Specification and Analysis of Self-Timed Circuits

M.A. KISHINEVSKY
Technical University of Denmark, Lyngby, Denmark
e-mail:mik@id.dth.dk

A.YU. KONDRAITYEV
The University of Aizu, Japan
e-mail:kondray@rsc.u-aizu.ac.jp

A.R. TAUBIN
The University of Aizu, Japan
e-mail:taubin@rsc.u-aizu.ac.jp

Received April 25, 1992; Revised November 3, 1992.

Abstract. The problems of self-timed behavior specification and verification are considered on the basis of an event model—Change Diagram (CD). The descriptive power of a CD model is demonstrated by comparing the CD with Signal Transition Graphs (STG). CD differs from STG by two types of causal relations (AND and OR) between events (in STG only AND-relation is presented). CD verification is shown to be reducible to an analysis of precedence and concurrency properties for events. These properties are hard to analyze directly by a cyclic CD. We suggest that the cyclic description be replaced by an equivalent acyclic one (called an unfolding) in order to solve the analysis problem. The notion of CD correctness is introduced, and the necessity and sufficiency of this notion for the implementation to be in self-timed class are shown. The polynomial algorithms for CD correctness verification are considered.

1. Introduction

Specification and analysis of self-timed circuits (STC) behavior constitute the intersection point of the interests of hardware designers and researchers who develop formal models of concurrency.

Among the models that have been sufficiently well studied, one may refer to the so-called global models which employ partially ordered sets of complete states of the circuit (i.e., values of the signals involved). Transition diagrams (TD) for Muller model [1] as well as trees of markings for Petri net [2] are examples of this kind. They provide rather comprehensive descriptions that contain the full information about the feasible states of the system.

With the growing dimensions of the circuit, however, the completeness of the description becomes a burden rather than an advantage as the description complexity (number of states) increases exponentially.

To reduce the representation complexity the event-based models were introduced: signal transition graphs (STG) [3], Petri net graphs [2], traces [4], and event-structures [5]. They specify the process by the local transitions of the components of a state. Usually these models are used only to specify the circuit behavior, while the problems of analysis and synthesis are quite difficult and can be solved only by the relevant global state description.

In recent CAD systems for STC [6], [7] their authors seek to overcome these challenges of analysis through the introduction of their own criteria of the circuit correctness. These attempts, however, ignore the important problem of the criteria consistency with Muller's fundamental principles. As a result, the class of implementable processes becomes artificially constrained, as some serviceable circuits are rejected as incorrect (as for semimodular takeover circuits considered in [3], [8], [9]).

The only possibility, we believe, to avoid such pitfalls is to impose a requirement to consider as correct all the event-based specifications which are equivalent to semimodular TD, so that the class of serviceable circuits is not restricted unnecessarily.

To provide an adequate event representation for the processes in STC, change diagrams (CD) were developed [10], [11]. This model allows the analysis
to be performed directly by the structure of a corresponding graph (without using global state models).

In this article we have the following major objectives:

1. We try to present a succinct review of the major concepts and results relevant to the STC analysis in terms of TD.
2. We discuss an outline of a CD event model that can be used to represent semimodular circuits that are not distributive.
3. We plan to consider the main ideas, results and techniques of STC analysis based on CD representation. Some aspects of using CD for STC design has already been presented in [11]–[14].

Note that in line with the Muller’s classical theory we consider here, only autonomous circuits (i.e., circuits without external inputs or with fixed signal values on external inputs). This is quite reasonable because any control circuit can be implemented within a class of speed-independent circuits as an autonomous one with control units to be inserted in control circuit wire breaks (see [10] for greater details).

Direct specification of the “open” circuits would demand the introduction of some means of external nondeterminism representation (e.g., free choice in STG model [3]). This, however, would increase the complexity of analysis and synthesis significantly (these problems are beyond the scope of this article).

2. Muller Model, Main Definitions and Results

In the Muller model [1], [10], every circuit element is represented by a combination of a function generator, which implements the corresponding logic function instantly, and an undetermined time delay at the component output. All the delays of physical processes, that are involved in signal processing and transmission within an element and along the connecting wire prior to its fork, are taken to be reduced to the output delay. The skew of the wire delays after the fork is considered to be negligible (in practice the latter requires from the skew to be less than minimal gate delay). A delay induced by a wire may be allowed for, if need be, by inserting an auxiliary component into the wire break—a buffer.

Note that down scaling of electronic components results in further growth of wire delays. Today it has become usual to say that a circuit is a set of wires interconnected by gates. In this context, attention is focused on the circuits whose behavior is independent of the wire delays. Obviously it would be excessively optimistic to hope that one might design a circuit which would be insensitive to the delays in all its wires, and in this respect one could only be thinking about securing the validity of some theoretical hypothesis through technological means. Discussions along these lines usually lead to demands to make all the after-fork wire delays more or less identical (isochronous fork for example [6]), and after this the circuits could be declared delay-independent.

For us it seems more natural to concentrate on the hypotheses about the wire and component delays and to keep squarely within the framework of the Muller theory, rather than pretend that we are dealing with a somewhat wider class of delay-insensitive circuits.

Let us now recall several major concepts of the Muller theory.

A circuit \( S \) is a set of gates \( \{z_1, \ldots, z_m\} \) with each gate input connected to strictly one gate output and with no two outputs interconnected between themselves.

A state of a circuit at a given moment of time is a set of the signal values on the outputs of gates. Behavior of the \( i \)-th gate can be described by the Boolean equation

\[
z_i' = f_i(z_1, \ldots, z_i, \ldots, z_m),
\]

where \( z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_m \) are the values in the inputs of the \( i \)-th gate, \( z_i \) is the signal value of the gate output, \( z_i' \) is its next value to replace the previous one, and \( f_i(z_1, \ldots, z_m) \) is an inherent function of the \( i \)-th gate.

A Muller model of circuit \( S \) (of \( n \) gates) is a set of \( n \) simultaneous equations of type (1) which describes the behavior of all the circuit gates.

We would say that gate \( z_i \) is excited, if \( z_i' \neq z_i \), otherwise it is stable. Circuit goes from one state to another by switching the excited elements (excited elements are usually marked by a star (*) in the Boolean code of a state).

Formally circuit behavior can be presented by a transition diagram (TD).

**Definition 1.** An \( m \)-variable TD is an oriented graph such that its vertices are labelled by Boolean \( m \)-tuples (diagram states), its edges correspond to the NEXT relation (\( \rightarrow \)) between the adjacent vertices, and that two following conditions are met:

1. If state \( u = (u_1, \ldots, u^m) \) directly precedes state \( w = (w_1, \ldots, w^m) \), then every digit \( u_i' \), where \( u_i' \) differs from \( w_i \) is considered to be excited and hence marked by * in \( u \);