From Proceedings of International Workshop on Timed Petri Nets,
Torino, Italy, July 1985, IEEE CS Press, pp. 199-207.

SIGNAL GRAPHS : FROM SELF-TIMED TO TIMED ONES

L. Ya. Rosenblum, A.V. Yakovlev

Computing Science Department of the Leningrad
Electrical Engineering Institute
Leningrad, USSR 197022

ABSTRACT

The paper aims at the following goals: 1) to make a bridge between the concepts of Petri Nets theory and the works concerned with self-timed systems (speed-independent circuits); 2) to draw the designers' attention to a signal graph model which is an interpreted marked graph by demonstrating its advantages in producing concise specifications of asynchronous system behaviour and being an alternative tool to timing diagrams, which are traditionally used for interface protocol specification; 3) to generalize the signal graph model to allow finite time delay values to be taken into account, and to suggest a new approach to the analysis of the temporal behaviour of dynamic systems aiming at reducing the complexity of analysis procedure.

1. Introduction

The Petri Nets theory /1-4/ is popular because it is a suitable and effective tool oriented to modelling, analysis and synthesis of parallel processes of different types. Most researchers are satisfied with such advantages of Petri Nets as: a) an ability to reflect asynchrony, parallelism and non-determinacy of processes and the dynamics of their operation; b) a simplicity of syntax, a comprehensibility and a transparency of the model's graphical appearance, and at the same time high functionability due to the large choice of the hierarchically and linguistically structured functional and syntax subclasses of the general formalism. Today one may readily confirm that the theory of Petri Nets has essentially become a quite independent scientific subject. Moreover its application areas are constantly expanding. The development of Petri Nets theory is motivated by the theory's internal needs but the main drive is made through the demands of system design.

It should be noted that in many aspects the Petri Nets theory is tightly linked with automata theory and is essentially a derivative of the latter. However, since it is developing in a relatively autonomous manner, the Petri Nets theory is gradually moving away from the classic automata theory, and it appears to be an appropriate moment to build bridges between Petri Nets and one of the most interesting parts of automata theory, which was for the first time studied in pioneer works by D.E. Muller /5,6/. Generally the latter approach follows the same demands as we require from the Petri Nets formalism, namely: to provide a tool for the description of dynamic behaviour of discrete systems. Although Muller's theory appeared almost at the same time as Petri nets i.e. more than twenty years ago, the Muller model for some extrinsic reasons had not actually been touched until well into the middle of the '70s. It was thought that the Muller model might be used only within the frames of low-level hardware design or at least of analysis and synthesis of a rather small class of devices - so called speed-independent circuits. As a matter of fact the Muller transition diagrams and the research into the lattice-theoretical properties of asynchronous circuits were obviously first attempts towards describing and analysing the parallel asynchronous processes. In fact the behaviour of a speed-independent circuit could represent a process characterized by such general properties as determinacy, persistancy, confluency, liveness, safety etc.

Petri Nets describe process behaviour on the basis of the 'condition-event' approach in contrast to the Muller model which is closer to the 'state-transition' concept. Attempts to generalize the latter one were a transition system by R. Keller /8/ and an asynchronous process by V. Varshavsky et al /9/. It is worth noting that the renaissance of Muller model is also necessary because it is capable to solve one of the most urgent state-of-art problems of VLSI design: the self-timing problem (see MIT Workshop Report on this topic /10/).

The results under discussion here are based on a signal graph. They essentially utilize both the approaches mentioned. From the one side signal graphs are interpreted marked graphs (subcass of Petri Nets), from the other side they are more compact specification tool for transition diagrams (Muller diagrams). It is established

that the important subclass of speed-independent circuits, so called distributive circuits /6,7/ can be described by signal graphs.

The analysis and sythesis of discrete devices implementing data transfer protocols is often carried out by using informal description languages like timing or voltage charts. These diagrams attract the designers attention by their 'customary clearness' which however is not so evident when attempts are undertaken to discover some of the qualitative properties of protocols. The organization of interface on the 'request-acknowledge' principle corresponds through some interpretation both to particular subclasses of Petri Nets and to semi-modular Muller diagrams. These diagrams have the necessary self-timing features. Signal graphs are therefore such a formal tool for the analysing and specifying of self-timed protocols.

Finally, we propose the generalisation of signal graphs for the case of the time introduction: timed signal graphs. In this case the analysis of the time behaviour is reduced to the search in transition diagram for particular states which, although violate some self-timing properties, may not lead to the bad behaviour if the time constraints of its transitions are satisfied. This analysis technique is the main distinctive feature of approach proposed in comparison to known ones.

2. Petri Nets and Marked Graphs: the main definitions

A Petri Net (PN) is a bipartite oriented graph $PN=\langle P,T,E,\mu^o\rangle$, where P is a finite set of places, T is a finite set of transitions, E is a finite set of arcs ($E \subseteq (PxT) \cup (TxP)$), μ^o is an initial marking ($\mu^o : P \to Z$, where $Z = \{0,1,2,..\}$, i.e. is a set of non-negative integers) /2/.

The sets of input and output places of the given transition $t_j \in T$ are denoted by $I(t_j)$ and $O(t_j)$ respectively. Similarly O(pi) and I(pi) are denotations of sets of transitions which are respectively output and input ones for the given place $pi \in P$. Marking is usually visualized by tokens in places.

The firing of some enabled transition (local action) generally causes the substitution of Petri Net marking (global state). Thus the dynamics of PN behaviour can adequately be described by $\langle \mu^o, \to, M\rangle$, where μ^o is an initial marking, $\to$ is a direct marking sequence relation and M is a set of markings reachable from μ^o. The depiction of this triple is an oriented graph, the vertices of which are labelled by vectors $\mu^j = \langle \mu_1^j, \mu_2^j, ..., \mu_n^j \rangle$, where $n = |P|$. Such a graph is called a marking diagram. If we label arcs of this diagram by the denotations of corresponding firing transitions then for this marking diagram and, therefore, for the Petri Net we can build another diagram which is called a cumulative diagram (of Petri Net transition firings). Initial marking is mapped on vector a^o, consisting of all zeros. The dimension of vector is equal to the cardinality $|T|$ of set T. For a transition $\mu^o \xrightarrow{t_k} \mu^1$ we build a vector a^1 which differs from a^o only by the presence of one in the kth component position which means that transition t_k has fired for the first time. When moving along the sequence of the making diagram it is quite straightforward to map (one-to-many) on vectors $a^o, a^1, ..., a^s$. The component values on each vector are equal to the numbers of corresponding transition firings which occur on this sequence. Thus it is clear that the cumulative diagram represents the history of Petri Net operation. More rigorously it can be defined as a set of non-negative integer vectors and a partial order relation ($a \leq b$ if $a_j \leq b_j$ for any j). So the cumulative diagram is a Hasse diagram.

Of further concern are the following definitions:

- safe, if for any reachable marking none of the places can store more than one token, i.e. for any $pi \in_p \mu_i \leq 1$;
- persistent, if for any reachable marking μ providing enabled transition $tj \in T$ all the markings which are reachable from μ either hold this transition enabled or imply it is fired on the sequence from μ(or a once enabled transition can not be disabled by the firing of other transitions);
- a marked graph, if each place pi has not more than one input and one output transition, i.e. if $|I(pi)| \leq 1, |O(pi)| \leq 1$.

A marked graph can also be defined as a monochromatic oriented graph $MG = \langle V,E,\mu^o\rangle$, where V is a finite set of vertices (analogous to transitions of the Petri Net), E is a finite set of arcs, E=VxV (analogous to the places together with one input arc and one output arc of the Petri Net). Some fundamental results in marked graphs are in /11/.

In a similar way as for Petri Nets a vertex $v \in V$ is enabled if each of its input arcs has at least one token. The firing rule for enabled vertices is straightforward. If a vertex fires then one token is removed from each of its input arcs and one token is added to each of its output arcs (indivisible operation). The cycle in graph having exactly one token is called a synchrocycle.

Following statements (formal proof is omitted here) concerning the classification of Petri Nets in terms of the lattice theory /12/ will further be basic for the establishing relationship between transition diagrams and signal graphs.

Statements

1. The cumulative diagram of a persistent Petri Net is a semimodular lattice with the zero element.
2. The cumulative diagram of a persistent and safe Petri Net is a distributive lattice with the zero element.
3. The cumulative diagram of a marked graph is a distributive lattice with the zero element.

3. Transition Diagrams and Muller Diagrams

A transition diagram (TD) is an oriented graph $\langle S,F\rangle$ where S is a finite set of vertices (states) and $F \subseteq SxS$ is a finite set of arcs (transitions). Each state is encoded by a vector consisting of n values of variables $xi(xi \in X$, where X is a finite set of discrete variables) given on the domain Z^f, which is a finite subset of non-negative integers. If $Z^f = \{0,1\}$, then TD is called binary. If $SuFSv$ then an arc is directed from Su to Sv and if in Su and $Sv(u \neq v)$ variable xi has different values, then in Su its value is marked by an asterisk. The variable marked by the asterisk is called enabled (or excited). For excited variable it is possible to perform an action concerned with changing the value of xi. The variables which are not marked by asterisk are called stable. The firing of variable xi (transition) is an action concerned with substitution of state Su in which xi is enabled by state Sv in which the value of xi has changed (xi becomes stable).

Therefore the TD describes the allowed sequences of variable value changes ordered by the relation F and the duration time of any transition is supposed to be arbitrary but finite.

It is accepted that in an initiated TD an initial state is explicitly labelled. Some definitions aiming at classification of TDs are given below. They are given only for a binary TD because the generalization to multiple-valued case is straightforward.

The state of TD is called:

- multiple (non-unique) if TD has at least one state that differs from the given one only by asterisk setting rather than by variables values (e.g., 10*1 and 1*01);
- bifurcate if it has more than one excited variable;
- detonant with respect to xi if it is bifurcate and variable xi is stable in it, and there are two or more states directly reachable from it in which this variable is excited, i.e. there are the following possible branches of variable states:

$$1 \prec \begin{matrix} 1* \\ 1* \end{matrix} \quad \text{or} \quad 0 \prec \begin{matrix} 0* \\ 0* \end{matrix}$$

- conflicting with respect to xi, if variable xi is excited in it and there exists such a directly reachable state in which xi is stable while it has the same value, i.e. the following transitions of xi are possible: $1* \rightarrow 1$ or $0* \rightarrow 0$.

The binary TD is called:

- contradictory if it has at least a pair of multiple states;
- sequential if it has no bifurcate and conflicting states;
- distributive if it has no detonant and conflicting states;
- semi modular if it has no conflicting states.

Classes of sequential, distributive and semi modular TD are denoted K,D and U respectively.

Statement 4. The following inclusion holds for these classes: $U \supset D \supset K$.

Let in the bifurcate state S k component variables be excited ($k \geq 2$). The subcube of (bifurcate) state S is a set of 2^k states derived from S by means of all perturbations of excited variable firings. If all states of the subcube are directly reachable from S then such a fragment of TD is called a hammock. A subhammock is a fragment of the hammock in which either some subcube states are not defined or some arcs belonging to the hammock are absent. TD is called regular if for each bifurcate state S oen of the following conditions holds:

1) all of its subcube states form a hammock;
2) some of subcube states form a subhammock, and the other subcube states (complementing subhammock to a hammock) are not elsewhere presented in TD.

A Muller circuit is a model given by a system of Boolean equations

$$xi=fi(x1,x2,\ldots,xi,\ldots,xn),\ i=1,2,\ldots,n.$$

An initiated model also has an explicit initial state, defined by fixed values of Boolean variables $x1,x2,\ldots,xn$, i.e. $B=\{0,1\}$ is their domain. A circuit variable is called excited if for some state $xi=fi$, and stable otherwise.

The tendency of the excited variables to become stable generates the dynamic behaviour of the circuit, i.e. transitions from an initial state to other states which can be determined according to the equation system. Consequently the firing of variables excited in these succeeding states causes new transitions and so on. It is obvious that the operation of a Muller circuit may be described by a binary transition diagram. However these diagrams have their own specific properties that is why they can be named distinctively. The TD generated by a Muller circuit is called a Muller diagram (MD).

Statements

5. A TD is an MD if and only if it is binary, non-contradictory and regular.
6. Not every MD is semimodular.

While the statement 6 is obvious the proof of the statement 5 is quite straightforward due to the rules of transition from a TD having the above mentioned properties to a Muller circuit. These rules consist of making a truth table mapping illustrated by the following example:

Example. Having analyzed the TD shown in Fig. 1 one can state that the TD is binary, non-contradictory and regular (hence, it is an MD). Besides that it is distributive. Table 1 is a truth table in which all $2^3 = 8$ vectors at dimension 3 are defined. After minimization of each function $xi=f(x1,x2,x3)$ $(i=1,2,3)$ one obtains the following system:

$$x_1 = x_2x_3 \vee x_1(x_2 \vee x_3),$$
$$x_2 = \bar{x}_1,$$
$$x_3 = \bar{x}_1.$$

Table 1

$x_1x_2x_3$	$x_1x_2x_3$
0 0*0*	0 1 1
1*0 0	0 0 0
0 1 0*	0 1 1
1 1*0	1 0 0
0 0*1	0 1 1
1 0 1*	1 0 0
0*1 1	1 1 1
1 1*1*	1 0 0

Generally, for any MD in which less than 2^n vectors of length n are defined one can derive a number of systems of equations. However, for a system with a given initial state one can derive a unique MD. The algorithm for forming an MD from a Muller circuit is based upon the computation of characteristic Boolean functions of reachability set states /15/. Here it is not significant and hence omitted.

4. Signal Nets and Signal Graphs

In order to build a bridge between Petri Nets and transition diagrams (in other words, Muller circuits) one should make an interpretation of Petri Nets, for example, by means of labelling transitions. The semantics of this labelling is expressed by changes of signal values which describe events in the system being modelled. Let $X=\{x1,x2,\ldots,xn\}$ be a set of discrete variables. Every variable xi has its own finite set of values (states) $\sigma_i^1,\ldots,\sigma_i^{ki}$, encoded say by non-negative integers. For each variable xi a set D(xi) of allowed changes δxi is defined.

Examples. If xi is a binary signal then $D(xi)=\{xi^{0-1},xi^{1-0}\}$. For the sake of conciseness xi^{0-1} will be denoted by xi^+ and xi^{1-0} by xi^-. If xi is a ternary signal and is defined on the domain {0,1,2}, where the values 0,1 and 2 correspond, say, to low, middle and high potential levels then $D(xi)=\{xi^{0-1},xi^{1-2},xi^{2-1},xi^{1-0}\}$. If xi models a bundle of parallel bus wires then two separate classes of states-information and transit (or spacer) - can be defined by 1 and 0 respectively.

So the interpretation means conferring labels on the Petri Net or marked graph, i.e. more formally it means definition of a partial function $\varepsilon:T \to D$ (or $\varepsilon:V \to D$, for signal graph), where $D=\bigcup D(xi)$. Triple $\langle MG,D,E\rangle$ is called a signal graph and triple $\langle PN,D,E\rangle$ is a signal (Petri) net. Here MG and PN are marked graph and Petri Net respectively.

Somewhat similar models were studied in /13, 14/. However, the main objectives of /13/ were the modelling of request-acknowledge systems and investigation of composition conditions for such systems. In /14/ the so called taxogram is discussed but it has no explicit marking mechanism.

Now one can notice that the interpretation (or labelling) function allows the derivation from a marked graph or Petri Net not only of a marking diagram but also of a transition diagram each state vertex of which corresponds to a marking vertex in the marking diagram. However this correspondence can not necessarily be one-to-one because of the contradictory states which are possible in TD. At the same time the signal graph generates not only a cumulative diagram of transition (vertex) firings but also cumulative diagram of variable firings. The vectors of this cumulative diagram are comprised of the numbers of variable firings so the vectors length is less because the number of variables is always less at least by 2 times than the number of vertices of the graph. But it must be said that the Hasse diagram structure of the former cumulative diagram under certain conditions can be the same as that of the latter one, i.e. the lattice properties may hold. This condition is expressed through definition of a valid labelling function.

The labelling function ε is called valid for given marked graph MG and set of allowed changes D if for any variable xi all its changes δxi belong to some synchrocycle. In other words, a valid labelling function avoids conflicts in determination of the next value of a given variable, i.e. no variable value changes can occur simultaneously with another change of the same variable.

Examples. A marked graph corresponding to the distributive transition diagram in Fig. 1 having been labelled by changes of signals x1, x2 and x3 becomes a signal graph and is shown in Fig. 2. Fig. 3 is an example of a signal net which can not be represented by a signal graph, because the non-interpreted Petri Net corresponding to Fig. 3 is not a marked graph.

Statements

7. A signal graph generates a distributive transition diagram if it has a valid labelling function. (Proof follows from the Statement 3)
8. A signal graph with a valid labelling function can generate a contradictive transition diagram.

A signal graph with a valid labelling function is called normal if for some allowed sequence of markings it has no subset of variables $X^1 \subset X$ which can proceed through the whole cycle of their values while the other variables (from $X \setminus X^1$) stay unchanged.

Statements

9. A normal signal graph function generates a non-contradictory distributive transition diagram.
10. A signal net generates TD which is semi-modular (distributive) if two conditions hold:
    1. the corresponding non-labelled Petri Net is persistent (and safe)
    2. for no marking there are two or more enabled transitions which are labelled with different changes of the same variable xi.

(Proof follows from the Statements 1 and 2)
Below a normal signal graph generating a

distributive Muller diagram will be called self-timed.

5. Timed Signal graphs

The above described signal graph model is a suitable tool for representation of parallel processes, in particular, distributive processes. Since firing time limits are not established in the signal graph the time delays can be arbitrary but finite. This model is predominantly meant for the investigation of self-timing properties (non-clocked behaviour) of phenomena. However a great deal of applications demands determination of time intervals. The most adequate examples are those of data transfer interfaces and digital controllers.

Let the signal graph be supplemented with another type of vertices denoted by T(K) which model inclusion of built-in delay with a value equal to K time units. It is also natural to admit the following denotations: $T(\geq K)$, $T(\leq K)$ and $T(K1 \leq, \leq K2)$ the semantics of which is quite obvious. We also allow the vertices of type δxi to be supplied with time attribute of the following kind: $K1 \leq t(\delta xi) \leq K2$ etc in a similar way as for the above 'pure' time vertices. Such modified signal graph will be called a timed signal graph.

Usually the data transfer bus protocols are specified by timing diagrams. For example, the timing diagram of the "read" operation protocol for the UNIBUS is shown in fig. 4. It has signals AC (modelling state of address and control lines bundles), D (state of data bus), MSYN and SSYN (states of master and slave synchronisation lines respectively). The approach proposed here exploits timed signal graph and significantly simplifies the description technique as well as protocol understanding itself. It also enables the designer to define the parallelism in the process specification. Such a graph for an ad hoc example is shown in Fig. 5. Vertices denoted with sub-indexes "m" and "s" imply the source entity - "master" and "slave" respectively. The time delays of 150ns and 75ns are brought by vertices $T(\geq 150)$ and $T(\geq 75)$ due to the UNIBUS requirement concerned with the compensation of the "skew" phenomenon.

Thus in a signal graph the vertex of type $T(K1 \leq, \leq K2)$ (pure time vertex) or $\delta xi(K1 \leq t(\delta xi) \leq K2)$ (with time attribute) fires not sooner than in K1 time units and not later than K2 time units (after it is enabled).

From the timed signal graph one can derive corresponding timed TD which however should contain some means for expressing time delays. That is why in such diagram beside the asterisk there is a symbol d. The corresponding value of time delay must be given in a supplement to the TD. For example, if after a change of signal xi from 0 to 1 there was a delay of 75ns then in those states which correspond to the newly set value xi=1 one should use as upper index a symbol 'd'. Among the immediate successors of the state containing $xi=1^d$ there must be a state differing from the given state only by having xi=1. Therefore in comparison with ordinary excitation of a variable which requires for semimodular diagram a compulsory firing of an excited variable, here one should have a new "timed" excitation which is not concerned with the switching of a variable but influences the allowance of the excitation of other variables. For the timed signal graph (Fig. 5) a corresponding timed TD is shown in Fig. 6.

The necessity of time delays d1 and d2 compensating signal skew is connected with the requirement that a synchronising signal (MSYN or SSYN) must be sensed by the recipient (slave or master) properly later than the completion of signal changes on the parallel bundle of wires. The satisfaction of this requirement helps to avoid reception errors. Since synchronisation and information lines may have quite different time parameters such compensating delays must be used. However specifying this protocol and taking into account the possible asynchrony of protocol signals we must look at this example with signal skew from another point of view. Strictly speaking we should model this protocol having in mind the following: the setting of parallel signals on the internal wires of master and the coresponding change of values of the outside bundle of lines that can be sensed by the slave must be specified separately, because when the master has information about the completion of setting of new bus values, it switches its synchronising signal MSYN only from its internal signals.

Such more elaborate consideration of transfer process causes us to specify the events of setting data or address on parallel buses in the master and out of it with two different signal graph vertices (see Fig. 7). For the sake of simplicity here used the denotation AC^{Δ} (instead of two AC^{+} and AC^{-}) to show by one vertex that address and control signals simply change their state without noticing that this change may be made through intermediate spacer state.

Moreover the vertices representing the outside events (AC_B^{Δ}, $D_B^{\pm}$, D_B^{-}) are obviously "hanging" vertices because each of these events, i.e. the completion of all necessary changes in the bundle of parallel wires can not be sensed by any other events. In fact, the slave transitions are occurred when indicating the master transitions only of wire signal MSYN. So abstracting from the exact vertex time values and accepting the time delays as unbounded but finite we are encouraged to establish that this protocol is incorrect because it is both non-semi modular (conflicts are possible) and unsafe ("hanging" vertices).

However we may conclude this protocol to be correct if we permit and hence provide in implementation the following time constraints between firing delays: $t(AC_B^{\Delta}) - t(MSYN^{+}) < 150ns$, $t(MSYN^{-}) - t(AC_B^{\Delta}) < 75ns$, $t(D_B^{\pm}(\text{or } D_B^{-})) - t(SSYN^{+}(\text{or } SSYN^{-})) < 75ns$.

With these constraints provided the timed signal

graph will be safe and will not have the conflicting states.

This example with interface led us to the necessity of discussion of some general aspects of temporal system analysis based on the properties of timed signal graph. It is worth noting what do we mean here by such analysis. Usually the Petri Net analysis is concerned with the procedure of investigation of some properties of the modelled system. It is well known that the key analysis problem is the reachability problem, the solvability of which is supposed to be proved. It is also known that the other analysis problems are transferable to the reachability problem. However when the question about Timed Petri Nets is arisen the correspodning problems are unsolvable for the general Timed Petri Net case. Not speaking much here about these problems (they are given at length in /2/) it can be stated that the investigation of self-timed properties of Muller circuits is also a significant analysis problem. This problem was under the study of D.E. Muller himself /6/ and some important results which gave the way to the analysis automation system based on the Muller model were issued in /15/.

The temporal behaviour analysis provides designer with the knowledge of presence of such undesirable effects as deadlocks, traps, hangups, conflicts, non-productive cycles (tempo-blocking) etc. The complexity of corresponding analysis procedures is often very high even for relatively small dimensions. This is because the analysis by whatever method used is always concerned with permutations of all possible system component time value combinations. In this sense the system analysis technique proposed here may be helpful from an idealogical point of view. This technique is based on a rather straightforward idea.

If the initial system model was a self-timed diagram or circuit which was semi modular then whatever time values the signal graph would be labelled by the system would operate correctly, i.e. undesirable effects would not be possible. That is why the following technique is allowable.

It is supposed that the system is initially represented by timed signal graph. Firstly, we build the transition diagram corresponding to this signal graph without account of the time values, i.e. as if this signal graph is self-timed. Secondly, if this diagram appears to be semi modular and hence correct in the self-timed sense we may deduce that the account of time delays is eliminated at all and the system is correct in its original timed sense.

Thirdly, if this diagram has some local violations of semi modularity and hence the system is incorrect in the self-timed sense then these local violation points must be analysed with account of the given time values. In fact, the introducing of time constraints on some system transitions restricts the set of its possible transition sequences (as well as permutations above mentioned). Thus if it appears that some of the sequences are not real due to time value relations then the local behaviour in violation points can be satisfactory in the timed sense. The following example illustrates this idea.

Example. The fragment of the timed signal graph is shown in Fig. 8a. Its variable changes are $x_1^+, x_1^-, x_2^+, x_3^-, x_4^+, x_4^-$. The corresponding time constraints are linked as attributes with all given vertices. The transition diagram fragment built in self-timed sense (see Fig. 8b) shows that there is a local violation of semi modularity in conflicting state 0100 when the completion of transition x_1^- is followed by the transition x_4^- but the previous transition x_4^+ is not yet completed, because of its "hanging" vertex. In order to validate this signal graph we must check the time constraints of two concurrent transitions: x_4^+ and x_4^- if the maximum time delay of x_4^+ is less than the minimum time delay of x_1^- i.e. the following relation holds:

$$K_2(x_4^+) < K_1(x_1^-).$$

6. Concluding remarks

The link between the Petri Nets and Muller's speed independent circuits (self-timed systems) is established through the bridge between persistent and safe Petri Nets, marked and signal graphs from one side and marking, transition diagrams can be represented by normal signal graphs in a more compact way. Signal graphs, hence, enable the designer of the discrete system to specify his timing diagrams in a more formal way. Timed signal graphs are introduced and their analysis technique with referring to time values, only if local self-timed correctness (semimodularity) is violated, is proposed. Therefore time values are a factor reducing the number of possible sequences and by that transforming the self-timed system to one which is dependent on the particular transition speeds. We hope that the approach proposed will give rise to significant reduction of temporal analysis complexity. However the formal proof of this fact is not yet obtained.

It should also be noted that the paper is written in an advertising manner and aims at drawing the readers attention to some research directions and problems rather than at giving the list of formal results in this interesting area.

References

1. Petri, K.A. Kommunikation mit Automaten - Technische Hochschule Darmstadt, 1962.

2. Peterson, J.L. Petri Net theory and the modelling of systems, New York, 1981.

3. Rosenblum, L. Ya. Petri Nets. Engineering Cybernetics, USSR Acad. Sci. 1983, No 5, 12-40.

4. Kotov, V.E. Petri Nets. Moscow:Nauka, 1984 (in Russian).

5. Muller, D.E., Bartkey, W.C. A theory of

asynchronous circuits. Annals of Computing Lab. of Harvard University 29 (1959), 204-243

6. Miller, R.E. Switching Theory, Vol. 2, New York:Wiley, 1965, Chapter 10 is a review of Muller's work on speed-independent circuits and includes a bibliography.

7. Varshavsky, V.I. et al. Aperiodic automata. Moscow:Nauka, 1976, (in Russian).

8. Keller, R.M. Vector replacement systems: a formalism for modelling asynchronous systems. TR 117 Princeton University, New Jersey, Dec. 1972.

9. Varshavsky, V.I., Rosenblum, L.Ya. et al. Asynchronous processes I, II Engineering Cybernetics, USSR Acad. Sci. 1980, No. 4, 137-142, No. 5, 138-143.

10. Bryant, R.E. Report on the Workshop on Self-timed Systems. Lab. for Comput. Sci. M.I.T., Cambridge, 1979.

11. Commoner, F. et al. Marked directed graphs, J. Comp. Syst. Sci. 5 (1971), No. 5, 511-523.

12. Birkhoff, G. Lattice Theory. Providence, Road Island, 1967.

13. Jump, J.R. Thiagarajan, P.S. On the interconnection of asynchronous control structures J. ACM 22 (1976), No. 4, 596-612.

14. Starodubtzev, N.A. Synthesis of control circuits for parallel computer systems. Leningrad:Nauka, 1984 (in Russian).

15. Varshavsky, V.I. et al. Analysis of asynchronous logical circuits I, II. Engineering Cybernetics USSR Acad Sci. 1982 No. 3, 137-149, No. 4, 84-97.

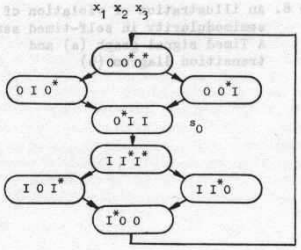


Figure 1. An elementary Muller diagram

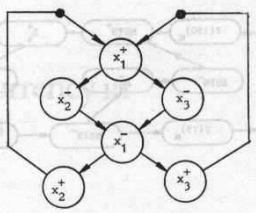


Figure 2. A signal graph corresponding to Muller Diagram in Figure 1

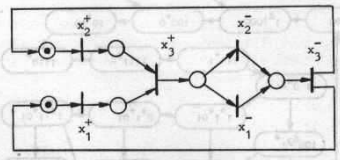


Figure 3. A Signal Petri Net

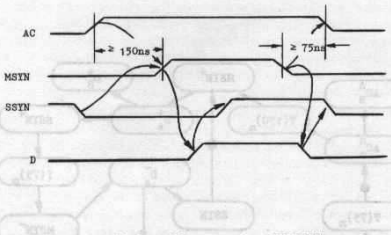


Figure 4. A timing diagram for UNIBUS read operation protocol

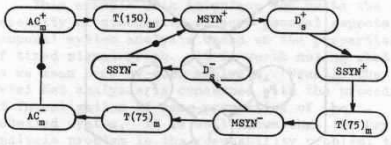


Figure 5. A timed signal graph for UNIBUS read protocol

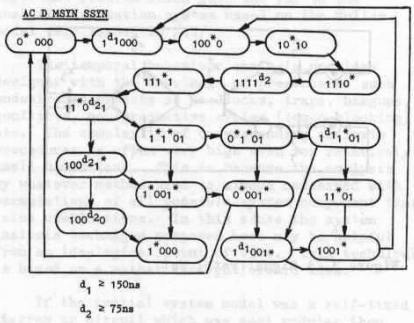


Figure 6. A timed transition diagram for UNIBUS read protocol

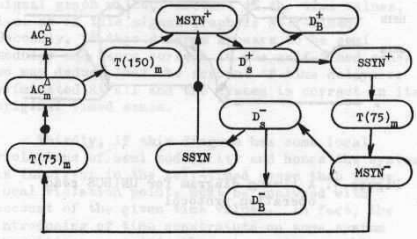
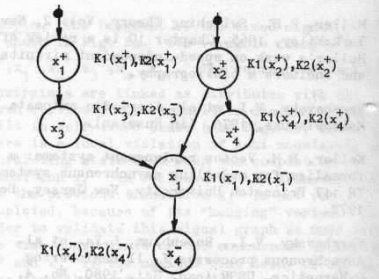
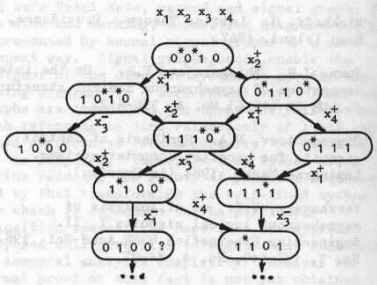


Figure 7. More adequate modelling of UNIBUS read protocol



a)



b)

Figure 8. An illustration of violation of semimodularity in self-timed sense A Timed signal graph (a) and transition diagram (b)