Global Routing Congestion Reduction with Cost Calibration Look-ahead

Leandro Nunes, Ricardo Reis

PGMICRO – Programa de Pós-Graduação em Microeletrônica
Instituto de Informática - UFRGS
Av. Bento Gonçalves, 9500
Porto Alegre - RS / Brazil
leandro.nunes@inf.ufrgs.br
reis@inf.ufrgs.br

***Abstract*— This work presents two techniques for defining and treating areas that have high interconnection demand in VLSI circuits, during global routing. These techniques are applied in two steps over all global routing flow. First technique is executed in the pre-routing phase, where are identified regions with high interconnection density, (i.e. source or destination are in a large number reducing its ability to allocate interconnection); the second technique is applied within iterative routing phase, identifying and protecting those regions from having higher levels of congestion. These techniques were included in an existing global routing flow, called GR-WL, to validate the impact of its implementation, through the extraction of three global routing metrics: wirelength, total value of maximum overflow (TOF) and maximum obtained overflow (MOF). By running experiments using these techniques, improvements were up to 16% in reduction total congestion. The results in overflow reduction are more relevant with benchmark circuits for which there is still no valid solutions in the literature. Furthermore, the running times achieved were up to 30% faster when compared to the reference implementation, with a maximum impact of 1.39% in the total wirelength.**

***Keywords* – microelectronics, physical design, CAD tools, global routing, routing grid, cost calibration.**

I. Introduction

In the deep sub-micron technologies in addition to the continuous increasing complexity of the current chip designs, the delay of interconnect wires becomes the first order term in the total delay calculation of the VLSI circuits.

As part of the VLSI project flow, the Physical Design phase is divided into three steps: partitioning, placement and routing. The partitioning algorithms are used to separate the circuit into logical and disjoint parts; placement is the responsible to set positions to every cell in the circuit area; and routing aims to interconnect the logically related cells [3].

In the routing phase, there are two major steps: global and detailed routing. The goal of global routing is to simplify and plan the high level interconnections scheme in order to reduce the complexity of the detailed routing process, where the exact location of each wire and pin will be defined.

An usual approach to model the global routing problem is to use a virtual grid where each edge represents a set of interconnection wires and each vertex a discrete area of the circuit. The goal is to balance the demand of wires in each region to provide a plan to the detailed router draw each individual wire [3].

In order to negotiate the demand and priority of the wires in a circuit region, the router usually set costs to each region. The costs are raised for the regions that has high interconnect demand.

The aim of the present work is to propose techniques to improve the global routing step on the early definition of appropriate costs to congested regions.

The decisions made at routing time will affect directly some relevant parameters of the resultant layout: power consumption, max operation frequency (clock), manufacturing yield, and some side effects like thermal dissipation, coupling capacitance and noise can be favored or avoided, depending on the routing scheme.

II. Problem Formulation and Related Works

From a global routing perspective, the routing region is divided into a set of global cells (also called *tiles*) that can be represented as a graph $G = (V, E)$, where V represent all circuit regions, denoted by $v_i \in V$. Each global edge $e_{ij} \in E$ corresponds to a logical connection between circuit regions that contains a capacity C_{ij}. The capacity define how many wires each edge support.

Considering the total number of existing nets N that represents a circuit, there is a list of nets $n_i \in N$. Each net n_i contains a set of pins P_i that are placed into a region v_i.

The common global routing goal is to connect all the pins of all the nets in N through the edges E, without to extrapolate the capacities of each edge [3]. If any edges have more interconnections than their capacity, then this edge has overflow.

In the state-of-the-art global routing literature, some metrics are used to measure the router performance [3] [4]:

- The wirelength (WL) that represents the sum of the size of all wires in the routing solution;
- The total overflow (TOF), that denotes the sum of all edges overflow and
- The maximum individual edge overflow (MOF) obtained in the routing solution.

Currently, more metrics are extracted from a routing solution, in order to attend to multi-objective routing algorithms [3] [4].

In the last six years, two global routing challenges promoted in the context of the ACM International Symposium on Physical Design (ISPD 2007 and 2008) were contributed to increase the interest on global routing tools and provide updated set of benchmarks that simplify the comparison of the academic tools [2][4][5].

The state-of-the-art routing tools [2][5] such as NTUgr, BoxRouter 2, NCTU-GR, GR-WL, GRIP, FastRoute 3.0, MaizeRouter, Archer and others, publish their results following the ISPD 2007 and 2008 [4] guidelines.

The global routing flow consists, commonly in four steps: preamble, initial routing, iterative routing and layer assignment.

In the **preamble** phase, the input data is stored in the routing data structure: for grid-based structures that are more common, the tool can model it using a 3D routing with all layers or 2D condensed grid, mapping all routing layers to a flat matrix, in order to reduce the effort of maze routing algorithms.

An optional step inside the preamble is the **net decomposition**, where the multi-net pin is decomposed into a set of two-terminal wires. The most popular approaches are FLUTE algorithm and MST expansion [4].

After the preamble, **initial routing** phase will provide a rough solution for nets routing, commonly using a lightweight algorithm such as pattern routing or monotonic routing [5].

The next step is **iterative routing**, where the routing solution is incrementally constructed. In that way, ripup-and-reroute (RRR) based techniques are widely employed [3] [5], combined with monotonic routing [5].

Once a solution for routing exists, **layer assignment** is the step that aims to distribute the generated wires in to the routing layers, in order to provide a final solution for the circuit.

III. GR-WL Routing Flow

GR-WL is an iterative, sequential and negotiation based global router with focus on wirelength reduction. This work was developed in [5].

The GR-WL routing flow supports MST or FLUTE to decompose the multi-pin networks into a set of two-pin wires. The best experimental results were obtained with MST algorithm decomposition.

The routing flow consists in an initial routing step, based on pattern routing “I” and “L”, where the small networks (half perimeter less or equals to 3) are routed.

An iterative routing step, based on ripup-and-reroute is executed after the initial routing. The approach to iterative routing uses an A*-based maze router combined with an monotonic routing algorithm.

Monotonic routing algorithm is disabled when the total congestion is 97.5% completed in order to reduce the routing execution time when the most complex nets that still remains to be routed without overflow.

During the iterative routing the expansion area is gradually incremented in order to control the span of the networks.

After the iterative routing, considering that a valid solution was found, a layer assignment step using a greedy algorithm is executed, in order to distribute the wires by the routing layers. Otherwise, after twenty iterative routing rounds without progress, the router flow is finished without success.

IV. Cost Calibration Look-ahead

After an analysis of GR-WL routing flow and benchmarks execution, it's possible to observe that the top offender nets are placed in high interconnect demanding areas

It is possible to see in the Fig. 1, that only 10% of the total iterative routing rounds are spent on solving more than 90% of the circuit nets, the remaining iterations are used to solve the 10% of nets and on overflow solving.

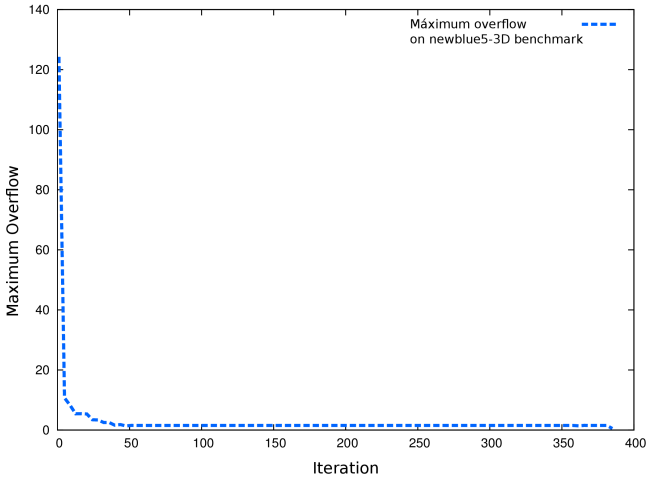

Fig. 1 Maximum overflow in newblue3 solving with GR-WL

One of the main challenges of a grid-based global router is to define appropriate costs to the grid, in order to provide to the routing algorithms the actual situation of the resource demand in each area of the circuit. So, the networks will avoid congested areas without generate low occupation islands.

In that way, the proposed technique aims to help the cost calibration on the routing grid, as early as possible, in order to spread the interconnection wires, in order to reduce the overflowed areas. The approach to achieve this goal is to calibrate the grid costs based on demand and congestion metrics.

The proposed cost calibration techniques were applied in two steps of GR-WL flow: just after initial routing, when the demand of each edge is evaluated; and during the iterative routing, when the occupation and actual congestion can be calculated.

A. *Initial routing cost calibration*

The process of cost calibration after the initial routing aims to classify and mark nodes that have high demand on interconnections.

In the current implementation, the criteria to select nodes in that areas is based on a configurable threshold value, that

represents the minimum occupation to a node be considered a high interconnection demand node. This value is represented by β.

Given a node n in the global routing grid, Cn represents the sum of capacities of all edges connected to n. The demand of n is denoted by D_n that means how many nets have a terminal place in the node n.

A node will be considered as a high demand node if and only if:

$$C_n / (D_n \times 100) > \beta$$

There are three parameters that guide the process of expansion and cost increment to the high interconnection demanding nodes: Max increment, denoted by ψ; increment amplitude, denoted by α; and cost distribution function.

The **max increment** represents the value that will be directly added to the historic cost of high interconnection demand nodes. The historic cost was chosen to be incremented, in order to affect permanently the cost of these nodes.

In order to treat the areas that the high interconnect demand nodes are located, the **increment amplitude** parameter will set the size of the perimeter around the nodes will have their costs adjusted.

There are some side effects that will occur if the value of ψ or α is not well tuned. The most visible side effect is the low congestion island that represents low overflow areas inside an overflowed area. Other possible side effect is the abnormal increment in wirelength, caused by the long detours caused by low congestion islands.

In the current implementation, three cost distribution methods were defined to evaluate their impacts.The first method will statically set the costs to ψ for all expanded nodes, limited to α nodes away from the high interconnect demand node. The Fig. 2 illustrates the resultant color map of the static function.

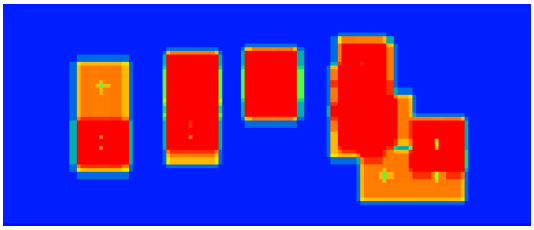

Fig. 2 Statically distributed costs to expanded high demand nodes.

Observing the resultant solutions of static functions was possible to verify low overflow islands and an increment in wirelength. In order to mitigate this problem, two alternative approaches were proposed.

Considering the impact of high interconnection demanding nodes will decrease in the nodes that are not close, a linear decreasing cost function was implemented. The Fig. 3 illustrates the color map of the influence of the cost increment variation using this function. Warmer colors in the figure represent historic cost increment areas.

At this point of the routing process, with all costs set to 1, even a small increase on historic cost will impact the final solution and provide information to the global router found alternative paths that avoid that regions.

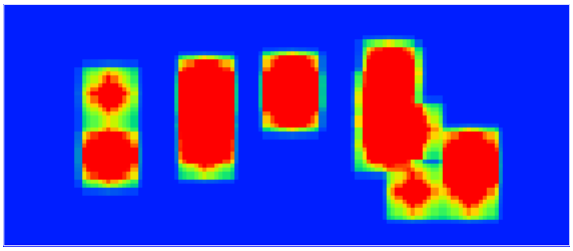

Fig. 3 Linearly distributed costs to expanded high demand nodes.

In order to offer a more detailed cost calibration and reduce the penalty on neighbor regions to avoid low overflow islands, the exponential function was proposed.

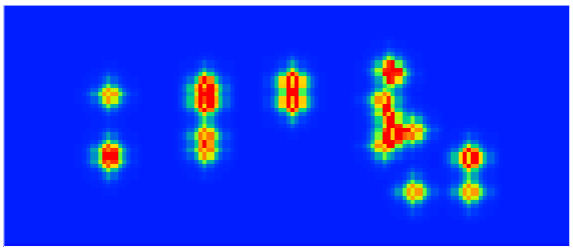

Fig. 4 Exponentially distributed costs to expanded high demand nodes.

After all high interconnection demand nodes and perimeter regions are marked and the historic cost was incremented, the routing flow can go ahead with iterative routing.

B. *Iterative Routing cost calibration*

During the iterative routing, this technique will calibrate the penalization value in order to encourage the nets to avoid congested areas early.

The cost calibration during iterative routing act in three steps, similar to [1], but with some adjustments:

- Critical node identification
- Congestion border expansion
- Cost increment

In the first step the critical nodes are identified, in order to find the overflowed zones. By critical node, we define the nodes where overflow equals to the maximum overflow.

It's common to found, in the regions that a high overflow is present, a border with an overflow gradient. These nodes are marked in the second phase of this technique (border expansion), in order to define more relevant areas than spare points of maximum congestion.

There is a tolerance, based on the overflow present in the neighbor nodes, in order to limit the expansion border width. This tolerance value, that was experimentally defined with 10% of the maximum congestion, and grows up to 40% in the final iterations.

For the high congestion nodes, the penalization cost is incremented by 80%. For the expansion border will have a linear decrement of 3% in cost increase, starting from 80%.

V. Experimental Results

The reference implementation (GR-WL) results, shown in the tables below, refers to a global router configuration that uses MST to decompose nets to two-pin wires, and via cost set to 1. It means that each via will be considered as a wire with size 1.

ISPD 08 benchmarks are used as input data, and a comparison with the original GR-WL [5] and NTUgr [1] is detailed with the tables I, II and III. Detailed information about the benchmark suite is available in [2].

For the cost calibration look-ahead techniques, the parameter values are defined with the following values: $\psi = 10$; $\alpha = 5$ and $\beta = 50$. The exponential cost distribution model was applied for initial routing and iterative routing cost increment steps.

In the Table I the overflow results are presented, only for the benchmarks that are not successfully solved. For these circuits, it's possible to observe a reduction up to 16% in the newblue1 (nb1) and newblue7 (nb7), for the total overflowed nodes (TOF) metric.

TABLE I
Total and maximum overflow metric.

	NTUgr		GR-WL		Ours.	
Circ.	**TOF**	**MOF**	**TOF**	**MOF**	**TOF**	**MOF**
bb2	118	4	122	2	138	2
bb4	410	10	780	6	1090	6
nb1	212	4	588	2	504	2
nb3	33636	374	36280	1194	37382	608
nb4	284	284	462	2	460	2
nb7	906	6	5172	4	4620	4

In the wirelength metric, it's possible to observe an increase up to 1.39%, when compared with the reference implementation (GR-WL), as a side effect of the interconnections spreading, due to the cost calibration.

TABLE II
Wirelength compared with NTUgr and GR-WL.

	NTUgr	GR-WL	Ours.
Circ.	**WL**	**WL**	**WL (diff. to GR-WL)**
ad1	5609943	6380538	6418268 (+ 0.59%)
ad2	5421602	6084400	6101664 (+ 0.28%)
ad3	13649616	15155118	15182090 (+ 0.17%)
ad4	12408382	13981761	13992148 (- 0.07%)
ad5	16358988	18102395	18149818 (+ 0.26%)
bb1	5949946	6685407	6741135 (+ 0.83%)
bb2	94667229	11204002	11181906 (- 0.2%)
bb3	13491167	15594684	15688085 (+ 0.59%)
bb4	24049722	27602383	27481693 (+ 0.56%)
nb1	4824320	5437199	5495484 (+ 1.07%)
nb2	7769446	9245356	9266090 (+ 0.22%)
nb3	11005863	13316366	13218553 (+ 0.73%)
nb4	13442722	15329779	15410331 (+ 0.52%)
nb5	24021512	26983599	27360317 (+ 1.39%)
nb6	18735225	20708074	20769146 (+ 0.29)
nb7	37485508	42224489	42503107 (+ 0.65%)

Execution time is reduced in the small circuits of the benchmark suite. It is an expected effect of the early cost calibration, applied in the proposed techniques.

In other hand, early cost calibration can cause a side effect in wirelength increasing if the cost is over or wrongly incremented. As shown in the Table II, the increase in wirelength was not significant (up to 1.39%).

TABLE III
Execution time (in minutes) compared with NTUgr and GR-WL, with net decomposition using MST.

	NTUgr	GR-WL	Ours.
Circ.	**Exec. Time (min)**	**Exec. Time (min)**	**Exec. Time (min)**
ad1	46.8	46.2	41.7
ad2	7.6	15.3	13.1
ad3	37.8	45.4	47.5
ad4	14.3	9.4	10.5
ad5	110.9	120.9	103.1
bb1	136.2	165.9	182.1
bb2	215.9	281.3	229.6
bb3	29.9	56.3	61.5
bb4	313.1	197.1	161.7
nb1	157.5	89.8	108.3
nb2	4.3	5.4	3.5
nb3	183.1	141.6	140.6
nb4	254.5	323.5	213.2
nb5	117.9	283.8	182.5
nb6	72.6	164.6	176.4
nb7	1421.0	974.2	1138.2

VI. Conclusions

In this paper we have presented two techniques to identify and treat areas with high interconnection demand and high congestion.

Three cost spreading functions were tested: linear, static and exponential. The best results were found with the exponential distribution.

The result shows that these techniques can impact positively to reduce the overflow in circuits with high level of congestion, without significant side effects on wirelength metric.

Future works will target improvements on the spreading amplitude parameter in order to dynamically quantify the size of cost increment area. Also, new cost distribution functions need to be experimented, in order to spread and reduce the congestion, and reducing the side effect impacts.

References

[1] H.-Y. Chen, C.-H. Hsu, and Y.-W. Chang, **High-Performance Global Routing with Fast Overflow Reduction.** Asia and South Pacific Design Automation Conference (ASP-DAC), Yokohama, Japan, pp. 582-587, January 2009.

[2] G.-J. Nam, C. Sze and M. Yildiz. **The ISPD Global Routing Benchmark Suite.** Proceedings of the 2008 international symposium on Physical Design (ISPD), pp. 156-159, New York, USA, 2008.

[3] J. Westra, P. Groeneveld, Y. Tan, P. H. Madden. **Global Routing: Metrics, Benchmarks and Tools**, 2008.

[4] M. D. Moffitt. **Global routing revisited**. In Proceedings of the 2009 International Conference on Computer-Aided Design (ICCAD '09). ACM, New York, NY, USA, 805-808, 2009.

[5] T. J. Reimann. **Roteamento Global de Circuitos VLSI,** Microelectronics master thesis, UFRGS, Porto Alegre, Nov. 2011