A Fast Petri Net Based Programmable Controller:  
From the Specification to a Self-Timed Implementation

Helmut Baur, Slavek Bulach
Department of Microelectronics
University of Ulm, Germany

Abstract

First published hardware implementations of Petri Net based controller circuits with flip-flops substituting each individual place and transition were developed by Patil in 1972. Later, other programmable approaches and implementations of Petri Net control algorithms on general purpose processors (µP, µC, PLC) have been reported. A novel PN-based dedicated programmable controller architecture is presented here whose asynchronous realization offers fast response times to external events and power efficiency for applications with variable signal arrival rates. This approach is based on the local execution of the enabled transitions in the net. The net structure is stored in memory as transition blocks and the firing conditions are tested by the controller. The asynchronous realization of the event-driven controller architecture provides implicit idle and wake-up modes.

The main theme of this talk is to present the basic concepts of the PNDU (Petri Net Decision Unit) and the development of its self-timed realization. Starting from the net structures supporting concurrent processes and the inclusion of I/O signals in the firing conditions, an encoding scheme for the storage of the net structure in EPROM memory is derived. The cyclic net execution algorithm consists of transition enabling checks and firings. New output signals are set after all enabled transitions of the current marking have been fired. Thereby, the internal sequential execution of the net appears concurrent to the environment. This net execution algorithm is mapped onto a system architecture that is optimized for reactive control tasks.

The objective of the first self-timed prototype presented at ACID’00 was to demonstrate the methodology of transforming an event-driven architecture into a self-timed implementation using VHDL and commercial circuit design tools such as Synopsys and Cadence. This gate-level PNDU realization exhibited high control cost. Additional work shows that by increasing module concurrency and decentralization of control functionality a significant improvement of the average processing speed is achieved. The talk will summarize lessons learned in designing self-timed circuits for programmable control applications.