# Equipotential Two-Sided Planar Routing for QCL Channels

Glauco Borges Valim dos Santos and Marcelo de Oliveira Johann
*Pontifícia Universidade Católica do Rio Grande do Sul – Campus Universitário II – BR 472 Km 7*
*CEP 97500-970 – Uruguaiana, RS – Brasil. Phone: +55(55)413-1515, Fax:+55(55)413-1280*
*E-mails: glauco@pucrs.campus2.br, johann@pucrs.campus2.br*

## Abstract

*In this work we investigate the reduction of the number of vias in Quickly Customized Logic (QCL) channels based on Single Row Routing Problem (SRRP) and Equipotential Two-Sided Planar Routing (ETSPR). Quickly Customized Logic (cited in [8] and [2]) is a gate-array architecture in which the customization of the IC is done in a single layer. The first step based on [4]'s SRRP Algorithm was done with good results in channel height reduction, missing some optimizations to be done.*

## 1. Introduction

The Quickly Customized Logic (QCL) model proposed in [7] has been utilized with success in ÁGATA project [1] because of the conditions it provides for the application of the Left Edge Algorithm [5], which produces optimal solutions in linear time. Such model consists of fixed vertical conductors in the lower metal layer called underpasses (represented in light gray in Fig.1(a)). Over them horizontal routing tracks (dark gray) are placed connecting to them through vias (dark).

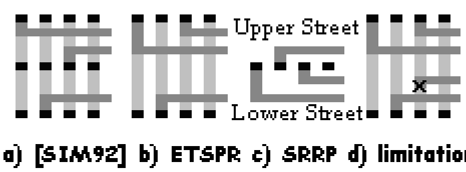


Figure 1. [7] model (a), ETSPR (b) and SRRP (c) models, and an ETSPR limitation (d).

Such channels are rounded by transistors rows from where assignments to underpasses come from. And generally, as the case of ÁGATA's gate-arrays, a river routing is used for making the connections of this assignment. The model created by [7] needs a track of vias between each pair of horizontal routing tracks. The reduction for only two tracks of vias, located in the channel's upper and lower boundaries, frees up additional routing tracks for real connections, or it may decrease the channel height, as considered earlier in [3]. Such new model is known as Equipotencial Two-Sided Planar Routing (ETSPR) in [6]. In [6] only a subset of the set of all the nets are routed over the cell.

Equipotential Two-Sided Planar Routing (Fig.1(b)), presents similarity with Single Row Routing Problem (SRRP [5]). SRRP is the problem of routing a set of nets with two or more terminals along a line of points (single row). The area above the single row (the line of points) is called **upper street** and the area below the single row is called **lower street**. Notice how the upper and lower streets of an SRRP representation in Fig.1(c) correspond, respectively, to the lower and upper region inside the channel in Fig.1(b). Besides that similarity, differences must be considered, as some connection patterns are possible in ETSPR and not in SRRP. (As shown in Fig.2(a)). Also SRRP lower and upper street are not necessarily directly correspondent to the upper and lower region of an ETSPR channel (Fig.2(b)). However, it is still possible to translate SRRP solutions to ETSPR.

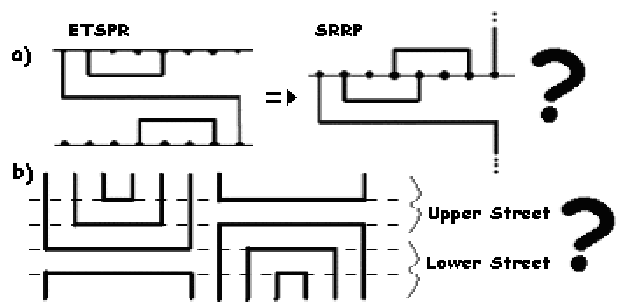


Figure 2. Differences ETSPR / SRRP.

Horizontal connections make underpasses more easily unreachable by others (horizontal connections) since they must be all planar in the ETSPR model (see Fig.1(d)), while they do not represent vertical constraints to each other in the original ÁGATA's model.

An approach based on [4] was selected from a set of three theoretical approaches defined for the problem, since its viability was more assured, there are previous works based on it with models similar to the model present here ([8] and [2]).

## 2 The approach based on [4]

Approaches based on [4]'s SRRP algorithm have been presented before, like [8] and [2], but since [8] does not satisfy the requirements for problems of considerable complexity (does not even consider the use of doglegs in its routing), and [2] does not describes clearly how it does that (and we couldn't get access to all its references), this work follows the same path but starting on [4] instead of the later works.

First some limitations of [4]'s algorithm must be considered. It does not limit the **Between Nodes Congestion** (***$C_B$*** **-** number of crossings between terminals [5]), what increases its chances of success. For ETSPR channel routing, some terminals are adjacent, what disables the utilization of underpasses between them for doglegs. Such cases limit $C_B$ across the channel, once each assigned underpass correspond to a point in the single row. It may be possible to shift adjacent or close terminals that present a $C_B$ greater than the number of free underpasses between them, in such way that the river routing could still be done. More tracks may be used for the river routing once satisfactory results are obtained in decreasing channel height.

In [4], all possible orders of nets that pass by each point of connection with the single row are considered. Fig.3(a) shows the nets that cross each point of connection in a single row. Based on a classification of these points over their positions, different behaviors are done. Fig.3(b), Fig.3(c) and Fig.3(d) shows the behavior for the first point of a net. In this case all the possible orders are considered for introducing this point in routing. For the first point only one order is considered, as shown in Fig.3(b). This order will result in different order possibilities when introducing the next net in routing in the next point (Fig.3(c)), and so on. Fig.3(d) shows only the orders for one of the orders in Fig.3(c), since the number of possibilities grows for each first point of each net that is inserted on routing. There are other behaviors for the last and the middle points of each net. Also is needed to check the streets overflow. This example was shown to demonstrate the high size of [4]'s search tree and try to justify the utilization of a simplified version.

Now, a simplified version of [4]'s single row routing algorithm is been implemented. This simplified version doesn't use the search tree proposed in the [4], but a simplified and more efficient version with reduced number of branches, allowed by linked lists, together with a new concept of storage of the segments. It is also less complete then [4] original proposition, but this is justified for the fewer necessity of memory and processement, and the use of some policies along the routing that try to get as the best solution (or at least one of the best) the first result instead of considering all the possible results for later selection.

Following [4] the information about **start**, **stafin** (correspondent to [4]'s middle point) and **finish** points (points of connection with the single row) of all segments is kept. A segment is the horizontal part of a net connection and its right hand edge correspond to the start point, his left hand edge to the finish point and points of connection with the single row in the middle of the segments are the stafin points. An iteration deals with each of this points, ordered by their *x* coordinates, introducing start points nets in routing, and excluding finish points nets, doing both for stafin point nets.

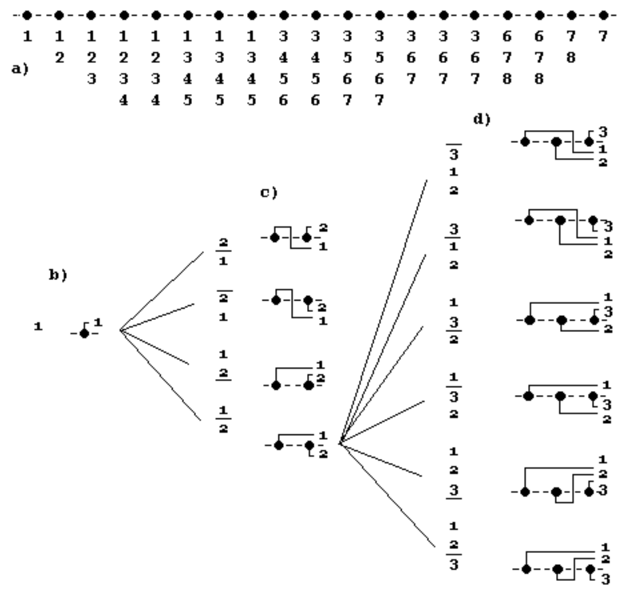


Figure 3. Example of [4]'s behaviour.

Each introduction inserts the start point net in one of two linked lists correspondent to the upper or lower street nets positions above or below that point. This is made following some policies created to assure a better routing. The policies will select the first street to insert the start point in routing, creating a branch in the seach three. Other branch will be created for the other street once the prior one does not succeed in getting a routing solution for the permitted height.

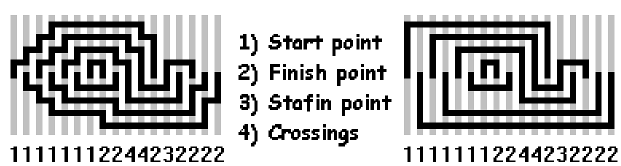


Figure 4. Example of stored data along the routing solution and correspondent layout

Each exclusion of a finish point implies in replacing all segments before him from his linked list to the other street, what implies in the increase of the $C_B$ between the current position and the prior one, as shown in Fig.11(b) when a finish point is found for the net of the segment pointed to by an arrow in Fig.11(a). This is the algorithm behavior. After all positions with start, stafin and finish points are covered the layout may be obtained from the information stored at each position, reducing also unnecessary bends (Fig.4).

## 3. Implementation

The algorithm was completely implemented in C in IBM-PC, running Linux. It was tested with routing problems of the channels of 5 test circuits: copel1, powpad11, powxor11, m8255 and timer14.

It was also implemented for validation a visualization toll in Java. The visualization tool prints the results over a horizontal centralized line representing the single row. Under each vertical connection to the single row and crossing is printed in light gray an underpass, and it is possible to see clearly the shifting necessities. For simplification, the tool prints the resulting layouts with no bends at all but some bends must be kept for the real layout to assure the perfect fit between SRRP's upper and lower streets in an ETSPR channel.

For now it is missing the implementation of the shifting procedure of the algorithm results once they present, many times, the necessity of free underpasses between points. In the case of points correspondent to adjacent or closest underpasses assigned to nets, this necessity of free underpasses was, many times, greater then the existing ones (none for adjacent underpasses). After replacing the nets that need to be shifted, the river routing should still be done somehow. It should reach underpasses in positions such as to satisfy between nodes congestion ($C_B$).

## 4. Results

The algorithm solved all channels problems with satisfactory results for 4 of the 5 circuits. In Tab.1 it is presented each tested channel with the original heights and the heights obtained with the algorithm implemented so far. The height presented for each circuit is the highest of all of its channels.

**Table 1. Results in the original and the new model.**

| Circuit name | Number of channels | Original height | Obtained Height |
|---|---|---|---|
| copel1 | 17 | 14 | 14 |
| m8255 | 11 | 14 | 10 |
| powpad11 | 12 | 14 | 8 |
| powxor11 | 12 | 14 | 9 |
| timer14 | 8 | 11 | 9 |

The only circuit with no satisfactory result was copel1, the more complex, but even so we can notice that at least it was not needed a channel height greater than the original one.

Some results represented by the visualization tool can be seen in the figures that follow. The tool originally uses different colors for the connections of different nets, but once this work may be seen in gray scale all connections are represented in dark.

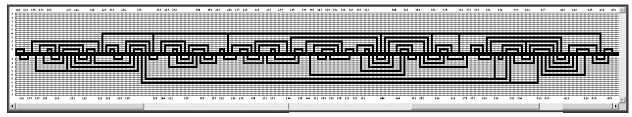

**Figure 5. M8255 channel 11.**

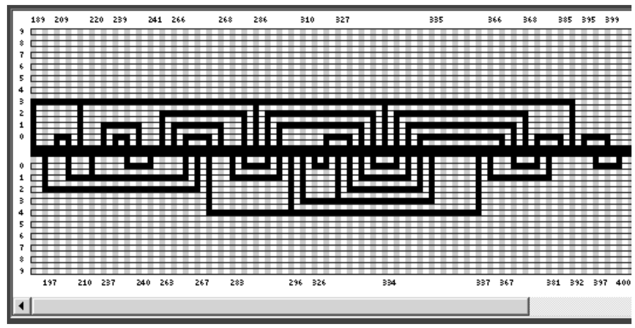

**Figure 6. M8255 channel 5.**

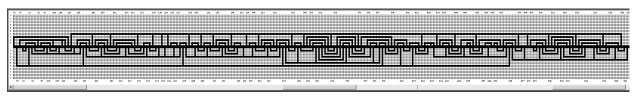

**Figure 7. Powdpad11 channel 8.**

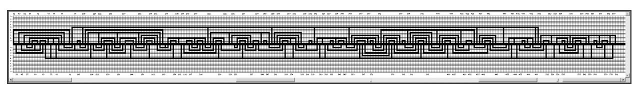

**Figure 8.Powxor11 channel 14.**

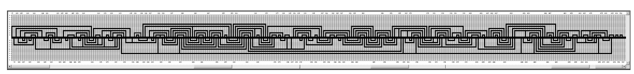

**Figure 9. Copel1 channel 8.**

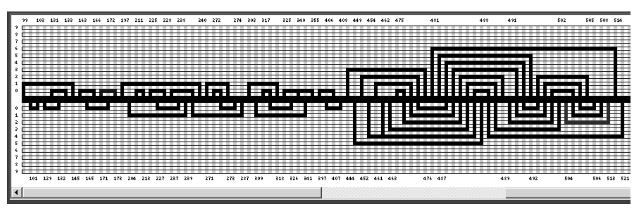

**Figure 10. Copel1 Channel 1.**

Fig.6 and Fig.10 may be more representative because their channels solutions are less complex than the other ones.

## 6. Conclusions and Future Work

What was implemented so far only provides SRRP connections, ignoring other shapes of connections allowed in ETSPR that could strongly improve the planar routing.

The first new shape to be considered is the same as the **Top-Bottom crossing** proposed by [6], and [2]'s **crank-**

**shaped wire**. Such a shape implies in a connection with the beginning in the channel upper boundary and another one in the lower boundary, as shown in Fig.2(a), and increase the number of branches in the search tree since it means an extra possibility. That means that when a finish point was found as shown in the example of the first image of the Fig.11(a), the algorithm will not proceed as usual, placing segments below the finish one for that street in the other street, as shown in Fig.11(b), with the SRRP representation at left and the ETSPR representation at right. In this case it is necessary to evaluate whether it is better to connect the finish point in the usual way or to perform the new shape.

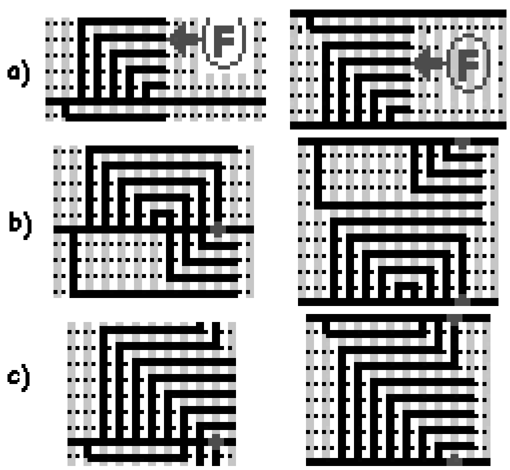


Figure 11 - New shapes in SRRP and ETSPR.

We have to look at the right hand side image of the Fig.11(a), where the ETSPR representation of the SRRP routing made until now is shown, and notice that the segment with a finish condition got only two segments above it and three below. So, if a connection like the one shown in Fig.11(c) is made, fewer crossings are needed, as it can be seen more clearly in the ETPSR representation.

Another optimization is to change the actual policies for insertion of start points in routing. Observations have shown that better choices could be done.

## 6. References


[1] L. Carro et al, “Ambiente ÁGATA de Projeto Versão Beta 2.0”. Tercer Workshop Iberchip. Proceedings… México, 1997. p.495-503.

[2] S-k. Dong, et al, “Two Channel Routing Algorithms for Quickly Customized Logic”. EDAC. Proceedings... IEEE, Los Alamitos, 1993, p.122-126.

[3] M.O. Johann, “Estruturas de Roteamento em Circuitos VLSI”, Porto Alegre: PPGC/UFRGS, 1997.

[4] S. Sahni, R. Raghavan, “The complexity of single row routing”, IEEE Trans. on Circuits and Systems, v CAS-31, n 5, 1984.

[5] N.A. Sherwani, Algorithms For VLSI Physical Design Automation, Kluwer, Massachusetts, 1993.

[6] N. Sherwani, S. Bhingarde, A. Panyam, Routing in the Third Dimension: From VLSI Chips to MCMs, IEEE Press, Piscataway, 1995.

[7] S.A. Simões et al, “Matriz gate array cmos avançada, configurável por um único nível de metal”. Congresso da SBMICRO. Anais...: SBMICRO/USP, São Paulo, 1992, p.281-291.

[8] Y. Sun, “A Channel Router for Single Layer Customization Technology”. ICCAD. Proceedings... 1991. p.436-439.