

Parallel Processing Application to Nonlinear Microwave Network Design

M. I. SOBHAY, MEMBER, IEEE, AND Y. A. R. EL-SAWY

Abstract—One of the objectives of this paper is to introduce microwave network designers to an important new development in computer-aided design techniques. The paper describes how parallel processing is applied to the CAD of nonlinear microwave circuits. The advantage of parallel processing is the significant reduction in computational time it offers, such that optimization becomes feasible even on a desktop computer. The developed programs run on an AT desktop with one Transputer board capable of concurrent processing speeds of over 80 MIPS at a processor speed of 20 MHz. A new representation of microwave and nonlinear circuits has been developed to suit the required parallelism. Applications to the analysis of nonlinear amplifiers and frequency multipliers are described.

I. INTRODUCTION

THERE ARE A number of available computer programs for analyzing linear and nonlinear microwave networks [2]–[4]. Both time-domain and harmonic balance methods are based on serial processing. They both require powerful computers, and no optimization is offered due to the inordinate time required. In order to reduce the computational time and hardware requirements, a new approach to the circuit simulation is required. This paper describes such an approach, one which is suitable for parallel processing. The process does not require iteration when solving the nonlinear network. Furthermore, each circuit element has an identifiable process and all the processes work in parallel. These properties result in a very efficient algorithm that can be mounted on a desktop AT computer with a Transputer (trademark of INMOS Ltd.) board. The algorithm also offers the possibility of optimization due to the improvement in computational speed and the fact that each element is assigned an identifiable process. Parallelism can be featured independently in the mathematical algorithm, the programming language, and the hardware. To obtain the full advantage of parallel processing all three stages of the computer-aided design procedure should incorporate some form of parallelism.

II. THE MATHEMATICAL ALGORITHM

A general nonlinear microwave network contains the following elements:

- linear resistors, voltage- and current-controlled nonlinear resistors, and parametric resistors;

- linear and nonlinear inductors and capacitors;
- linear and nonlinear controlled sources;
- independent sources;
- distributed delay elements (transmission lines, microstrips, etc.).

Any CAD procedure has to satisfy both Kirchhoff's laws and the I – V relations of the elements (linear, nonlinear, integral, differential, delay, etc.). Instead of representing the network by using circuit diagrams, the new theoretical development derives a signal flow or simulation diagram for each network and can be summarized, in a simple form, as follows.

After defining a network tree, Kirchhoff's laws are expressed in the hybrid form

$$\begin{bmatrix} i_t \\ v_c \end{bmatrix} = \begin{bmatrix} 0 & D \\ -D^T & 0 \end{bmatrix} \begin{bmatrix} v_t \\ i_c \end{bmatrix} \quad (1)$$

where the subscripts t and c refer to the tree and cotree elements respectively and the matrix D is the dynamical transformation matrix [1]. Equation (1) is a hybrid form of Kirchhoff's laws, with some elements represented by Kirchhoff's first law and some by the second law.

The I – V relations are given by

$$\begin{aligned} i_t &= f_1(v_t) \\ v_c &= f_2(i_c) \end{aligned} \quad (2)$$

where f_1 and f_2 are general functions that could be nonlinear and/or differential.

Fig. 1 shows a block simulation diagram of the circuit, which is a representation of (1) and (2). This process is very general and can be applied to any linear or nonlinear network.

The advantage of this formulation is that each element in the circuit is considered a "process" and all processes together with Kirchhoff's first and second laws operate in "parallel." This is certainly not the case when trying to solve a circuit from its circuit diagram, as all the I – V relations in the circuit are interdependent and it is very difficult to define any parallelism.

Fig. 2 clarifies the difference between the two approaches. The simple rectifier circuit shown in Fig. 2(a) is represented by the simulation diagram of Fig. 2(b). In Fig. 2(b), the processes can be clearly defined and are realistically simulated on parallel processors.

Manuscript received March 28, 1989; revised June 26, 1989.
The authors are with the Electronic Engineering Laboratories, University of Kent at Canterbury, Canterbury, Kent, CT2 7NT, U.K.
IEEE Log Number 8931088.

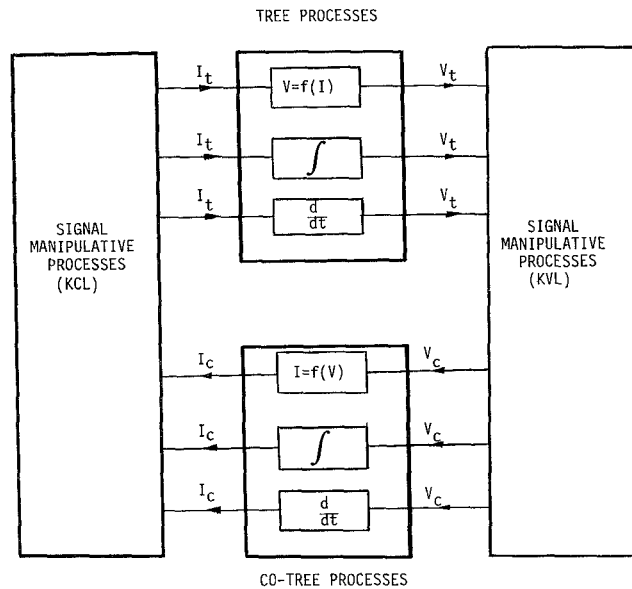


Fig. 1. Block simulation diagram.

The advantages of this formulation are:

- 1) Either a parallel or a sequential algorithm can be written to analyze the system.
- 2) No iterations are required to obtain the solution.
- 3) Each element has a clearly defined process. Thus varying the element values to optimize the network response is easily achieved without reformulating the equations.
- 4) There are no restrictions on either the topology or the type of process. Each process can be a linear, nonlinear, or any desired function.
- 5) The overall system can be easily configured on a number of Transputers.

The above formulation has been implemented on a system of Transputers. Fig. 2(c) shows the results of the simple rectifier circuit shown in Fig. 2(b).

III. BASIC PROCESS

A library of processes had to be developed to simulate all the possible circuit elements together with basic signal manipulative processes such as addition, subtraction, and fan-out.

1) *Lumped and Nonlinear Elements:* All linear and nonlinear elements can be represented as shown in Fig. 3(a). The inputs to the process are currents and voltages on the element itself (designated as the input i) or any other signal in the network (designated as control c). The output o is given by the function $f(i, c)$. This function can be a simple multiplication (for resistors or conductors), integration or differentiation (for capacitors and inductors), or any other linear or nonlinear function. Thus modeling any active or passive device is achieved by determining the input i , the control c , and the function $f(i, c)$.

2) *Distributed Elements:* Although distributed elements can also be represented by the general process shown in

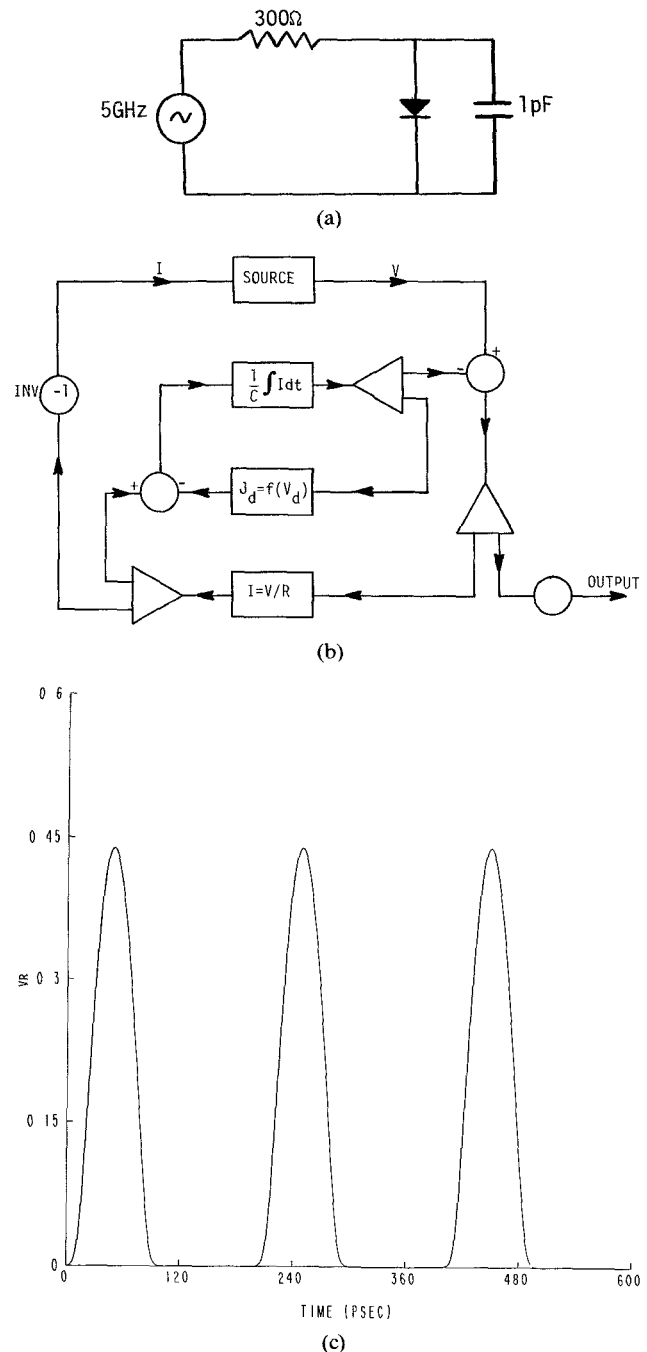


Fig. 2. Rectifier circuit and response from Transputer. (a) Rectifier circuit. (b) Parallel processes of rectifier circuit. (c) Response in time domain.

Fig. 3(a), it is useful to give more details of the process involved. An example is given in Fig. 3(b), which is a process representing a lossless transmission line. The inputs to the process are the input and output voltages V_1 and V_2 of the transmission line, and the outputs from the process are the currents I_1 and I_2 . Other combinations are also possible.

3) *Signal Manipulation Process:* The three signal manipulation processes are shown in Fig. 3(c)–(f). These are the signal addition (or subtraction), the delta (or fan-out) process, and a delay process.

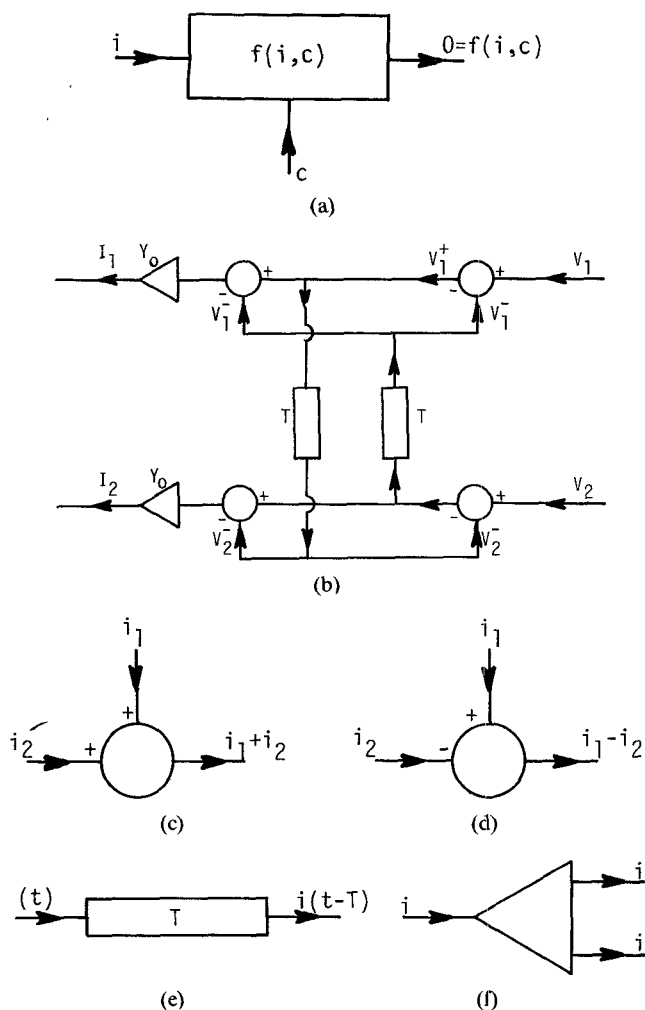


Fig. 3. Basic processes. (a) Linear or nonlinear process. (b) Distributed elements. (c) Signal addition. (d) Subtraction. (e) Delay. (f) Delta or fan-out.

4) *Probe Process*: For interactive testing of the circuit, the user may wish to attach a “probe” to any point in the circuit to inspect the signal at that point.

IV. THE “MAESTRO” PROCESS

Together with the basic processes, a controlling “Maestro” process was written to assume overall control for running the program, ensuring convergence and establishing the step size. The resulting step size ΔT is specified by the user as the time interval for which the results are required. All the sources are stepwise approximated with a time step ΔT . However, solving the network with an interval ΔT may not result in a convergent solution. The Maestro process checks the convergence by calculating the first- and second-order moments of the response and determines the solution step size δt as the largest time step for which a solution is convergent. It also checks whether the solution has reached a steady state by checking the difference between successive solutions. If the difference is smaller than a predetermined error, the time is advanced to the end of the result step ΔT . This process ensures the

convergence of the results and maximizes the speed of the program.

V. THE PROGRAMMING LANGUAGE

Sequential programming languages deal with processes as a number of statements to be executed strictly in sequence. They are not well equipped to construct programs for multiple processor systems because their very design assumes the sequential execution of instructions.

Some conventional sequential programming languages (such as C and Fortran) have been modified to allow concurrent programs to be written, but ensuring that concurrent parts of the program are synchronized is the programmer’s responsibility. This leads to a more difficult programming than ordinary sequential programming.

OCCAM [5], [6] is a simple programming language which allows the user to express and reason about parallel designs. It is based upon the concept of the “process.” OCCAM parallel processes communicate only by synchronized unbuffered messages along dedicated “point-to-point” channels. Each process inputs in parallel from all its input channels, performs some action (activity) inside the process, and outputs in parallel to all its output channels.

OCCAM is the first language to be based upon the concept of parallelism, in addition to sequential execution and to providing automatic communication and synchronization between concurrent processes. The choice of OCCAM as the programming language for nonlinear microwave networks contributes to the impressive programming efficiency obtained.

VI. HARDWARE

The program has been implemented on a desktop AT computer with the addition of five Transputers. The Transputer [7], [8] is a processor which implements the process model of communication embodied in its native parallel programming language, OCCAM. Each Transputer has four hardware links, each of which can be mapped into a bidirectional pair of OCCAM channels. Any number of Transputers can be used. However, the system of five Transputers was found adequate for good programming efficiency. One of the five Transputers was used as a “host” to act as an interface between the user and the remaining four-Transputer network, which carries out the execution of the algorithm. The host Transputer has 2 Mbytes of dedicated DRAM and each of the other four Transputers has 256 kbytes of dedicated DRAM. Theoretically, the computation time is inversely proportional to the number of Transputers used. In practice, the efficiency of configuring the network and the finite communication time between Transputers will affect the reduction in time.

VII. CONFIGURATION OVER A TRANSPUTER NETWORK

The OCCAM program is easy to distribute over a Transputer network of arbitrary size by simply partitioning the

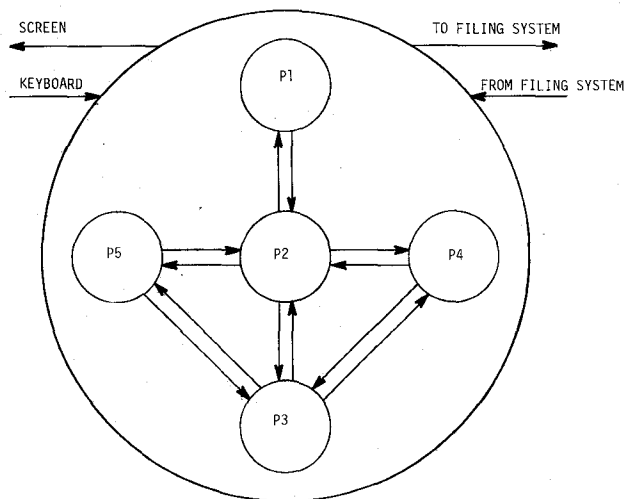


Fig. 4. The main OCCAM program.

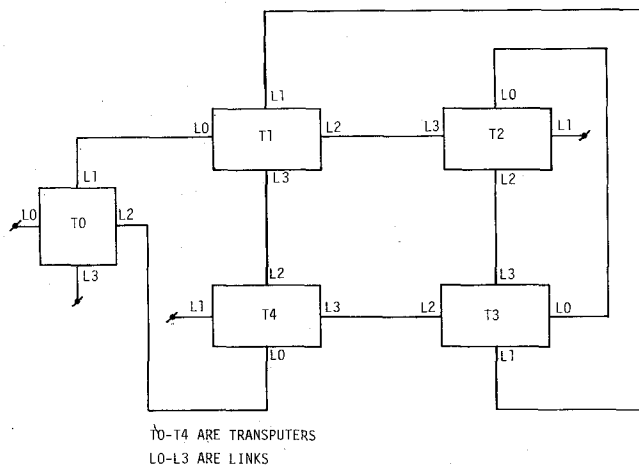


Fig. 5. Five-Transputer network (T414B). T1-T4 are Transputers. L0-L3 are Links.

network of processes. Because of the parallel design of the simulation there is great flexibility as to how this can be done. Almost any partition from all the processes on one Transputer down to one process per Transputer is possible (although perhaps not sensible).

Since inter-Transputer links (like the OCCAM communication channels that they implement) are point-to-point, the size of the network is never limited by the usual contention problems which arise from more normal methods of combining many processors (e.g. on a common bus).

The individual OCCAM processes of the parallel processing scheme shown in Fig. 1 are partitioned into five main processes, shown in Fig. 4, to be configured on the five Transputer networks shown in Fig. 5. The main five processes are as follows:

- P1 is the Maestro process and the input/output procedures.
- P2 includes all the TREE processes.
- P3 includes all the COTREE processes.

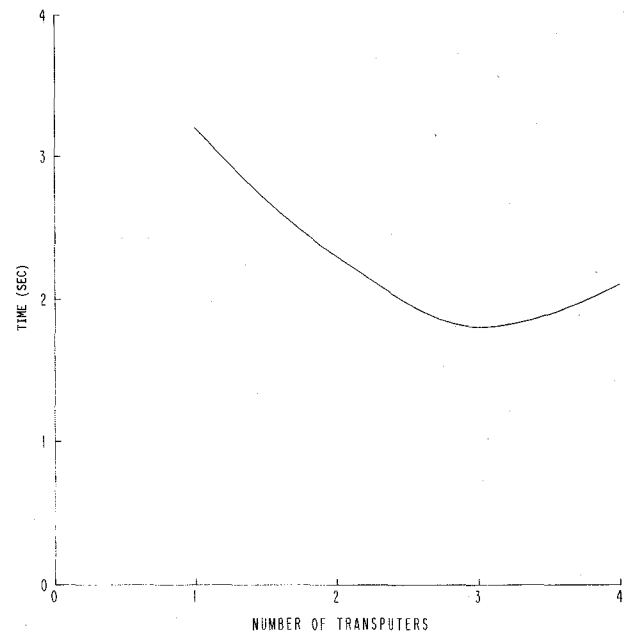


Fig. 6. Speedup curve for Transputer network.

P4 includes all the signal manipulation processes of the KVL side (shown in Fig. 1).

P5 includes all the signal manipulation processes of the KCL side (shown in Fig. 1).

VIII. PERFORMANCE MEASURES OF CONFIGURATION OVER A TRANSPUTER NETWORK

Configuration of our developed program over a Transputer network can achieve, at maximum efficiency, no more than a linear speedup, that is, N Transputers can at best run the program in $1/N$ of the time it takes for one Transputer to run it. However, this speedup is not achieved for two main reasons.

- 1) The Transputers are required to synchronize, at certain instances during the processing, in order to transfer data between each other. This results in some Transputers waiting and being idle while others reach the synchronization point.
- 2) The communication time needed to transfer data, over the hard links, from one Transputer to another affects the reduction in time.

Fig. 6 shows the speedup curve of the nonlinear amplifier shown in Fig. 7(a). The amplifier has been configured over an array of Transputers of increasing size, starting with one Transputer up to four Transputers.

Fig. 6 shows that the time decreases as the number of Transputers increases up to three and then increases as the fourth Transputer is added to the array.

The efficiency (η) of configuring over N Transputers is given by

$$\eta = \frac{\text{time taken over one Transputer}}{\text{time taken over } N \text{ Transputers} \times N} \times 100.$$

TABLE I

NUMBER OF TRANSPUTERS	TIME (SEC.)	η
1	3.2	
2	2.3	70%
3	1.8	60%
4	2.1	38%

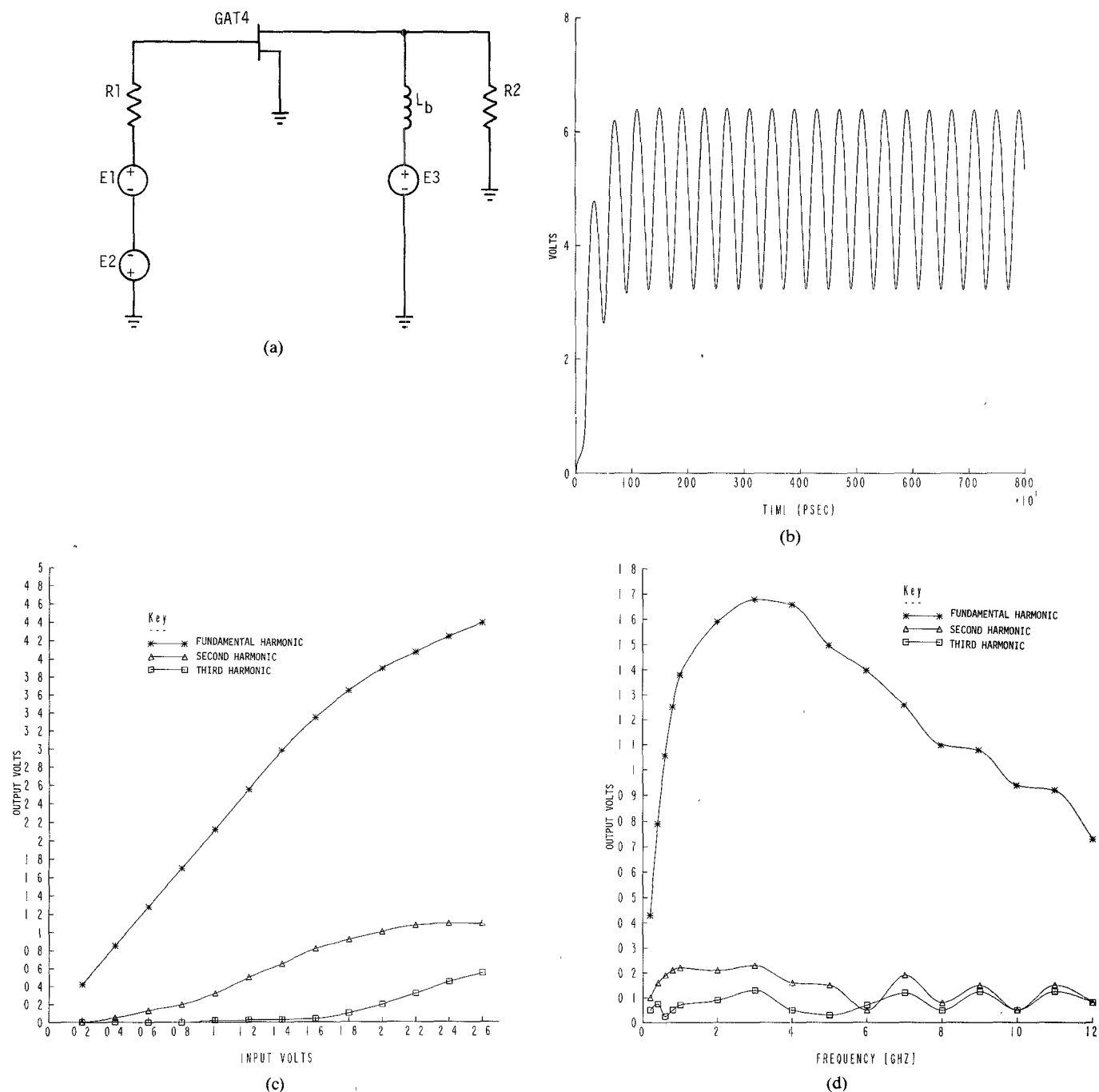


Fig. 7. (a) Nonlinear amplifier. (b) Response in time domain of nonlinear amplifier. (c) Saturation characteristics of the amplifier. (d) Large-signal frequency response of the amplifier.

Table I shows the efficiency of configuring the nonlinear amplifier over a Transputer network. The performance analysis shows that there must be a compromise between the problem size and the number of Transputers used in the array to achieve a high performance.

IX. APPLICATIONS

The developed program has been applied to several microwave networks. In each case the results were compared to those obtained by ANAMIC [3], which is a time-domain simulation program. Although ANAMIC can run on a wide range of workstations, the present results were obtained using a VAX 8500 mainframe.

The following summarizes some of the results obtained.

1) *Nonlinear Amplifier*: The amplifier circuits shown in Fig. 7(a) were analyzed by both programs. Identical results were obtained with ANAMIC taking 3.5 s on the VAX 8500 and the parallel algorithm taking 3.2 s on the AT computer. The advantage is impressive, especially when realizing that we are comparing a powerful mainframe with a small AT computer and that further development of the new algorithm is expected to reduce the time even further. The addition of more Transputers will also reduce the computational time.

The initial result from the program is the waveform in the time domain as shown in Fig. 7(b). The results could be analyzed further to obtain the saturation characteristics and the frequency response as shown in Fig. 7(c) and (d).

2) *Frequency Doubler*: The MESFET 4–8 GHz frequency doubler circuit shown in Fig. 8(a) was analyzed from 0 to 8000 ps at 2 ps intervals and took 130 s of computer time. Further analysis of the results gives a 5.5 dB conversion gain for the doubler. The waveform and the frequency response are shown in Fig. 8(b) and (c), respectively.

X. CONCLUSIONS

A powerful new algorithm has been developed for analyzing nonlinear microwave networks using parallel processors. The system offers the possibility of analyzing and optimizing these circuits using low-cost desktop computers and eliminates the need for workstations or mainframes. By identifying every circuit element with a process, efficient optimization is feasible. A further increase in speed is possible by the simple addition of more Transputers without any further modification of the software.

REFERENCES

- [1] M. I. Sobhy and M. H. Keriakos, "Computer aided analysis and design of networks containing commensurate and non-commensurate delay lines," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-28, pp. 348–358, Apr. 1980.
- [2] V. Rizzoli *et al.*, "User-oriented software package for the analysis and optimisation of non-linear microwave circuits," *Proc. Inst. Elec. Eng.*, pt. H, vol. 33, pp. 635–640, Oct. 1986.
- [3] M. I. Sobhy and A. K. Jastrzebski, "Computer-aided design of microwave integrated circuits," in *Proc. 14th European Microwave Conf. (Liege)*, 1984, pp. 705–710.
- [4] M. I. Sobhy and A. K. Jastrzebski, "Direct integration methods of non-linear microwave circuits," in *Proc. 15th European Microwave Conf. (Paris)*, Sept. 1985, pp. 1110–1118.
- [5] D. Pountain and D. May, *A Tutorial Introduction to OCCAM Programming*, INMOS Company, 1988.
- [6] R. D. Dowsing, *Introduction to Concurrency Using OCCAM*. New York: Van Nostrand Reinhold, 1988.
- [7] A. Carling, *Parallel Processing, the Transputer and OCCAM*. Wilmglow: Sigma Press, 1988.
- [8] *IMS T414 Transputer*, INMOS Company, Product Data, 1988.

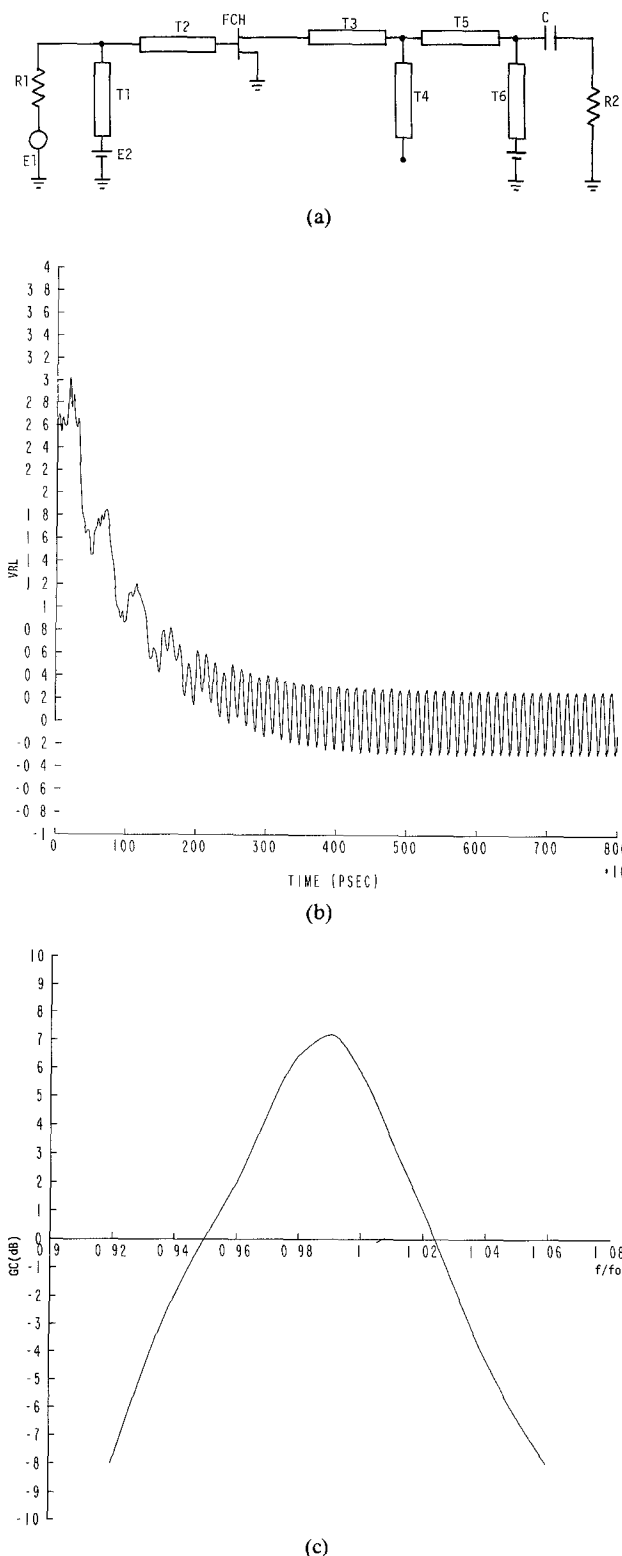


Fig. 8. (a) Circuit diagram of 4–8 GHz doubler. (b) Response in time domain of the doubler. (c) Frequency response of the doubler.



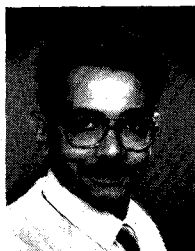
M. I. Sobhy (M'60) received the B.Sc. degree in electrical engineering from the University of Cairo, Cairo, Egypt, in 1956 and the Ph.D. degree from the University of Leeds, Leeds, England, in 1966.

He was a Teaching Assistant in the Department of Electrical Engineering at the University of Cairo until 1962, when he joined the University of Leeds, first as a research student and later as a Lecturer working on microwave ferrite devices. In 1966 he joined Microwave Associates Ltd., Luton, England, as a Research Engineer, where he worked on the development of microwave solid-state devices. He joined the University of Kent at Canterbury, Canterbury, England, in 1967, where he is now leading a research group engaged on projects on solid-state devices and microwave circuits. He is also a consultant to a number of industrial establishments.

Dr. Sobhy has published more than 60 papers in the fields of microwave circuits, computer-aided design of nonlinear circuits, digital

filters, switched capacitor filters, and microwave solid-state devices. He is currently Director of the Electronic Engineering Laboratories at the University of Kent. Dr. Sobhy is a Fellow of the Institute of Electrical Engineers, London, England.

✱



Y. A. R. El-Sawy was born in Cairo, Egypt, in 1950. He received the B.Sc. and M.Sc. degrees from the Military Technical College, Cairo, in 1973 and 1986, respectively.

He joined the teaching staff at the Military Technical College in 1978. He has been on leave since 1987, working on a Ph.D. project at the University of Kent, Canterbury, U.K. He is currently working on computer-aided circuit design techniques using parallel processing.