УДК 004.002

Melnyk V.
Lutsk national technikal university

CIRCUIT SIMULATOR THROUGH UNIX SOCKET CONNECTION WITH PARALLEL WAVEFORM RELAXATION

Abstract

A circuit simulator with parallel waveform relaxation (PWR) has been made to be effective through distributed computing. Its work is comparable with other PWR using parallel computing [1]. It is more flexible, economical and resource independent.

KEY WORDS: circuit simulator, waveform relaxation, parallel computing, socet, UNIX, network
Fig. 8., Ref. 5.

Introduction

Parallelization of both direct method and the relaxation method [2] has been researched on different computing platforms for many years. Another way of parallelization - distributed computing thought UNIX Berkeley socket interface [3] for the client and server model has been developed (fig. 1). Such a computer model contains a number of node computers that are called client and a gateway which is used to share data and to synchronize execution among clients. Thus, the gateway should have a large memory size for all clients to access; Share Memory applies for this purpose. Considering this computer model, we can develop a les of cost and effective, high flexibility parallel computing environment for the PWR simulator. This computer model can be separated into three parts: user terminal, clients and gateway.

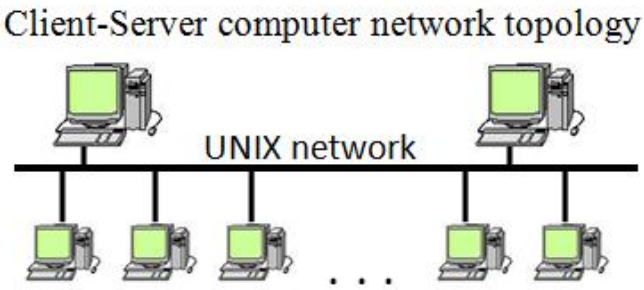

Fig. 1 Client/Server computer model

Programming technique description

Clients: Contain server socket interface and simulator based on MNA method [4]. The main function of the client is to solve individual subcircuit and transfer results to gateway. *Gateway*: Both, server socket and client socket interface and share memory that has been employed for Inter-process Communication (IPC) [3]. The gateway synchronizes the message flow; partitions circuit to subcircuits, orders the subcircuits and more importantly schedules the subcircuits [5]. *User Terminal*: Owns client socket for transferring circuit information and results between user terminal and gateway. The relationships among clients, gateway and user terminal are shown in fig. 2.

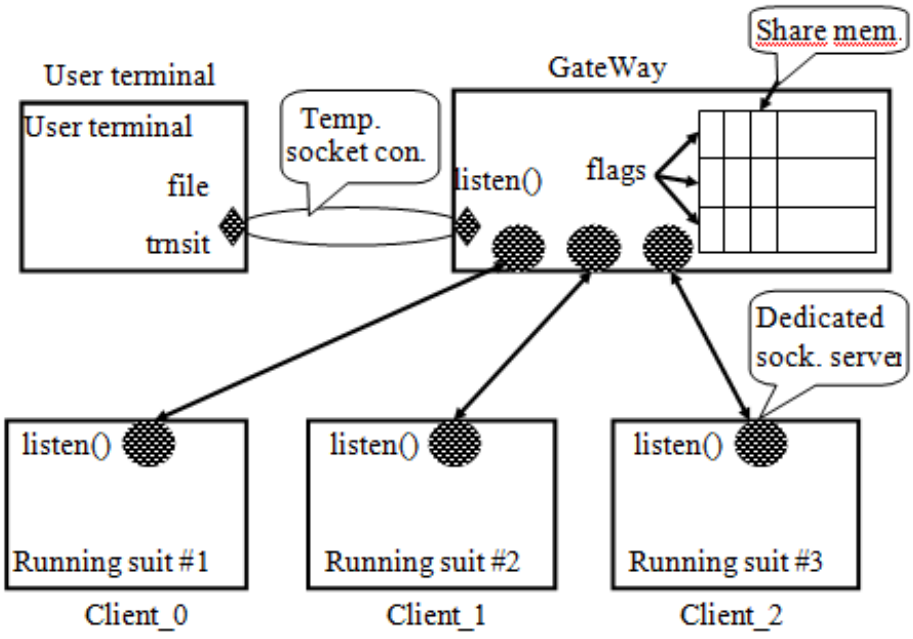

Fig. 2. Simulator program structure

Firstly, circuit information is read in, analyzed and then transferred to gateway by a user terminal program though socket

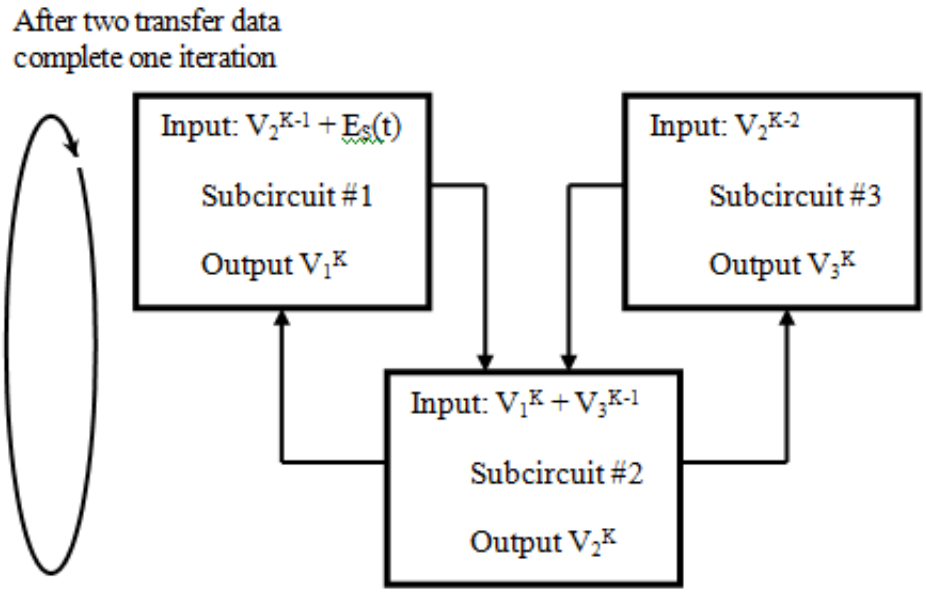

Fig. 3. Subcircuits Dependency among clients

connection. Secondly, gateway receives circuit data, distributes and stores it into share memory and sets flags to activate specified clients. Thirdly, at the same time, client processes have been triggered so that there are message flows among clients and gateway concurrently until convergence criterion of the waveform is met. Finally, user terminal which has a socket connection will close after the user receives simulation results from gateway while socket connection among clients and gateway still connect so that the gateway can wait for another simulation request.

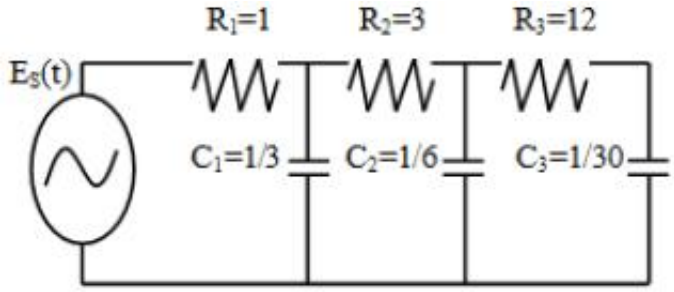

Fig. 4. Circuit for testing PWR simulator

Data dependency

Considering the parallelization for parallel waveform relaxation simulator, data-dependent analysis among clients is very important for optimizing simulation time. The presence of dependence between two computations implies that they cannot be performed in parallel. fig. 3 shows the data dependency among clients and gateway using example circuit in fig. 4 with Gauss-Seidel algorithm. Also, the dependency equations have been listed in (1), (2) and (3) with Gauss-Seidel algorithm; besides, a Gauss-Jacobi version of the WR algorithm for (1), (2) and (3) can be obtained by replacing for each statement with for all statement and adjusting the iteration indices.

$$\#1: (G_1 + G_2 + C_1 \, d/dt) V_1^{\uparrow}(k+1) + G_2 V_2^{\uparrow} k - E_S(t) G_1 = 0 \quad (1)$$

$$\#2: (G_2 + G_3 + C_2 \, d/dt) V_2^{\uparrow}(k+1) + G_2 V_1^{\uparrow}(k+1) - G_3 V_3^{\uparrow} k = 0 \quad (2)$$

$$\#3: (G_3 + C_3 \, d/dr) V_3^{\uparrow} k - G_3 V_2^{\uparrow} k = 0 \quad (3)$$

$$\|V_1^{k+1} - V_1^k\| < \varepsilon \;\; or \;\; \|V_2^{k+1} - V_2^k\| \le \varepsilon \;\; or \;\; \|V_3^{k+1} - V_3^k\| \le \varepsilon \quad (4)$$

As we can see from fig. 4, dependency affects the parallelism where two data flows will achieve one iteration because of dependency between V_1, V_2 and V_3. At first data transfer cycle, subcircuit #1 and subcirucuit #3 return results which are used for input of subcircuit #2 in next data transfer cycle where subcircuit #1, subcircuit #2 and subcircuit #3 are partitioned subcircuits by gateway. Therefore parallel processing has been achieved by #1 and #3 running concurrently.

Simulation results

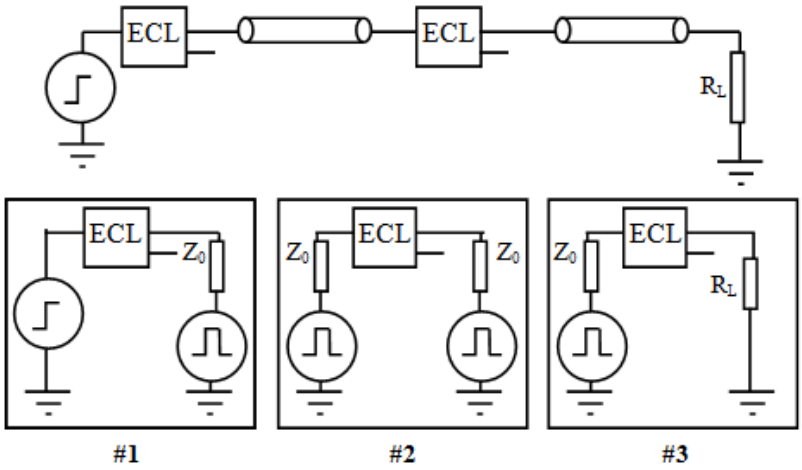

Fig. 5. A bipolar logic system interconnected with lossy line #1, #2 and #3

In the example shown on fig. 4, we use both Gauss-Jacobi and Gauss-Seidel [5] iterative methods to evaluate the circuit simulator. Actually, we found that Gauss-Seidel iterative method has shown superior then Gauss-Jacobi iterative method within same number of iterations. Since from the PWR simulator, the number of iteration for Guass-Jocabi iterative method needs 12 iterations while Gauss-Seidel only needs 6 iterations for the same convergent condition. We also can explore, that data generated from the simulator should on approaches the analytical results as number of iteration increases for both iterative methods.

The Parallel Waveform Relaxation is performed by the simulator to obtain waveform information. Both Gauss-Seidel method and Gauss-Jacobi iterative methods have been used to simulate the example circuit in fig. 4 and the waiting results. The analytic solution of V_1, V_2, and V, derived from the Laplace Transformation are given by (5), (6) and (7). It is obvious to see that the waveform generated by the project simulator converges to the analytical solution after certain iterations.

$$V_1(t) = 1 - 0.125(3e^{-t} + 2e^{-3t} + 3e^{-5t}) \quad (5)$$

$$V_2(t) = 1 - 0.125(9e^{-t} + 2e^{-3t} + 3e^{-5t}) \quad (6)$$

$$V_3(t) = 1 - 0.125(15e^{-t} + 10e^{-3t} + 3e^{-5t}) \quad (7)$$

Transmission line analysis

It will be demonstrated in more detail that this project simulator is most suitable for simulation of VLSI interconnected circuits as indicated in fig. 5, taking advantage of the natural boundaries provided by

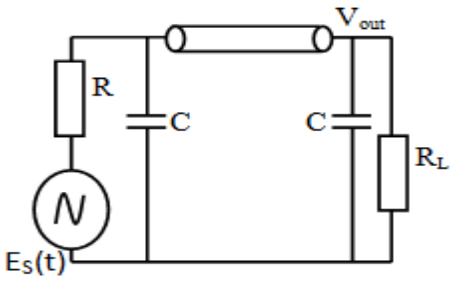

Fig. 6. Transmission line circuit with RC elements

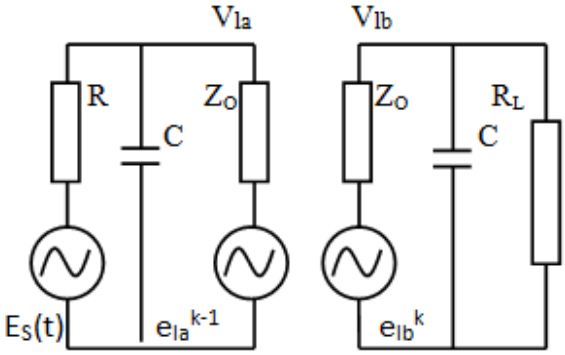

Fig. 7. Transmission line circuit modeling

transmission lines for circuit partitioning. For example, as, shown in fig. 6,7, the circuit is consist of RC elements. We can partition it into two sub-circuits. Each one is analyzed in different computer and the output is as shown in fig. 8.

Conclusions

The method of Parallel Waveform Relaxation using different iterative methods to simulate a simple circuit is described in this paper. The presentation for the case of linear circuits is not shown. The method works for nonlinear circuits as well and will be described in more detail in the future.

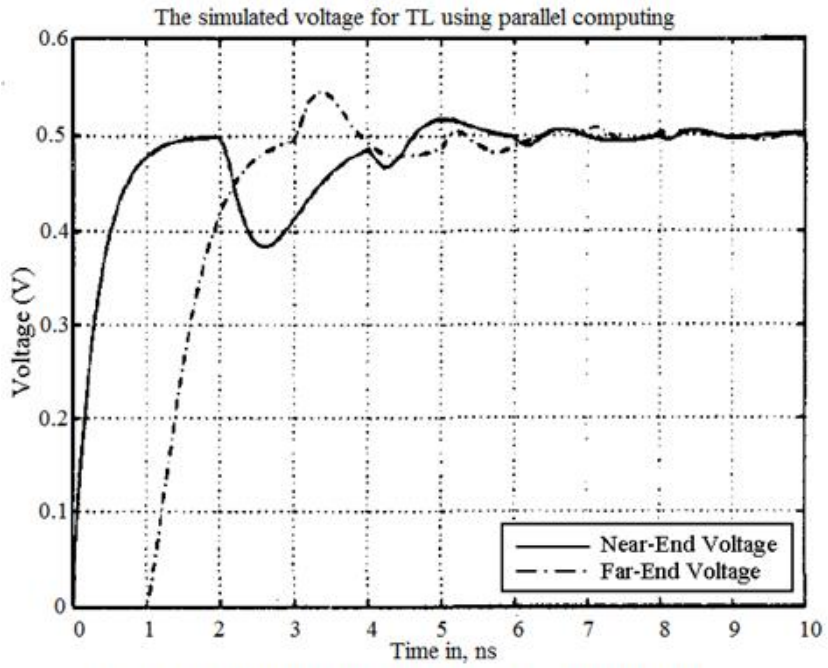

Fig. 8. The Discrete-Time simulated voltage waveform for transmission line modeling delay of transmission line in 1ns

References

1. S. Mattisson. CONCISE: A Concurrent Circuit Simulation Program. Dep. Appl. Electron. 1986, – Res. Lund Univ., Lund.
2. F.Y. Chang, 'Waveform Relacation Analysis of RLCG Transmission Lines", *IEEE Trans. Circuits and Systems.* 1990, – vol. CAS-37 No.11, pp. 1394-1415.
3. UNIX Network Programming, W. Richard Stevens Prentice Hall
4. C.W. Ho et al. 'The modified nodal approach to network analysis," IEEE Trans. Circuits and Systems. June 1995.– vol. CAS-22, pp.504-509.
5. J. White and A. L. Sangiovanni-Vincentelli, Partitioning algorithms and parallel implementations of waveform relaxation algorithms for circuit simulation. Proc. IS CAS, K Toy, June 1995.

Мельник В.М. Циклічний симулятор реалізований через з'єднання UNIX coketom з паралельною релаксацією хвильової форми сигналу. *Циклічний симулятор з паралельною релаксацією хвильової форми сигналу був змодельований для ефективності розподілених обчислень на розподілених комп'ютерах. Його робота порівнюється з іншими подібними симуляторами, які використовуються в паралельних обчисленнях. Це більш гнучкий механізм, економічний, з незалежним ресурсом.*

КЛЮЧОВІ СЛОВА: циклічний симулятор, релаксація хвильової форми, паралельні обчислення, сокет, UNIX, мережа.

Мельник В.М. ***Цыклический симулятор реализированный через соединение UNIX coketom с паралельной релаксацией волновой формы сигнала.*** *Циклический симулятор с параллельной релаксацией волновой формы сигнала был смоделирован для эффективности распределенных вычислений на распределенных компьютерах. Его робота сравнивается и иными подобными симуляторами, которые используются в параллельных вычислениях. Это более гибкий механизм, экономичнее, с независимым ресурсом.*

КЛЮЧЕВЫЕ СЛОВА: циклический симулятор, релаксация волновой формы, параллельные вычисления, сокет, UNIX, сеть.