Parallelization of Circuit Simulation

Students
Gaurav Trivedi
Nilesh J. Bhattad
G. Anil Kumar
V. Venkat Rao

Faculty
H. Narayanan (EE)
Madhav P. Desai (EE)
Sachin Patkar (Maths)

EE Department,IITB

Motivation

- To solve large circuits through *easily available, weakly connected* cluster of computers.
- Every general purpose circuit simulator has a *DC-Analyzer* in its innermost loop. So, we parallelize the DC analyzer.

Application

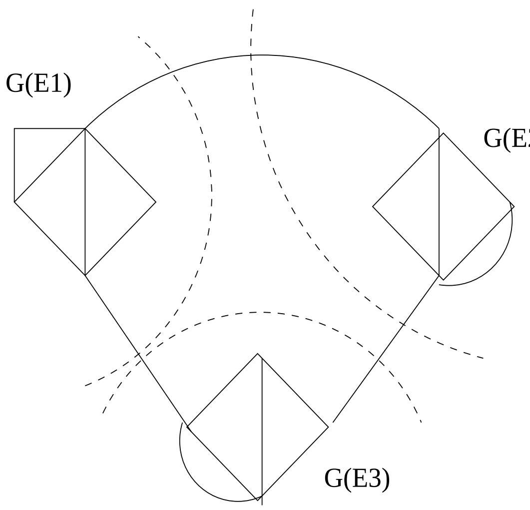
DC-Analysis can be used to

- Solve large *sparse linear equations*.
- Solve *pipe flow* problems
- Solve large *Combinatorial Optimization* problems (e.g. Min Cost flow problem)

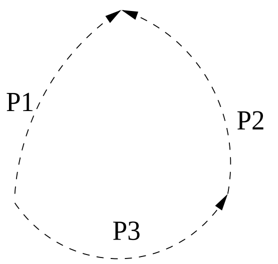
Introduction

- We decompose the circuit into *k-multiports*.
- These decomposed k-multiports are solved independently in terms of *port conditions*. This also gives *port behaviour*.
- We use *port behaviour* and *connection constraints* between multiports for computing the port voltages and currents.
- We use the decomposed multiports' solution in terms of *port conditions* to get voltages and currents inside the multiports.

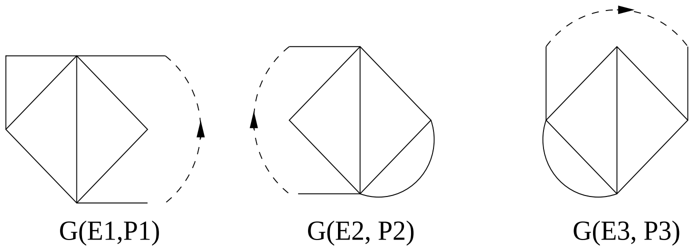
Multiport Decomposition



Original Network



Port Connection Diagrapm



Individual Network

Algorithm

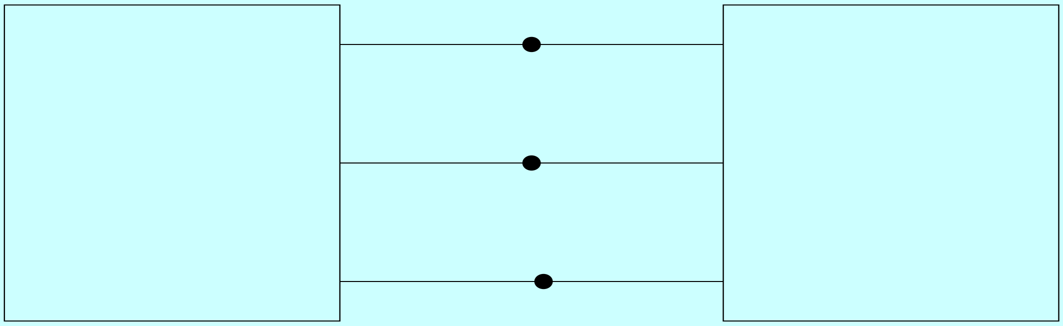


Figure 1: An example circuit

Algorithm

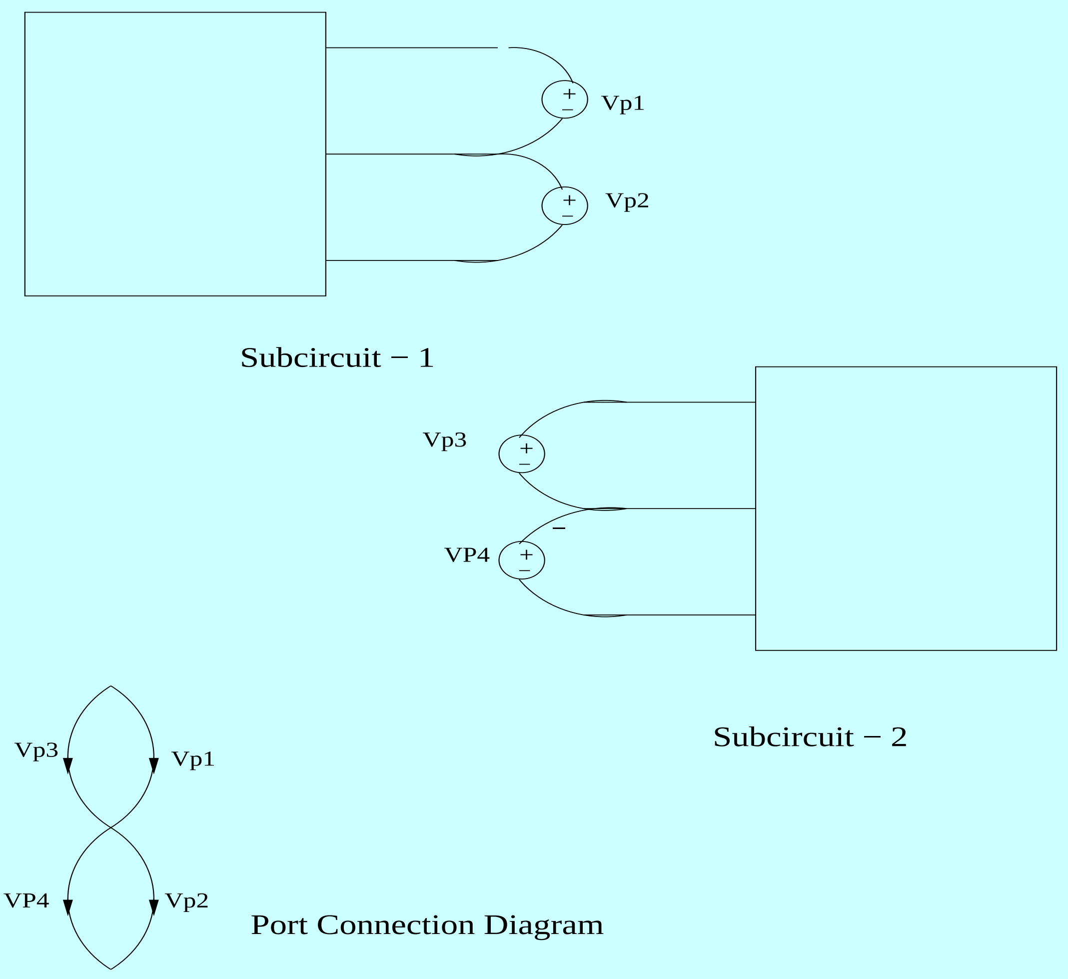
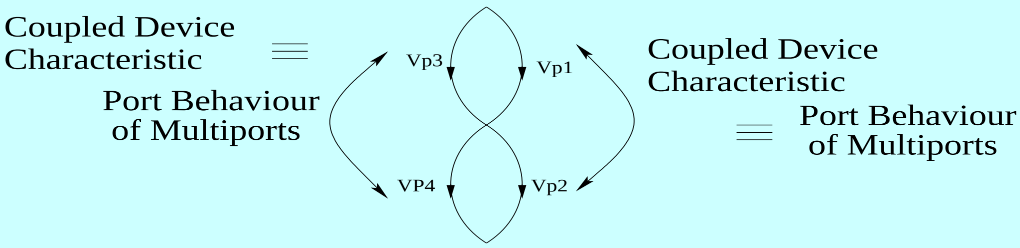


Figure 2: Decomposed Circuit

Algorithm



Port behaviour is now
device characteristics

Solve 'Port Connection Network'
and get Port Conditions

Figure 3: Port Behaviour Extraction

Algorithm

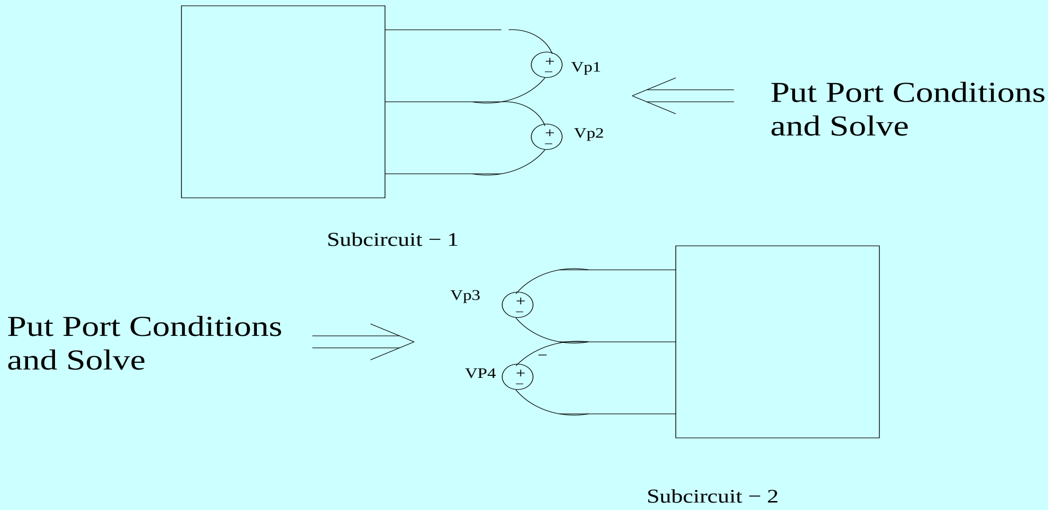


Figure 4: Solving Individual Multiports

Experimental Setup

- Experiments were performed on a cluster of *9-processors* using a *10/100 Mbps unmanaged ethernet switch*.
- In this setup, one processor acted as *Master Processor* and others were *Slave Processors*.
- We used *Multiport Decomposition*.
- The speedup obtained approaches the number of processors provided the percentage of ports relative to total number of nodes is small *(.5%- 1%)*.
  - As far as possible, access to the *hard disk* is avoided.

Circuit Description

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

with *one slave processor*

Circuit	t_{total} (in sec)	t_{comm} (in sec)
g60k	17	0
g105k	36	1
g200k	82	4
g300k	125	4
g400k	222	7
g500k	282	7
*g600k	417	10
*g700k	905	11

* spilling out of RAM.

Results: 8-block partition

Time for solving *Port Connection Diagram*

by *2, 4* **and** *8 Slave Processors*

Circuit	$^*t_{2proc}$ (in sec)	$^*t_{4proc}$ (in sec)	$^*t_{8proc}$ (in sec)
g60k	9	5	3
g105k	19	10	5
g200k	40	22	12
g300k	62	34	19
g400k	121	61	33
g500k	137	77	41
g600k	223	115	64
g700k	437	254	156

* t_{2proc}, t_{4proc} and t_{8proc} show the time spent in calculating *Port Behaviour* using two, four and eight slave processors respectively.

Results: 8-block partition

with *8 slave processors*

Circuit	t_{total} (in sec)	t_{comm} (in sec)
g60k	4	0
g105k	7	1
g200k	14	4
g300k	21	4
g400k	38	7
g500k	49	7
g600k	76	10
g700k	220	11

Conclusion

- *Multiport Decomposition* is a promising method of parallelization of circuit simulation.
- *Port Connection Behaviour* has been fully parallelized.
- Solution of the *Port Conductance Matrix* needs to be parallelized.
- 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 $< 1\%$)*.
- The 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.