E8.

VLSI Routing Methodology for Quadruple Metal Sea-of-Cells Design

Howard H. Chen and C. K. Wong
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598
U.S.A.
(914) 945-1385

ABSTRACT

This paper presents a general sea-of-cells routing methodology for the most advanced VLSI technology today, where four layers of metal are available for signal interconnection. As routing channels are defined over the cells, the entire chip can be fully populated with circuit cells. The routing process starts with multi-layer global routing, followed by vertical line packing and horizontal channel routing. The global wires are converted to pseudo pins in the vertical channel, and horizontal channels are completed by using a novel four-layer channel router.

1. Introduction

The most advanced VLSI technology today has made possible the design of large and complicated circuits with four layers of metal. The additional layers of metal make the traditional standard cell and gate array design give way to the denser sea-of-cells or sea-of-gates design style. Fig. 2 shows a sea-of-cells chip design, where rows of cells are placed immediately adjacent to each other, and routing channels are defined over the cells. The abutment of neighboring cell rows allows the cells to share a common power bus. Large macros are also designed to match the underlying power bus structure. Since more than two routing layers are available over the cells, there is no need to reserve routing channels between adjacent rows of cells. The design of dense macro cells with minimal porosity can also be achieved, as global wires are allowed to run over the cells in both horizontal and vertical directions.

2. Sea-of-Cells Routing Methodology

Without proper floorplanning, a general sea-of-cells design can only be wired by a maze router, in order to handle various blockages and randomly located pins all over the chip [1]. However, if all the basic cells have the same height and the pins are only located at the top, middle, and bottom of each cell, we can take advantage of this row structure and generate routing channels over the cells. The pins don't have to be perfectly aligned horizontally, as long as a horizontal channel boundary can be generated along the pins. If the pins only exist in the middle of circuit cells, the channel boundary will be in the middle of cells. The upper half of one cell row will be combined with the lower half of the cell row above to form one routing channel. Similarly, if pins only exist on the top and bottom of circuit cells, two routing channels will be generated per circuit row. When pins are located at top, middle, and bottom of circuit cells, as many as 3 routing channels can be generated per circuit row.

The routing methodology for a row-structure sea-of-cells design can be divided into four steps: namely, global routing, vertical line packing, horizontal channel routing, and final overflow embedding.

2.1. Global Routing

To determine which routing channels should be used for each net, we must perform global routing and generate the global path for each interconnection. The global routing grid is a coarse grid, where each horizontal row of cells corresponds to a horizontal routing channel and each vertical column of global cells forms a vertical channel. While the horizontal boundaries of global cells coincide with the horizontal channel boundaries, the positions of vertical boundaries are determined artificially. The number of wiring tracks available (capacity) in a vertical channel is usually adjusted according to the capacity in a horizontal channel to avoid any directional bias during global routing.

A global path consists of a series of global segments assigned to designated horizontal or vertical channels. For quadruple metal technology, we assume that M1 and
M3 layers are used for horizontal connections, and M2 and M4 layers are used for vertical connections. If the traditional two-dimensional global routing were to be used, the total number of tracks available on M1 and M3 must be lumped together as the horizontal channel capacity. Similarly the total number of tracks available on M2 and M4 will be used as the vertical channel capacity.

However, if the layer assignment of global segments is desired, a three-dimensional global router must be used. In 3D global routing, each layer has its own capacity; i.e., horizontal channels will have capacity constraints on M1 and M3, and vertical channels will have capacity constraints on M2 and M4. The use of M4 layer for vertical pin access is discouraged, because it will also block the vertical track on M2 layer. Wrong-way wires, which do not run in the preferred direction on their respective layers, may be used occasionally for pin access or via reduction.

2.2. Vertical Line Packing

After global routing, the vertical wire segments are to be embedded first by vertical line packing. If the layers of vertical segments are not determined during global routing, they will also be assigned at this time. Each vertical column of global cells forms one vertical channel, and each vertical channel can be routed independently. We use a line packing algorithm to pack all the vertical wire segments which are globally assigned to the same segments with physical pins of the same net to avoid unnecessary wire jogging. When alignment is not possible, a dogleg may be introduced to break the vertical segment, so that the divided segments can be packed separately. During the line packing process, the M2 layer will be packed first if there still are tracks available. The remaining vertical segments are then packed on M4, which is free of blockages, subject to a set of constraints to minimize the wire densities in horizontal channels.

2.3. Horizontal Channel Routing

When all the vertical global segments are embedded, they will be converted to pseudo pins on each horizontal channel boundary which they cross. Since the physical pins are also located on or near the boundary, all the physical pins and pseudo pins of a net can be connected by a horizontal channel router.

A general four-layer channel routing algorithm is developed for the quadruple metal technology. The four-layer channel router makes horizontal connections on M1 and M3, and generates vertical segments on M2 and M4. Due to the complexity of four-layer interconnections, the channel routing constraints not only depend on the pin locations, but also the layer of each pin. If two pins on different layers (M2 and M4) overlap each other at the same location, they can only be connected to vertical segments on their assigned layers. However if a pin is not overlapped by another pin, it can be accessed from either M2 or M4. When two pins with the same x location are on opposite boundaries of a channel, it is desirable to have them on different layers to avoid the vertical constraint.

From the location and layer of each pin, we can determine the layer of its vertical segment, the channel routing constraints, and the layer of its horizontal connection. Although stacking via is allowed, the layer change from M4 directly to M1 is undesirable, because it also blocks the M2 and M3 tracks. To generate the constraint graph [2], we first assume that all the horizontal connections are on M1 and introduce only the M1 constraints. Then we assume that all the horizontal connections are on M3 and generate the M3 constraints. Inter-layer constraints between M1 and M3 will not be generated until the routing starts. Fig. 1 shows the weighted constraint graphs WCG1 and WCG3 for M1 and M3 respectively. Each horizontal connection will be represented by one node in WCG1 and one node in WCG3. In each constraint graph, the vertical constraints are represented by directed edges and horizontal (nonoverlapping) constraints are represented by undirected edges [3]. The weight of each edge, which represents the minimum separation required between the corresponding nodes, is calculated according to the design rules (wire width, spacing) on each layer.

As we proceed to embed the horizontal segments, the graph will be updated accordingly. If net A is embedded on M1, the corresponding node A in WCG3 will be deleted. Similarly, if net B is assigned to M3, the redundant node B in WCG1 will be deleted. Inter-layer constraints between net A on M1 and net B on M3, which are represented by dotted edges in Fig. 1, will be added between WCG1 and WCG3. The wire embedding and graph updating process continues until all the wires are embedded. The final WCG1 and WCG3 will be merged into one graph, where only one node exists for each net, and all the undirected edges are assigned directions to represent the relative positions between wires.
2.4. Rip-Up and Reroute

In the sea-of-cells design, where power structures are fixed and interconnections are made over the cells, the routing regions cannot be expanded. In order to achieve 100% routing completion, it may be necessary to use rip-up and reroute to embed the overflows. Local rip-up and reroute will be performed within the routing channel and its two adjacent channels. As a last resort, global rip-up and reroute can also be used to change the global paths of overflown connections.

3. Conclusions

A general VLSI routing methodology has been presented for the state-of-the-art quadruple metal technology. Using the sea-of-cells design style, where routing channels are defined over the cells, more than 20,000 circuits can be easily embedded in a chip with 50,000 basic cells. The chip image of Fig. 2 and its power bus structure on different routing layers are shown in Fig. 3, Fig. 4, and Fig. 5. Following the preferred routing direction on each layer, this routing methodology efficiently utilizes the four routing layers over the cells and thus provides a systematic solution for VLSI sea-of-cells design.

4. References


5. Acknowledgement

The authors would like to thank Dr. Donald T. Tang and Dr. Zeev Barzilai of IBM Research Division for their support of this research.
Fig. 2. Sea-of-Cells Design.

Fig. 3. M1 Image.
Fig. 4. M2 Image.

Fig. 5. M3 Image.