UNIVERSITÄT OSNABRÜCK

VLSI Design with Genetic Algorithms

Volker Schnecke, Oliver Vornberger

University of Osnabrück
Dept. of Math./Computer Science
D-49069 Osnabrück
volker@informatik.uni-osnabrueck.de
http://brahms.informatik.uni-osnabrueck.de/prakt/prakt.html

Integrated circuits consist of many electronic components, built by layering several different materials in a well defined fashion on a silicon wafer. For this, the circuit description has to be transformed into a geometric description, the *layout* of the chip. This process is called the *physical design* of a chip. The physical design is the most time consuming step in the VLSI design cycle. Due to the complexity of the problem, it is usually broken into various sub-steps:

1. Because a VLSI chip may contain several million transistors, the circuit has to be *partitioned*, i.e. the components have to be grouped into blocks. This partitioning is done with respect to the size of the blocks, the number of blocks and the number of interconnections required between the blocks.
2. In the *floorplanning* phase, the building blocks are exactly positioned on a chip. In this phase the aspect ratio of a block can be varied within a prespecified range. The goal of this phase is to find a minimum area arrangement for the blocks that allows completion of interconnections between them.
3. The *routing phase* consists of two subphases. In the *global routing* the 'loose route' of a wire through different regions of the routing space (space on the chip which is not occupied by the building blocks) is computed. In the *detailed routing* phase the exact routes in the channels between the blocks are determined.
4. The last phase in the physical design of a VLSI-chip is the *compaction*, which is the task of compressing the layout in all directions such that the total area is reduced.

In this project the floorplanning and the routing is done in one step using a *genetic algorithm*. 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 hereditaring 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.

The main problem in the development of a genetic algorithm is to find good encodings for the individuals 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. The leaves of this tree represent the building blocks of the layout as they are determined by the partitioning phase. Such a building block is a rectangular block with a set of pins on the walls which have to be connected via a net with pins of other blocks of the circuit. If there exist different layout-alternatives for this block, the alternatives are described by a *shape-function*, which is a list of feasible height-/width-combinations. 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, the total (detailed) routing inside this block can be done (c. f. Fig. 1).

Building the tree in a bottom-up fashion, the root of this tree represents the layout for a 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 (like shown in Fig. 2), and an efficient parallelization of the genetic algorithm.

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

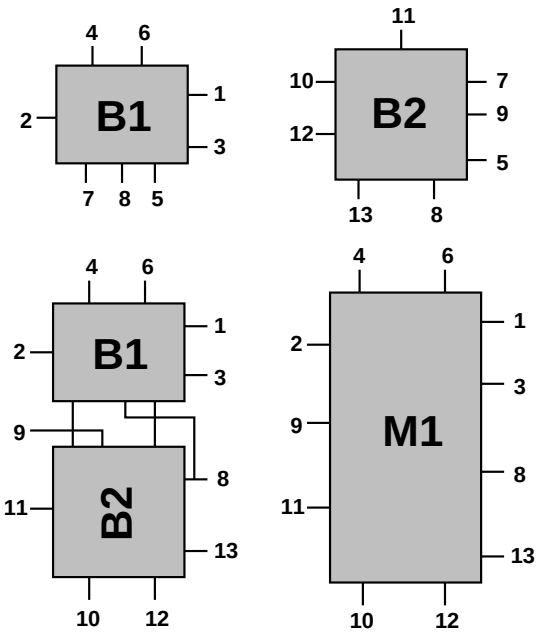

Fig. 1: Combining two blocks to a meta-block

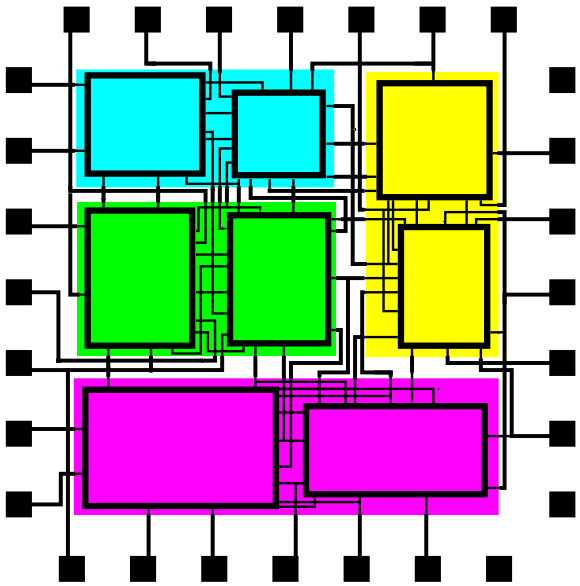

Fig. 2: The layout of a VLSI-chip