Parallel Circuit Simulation on Graphical Processing Unit

Aram Baghdasaryan

Synopsys Armenia CJSC
3rd year PhD student, State Engineering University of Armenia
Yerevan, Armenia
aramb@synopsys.com

Abstract— So high integration of IC design and mix VLSI design have brought new complexity in IC design. This complexity brings new challenges for simulation IC time. There is interest to speed up Spice [1] simulation because for large IC simulation can take several days. Average 75% percent of simulation time is spent in evaluating transistor model equations. This report is discussing accelerating transistor model evaluation using Graphical Processing Unit (GPU). For speed up simulation time also used scheduling algorithms which help schedule tasks according to running time criteria. According to results method which is represented in this paper sped up simulation to 2.5 times.

Keywords—GPU, scheduling algorithms, simulation, speed up, parallel.

I. Introduction

Nature is analog and interaction with nature is also analog. Analog circuits are necessary where area, power, and high frequency operation can't be performed by digital circuits. Analog circuits are used in microprocessor supervisory circuits, massively parallel analog signal processors, switched-current filters and etc. There was a phenomenal growth of integrated circuits industry during last decades. In the middle of 60's appeared simple gates and operational amplifiers, in the 70's microprocessors and analog-to-digital conversion were discovered. Approximately 60% percent of CMOS, BiCMOS were mixed analog and digital parts. Analog design becomes part of the most digital circuits. Though high integration of IC design and mix VLSI designs have brought new complexity in IC design. The lack of analog circuit design formulation, circuit independent design procedure make analog design simulation complex and time consuming process. Simulation for large analog design can take several days. Though these growths in IC design bring new challenges to computer aid systems.

If simulation results don't satisfy specification, then simulation is repeated several times until it satisfies the specification. If after several iterations results don't satisfy specification, then designer should change circuit design and repeat same iterations as described above. This process is time consuming and can increase simulation time, that's why iterations are limited.

Laws for circuit theory (Kirchhoff's, Ohm lows) are not enough to design functional circuit. Analog circuit designer should also know other techniques, knowledge to design circuits. There are several approaches which help predict circuit behavior.

A. Analytic design equation

Simple analytic design equations predict sufficiently accurate circuit behavior. Many methods (small signal modeling, analysis method) were discovered for solving these complex equations without much loss of accuracy. Another approach is qualitative relationships base approach.

B. Qualitative relationships

Qualitative relationships between circuit performance and design variables help to understand circuit behavior. For example voltage gain of a CMOS amplifier depends on DC bias current's value and voltage gain of CMOS can be improved if DC bias current is reduced. This type of knowledge helps designers to choose appropriate values for variables which lead to circuit optimization.

II. Simulation

IC design complexity brings new challenges to computer aid systems. The first automation tools were optimization phase. These tools are limited due to several reasons.

- Good starting point. If designers specify bad starting point, it will bring bad circuit design. This issue overcomes with random starting points, but these methods are time consuming process.
- Circuits optimize knowledge. Designer has to define parameters which optimizer tool can change, in the other cases bad parameters can degrade optimization process.
- These systems are slow due to they involve simulation in the optimization loop.

Besides these limitations, CAD tools improve analog design in following ways.

- Reduce design time. This will help to enter market early.
- Make design process simple. It will allow to designers implement standard analog cells very quickly.

- Reduce probability of errors in design. Automation systems help to decrease the design cycle/success ratio.
- Improves manufacturing yield. Computer aid systems can improve manufacture yield and reduce profitability.
- Reduce production cost. This will help to reduce time for analog design which reduces production cost.

III. Previous Work

IC design parallel simulation isn't new topic and there are many researches related to speeding up simulation time. An increasing number of elements in integral circuits (IC) bring new challenges for simulation tools. Nowadays simulation with Spice or with direct method simulators on scalar processor is a time consuming process. In circuit exist parallelism and it can be used to speed up simulation. There are two ways for reducing simulation time: develop effective algorithms or use more powerful systems.

A. Algorithms

Multilevel Newton algorithm and waveform relaxation algorithm are used for circuit decomposition [2]. According to decomposition circuit is divided into sub circuits (Fig. 1) and specification for one level should be satisfied by low level. Instead of satisfaction large of specification in top level which is difficult now in every stage design should satisfy sub specification.

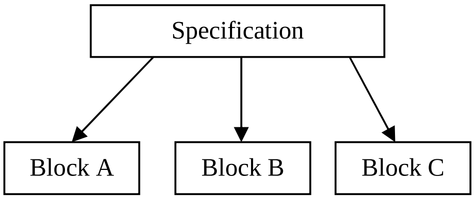

Fig. 1. Sequential decomposition

Circuit is divided according to function and structure.

- Functional
- Structural

According to functional approach sub-circuit should correspond well defined function. Structural division related to input/output characteristics of circuit block. There are 4 types according to this declaration (Models, Building blocks, Task blocks and Primitive blocks).

Models are complex related to other blocks because they contain a significant number of devices. Examples of module are operational amplifiers, comparators, voltage references and etc. Building blocks perform simple function. They are involved in module. A typical example of building example is mirror which performs simple function of mirroring.

Here are the advantages of hierarchical approach.

- It helps to solve easy problems because difficult problems are divided to small and easier sub problems. It contains following approach "divide and conquer".
- It is possible to cover wide range of performance with hierarchical design approach, because single circuit can be added in different circuit design architectures.

The disadvantage of this algorithm is that a lot of feedbacks in circuits can increase simulation time.

B. Hardware

New shift in hardware design (multi-cores, cluster) brings new challenges to simulations tools. The most methods which perform IC simulation on clusters and supercomputers [3] not used shared memory and there are interconnections between processors which increase simulation time. In case of simulation on GPU [4] there is shared memory which helps to reduce simulation time.

IV. Scheduling Algorithms

Scheduling necessity appears in multicore architecture when there are several tasks which are ready to be executed [5]. There are two possible situations for running tasks. Executed task can be displaced by other tasks or block other tasks until its completion. According to this approach there are exist two types of scheduling algorithms` Non-preemptive and Displace.

According to non-preemptive scheduling, task can be executed as much as it requires for completion. Other tasks must wait and they can be executed when previous task completes or wait completion of input or output operations. This method is very simple, but there exist risk related to occupation processor when according to execution occurs error and current task can't give control to other tasks. According to displace scheduling, every task has the same execution time (quota). When execution time expires, task execution is interrupted and time quota assign to the next task. There is no risk of blocking task execution as in non-preemptive scheduling algorithm. Here are widespread algorithms related to displace methodology.

A. First come first served.

The easiest scheduling algorithm is FCFS .When task is ready to be executed it is added in the end of the queue which contains list of ready tasks. Task is selected to run from begging of the queue. The advantage of this algorithm is that it easy to implement.

B. Round Robin (RR)

RR is a modification of scheduling algorithm FCFS. Difference from FCFS algorithm is that tasks, which are ready to be executed, are stored in the cycle queue. Every task has 10-100 ms execution time and when this time expires the next task start to run. There are two possible cases. 1. Execution time for task is less than time quota. In this case task will be removed before time quota expires and other task will be executed. 2. Execution time is bigger than time quota. In this case task running time will be equal to time quota. When according to big quota time majority tasks complete execution than RR algorithm execution time is equal to FCFS execution time. In case of little time quota in theory average waiting and running times are short, but in real systems switching time between tasks increases running time.

C. Shortest-Job-First (SJF)

FCFS, SJF algorithms performance depends on sequence of tasks. Algorithm performance will increase, when short execution task run at first. According to this criteria working algorithm called Shortest-Job-First (SJF).If short execution tasks are several, then their running sequence will be selected by FCFS algorithm. There are two types of SJF algorithm Non-preemptive and Displace. Task running process doesn’t depend on which new tasks are generated according to this time in system in displace algorithm. According to displace scheduling when new tasks appears which execution time is smaller than running task execution time than algorithm displace running task and give processor resources to new task.

V. METHOD FORMULATION

In the Fig. 2 represented general structure of algorithm.

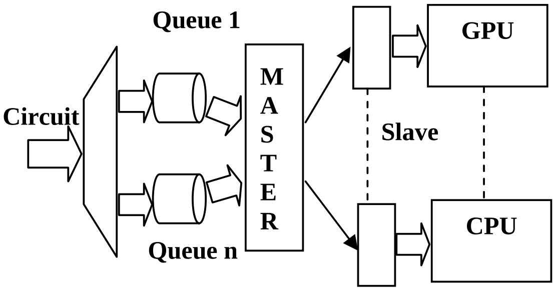

Fig. 2. General structure of the algorithm

At the first stage circuit is divided into sub circuits according to hierarchical approach. Parallel simulation scheduling and synchronization is implemented in master scheduler. Tasks scheduling is selected according to their execution times and depend on their execution times selected on of the scheduling algorithms. At the first all tasks are executed according to Shortest Job First algorithm. If several tasks execution time are equal, then these tasks are selected according to Round Robin algorithm. In the second stage tasks are chosen according to following approach. Task which contains many interaction is running on GPU due to it contains multiple threads and these interactions can be done parallel. Synchronization for running in GPU and CPU is done by the slave schedulers. In the Fig. 3 represented GPU general structure [6].

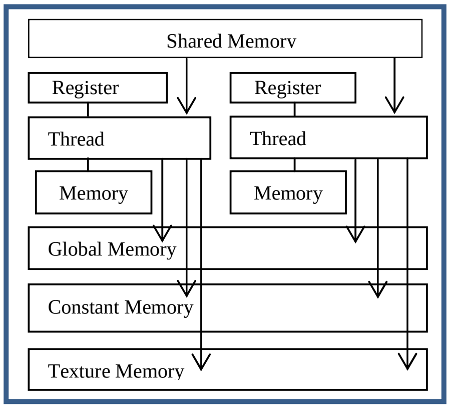

Fig. 3. General structure of algorithm

GPU has the following types of memory:

- Constant
- Texture
- Global
- Local

It consists of the following components:

- One set of register per processor.
- Share memory which is used by all processors
- Read only shared memory that is used by all processors which speed up reading operation from constant memory
- Read only shared memory that is used by all processors which speed up reading operation from texture memory

Global and locale memories aren’t cash memory and they are used for reading, writing operations. Accessing to the local memory is faster than to the global memory, but the local memory is smaller compare to global memory. For reducing simulation it’s preferable to store in local memory if the data size is small. A single floating point value to reading or writing from the global memory take 400 to 600 clock cycles. The latency is possible to reduce if there are instructions which can wait until global memory the end of reading or writing process. CUDA Programing method represented in the Fig. 4.When program is written in CUDA then computing device is GPU. It can execute large number of threads in parallel. In the thread code which was executed called kernel. GPU operates as co-processor for CPU. A thread block contains several threads which can be executed to run parallel which help reduce total simulation time. Every grid contains several blocks. This architecture allows running parallel maximum threads which reduce simulation time. Synchronization in the block is done in following way. All threads are suspended until they all reach synchronization point. Numbers of threads in each block are equal and block size decides programmer. Every thread has its number and it can be views in the code as 1, 2, and 3… dimension value. This method sped up simulation to 2.5 times.

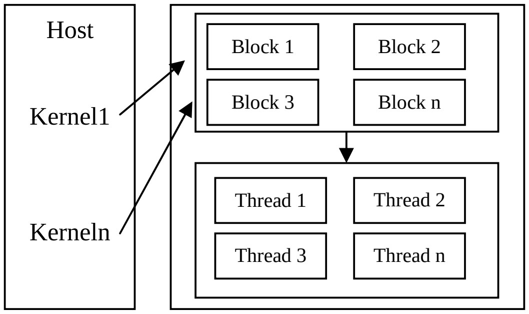

Fig. 4. GPU programming model

In this paper transistor equation simulation is done on GPU which help to reduce simulation time due to CPU contains many parallel threads which help to do simulation in parallel. The most methods which perform IC simulation on clusters and supercomputers not used shared memory and there are interconnections between processors which increase simulation time. In case of simulation on GPU there is shared memory which helps to reduce simulation time.

Parallel simulation scheduling and synchronization is implemented by scheduling algorithms which help to choose best scheduling algorithm depending on task simulation time and sequences. According to this method simulation time is speed up to 2.5 times.

VI. Experimental Results

The experiment environment included Intel 2.4GHz CPU, 4GB memory, NVIDIA 8800GTS display card, CUDA SDK2.0, Visual Studio 2005 C++ programming platform, and Windows 7 operating system. As the results are shown simulation time is speed up 2.5 times as shown in Table I.

TABLE I
Simulation times for GPU and CPU

Trans Number	Total Eval	CPU-alone simulation time (s)	GPU+CPU simulation time (s)	Speed up
300	$1.4x10^7$	44	33	1.3
1200	$2.3x10^7$	102	41	2.5
1100	$4.5x10^8$	550	230	2.4
510	$1.7x10^7$	27	20	1.35
1000	$5.8x10^7$	132	74	1.78
2020	$1.95x10^8$	490	220	2.2
3100	$1.3x10^7$	452	235	1.92

VII. Conclusion

In this paper represented new parallel simulation method, according which simulation time is speed up to 2.5 times**.** The method is implemented by slave and master schedulers. At the first stage circuit is divided into sub circuits according hierarchical approach. Parallel simulation scheduling and synchronization is implemented in master scheduler. Task which contains many interaction is running on GPU due to it contains multiple threads and these interactions can be done parallel. Synchronization for running in GPU and CPU is done by the slave schedulers. This help to speed up simulation.

References

[1] Nagel, L., "*SPICE: A computer program to simulate computer circuits*," in University of California, Berkeley UCB/ERL Memo M520, May., 1995.

[2] Chen, R., "*Solution of Large-scale Circuits by Partitioning, Proc. IEEE TENCONI82*", Hong Kong, Dec., 1982, pp.71-77.

[3] Parkhurst, J., "*From single core to multi-core: preparing for a new exponential*" International conference on Computer Aided Design, Nov., 2006.

[4] Owens, J., "*GPU architecture overview*," in SIGGRAPH '07: ACM SIGGRAPH 2007 courses, (New York, NY, USA), 2007, p. 2.

[5] Chen, X., Wu, W., Wang, Y.,Yu, H., Yang, H., "*An scheduler based data dependence analysis and task scheduling for parallel circuit simulation*" Circuits and Systems II: Express Briefs, IEEE Transactions on, Vol. 58, No. 10, Oct., 2011, pp. 702 –706.

[6] Luebke, D., Harris, M., Govindaraju, N., Lefohn, A., Houston, M., Owens, J., Segal, M., Papakipos, M., Buck, I., "*GPGPU: general-purpose computation on graphics hardware*" in SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing, (New York, NY, USA), p. 208.