Vectorized Program Architectures for Supercomputer-Aided Circuit Design

VITTORIO RIZZOLI, MEMBER, IEEE, MAURIZIO FERLITO, AND ANDREA NERI

Abstract —Vector processors (supercomputers) can be effectively employed in MIC or MMIC applications to solve problems of large numerical size such as broad-band nonlinear design or statistical design (yield optimization). In order to fully exploit the capabilities of a vector hardware, any program architecture must be structured accordingly. This paper presents a possible approach to the "semantic" vectorization of microwave circuit design software. Speed-up factors of the order of 50 can be obtained on a typical vector processor (Cray X-MP), with respect to the most powerful scalar computers (CDC 7600), with cost reductions of more than one order of magnitude. This could broaden the horizon of microwave CAD techniques to include problems that are practically out of the reach of conventional systems.

I. Introduction

THE ADVENT OF monolithic circuits technology and, to a lesser extent, the ever increasing miniaturization of hybrid circuits generate a growing need for enhanced capabilities of microwave circuit design software. In fact, the limited (if any at all) amount of trimming available in such circuits makes for extremely accurate and reliable modeling and for systematic solutions to problems that have been traditionally tackled by semi-empirical approaches or even by trial-and-error techniques.

In the last few years, a considerable effort has been devoted to the modeling problem [1], so that fast and accurate computer models of most microwave components of common use are now available. On the other hand, analysis algorithms and program architectures have remained substantially unchanged with respect to those in use in the early seventies [2]. As a consequence, today's marketplace offers a choice of computer programs for the design of linear microwave circuits which are fast and accurate, especially if the models adopted allow the use of modern optimization algorithms based on analytic gradient evaluation [3]. However, general purpose systematic solutions to more advanced problems, such as nonlinear broad-band or statistical design, have not yet been found. This is mostly due to a lack of computational power, in the sense that the combination of conventional design programs and computer systems does not yield acceptable cost-to-performance ratios.

From this viewpoint, the exploitation of vector processors (such as Cyber 205, Cray X-MP, or similar) can considerably broaden the horizon of present-day microwave CAD. A first essential aspect is that using a vector processor does not simply mean resorting to a more powerful computer. On such a machine, the same set of operations can be performed with a relative speed roughly ranging from 1 to 10, depending on program architecture only. The key point is that all those operations that are similar in structure must be grouped together as far as possible, and then carried out in parallel (in the pipeline sense) by the specialized vector hardware. The principal aim of this paper is to show that this very basic concept can be effectively applied to microwave circuit design programs.

It is not difficult to understand why this is possible. Designing a microwave circuit by optimization is a highly repetitive job, in the sense that most of the CPU time is spent in executing structurally similar operations over and over again. For instance, a typical circuit may contain several microstrip sections, each requiring the same set of operations to be carried out with different numerical operands. This is true to a much greater extent when it comes to nonlinear designs. In such cases, the cost of one objective function evaluation is mostly due to the repeated analysis of the linear part of the network at a large number of frequencies; as an example, for a broad-band mixer, one has to consider the local oscillator harmonics and the sidebands generated by each input frequency, within a descrete set selected to cover the band of interest. A conceptually similar situation occurs in statistical design, e.g., evaluating the production yield by a Monte-Carlo method practically means analyzing the same circuit hundreds of times with variable circuit parameters. Thus, the nature of the problem inherently lends itself to vectorization.

This paper introduces a vectorized program architecture allowing the capabilities of vector processors to be fully exploited in microwave CAD. After presenting some general guidelines, a specific example of application is carried out and its performance is discussed in detail. It is shown that a vectorized code running on a supercomputer can work out a typical design at a cost more than one order of

Manuscript received April 16, 1985; revised July 29, 1985. This work was supported in part by the Italian Ministry of Public Education.

V. Rizzoli is with the Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna, Bologna, Italy.

M. Ferlito is with HS Elettronica Progetti s.r.l., Bologna, Italy.

A. Neri is with Fondazione Ugo Bordoni, Rome, Italy.

IEEE Log Number 8405819.

0018-9480/86/0100-0135$01.00 ©1986 IEEE

magnitude lower than its scalar counterpart running on today's most powerful scalar machines.

II. Scalar and Vectorized Architectures

In order to illustrate the vectorization strategy, we will consider one of the most powerful and best-known approaches to microwave circuit analysis, usually referred to as the "Subnetwork-Growth Method" (SGM) [2], [4]. According to this method, a general microwave circuit is thought of as being generated by the interconnection of a number of elementary building blocks (circuit components), and the design problem is reduced to a nonlinear optimization by performing the following steps:

1) the scattering matrices of all individual circuit components are computed at a given frequency;
2) the circuit components are interconnected according to a predetermined optimum sequence until the scattering matrix of the overall network is generated;
3) steps 1) and 2) are repeated at a number of discrete frequencies covering the band of interest;
4) the scattering matrices are used to derive the electrical performance; the latter is compared with the design goals and an objective function encompassing all the specifications is generated;
5) the objective function is minimized by a suitable optimization scheme.

The above algorithm in its usual implementation exhibits a markedly scalar behavior. In fact, SGM programs running on typical vector processors, such as the Cyber 205 and the Cray X-MP, have virtually the same performance in the scalar and vector modes, the speed-up due to automatic vectorization being as low as a few units percent. This clearly suggests that the efficiency of MIC design programs can be significantly increased by "semantic" vectorization only, that is, by thoroughly restructuring the analysis and design algorithms in such a way as to match the computational capabilities of the vector hardware. The repeated execution of steps 1) and 2) is usually responsible for a major fraction of the CPU time requirements (typically more than 90 percent); their relative importance is strongly dependent on the particular job, but is comparable on the average, so that they are both natural candidates for vectorization.

As a first important point, the traditional approach to broad-band network analysis summarized by steps 1)–3) must be abandoned. According to this scheme, a broadband analysis is generated by carrying out sequentially a set of complete single-frequency analyses. This is a typically scalar concept because consecutive operations are generally different from each other.

In a vectorized logic, a sequence of single-frequency analyses is changed into a single multifrequency analysis. This means that any physical parameter or numerical operand to be dealt with in evaluating the network performance is first replaced by the vector of the different values taken by the same quantity at all frequencies of interest. Each operation is then executed at all such frequencies, i.e., it acts on the entire vector operand, before the subsequent operation gets started. The memory requirements of the program are increased considerably in this way because one has to store the scattering matrices of the circuit components and the intermediate results generated by the SGM at all design frequencies. However, this usually does not represent a problem for a supercomputer, since large memories are always available in such machines (typically over 1 million 64-bit words). For the present application, it was found that a program allowing up to 128 frequencies can easily fit a one-million-words memory.

With the above approach, the length of all vector operations equals the number of design frequencies. This can provide reasonable speed-up factors in broad-band nonlinear design where several harmonics of each frequency of operation must be considered. On the other hand, in order to optimize the performance of a vector processor, it is generally convenient that the calculations to be carried out are grouped into the minimum number of vector operations with each having the maximum length. To enhance the degree of code vectorization, further actions can be taken in the following way.

Concerning step 1), we first note that a typical microwave circuit usually contains few kinds of circuit components, each one occurring several times in the network topology. As an example, let us consider the parallel-coupled filter illustrated in Fig. 1, which we will use as a benchmark throughout the paper. This circuit only includes two types of elements, i.e., the symmetric coupled-microstrip pair (numbers 1–6) and the lumped capacitance (numbers 7–18). It is quite obvious that the computation of the scattering matrix of all physically similar components requires the same sequence of operations to be carried out with different operands; such components may then be analyzed in parallel in the pipeline sense. This naturally leads to the concept of "supercomponent" or "vector component," defined as the set of all similar circuit components, each taken at all design frequencies. A pictorial representation of this idea is attempted in Fig. 2. As an example, the network of Fig. 1 is made of just two supercomponents: a vectorized coupled microstrip section of size $6N$ and a vectorized capacitance of size $12N$, N being the number of frequencies.

Thus, the combination of steps 1) and 3) of the scalar SGM can be converted into a very efficient vectorized procedure in the following way: each supercomponent—just one of each kind for a given network—is computed and stored by a dedicated subroutine to be called only once for each objective function evaluation. As a side effect, the code is also considerably improved from the scalar viewpoint, thanks to the elimination of a number of time-consuming procedures such as subroutine calls. This is by no means a minor point when dealing with vector processors: the CPU is so fast that the overall computational speed can be seriously limited by nonproductive operations such as data transfers to and from subroutines.

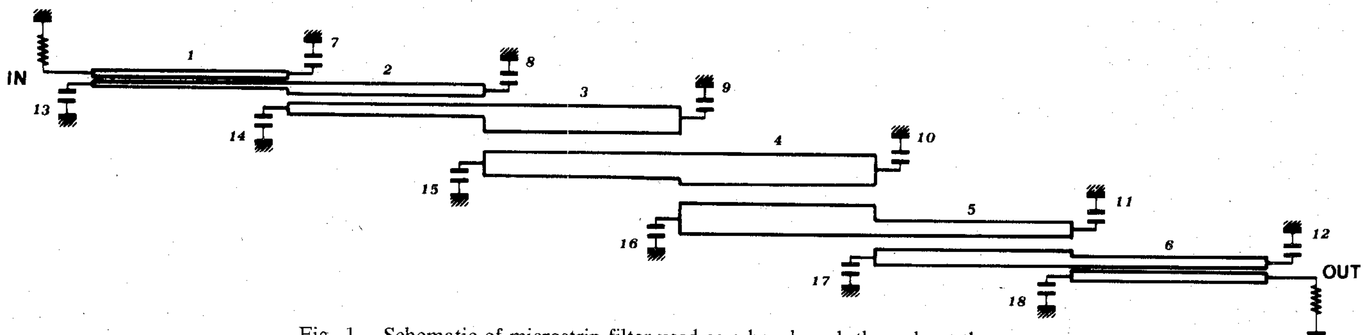

Fig. 1. Schematic of microstrip filter used as a benchmark throughout the paper.

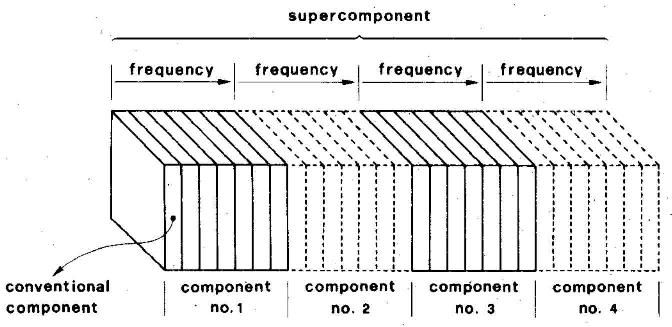

Fig. 2. Pictorial representation of the supercomponent concept.

As for step 2), we first recall that the computation of the scattering matrix of a microwave circuit by the SGM is obtained through a stepwise combination of couples of individual components according to a predetermined sequence [2], [4]. For the filter of Fig. 1, this is schematically illustrated in Fig. 3; in this case, the sequence is established in such a way that the connections leading to intermediate subnetworks with a minimum number of ports are carried out first. It is quite evident that the connections operated by the SGM can be grouped into sets of "topologically similar connections" (TSC). Two TSC's are such that the number of ports of both the combined and the resulting subnetworks are the same, regardless of their physical structure. Since the circuit components and the subnetworks generated by the SGM are always described in the same way (e.g., by the scattering matrix), any two TSC's clearly require the same sequence of operations.

Thus, the combination of steps 2) and 3) of the scalar SGM can be vectorized by application of the following simple rule: all TSC's which do not lead to recursion may be carried out in parallel (in the pipeline sense) at all design frequencies. This means that the sequential architecture of the scalar interconnection algorithm is replaced by a tree-like structure in its vectorized counterpart; for the filter of Fig. 1, this mechanism is illustrated in Fig. 4. As a result, most connections are carried out in the first few steps, with relatively long vector operations because of the large number of TSC's, and are thus very efficient, while only a limited number is performed in the final steps, with vector sizes equal to the number of frequencies. The execution speed is thus considerably enhanced.

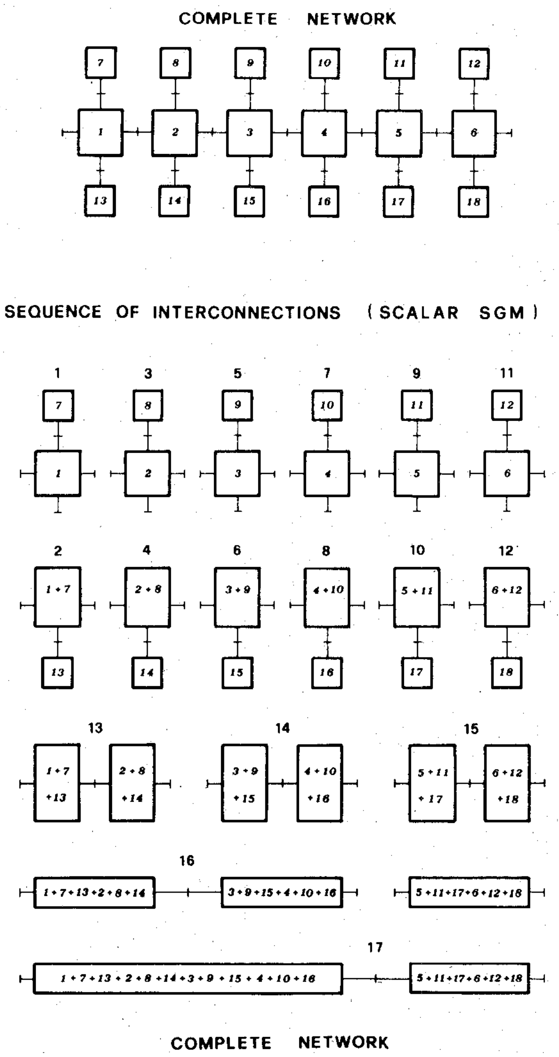

Fig. 3. Sequence of interconnections performed by the scalar subnetwork-growth algorithm.

The program architecture described in this section requires a large amount of topological preprocessing before entering the optimization loop. For instance, the circuit components must be reordered so that structurally similar ones are stored in contiguous memory locations, thus giving physical consistency to the supercomponent concept. The TSC's to be carried out in parallel must be recognized and the tree-like sequence of connections must be established, and so on. This may represent a difficult job from the programmer's viewpoint, but clearly has no appreciable effects on CPU time requirements.

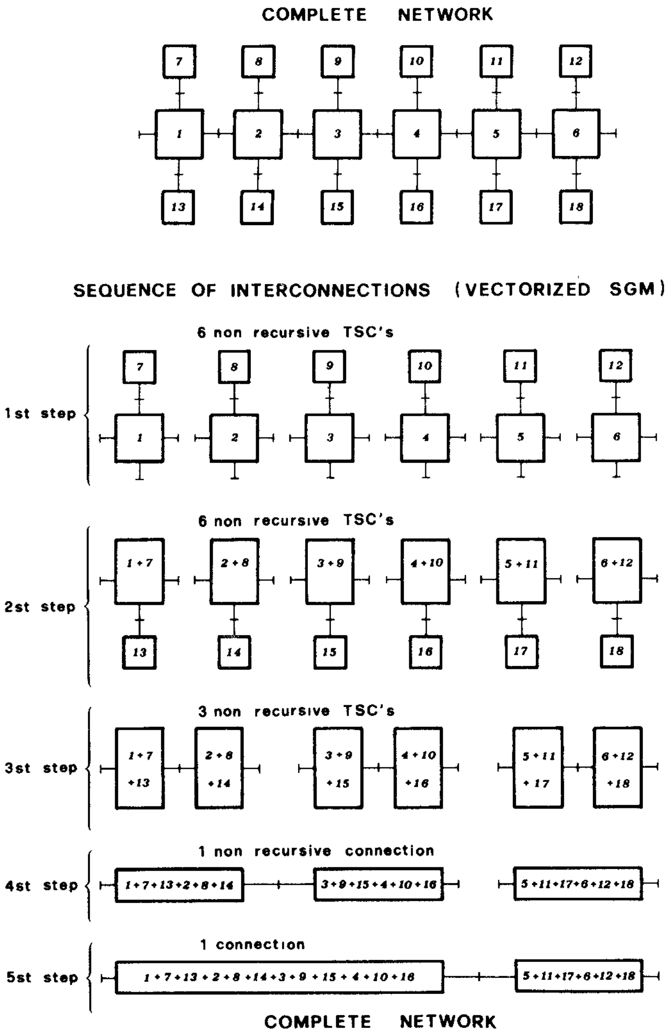

Fig. 4. Sequence of interconnections performed by the vectorized subnetwork-growth algorithm.

III. Performance of Vectorized Codes

To show the potential impact of vectorized codes on microwave circuit design, we present in this section a performance comparison. On one side we have a vectorized architecture based on the ideas discussed in Section II, running on two typical present-day supercomputers such as the two-pipeline Cyber 205 and Cray X-MP/12; on the other side we consider the scalar version of the same program running on a very powerful scalar mainframe such as the CDC 7600. The benchmark consists of a multifrequency analysis of the microstrip parallel-coupled filter depicted in Fig. 1. Three versions of the analysis program were written, each one very carefully optimized for a specific machine. Thus, to the best of the authors' capabilities, the numerical results given below represent good estimates of the kind of performance that the above mentioned computer systems can provide in microwave circuit design applications. Also, the analysis of a number of different topologies showed that such results may be considered as typical for a broad class of microwave circuits.

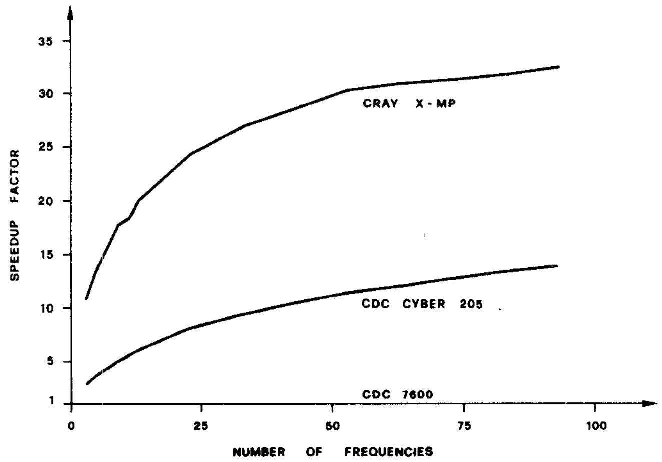

Fig. 5. Performance comparison between two supercomputers and the CDC 7600.

In Fig. 5, we plot the performance of the two vector processors against the number of frequencies, with the CDC 7600 taken as reference, i.e., its performance conventionally set to 1. Since the circuit analysis is mostly carried out by vector operations whose lengths are multiples of N, the supercomputers become more and more efficient as the number of frequencies increases. The absolute values of the speed-up factors are quite interesting; for instance, at $N = 50$, the Cray X-MP is about 30 times faster than the 7600, and is almost 11 times faster with a number of frequencies as small as three. This is a clear indication that the use of vectorized codes on supercomputers can considerably improve the cost-effectiveness of even conventional designs involving a limited number of frequencies. On the other hand, a speed-up factor of 30 is good enough for large-size designs involving many frequencies (such as broad-band nonlinear designs) to become practically accessible. As N becomes very large, a saturation is observed in the performance of both supercomputers; this is related to the fact that the computational speed approaches the machine asymptotic flop rate.

To understand how good a job was done in vectorizing the analysis program, we carried out a detailed examination of the peak performance reported in Fig. 5, namely, a speed-up factor of 32.6 (at $N = 93$) for the Cray machine. This figure results as a combination of the following three contributions:

hardware speed-up	2.65	
scalar optimization speed-up	1.90	
vectorization speed-up	6.48.	(1)

The "hardware speed-up" was measured by running on the Cray in scalar mode the code optimized for the CDC 7600. This factor is very similar to the ratio between the clock cycles of the two machines (9.5 and 27.5 ns, respectively), showing that the same job runs on the two computers in about the same number of clock cycles; it is thus essentially related to the more advanced technology of the Cray machine.

The scalar optimization speed-up is the performance ratio between the vectorized code and the scalar code, i.e., the one optimized for the CDC 7600, when both are run on the Cray in scalar mode (vectorization inhibited). It provides a quantitative check of the statement that vectorization often provides a good deal of scalar optimization, too, due to the elimination of a number of useless procedures. Generally speaking, the point is that the usual trade-off between memory occupation and CPU time can be pushed all the way towards computational efficiency in the supercomputer case because memory resources are abundant and cheap.

The last factor in (1) is just the conventional vector performance of the supercomputer, i.e., the CPU time ratio obtained when the same code is run with the vectorization option of the compiler switched off and on, respectively. According to available statistical evaluations of the Cray machine [5], a vector performance of 6.48 means that about 94 percent of the code is vectorized, which can be considered an excellent result [5]. As a final point, we note from Fig. 5 that the relative performance of the two supercomputers approaches the ratio between their clock cycles as the number of frequencies becomes large. These cycles are 20.5 ns for the Cyber 205 and 9.5 ns for the Cray X-MP (with a ratio of 2.105), while the performance ratio given by Fig. 5 is 2.35 at $N = 93$. This means that practically the same vector efficiency is achieved on the two machines despite the very different architectures of their vector hardware. At a number of frequencies much larger than 100, the performance comparison would probably be more favorable to the Cyber 205; however, $N > 100$ may be considered very unlikely for microwave applications. On the other hand, when the number of frequencies is small, the Cray machine offers a superior vector performance (the ratio goes up to 3.89 for $N = 3$) because its register-to-register vector operations can handle short vectors more efficiently than the memory-to-memory approach of the 205. In this respect, the best supercomputer for running microwave applications should be the new Fujtsu VP200 [6], because of its unique capability to dynamically allocate the length of its vector registers. In fact, the number of frequencies in microwave design jobs can change randomly from case to case. Unfortunately, it was not possible for the authors to test their programs on this kind of machine.

Converting CPU performance information into cost information in a meaningful way is usually very difficult because cost policies are obviously quite variable. Nevertheless, in this particular case, we feel that a cost comparison can be attempted for the very simple reason that there is just one supercomputer installed in our country at the time of this writing: a Cray X-MP/12 located at the computer center CINECA, near Bologna.[1] This system was acquired under financial support of the Italian Ministry of Public Education and may be accessed by any kind of public or private organization throughout the country via the telephone network. It now acts as a national resource of computational power available to the entire scientific and technical community. Thus, the figures are significant in absolute in Italy, and may represent a useful reference for microwave engineers from abroad.

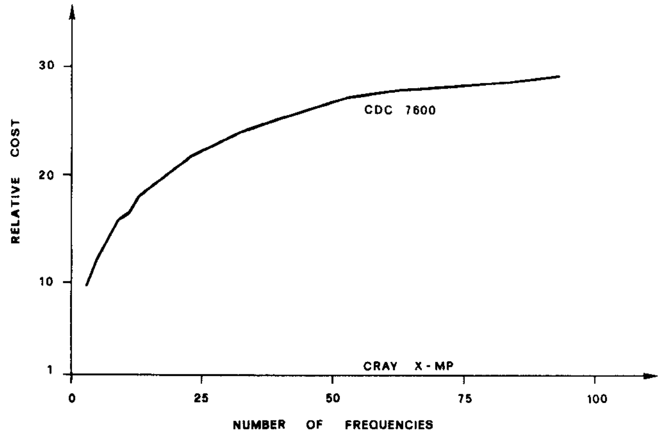

Fig. 6. Cost comparison between the CDC 7600 and the Cray X-MP/12 of CINECA.

In Fig. 6, we compare the costs of running the same multifrequency analysis of the microstrip filter on the CDC 7600 (scalar code) and on the Cray X-MP (vectorized code) of CINECA at current rates (July 1985). The Cray cost is used as reference, and conventionally set to one, for ease of comparison. The beneficial effects of using a vectorized code on the supercomputer are quite evident with cost reductions ranging from a factor of the order of nine with a small number of frequencies to more than 28 in the case of large-size problems ($N > 70$).

IV. Application to Nonlinear Design

In this section, we discuss briefly the perspective application of vectorized codes for microwave circuit analysis to the solution of a typical large-size problem such as broadband nonlinear design.

The peculiar aspect of a nonlinear circuit is that its electrical performance depends on both circuit topology and steady-state electrical regime. Thus, the designer must simultaneously find a network and a set of voltage and current waveforms satisfying both the network equations and the design goals. Nonlinear design techniques are not as well established as their linear counterparts; for this reason the following discussion is essentially based on the authors' own experience and is mainly intended to present some preliminary results and qualitative guidelines. As a starting point, let us refer to a recently proposed numeric approach [7] allowing a nonlinear design problem to be reduced to an optimization via the following steps.

1) The circuit is subdivided into a linear part and a nonlinear part (named the "device") for which an analytic time-domain description is available. The set of the design variables thus includes the unknown parameters of the linear subnetwork and the voltage harmonics at the device ports.

2) The linear subnetwork is analyzed at the design frequencies and at all required harmonics.

3) Using the voltage harmonics and the results of 2), the network performance is computed and compared with the design specifications, thus generating a first contribution to the objective function.

[1] CINECA, Centro di Calcolo Interuniversitario dell'Italia Nord-Orientale, Via Magnanelli 6/3, 40033 Casalecchio di Reno, Bologna, Italy.

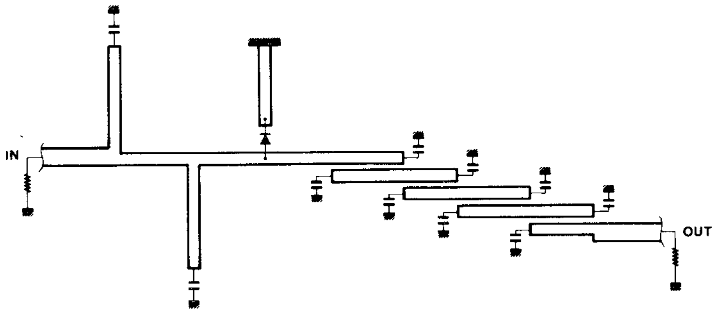

Fig. 7. Schematic of microstrip frequency multiplier by four.

4) The current harmonics at the device ports are computed twice, that is, i) by a conventional frequency-domain analysis of the linear subnetwork, and ii) by the time-domain device equations making use of the fast Fourier transform (FFT). The rms difference between the two results, the harmonic-balance error, is used as a second contribution to the objective function.

5) The overall objective function is simultaneously minimized with respect to all unknowns by a suitable optimiztion scheme, until a minimum close enough to zero is determined.

Steps 3) and 4) usually require only a limited number of algebraic operations and a few applications of the FFT, which may be carried out most efficiently by existing library subprograms. Thus, in an average nonlinear design, most of the CPU time is spent on step 2). It is then possible to give a rough estimate of the relative cost of typical nonlinear and linear design problems. Given a network topology and a number of variable circuit parameters, the number of single-frequency linear analyses required to complete a nonlinear design is usually much larger than for its linear counterpart. This is due to the combined effect of three factors, that is, i) an increased number of unknowns (because the harmonics must also be found), ii) a more critical optimization problem (because an absolute minimum must be reached), and iii) the need to consider a number of harmonics of each design frequency. On the other hand, the large number of frequencies to be dealt with (especially in the broad-band case) makes the use of vectorized analysis programs extremely attractive. As an example, running on a supercomputer a seven-frequency design taking six harmonics into account and requiring 10 000 single-frequency linear analyses, will cost only 3.5 times as much as a scalar design requiring 100 circuit analyses with the same topology, according to Fig. 5, in terms of CPU time.

The problem of gradient evaluation deserves some special attention. Some of the most advanced and sophisticated optimization methods that are presently available [3] rely upon analytic gradient evaluation. On the other hand, modern highly accurate models of many microstrip components of common use (e.g., [8], [9]) are so complicated that an analytic computation of the derivatives hardly seems a viable approach. This is true to an even greater extent when the circuit contains certain kinds of components that can only be computed with acceptable accuracy by electromagnetic simulation (e.g., two-dimensional microstrip devices such as the radial stub [10]). In the nonlinear case, the analytic approach is out of the question because the computation of the objective function involves all-numeric procedures such as the FFT.

When it comes to numeric gradient evaluation, one can take advantage of the large central memory available in a supercomputer to obtain a further significant increase of the computational speed with respect to a scalar machine. In fact, the ability to store the component matrices and the intermediate results of the SGM at all frequencies (Section II) can be used to effectively reduce the number of required operations. In the case of quasi-Newton methods, this was found to provide on the average an additional speed-up factor of 1.4 for the Cray code as compared to its CDC 7600 equivalent. From Fig. 5, the overall speed-up factor is then found to range between 15.2 and 45.6 (depending on N), while Fig. 6 yields a cost benefit from 13.6 up to 41.1.

As a final point, we consider briefly the stability problem. Nonlinear circuits contain pumped nonlinear devices such as diodes, FET's, and so on, and are thus natural candidates to parametric instabilities and spurious oscillations. Analyzing the stability of a nonlinear circuit is a difficult job because one has to deal with the effects of perturbing a large-signal nonsinusoidal steady-state regime; thus, the resulting numerical problem may be too large and time-consuming for systematic applications. Once again, vector processors can help to work out a general solution of practical usefulness. For instance, according to a recent work [11], generating the Nyquist stability plot of a typical nonlinear circuit operated in steady-state regime may require a CPU time of the order of one second on a Cyber 205, which is very acceptable for practical purposes.

As an example, we consider the microstrip circuit schematically illustrated in Fig. 7, which can be considered a typical specimen of nonlinear microwave network. This circuit was designed to act as a broad-band frequency multiplier by four, with an 800-MHz output band centered around 18 GHz. The diode used was a Microwave Associates "dual-mode" varactor whose voltage–capacitance characteristic is provided in the manufacturer's catalog. The broad-band design used five frequency points and 14 circuit variables, and required less than 36 CPU seconds on a Cray X-MP, of which around 30 were spent in the linear subnetwork analysis. The cost of the same design on a CDC 7600 (scalar code) would have been around 1050 CPU seconds, with an overall speed-up factor of about 30.

The multiplier was found to be parametrically stable and spurious-free throughout the band of interest; the time required to compute a Nyquist stability plot following the method described in [11] was approximately 0.5 s for each frequency point.

V. Conclusion

The use of vectorized codes running on supercomputers presently provides some of the best chances of developing powerful CAD tools allowing general-purpose nonlinear designs to be performed at the same cost levels as today's linear applications. Some basic concepts leading to highly efficient vectorized program architectures for linear and nonlinear microwave circuit design have been outlined in the present paper.

The application software described herein is still at the prototype level, and the results have essentially the meaning of a proof of feasibility. On the other hand, a very refined user-oriented scalar program for single-frequency nonlinear design has been in use for the last few years with excellent results [7]. Thus, the available know-how provides a suitable background, in view of the development of a fully vectorized nonlinear broad-band design tool, to become as handy and powerful as currently available linear CAD programs.

References

[1] R. H. Jansen, "Computer-aided design of hybrid and monolithic microwave integrated circuits," in *Proc. 13th European Microwave Conf.* (Nürnberg, W. Germany), Sept. 1983, pp. 67–78.

[2] V. A. Monaco and P. Tiberio, "Computer-aided analysis of microwave integrated circuits," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-22, pp. 249–263, Mar. 1974.

[3] J. W. Bnadler *et al.*, "Case study in passive circuit CAD," in *Proc. 14th European Microwave Conf.* (Liège, Belgium), Sept. 1984, pp. 740–745.

[4] K. C. Gupta, R. Garg, and R. Chadha, *Computer-Aided Design of Microwave Circuits.* Dedham, MA: Artech House, 1981, pp. 338–353.

[5] G. Erbacci and U. Fabbri, "Vectorization and optimization of Fortran programs for the Cray X-MP/12," CINECA-Monografie Cray no. 4, pp. 45–48, Jan. 1985.

[6] *DPMA Int. Conf. on Supercomputers*, Paris, France, Apr. 1984.

[7] V. Rizzoli, A. Lipparini, and E. Marazzi, "A general-purpose program for nonlinear microwave circuit design," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-31, pp. 762–770, Sept. 1983.

[8] M. Kirschning and R. H. Jansen, "Accurate wide-range design equations for the frequency-dependent characteristics of parallel coupled microstrip lines," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-32, pp. 83–90, Jan. 1984.

[9] I. Wolff and G. Kibnuka, "Computer models for capacitors and inductors," in *Proc. 14th European Microwave Conf.* (Liège, Belgium), Sept. 1984, pp. 853–858.

[10] F. Giannini, R. Sorrentino, and J. Vrba, "Planar circuit analysis of microstrip radial stub," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-32, pp. 1652–1655, Dec. 1984.

[11] V. Rizzoli and A. Lipparini, "General stability analysis of periodic steady-state regimes in nonlinear microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-33, pp. 30–37, Jan. 1985.

Vittorio Rizzoli (M'79) was born in Bologna, Italy, in 1949. He graduated from the School of Engineering, University of Bologna, Bologna, Italy, in July 1971.

From 1971 to 1973, he was with the Centro Onde Millimetriche of Fondazione Ugo Bordoni, Pontecchio Marconi, Italy, where he was involved in a research project on millimeter-waveguide communication systems. In 1973, he was with the Hewlett-Packard Company, Palo Alto, CA, working in the areas of MIC and microwave power devices. From 1974 to 1979, he was an Associate Professor at the University of Bologna, teaching a course on microwave integrated circuits. In 1980, he joined the University of Bologna as a full Professor of Electromagnetic Fields and Circuits.

His current research interests are in the fields of MIC and MMIC, with special emphasis on nonlinear circuits. He is also heading a research project aimed at the development of vectorized software for microwave circuit design applications.

Maurizio Ferlito was born in Bologna, Italy, in 1955. He received a degree in electronic engineering from the University of Bologna, Bologna, Italy, in 1981.

After graduating, he joined Fondazione G. Marconi, Pontecchio Marconi, Italy, where he obtained a research grant under which he carried out a study concerning the theoretical and experimental characterization of MIC passive components. In the same period, he collaborated with the Department of Electronics, University of Bologna and Elettronica S.p.A., Rome, Italy, in the field of automated measurements on microwave planar lines and discontinuities. In 1983, he obtained a new grant sponsored by Italtel S.p.A., from Fondazione G. Marconi, in order to carry out research on the computer-aided statistical design of microwave circuits. Since 1985, he has been with HS Elettronica Progetti s.r.l., Bologna, Italy.

Andrea Neri was born in Bologna, Italy, in 1957. He graduated in electronic engineering from the University of Bologna, Bologna, Italy, in July 1981.

In 1983 and 1984, he obtained research grants issued by Fondazione G. Marconi, Pontecchio Marconi, Italy, and Selenia S.p.A., Rome, Italy, to work on dielectric resonators and their applications in MIC. In 1985, he joined Fondazione U. Bordoni, Rome, Italy, where he is currently involved in research on nonlinear microwave circuit design. His main fields of interest are the characterization of microwave components by means of electromagnetic methods and the application of supercomputers in MIC and MMIC design.