Parallelization of DC-Analyzer

G. Anil Kumar* **Gaurav Trivedi†** **Madhav P. Desai‡** **H. Narayanan§**

Abstract

Physical problems offer scope for macro level parallelization of solution by their essential structure. For parallelization of electrical network simulation, the most natural structure based method is that of *Multiport Decomposition*. In this paper this method is used for the simulation of electrical networks consisting of resistors, independent and controlled sources using a distributed cluster of weakly coupled processors. Results are presented for the cases where the number of processors are 1,2,4,8 and for circuit sizes upto 700,000 nodes and 1.4 million edges. We use a cluster of Pentium IV processors linked through a 10/100MBPS ethernet switch.
Keywords: Multiport Decomposition, Parallel processor, simulation.

INTRODUCTION

It is becoming increasingly necessary to solve very large circuits accurately. This is both because of higher chip densities and also because of the incorporation of high frequency effects. Usually such circuits are too large to be solved on a single processor and some form of distributed computing has to be resorted to. We have chosen the model of cluster computing in which all the processors are connected through weak media. We used a 10/100 Mbps unmanaged Ethernet switch for interconnecting processors. This level of resources is easy to obtain in almost any computational laboratory or software house. We have solved DC circuits of sizes upto $700,000$ nodes and 1.4 million edges by our technique using such resources in only a few minutes.

The general approach used by us to parallel process a circuit simulator is to decompose the circuit into k 'multiports'. These decomposed k multiports are solved independently. After obtaining the port behaviour we use the connection constraints between multiports for computing the port voltages and currents. For best results during parallel computing, communication among processors should be a minimum. It is necessary therefore that decomposed blocks should have minimum interconnection between themselves. We present our method of solution in Section 2.

Every general purpose circuit simulator has a *DC-Analyzer* at its core. In any iteration of non-linear circuit analysis a circuit simulator solves a circuit consisting only of elements: resistors (R),controlled sources (CS), voltage sources (V) and Current sources (J). Thus, parallelization can be done in two ways. First, by parallelizing each iteration of the non-linear analysis and second, by parallelizing the *DC-Analyzer* itself. The first approach is adopted by [4]. We adopt the second approach [7, 2, 10] and show that parallelizing the *DC-Analyzer* may yield significant speedup in overall circuit simulation. The algorithm used for parallel *DC-Analysis* is explained in Section 3. Results are discussed in Section 4.

MULTIPORT DECOMPOSITION

A port conventionally means a pair of terminals in a network which can be used to connect it to another network in such a way that the current entering through one terminal of the network is equal to the current coming out of another terminal of the same network. Now, *Multiport* can be viewed as more than one port in a network to make connections with external networks. For studying the behavior of a network on another network when both are connected, we use, for formal purposes, a device (*norator*) which permits every current voltage combination. When connected across a pair of external terminals of a network it automatically ensures that the current leaving the network through one terminal equals the current entering the other since these currents must equal the current flowing through the norator. In further discussions an electrical network could include some extra devices which are norators. Norators are specified as ports.

A electrical circuit is divided into k parts by using *Multiport Decomposition* [8] in such a way that there is no interaction between the k parts. The decomposed

*Texas Instruments, Wind Tunnel Road, Murugeshpalya, Bangalore-560017. email: anilk@india.ti.com

†Department of Electrical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai-400076. email: trivedi@ee.iitb.ac.in

‡Department of Electrical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai-400076. email: madhav@ee.iitb.ac.in

§Department of Electrical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai-400076. email: hn@ee.iitb.ac.in

parts may be large but the number of ports should be minimum.

A network N, which is to be decomposed into two multiports, is given in figure 1. N_A and N_B are the subnetworks of N. Assume that the devices in both of the network are decoupled. It can be seen that the currents in N_B can affect the currents in N_A or vice-versa according to the KCE constraint at n_1, n_2. Simultaneously voltages in $N_A(N_B)$ can affect the voltages in $N_B(N_A)$ because voltage $v_1 = v_{n_1} - v_{n_2}$. Thus the constraint on all currents and voltages of N remain the same if we replace the network by the decomposed network. Both the original and decomposed networks are shown in figure 1. In the figure 1 norators are represented by dotted lines. The conditions $v_P = v_{P'}, i_P = -i_{P'}$, are imposed by the *port connection diagram* which depicts the two norators in parallel. This can be written in the form of matrices also, which is given below.

Let $[A'_{rA}A'_{rB}], [A_{rA}A_{rP}], [A_{rB}A_{rP'}]$ be the reduced incidence matrices of graphs $G, G_{AP}, G_{BP'}$ respectively and $[B'_A B'_B], [B_A B_P], [B_B B_{P'}]$ are the loop matrices (whose rows span the current space of graphs $G, G_{AP}, G_{BP'}$ respectively).

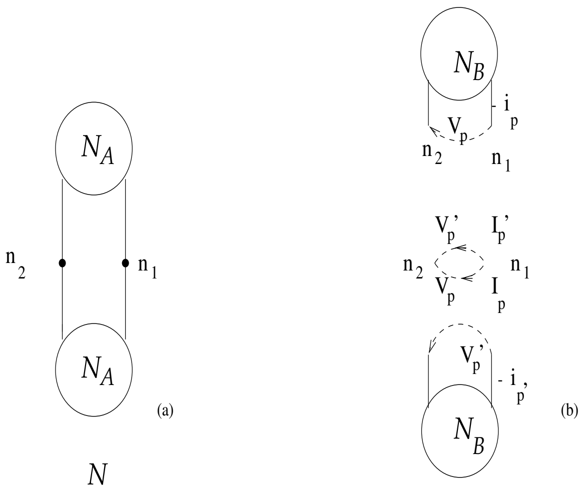

Figure 1: Multiport Decomposition Technique

$$\begin{bmatrix} A_{rA} & A_{rP} \end{bmatrix} \begin{bmatrix} i_A \\ i_P \end{bmatrix} = 0 \quad (1)$$

$$\begin{bmatrix} A_{rB} & A_{rP'} \end{bmatrix} \begin{bmatrix} i_A \\ i_{P'} \end{bmatrix} = 0 \quad (2)$$

$$\begin{bmatrix} I & I \end{bmatrix} \begin{bmatrix} i_P \\ i_{P'} \end{bmatrix} = 0 \quad (3)$$

$$\begin{bmatrix} B_B & B_P \end{bmatrix} \begin{bmatrix} v_A \\ v_P \end{bmatrix} = 0 \quad (4)$$

$$\begin{bmatrix} B_B & B_{P'} \end{bmatrix} \begin{bmatrix} v_B \\ v_{P'} \end{bmatrix} = 0 \quad (5)$$

$$\begin{bmatrix} I & -I \end{bmatrix} \begin{bmatrix} v_P \\ v_{P'} \end{bmatrix} = 0 \quad (6)$$

A more general example of Multiport Decomposition is shown in figure 2.

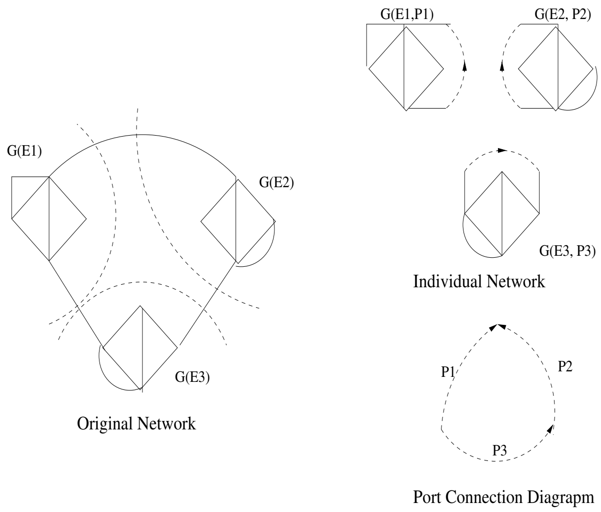

Figure 2: Multiport Decomposition Technique

ALGORITHM

The method used for circuit decomposition is *Multiport Decomposition*. The analysis is performed by *DC-Analyzer* [1] which uses the *Two-Graph Method* [1]. An elementary partitioner was used to obtain the graph of the network into subgraphs of approximately equal size with very few common nodes. *PVM* [5, 9] is used for communicating among processors. Assuming that the DC-Analyzer was available on each processor, the algorithm for parallel DC analysis of electrical network is given below:

1. Partition a large electrical network using the graph partitioner.
2. Form multiports from the partitioner output.
3. Extract *Port Connection Diagram* from the partitioned blocks.
4. Send multiports to each processor.
5. Compute *Port Characteristic* i.e. the matrix G and the vector b in the equation $i = Gv + b$, where v, i are the port voltages and currents, by solving each multiport repeatedly by keeping each one of the port voltages to be 1 and all the others to be zero.
6. Using the port characteristics as coupled device characteristics of the group of ports corresponding to each multiport, write nodal equations for the port connection network and compute the branch voltage of each port edge. These latter are the actual voltages of the corresponding ports in the multiports.
7. Solve each multiport once more for the port voltages computed above. This gives all branch voltages and currents of the original network.

The way in which port characteristic [7, 2, 10] is extracted, is given below:

Mathematical model of the port characterization

In a multiport, port sources can be taken to be either voltage or current sources. In the present method port sources are considered as voltage sources and port behavior is of conductance type. Port behavior is given by $i = Gv + b$. Port connection diagram may be represented by the reduced incidence matrix $[A_1 A_2 ......A_k]$. From port connection diagram, we can write

$$\begin{bmatrix} A_1 & A_2 & \cdots & A_k \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_k \end{bmatrix} = 0 \qquad (7)$$

But we know from the behavior of port sources, $i_1 = G_1 v_1 + b_1$, $i_2 = G_2 v_2 + b_2$ ....etc. $i_1, \cdots, i_k, v_1, \cdots, v_k$ and $b_1, \cdots, b_k$ all are vectors and $G_1, \cdots, G_k$ are the matrices. Thus the equation 7 can be modified as

$$\begin{bmatrix} A_1 & A_2 & \cdots & A_k \end{bmatrix} \begin{bmatrix} G_1 v_1 + b_1 \\ G_2 v_2 + b_2 \\ \vdots \\ G_k v_k + b_k \end{bmatrix} = 0 \qquad (8)$$

or

$$\begin{bmatrix} A_1 & A_2 & \cdots & A_k \end{bmatrix} \begin{bmatrix} G_1 & 0 & \cdots & 0 \\ 0 & G_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & G_k \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix}$$

$$= - \begin{bmatrix} A_1 & A_2 & \cdots & A_k \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix} \qquad (9)$$

v_1, v_2, v_3 .... can be represented in terms of $v_1 = A_1^T v_{n_1}$, $v_2 = A_2^T v_{n_2}$,...., $v_k = A_k^T v_{n_k}$. So the equation turns out to be:

$$\mathbf{AGA^T} \begin{bmatrix} v_{n_1} \\ v_{n_2} \\ \vdots \\ v_{n_k} \end{bmatrix} = - \begin{bmatrix} A_1 & A_2 & \cdots & A_k \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix} \qquad (10)$$

where,

$$\mathbf{A} = \begin{bmatrix} A_1 & A_2 & \cdots & A_k \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} G_1 & 0 & \cdots & 0 \\ 0 & G_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & G_k \end{bmatrix}$$

and

$$\mathbf{A^T} = \begin{bmatrix} A_1^T \\ A_2^T \\ \vdots \\ A_k^T \end{bmatrix}$$

Here $v_{n_1}, v_{n_2}, ..., v_{n_k}$ are the node voltages of port connection diagram. The coefficient matrix on the left side of the equation 10 will be called *port conductance matrix.* Current in each port sources is calculated using DC-Analyzer by setting each port source as a 1-volt source at a time and rest all port sources and internal sources zero in a multiport. If the number of port sources in a multiport is p, the DC-Analyzer will be called $p + 1$ times for analysis. Each time it calculates the current vector. The current vector obtained is the same as that of the column of the port conductance matrix corresponding to the port voltage source set to 1-volt. After p such calculations, submatrix G_i will be obtained. In the same way, b_i is obtained by solving multiport on each processor by setting all the port sources zero and all the internal sources live. After calculating G_i and b_i for $i = 1, ...n$, node voltages of port connection diagram, can be calculated by LU-Decomposition. LU-Decomposition [6, 3] also may be parallelized, if it takes significant time in calculation. After obtaining the node voltages of port connection diagram, the port voltages and currents can be computed and thence the node voltages and branch currents of each multiport.

RESULTS

Parallelization of DC-Analyzer was implemented using the network of PIV 1.6 GHz processors each having 256MB RAM. Processors used were connected through a 10/100 Mbps Switch. DC-Analyzer is called $p + 1$ times for the p-ports in a multiport to compute the currents in port voltage sources when it is parallelized. The best case of parallelization is where the speedup obtained approaches the number of processors. This is achieved using *Multiport Decomposition* provided the percentage of ports relative to total number of nodes is small (.5%- 1%).

The multiport decomposition method was used to solve a number of large DC networks through parallelization. In every case the network had a planar grid structure- the width and height, in terms of number of nodes, has been specified in the results (eg. the network g700k is 5000x140, width being 5000 nodes and height, 140 nodes). For this kind of structure, a partition into 8 blocks, would simply be that obtained by seven equally spaced vertical cuts resulting in 14x139 ports. However, to have a better idea of the performance of our technique for a general circuit, the partitioning was done by a general purpose public domain partitioner. We found that the number of ports in the larger circuits has come out to be larger than the optimum. This is an area where considerable improvement can be made.

An essential precaution in dealing with large circuits is to avoid frequent access of the hard disc as this is extremely slow in comparison with operations within the RAM. So, as far as possible, we have attempted to bring the required circuit entirely within the RAM and perform all the calculations. In the case of g600k and g700k this has not been entirely feasible and one can see that the times for computation increase abruptly from g500k to g600k and the latter to g700k.

Note that in the following discussion we mean by parallel processors a distributed cluster of weakly coupled processors. We used a master processor for scheduling the individual multiport solution tasks to different slave processors. After the port characterization was computed by the slaves, this was communicated to the master and the latter by itself solved the port connection network.

The tables contain the following data: A circuit description table gives names of circuits against number of nodes and edges. In each circuit voltage and current sources are kept constant at 14% and 7% of the circuit size respectively. Each subsequent table corresponds to a certain fixed number of slave processors. Columns 1,2 are self explanatory. Column 3 gives the input read time. Column 4 is the maximum time a processor took to compute the port characterization of the individual multiports (other processors which finished their task earlier would remain idle until this duration is completed). Column 5 gives the time taken to compute the final solution after port voltages are obtained. Column 6 gives the time taken to solve the port connection network by the master. The last column contains the total time taken in the communication between master and the slaves.

The time during which the master was active can be taken to be the communication time + the port connection network solution time. Where this can be neglected, it can be seen that the solution time for the entire network is inversely proportional to the number of slave processors. The coefficient matrix for the port connection network would be dense and as the network becomes larger (a few million nodes) this part of the computation would begin to dominate in the overall computation, unless it is also parallelized.

We used 8 block partitions throughout, independent of the number of slave processors. This is so that in each choice of number of slave processors $(2, 4, 8\ )$ the task of solving the eight multiports is divided equally between the slave processors. For comparison, we have given the results for a 10 block partition with 2 slave processors. It can be seen that the results are not substantially different. However, the time for solution of the port connection network is higher as is to be expected since when the number of blocks goes up the port connection network tends to increase in complexity. We note that times given are approximate and rounded to the nearest integer.

Circuits Used

Circuit	Nodes	Edges
g60k	60,000	118940
g105k	105,000	208430
g200k	200,000	397420
g300k	300,000	596170
g400k	400,000	795900
g500k	500,000	994900
g600k	600,000	1194880
g700k	700,000	1394860

Results: 8-block partition

Here, t_{ip} is the time for reading the input. t_{pc} is the time taken for port characterization. t_{mul} is the time taken to compute the coefficient matrix of the port connection network solution. t_{pcn} is the time taken in port connection diagram solution. t_{comm} is the time taken in communication.

With one slave processor

Circuit	No of Blocks	t_{ip} (in sec)	t_{pc} (in sec)	t_{mul} (in sec)	t_{pcn} (in sec)	Total Time (in sec)	t_{comm} (in sec)
g60k	8	0	16	0	1	17	0
g105k	8	0	35	0	1	36	1
g200k	8	0	80	0	2	82	4
g300k	8	0	122	1	2	125	4
g400k	8	0	217	1	4	222	7
g500k	8	1	272	1	8	282	7
g600k	8	0	404	1	12	417	10
g700k	8	1	840	1	63	905	11

With two slave processors

Circuit	No of Blocks	t_{ip} (in sec)	t_{pc} (in sec)	t_{mul} (in sec)	t_{pcn} (in sec)	Total Time (in sec)	t_{comm} (in sec)
g60k	8	0	9	0	1	10	0
g105k	8	0	19	0	2	21	1
g200k	8	0	40	0	3	43	4
g300k	8	0	62	0	2	64	4
g400k	8	0	121	1	4	126	7
g500k	8	0	137	1	7	145	7
g600k	8	1	223	1	12	237	10
g700k	8	1	437	1	65	504	11

With four slave processors

Circuit	No of Blocks	t_{ip} (in sec)	t_{pc} (in sec)	t_{mul} (in sec)	t_{pcn} (in sec)	Total Time (in sec)	t_{comm} (in sec)
g60k	8	0	5	0	1	6	0
g105k	8	0	10	0	1	11	1
g200k	8	0	22	1	2	25	4
g300k	8	0	34	0	2	36	4
g400k	8	0	61	1	3	66	7
g500k	8	1	77	1	7	85	7
g600k	8	0	115	1	13	129	10
g700k	8	0	254	2	63	319	11

With eight slave processors

Circuit	No of Blocks	t_{ip} (in sec)	t_{pc} (in sec)	t_{mul} (in sec)	t_{pcn} (in sec)	Total Time (in sec)	t_{comm} (in sec)
g60k	8	0	3	0	1	4	0
g105k	8	0	5	0	2	7	1
g200k	8	0	12	0	2	14	4
g300k	8	0	19	0	2	21	4
g400k	8	0	33	1	4	38	7
g500k	8	1	41	1	7	49	7
g600k	8	0	64	1	11	76	10
g700k	8	0	156	1	63	220	11

Results: 10-block partition

With two slave processors

Circuit	No of Blocks	t_{ip} (in sec)	t_{pc} (in sec)	t_{mul} (in sec)	t_{pcn} (in sec)	Total Time (in sec)	t_{comm} (in sec)
g60k	10	0	9	0	2	11	0
g105k	10	0	19	0	4	23	1
g200k	10	0	44	1	7	52	4
g300k	10	0	64	1	19	84	6
g400k	10	0	109	1	21	131	7
g500k	10	0	140	1	37	178	9
g600k	10	0	199	1	81	281	10
g700k	10	0	299	2	189	490	11

Conclusion

In this paper a method of parallelization of circuit simulation is outlined which is based on the structure (topology) of the network viz. *Multiport Decomposition.* We use this method to parallelize the DC-Analyzer noting that since a DC-Analyzer lies at the core of every general purpose simulator, this would amount to parallelizing the circuit simulator. Our results show speedups proportional to the number of processors provided the blocks into which the network is broken up do not have many ports in between (number of ports $< 1\%$). We find it interesting that a method which uses no sparsity exploiting technique except the simple structural one of multiport decomposition does as well as can be hoped for with the best kind of parallelization. We note that circuits of sizes up to $700,000$ nodes and 1.4 million edges have been solved by this technique in a few minutes using facilities easy to obtain in any computational laboratory or software house.

References

[1] Batterywala, S. H., and Narayanan, H. Efficient dc analysis of rvj circuits for moment and derivative commutations of interconnect networks. *12th International Conference on VLSI Design* (1999), 169–174.

[2] Bhattad, N. J. Parallelization of circuit simulator. Master's thesis, Department of Electrical Engineering, Indian Institute of Technology, Bombay, January 2001.

[3] Dongara, J., Duff, L., Sorensen, D., and der Vorst, H. V. *Numerical Linear Algebra for High-Performance Computers.* Society of Industrial and Applied Mathematics, Philadelphia, 1998.

[4] Frohlich, N., Riess, B. M., wever, U. A., and Zheng, Q. A new approach for par-

allel simulation of vlsi circuits on a transistor level. *IEEE Transaction on Circuits and system -1:Fundamental Theory and Applications 45*, 6 (1998), 601–613.

[5] Gist, A., Beguelin, A., Dongara, J., Jiang, W., Manchek, R., and Sunderam, V. *PVM 3 User Guide and Reference Manual*. Oak Ridge National Laboratory, Engineering Physics and Mathematics Division, Mathematical Science Section, Oak Ridge, Tennessee, 378331, USA, September 1991. operated by Martin Marietta Energy Systems.

[6] Golub, G., and Loan, C. V. *Matrix Computations - 2nd Edition*. Johns Hopkins University Press, Baltimore, Maryland, 1989.

[7] Kumar, G. A. Parallelization of circuit simulator. Master's thesis, Department of Electrical Engineering, Indian Institute of Technology, Bombay, January 2002.

[8] Narayanan, H. *Submodular Function and Electrical Networks, Annals of Discrete Mathematics*, vol. 54. North Holland, Amsterdam, The Netherlands, 1997.

[9] Pvm http site. http://www.epm.ornl.gov/pvm/.

[10] Trivedi, G. Parallelization of circuit simulator. Master's thesis, Department of Electrical Engineering, Indian Institute of Technology, Bombay, January 2000.