Global Routing and Parallelism

Roger Caputo[†], Diego Tumelero[‡], Marcelo Johann[†] and Ricardo Reis[†]

[†] *PGMicro, Instituto de Informática*

[‡] *PPGC, Instituto de Informática*

Universidade Federal do Rio Grande do Sul, Brazil

{rcllanos, dtumelero, johann, reis}@inf.ufrgs.br

***Abstract*— This paper addresses the topic of global routing (GR) in VLSI (Very-Large-Scale Integration) and the use of parallelism as an approach to improve its performance. We review the history of GR and give a look at recent work that has contributed to the state-of-the-art in the field. Academic routers with better results in the past decade are briefly compared and also is shown the work developed by the Microelectronics Group (GME) of the Universidade Federal do Rio Grande do Sul (UFRGS). The use of graphics processing units (GPUs) and a combination of routing algorithms are promising approaches to reduce execution time and ameliorate GR's resolution for open challenges.**

***Keywords*— Global routing, parallelism, GPU, VLSI, EDA**

I. INTRODUCTION

In the design of integrated circuits, the global routing plays a key role and is one of the main challenges that Electronic Design Automation (EDA) tools must face. Routing is a very complex combinatorial problem. It has become a more elaborated process inside EDA due to the increasing number of transistors per die and the advent of a myriad of rules and constraints that each technology advance and device shrinking bring with. Despite all the complexities related to Very-Large-Scale Integration (VLSI) designs, make a circuit routable is arguably the most important task of physical synthesis, even more important than timing closure [1]. A design that does not accomplish the time metrics but is routable instead may require much less effort to be finished than another one that closes on timing (using for example Steiner estimates) but is *unroutable*.

A lot of research, leaded by different groups, has been done in recent years, generating a comprehensive understanding of the basic principles of global routing problem and presenting different approaches for timing, wirelength, congestion, runtime and/or 3D routing. After 2007, the International Symposium on Physical Design (ISPD) contests [2], [3] encouraged the improvement of academic routers, propitiating a competition space to share results and a great opportunity to know the latest industry benchmarks. However, some have questioned the scope of the problem specification and the via capacity consideration for multi-layer routing promoted by these contests, exposing that the research community has converged its effort to the wrong problem [4]. Certainly, the academic formulation has known limitations and particular characteristics that differentiate it from the industrial proposal.

In this paper, is addressed the state-of-the-art of global routing. In the sections III and IV are reviewed the basic principles behind modern approaches to the problem and are presented the most popular academic routers. The use of some parallelism techniques and their importance for global routing runtime is shown in section V. In the last part is introduced recent work developed by the Microelectronics Group (GME) of the *Universidade Federal do Rio Grande do Sul*, Brazil (UFRGS) related to GR.

II. BACKGROUND

Global routing precedes the detailed routing and follows placement and clock tree synthesis in the physical synthesis. It receives a given placement result with fixed locations of blocks and pins.

The global router must distribute the interconnections specified on the netlist across the available routing channels respecting the imposed constraints. After placement, the global router partitions the routing region (the circuit) into tiles and then decides the paths between the tiles for all nets. Figure 1 shows the general process of GR.

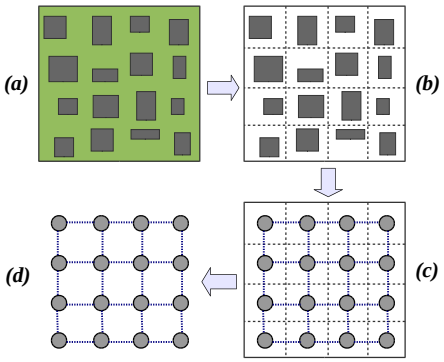

Fig. 1 (a) Placement result. (b) Circuit partitioned. (c) Cells and boundaries representation. (d) Global routing graph

In order to treat mathematically the global routing problem, it is characterized using the graph theory, where the circuit is represented by a grid-graph G that specifies two groups of components, a set of vertices V and a set of edges E. Each $v_i \in V$ denotes a particular region (cell or tile) of a metal layer; meanwhile each $e_{ij} \in E$ corresponds to a boundary between adjacent tiles. As explained in [5], these frontiers have a maximum allowable resource (m_{ij}) which can be used to measure the *overflow*, denoted as the total

amount of demand that exceeds the edge's capacity. At last, there is a set of nets N, and each $n_i \in N$ is composed of a set P_i of pins. Each one of these pins corresponds to a vertex v_i.

A solution is reached when all nets are routed while meeting the capacity constraint of each edge in the minimum runtime and satisfying any other constraint like wirelength, if specified. In some approaches, the use of techniques for parallelization shows results with substantial improvement in terms of execution speed [6].

III. Basic Algorithms

Numerous algorithms for global routing have been presented over the past three decades [7]. Nevertheless, the routers with better performance employ only a subset of these algorithms, using them like ingredients for a successful recipe. A brief description of the basic algorithms used in global routing is presented below.

- *Maze routing* [5] seeks the shortest path (avoiding obstacles) between two points on a grid. It is known as a *brute-force* method because allows all possible paths employing breadth-first search, Dijkstra's algorithm and A* search.
- *Pattern routing* [5] makes point-to-point connections following a small number of fixed shapes, usually minimal-length 'L' and 'Z' paths. It examines fewer grid edges but does not provide guarantees of best local solution.
- *Monotonic routing* follows a similar description as Pattern routing but with a less limiting technique and based on the monotonic function concept.
- *Steiner trees* are used to route multi-pin nets finding minimum total wirelength.
- *Multi-commodity flow (MCF)* uses the flow problem to solve a linear programming relaxation of GR [8].

IV. Most Famous Global Routers

Remarkable progress has been achieved recently in routing algorithms and EDA tools by university-industry researchers groups. Some routers have gained wide popularity inside academic spheres because of their performance, demonstrated at routing contests.

FastRoute [9], *FastRoute* 2.0 [10] and *FastRoute* 4.0 [11]. FastRoute use a congestion map to deform the structure of a Hanan grid [12] during Steiner tree generation followed by edge shifting and pattern routing. In [10], the router is enhanced with monotonic routing and multi-source multi-target maze routing. The latest version addresses the via number optimization problem throughout the entire global routing flow.

BoxRouter [13] and *BoxRouter* 2.0 [14]. The main idea of BoxRouter is to progressively expand a box initiated from the most congested region of the chip, applying an integer linear programming (ILP) formulation considering L-shaped patterns to re-route connections between successive boxes. The BoxRouter 2.0 is an improvement that uses a dynamic version of A* search and incorporates topology-aware rip-up to move wires from congested regions without changing the net topology.

Fairly Good Router (FGR) [15] is based on the PathFinder router originally developed for Field-Programmable Gate Arrays. It uses a particular function for congestion penalty and performs a fast layer assignment followed by a 3D clean-up.

MaizeRouter [16]. It uses two elementary edge-based operations (extreme edge shifting and edge retraction) and manipulates individual segments of nets one-at-a-time. Its approach is founded on interdependent net decomposition, in which routing solutions are implicitly maintained by collections of intervals instead of defined topologies.

NTHU-Route 2.0 [17] is based on rip-up and re-route. It uses a history-based cost function to distribute overflow and employs an identification method to specify the order for rip-up congested regions. Wirelength reduction is achieved through an adaptive multi-source multi-sink maze routing method.

Table I shows a comparison of the routers presented above, showing most of the known techniques used by each router.

TABLE I
Academic Global Routers Comparison

	FastRoute	FastRoute 2.0	FastRoute 4.0	BoxRouter	BoxRouter 2.0	FGR	MaizeRouter	NTHU-Route 2.0
Pattern routing	■	■	■	■	■		■	■
Monotonic routing		■	■					■
Maze routing	■	■	■	■	■	■	■	■
A* search					■	■	■	
FLUTE dependence	■	■	■	■	■		■	■
Topology reconstruction	■	■	■		■	■	■	■
Incremental						■	■	
Edge "sliding"		■	■				■	
Resource sharing	■	■	■			■	■	
ILP or MCF				■	■			
Congestion manipulation	■	■	■		■	■	■	■
History-based						■	■	■
Layer Assignment			■			■	■	■
Open source						■	■	

* Based on [5]

V. Parallelizing the Global Routing

Like many other processes in real world, some computer programs work sequentially, performing one operation after another, but many complex or large computational problems would take too much time to be completed using sequential processing. Those problems can often be divided into smaller ones and then computed concurrently. In other words the execution of tasks or calculations needed to solve the problem could be done at the same time (in parallel). Parallelism has

been employed for many years, mainly in high-performance computing. Focusing on power consumption reduction, parallel computing has become the dominant paradigm in computer architecture, mainly in the form of multicore processors [18].

In global routing, despite the use of heuristic methods or the implementation of improved algorithms, the process is still sluggish. This occurs for circuits with high integration scale and due to imposed constraints or physical variations attributable to fabrication processes like chemical mechanical planarization [1]. The execution time is an important constraint inside VLSI projects. The GR demands a large runtime when executed sequentially and a way to speed up the process is using parallelism (when possible) if the algorithm is scalable. That is whether it can be accelerated linearly with the use of various processors and whether the serial portion of the application is not too big to make the total runtime converge to it (*Amdahl's Law* [19]).

Madden [20] exposes some myths of parallel computing, he shows that the fact an algorithm is scalable, does not mean necessarily it will have high performance. In some cases is better to use the serial approach than the parallel, because the amount of effort and quantity of resources required to achieve a quality solution are important constraints. Much of the success of parallelism falls on the algorithmic efficiency and the main performance limitation of parallel approaches is due to some intrinsic serial characteristics of the tools.

From the state-of-the-art on global routing using parallel approaches, two main methods can be distinguished:

- Routing every net independently [21], this method leads to a problem because not all nets are independent, a situation that ends up generating sequential blocks with various nets and difficult a priori exploration of massive parallelism.
- The second method consists in partitioning the circuit [6] and treats each new part independently. With this approach, the biggest challenges are the partitioning itself and the interconnection of the boundaries partitions.

VI. Graphics Processing Units

We are in the golden era of computing through graphics processing units (GPUs), attributable to recent advances of the two largest companies in the industry, *AMD* and *nVIDIA* with the *CUDA* [22] technology and the open standard *OpenCL* [23]. This last one is a framework for writing programs that execute across heterogeneous platforms consisting not only of GPUs but also CPUs, digital signal processors and others.

What makes interesting of the GPU computing is its high processing capacity with a low cost on energy consumption, bigger area for components means more complexity and hence better performance. It is supported by the Pollack's rule [24], which states that performance of a chip is approximately equal to the square of its complexity ($performance \approx \sqrt{complexity}$). Put differently, "high-end video cards" have a considerably higher performance compared to high-end processors, with almost equivalent energy consumption. This is only possible due to the fact that the complexity on a GPU is partitioned over several smaller processing units (called *CUDA Cores* in *nVIDIA* models), also known as *shader units*. The argument of Fred Pollack is the same used by Intel to sell their CPUs with multiple processing cores [25].

The use of GPUs and parallel model for routing algorithms is an approach that could uncover an excellent solution to reduce execution time and improve router performance. Besides the GPUs, other form of parallelism is the use of a set of computers interconnected in *clusters* or *grids*. Clusters are groups of computers in a controlled environment and often dedicated to perform a specific task. Grids in this sense are no-common machines at different places that generally cede part of their processing capacity to projects through the internet [26]. In both cases the programming methodology is similar, they are multi-computers and are generally used message passing protocols for communication between nodes.

VII. Work Developed at the UFRGS

Important research has been done at the *Universidade Federal do Rio Grande do Sul* in the field of EDA. Despite the valuable investigation constantly performed at UFRGS that covers all steps of VLSI projects, here only will be referenced a couple of works related to global routing. Approaches that have shown improvements over the best ranked routers of the ISPD'08 contest.

Reimann [27] presented a global router that uses as principal tool the *rip-up and re-route* technique, 'with a differential method for sorting the nets'. In his approach were developed two versions of the same router, one (*WL version*) to achieve the shortest wirelength and the other (*RT version*) to seek the convergence of the solution as quickly as possible, with the lower number of iterations. *Minimum Spanning Tress* (MST) and *FLUTE* are the two forms used to build the routing nets. The Figure 2 shows the execution flow of the router.

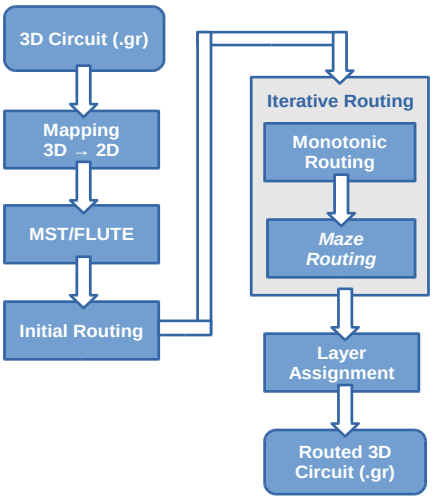

Fig. 2 Execution flow [27]

Despite the global router does not use techniques to identify congestion areas, nor post-routing optimizations and avoids any form of tuning for the benchmark circuits; it shows that is able to generate good results when compared with others academic solutions for the 3D circuits exhibited at ISPD'08. The WL version of the tool presents a difference, on average, of 1.78% more for the wirelength metric without considering the cost of the vias and 15.56% considering the via cost like unit of the wirelength; as for the RT version the difference was 3.82% and 17.03% more respectively.

In the context of this paper, is presented another work but focused on a different process of the physic synthesis. It is relevant by its approach and because the technology used on it can be extrapolated to the GR problem.

The placement is the VLSI process where the basic logic elements (cells) are organized inside the integrated circuit. This step is a NP-Hard combinatory problem and one form to find a satisfactory solution is using quadratic functions.

In [28], Flach et al., present an interesting form of using the power processing of a GPU to treat the cell placement problem. When the paper was written, there was not available a programming language like *CUDA* to employ easily the GPU for general purposes, thus the authors implemented a quadratic placement on *OpenGL*. They use *OpenGL* directives to manipulate elements of texture (matrix of pixels compounded of red, green, blue and alpha color channel), converting those elements in the tool memory and implementing simple algebra operations like *matrix-vector multiplication* and *dot product*.

The results present significant improvements of performance. In the *dot product*, *multiply and add*, and the *sparse-vector multiplication* operations the tool had a speed-up on runtime in the order of 1.95x, 3.45x and 3.05x respectively.

VIII. CONCLUSIONS

In this work, we took a look at the panorama of global routing nowadays, exposing the best ranked routers and comparing their characteristics.

We presented two important works developed at the UFRGS in the specific field. The Reinmann's approach shows results close to the obtained by the ISPD'08 contest competitors. From this can be inferred that the success of a router is not necessarily related to its complexity. In some cases the simplicity of a solution guarantees its success.

The work of Flach et. al, is important not only because it covers a physical synthesis step indispensable for the global routing process but also because it demonstrates the feasibility of using GPUs (employing either *CUDA* or *OpenGL*) to speed up computational problems like GR. Through the time, has been noticed that approaches originally conceived for different problems could be easily extrapolated for implementation inside EDA tools.

Most of modern global routers are a combination of well-known techniques and algorithms. How they are implemented inside the router and novel approaches could make a difference in terms of quality results.

The use of parallelism methods is a promising proposal to reduce the execution time of global routers. Despite the large power processing that could be achieved with those methods, they demand a considerable amount of resources that end up being a restriction in many cases. As exposed previously, various considerations have to be done when using parallelism as an approach in global routing inside EDA tools.

In order to conquer the open challenges, is not enough with just analyze current methods of global routing and learn from them, is equally important to have an open mind to imagine new efficient approaches.

REFERENCES

[1] C. J. Alpert, Z. Li, M. D. Moffitt, G. Nam, J. A. Roy, and G. Tellez, "What Makes a Design Difficult to Route", *Proceedings of the 2010 International Symposium on Physical Design* (*ISPD'10)*, pages 7-12, 2010.

[2] G. J. Nam, C. C. N. Sze, and M. C. Yildiz, "The ISPD global routing benchmark suite", *Proceedings of the 2008 International Symposium on Physical Design (ISPD'08)*, pages 156–159, 2008.

[3] G. J. Nam, M. C. Yildiz, D. Z. Pan, and P. H. Madden, "ISPD placement contest updates and ISPD 2007 global routing contest", *Proceedings of the 2007 International Symposium on Physical Design (ISPD'07)*, page 167, 2007.

[4] M. D. Moffitt, "Global Routing Revisited", *Proceedings of the 2009 International Conference on Computer-Aided Design* (*ICCAD'09)*, pages 805-808, 2009.

[5] M. D. Moffitt, J. A. Roy, and I. L. Markov, "The Coming of Age of (Academic) Global Routing [Invited Paper]", *Proceedings of the 2008 International Symposium on Physical Design (ISPD'08)*, pages 148-155, 2008.

[6] Y. Han, D. M. Ancajas, K. Chakraborky, S. Roy, "Exploring high throughput computing paradigm for global routing", *Proceedings of the International Conference on Computer-Aided Design (ICCAD'11),* pages 298-305, 2011.

[7] J. Hu and S. S. Sapatnekar, *A survey on multi-net global routing for integrated circuits*, Integration, the VLSI Journal, vol. 31, no. 1, pages. 1-49, 2001.

[8] C. Albrecht, "Global routing by new approximation algorithms for multi-commodity flow", *IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems (TCAD)*, vol. 20, no. 5, pages 622-632, 2001.

[9] M. Pan and C. Chu, "FastRoute: A step to integrate global routing into placement", *Proceedings of the 2009 International Conference on Computer-Aided Design* (*ICCAD'06)*, pages 464- 471, 2006.

[10] ——, "FastRoute 2.0: A high-quality and efficient global router", *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC'07)*, pages 250-255, 2007.

[11] Y. Xu, Y. Zhang and C. Chu, "FastRoute 4.0: Global Router with Efficient Via Minimization", *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC'09)*, pages 576-581, 2009.

[12] M. Hanan, "On Steiner's problem with rectilinear distance", *SIAM Journal of Applied Mathematics, vol. 14*, pages 255–265, 1966.

[13] M. Cho and D. Z. Pan, "BoxRouter: A new global router based on box expansion and progressive ILP", *Proceedings of the 43rd Design Automation Conference (DAC'06)*, pages 373-378, 2006.

[14] M. Cho, K. Lu, K. Yuan, and D. Z. Pan, "BoxRouter 2.0: Architecture and implementation of a hybrid and robust global router", *Proceedings of the International Conference on Computer-Aided Design (ICCAD'07)*, pages 503-508, 2007.

[15] J. A. Roy and I. L. Markov, "High-performance routing at the nanometer scale", *Proceedings of the International Conference on Computer-Aided Design (ICCAD'07)*, pages 496-502, 2007.
[16] M. D. Moffit, "MaizeRouter: Engineering an effective global router", *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC'08)*, pages 226-231, 2008.
[17] J.-R. Gao, P.-C. Wu, and T.-C. Wang, "A new global router for modern designs", *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC'08)*, pages 232-237, 2008.
[18] K. Asanovic, et al., "The Landscape of Parallel Computing Research: A View from Berkeley", Technical Report No. UCB/EECS-2006-183, University of California, Berkeley, 2006.
[19] G. M. Amdahl, "Validity of the single-processor approach to achieving large scale computing capabilities", *Proceedings AFIPS Conference*, pages 483–485, 1967.
[20] P. H. Madden, "Dispelling the Myths of Parallel Computing", *IEEE Design & Test of Computers, February*, pages 1–1, 2012.
[21] T.-H. Wu, A. Davoodi, J. T. Linderoth, "A parallel integer programming approach to global routing", *Proceedings of the 47th Design Automation Conference (DAC'10)*, pages 194-199, 2010.
[22] nVIDIA, "CUDA Parallel Computing Platform" [Online]. Available: http://www.nvidia.com/object/cuda_home_new.html
[23] Khronos Group, "The open standard for parallel programming of heterogeneous systems", OpenCL [Online]. Available: http://www.khronos.org/opencl
[24] P. P. Gelsinger, "Microprocessors for the New Millennium: Challenges, Opportunities, and New Frontiers", in *IEEE International Solid-State Circuits Conference*, pages 22-25, 2010.
[25] S-L. Garver, B. Crepps, "The New Era of Tera-scale Computing", Intel Software [Online]. Available: software.intel.com/en-us/articles/the-new-era-of-tera-scale-computing
[26] M. A. R. Dantas, *Computação Distribuída de Alto Desempenho: redes, clusters e grids computacionais*, Axcel Books do Brasil Editora, Rio de Janeiro, 2005.
[27] T. J. Reimann, *Roteamento Global de Circuitos VLSI*, Master Dissertation, Institute of Informatics, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.
[28] G. Flach, M. Johann, R. Hentschke, and R. Reis, "Cell placement on graphics processing units", *Proceedings of the 20th Annual Conference on Integrated Circuits and Systems Design (SBCCI'07)*, 2007, page 87.