Department of Applied Analysis
and Computer Science

Research Report CS-73-07

April, 1973

PETRI NETS
AND
ASYNCHRONOUS CONTROL NETWORKS

by

M. Yoeli

INTRODUCTION

Bruno and Altman [18] have developed an interesting theory of asynchronous control structures of modular design. This theory has been further extended by Altman and Denning [19]. However, the exposition relies to some extent on engineering intuition, rather than on mathematically rigorous models. On the other hand, Patil [4] and Dennis [5] have demonstrated the applicability of Petri nets to the precise specification of asynchronous control structures. It thus becomes a challenging task to reformulate the work in [18], [19] in a mathematically rigorous way by means of Petri nets.

The second part of this Report (Sections IV-VI) is devoted to this task. A similar attempt has also been undertaken by Jump and Thiagarajan ([16],[17]), but our approach differs considerably from theirs.

The first part (Sections I-III) of this Report presents a unified introduction to Petri nets, covering two different versions, which appear in the literature. Emphasis is both on mathematically precise definitions as well as suitable system-oriented interpretations. This first part relies heavily on [3], [7], [9], [10] and uses freely material from these sources.

Throughout this report we have stated theorems precisely, but have omitted their proofs.

A comprehensive annotated bibliography on the theory and applications of Petri nets is given at the end of this Report.

This work was supported by the National Research Council of Canada, Grant No. A-1617.

I. BOOLEAN-TYPE PETRI NETS

Def.1.1 a) A *Petri Graph* is a system

$$P = \langle S,T,R\rangle$$

where S is a finite set of *places*

T is a finite set of *transitions*

and $R \subseteq (S \times T) \cup (T \times S)$.

The place s is an *input place* of $t \in T$, iff sRt and an *output place* of t, iff tRs.

b) A *Boolean-type Petri Net* (BPN) is an ordered pair $\langle P,m\rangle$ where

$P = \langle S,T,R\rangle$ is a Petri Graph and m is a *B-type marking* of P, i.e. a function $m:S \to \{0,1\}$.

In the system-oriented interpretation of a BPN, a place s corresponds to a certain condition which is either satisfied by the system [i.e., m(s) = 1] or does not hold [m(s) = 0]. A "fireable" transition, to be defined next (Def.1.2), will correspond to an event (change of conditions) which may occur in the system.

Petri nets are conveniently represented by means of a diagram as shown in Fig.1.1. The symbols used in BPN-diagrams are listed below:

s ○ = place s with marking m(s) = 0

s ⊙ = place s with *token*, i.e. m(s) = 1.

t □ = transition t

s ○→□ t = sRt

t □→○ s = tRs

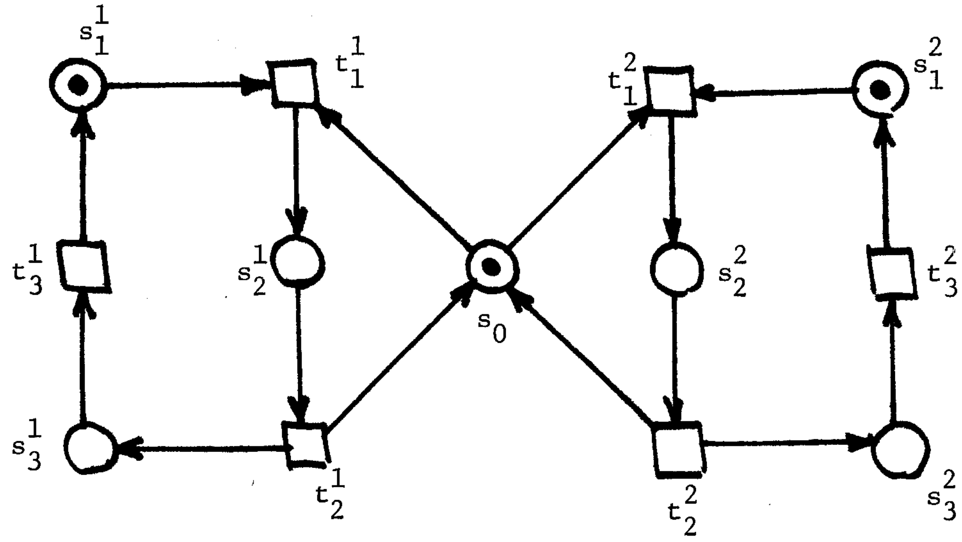


FIG.1.1 - Example of BPN <P,m> .

SYSTEM INTERPRETATION OF FIG.1.1

The Petri graph of Fig.1.1 represents e.g. a 1-server, 2-customer system, where the places correspond to system conditions as specified below ($i \in \{1,2\}$):

s_1^i - Customer in lane i requests service

s_0 - Server free

s_2^i - Customer in lane i is being served

s_3^i - Lane i is free.

The marking shown corresponds to a system state, in which the server is free and two customers request service simultaneously.

The transitions in this example may be interpreted as the following events:

t_1^i - The server starts serving the customer in lane i

t_2^i - The server finishes serving the customer in lane i and the customer leaves

t_3^i - A new customer enters lane i.

Note that the events represented by t_1^1 and t_1^2 may not occur simultaneously, since there is only one server. We shall refer to such a situation as a conflict (see below, Def.1.3).

Def.1.2 Let <P,m> be a BPN. The transition t is fireable (in <P,m>) iff

$$(\forall s \in S)[sRt \rightarrow m(s) = 1]$$

Thus in Fig.1.1, the fireable transitions are t_1^1 and t_1^2.

For a fixed Petri graph P we set

$$F_m \triangleq \{t \mid t \text{ is fireable in } <P,m>\} .$$

Def.1.3 Let <P,m> be a BPN and $V \subseteq F_m$. V is simultaneously fireable iff

$$(\forall s \in S)[|\{t \in V \mid sRt\}| \leq 1].$$

<P,m> is conflict-free iff F_m is simultaneously fireable.

In Fig.1.1, $F_m = \{t_1^1, t_1^2\}$ but F_m is not simultaneously fireable, i.e. the BPN is not conflict-free.

Def.1.4 Let <P,m> be a BPN, and let V be a set of transitions which is simultaneously fireable. We define $m*V$ to be the B-type marking n of P specified by:

$$n^{-1}(1) = [m^{-1}(1)-R^{-1}(V)] \cup R(V)$$

where $R(V) \underline{\underline{\Delta}} \{s|(\exists t \in V)tRs\}$

$R^{-1}(V) \underline{\underline{\Delta}} \{s|(\exists t \in V)sRt\}$

and $m^{-1}(1) \underline{\underline{\Delta}} \{s|m(s) = 1\}$

We shall say that the net <P,n> is obtained from <P,m> by firing V.

If $V = \{t\}$, then $R(V)$ is the set of output places of t and $R^{-1}(V)$ is the set of its input places.

Referring again to Fig.1.1 and writing $m*t$ for $m*\{t\}$, we have e.g.

$$m*t_1^1 = m_1\text{, where } m_1^{-1}(1) = \{s_1^2,s_2^1\}.$$

Now $F_{m_1} = \{t_2^1\}$ and $m_1*t_2^1 = m_2$, where $m_2^{-1}(1) = \{s_1^2,s_0,s_3^1\}$ and $F_{m_2} = \{t_3^1,t_1^2\}$. If we now assume that only t_3^1 fires, then the system returns to the state represented by m, since $m_2*t_3^1 = m$.

II. INTEGER-TYPE PETRI NETS

Let N denote the set of non-negative integers.

Def.2.1 An Integer-type Petri Net (IPN) is an ordered pair $<P,m>$ where P is a Petri graph and m is an I-type marking, i.e. $m:S \to N$.

Example 2.1

If the place $\underset{s_o}{\odot}$ in Fig.1.1 is replaced by $\underset{s_o}{\textcircled{..}}$, i.e. $m(s_o) = 2$, we obtain an IPN which may be interpreted as a 2-server, 2-customer system. The modified firing rules are specified in the following definitions.

Def.2.2 Let $<P,m>$ be an IPN. The transition t is fireable (in $<P,m>$) iff $(\forall s \in S)\ [sRt \to m(s) > 0]$.

Def.2.3 Let $<P,m>$ be an IPN and V a set of transitions of P. V is simultaneously fireable (in $<P,m>$) iff

$$(\forall s \in S)\ [m(s) \geq |\{t | t \in V \wedge sRt\}|]$$

We again denote by F_m the set of all fireable transitions in $<P,m>$ and call $<P,m>$ conflict-free iff F_m is simultaneously fireable.

Def.2.4 Let $<P,m>$ be an IPN, and let V be a set of transitions which is simultaneously fireable. We define $m*V$ to be the I-type marking n of P specified by:

$$(\forall s \in S)[n(s) = m(s)-|\{t \in V|sRt\}| + |\{t \in V|tRs\}|]$$

The firing of a single transition consists in decreasing the marking of its input places by 1 and increasing the marking of its output places by 1.

Hence the marking m*V may be obtained by firing each transition in V separately.

The IPN of Example 2.1 is conflict-free and we have:

$F_m = \{t_1^1, t_1^2\}$

$m*F_m = m_1$, where $m_1(s_2^1) = m_1(s_2^2) = 1$ and $m_1(s) = 0$ for all other places.

If we set $m_2 = m_1 * F_{m_1}$ and $m_3 = m_2 * F_{m_2}$, then $m_3 = m$.

The following two definitions apply to both BPN's and IPN's.

Def.2.5 Let <P,m> be a Petri net, and σ a sequence of markings

$m = m_0, m_1, \ldots, m_k \quad k \geq 0.$

σ is a single-firing sequence for <P,m> iff either k = 0 or there exist transitions t_i, $1 \leq i \leq k$, such that $m_i = m_{i-1} * t_i$. The outcome of σ is the marking m_k.

σ is a total-firing sequence for <P,m> iff $<P,m_i>$ is conflict-free for $0 \leq i \leq k-1$ and $m_{i+1} = m_i * F_{m_i}$.

The total-firing sequence σ terminates iff $F_{m_k} = \phi$. We set $\hat{\sigma} = \bigcup_{i=0}^{k-1} F_{m_i}$.

Def.2.6 A Petri net <P,m> is live, iff for every single-firing sequence σ for <P,m>, with outcome m' and every transition t of P, there exists a single-firing sequence σ' for <P,m'> with outcome m" such that $t \in F_{m''}$.

Def.2.7 An IPN <P,m> is k-safe, k ≥ 1, iff for every single-firing sequence σ for <P,m> with outcome m', and every place s of P, $m'(s) \leq k$.

III. MARKED GRAPHS

In this section we summarize some of the results on marked graphs from Ref.[7].

Def.3.1 A Marked Graph (MG) is an IPN <P,m> in which all places have indegree 1 and outdegree 1.

Def.3.2 Let <P,m> be an MG, and C a directed circuit of P. The token count m/C is defined by:

$$m/C \triangleq \sum_{s \in C} m(s)$$

Lemma 3.1 The token count of a directed circuit in an MG does not change by firing.

Theorem 3.1 An MG <P,m> is live iff the token count of every directed circuit is positive.

Theorem 3.2 A live MG is 1-safe iff every place is in a directed circuit with token count 1.

IV. CONTROL NETS

Def.4.1 A Control Net is a triple C = <P,m,L>, where

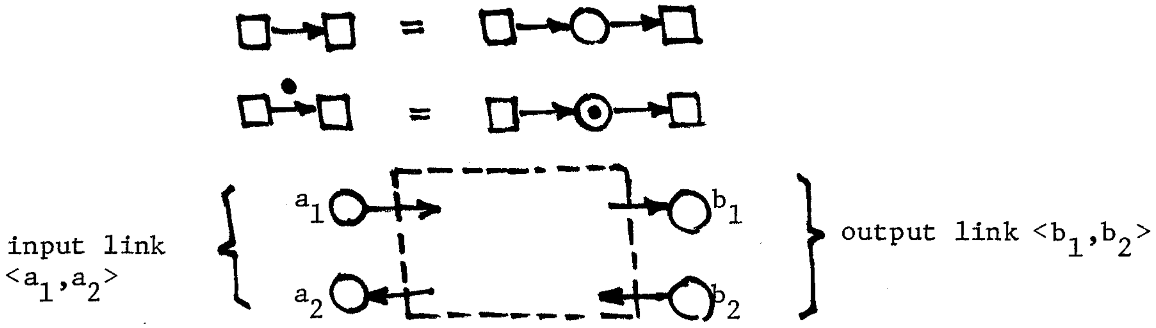
1) <P,m> is a BPN in which all places have indegree ≤ 1 and outdegree ≤ 1. A place with indegree 0 is an input terminal of C, and a place with outdegree 0 is an output terminal of C. An input (output) link of C is an ordered pair, the first component of which is an input (output) terminal and the second component an output (input) terminal.

2) L is a set of links of C, such that every terminal of C belongs to exactly one link in L.

3) $(\forall s \in S)$ [s is a terminal of C → m(s) = 0].

Examples of Control Nets

The following additional symbols will be used in diagrams representing control nets.



Figures 4.1-4.5 show some basic control nets (control modules), which may be used to form composite control nets, as discussed later.

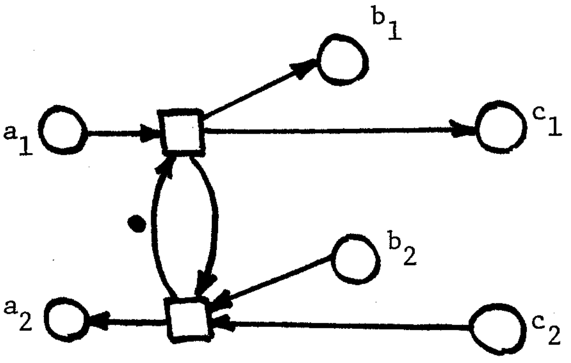


Fig.4.1 - WYE(W) - module

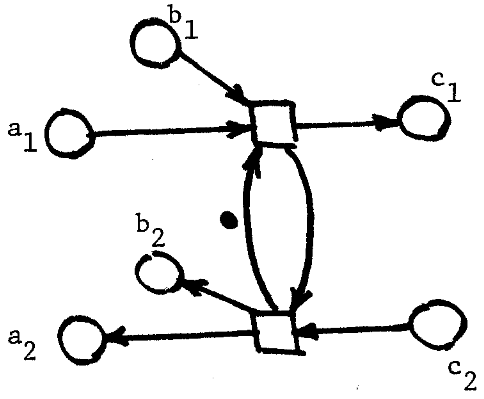


Fig.4.2 - JUNCTION(J) - module

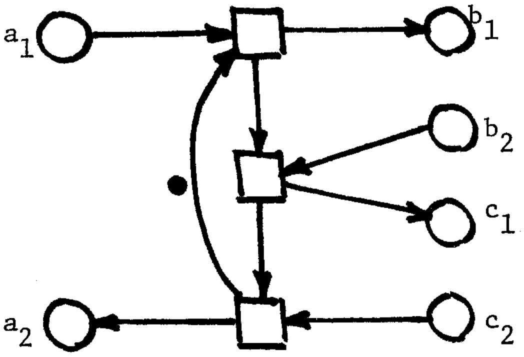


Fig.4.3 - SEQUENCE (S) - module

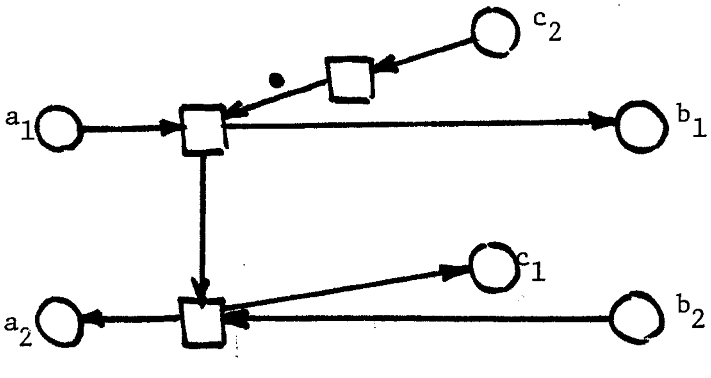


Fig.4.4 - TRIGGER (T) - module

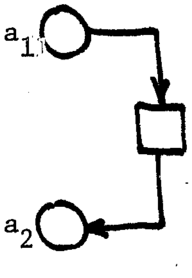


Fig.4.5 - SINK - module

Interpretation of Basic Control Nets

The SINK-module (Fig.4.5) may be considered to represent the control part of an operational unit, functioning asynchronously. Upon receipt of a START-signal (this corresponds to placing a token in a_1) the unit performs a well-defined single task (the transition fires). It indicates completion of the task by returning a DONE-signal (this corresponds to the appearance of a token in a_2).

Consider now e.g. the S-module (Fig.4.3), and assume that SINK-modules are connected to output links b and c, as shown in Fig.4.6.

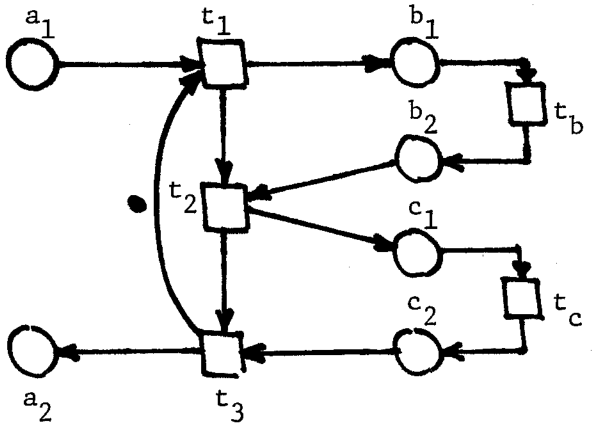


Fig.4.6 - S-module with output link terminations

This control net represents a system with two operational units which are to be activated sequentially.

If the system receives a START-signal (i.e. a token is placed in a_1), the events represented by the following transitions will occur in the order indicated:

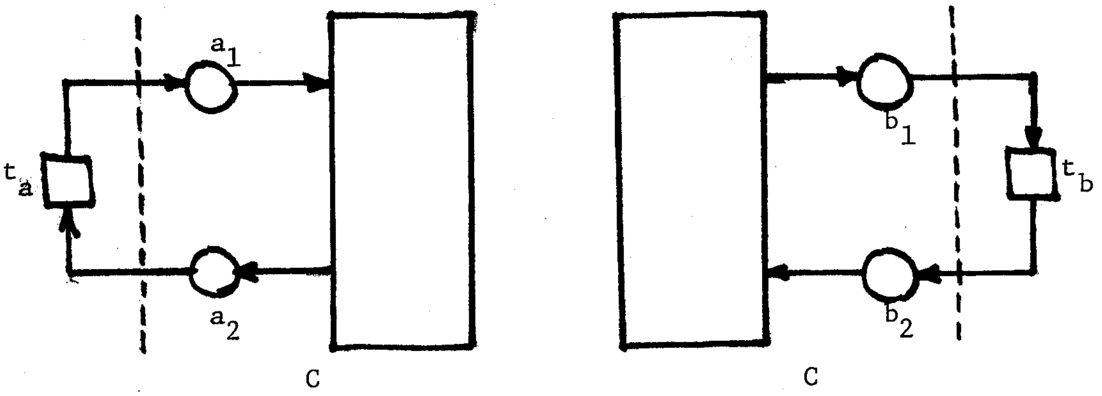
$t_1, t_b, t_2, t_c, t_3.$

Thus the second operational unit (t_c) will be activated only after the first unit (t_b) has issued a DONE-signal. After the second unit has completed its task, the system issues a DONE-signal (i.e. a token appears in a_2).

If an attempt is made to restart the system before it has issued the DONE-signal, this attempt will remain ineffective (since t_1 becomes refireable only after t_3 has fired).

If this and similar precautionary measures are not required, the internal connections between transitions t_1, t_2, t_3 in Fig.4.6 may be omitted.

Def.4.2 Let C be a control net, and $a = \langle a_1, a_2 \rangle$ one of its input links. By *terminating* link a we mean the connection of an *outside transition* t_a as shown in Fig.4.7(a). The *termination* of an output link $b = \langle b_1, b_2 \rangle$ is similarly defined (see Fig.4.7(b)).



(a) Termination of Input Link (b) Termination of Output Link

Fig.4.7 - Link Terminations

Def.4.3 Let C be a control net and $a = \langle a_1, a_2 \rangle$ one of its input links. By starting input link a we mean the placing of a token in its input terminal a_1.

Def.4.4 A control net contains a deadlock iff the BPN obtained by starting all its input links, and terminating all its input and output links is not live.

One easily verifies the following (cf. Thm.3.1).

Theorem 4.1 If a control net contains a directed circuit with token count 0, then it contains a deadlock.

An example of a deadlocked control net, obtained by "cascade-connecting" an S-module (Fig.4.3) and a J-module (Fig.4.2) is shown in Fig.4.8.

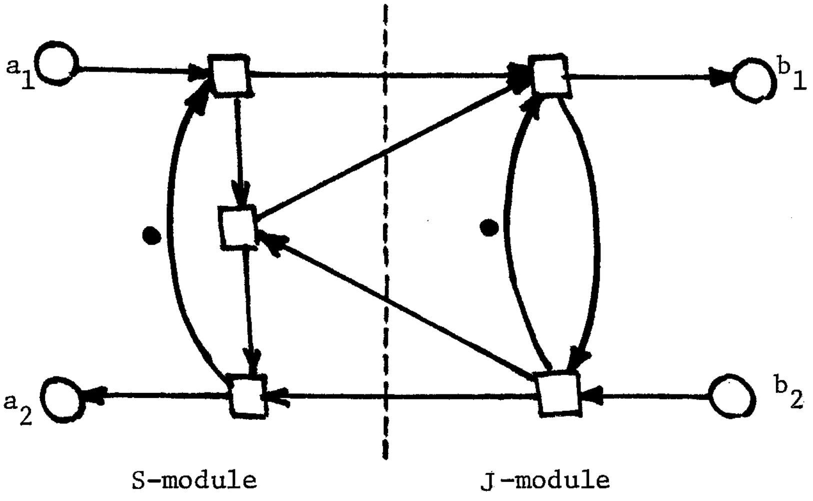


Fig.4.8 - Example of Control Net with Deadlock

On the other hand, none of the basic control nets (Figs.4.1-4.5) contain deadlocks.

V. COMPOSITION OF CONTROL NETS

Def.5.1 Let C_1 and C_2 be control nets. Let $\beta = \langle b^1,\ldots,b^k\rangle$ $(k \geq 0)$ be a sequence of different output links of C_1 and $\alpha = \langle a^1,\ldots,a^k\rangle$ a sequence of different input links of C_2. The β/α-composition C of C_1 and C_2 is the control net obtained from C_1 and C_2 by identifying links b^i and a^i for all i, $1 \leq i \leq k$, as indicated in Fig.5.1.

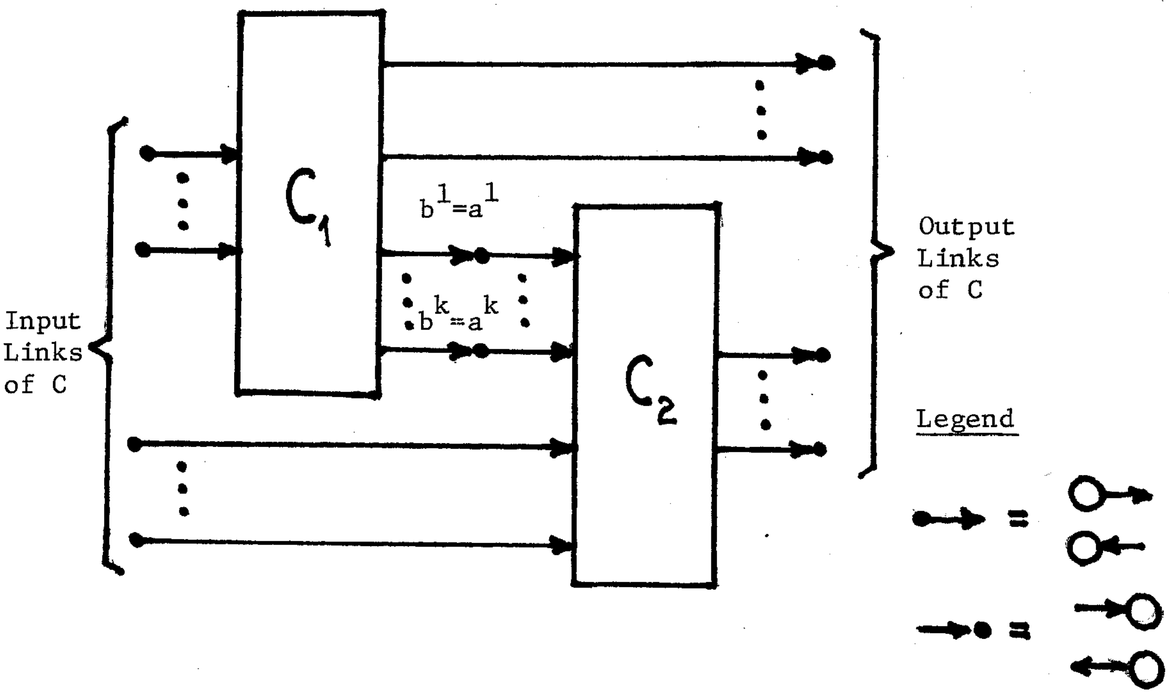
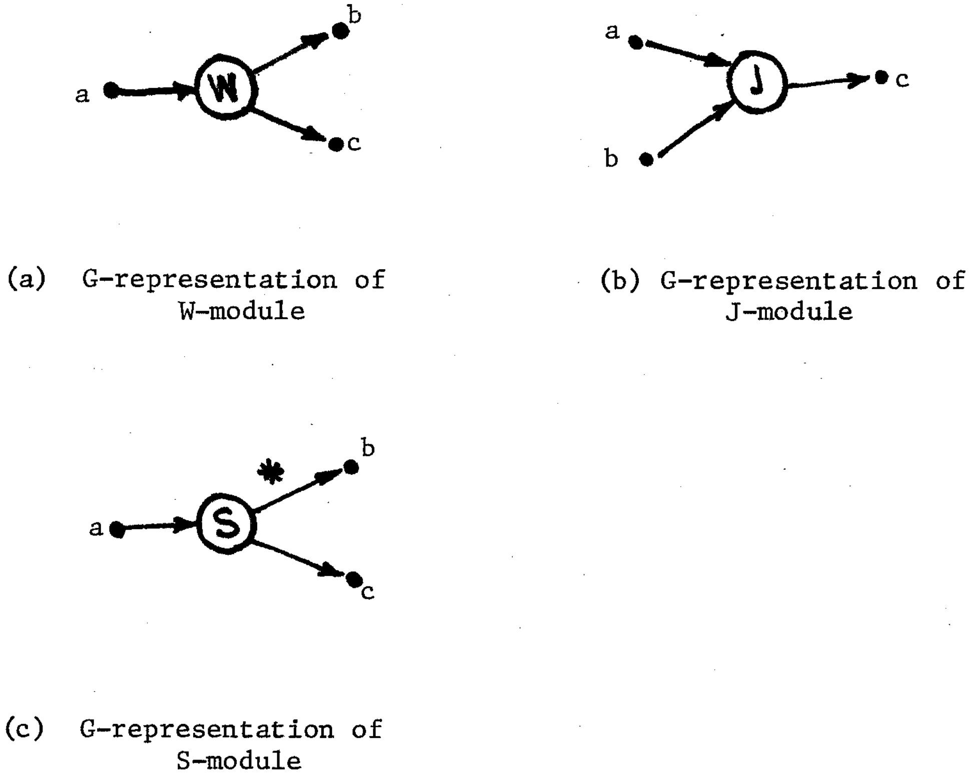


Fig.5.1 - Composition of Control Nets

In the sequel we shall be concerned with compositions of the basic control modules W, J, and S. In order to simplify the representation of such composite control nets, we use the G-representation of these modules, as shown in Fig.5.2.



(a) G-representation of W-module

(b) G-representation of J-module

(c) G-representation of S-module

Fig.5.2 - G-Representations of W-, J-, and S-Modules

In this G-representation a link is represented by a single dot (•). An arrow pointing towards the module indicates an input link, an arrow pointing away from the module indicates an output link.

The star (*) in Fig.5.2(c) is used to indicate the output link of the S-module which is to be activated first (primary output link).

Fig.5.3 shows the G-representation of the control net of Fig.4.8.

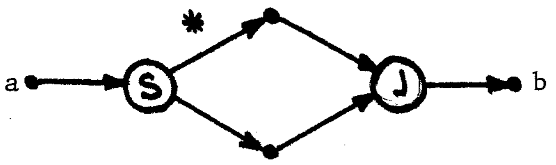


Fig.5.3 - G-representation for Fig.4.8

Def.5.2 A WSJ-net is a control net obtained from W-, S-, and J-modules by (repeated) composition.

For WSJ-nets the converse of Theorem 4.1 also holds:

Theorem 5.1 A WSJ-net contains a deadlock iff it contains a directed circuit with token count 0.

The following Theorem 5.2 is a mathematically precise formulation of a result first stated in [19].

Theorem 5.2 A WSJ-net C contains a deadlock iff there exists an S-module S and an output link b of C such that the G-representation of C contains a directed path from S via its primary output link to b, as well as a directed path from S via its secondary output link to b.

VI. EQUIVALENCE OF CONTROL NETS

Def.6.1 Let C be a Control Net, A a set of input links of C, and B a set of output links of C. An A/B experiment on C consists of terminating all output links in B and starting all input links in A. The experiment terminates if there exists a terminating total-firing sequence σ with outcome m' for the Petri Net $<P_B,m_A>$ where P_B is the corresponding extension of P, and m_A the extension of m.

Assume now that the A/B-experiment on C terminates, that the corresponding firing sequence is σ, and that the outcome of σ is m'. The outcome of this experiment is the ordered pair <A',B'> where $A' \underline{\underline{\Delta}} \{a|a$ is an input link of C and $m'(a_2) = 1\}$ and $B' = \{b \in B|t_b \in \hat{\sigma}\}$

Def.6.2 A control net C is well-formed iff

1) C contains no deadlock

2) Every A/B-experiment terminates and its outcome <A',B'> satisfies the condition $A' \subseteq A$.

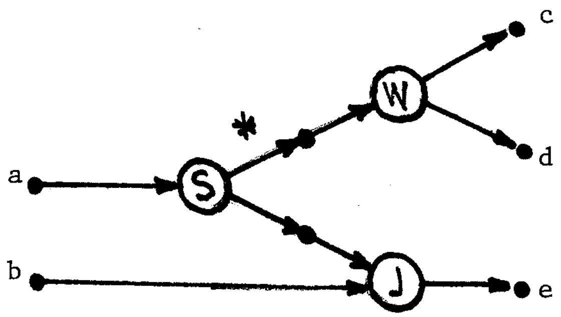
3) If A is the set of all input links of C, and B the set of all its output links, then the outcome of this A/B-experiment is <A,B>.

For WSJ-nets we have the following result.

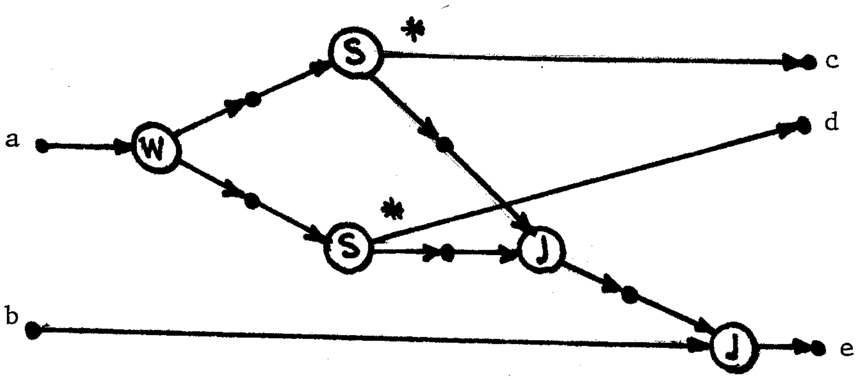
Theorem 6.1 A WSJ-net is well-formed iff it contains no deadlock.

Def.6.3 Let C and C' be well-formed control nets. C and C' are equivalent iff there exists a 1-1 correspondence between their input links as well as a 1-1 correspondence between their output links such that corresponding experiments always yield corresponding outcomes.

Fig.6.1 illustrates Def.6.3. Corresponding links of C and C' have the same label.



(a) a well-formed control net C



(b) a well-formed control net C' equivalent to C

Fig.6.1 - Equivalent well-formed Control Nets

Def.6.4 Let C be a WSJ-net and B its set of output links. We define a precedence relation < on B as follows:

b < b' iff there exists an S-module S such that the G-representation of C contains a directed path from S via its primary output link to b, and a directed path from S via its secondary output link to b'.

Referring to Fig.6.1, we have for both the control nets C and C':

$< \; = \{\langle c,e\rangle,\langle d,e\rangle\}$

Def.6.5 Let C be a control net, A the set of its input links, and B the set of its output links. We define an i/o-relation $\rho \subseteq A \times B$ as follows:

$a\rho b$ iff the G-representation of C contains a directed path from a to b.

Referring again to Fig.6.1, we have for both C and C':

$\rho = \{\langle a,c\rangle,\langle a,d\rangle,\langle a,e\rangle,\langle b,e\rangle\}$

The following theorems are precise formulations of results stated in [18], [19].

Theorem 6.2 Let C and C' be well-formed WSJ-nets. C and C' are equivalent iff their links can be relabeled such that their precedence relations coincide, as well as their i/o relations.

Theorem 6.3 Let C be a well-formed WSJ-net. There exists an equivalent well-formed WSJ-net C' which is the composition of control nets C_W, C_S, C_J (in this order; see Fig.6.1(b) for an example), where C_W contains only W-modules, or is a trivial through-connection, and similarly for C_S and C_J.

ANNOTATED BIBLIOGRAPHY

A. Petri Nets

[1] C.A. Petri, Kommunikation mit Automaten, Institut für Instrumentelle Mathematik, Universität Bonn, Hft.2, Bonn, 1962.
This is the first paper on the subject. Now out-of-print.

[2] A.W. Holt et al., "Information System Theory Project", Applied Data Research, Inc., Princeton, N.J., 1968. AD676972.
The first comprehensive U.S. account of Petri nets.

[3] A.W. Holt and F. Commoner, "Events and Conditions", in: Record of the Project MAC Conference on Concurrent Systems and Parallel Computations (June 1970, Woods Hole, Mass.) ACM, New York, 1970, pp.3-52.
An extensive discussion of restricted types of Petri nets (marked graphs and transition diagrams).

[4] S.S. Patil, "Macro-modular Design of Asynchronous Circuits", Memo MAC-M-414, Project MAC, M.I.T., May 1969.
An application of Petri nets to the specification and design of modular asychronous control structures.

[5] J.B. Dennis, "Modular, Asynchronous Control Structures for a High Performance Processor" in: Record etc. (see [3]), pp.55-80.
This paper specifies various basic control modules by means of Petri nets and applies them to the synthesis of a particular asynchronous control network.

[6] H.J. Genrich, "Einfache Nicht-Sequentielle Prozesse" GMD (Gesellschaft für Mathematik und Datenverarbeitung), Bonn, Bericht Nr.37, 1971.
Heavily formalized study of marked graphs. Not recommended as introductory text.

[7] F. Commoner, A.W. Holt, S. Even, A. Pnueli, "Marked Directed Graphs", J. Comp. Syst. Sc. 5, pp.511-523, 1971.
Strongly recommended introduction to marked graphs. Summarizes in easily readable form some of the results of [3] and [6].

[8] R.M. Shapiro and H. Saint, "A New Approach to Optimization of Sequencing Decisions", in: Ann. Rev. in Automatic Programming, 6, Pt.5, pp.257-288, 1970.
This paper deals with the problem of generating efficient programs for computers with parallel operation capabilities, from algorithms expressed e.g. in FORTRAN. The first step is the production of a "desequenced" version of the algorithm, expressed as a Petri net. Next, the functioning of the target processor is represented by a Petri net. Finally, these two nets are suitably "combined".

[9] H.J. Genrich und K. Lautenbach, "Synchronisationsgraphen", GMD-I5 Bericht, 1971. To be published.
An interesting application of linear network algebra and linear programming to the study of marked graphs. Both mathematically precise and easily readable (if you read German).

[10] D. Tsichritzis, "Modular System Description", T.R. No.33, Dept. of C.S., Univ. of Toronto, 1971. Studies decompositions of large Petri nets into smaller modules, as efficient tool to describe Operating Systems.

[11] D. Tsichritzis, "Topics in Operating Systems Revisited", T.R. No.38, Dept. of C.S., Univ. of Toronto, 1972.
An essay on Petri nets and Operating Systems.

[12] D. Tsichritzis, Lecture Notes on Operating Systems, T.R. No.44, Dept. of C.S., Univ. of Toronto, 1972.
Contains a discussion of the applicability of Petri nets to Operating Systems.

[13] G. Tourlakis, Petri Nets, Internal Report, Dept. of C.S., Univ. of Toronto.
Petri nets (theory and applications) made easy, with an Appendix for the Mathematician.

[14] P.A. Bernstein, Description Problems in the Modeling of Asynchronous Computer Systems T.R. No.48, Dept. of C.S., Univ. of Toronto, 1973.
A comparative survey of models for parallel systems, including Petri nets.

[15] R.E. Miller, "Some Theoretical Studies of Parallel Computation", in: Proc. Internat. Computing Symp., Venice, April 12-14, 1972.
A comparison of theoretical models for parallel computation, including Petri nets.

[16] J.R. Jump and P.S. Thiagarajan, "On the Equivalence of Asynchronous Control Structures", 13-th Annual Switching and Automata Theory Symp., Oct.1972, pp.212-223.
This and the following paper is an interesting attempt to apply available results on marked graphs to a mathematical theory of asynchronous control networks.

[17] J.R. Jump and P.S. Thiagarajan, "On the Interconnection of Asynchronous Control Structures",to be published.

B. Asynchronous Control Networks

[18] J. Bruno and S.M. Altman, "A Theory of Asynchronous Control Networks", IEEE Trans. Comp., C-20, No.6, June 1971, pp.629-638.

[19] S.M. Altman and P.J. Denning, "Decompositions of Control Networks" in: Record etc. (see [3]), pp.81-92.