

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Computation Structures Group Memo 98

Proposed Research on
Concurrent Systems

Part III of a Proposal to NSF
for Further Research in Computation Structures

by

Suhas S. Patil

February 1974

PART III CONCURRENT SYSTEMS

Concurrency, or interacting parallel activity, arises in systems when there are independent but interacting parts in the system. The study of concurrency is important because it manifests itself in both hardware and software systems, especially when different parts of the system must coordinate some activity or communicate with each other. We have studied concurrency by studying the problems of description and realization of modular systems [35, 47] and arbiters [45,48], and by studying properties of Petri nets. We have found transformations of Petri nets into asynchronous circuit realizations [46], transformations that preserve Petri net behavior [4], and means for analyzing throughput of asynchronous systems [50]. We have studied the requirements for preserving liveness under interconnection of system parts and have shown that many open decision problems concerning liveness and reachability are reducible to one another [23]. Our reducibility results put us in excellent position to attack the main open problem of liveness and reachability for Petri nets and Vector Addition Systems, which is whether liveness and reachability are decidable properties of systems. Solving these problems for either the Vector Addition System or Petri nets will be enough because we have shown that these two systems are reducible to each other [23]. Besides pursuing decidability questions of Petri nets we would like to find behavior preserving transformations on nets that eliminate non-promptness and we would like to complete the theory of persistent Petri nets.

There had always been a question about how the basic arbiter in an asynchronous system can be physically realized. We have found interesting structures for arbiters which are of both theoretical and practical interest. This has led us to some interesting questions concerning realizability and response time

of digital systems, which we propose to study along with the study of Petri nets.

A. PETRI NETS

In the formal representation and analysis of concurrent systems the Petri nets of Holt [24, 25, 49] have been very useful. Problems of liveness and deadlock [13, 21, 25], description and realization of digital systems [14, 35, 40, 42, 47, 52], flow of control and semaphores [32, 41, 44], throughput of computation [50], equivalence and canonical forms for control [4, 8, 27] and vector addition systems [23, 28, 29, 31] have found precise representation in Petri nets, and the analysis of Petri nets and their subclasses has provided useful results and deep insight into these problems. Other related graph representations of concurrent systems have also been studied specifically with regard to proper termination of computation [2, 3, 20, 26, 28].

The usefulness of Petri nets in the study of concurrent systems comes from their simple structure which permits analytic treatment and yet allows precise representation of the pertinent aspects of the systems. The graphical form of Petri nets contributes immensely towards providing an intuitive perception of the problem.

A1. DECIDABILITY QUESTIONS FOR PETRI NETS

Ever since M. Rabin's proof showing that the question whether the reachability set of one Vector Addition System is a subset of another Vector Addition System is undecidable, there has been an interest in other decidability questions about Vector Addition Systems and Petri nets.

R. Keller has pointed out that Petri nets without self-loops correspond to Vector Addition Systems whose Period Vectors have coordinates restricted to the set $\{-1, 0, +1\}$, whereas a form of Petri nets generalized to permit multiple-

arcs between places and transitions instead of just single arcs can be used as a graphical representation for his Vector Replacement System [31]. What we have shown is that there is a complete isomorphism between generalized Petri nets and ordinary Petri nets, and thus any question about Vector Addition Systems or Vector Replacement Systems can be translated into an equivalent question about either ordinary or generalized Petri nets [23].

In Petri net terms, Rabin's unsolvable problem becomes: Given two Petri nets with the same number of places, is the Marking Class of one a subset of the Marking Class of the other?

Using Petri net constructions, we have been able to present a simplified version of Rabin's proof. This proof, which is presented in the accompanying Attachment C is representative of our methodology.

Some open decidability questions about Petri nets, and therefore about Vector Addition Systems, are the following.

RP - Reachability Problem: Is a particular marking reachable by a net?

LP - Liveness Problem: Is a given Petri net live?

EM - Equality of Marking: Do two nets have equal marking classes?

CM - Common Marking Problem: Is the intersection of the marking classes of two nets non-empty?

We have shown that the reachability problem is reducible to the liveness problem, and we believe the converse is also true. As steps toward proving the latter conjecture, we have shown that the reachability problem, the Zero Reachability Problem (ZRP-reachability of zero marking), and the Sub-reachability Problem (SRP-reachability of a submarking) are all reducible to each other, and that the liveness problem and the sub-liveness problem (SLP-liveness of a given transition in a Petri net) are reducible to each other. The remaining step to be completed is showing that some form of the liveness problem is

reducible to a reachability question, thus making all of these decision problems reducible to each other.

We propose to find answers to the decidability questions by:

- Developing Petri net constructions that permit us to relate the various problems to each other and establish whatever recursive reducibilities exist among these problems.
- We know some properties which if true for Petri nets will make the Reachability and Liveness problems undecidable and if some other properties hold we know these problems will be decidable. We propose to investigate these properties further in hope of answering whether the Reachability Problem, Liveness Problem, Equality of Marking Problem and the Common Marking Problem are decidable.

A2. PROMPT NETS

If we identify certain transitions of a Petri net as input/output transitions and other transitions as internal transitions, then the net is said to be prompt if there can be at most a bounded number of firings of the internal transitions between successive firings of input/output transitions. Thus if a Petri net is viewed as a black box and the firing of any of the input/output transitions is viewed as interaction with the outside world, promptness will guarantee that there will be no more than a bounded number of events inside the box between successive interactions with the outside world. Nets which respond to inputs within a fixed number of firings of internal transitions are a special kind of prompt net.

Since the lack of promptness implies uncertainty about the response time and the resources consumed by the system represented by the net, transformations

of nets which preserve the external functional behavior of the net but eliminate the non-promptness are of great interest. The need to transform non-prompt nets into prompt nets also arises in the process of reducing one net to another to show their equivalence.

The nature of the non-promptness of nets can be characterized by the type of nets that result when the input/output transitions of a net are deleted together with their associated arcs; if the resulting nets are state machines, then we say that the net has non-promptness of the state machine type. Thus we can have non-promptness of marked graph type, free choice net type, state machine decomposable net type and of the types of other classes of subnets we know. We have found transformations that eliminate the state machine type of non-promptness which involve collapsing all the places of the state machine into one place with number of tokens equal to the sum of tokens in all places of the state machine [Appendix]. The class of nets we propose to study includes non-safe nets, i.e. nets which may have multiple tokens in the places.

- We propose to find answers to the open question of whether all Petri nets can be transformed into functionally equivalent prompt nets, and we propose to find transformations to eliminate unpromptness of the other types such as marked graph type, free choice type and state machine decomposable type if such transformations exist.

A3. PERSISTENT NETS

The theory of marked graphs [4, 13, 19, 21, 25, 27] is now well understood. Marked graphs are a subclass of the class of persistent Petri nets. A persistent net is one in which an enabled transition is disabled only as the result of it being fired. We propose to study persistent nets to obtain the

marked graph-like results to cover the persistent nets. In particular we propose the following.

- We propose to study the liveness and deadlock question for persistent nets.
- Transformation of non-safe persistent nets into equivalent marked graphs.

B. DIGITAL SYSTEMS

Faithful realization of a system description as a physical system is an important problem. Digital systems are realized either as asynchronous or as synchronous systems. Work on speed-independent circuits [16, 30, 35, 38, 39, 46], and modular realization of digital systems [5, 6, 7, 9, 12, 14, 15, 17, 18, 36, 37, 42, 45, 46, 47] is an attempt to realize correctly functioning asynchronous systems. We have investigated the asynchronous modular realization of Petri nets [18, 42, 46, 47] and parallel schemata [15, 47]. In the physical realization of many systems one requires an arbiter to control access to resources or resolve conflicts. In such systems the correctness of the system critically depends on the correct functioning of arbiters. Synchronous systems use both arbiters and synchronizers. Practical and theoretical evidence indicates that arbitration cannot be always performed in a fixed amount of time [10, 11, 33]. We have shown that synchronizers and arbiters are very closely related; one can realize a synchronizer from a bounded delay arbiter and a bounded delay arbiter from a synchronizer. We have also shown that a synchronizer which operates in a fixed number of clock periods can be realized from a synchronizer which takes an unbounded number of clock periods under critical operation [1, 48] (Attachment D).

If the assertion that arbitration cannot be done in a fixed length of time is true, the synchronizers and arbiters used in synchronous systems will have a finite probability of failure. The schemes for reducing this probability to an acceptable low value then becomes very important. We have investigated iterative structures for improving both synchronizers and arbiters. The iterative structures achieve low probability of error from the fact that the probability of the failure of a two-stage device is equal to the product of the probability of the failure of the individual stages. We have also found a new principle for realization of asynchronous arbiters which enabled us to obtain what appears to be the smallest physical realization of an asynchronous arbiter. This realization requires only a single flip-flop and two threshold elements, which is considerably smaller than the asynchronous arbiter circuits reported so far.

Our work on synchronizers and arbiters has led us to the hypothesis that if we accept the premise that arbitration cannot be done in a bounded length of time, then no non-functional computation can be physically realized in bounded length of time. At the same time we hypothesize that all functional computations that involve only a bounded number of steps can always be physically realized in asynchronous systems which take only a bounded length of time to perform the computation. Surprisingly the functional system cannot be correctly realized as a synchronous system because the synchronization problem makes it impossible for one to be assured that the data sent to a synchronous system will be correctly received by the system. Many times a functional computation is realized in a system which is internally non-functional even though at the output terminal it is functional. Such a system may take an unbounded length of time to respond to inputs because of the internal non-functionality. We have examples which show that for some class of system realizations if one is

provided with many copies of such systems one is able to construct a bounded delay system out of them.

- We propose to fully explore the philosophical questions raised above in the hope of gaining insight into the fundamentals of physical realizability of systems.
- We propose to examine other interesting structures for realization of synchronizers and arbiters and study multi-input arbiters and the synthesis of multiple arbiter systems.

We have used a digital laboratory for our work on arbiters and asynchronous digital circuits. We built and tested iterative arbiters and multi-input arbiters in this laboratory. The laboratory has also provided many undergraduate students with interesting projects on asynchronous digital systems and has been of great educational benefit to them. We would like continued use of the laboratory and request \$2,000 per year for components and materials needed in the laboratory.

Appendix

Transformation of Non-Prompt Nets into Prompt Nets

We shall again state what promptness means: A system is prompt if it performs at most a bounded number of internal transactions between any successive input/output transactions. The lack of promptness means that the system may make an unbounded number of internal transactions between two input/output transactions. As an illustration of a non-prompt system consider the system consisting of the four cyclic processes handling semaphores shown in Figure 1. In this system, the input is received through semaphore X and output is sent through Y and Z, and therefore all semaphore primitives that operate on these semaphores will be regarded as performing input/output transactions. The semaphores L and R are internal semaphores. Basically, the action of the processes is to increment either semaphore Y or semaphore Z for each increment made to semaphore X. If incrementing a semaphore is regarded as giving a token of information and decrementing as taking a token away, we can describe the actions of the four processes as follows: Process P_1 takes the token of information from X and sends it to process P_2 whereupon the process P_2 juggles the token between semaphore variables L and R just the way a juggler might juggle a ball between the left and right hands. Process P_3 can take the token only when it is in semaphore L and process P_4 only when it is in semaphore R. The system is non-prompt because the process P_2 may juggle the token an unbounded number of times before either process P_3 or P_4 gets it.

The Petri net of Figure 2a presents a schematic representation of the above system where the semaphores are represented by places and the instructions by transitions. Transitions t_a , t_d and t_e are input/output transitions while transitions t_b and t_c are internal transitions. Because transitions t_b and t_c may fire an indefinite number of times between the firings of t_a , t_d and t_e ,

this net is not prompt. But this net can be transformed into a prompt net by collapsing the subnet consisting of places L and R and the transitions t_b and t_c into one place S. The resulting prompt net is equivalent to the non-prompt net as far as the functional input/output behavior is concerned. The prompt semaphore system resulting from this transformation is shown in Figure 3.

We have chosen Petri nets for studying the transformation of non-prompt systems into prompt systems because Petri nets provide a convenient model for the representation of such aspects of systems and because they are capable of being analyzed easily. The questions we will be asking are these: What class of non-prompt nets can be transformed into functionally equivalent prompt Petri nets and what these transformations are.

It appears that the reduced net that is obtained by deleting all the input/output transitions together with the arcs connected to them, completely characterizes the type of non-promptness the net has. For example, we say that a net has non-promptness of the state machine type if the reduced net is a non-empty collection of state machines. The transformation of such a net involves collapsing each state machine into a single place with a token count equal to the sum of the tokens of all the places in the state machine. For example, the net of Figure 4a is transformed into a prompt net by collapsing the state machine consisting of places w, x, y, and z and transitions t_a , t_b , t_c , t_d and t_e into one single place X as shown in Figure 4b.

Our study has shown that this simple transformation does not work for the other types of non-promptness. We propose to study non-promptness of the marked graph type, the free choice net type and the state machine decomposable net type, and investigate the question whether all non-prompt Petri nets can be transformed into prompt nets. If all nets cannot be transformed into prompt nets we propose to find a net that cannot be so transformed and produce a

formal proof to establish that fact.

The study of promptness is useful because in reducing a net to show that it is functionally equivalent to another, one often encounters nets which are non-prompt, and the non-promptness must be eliminated in order to show that the nets are equivalent. Furthermore, the study of promptness should provide new insight into the fundamentals of systems.

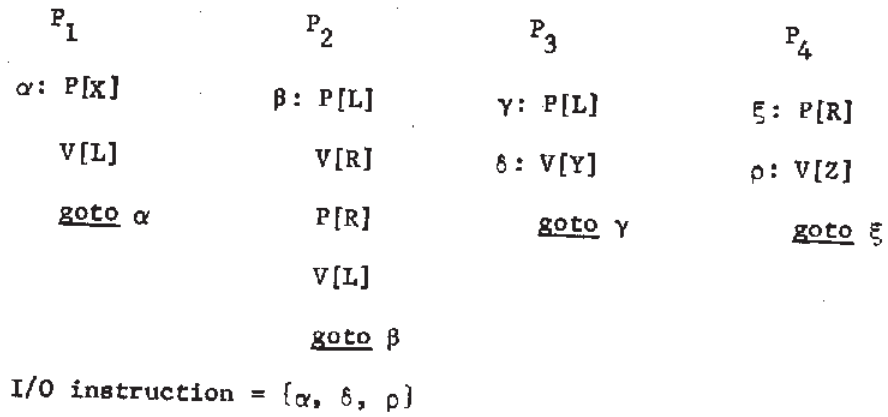


Figure 1. A non-prompt semaphore system.

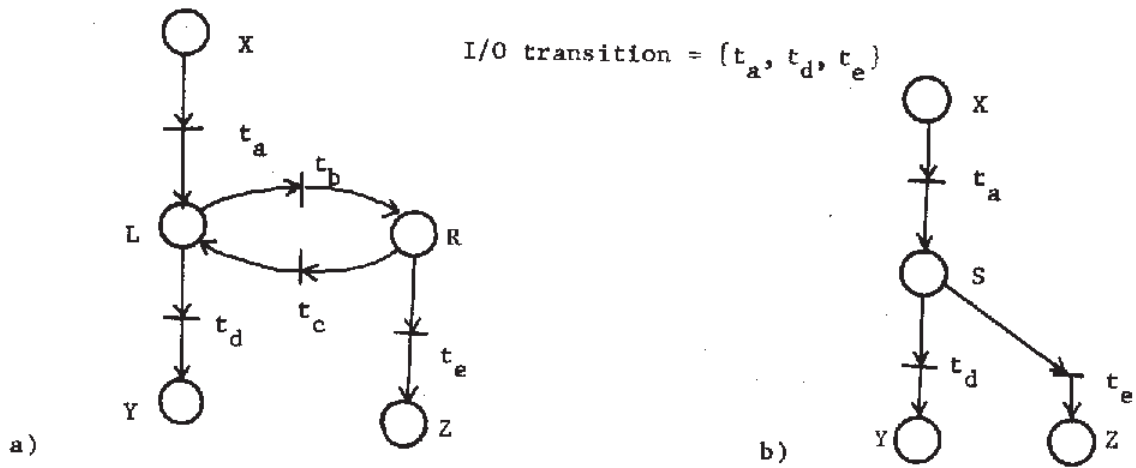


Figure 2. Petri net representation and transformation into a prompt system.

P_1	P_2	P_3
$\alpha: P[X]$	$\gamma: P[S]$	$\xi: P[S]$
$V[S]$	$\delta: V[Y]$	$\rho: V[Z]$
<u>goto</u> α	<u>goto</u> α	<u>goto</u> ξ
I/O instructions = $\{\alpha, \delta, \rho\}$		

Figure 3. Prompt semaphore system.

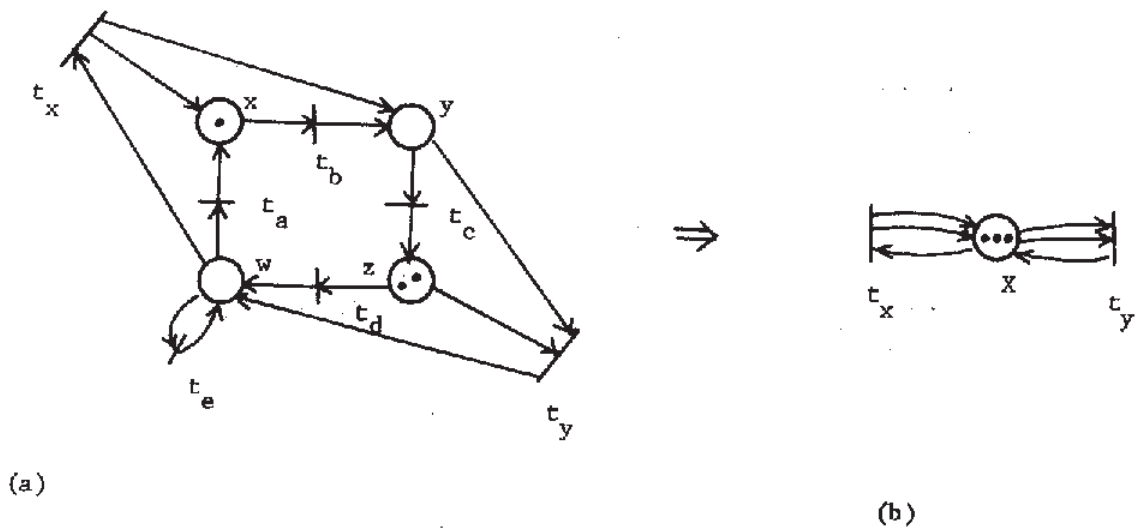


Figure 4. Transformation of the non-promptness of the state machine type.

REFERENCES

1. Adams, R. L., D. R. Castaldo, and G. W. Kurz. Real-time Detection of Latch Resolution Using Threshold Means. U. S. Patent 3,515,998, June 2, 1970.
2. Baer, J. L. Graph Models of Computations in Computer Systems. Report 68-46, Department of Engineering, University of California, Los Angeles, California, October 1968.
3. Baer, J. L., D. P. Bovet, and G. Estrin. Legality and other properties of graph models of computations. J. of the ACM, Vol. 17, No. 3 (July 1970), pp 543-554.
4. Baker, H. G. Equivalence Problems of Petri Nets. S.M. Thesis, Department of Electrical Engineering, MIT, Cambridge, Mass., 1973.
5. Bell, G. C. and A. Newell. Computer Structures: Readings and Examples. McGraw Hill Book Co., New York, N. Y., 1971.
6. Bell, G. C. and A. Newell. The PMS and IPS descriptive systems for computer structures. AFIPS Conference Proceedings, Vol. 36, Spring 1970, pp 351-374.
7. Bell, G. C., J. Grason and A. Newell. Designing Computers and Digital Systems, Digital Press, Maynard, Mass., 1972.
8. Bruno, J. and S. M. Altman. Asynchronous control networks. IEEE Conference Record. Tenth Annual Symposium on Switching and Automata Theory, 1969, pp 61-73.
9. Bernstein, P. Description Problems in the Modelling of Asynchronous Computer Systems. Report TR-48, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 1973.
10. Catt, I. Time loss through gating of asynchronous logic signal pulse. IEEE Trans. on Electronic Computers, Vol. EC-15, No. 1 (February 1966), pp 108-111.
11. Chaney, T. J. and C. E. Molnar. Anomolous behavior of synchronizer and arbiter circuits. IEEE Trans. on Computers, Vol. C-22, No. 4 (April 1973), pp 421-422.
12. Clark, W. A. Macromodular computer systems. AFIPS Conference Proceedings, Vol. 30, Spring 1967, pp 335-336.
13. Commoner, F., A. W. Holt, S. Even, and A. Pnueli. Marked directed graphs. J. of Computer and System Sciences, Vol. 5, May 1971, pp 511-523.
14. Dennis, J. B. Modular, asynchronous control structures for a high performance processor. Record of the Project MAC Conference on Concurrent Systems and Parallel Computation, ACM, New York 1970, pp 55-80.
15. Dennis, J. B. and S. S. Patil. Computation Structures. Notes for Subject 6.232, Department of Electrical Engineering, MIT, Cambridge, Mass., 1971. Most of the relevant concepts were included in the edition of the notes used for a Summer Conference at Princeton University, 1968.

16. Dennis, J. B., and S. S. Patil. Speed independent asynchronous circuits. Proceedings of the Fourth Hawaii International Conference on System Sciences, 1971.
17. Friedman, A. D., and P. R. Menon. Systems of asynchronously operating modules. IEEE Trans. on Computers, Vol. C-20, 1971, pp 100-104.
18. Furtek, F. C. Modular Implementation of Petri Nets. S.M. Thesis, Department of Electrical Engineering, MIT, Cambridge, Mass., September 1971.
19. Genrich, H. J. Simple Nonsequential Processes. Gesellschaft fur Mathematik und Datenverarbeitung, Bonn, 1971.
20. Gostelow, K, P. Flow of Control, Resource Allocation and Other Proper Terminations of Programs. Ph.D Thesis, Computer Science Department, University of California, Los Angeles, Calif., 1971.
21. Hack, M. H. Analysis of Production Schemata. S.M. Thesis, Department of Electrical Engineering, MIT, Cambridge, Mass., 1972.
22. Hack, M. H. A Petri Net Version of Rabin's Undecidability Proof for Vector Addition Systems. Computation Structures Group Memo 94, Project MAC, Cambridge, Mass., December 1973.
23. Hack, M. H. Forthcoming report on reducibility of liveness and reachability problems.
24. Holt, A. W. Final Report of the Information System Theory Project. Technical Report RADC-TR-68-305, Rome Air Development Center, Griffiss Air Force Base, New York, 1968.
25. Holt, A. W., and F. Commoner. Events and Conditions. (In three parts), Applied Data Research, New York 1970. [Chapters I, II, IV and VI appear in Record of the Project MAC Conference on Concurrent Systems and Parallel Computation, ACM, New York 1970, pp 3-52.]
26. Holt, Richard C. On Deadlock in Computer Systems. Technical Report 71-91, Department of Computer Science, Cornell University, Ithaca, New York, January 1971.
27. Jump, J. R., and P. S. Thiagarajan. On the equivalence of asynchronous control structures. SIAM Journal of Computing, Vol. 2, No. 2 (1973), pp 67-87.
28. Karp, R. M., and R. E. Miller. Properties of a model for parallel computations: determinacy, termination, queueing. SIAM Journal of App. Math., Vol. 14, No. 6 (November 1966), pp 1390-1411.
29. Karp, R. M., and R. E. Miller. Parallel program schemata. J. of Computer and System Sciences, Vol. 3, No. 2 (May 1969), pp 147-195.
30. Keller, R. M., and D. F. Wann. Analysis of Implementation Errors in Digital Computing Systems. Technical Report 6, Computer Systems Laboratory, Washington University, St. Louis, Missouri, March 1968.

31. Keller, R. M. Vector Replacement Systems: A Formalism for Modeling Asynchronous Systems. Technical Report 117, Princeton University, Computer Science Laboratory, December 1972.
32. Kosaraju, S. R. Limitation of Dijkstra's Semaphore Primitives and Petri Nets. Hopkins Computer Research Report 25, Computer Science Program, The John Hopkins University, Baltimore, Maryland, May 1973.
33. Littlefield, W., and T. J. Chaney. The Glitch Phenomenon. Technical Memorandum 10, Computer Systems Laboratory, Washington University, St. Louis, Missouri, December 1966.
34. Miller, R. E. A comparison of some theoretical models of parallel computation. IEEE Transactions on Computers, Vol. C-22, No. 8 (August 1973), pp 710-717.
35. Misunas, D. Petri nets and speed-independent design. Comm. of the ACM, Vol. 16, No. 8 (August 1973), pp 474-481.
36. Muller, D. E. Flow chart methods for the logical design of an asynchronous control. File No. 262, Digital Computer Laboratory, University of Illinois, January 1959. [Presented at the UNESCO International Conference on Information Processing, Paris, 1959.]
37. Muller, D. E. Asynchronous logics and application to information processing. Switching Theory in Space Technology, Stanford University Press, Stanford, California, 1963.
38. Muller, D. E. The general synthesis problem for asynchronous digital networks. IEEE Conference Record. Eighth Annual Symposium on Switching and Automata Theory, 1967, pp 70-82.
39. Muller, D. E., and W. S. Bartky. A theory of asynchronous circuits. Proceedings of an International Symposium on the Theory of Switching, The Annals of the Computation Laboratory of Harvard University, Vol. 29, Part I, Harvard University Press, Cambridge, Mass., 1959, pp 204-243.
40. Noe, J. D., and G. J. Nutt. Macro E-nets for representation of parallel systems. IEEE Trans. on Computers, Vol. C-22, No. 8 (August 1973), pp 710-717.
41. Parnas, D. L. On a Solution to the Cigarette Smoker's Problem (Without Conditional Statement). Technical Report, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1972.
42. Patil, S. S. Coordination of Asynchronous Events. Report MAC-TR-72, Project MAC, MIT, Cambridge, Mass., June 1970.
43. Patil, S. S. Closure properties of interconnections of determinate systems. Record of the Project MAC Conference on Concurrent Systems and Parallel Computation, ACM, New York 1970, pp 107-116.

44. Patil, S. S. Limitations and Capabilities of Dijkstra's Semaphore Primitives for Coordination Among Processes. Computation Structures Group Memo 57, Project MAC, MIT, Cambridge, Mass., February 1971.
45. Patil, S. S. Forward Acting n x m Arbiter. Computation Structures Group Memo 67, Project MAC, MIT, Cambridge, Mass., April 1972.
46. Patil, S. S. Circuit Implementation of Petri Nets. Computation Structures Group Memo 73, Project MAC, MIT, Cambridge, Mass., October 1972.
47. Patil, S. S., and J. B. Dennis. Description and realization of digital systems. Proceedings of the Sixth Annual IEEE Computer Society International Conference, San Francisco, Calif., September 1972.
48. Patil, S. S. Synchronizers and Arbiters. Computation Structures Group Memo 91, Project MAC, MIT, Cambridge, Mass., October 1973.
49. Petri, C. A. Communication With Automata. Supplement 1 to Technical Report RADC-TR-65-377, Vol. 1, Griffiss Air Force Base, New York 1966. [Originally published in German: Kommunikation mit Automaten, University of Bonn, 1962.]
50. Ramchandani, C. Analysis of Asynchronous Concurrent Systems by Timed Petri Nets. Ph.D Thesis, Department of Electrical Engineering, MIT, Cambridge, Mass., 1973.
51. Robertson, J. E. Problems in the physical realization of speed independent circuits. Proceedings of the Second Symposium on Switching Circuit Theory and Logical Design, AIEE, October 1961, pp 106-108.
52. Seitz, C. L. Graph Representations for Logical Machines. Ph.D Thesis, Department of Electrical Engineering, MIT, Cambridge, Mass., 1971.
53. Swartwout, R. E. New techniques for designing speed independent control logic. Proceedings of the Fifth Annual Symposium on Switching Circuit Theory and Logical Design, IEEE, New York, 1964, pp 12-29.
54. Thiagarajan, P. S. A Study of an Extended Class of Asynchronous Control Systems. Ph.D Thesis, Department of Electrical Engineering, Rice University, Houston, Texas, 1973.