

Design of VLSI-Layouts by Genetic Algorithm

Volker Schneck, Oliver Vornberger
University of Osnabrück

Keywords: Genetic Algorithms, VLSI-Design, Floorplanning, Routing

Introduction

VLSI design is a problem area with many non numerical, but compute bound problems. Because of the increasing complexity of VLSI circuits, there is a growing need for new solution methods. In this project a parallel genetic algorithm for the physical design of VLSI chips is developed. The algorithm solves different – normally consecutive – problems simultaneously: Together with the placement of cells the total wiring on a chip is calculated.

VLSI design

Today's VLSI (Very Large Scale Integrated) microchips contain some million transistors. The design cycle for these chips consists of different serial steps (e.g. system specification, functional design, logic design, circuit design, physical design). The *physical design* describes the transformation of a circuit description (which is the result of the preceding circuit design process) into the *layout* of a chip which is needed for the following fabrication step. The layout includes the geometrical description of the building blocks and the information for the routes of the interconnections between them (Fig. 1 shows a schematic representation of a layout). The main concern in the physical design of VLSI chips is to find a layout with minimal area, further the total wirelength has to be minimized. For some critical nets there are hard limitations for the maximal wirelength.

Due to its complexity, the physical design is normally divided into various sub-steps: The circuit has to be *partitioned* to get some (up to 50) macro cells which have to be placed on the chip (*floorplanning*). In the *routing phase* pins on the walls of these modules have to be connected. This is usually done in two steps: In the *global routing* the 'loose' routes are determined, while in the *detailed routing* the exact routes for the wires in each channel between two modules are computed. The last step in the physical design is the *compaction* of the layout, where it is compressed in all dimensions so that the total area is reduced.

In this project the floorplanning and the routing is done in one step using a *genetic algorithm*. The input to the floorplanning process are the rectangular building blocks (macro cells) and a number of interconnections between these blocks, described by a netlist. Most nets are two-terminal nets, i.e. they only connect two modules on the

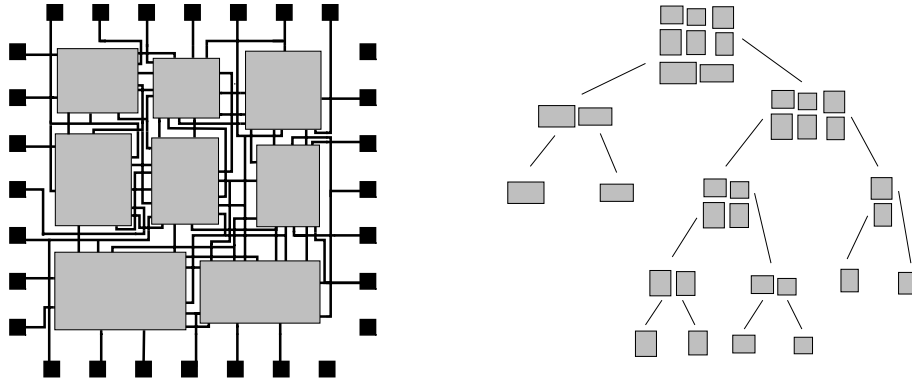


Figure 1: A layout of a chip (left) and its genotype representation (right)

chip, but there are also some nets (e.g. power or ground), which connect all blocks. At this stage (after the partitioning of the circuit) there are good estimates for the area of the macro cells, but their exact dimensions can still be varied over a wide range. The different aspect-ratios of these flexible cells are described by *shape-functions* (lists of feasible width/height-combinations).

Genetic Algorithms

Genetic algorithms are a very powerful technique to solve hard optimization problems. Like in the natural evolution there is the *population*, which is a set of *individuals*. In the application these are single solutions of the optimization problem. There are genetic operators, which change the population during a *generation* by creating new individuals and selecting some individuals for surviving to the next generation. After a certain time the *fitness* (solution quality) of the individuals converges to a global optimum.

Like in nature there is a *mutation* operator, which randomly changes an individual and so prevents the algorithm from sticking to a local optimum. The *crossover* operator creates an offspring out of two ‘parent’-individuals and thereby hereditating their good qualities to the new individual. At the end of every generation a fixed number of individuals is selected from the grown population due to their fitness to survive to the following generation. This is done by the *selection* operator which together with mutation and crossover simulates Darwin’s ‘survival of the fittest’ and yields a (nearly) optimal solution even in very complex solution spaces.

Combined floorplanning and routing

The main problem in the development of a genetic algorithm is to find good encodings for the individuals (*genotypes*) and the genetic operators. Here each individual describes a correct layout for the circuit of the VLSI chip. This layout is encoded by a binary tree (cf. Fig. 1). The leaves of this tree represent the building blocks of the layout and the inner nodes of the tree describe meta-blocks (partial layouts), which have the same

representation like the basic blocks. When building such a meta-block out of two blocks or meta-blocks, the total (detailed) routing inside this block can be done.

Building the tree in a bottom-up fashion, the root of this tree represents the layout for the complete chip. The mutation and crossover operators operate on the tree by exchanging leaves or changing the wiring-routes inside a meta-block (mutation) and combining subtrees of two individuals to a new solution (crossover). The convergence-ratio of the fitness (layout-area) is increased by using shape-functions also for the meta-blocks, so that every inner node of the tree – including the root – describes more than one layout alternative, provided that the blocks have different aspect ratios.

For real-world VLSI chips there are 10 to 50 building blocks and up to 500 nets (full custom design). The actual implementation includes a visualization of the layout by showing the blocks and the routing space. Future versions will include a visualization of the detailed routing, too, and an efficient parallelization of the genetic algorithm.

This project is part of the BMBF-project ‘HYBRID–Applications of Parallel Genetic Algorithms in Combinatorial Optimization’. It has started in April 1994 and will end in December 1996.

Contact address

Contact: Dipl.-Inform. Volker Schneck, Prof. Dr. Oliver Vornberger
Institution: University of Osnabrück
Department of Mathematics and Computer Science
Mail: D-49069 Osnabrück
Germany
Phone: +49 541 969 2397
Fax: +49 541 969 2770
Email: volker@informatik.uni-osnabrueck.de
WWW: <http://brahms.informatik.uni-osnabrueck.de/prakt/prakt.html>