Structural Analysis of Signal Transition Graphs

F. García-Vallés and J.M. Colom
Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza

Abstract

Signal Transition Graphs (STGs) are net systems with an interpretation, modelling the behaviour of asynchronous digital circuits by sequences of changes of the circuit signals. Although they have became very popular, only restricted syntactical subclasses (free choice nets) are considered in general. Moreover, the analysis techniques applied to these models are essentially based on the state space obtained from the STG, giving rise to the classical state space explosion problem. In this paper we propose a model based on the STGs, that is a net system without interpretation allowing to represent the same circuit behaviour as a classical STG. This new model consists of the underlying net system of the STG plus a set of places representing the values of the circuit signals. Considering this model, we show how general structural analysis techniques can be applied to verify properties of the system, as consistency, CSC or USC. These analysis techniques avoid the construction of the state space. Moreover, we show that with this kind of model no restrictions on the kind of Petri nets to be used are needed.

Topics: Modelling asynchronous circuit behaviour, Structural methods of verification, Application of Petri Nets in asynchronous circuits, Signal Transition Graphs.

1 Introduction

State based models have been traditionally used in the description of digital systems. In the last years, great attention has been devoted to event based models, because of its advantages to model asynchronous systems. In particular, a Petri net based model, the Signal Transition Graph, has been widely used [2, 17]. This model is able to represent in a very compact way the behaviour of almost any asynchronous system, excepting those that exhibit *or causality* [21].

In spite of being a model able to express concurrency, definitions of the basic properties to study these systems, and even its analysis, have traditionally been carried out at the state graph level. Therefore the STG model becomes just an element of description of the system, useless in the analysis or synthesis processes. An alternative way to perform these tasks could make use of the underlying structure of the STG, taking into account the great amount of results in Petri nets theory already known. Pastor et alt. [15] follow this approach by developing structural algorithms to solve analysis and synthesis problems of asynchronous circuits and show its adequacy for some subclasses. However, the general basis to carry out this methodology is not well established. The problem arises from the fact that STGs are interpreted net models. Therefore, many questions that can be posed on the STGs cannot be solved only using the underlying net system (ad hoc algorithms dealing with the signals are needed). We propose, as solution, to represent in net terms all information concerning signals. In essence, the idea is to represent the values of the signals by means of places and the signal changes by means of arcs connecting these places with the transition producing the change. At this point all the questions concerning STGs can be expressed in terms of a property of this new net, and we can use indigenous techniques (not necessarily based on the net space) of Petri nets to answer the questions.

Another important question in STGs concerns the constraints that are usually imposed to the underlying net model: free choice nets, binarity or liveness are some of the more common. Most of them does not seem to be related with the systems under study. In this sense, Yakovlev [20] *extends* the more usual STG model showing that several restrictions are not necessary, and more general models are useful in some cases. Moreover, some usual definitions mix non-related concepts from the asynchronous systems and Petri nets fields. This situation has left to multiplicity of definitions and, in general, to a lack of clarity. Therefore it is necessary to clarify basic concepts, review the definitions removing useless hypothesis, and to use Petri net terminology if already exists. In this paper we address this clarification problem for the case of the consistency property of STGs.

Structure based methods have been widely applied in Petri net analysis. However, only recently Pastor has introduced its application to STGs. The STGs he works with are limited to the free choice subclass, and then he uses results from free choice theory. But it is possible to apply more general results of structure analysis of Petri nets, opening a way to deal with non free choice STGs. In this sense, we propose the use of linear algebraic techniques for the analysis, using two representative properties of STGs, USC and CSC, to illustrate the approach. The use of these methods allows to deal with generalized Petri Nets (therefore the full descriptive power of PNs can be used without any limitations because the analysis algorithms) with low practical computational complexities (using, for example, standard algorithms to solve Linear Programming Problems). Unfortunately, these structural analysis methods only give necessary or sufficient, but in general no both, conditions on the verification of a given property. The problem arises from the linear description of the reachability set, the net state equation, that has non-reachable solutions.

The paper is organized as follows. Section 2 presents definitions and notation concerning Petri nets and the state space model of the system. Consistency property is introduced in section 3, and a new characterization at the STG level based on the implicitness of certain places is presented. A sufficient condition to verify the consistency property on a standard analysis is also presented. Sections 4 and 5 deal with the unique and the complete sate coding properties respectively, developing a structural approach to their analysis.

2 Basic Definitions

2.1 Nets and Net Systems

We assume the reader has a basic knowledge of Place/Transition net models (see, for instance, [14]), elementary graph theory (see, for instance, [4]). However, in this section we recall the basic concepts and notation that are going to be used.

Nets. $\mathcal{N} = \langle P, T, F, W \rangle$ is a *net* if P and T are finite, no empty and disjoint sets of *places* ($|P| = n$) and *transitions* ($|T| = m$), respectively, $F \subseteq (P \times T) \cup (T \times P)$ and $W : F \to \mathbb{N}$. We require every net to be *connected*.

Let $\mathcal{N} = \langle P, T, F, W \rangle$ be a net. *Pre-* and *post-sets* of elements in $P \bigcup T$ are denoted by the dot-notation: ${}^\bullet x = \{y | (y, x) \in F\}$ and $x^\bullet = \{y | (x, y) \in F\}$. This notion is extended to sets of elements; ${}^\bullet X$ is the union of the pre-sets of elements of X, $X^\bullet$ is the union of the post-sets of elements of X.

Let P and T be arbitrarily but fixed ordered. The *Pre-incidence matrix* $\mathbf{Pre} : P \times T \to \mathbb{N}$ of $\mathcal{N}$ is defined by $\mathbf{Pre}[p, t] =$ if $(p, t) \in F$ then $W((p, t))$ else 0. The *Post-incidence matrix* $\mathbf{Post} : P \times T \to \mathbb{N}$ of $\mathcal{N}$ is defined by $\mathbf{Post}[p, t] =$ if $(t, p) \in F$ then $W((t, p))$ else 0. The *incidence matrix* $\mathbf{C} : P \times T \to \mathbb{Z}$ of $\mathcal{N}$ is defined by $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$.

Let $\mathbf{A}$ be anyone of the three matrices $\mathbf{C}$, $\mathbf{Pre}$ or $\mathbf{Post}$. $\mathbf{A}[\Pi, \Theta]$ denotes the submatrix of $\mathbf{A}$ corresponding to rows of places in $\Pi \subseteq P$ and columns of transitions in $\Theta \subseteq T$.

Net Systems. A function $\mathbf{m} : P \to \mathbb{N}$ is called *marking*. Markings are represented in vector form. A *Net system* is a pair $\langle \mathcal{N}, \mathbf{m_0} \rangle$ where $\mathcal{N}$ is a P/T net and $\mathbf{m_0}$ is the *initial marking*. A transition $t \in T$ is *enabled* at $\mathbf{m}$ if $\mathbf{m} \geq \mathbf{Pre}[P, t]$. If t is enabled at $\mathbf{m}$, then t may *fire* yielding a new marking $\mathbf{m}'$ given by $\mathbf{m}' = \mathbf{m} + \mathbf{C}[P, t]$. The fact that the firing of t changes the marking $\mathbf{m}$ into the marking $\mathbf{m}'$ is denoted by $\mathbf{m}[t\rangle\mathbf{m}'$. A *firing sequence* from $\mathbf{m}$ is a sequence $\sigma = t_1 t_2 \ldots t_r \in T^*$ such that $\mathbf{m}[t_1\rangle\mathbf{m}_1[t_2\rangle\mathbf{m}_2 \ldots \mathbf{m}_{r-1}[t_r\rangle\mathbf{m}_r$. $\mathbf{m}[t\rangle$ ($\mathbf{m}[\sigma\rangle$) denotes the marking into which $\mathbf{m}$ is transformed by the firing of t (σ). The set of firing sequences (the *language*) is denoted by $\mathrm{L}(\mathcal{N}, \mathbf{m_0})$ and the *reachability set* of $(\mathcal{N}, \mathbf{m_0})$ is $\mathrm{RS}(\mathcal{N}, \mathbf{m_0}) = \{\mathbf{m_0}[\sigma\rangle \mid \sigma \in \mathrm{L}(\mathcal{N}, \mathbf{m_0})\}$.

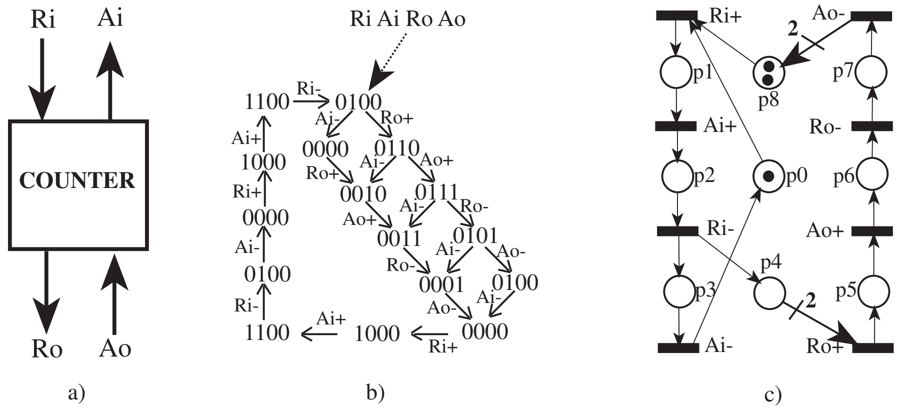

Figure 1: a) A module 2 counter with input and output signals following a 4-phase handshake protocol. b) Representation of the behaviour by a STD model (removing the arc labels) and an ALTS model (removing the binary vectors). c) The Signal Transition Graph.

A net system is *bounded* when there exists a bound $b = \max_{p \in P}\{\max_{\mathbf{m} \in \mathrm{RS}(\mathcal{N},\mathbf{m_0})}\{\mathbf{m}[p]\}\}$. A net system is said to be *safe* (or *binary*) if the bound b is equal to one.

2.2 State Transition Diagram

An usual way to specify the behaviour of a digital system is by means of a state based model called *State Transition Diagram* (*STD*) [8, 10]. A state in the model is defined by an assignment of values to the system signals. A transition between states is a change of a signal value. In the sequel we denote system signals by letters a, b, c, ...

Definition 1 (State Transition Diagram [10]) *A State Transition Diagram (STD) is a 4-tuple $\langle V, E, A, \lambda\rangle$, where V is a set of states, $E \subseteq V \times V$ is a set of transitions, A is a set of system signals, and $\lambda : V \to B^{|A|}$ ($B = \{0, 1\}$) is a function labelling each state with a binary vector of signal values. Moreover, $A = X \cup Z$, where X and Z are disjoint sets of* input *and* output *signals, respectively.*

The figure 1b without the labels in the arcs is a STD corresponding to the behaviour of the module 2 counter depicted in figure 1a. The system communicates with its environment by means of a 4-phase handshake protocol, producing a 4-phase output per two 4-phase inputs. Each state is represented (function λ) by a binary vector of values of the system signals (R_i, A_i, R_o, A_o).

In order to represent the behaviour of a digital system by means of a STD model, an additional constraint must be added: for all transition between two states, one and only one signal changes its value. STD models satisfying this property are said to be *consistent*.

Definition 2 (Consistent State Transition Diagram) *A State Transition Diagram, $\langle V, E, A, \lambda\rangle$ is* consistent *if and and only if for all $(v, v') \in E$, the Hamming distance between the two binary vectors $\lambda(v)$ and $\lambda(v')$ is equal to one.*

Sometimes it is useful to represent the behaviour of the systems by means of the sequence of signal changes instead of the sequence of states. Associated with a consistent STD, it is possible to build a model where signal changes are explicitly shown. The following definitions show how this new model is obtained from the STD.

Definition 3 (Arc-labelled Transition System) *An Arc-labelled Transition System (ALTS) is a 4-tuple $\langle V, E, A, \delta\rangle$ where V is a set of states, $E \subseteq V \times V$ is a set of transitions, A is a set of signals, and $\delta : E \to A \times \{+, -\}$ is a function labelling each arc with a signal change.*

Definition 4 (ALTS associated with a consistent STD) *Let $\langle V, E, A, \lambda\rangle$ be a consistent STD, the Arc-labelled Transition System associated with it is the 4-tuple $\langle V, E, A, \delta\rangle$ where function δ is defined as follows: Let $e = (v, v') \in E$ be a transition and $a \in A$ the signal on which a change has been produced (i.e. $\lambda(v)[a] \oplus \lambda(v')[a] = 1$), $\delta(e) =$* **if** $\lambda(v)[a] = 0$ **then** a^+ **else** a^-.

In the presented example the figure 1b, removing the state labels, is the corresponding ALTS. Note that, if the STD was not consistent and we would want to obtain the ALTS associated with it, at least one arc should have more than one label.

2.3 Signal Transition Graphs

Another way to specify the behaviour of digital systems is based on events instead of states. Signal Transition Graphs, (STGs) [2, 17], are a kind of these models. A STG is a Petri Net whose transitions carry labels of the form a^+, a^-, where a is a circuit signal. The occurrence of a transition with label a^+ *raises* a, i.e. sets its value to 1, while the occurrence of a transition with label a^- *falls* a, i.e., sets its value to 0. Given a signal a, we will denote by a^* either a^+ or a^-.

Definition 5 (Signal Transition Graph) *A Signal Transition Graph $\mathcal{N}^A$ is a tuple $\mathcal{N}^A = \langle \mathcal{S}, A, \Lambda\rangle$, where $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0}\rangle$ is a net system, $\mathcal{N} = \langle P, T, F, W\rangle$ is a net such that $\nexists\, t_1, t_2 \in T \mid {}^\bullet t_1 = {}^\bullet t_2$ and $t_1{}^\bullet = t_2{}^\bullet$, $\mathbf{m_0}$ is an initial marking of $\mathcal{N}$, A is a set of signals, and $\Lambda : T \to A \times \{+, -\}$ is a function that assigns a signal change to each transition of the net. Moreover, $A = X \cup Z$, where X and Z are disjoint sets of* input *and* output *signals, respectively.*

In this definition we exclude nets with pairs of transitions with the same pre and post sets, because there would be some problems in associating an ALTS to a net in that case (see the definition below).

An STG describing the counter example is shown in the figure 1c. Signal changes label the transitions, and the behaviour of the system in terms of event sequences is defined by the firing sequences of the underlying net system, considering labels instead of transition names.

A Signal Transition Graph is called *binary* if its underlying net system is binary. The example in the figure 1c is not binary; several reachable markings, as the initial one, have places with several tokens.

From a Signal Transition Graph is straightforward to obtain a state based model of the behaviour of the modelled system. This state based model is the reachability graph of the underlying net system of the STG. The nodes of this graph are the reachable markings and there exists an arc from a marking $\mathbf{m}$ to a marking $\mathbf{m}'$ if $\mathbf{m}'$ is reachable from $\mathbf{m}$ firing a single transition. Normally, the arcs of the reachability graph are labelled with the name of the fired transition. If the arcs are labelled with the signal change produced by firing the corresponding transition ($\Lambda(t)$ instead of t), then we obtain the ALTS associated with the STG.

Definition 6 (ALTS associated with a STG) *Let $\mathcal{N}^A = \langle \mathcal{S}, A, \Lambda\rangle$ be an STG, where $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0}\rangle$ and $\mathcal{N} = \langle P, T, F, W\rangle$; the Arc-labelled Transition System associated with $\mathcal{N}^A$ is the 4-tuple $\langle V, E, A, \delta\rangle$ where $V = \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$, $E = \{(m, m') | m, m' \in V$ and $\exists t \in T$, such that $m[t\rangle m'\}$, and for all $(m, m') \in E$, with $m[t\rangle m'$, $\delta(m, m') = \Lambda(t)$.*

The ALTS associated with the STG in figure 1c is the previously presented in the figure 1b, removing the binary vectors appearing in the states.

3 Consistent Signal Transitions Graphs

STGs are used to specify the behaviour of digital systems, and their signals directly correspond to circuit signals. Note that for a given signal, a, in a circuit, the changes a^+ and a^- always alternate. Moreover, from a given initial state, two behaviours such that in one of them the first change for a signal a is a^+, and in the other a^- cannot exist. However, in the STG definition nothing is said about these constraints on the firing sequences; that is, it is possible to build STGs that are not specifying the behaviour of any digital system. The *consistency* of a STG is related to this idea: STGs specifying the behaviour of a digital system are called *consistent* STGs.

Several definitions have been given in the literature to express the idea of consistency of a STG. However, most of them does not only constrain the class of STGs to those that represent the behaviour of a digital system, but also they include constraints related to, for instance, the implementation method from the STG to obtain a circuit or the basic library to build the circuits. Examples of these definitions are: *live* STGs [2, 8, 9, 12, 13, 19], *correct* STGs [5, 6, 10, 15], *implementable* STGs [7, 11] and *well-formed* STGs [2]. Liveness and/or safeness of the underlying net system, finiteness of its space state, net structure corresponding to the net subclasses marked graph or free choice, are some typical constraints imposed in these definitions. Obviously some of these restrictions are not related to the idea of consistency.

A good definition of this property in terms of the STD associated with the STG was given in [10]. Other authors have defined it directly from the STG, giving several characterizations that are not equivalent to the original definition.

In order to represent the behaviour of a digital system, the STG must be consistent. Consistency is a property of the firing sequences of the net system (or, more precisely, the sequences of signal changes). This property specifies constraints in a sequence (or in several related sequences) to the appearing of changes of a given signal. In this sense, in the definition of consistency no additional constraints are needed, e.g. finiteness of the state space, liveness, and so on.

In the sequel, we present the classical definition of consistency for bounded STGs in terms of the consistent associated STD. Later we will see that this definition is equivalent for bounded STGs to one expressed in terms of the constraints that must fulfill the firing sequences. Moreover, we will see that the boundedness property can be removed from the definition of consistency, but in this case the definition must be made in terms of firing sequences, since the STD has an infinite number of states. In any case, bounded STGs will be considered through the paper.

In this section we define consistency following the definition in [10], we give a new characterization of the property directly at the STG level, and we relate it with other properties introduced in the literature dealing with consistency, showing the relations between them.

Definition 7 (Consistent Signal Transition Graph) *Let $\mathcal{N}^A$ be a STG with bounded underlying net system, and let $A^{\mathcal{N}}$ be its associated ALTS. $\mathcal{N}^A$ is said to be* consistent *if there exists a consistent STD for which $A^{\mathcal{N}}$ is the associated ALTS. This STD is said associated with $\mathcal{N}^A$.*

The STG in the figure 1c is consistent, and in this case, its associated STD is unique. In general, a consistent STG can have several associated STDs. This is possible if all transitions with the label corresponding to a given signal are not firable.

From the definition 7 we can define a bijection between the reachable states of the STG and the states of an associated STD. Therefore it is possible to define a function that associates a *code* (the binary vector of signal values) with every reachable marking. We will call it the *coding function.*

Definition 8 (The Coding Function) *Given a consistent STG and an STD $\langle V, E, A, \lambda\rangle$ associated with it, the Coding function is $\mathcal{C} = \lambda \circ MtoV$, where $MtoV : \mathrm{RS}(\mathcal{N}, \mathbf{m_0}) \rightarrow V$ is a bijection.*

3.1 A new characterization of Consistent Signal Transitions Graphs

In a consistent STG, each state determines the value of each signal, and this information (the signal values in each state) is implicitly included in the model. Therefore, in a consistent STG the information about the signal values is redundant. We propose a new approach to characterize consistent STGs (and to analyze their properties) based on the explicit representation of the signal values in net terms and in an explicit way. This information will be included in the STG model by means of places with

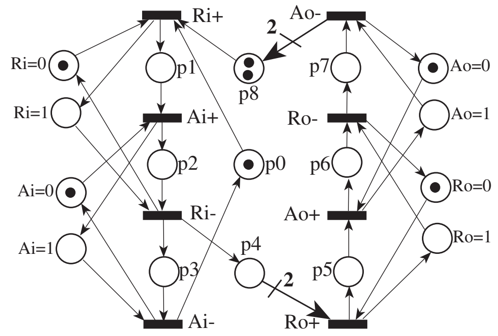

Figure 2: Extended STG corresponding to the STG of figure 1c.

the right initial marking, and in order to have an equivalent model to the original one, the firing sequences have to be preserved. Informally, a place we can remove (or add) from (to) a net system without changing its firing sequences is known as an *implicit place*, and in fact it is a redundant place.

Definition 9 (Implicit Place) *Let $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ be a net system and $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m_0}' \rangle$ the net system resulting from removing place p from $\mathcal{S}$. The place p is* implicit *iff* $\mathrm{L}(\mathcal{N}, \mathbf{m_0}) = \mathrm{L}(\mathcal{N}', \mathbf{m_0}')$, *i.e., the removing of place p preserves all firing sequences of the original net.*

The key of the approach is to add to the STG a pair of places for each signal that explicitly represent the signal value. For example, for a signal a we add the places $a = 0$ and $a = 1$. If place $a = 0$ ($a = 1$) is marked in a state, it means that the value of the signal a is 0 (1) in that state. Transitions with label a^+ (a^-) take a token from the place $a = 0$ ($a = 1$) and put it in the place $a = 1$ ($a = 0$).

As a first step, the initial marking of the added places will not be completely determined, but a restriction will be imposed: only one of the two places representing the value of a signal can be marked; that is, the value of a signal can be one or zero, but not both. Therefore we will obtain a family of STGs.

Definition 10 (Extended STG Dictionary) *Let $\mathcal{N}^A = \langle \mathcal{S}, A, \Lambda \rangle$ be an STG, where $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$, $\mathcal{N} = \langle P, T, F, W \rangle$, A a set of signals, and $\Lambda : T \to A \times \{+, -\}$. The Extended Signal Transition Graph Dictionary associated with $\mathcal{N}^A$, $D^A = \langle \mathcal{S}^A, A, \Lambda \rangle$, is a family of STGs, where $\mathcal{S}^A = \langle \mathcal{N}', \mathbf{m_0}' \rangle$, $\mathcal{N}' = \langle P \cup P_{A=0} \cup P_{A=1}, T, F', W' \rangle$, parameterized by the initial marking of places belonging to $P_{A=0} \cup P_{A=1}$ and satisfying the following conditions:*

1. *The functions $Val_0 : A \to P_{A=0}$ and $Val_1 : A \to P_{A=1}$ are bijective. We will use the following notation: Let $a \in A$ be a signal, $Val_0(a) = a_0$ and $Val_1(a) = a_1$.*

2. *$F' = F \cup F_N$ and for all $f \in F$, $W'(f) = W(f)$.*

3. *For all $t \in T$, $(a_0, t) \in F_N$ and $(t, a_1) \in F_N$ iff $\Lambda(t) = a^+$. In that case $W'((a_0, t)) = 1$ and $W'((t, a_1)) = 1$.*

4. *For all $t \in T$ $(a_1, t) \in F_N$, $(t, a_0) \in F_N$ iff $\Lambda(t) = a^-$. In that case $W'((a_1, t)) = 1$ and $W'((t, a_0)) = 1$.*

5. *For all $p \in P$, $\mathbf{m_0}'[p] = \mathbf{m_0}[p]$, and the initial marking of each pair of places corresponding to a signal a must satisfy the constraint: $\mathbf{m_0}'[a_0] + \mathbf{m_0}'[a_1] = 1$*

Once we have structurally defined the pair of places for every signal, the problem is to determine their initial markings, if it is possible. As the next theorem shows, such marking exists if and only if the STG is consistent.

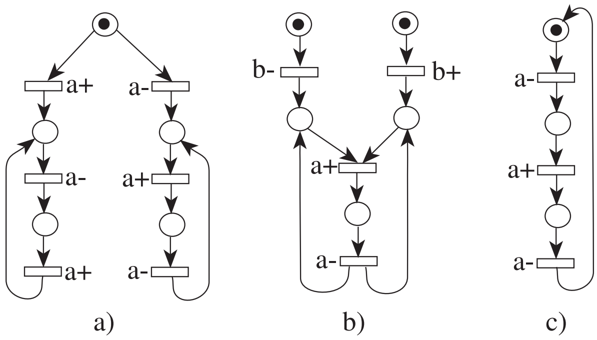

Figure 3: Several counterexamples.

Theorem 11 *Let $\mathcal{N}^A = \langle \mathcal{S}, A, \Lambda \rangle$ be a bounded STG, where $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$, $\mathcal{N} = \langle P, T, F, W \rangle$, $A = \{a, b, c, \ldots\}$, and $\Lambda : T \to A \times \{+, -\}$, and let $D^A = \langle \mathcal{S}^A, A, \Lambda \rangle$ be the Extended STG Dictionary associated with $\mathcal{N}^A$, where $\mathcal{S}^A = \langle \mathcal{N}', \mathbf{m_0}' \rangle$ and $\mathcal{N}' = \langle P \cup P_{A=0} \cup P_{A=1}, T, F', W' \rangle$. The STG $\mathcal{N}^A$ is consistent if and only if there exists a STG belonging to D^A, such that $\forall p \in P_{A=0} \cup P_{A=1}$, p is an implicit place.*

Note that that theorem does not guarantee the uniqueness of the STG in D^A. However, it is unique if each signal in the STG can change its value at least once, as the following corollary states.

Corollary 12 *Let $\mathcal{N}^A$ be a consistent STG as in theorem 11, where for all signal $a \in A$, there exists at least a transition $t \in T$ such that t is firable and $\Lambda(t) = a^*$. The STG in D^A that fulfils theorem 11 is unique.*

From theorem 11, we have a characterization of the set of consistent STGs: the set of STGs such that if we add places representing the value of the signals, these places are implicit. A STG model of this kind allows to specify the behaviour of a digital system and we will call it *Extended STG associated with a consistent STG*, or *Extended STG* for short. An Extended STG model will be denoted $E\mathcal{S}^A$ if $\mathcal{S}^A$ is the corresponding consistent STG.

The figure 2 shows the Extended STG associated with the STG in figure 1c.The reader can easily check that the behaviour of both net systems is the same.

In the sequel, we are going to present other properties introduced in the literature that have been used as part of consistency definitions. Relations between them and with consistency are presented as corollaries, that are easily proved from theorem 11. The first one deals with the alternation between the occurrence of the changes from zero to one and from one to zero for each signal and firing sequence.

Definition 13 (Switchover correctness [6, 15]) *Let $\mathcal{N}^A = \langle \mathcal{S}, A, \Lambda \rangle$ be a STG. A firable sequence in $\mathcal{N}^A$ is switchover correct if for all signal $a \in A$, a^+ and a^- alternate. A STG is switchover correct if all the firable sequences are switchover correct.*

Corollary 14 *A consistent STG is switchover correct.*

The converse of the previous corollary is not true. A counterexample is presented in figure 3a. The STG is switchover correct but is not consistent. It is not possible to assign a consistent code to the signal a in the initial state.

The following corollary completely characterizes the consistency of a STG in terms of its firing sequences; it is necessary to take into account the first occurrence of a transition firing for a given signal in all the sequences.

Corollary 15 *Let $\mathcal{N}^A = \langle \mathcal{S}, A, \Lambda \rangle$ be a bounded STG, where $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$, $\mathcal{N} = \langle P, T, F, W \rangle$, $A = \{a, b, c, \ldots\}$, and $\Lambda : T \to A \times \{+, -\}$. $\mathcal{N}^A$ is consistent if and only if: i) $\mathcal{N}^A$ is switchover correct; and ii) Given a signal $a \in A$, for all firable sequence the first appearing of a change on signal a is of the same type.*

This corollary represents an equivalent definition of consistency in terms of the firing sequences. It is part of the definition of *correctness* in [6], where point ii) is called the *initial stability condition*. Observe that if we use the definition of consistency in terms of firing sequences, theorem 11 also holds if we remove the boundedness condition (this is the because implicit places are characterized in terms of firing sequences preservation).

Another interesting property is called *signal autoconcurrency*, and deals with the possibility of the concurrent firing of two transitions whose labels correspond to the same signal.

Definition 16 (Autoconcurrent Signal [6, 15]) *Let $\mathcal{N}^A = \langle \mathcal{S}, A, \Lambda \rangle$ be a STG, where $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ and $\mathcal{N} = \langle P, T, F, W \rangle$. A signal $a \in A$ is autoconcurrent, if there exists $m \in \mathrm{RS}(\mathcal{S})$ and $t_1, t_2 \in T$ such that $\Lambda(t_1) = a^*, \Lambda(t_2) = a^*, m[t_1 t_2\rangle$ and $m[t_2 t_1\rangle$.*

Corollary 17 *In a consistent STG, there is no autoconcurrent signal.*

The converse of previous corollary is not true, as the example in the figure 3a shows.

In the general case, there does not exist relation between autoconcurrency and switch-over correctness. The STG in the figure 1c is consistent, and therefore switchover correct and without autoconcurrent signals. The examples in figures 3b and 3c show the other non trivial cases: the first is switchover correct and has an autoconcurrent signal, and the second the contrary.

3.2 A Structural Condition for STG Consistency

In theorem 11, we presesented a characterization of STG consistency in terms of implicit places. From this characterization we present in this section a new method inside strutcture analysis that is applicable to any type of net system. The approach is absolutely general, it does not assume a given type of net system. The drawback is that the technique gives only a sufficient condition of STG consistency.

The next property gives a method to check if a place with a given initial marking is implicit. It is based in the solution of a linear programming problem (LPP 1), and therefore it has polynomial time complexity.

Property 18 ([3]) *Let $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$ be a net system. A place p of $\mathcal{N}$, with initial marking $\mathbf{m_0}[p]$, is an implicit place if the LPP 1 has at least one feasible solution and $\mathbf{m_0}[p] \geq z$, where z is the optimal value of LPP 1.*

$$\begin{array}{rll} z = & \mathit{min.} & \mathbf{y} \cdot \mathbf{m_0} + \mu \\ & \mathit{s.t.} & \mathbf{y} \cdot \mathbf{C} \leq \mathbf{C}[p, T] \\ & & \mathbf{y} \cdot \mathbf{Pre}[P, t] + \mu \geq \mathbf{Pre}[p, t] \quad \forall t \in p^\bullet \\ & & \mathbf{y} \geq 0, \mathbf{y}[p] = 0 \end{array} \tag{1}$$

From this property is straightforward to obtain an algorithm to check if a STG is consistent. For every signal a, we compute the optimal solutions of LPP 1 for the pair of places representing this signal. If the addition of both optimal values is less or equal to 1, this signal does not produce any consistency violations.

Proposition 19 *Let $\mathcal{N}^A$ be a STG. If the algorithm* Consistent_STG($\mathcal{N}^A$) *returns true, then $\mathcal{N}^A$ is consistent.*

Theorem 11 gave a condition about the initial marking of the added places corresponding to a signal: its addition has to be one. However we have relaxed this condition in the algorithm: its addition has to be less or equal to one. The reason is that if for a given signal all its associated transitions are dead in the initial marking, then the places representing this signal are implicit with a marking greater or equal to zero. This fact does not contradict the theorem 11, because in this case

Algorithm 1 Consistent_STG

Input: A STG $\mathcal{N}^A$ as in definition 10
Output: Boolean
begin
  Compute $\mathcal{N}' = \langle P \cup P_{A=0} \cup P_{A=1}, T, F', W' \rangle$ following definition 10
  For all $a \in A$ do
    Compute z_0 as the optimal solution of the LPP 1 with $p = Val_0(a)$
    Compute z_1 as the optimal solution of the LPP 1 with $p = Val_1(a)$
    If $z_0 + z_1 > 1$ **then return**(false)
  end for
  return(true)
end

an initial marking whose addition is one always exists (obviously there exist two, and we can choose any of them arbitrarily).

The reason why this approach is only a sufficient condition of consistency is in the LPP 1. The optimal solution of the problem is not, in the general case, the *minimum* initial marking that makes the place an implicit place. If we restrict the problem to marked graphs, then the optimal solution is the minimum, and the condition is necessary and sufficient. However, even for the class of live and safe free-choice P/T systems is only sufficient, because the net with the added places usually is not free-choice any longer.

4 The Unique State Coding Property

Given a behavioral specification of a sequential digital system, the corresponding states can be coded in different ways. The coding is chosen according to diverse criteria; for instance, a typical goal is to minimize the number of bits of the code. Thinking of STG specifications, where the signals are present in the model, a criteria could be to code the states with the value of the signals in the model, if it is possible. The Unique State Coding property deals with this possibility: if each state in the system implies a different vector of signal values, this vector can be used to code the states. Note that this property is not constrained to asynchronous circuits, but it is also applicable to synchronous designs [16].

To formally define the USC property, we will use the coding function. Therefore this property is applied to consistent STGs only.

Definition 20 (Unique State Coding [18]) *A consistent STG has the Unique State Coding Property if its coding function $\mathcal{C}$ is injective.*

We can use the Extended STG model to characterize the property without referring to the coding function (and therefore to the STD model). This fact will allow us to study the property directly from the STG.

Theorem 21 *Let $\mathcal{N}^A = \langle \mathcal{S}, A, \Lambda \rangle$ be a consistent STG, where $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$, $\mathcal{N} = \langle P, T, F, W \rangle$, $A = \{a, b, c, \ldots\}$, and $\Lambda : T \rightarrow A \times \{+, -\}$, and let $\mathcal{N}'^A = \langle \mathcal{S}', A, \Lambda \rangle$ be the underlying net system of the Extended STG associated with it, where $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m_0}' \rangle$, $\mathcal{N}' = \langle P', T, F', W' \rangle$. $\mathcal{N}^A$ has the Unique State Coding Property if and only if there do not exist two markings $\mathbf{m}_1, \mathbf{m}_2 \in \mathrm{RS}(\mathcal{N}', \mathbf{m_0}')$, $\mathbf{m}_1 \neq \mathbf{m}_2$, such that for all $p \in P' \setminus P$, $\mathbf{m}_1(p) = \mathbf{m}_2(p)$.*

As we have stated before, USC property guarantees that each state of the circuit whose behaviour is described by a STG is uniquely determined by the value of the signals. Therefore it is possible to distinguish two states using only information about the signal values. But the question is: Is this situation intrinsically bad if we are thinking of using the signals as state variables? The answer is not always. If several STD states have the same code but the behaviours that follows them are the same, an equivalence relation between STD states can be defined, and the problem disappears if we refer the USC property to the equivalence classes. This equivalence relation was defined in [10], for both the STD states and the STG markings. This fact makes that the USC property does not correspond exactly with its intended purpose, that is, a way to know if the states of the behaviour specified by the STG can be represented by the signal values. Moreover, and from a practical point of view, while it is very easy to detect and correct this problem at the STD level, it is very hard at the STG level. If USC property applies to STG markings without considering the equivalent classes, it is only a sufficient condition of the claim.

A sufficient condition to check the USC property is easily deduced from the theorem 21. It is based on an integer programming problem. However, the use of the state equation to describe the set of reachable states gives only a sufficient condition (if $\mathbf{m} \in \mathrm{RS}(\mathcal{N}, \mathbf{m_0})$, then there exists a nonnegative integer vector $\boldsymbol{\sigma}$ such that $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$). The problem is well known long time ago: there can be integer solutions of the state equation that are not reachable markings. This solutions are called spurious solutions [3].

Corollary 22 *Let $\mathcal{N}^A$ and $\mathcal{N}'^A$ be a consistent STG and the ESTG associated with it, as in theorem 21. $\mathcal{N}^A$ has the Unique State Coding property if the IPP 2 has no solution, where $\mathbf{C}$ is the incidence matrix of $\mathcal{N}^A$, $\mathbf{C}' = \mathbf{C}[\mathbf{P}' \setminus \mathbf{P}, \mathbf{T}]$, is a matrix formed by the rows corresponding to the added places in $\mathcal{N}'^A$, and b is the bound of the net system.*

$$
\begin{array}{l}
\mathbf{m_1} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}_1 \\
\mathbf{m_2} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}_2 \\
\mathbf{C}' \cdot \boldsymbol{\sigma}_1 = \mathbf{C}' \cdot \boldsymbol{\sigma}_2 \\
\mathbf{v} \cdot \mathbf{m_1} - \mathbf{v} \cdot \mathbf{m_2} > 0 \ , \quad \textit{where} \quad \mathbf{v}[x] = (b+1)^x \ , \quad x = 0 \ldots |P| - 1 \\
\mathbf{m_1}, \mathbf{m_2}, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2 \in \mathbb{Z}^+
\end{array}
\tag{2}
$$

Note the way to express that the two markings are different. We associate a number to each marking. As the problem is symmetric with respect to the two markings, by imposing that the difference between them is greater than zero, we impose that the markings are different.

5 The Complete State Coding Property

In the definition of STG we have distinguished two types of signals: input signals, that correspond to inputs to the digital system, and output signals, that are the responses of the circuit. The system itself does not control input signals. The environment of the system receives the output signals as inputs, and produce as outputs the inputs to the system. Note that the STG specifies *both* the behaviour of the system and the behaviour of the environment (in fact if the digital system operates in a different environment that does not fulfill the STG specification, we cannot know how the system will behave, because such case is not specified). In other words, we can consider that the STG specifies a closed system, that is the system itself augmented with its environment.

Now consider again the question about the coding of the states by means of the signals. Suppose that two states with the same code have not to produce a change in any outputs, but they have to wait for an input change, different in each case. Note that the environment of the system, that also fulfils the STG specification, knows in some way which state the system is in, and therefore it will send the right input change corresponding to each of the two states. The same argument could be applied if the states had to generate an output change, but this change was equal in both. This idea is expressed by means of the *Complete State Coding Property*: two states with the same code will be distinguished if the outputs they have to produce are the same, though the possible inputs are different in both (the environment will send the right input in each case).

Definition 23 (Complete State Coding [10]) *A consistent STG* $\mathcal{N}^A = \langle \mathcal{S}, X \cup Z, \Lambda \rangle$*, where* $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$*,* $\mathcal{N} = \langle P, T, F, W \rangle$*, has the Complete State Coding Property if and only if for all* $\mathbf{m}_1, \mathbf{m}_2 \in \mathrm{RS}(\mathcal{S})$*,* $\mathbf{m}_1 \neq \mathbf{m}_2$*, such that* $\mathcal{C}(\mathbf{m}_1) = \mathcal{C}(\mathbf{m}_2)$*, and for all* $t \in T$ *such that* $\Lambda(t) \in Z$ *and* $\mathbf{m}_1[t\rangle$*, there exists* $t' \in T$ *such that* $\Lambda(t') = \Lambda(t)$ *and* $\mathbf{m}_2[t'\rangle$*.*

As with the USC property, we give a characterization of the CSC property directly at the STG level, that does not refer to the coding function. As in the former, the Extended Signal Transition Graph is used.

Theorem 24 *Let* $\mathcal{N}^A = \langle \mathcal{S}, A = X \cup Z, \Lambda \rangle$ *be a consistent STG, where* $\mathcal{S} = \langle \mathcal{N}, \mathbf{m_0} \rangle$*,* $\mathcal{N} = \langle P, T, F, W \rangle$*,* $A = \{a, b, c, \ldots\}$*, and* $\Lambda : T \to A \times \{+, -\}$*, and let* $\mathcal{N}'^A = \langle \mathcal{S}', A = X \cup Z, \Lambda \rangle$ *be the Extended STG associated with it, where* $\mathcal{S}' = \langle \mathcal{N}', \mathbf{m_0}' \rangle$*,* $\mathcal{N} = \langle P', T, F', W' \rangle$*.* $\mathcal{N}^A$ *has the Complete State Coding Property if and only if for all* $\mathbf{m}_1, \mathbf{m}_2 \in \mathrm{RS}(\mathcal{S}')$*,* $\mathbf{m}_1 \neq \mathbf{m}_2$*, such that* $\forall p \in P' \setminus P$*,* $\mathbf{m}_1(p) = \mathbf{m}_2(p)$*, and for all* $t \in T$ *such that* $\Lambda(t) \in Z \times \{+, -\}$ *and* $\mathbf{m}_1[t\rangle$*, there exists* $t' \in T$ *such that* $\Lambda(t') = \Lambda(t)$ *and* $\mathbf{m}_2[t'\rangle$*.*

An IPP-like formulation of this problem is also possible. For the lack of space we constrain ourselves to the case of binary STGs. *Linear Enabling Functions* [1] can be used to extend the result. It is only a sufficient condition by the same reason explained in the case of the USC property, the spurious solutions of the state equation.

Proposition 25 *Let* $\mathcal{N}^A$ *and* $\mathcal{N}'^A$ *be a consistent binary STG and an ESTG associated with it, as in theorem 21.* $\mathcal{N}^A$ *has the Complete State Coding property if the IPP 3 has no solution, where* $\mathbf{C}$ *is the incident matrix of* $\mathcal{N}^A$*,* $\mathbf{C}' = \mathbf{C}[\mathbf{P}' \setminus \mathbf{P}, \mathbf{T}]$ *is a matrix formed by the rows corresponding to the added places in* $\mathcal{N}'^A$*, and* $T_a = \{t \in T \mid \Lambda(t) = a^*\}$*.*

$$\begin{array}{l}
\mathbf{m_1} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}_1 \\
\mathbf{m_2} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}_2 \\
\mathbf{C}' \cdot \boldsymbol{\sigma}_1 = \mathbf{C}' \cdot \boldsymbol{\sigma}_2 \\
\mathbf{v} \cdot \mathbf{m_1} - \mathbf{v} \cdot \mathbf{m_2} > 0 \ , \quad \textit{where} \quad \mathbf{v}[x] = 2^x \ , \quad x = 0 \ldots |P| - 1 \\
\sum_{a \in Z} \quad \mathbf{1}^T \cdot \boldsymbol{\sigma}_a = 1 \\
\forall a \in Z \ , \quad \mathbf{m_1} \geq \mathbf{Pre}[P, T_a] \cdot \boldsymbol{\sigma}_a \\
\forall a \in Z \textit{ and } \forall t \in Z \quad \textit{s.t.} \quad \Lambda(t) = a^* \ , \quad \mathbf{1}^T \cdot \mathbf{Pre}[P, t] - \mathbf{m_2}^T \cdot \mathbf{Pre}[P, t] \geq \mathbf{1}^T \cdot \boldsymbol{\sigma}_a \\
\mathbf{m_1}, \mathbf{m_2}, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \boldsymbol{\sigma}_a \in \mathbb{Z}^+
\end{array} \tag{3}$$

6 Conclusions

Classical STGs are interpreted net systems with information on signal changes. In this paper we have proposed a new model for the description of the behaviour of digital systems that includes all information related to signals in net terms. This new model is the original underlying net system of the STG with an additional set of places representing the values of the signal and their changes. The advantages of this new model are: (1) All questions on STGs can be formulated in net terms because the new model contains all information of the original STG in net terms; (2) We can use standard analysis methods inside the rich body of results about Petri Net theory.

A second question addressed in the paper is the clarification of the consistency property and its relation with other classical results of STGs. As a consequence, we have presented a characterization of the consistency property of a STG based on the detection of some implicit places in our extended STG.

We have proposed the use of structural analysis methods of Petri Nets based on linear algebraic techniques for the analysis of STGs properties. Obviously, these methods need our extended STG. The advantages of the approach are: (1) We can analyze STGs represented by generalized Petri Nets (so, the full descriptive power of PNs can be used without the limitations due to the analysis algorithms); (2)

We can use standard and general algorithms to analyze STGs (algorithms to solve linear programming problems) with, in general, good practical computational complexities. Nevertheless, the use of the net state equation, as a linear description of the reachability set, introduces spurious (non reachable) markings that make that the obtained conditions are necessary or sufficient, but not both. In any case, the approach is more general than previous structural analysis techniques.

References

[1] J.L. Briz, and J.M. Colom. Implementation of weighted place/transition net based on linear enabling functions. In R. Valette, editor, *Application and Theory of Petri Nets 1994*, volume 815 of *Lecture Notes in Computer Sciences*, pages 99–118. Springer Verlag, Berlin, 1994.

[2] T.-A. Chu. *Synthesis of Self-Timed VLSI Circuits from Graph-Theoretic Specifications.* PhD thesis, MIT Laboratory for Computer Science, June 1987.

[3] J.M. Colom and M. Silva. Improving the Linearly based Characterization of P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Sciences*, pages 113–145. Springer Verlag, Berlin, 1991.

[4] N. Deo. *Graph Theory with Applications to Engineering and Computer Science.* Prentice-Hall, Inc. Englewood Cliffs, N.J., 1974.

[5] M. Kishinevsky, A. Kondratyev, A. Taubin, and V. Varshavsky. Analysis and identification of self-timed circuits. *Formal Methods in System Design*, 4(1):33–75, 1994.

[6] A. Kondratyev, J. Cortadella, M. Kishinevsky, E. Pastor, O. Roig and A. Yakovlev. Checking Signal Transition Graph Implementability by Symbolic BDD Transversal. In *Proc. The European Design and Test Conference*, pages 325–332, March 1995.

[7] A. Kondratyev and A. Taubin. Verification of speed-independent circuits by stg unfoldings. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 64–75, November 1994.

[8] L. Lavagno, K. Keutzer, and A. Sangiovanni-Vincentelli. Algorithms for synthesis of hazard-free asynchronous circuits. In *Proc. ACM/IEEE Design Automation Conference*, pages 302–308. IEEE Computer Society Press, 1991.

[9] L. Lavagno, C. Moon, R. Brayton, and A. Sangiovanni-Vincentelli. Solving the state assignment problem for signal transition graphs. In *Proc. ACM/IEEE Design Automation Conference*, pages 568–572. IEEE Computer Society Press, June 1992.

[10] L. Lavagno. *Synthesis and Testing of Bounded Wire Delay Asynchronous Circuits from Signal Transition Graphs.* PhD thesis, U.C. Berkeley, November 1992. Technical report UCB/ERL M92/140.

[11] L. Lavagno, Cho W. Moon, Robert K. Brayton, and Alberto Sangiovanni-Vincentelli. An efficient heuristic procedure for solving the state assignment problem for event-based specifications. *IEEE Transactions on Computer-Aided Design*, 14(1):45–60, January 1995.

[12] L. Lavagno, Kurt Keutzer, and Alberto Sangiovanni-Vincentelli. Synthesis of hazard-free asynchronous circuits with bounded wire delays. *IEEE Transactions on Computer-Aided Design*, 14(1):61–86, January 1995.

[13] C. W. Moon. *Synthesis and Verification of Asynchronous Circuits from Graphical Specifications.* PhD thesis, U.C. Berkeley, 1992.

[14] T. Murata. Petri nets: properties, analysis, and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[15] E. Pastor. *Structural Methods for the Synthesis of Asynchronous Circuits from Signal Transition Graphs* PhD Thesis, Universidad Politécnica de Cataluña, 1996.

[16] I. Pomeranz and K. -T. Cheng. State Assignment Using Input/Output Functions. In *Proc. ACM/IEEE Design Automation Conference*, pages 573–577. IEEE Computer Society Press, June 1992.

[17] L. Y. Rosenblum and A. V. Yakovlev. Signal graphs: from self-timed to timed ones. In *Proceedings of International Workshop on Timed Petri Nets*, pages 199–207, Torino, Italy, July 1985. IEEE Computer Society Press.

[18] P. Vanbekbergen, F. Catthoor, G. Goossens, and H. De Man. Optimized synthesis of asynchronous control circuits form graph-theoretic specifications. In *Proc. International Conf. Computer-Aided Design (ICCAD)*, pages 184–187. IEEE Computer Society Press, 1990.

[19] P. Vanbekbergen, G. Goossens, F. Catthoor, and H. J. De Man. Optimized synthesis of asynchronous control circuits from graph-theoretic specifications. *IEEE Transactions on Computer-Aided Design*, 11(11):1426–1438, November 1992.

[20] A.V. Yakovlev. On limitations and extensions of STG model for designing asynchronous control circuits. In *Proc. International Conf. Computer Design (ICCD)*, pages 396–400. IEEE Computer Society Press, October 1992.

[21] A.V. Yakovlev, M. Kishinevsky, A. Kondratyev, and L. Lavagno. OR causality: modeling and hardware implementation. In R. Valette, editor, *Application and Theory of Petri Nets 1994*, volume 815 of *Lecture Notes in Computer Sciences*, pages 568–587. Springer Verlag, Berlin, 1994.