functions (like parity) have an optimal two-level implementation that is exponential in size, but a multi-level implementation that is linear in size; however, we found no such functions in the state machine benchmark set. In all of our experiments we found no state assignment that produced an extremely bad two-level implementation but a very good multi-level implementation.

IV. Conclusions

First, optimization techniques that take advantage of don't care conditions give better implementations of state-assigned logic and expose greater variation in the quality of different state assignments.

Second, our experiments showed several important properties of state assignments implemented in multi-level logic. Deterministic experiments (like those of *Devadas* et al.) show that a two-level assignment algorithm can do surprisingly well compared to the best available multi-level assignment algorithm. Our random experiments show that, for machines like those in our benchmark set, state assignments that give good multi-level implementations also give good two-level implementations. Understanding the relationship between two-level and multi-level implementations of encoded logic should help in the development of better multi-level state assignment programs.

Acknowledgment

The authors have greatly benefited from and appreciate their discussions with Srinivas Devadas through electronic mail during the development of these new experimental results.

References

[1] K. A. Bartlett *et al.*, "Multilevel logic minimization using implicit don't cares," *IEEE Trans. Computer-Aided Design*, vol. 7, pp. 723–740, June 1988.

[2] R. K. Brayton, C. McMullen, G. D. Hachtel, and A. Sangiovanni-Vincentelli. *Logic Minimization Algorithms for VLSI Synthesis*. New York: Kluwer Academic, 1984.

[3] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. Wang, "MIS: A multiple-level logic optimization system," *IEEE Trans. Computer-Aided Design*, vol. CAS-6, pp. 1062–1081, Nov. 1987.

[4] S. Devadas, State assignment experiments, private communication, May 1988.

[5] S. Devadas, H.-K. Ma, A. R. Newton, and A. Sangiovanni-Vincentelli, "MUSTANG: State assignment of finite state machines targeting multilevel logic implementations," *IEEE Trans. Computer-Aided Design*, vol. CAD-7, pp. 1290–1300, Dec. 1988.

[6] B. Lisanke, "Finite-state machine benchmark set. Preliminary benchmark collection," Sept. 1987.

[7] J. W. Tukey, *Exploratory Data Analysis*. Reading, MA: Addison-Wesley, 1977.

[8] W. Wolf, K. Keutzer, and J. Akella, "A kernel-finding state assignment algorithm for multi-level logic," in *Proc. IEEE/ACM 25th Design Automation Conf.*, June 1988.

Comments on "Simulation of Nonlinear Circuits in the Frequency Domain"[1]

GEORGE W. RHYNE AND MICHAEL B. STEER

In the above paper,[1] the authors present an excellent review of the harmonic balance method for nonlinear circuit analysis and detail three techniques for solving the associated system of nonlinear equations: optimization, relaxation, and Newton's method. In their discussion of optimization, however, they claim that optimization is inefficient, that information is lost in calculating the cost function to be minimized, and that it is difficult to exploit the structure of the node equations when using optimization methods. The authors conclude that relaxation and Newton's method are the preferred techniques. In this paper we address these issues and comment on the relationship between the optimization approach and Newton's method.

As discussed in the above paper,[1] the simulation of nonlinear circuits using nodal analysis and the harmonic balance method requires solving the nonlinear system of equations resulting from the frequency-domain statement of Kirchoff's current law. This system is written as

$$\boldsymbol{F}(\boldsymbol{V}) = \boldsymbol{0} \tag{1}$$

where $\boldsymbol{F}$ is a complex vector whose elements are the sum of phasor currents at each node and frequency and $\boldsymbol{V}$ is the complex vector of node voltage phasors. This system can be solved directly (e.g., using Newton's method) or the solution can be found through minimization of an appropriate cost function. In the above paper,[1] the cost function

$$\epsilon(\boldsymbol{V}) = \boldsymbol{F}^*(\boldsymbol{V})\,\boldsymbol{F}(\boldsymbol{V}) \tag{2}$$

(where the asterisk represents the conjugate transpose) is considered and the relative merits of optimization discussed based on this choice. Using the notation of the above paper,[1] where $\overline{X} = [\mathrm{Re}\{X\} \;\; \mathrm{Im}\{X\}]^T$ refers to the equivalent real vector corresponding to the complex number X and using a similar convention for matrices and vectors, we form the corresponding cost function

$$\epsilon(\overline{\boldsymbol{V}}) = \tfrac{1}{2}\,\overline{\boldsymbol{F}}(\overline{\boldsymbol{V}})^T\,\overline{\boldsymbol{F}}(\overline{\boldsymbol{V}}) = \tfrac{1}{2}\sum_{i=1}^{2HN}\overline{\boldsymbol{F}}_i^2(\overline{\boldsymbol{V}}) \tag{3}$$

where there are H harmonics and N nodes, and the factor of $1/2$ has been added for convenience. In this case, optimization is a nonlinear least squares problem. Nonlinear least squares problems having zero or small residuals, e.g., where there is a solution $\hat{V}$ such that $\overline{\boldsymbol{F}}(\hat{\boldsymbol{V}}) \approx \boldsymbol{0}$, can be solved using the Gauss–Newton method. This method uses the iteration

$$\overline{\boldsymbol{V}}_{k+1} = \overline{\boldsymbol{V}}_k - \left(\overline{\boldsymbol{J}}(\overline{\boldsymbol{V}}_k)^T\,\overline{\boldsymbol{J}}(\overline{\boldsymbol{V}}_k)\right)^{-1}\overline{\boldsymbol{J}}(\overline{\boldsymbol{V}}_k)^T\,\overline{\boldsymbol{F}}(\overline{\boldsymbol{V}}_k) \tag{4}$$

where $\overline{\boldsymbol{J}}$ is the Jacobian of $\overline{\boldsymbol{F}}$ [1]. In the situation we are considering, the number of nodes at which Kirchoff's current law is applied equals the number of unknown node voltages. Thus the Jacobian is square and, assuming it to be nonsingular, the above iteration simplifies to

$$\overline{\boldsymbol{V}}_{k+1} = \overline{\boldsymbol{V}}_k - \overline{\boldsymbol{J}}(\overline{\boldsymbol{V}}_k)^{-1}\,\overline{\boldsymbol{F}}(\overline{\boldsymbol{V}}_k). \tag{5}$$

This is the same iteration obtained by applying Newton's method directly to (1). Clearly, with this particular cost function and in this application, Newton's method and optimization are directly related and computationally equivalent and we cannot conclude that either method is preferred.

The authors[1] also state that optimization is unwieldy due to the large number of variables. While this is a true statement, it is misleading as the number of unknowns is independent of the solution method. Solution via Newton's method, for example, also suffers from the fact that there is a large number of variables.

Finally, the authors comment that it is difficult to exploit the structure of the harmonic balance-node admittance equations when using optimization. This depends, however, on the optimization strategy that is employed. If the optimization is formulated as discussed above, the structure of the equations can be exploited in the same manner as in their Newton's method approach with the same efficiency.

In conclusion, we point out that optimization has the added advantage that design objectives may be included in the formulation

Manuscript received Aril 18, 1988. This work was supported in part by the National Science Foundation through a Presidential Young Investigator Award under Grant ECS-8657836. The review of this paper was arranged by Associate Editor M. R. Lightner.

G. W. Rhyne was with North Carolina State University, Raleigh, NC. He is now with Texas Instruments, Dallas, TX 75265.

M. B. Steer is with the Center for Communications and Signal Processing, Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695-7911.

IEEE Log Number 8927946.

[1] K. S. Kundert and A. Sangiovanni-Vincentelli, *IEEE Trans. Computer-Aided Design*, vol. CAD-5, pp. 521–535, Oct. 1986.

of the cost function [2]. In this way, the circuit performance can be optimized simultaneously with the steady state analysis. This, however, may result in the residual ($\bar{F}(\hat{V})$) being so large that the Gauss–Newton method cannot be used. In this case, the iteration scheme will not reduce to (5).

Reply[2] *by K. S. Kundert and A. Sangiovanni-Vincentelli*[3]

Rhyne and Steer bring up several interesting points that were only briefly discussed in our paper. Their points, as we understand them, are that using optimization techniques to solve the harmonic balance equations can be made as efficient as using root finding methods (in particular Newton–Raphson), and that by using optimization it is possible to combine the two tasks of solving the harmonic balance equations and choosing circuit parameters to maximize circuit performance into one operation. We feel that while they have started from a minimization problem and derived an iteration formula that is identical to Newton–Raphson, in doing so they have added enough constraints so that their method is no longer capable of finding arbitrary minima (and so, in our opinion, can no longer be considered an optimization method), but only those minima associated with roots of the underlying problem. Thus they have simply rederived the classic Newton–Raphson algorithm in a circuitous manner. As for combining performance optimization with equation solution, Rhyne and Steer themselves admit that the approach they propose is often inappropriate. However, there are other dangers in combining the two operations that do not appear to be well appreciated.

The cost function

$$\epsilon(V) = \tfrac{1}{2}F^*(V)\,F(V) \tag{1}$$

has two important characteristics. First, each root of F corresponds to a global minimum of ϵ, and at these points, $\epsilon(V) = 0$. Second, at each V that is a local minima of ϵ, but is not a root of F, the Jacobian of F [that is $J_F(V) = dF(V)/dV$] is singular. We seek $\hat{V}$, a global minimizer of ϵ. If F is continuously differentiable, then a necessary condition for $\hat{V}$ to be a minimizer of ϵ is that the gradient of ϵ at $\hat{V}$ be zero, i.e.,

$$\nabla\epsilon(\hat{V}) = 0. \tag{2}$$

This problem can be solved by using a wealth of techniques such as steepest descent, conjugate gradient, and Newton's method. However, solving (2) with Newton–Raphson is more difficult than solving $F(V) = 0$ because the equation involves the first derivative of the original function F, so applying Newton–Raphson requires knowing the second derivatives of F. In fact, the Newton–Raphson iteration used to solve (2) is

$$\nabla^2\epsilon(V^{(k)})\,[V^{(k-1)} - V^{(k)}] = -\nabla\epsilon(V^{(k)}) \tag{3}$$

where $\nabla^2\epsilon$ is the Hessian of ϵ.

From (1), it is easy to show that

$$\nabla\epsilon(V) = J_F^*(V)\,F(V) \tag{4}$$

and

$$\nabla^2\epsilon(V) = \frac{dJ_F(F)}{dV}F(V) + J_F^*(V)\,J_F(V). \tag{5}$$

Clearly, the Hessian is denser than is the Jacobian and is, therefore, considerably more expensive to LU factor. This is the reason for our claim that applying optimization to solve the harmonic balance equations was expensive because of the large number of unknowns. We did not intend to imply that the number of unknowns is larger with optimization than with root finding techniques such as Newton–Raphson or nonlinear relaxation, but rather that the computational complexity was higher with optimization because there is less sparsity to exploit.

Using (4) and (5), we can rewrite (3) as

$$\left[\frac{dJ_F(V)}{dV}F(V) + J_F^*(V)\,J_F(V)\right][V^{(k+1)} - V^{(k)}] = -J_F^*(V)\,F(V). \tag{6}$$

Rhyne and Steer propose to avoid computing the second derivative terms in the Hessian by exploiting the fact that we are solving for a root of F. When V is near the solution $\hat{V}$, then $F(V)$ should be small, and so the second derivative term $[dJ_F(V)/dV\,F(V)]$ in (6) can be dropped. Thus (6) becomes

$$J_F^*(V)\,J_F(V)\,[V^{(k+1)} - V^{(k)}] = -J_F^*(V)\,F(V) \tag{7}$$

which is referred to as the Gauss–Newton algorithm. By eliminating the second derivative term, the utility of this method as an optimization algorithm is greatly decreased because $F(V)$ must be small. Note that (7) is now in the form of the normal equation for a linear least squares problem. This equation is notoriously ill-conditioned because the condition number of $J_F^*(V)\,J_F(V)$ is the square of $J_F(V)$. Thus finding the roots of $F(V)$ using (7) is not only computationally more expensive (because $J_F^*(V)\,J_F(V)$ is denser than $J_F(V)$), but it may also be numerically unstable.

By assuming that $J_F(V)$ is square and nonsingular, it is possible to cancel $J_F^*(V)$ from both sides of (7), resulting in the following iteration:

$$J_F(V)\,[V^{(k+1)} - V^{(k)}] = -F(V). \tag{8}$$

This iteration is identical to that which results when Newton–Raphson is applied to $F(V) = 0$, hence, the claim by Rhyne and Steer that optimization can be made as efficient as Newton–Raphson when applied to solving the harmonic balance equations. However, by assuming that $J_F(V)$ is nonsingular, any possibility of finding a minimum of ϵ that is not a root of F has been eliminated. (Recall that minima occur where either F is zero or where J_F is singular.) Thus (8) can only be seriously considered a root finding method, not an optimization method. This, of course, comes as no surprise; since (8) is identical to the Newton–Raphson iteration, it can have no hidden powers not shared by Newton–Raphson. In fact, the steps that lead up to (8) are just a rather *round-about rederivation of the standard Newton–Raphson algorithm.*

It may be that optimization methods can be used efficiently to solve large systems of equations, but this has yet to be shown.

One question still remains; if it is necessary to both optimize circuit performance and solve the harmonic balance equations, should these two operations be combined into one optimization process? This idea was first suggested by Lipparini *et al.* [1]. They propose augmenting the cost function to be minimized with a contribution related to circuit performance, here denoted E. The list of design parameters is denoted p. The problem statement becomes

$$\min_{V,p}\,[F^*(V,p)\,F(V,p) + E^2(V,p)] \tag{9}$$

with the added constraint that $E = 0$ when all specifications are met and $E > 0$ otherwise. This approach is attractive because simple unconstrained optimization methods are used.

Unfortunately, a serious problem exists; (9) is not an accurate statement of the problem. The correct problem statement is

$$\min_{p} E^2(V,p) \quad \text{subject to} \quad F(V,p) = 0. \tag{10}$$

By using (9), we allow the optimizer to tradeoff satisfying Kirchoff's current law to improve circuit performance. The flaw,

[2]Manuscript received February 7, 1989. The review of this paper was arranged by Associate Editor M. R. Lightner.

[3]The authors are with the University of California, Berkeley, CA 94720.

IEEE Log Number 8927947.

of course, is that if Kirchoff's current law is not satisfied, the solution calculated is not feasible, and therefore, the actual circuit performance is not being measured.

Assuming something can be done to assure that Kirchoff's current law is satisfied, then there are several other important considerations. To treat (9) as a nonlinear least squares problem it is necessary to augment the list of equations with the performance cost function to be minimized and augment the list of variables with the optimizable parameters. Let

$$X = \begin{bmatrix} V \\ p \end{bmatrix}; \qquad G(X) = \begin{bmatrix} F(X) \\ E(X) \end{bmatrix}. \tag{11}$$

The cost function to be minimized becomes

$$\epsilon(X) = \tfrac{1}{2} G^*(X)\, G(X). \tag{12}$$

Augmenting the lists of equations and variables presents several problems. First, the new equations in G create new rows and columns in the Jacobian J_G that do not have the same structure as in J_F, making exploitation of the sparsity of Jacobian more difficult. Second, the number of design parameters in p is not usually the same as the number of cost functions in E. Thus J_G is not square. When applying Newton–Raphson to a system with more variables than equations, it is necessary to solve for the new iterate either by forming the normal equation, which effectively increases the number of equations until equal to the number of unknowns, or to solve the iteration equation with a method that is suitable for under-determined systems, such as QR factorization. The normal equation approach is written as

$$J_G^*(X^{(k)})\, J_G(X^{(k)})\, [X^{(k+1)} - X^{(k)}] = -J_G^*(X^{(k)})\, G(X^{(k)}). \tag{13}$$

As mentioned before, this equation is ill-conditioned and not nearly as sparse as when Newton–Raphson is applied directly to $F(V) = 0$. QR factorization would be applied to

$$J_G(X^{(k)})\, [X^{(k+1)} - X^{(k)}] = -G(X^{(k)}). \tag{14}$$

However, it is not possible to exploit sparsity in any significant way using QR factorization, and so this approach is impractical for large $J_G(X)$.

Lipparini's approach is not only dangerous, but it is also overly expensive for large problems. Adaptations that force Kirchoff's current law to be satisfied in Lipparini's approach, such as penalty functions, make the method even more inefficient. It is best to avoid this approach. Methods for solving (10) directly do exist, but in our opinion the two-level optimization/Newton–Raphson approach is the most promising. With the two-level approach, the best method can be used for each problem, Newton–Raphson for the large system of nonlinear equations and unconstrained optimization with a small number of parameters.

References

[1] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[2] V. Rizzoli, C. Cecchetti, A. Lipparini, and A. Neri, "User-oriented software package for the analysis and optimisation of nonlinear microwave circuits," in *Proc. Inst. Elect. Eng.*, pt. H, pp. 385–391, Oct, 1986.

[3] A. Lipparini, E. Marazzi, and V. Rizzoli, "A new approach to the computer-aided design of nonlinear networks and its application to microwave parametric frequency dividers," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-30, pp. 1050–1058, July 1982.

Two-Dimensional Analysis of a Merged BiPMOS Device

J. B. KUO, MEMBER, IEEE, G. P. ROSSEEL, MEMBER, IEEE, AND R. W. DUTTON, FELLOW, IEEE

***Abstract*—The BiPMOS device associated with n-well BiCMOS technologies consumes substantial area for isolation. A merged BiPMOS device structure is introduced to reduce the device size for BiCMOS VLSI. The performance of the merged BiPMOS device has been analyzed by PISCES-2B [1]. Comparisons between the merged BiPMOS device and the conventional one show that the merged BiPMOS device, which occupies a much smaller area, has a comparable performance.**

I. Summary

Fig. 1(a) shows a BiCMOS inverter. The PMOS and bipolar devices are the most important part during the pull-up transient [2]–[4]. The BiPMOS device in conventional BiCMOS structures is composed of bipolar and PMOS devices with substantial space for isolation between the two devices. Drain contact of the PMOS device and base contact of the bipolar device are wired by interconnects, which may bring substantial parasitics. Consequently, the performance of the BiPMOS device may be degraded. In this paper, a merged BiPMOS device structure is introduced to reduce the device size for BiCMOS VLSI. The merged BiPMOS device as shown in Fig. 1(b) is based on a 2 μm BiCMOS technology [5], [6] with a buried layer and a 0.8 μm 1×10^{16} cm^{-3} epi layer. The PMOS and the bipolar devices are located in the same n-well, which serves as the collector for the bipolar device and the substrate for the PMOS device. Instead of being connected by interconnects as in standard BiCMOS structure, the P+ drain region is placed against the extrinsic bipolar base region. This arrangement shrinks the length of the BiPMOS device to 5 μm. The PMOS device has an effective channel length of 1.2 μm and the bipolar one has a base width of 0.26 μm and a peak concentration of 2×10^{17} cm^{-3}.

A dc analysis has been carried out for the merged BiPMOS and the standard one with a simplified cross section [7] as shown in Fig. 1(c). The source contact of the PMOS device and the collector contact of the bipolar device are connected together via the collector/source electrode. The standard device has a bipolar device with an identical emitter width and a PMOS device with an equivalent channel length, which are separated by an oxide layer of 5000 Å. The drain contact of the PMOS device and the base contact of the bipolar device are connected by a common electrode. The substrate of the PMOS device is connected to the back gate electrode via a 100 kΩ resistance, accounting for the effective resistance in the neutral substrate region.

Applying a bias between the gate and the collector/source electrodes (V_{GS}) from 0 V to −5 V and a collector/source-to-emitter bias (V_{CE}) from 5 V to 1 V, emitter currents in A/μm have been extracted as shown in Fig. 2(a). Solid lines show the case with the merged BiPMOS device and the dashed ones with the standard one. The merged BiPMOS device has a threshold voltage of −0.8 V, which is defined as the V_{GS} as the magnitude of the emitter current

Manuscript received October 11, 1988; revised March 3, 1989. This work was supported by the U.S. Army Research Office under Contract DAAL03-87-G-0076 and by the R.O.C. National Science Council under Contract NSC77-0404-E002-44. The review of this paper was arranged by Associate Editor J. G. Fossum.

J. B. Kuo was with the Center of Integrated Systems, Stanford University, Stanford, CA. He is now with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

G. P. Rosseel and R. W. Dutton are with the Center of Integrated Systems, Stanford University, Stanford, CA 94305.

IEEE Log Number 8928246.