Semiannual Report: July 1991 - December 1991

Asynchronous Design for Parallel Processing Architectures

Contract No.: N00014-89-J-3036
Principal Investigator: Teresa H.-Y. Meng
CIS 132, Stanford University, Stanford, CA. 94305
Phone Number: (415) 725-3636
E-Mail Address: meng@tilden.stanford.edu

The objective of this research is to provide a design methodology for connecting heterogeneous hardware modules that have inherently different functional and timing behavior. With the constraints dictated by the system-level interaction, we need to adopt a modular design approach without compromising the global performance. The main task of this effort will be the development of the theory for optimal interface circuit synthesis from a high-level specification, with emphasis on testability and performance.

In the past six months, we have concentrated on the gate-level synthesis of asynchronous control circuits and introducing timing information into the synthesis process to increase circuit performance.

Gate-Level Synthesis of Speed-Independent Circuits

In our previous work we have addressed the problem of the testability for two kinds of asynchronous design: speed-independent circuits [1] and self "timed" circuits [2]. The testability results are based on the fact that asynchronous circuits are usually implemented by a collection of gates (the gate-level implementation). The next goal is to provide a synthesis tool that will generate a gate-level representation of a specified asynchronous circuit which is suitable for a standard-cell or gate-array implementation.

It has been shown that arbitrary delay elements can be added to a hazardous timed circuit to create a hazard-free design. We investigate the synthesis of gate-level hazard-free circuits *without* added delay elements, because such elements are bound to degrade circuit performance. We have developed heuristic algorithms to synthesize gate-level speed-independent circuits from a state-transition-diagram specification. The synthesis algorithms assume a realistic set of gates as building blocks, namely, AND gates, OR gates, and C-elements, with inverters attached to inputs if desired. The synthesized circuits are guaranteed to be hazard-free under all possible relative delays among the gates [3]. We are currently incorporating these algorithms into a synthesis system. When combined with translators of high-level language descriptions, such as communicating sequential processes (CSP) or signal transition graphs (STG), we can use this synthesis system to synthesize gate-level implementations from existing high-level languages.

One important feature of our algorithms is that they incorporate speed-independence-preserving logic optimizations. These optimizations are important for two reasons. First, we have found that significant savings in area and delay can be achieved in speed-independent designs, which in all existing synthesis systems has not been considered. Secondly, the speed-independence-preserving optimizations are of theoretical importance. While logic optimizations-transformations are well known for synchronous circuits and for some restricted class of asynchronous circuits, logic optimizations for speed-independent circuits have yet to be exploited.

Synthesis of Timed Asynchronous Circuits

One of the primary arguments against the use of asynchronous circuits is the belief that their advantages are paid for in area or delay. In this work we show that we can reduce the circuit area

and *increase* circuit performance by incorporating timing information into the synthesis process. Contrary to other approaches to timed circuit synthesis, we use timing information not to eliminate hazards, as hazards do not exist as a result of correct gate-level synthesis, but to optimize the circuit for better performance. Our synthesis procedure significantly reduce the circuit area by using conservative delay assumptions to remove redundant circuitry fron their speed-independent counterparts. The resulting implementation, while not speed-independent, is guaranteed to be hazard-free under the given delay assumptions.

The synthesis process starts with a circuit specification at the STG level, with each delay assumption given in a range (minimal delay and maximal delay of a particular transition). These delay assumptions can be obtained from commercially available data sheets, if the transition is for an external signal, or derived from a reasonable timing model, if the transition is for an internal signal which is to be synthesized. Not all transitions need to be provided with a delay assumption, as we allow gate delay of zero and infinity as part of the specification. Of course, the more timing information there is, the better optimized the circuit can be. The caution is that all these delay assumptions will have to be *satisfied* in the final circuit for it to function correctly.

Our synthesis algorithm takes the STG specification and translates it to a set of event-rules. These event-rules determine the precedence, or the causal relations, among the signal transitions. By tracing the STG, we can delete, rather conservatively, redundant conditions for each signal transition implied by the delay assumptions. As a result, the number of event-rules is reduced and the circuit is simplified. We have proved that the tracing algorithm will terminate, and that it is correct; i.e. the algorithm will not delete conditions that will be later found to be necessary to insure hazard-freedom of the circuit. In other words, our algorithm is a sufficient technique, but may not be necessary. Though this algorithm is conservative, we have tested it on the synthesis of various interface circuits. Most of the circuits that we have synthesized with only timing assumptions on external signals show a 50% reduction in circuit complexity.

Future research for the next two quarters:

Currently, we are considering developing our system on top of SIS, the Berkeley synthesis system for sequential circuits. Using SIS, we will be able to spring-board off many existing algorithms and will have common grounds for comparisons with the timed circuit synthesis results as well. On the synthesis side, we need to formally prove that our gate-level algorithms will generate correctly working speed-independent circuits for *all* legal specifications. We would also like to formalize our results on testability, and combine the testability knowledge with the synthesis algorithms to provide synthesis for testability options.

For the synthesis of timed circuits, we plan to determine the necessary conditions for our tracing algorithm. We will also apply our techniques to a larger class of circuits, for example the circuits with conditionals. The final goal is to combine the two synthesis strategies, gate-level implementation of timed circuits, into one homogeneous design tool.

References:

[1]. Peter A. Beerel and Teresa H.-Y. Meng, "Semi-Modularity and Testability of Speed-Independent Circuits", submitted to *Integration, the VLSI Journal*, November 1991.

[2]. Peter A. Beerel and Teresa H.-Y. Meng, "Testability of Asynchronous Self-Timed Control Circuits with Delay Assumptions", *Proc. 28th ACM/IEEE Design Automation Conference*, pp. 446-451, June, 1991.

[3]. Peter A. Beerel and Teresa H.-Y. Meng, "Gate-Level Synthesis of Speed-Independent Control Circuits", to appear in *1992 ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems.*, March 1992.