論文の内容の要旨

論文題目　Communication-Oriented Hardware Synthesis and Its Formal Verification
（通信指向ハードウェア合成とその形式的検証）

氏名　高 尚华

As VLSI process technology advances, the gap between gate delay and (global) interconnect delay is getting wider and wider. This makes interconnect delays (especially global interconnect delays) one of the most important factors that can affect performance. Under such situation, many state-of-the-art researches in high level synthesis try to consider the effect of interconnect delays. These researches indeed achieve better performance compared with traditional ones that ignore interconnect delays. However, when applications contain large loops (e.g., digital signal processing algorithms), there is still much room to improve performance, since usually, such applications contain computations that can be executed concurrently, they offer the possibilities of exploiting the parallelism.

In this work, we, for the first time, propose the idea to utilize pipelining techniques and take interconnect delays into account together into high level synthesis so as to improve its quality. This is not an easy task, since interconnect delays are unknown at the stage of high level synthesis. Usually they are not understood until after placement and routing.

In order to facilitate the estimation of interconnect delays, we propose a novel architecture, called regular distributed-register architecture for dynamic routing (RDR-dr). It divides a chip into a two dimensional array of islands, and the interconnect delay can be roughly estimated from the location of related islands.

Then, with the estimated interconnect delay information, we try to solve the problem of integrating both the effect of interconnect delay and pipelining techniques into high level synthesis by using exact solution methods such as Integer Linear Programming (ILP) and Boolean Satisfiability (SAT). We formulate the problem in the form of ILP/SAT, and then utilize commercial existing ILP/SAT solvers to find the solutions. As a result, for small examples, the exact methods can solve. But for relatively large examples, the exact methods can not solve within eighteen hours.

In order to improve the efficiency, we then propose a heuristic pipeline scheduling algorithm for array based architectures considering interconnect delays, and develop corresponding

interconnect-aware pipeline synthesis system. The proposed method has the following two characteristics: 1) it separates the consideration of interconnect delay from computation delay, and allows concurrent data transfer and computation; 2) it belongs to modulo scheduling framework, in the sense that all iterations have identical schedules, and are initiated periodically. We evaluate the proposed heuristic method from different points of view and the experimental results show that: (a) our proposed interconnect-aware pipeline synthesis outperforms the existing works in high level synthesis which consider the effect of interconnect delays or utilize pipelining techniques; (b) the quality of our heuristic method is almost the same as that of exact methods; (c) our proposed algorithm is able to solve large practical problems, and for an example with 883 nodes, it takes only 2 seconds.

Since the pipeline synthesis with interconnect delay considered needs complicated algorithms, the synthesizer may contain bugs. To check the correctness of the synthesizer, we propose an approach which applies a combination of induction method and symbolic simulation technique. Based on this approach, we develop a prototype equivalence checker. The inputs to the checker are a loop in high level description and a pipelined loop. In the experiment, we verify several loop examples before and after pipeline synthesis including ones that bugs are intentionally inserted. The results show that: 1) for the examples in which we intentionally insert bugs, our prototype equivalence checker detects the not-equivalence in very short runtime; 2) for the examples obtained from automatic interconnect-aware pipeline synthesis, our developed prototype equivalence checker judges that the loops before and after pipeline synthesis are equivalent. This result demonstrates that our developed interconnect-aware pipeline synthesizer works correctly.

As a conclusion, an efficient interconnect-aware pipeline synthesizer is developed. At the same time, its correctness is confirmed. If introducing the synthesizer with existing design flow which starts design from RTL level, one can get a higher performance chip in shorter turn-around time. We are sure that the tool is more useful in future as process technology advances.