=0}^{n}\left(\frac{1}{3x_i}\left(b_{i+1}\,x_{i+1}\,c_{in\,i+1} + x_i c_{p\,i}\right)\right) = \sum_{i=0}^{n}\left(\frac{b_{i+1}\,x_{i+1}\,c_{in\,i+1}}{3x_i} + \frac{c_{p\,i}}{3}\right) \quad (3)$$

For optimization we have:

$$b_i \frac{x_i}{x_{i-1}} \frac{c_{in\,i}}{3} = b_{i+1} \frac{x_{i+1}}{x_i} \frac{c_{in\,i+1}}{3} \qquad 1 \le i \le n$$

which can simply be solved analytically. If for some j and k, $2 \le k, j \le b_{i+1}$, x_{ij} be different from x_{ik}, that is x_i be different, there is no analytical solution for this generalized problem and numerical ways should be taken. The method, stated in the following section using simplifying assumptions, deal with this problem and can be similarly applied to the generalized problem.

3 Fitting routine

General equation (2) for considering power can be written as equation (4) in which P is a power constraint:

$$\begin{aligned} &\min D = \sum_{i=0}^{n}\left(\frac{1}{3x_{i1}}\left(\sum_{j=1}^{b_{i+1}}\left(x_{i+1\,j}\,c_{in\,i+1\,j}\right) + x_{i\,1}c_{p\,i}\right)\right) \\ &\textit{subject to } V_{DD}^2 \sum_{i=0}^{n}\sum_{j=1}^{b_{i+1}}\left(\left(x_{i+1\,j}\,c_{in\,i+1\,j}\right) + x_{i\,1}c_{p\,i}\right) \le P \end{aligned} \quad (4)$$

Like equality assumption in expression (3) we have equation system (5):

$$\begin{aligned} &\min D = \sum_{i=0}^{n}\left(\frac{b_{i+1}x_{i+1}c_{in\,i+1}}{3x_i} + \frac{c_{p\,i}}{3}\right) \\ &s.t. \quad \sum_{i=1}^{n}\left(x_i\left(c_{in\,i}b_i + c_{p\,i}\right)\right) \le c_s \end{aligned} \quad (5)$$

where $c_{s0} = \dfrac{P}{V_{DD}^2} - \left(x_0 c_{p\,0} + x_{n+1}c_{in\,n+1}b_{n+1}\right)$.

For simplicity it is considered that all the gates are inverters. Hence (5) is converted to the following form:

$$\begin{aligned} &\min D = n + 1 + \sum_{i=0}^{n}\frac{x_{i+1}}{x_i} \\ &s.t. \quad 6\sum_{i=1}^{n} x_i \le c_{s0} \end{aligned} \quad (6)$$

by replacing $c_s = c_{s0}/6$ the solution of (6) is identical to the solution of the following equation:

$$\begin{aligned} &\min D = \sum_{i=0}^{n}\frac{x_{i+1}}{x_i} \\ &s.t. \quad \sum_{i=1}^{n} x_i \le c_s \end{aligned} \quad (7)$$

© IEICE 2009
DOI: 10.1587/elex.6.889
Received March 08, 2009
Accepted June 04, 2009
Published July 10, 2009

This system's solution can only be gotten by numerical solutions. To present a faster solution for this Geometrical Programming problem with respect to nonlinear algorithms, an approximation is made using curve fitting on data experienced. After getting an (appropriate) approximation on the initial points for the problem's variables, we are now able to solve the problem numerically.

Lagrange Multipliers method, gives following nonlinear equation system made of $n+1$ equations in $n+1$ variables:

$$\begin{cases} \dfrac{\frac{x_{i+1}}{x_i} - \frac{x_i}{x_{i-1}}}{x_i} = \lambda \quad 1 \le i \le n \\ \dfrac{\frac{x_{n+1}}{x_n} - \frac{x_1}{x_0}}{c_s} = \lambda \end{cases} \tag{8}$$

Having x_n and x_1 or x_n and x_{n-1}, solves equation (8). Optimum curves, fitted on these variables, are gained by minimizing sum of squares of various fitted curves and experimental data differences at solution space. The least square problem solving is done by using Trust-Region sub-space method and based on Interior-Reflective Newton technique [11, 12].

By numerical solution of equation (7) using MATLAB® for different values x_0, x_{n+1}, n, and c_s, some experimental data is exploited. The algorithm can be chosen among the convex problem solving algorithms, especially geometric ones, because the programming problem is geometric and sparse [7]. For this purpose, the recently published algorithms in [13] have been used to be applied to MATLAB® software for upgrading the accuracy of results. In GGP library, in fact, the equation (8) is used together with the KKT (Karush-Kuhn-Tucker) method [13].

4 Results

Equation (9) exhibits the best fitting under the conclusions based on the following results. Obviously more accurate fitted curves need more speed in processing of instructions of nonlinear least square problem's solution.

$$x_n = \frac{1}{\sqrt{c_s}}\left(\frac{1}{2}x_{n+1} + 7\right)(.02n + 10)^2 \times \left(e^{-2.7x_0 - 0.7} - \frac{1}{2}\right) + x_{n+1} + 23 \tag{9}$$

Accuracy of this fitting can be inferred from figures (1) and (2) which are exploited from simulations made by MATLAB® for different values as follows. x_n in solution space of equation (7) is related to four variables n, c_s, x_n, and x_{n+1}, therefore in three dimensional plot a parameter would be constant.

In the figures, n ranges from 50 to 150 gates, c_s from 200 to 300, x_0 from 1 to 41 and x_{n+1} from 50 to 150 times of gates with minimum size transistors and equal rise and fall times.

Points on net planes are resulted in solving respective equations systems by GP algorithm and the net planes themselves are produced by their interpolation. Solid planes are obtained by formula (9). It is obvious that the adjacency of solid and net planes is resulting from the accuracy of formula

© IEICE 2009
DOI: 10.1587/elex.6.889
Received March 08, 2009
Accepted June 04, 2009
Published July 10, 2009

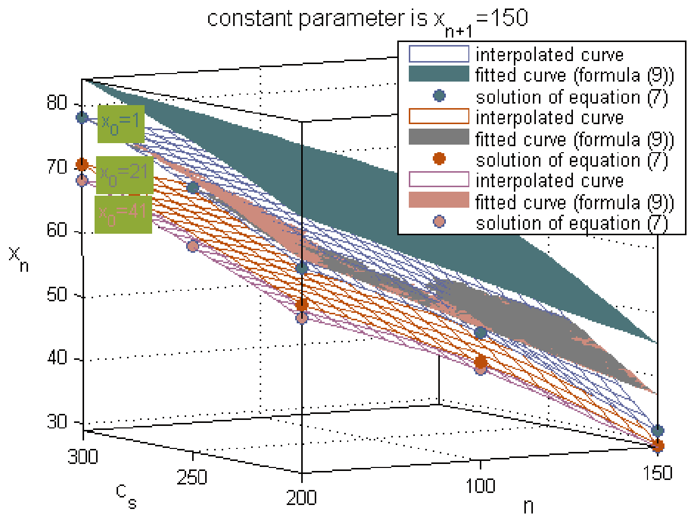

Fig. 1. relation between x_n and three variables n, c_s, and x_0.

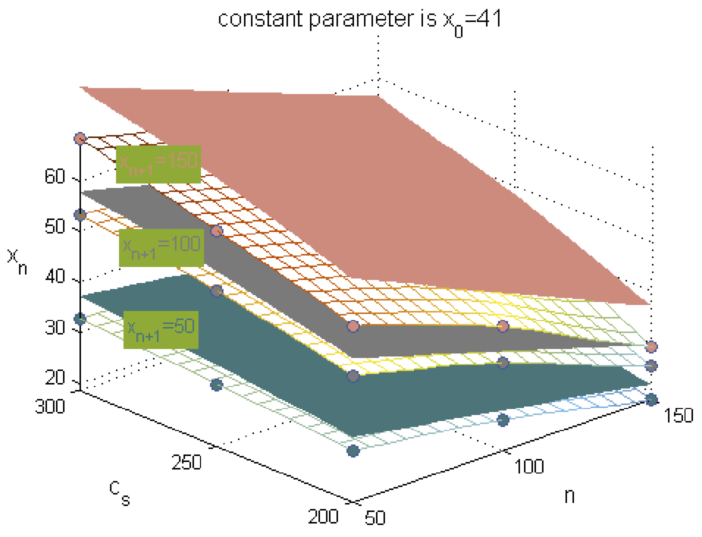

Fig. 2. relation between x_n and three variables n, c_s, and x_{n+1}.

(9) which is one the fitted curves on table I. This adjacency has been exactly explained using numerical errors in Table I.

Table I shows different curves fitted on data exploited from solving equation (7). Patterns of these functions have been selected to be exponential because of using power constraint in optimization problem [14]. The precision of the calculations for GP solver is 1.1×10^{-11} [15] and for MATLAB curve fitting is 10^{-6}, so values of the table have become rounded. It should be noted that the least square errors, are sum of square of differences between fitted pointes and actual points for x_n and in fact squares of differences have been added on the overall set of exploited points of x_n. This error has not unit because x_n has not unit. x_n is the ratio of gate's size to feature size gate and error is sum of square of differences between x_n.

Proximity of an initial point to the final solution is very effective on

© IEICE 2009
DOI: 10.1587/elex.6.889
Received March 08, 2009
Accepted June 04, 2009
Published July 10, 2009

numerical solution process. Final solution precision and numerical solving algorithm's convergence rate are extremely dependent upon the initial point selection [13]. In fact, number of iterations of GP algorithm for equation (7) depends on how good the initial points are. Relative errors on the Table I illustrate the accuracy of corresponding fitted curves. This error is the ratio of sum of absolute errors to sum of absolute values.

Table I. fitting of different curves on data exploited.

Relative Error	Least Square Error	Fitted function
0.170%	47.3759	$x_n = (0.9c_s - 0.07)^{-0.46}(0.69x_{n+1} + 7.21)(.02n + 10)^{1.8}(e^{-2.7x_0 - 0.7} - \frac{1}{2}) + x_{n+1} + 23$
0.173%	48.6591	$x_n = \frac{1}{\sqrt{c_s}}(\frac{1}{2}x_{n+1} + 7)(.02n + 10)^2(e^{-2.7x_0 - 0.7} - \frac{1}{2}) + x_{n+1} + 23$
0.178%	51.6752	$x_n = (0.35c_s - 6)^{1.84}(0.05x_{n+1} + 6.48)(e^{-0.1n - 6.72} + 0.0004)$ $(e^{-0.17x_0 + 0.18} + 0.8) + 0.3x_{n+1} + 7.2$
0.187%	57.2496	$x_n = (0.77c_s - 3.75)^{0.56}(0.29x_{n+1} + 4.38)(e^{-0.006n - 1.3} + 0.202)$ $(e^{-2.71x_0 - 0.71} + 0.18) + 0.3x_{n+1} + 4.73$
0.194%	61.4041	$x_n = (0.54c_s - 4.34)^{\frac{3}{2}}(0.07x_{n+1} + 6.54)(e^{-0.1n - 6.41} + 0.0007)$ $(e^{-0.17x_0 + 0.02} + 0.97) + 0.27x_{n+1} + 6.95$
0.231%	87.2068	$x_n = (0.4c_s - 0.33n)^{-0.076}(2.23x_{n+1} + 0.04x_0)$ $(1.45n + 2.73)^{0.01}(e^{-3.19x_0 - 1.19} - 1.68) + 4.57{x_{n+1}}^{0.93}$
0.360%	211.5868	$x_n = (0.99c_s - 0.49)^{-0.72}(0.97x_{n+1} + 0.17x_0)$ $(.023n + 10.47)^{1.69}(e^{-2.99x_0 - 0.99} - 0.35) + 2.82{x_{n+1}}^{0.76}$
0.407%	270.1156	$x_n = (1.004c_s + 3.01)^{0.14}(1.28x_{n+1} + 0.012x_0)^{0.84}$ $(-0.99x_{n+1}^{-0.89} + 0.92)(e^{2.71x_0 - 0.71} + 1.2) - 0.98x_{n+1} - 0.08x_0$
0.521%	443.4422	$x_n = (0.75c_s - 0.01n)^{-0.11}(1.46x_{n+1} + 0.1x_0)^{1.03}$ $(1.28x_{n+1}^{0.087})(e^{2.99x_0 - 0.99} - 0.92) + 2.24x_{n+1}0.05x_0$

Another problem in numerical (even analytical) solving nonlinear problems is the locally optimal solutions which are not globally optimal. A common way to deal with this problem is using initial points that are close to generally optimal solutions. Here low relative errors on the table show the proximity of approximations to final solution and can help to get rid of this problem.

5 Conclusion

Automated design of integrated circuits by algorithms with the ability of creating quick and low power circuits is inevitable. A technique was proposed that uses the Logical Effort idea for low power design in order to increase its performance. Generally speaking, simultaneously considering speed and power parameters would be resulted in numerical solutions for known problems, that is, nonlinear geometric systems. This article tried to ease their solutions using an almost accurate approximation of initial point to speed

DOI: 10.1587/elex.6.889
Received March 08, 2009
Accepted June 04, 2009
Published July 10, 2009

up the numerical solution. This method for solving GGP problems is considering effective parameters in VLSI designs and can improve the algorithms' speed and convergence rate.

© IEICE 2009
DOI: 10.1587/elex.6.889
Received March 08, 2009
Accepted June 04, 2009
Published July 10, 2009