Translating Concurrent Communicating Programs into Asynchronous Circuits

Erik Brunvand

September, 1991

CMU-CS-91-198

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Computer Science.

This research was sponsored in part by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597 and in part by an IBM Graduate Fellowship.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of IBM Corporation or the U.S. government.

Keywords: asynchronous circuits, self-timed circuits, occam, silicon compilers

Abstract

As VLSI technology improves, the number of devices that can be built on a chip, and the speed of those devices continue to increase. These improvements allow much more complicated systems to be considered than were possible a short time ago. Along with these improvements, however, come many challenges directly associated with the speed and scale of the new circuits. This thesis presents a method for taming the complexity of large and fast VLSI systems.

As chips get larger, and delays in signal propagation even inside a single chip become more significant, systems that are designed around a global synchronizing signal such as a clock become more difficult to design. One alternative is to design the system hierarchically as a set of subsystems each operating at its own rate and cooperating through communication. These subsystems can be built using *asynchronous* or *self-timed* circuit techniques which allow the circuits to run at their own speed without synchronizing to a global clock signal. The act of communicating synchronizes the processes involved in the communication and allows data to be shared between processes.

Another problem of large systems is related to the very complexity of the system. One method for taming this complexity is to use automatic methods for generating circuits from behavioral descriptions. Such a system is usually called a *silicon compiler*. This allows the designer to abstract away details of the low-level circuits and think of system behavior in terms of high level programs. Because the generated circuits faithfully mimic the behavior of the program, the resulting circuits are correct by construction. In order to design efficient systems in this way there must be a way for the programmer to reason about the resulting circuit based on the program text. The translation process must be sufficiently transparent to give the programmer some idea of how different program alternatives will affect the compiled circuit.

Combining these ideas, I present a method for designing a VLSI system as a concurrent program written in a subset of OCCAM and automatically translating that system description into an asynchronous circuit. The translation procedure is syntax-directed, and the resulting circuits are improved using correctness-preserving circuit-to-circuit transformations. A compiler has been constructed that automatically performs the translation and transformation.