Assignment Space Exploration Approach to Concurrent Data-Path / Floorplan Synthesis

Koji Ohashi

School of Information Science,
Japan Advanced Institute of Science and Technology

February 15, 2000

Keywords: High-Level Synthesis, Scheduling, Assignment, Layout, Branch-And-Bound Search.

High level synthesis is the task to synthesize the RT level structure in the structural domain from the algorithm in the behavioral domain. Major criteria for discussing design optimality are the speed and area of VLSIs. In addition, transmission delay by wires, power consumption, testability, etc. are also becoming important factors in the high–level synthesis. As the geometrical design rules of VLSIs becomes finer into the order of deep sub–micron, the impact of wires to VLSI performance becomes larger relatively to the other components. To discuss the total computation time of a given application algorithm, not only the number of control steps but also the clock period should be taken into account of. Especially in deep sub–micron era, signal transmission delay induced by wires will become a dominant factor for deciding a clock period. On the other hand, the power(dynamic power) consumption in MOS LSIs depends mainly on the supply voltage(logical voltage swing), the load capacitance of each gate and the number of signal transitions. If we can navigate the design so that as many as possible operations and data are assigned and geometrically located closely, and the number of distant data transfer is minimized, such design will result in the suppression of signal transitions which need to drive large capacitive load.

Since the layout(floorplanning) problem needs connectivity information between modules, several approaches in which floorplanning is incorporated into high–level synthesis at binding phase have been proposed. However, these approaches assume a scheduled DFG as their input. In fact, the high–level synthesis contains several interdependent sub–problems, most of which are known to be $\mathcal{NP}$–hard. Most of synthesis systems solve these sub–problems sequentially, and the scheduling is the first one to be solved in many systems. As a result, the optimality of final performances, especially those relevant

to module connectivity and layout(floorplan), may be missing in such scheduling–first strategy.

In this research, an assignment–driven approach to the data–path synthesis incorporated with one–dimensional floorplanning is proposed. In our approach, scheduling and one–dimensional floorplanning, both of which are driven by iteratively generated functional unit and register assignment(binding), are performed fully concurrently. Branch–and–bound like assignment space exploration is adopted for generating assignments, and system performances evaluated from both assignment–driven scheduling and floorplanning are used for pruning.

As for the one–dimensional floorplanning, the minimization of total wire length is known to be $\mathcal{NP}$–hard, and various iteration– based methods and constructive methods have been proposed. Considering that floorplanning will be evaluated repeatedly within the assignment space exploration process, local refinement approach is adopted for floorplanning. When the branching proceeds one step(that is, one object is newly assigned to one module, say r), RTL architectural topology maintained in the synthesis procedure changes in the following patterns, (a)module r is newly appended into the topology, (b)some interconnections between module r to other module are newly appended, and/or (c)the number of data transfer on some nets containing module r is increased by one. Based on this observation, the local refinement process is constructed as follows. First the initial floorplan is set to empty. To obtain a floorplan for a certain partial assignment solution, the floorplan for its parent partial solution is used as its initial solution, and refinement is achieved by replacing each pair of modules, between which a new connection request is appended, to their best positions while the relative locations(ordering) of the other modules are maintained.

Finally, the proposed method is implemented on a workstation using C language, and is applied to some benchmarks. As a result, we confirmed that our method can handle the tradeoff between several VLSI performances, such as area, execution time and power, depending on the optimization priority.

RTL–level floorplanning for tighter estimation of layout relevant VLSI performances and heuristics for the exploration of vast assignment space are remained as future problems.