May 1968 USCEE Report 278

UNIVERSITY OF SOUTHERN CALIFORNIA

THE SYNTHESIS OF PERIODIC SEQUENTIAL MACHINES

R. A. King

W. S. Meisel

ELECTRONIC SCIENCES LABORATORY

USC Engineering

May 1968 USCEE Report 278

The Synthesis of Periodic Sequential Machines

R. A. King and W. S. Meisel

This research was supported in part by NASA Grant No. NGR-05-018-044, Supp. #1.

ABSTRACT

The effects of loops upon the conversion of a sequential machine into a periodic machine are examined, and theorems and corollaries derived which predict the allowable periods for non-trivial periodic machine equivalents of sequential machines. A periodization procedure is developed and presented. The theorems and corollaries are initially derived for minimal, strongly connected sequential machines, and then extended to include non-minimal, non-strongly connected sequential machines, as well.

Preface

This report represents work begun in the summer of 1967. About the time the work was complete, Gill and Flexer published results similar to those presented herein. The approach and proofs differ sufficiently that we thought it worthwhile to make our work available in this form.

This work was performed in part in partial fulfillment of the requirements for Mr. King's Master of Science degree (Electrical Engineering).

The author's would like to express their appreciation to Professor Kaprielian and Professor McGhee for their comments on this research.

TABLE OF CONTENTS

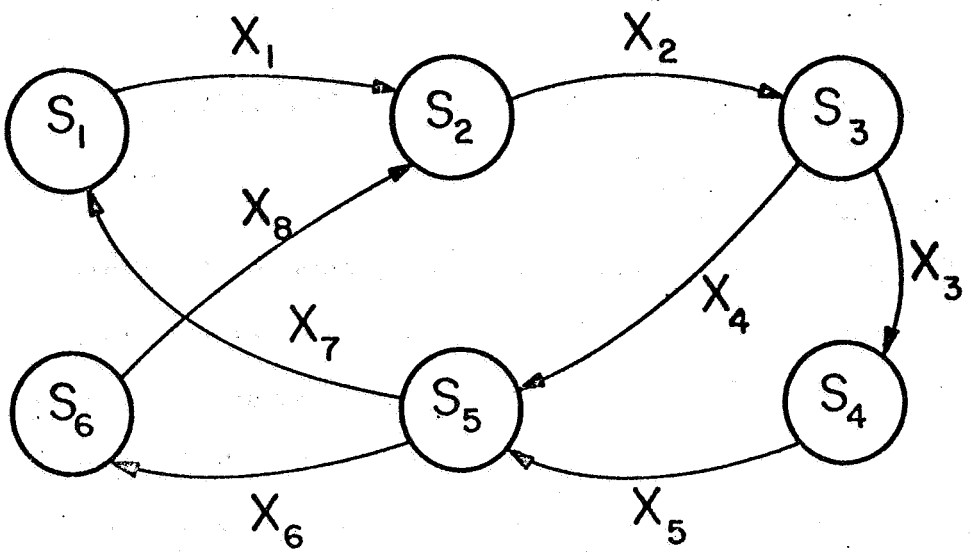
I. INTRODUCTION TO PERIODIC MACHINES

A. Preliminary Definitions

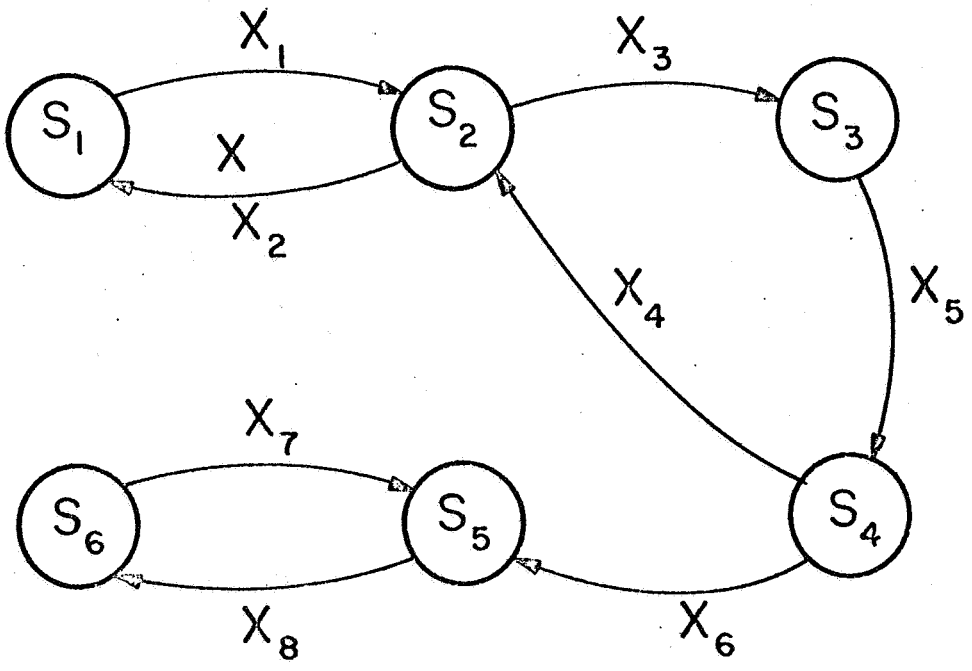
In this paper, we are concerned with a class of sequential machines in which transitions occur at particular intervals, e. g. synchronous (or clocked) sequential machines. The symbol M will be used to designate such machines. The input states to the machine M are denoted by the symbols X_i, the internal states by S_i and the output symbols by Z_i. The machine will be described by its state diagram, and will be a minimal machine. The name given the combination of input and internal states will be total state.

We shall further restrict the discussion, in general, to strongly connected machines; that is, machines for which there exists some input sequence $X_1, X_2, \ldots X_n$ which will cause the machine to progress from any given S_i to S_j, where i may equal j. An example of such a machine is shown in Figure 1a. Figure 1b portrays a machine which is not strongly connected; the input sequence X_3, X_5, X_6 (beginning in state S_2) will cause the machine to sequence to state S_5, from which there is no finite sequence causing a return to S_2.

In the later sections of this paper, certain of these restrictions are examined, and the findings extended to cover those cases.



a) Strongly Connected Machine



b) Non-Strongly Connected Machine

Figure 1: Classes of Sequential Machines

B. Periodic Machines

A periodic machine is a machine which consists of a set of input symbols, internal states, and output symbols, as defined previously. In addition, the periodic machine has a set of next-state functions which vary with time. Functionally, a periodic machine can be considered as shown in Figure 2, with input and output combinational logic blocks and a memory. The input and output logic blocks vary in composition with time in a periodic manner; they may be considered as being replaced with other blocks at each clock. Sequential machines whose next state functions do not vary with time may be considered as a special class of periodic machines whose period is 1.

The primary advantage of periodic machines lies in the potential of lesser memory capacity required. As an example, the conventional fixed machine of Figure 3 requires 8 states, whereas the equivalent periodic machine shown in Figure 4 requires only 2 states at any given time period. The saving in logical elements is demonstrated by the implementations of the fixed and periodic machines shown in Figures 5 and 6. The periodic representation assumes the existence of a multi-phase clock system; this is discussed later in this section.

In general, however, the lessening in memory capacity achieved in the transformation from fixed to periodic machine is accompanied by a like amount of memory required to remember in which time period the periodic machine is operating. In addition, the added complexity of the additional combinational circuitry required

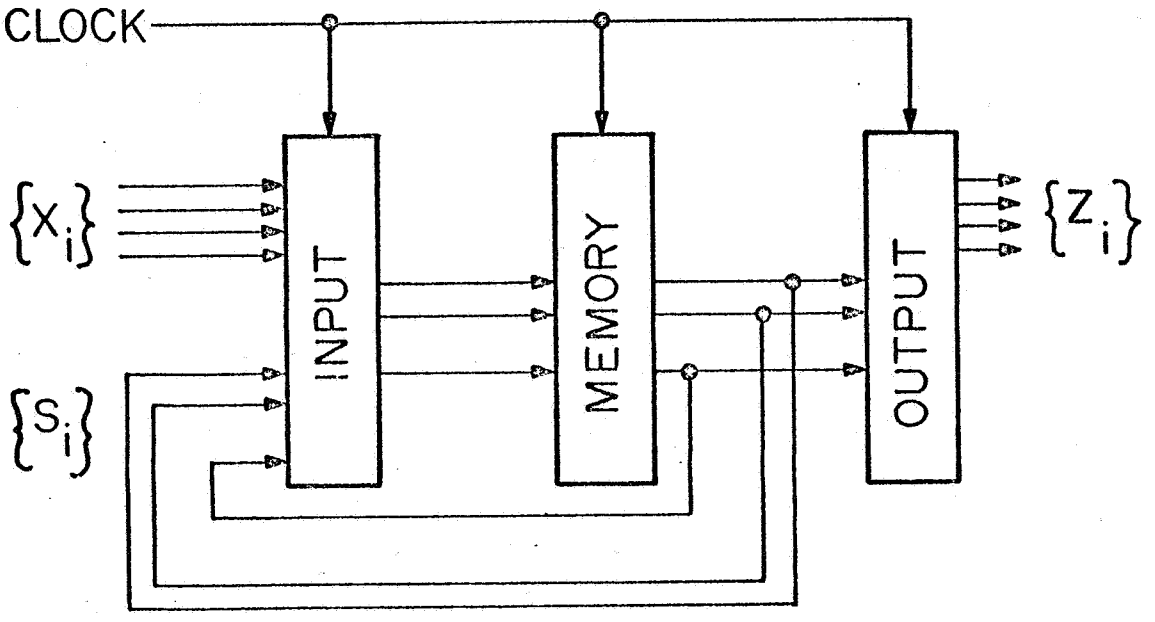


Figure 2: Generalized Periodic Machine

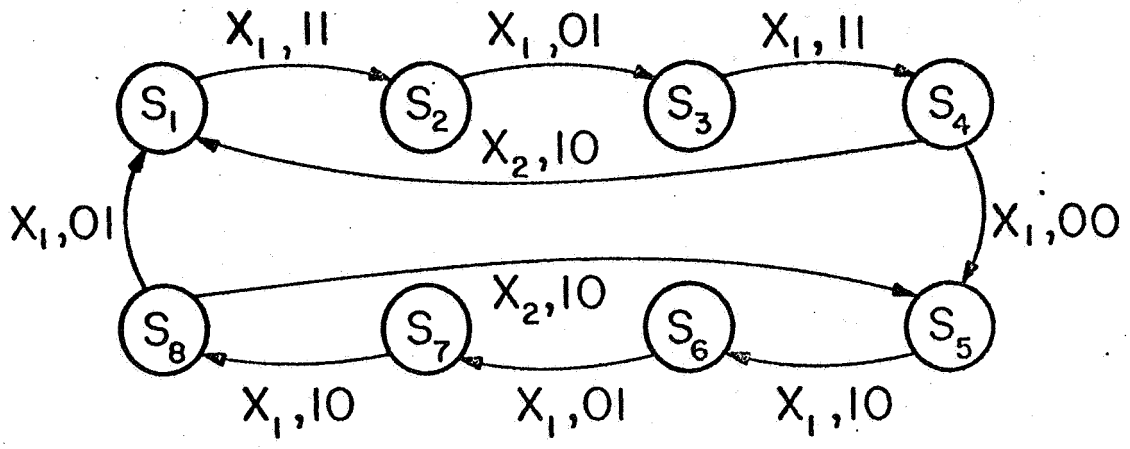


Figure 3: Eight State Sequential Machine

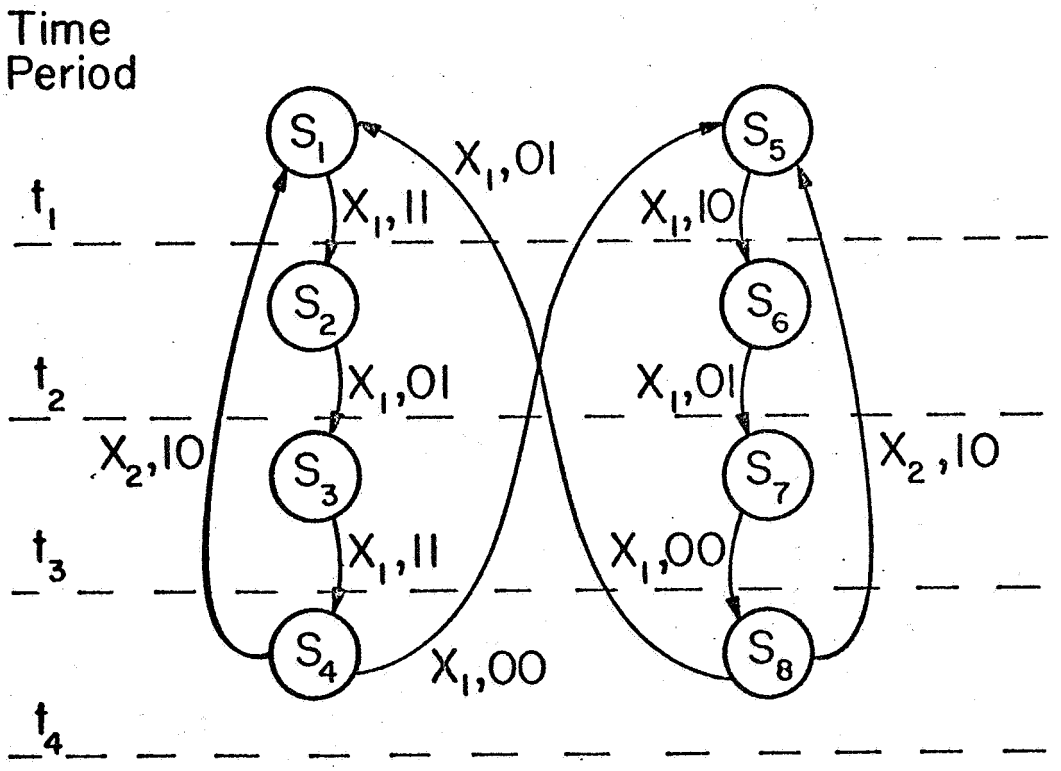


Figure 4: Periodic Equivalent of Machine in Figure 3

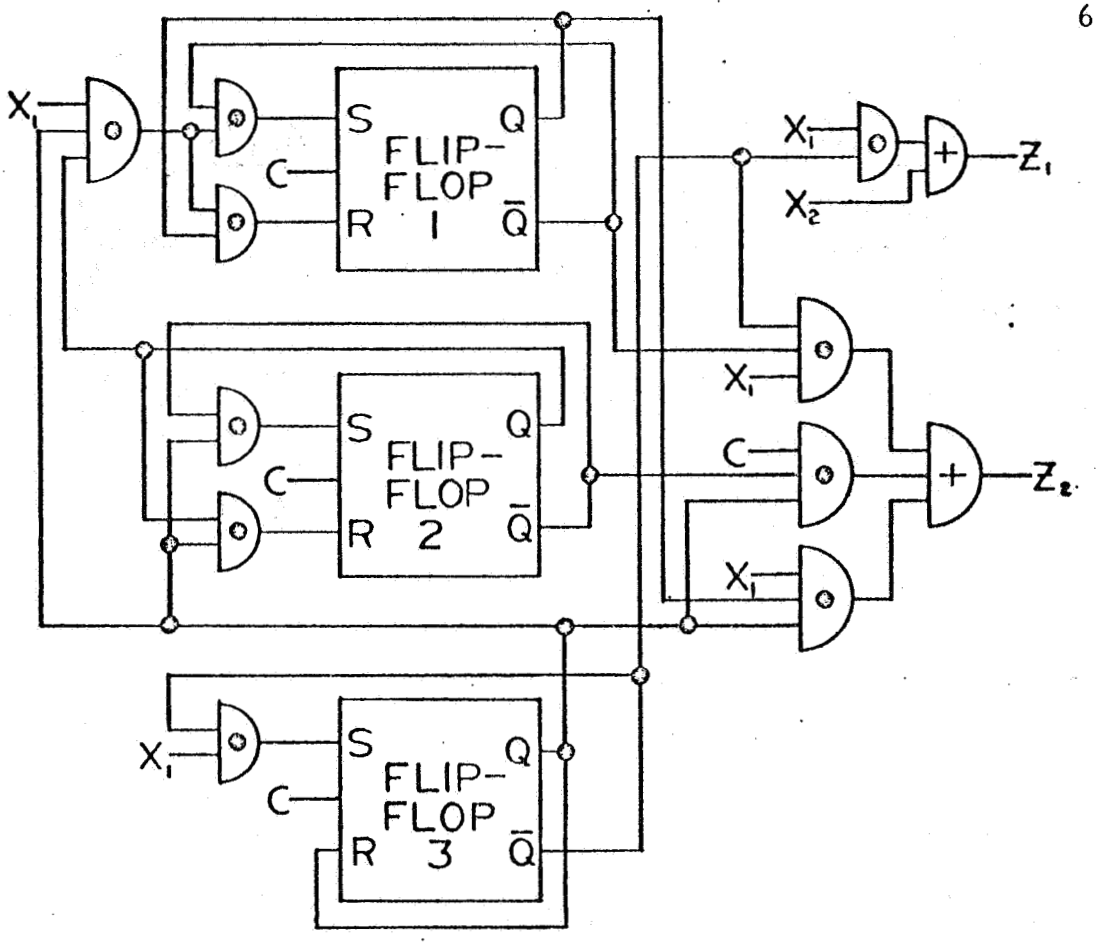


Figure 5: Implementation of Conventional Machine

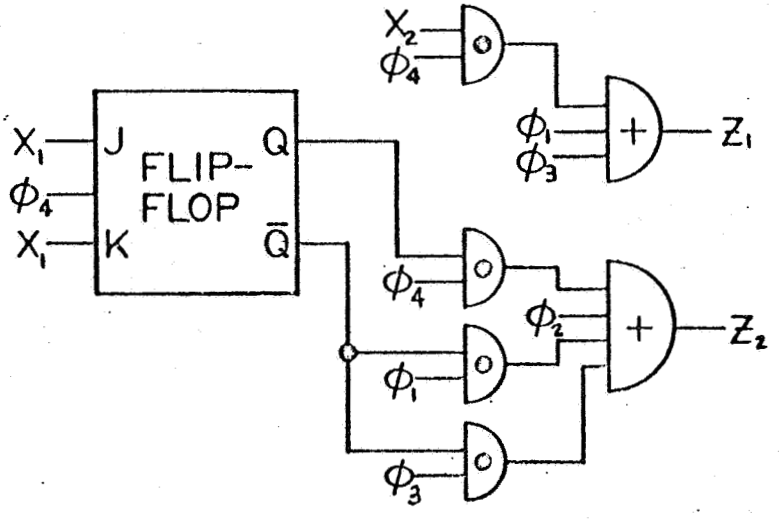


Figure 6: Implementation of Periodic Machine

may be a disadvantage.

There are equipments in which some centralized timing source already exists, such as a multi-phase clock system [5] or a "p-time counter". The availability of such a source restricts the choice of period of the periodic machine but does provide a "free" memory of the clock period in which the periodic machine is operating, and thus obviates the necessity for such a memory to be included in the periodization. Additionally, Meisel [3] has shown that it is possible to utilize variable-threshold threshold logic and an analog signal as the timing memory. For such cases, the advantages mentioned above for periodic machines may be obtained without the accompanying cost of clock memory and logic. It is to those cases that the research preceding this Thesis was directed.

The primary problem to be solved is the derivation of a procedure for the transformation of fixed machines to equivalent periodic machines, if such a transformation will result in a saving of logical elements.

One method whereby the transformation is particularly easy .is where appropriate partitioning of the states can be accomplished [1]. A partition is a collection of substates which are pairwise disjoint, and whose union is the entire collection of states of M. In this case, the transition from one collection of substates $\{S_i\}$ to another $\{S_j\}$ is allowed, but to no other; the $\{S_j\}$ must have a transition to $\{S_k\}$ etc. When all states can be partitioned into such classes, and there is a transition from the final partition $\{S_n\}$ to the initial partition $\{S_i\}$, then closure is said to have occurred and the partitioning

is complete and meaningful. Unfortunately, it is not always apparent which partitioning of a given fixed machine will prove successful; this paper explores an alternate means of determining the feasibility of conversion of a given fixed machine into a periodic machine, by examination of the "loops" of the fixed machine.

II. ANALYSIS AND SYNTHESIS PROCEDURES

A. Nature of the Problem

Given a minimal, strongly connected, sequential machine M, it is desired to determine under what conditions, if any, a corresponding non-trivial periodic machine can be devised.

At the same time, it would be desirable to be able to specify the period of such a machine, to suit the operational environment; hence, there is a requirement for an alogrithmetic method for achieving the transformation, if one exists.

Finally, the creation of a method which minimizes the number of states during any given period of the periodic machine should be investigated.

B. Definitions

Prior to further discussion of the above topics, it is necessary to formulate the following definitions.

Definition: If a conventional sequential machine is represented as an equivalent periodic machine, it will be said to have been periodicized.

Definition: Let M be a sequential machine as previously defined. If there exists an input sequence of length n; X_1, $X_2 \ldots X_m$ which, when M is initially in state S_1 induces the sequence of states S_2, S_3, . . . , S_n, S_1, where none of the S_i (i = 2, 3, . . . , n) are S_1; then M contains a loop of period n.

In particular, if there exists an input which does not change the internal state of the machine, M has a loop of period 1.

For example, Figure 7 indicates a machine with loops of period 1, 3, and 4.

Definition: It is clear that any machine M, of m states, can be periodicized into a machine of period T by duplicating the m states T times, and modifying the transitions accordingly; such a periodization is termed trivial.

Figure 8 presents a two-state machine which is trivially periodicized. From this definition, it may also be seen that conventional sequential machines can be said to be trivially periodic, with period 1.

Definition: In a periodic machine, the group of states which may occur at a particular time is called the state set of that time.

C. Loop Dominance in Sequential Machines

In this section, the discussion is centered about strictly periodic machines; the extension of the theorems derived to machines containing a transient set of states as well as a strictly periodic set is discussed in a later section of this paper. We also do not in general discuss nor show on state diagrams the outputs, to simplify the diagrams.

Lemma 1: In a strongly connected periodic machine M, if a state appears in every state set, then the periodic representation is trivial.

Proof: Consider a periodic machine in which state S_i appears

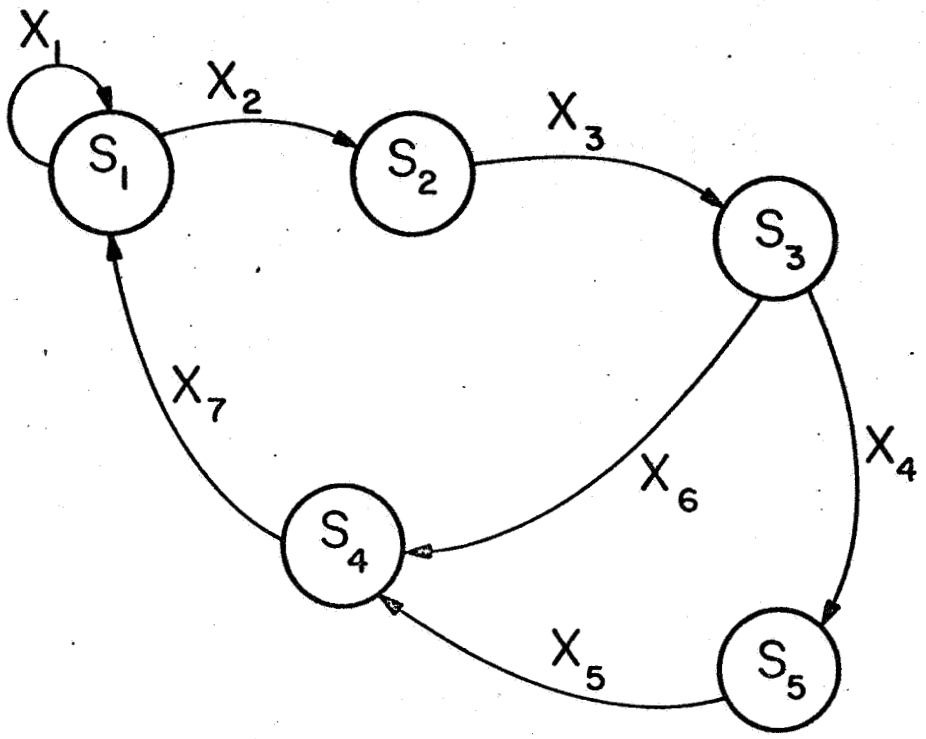
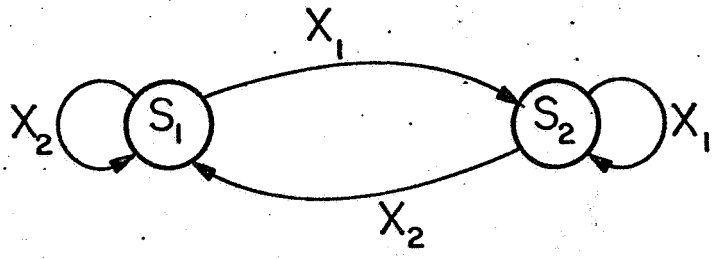
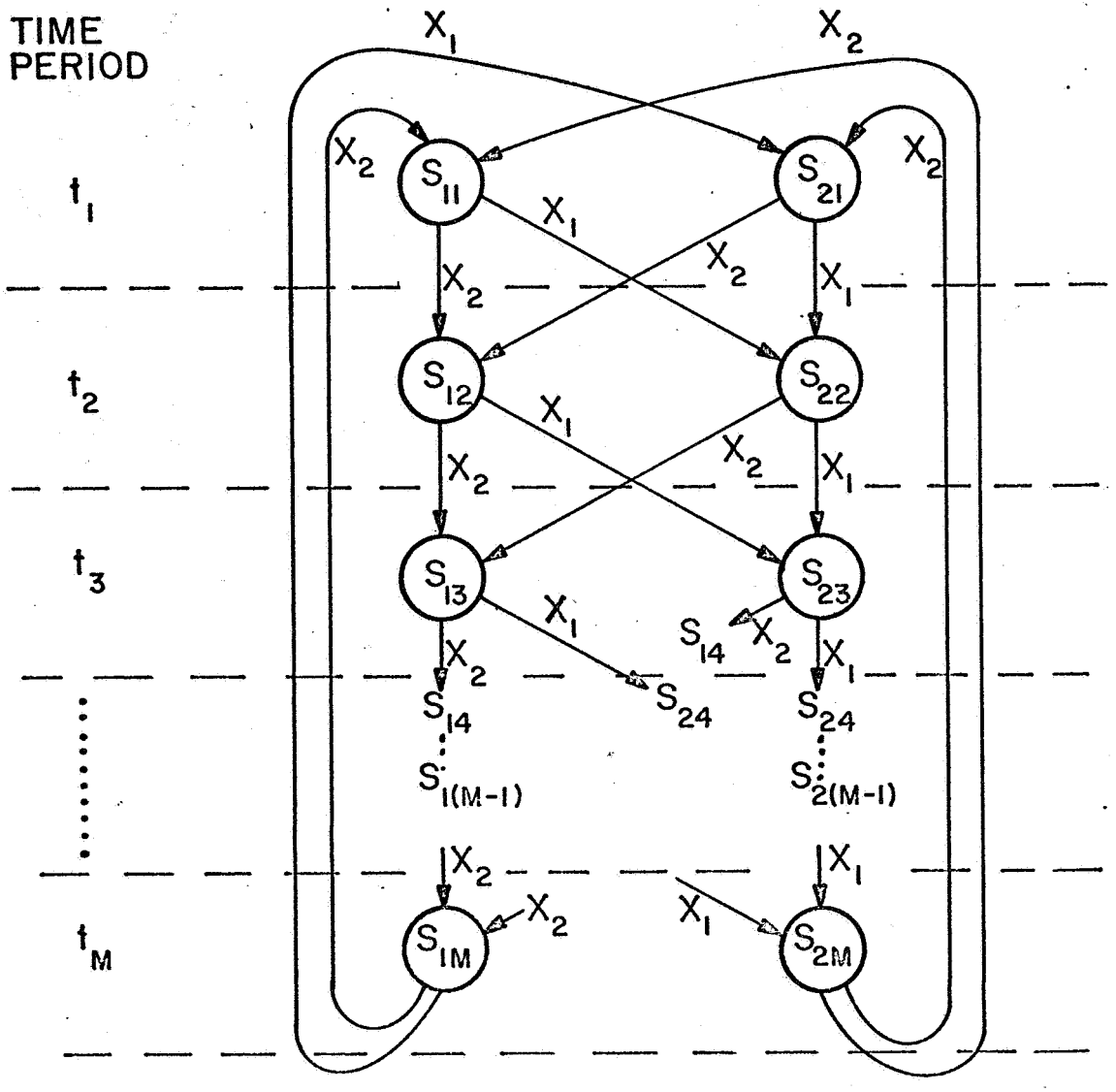


Figure 7: Sequential Machine with Multiple Loops



a) SEQUENTIAL MACHINE



b) TRIVIAL PERIODIC MACHINE WITH PERIOD T_M.

Figure 8: Demonstration of Trivial Representation

in every state set. Since state S_i of state set 3 which we denote by S_{i3} has at least one predecessor state S_{h2} and at least one successor state S_{j4}, they also must appear in every state set. In addition, the predecessor state of S_{h2} and the successor state of S_{j4} must appear in every state set. By induction, every state must appear in every state set, since M is strongly connected, and the representation is hence trivial.

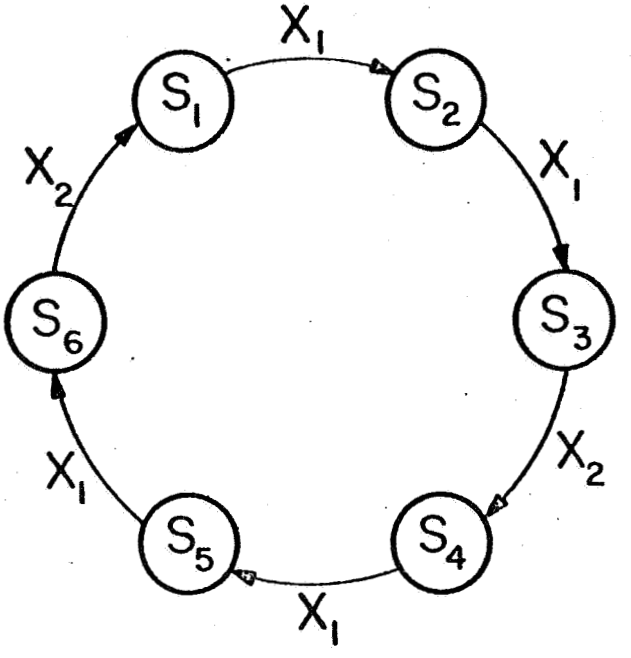
Theorem 1: Let M be a minimal, strongly connected sequential machine with a loop of period n. In any non-trivial periodic machine M_p which is the equivalent in the usual sense to M, the period T must be a multiple of one of the prime factors of n.

Proof: (By contradiction) Assume that a non-trivial periodic machine M_p exists, with a period T which is not a multiple of one of the prime factors of n, and yet is equivalent to M. Then, since M and M_p are equivalent, any state S_i of M in the loop of length n must have an equivalent state in M_p. Let us denote that state S_i^*. Since M has a loop of period n, there exists a sequence of inputs yielding a sequence of distinct states $S_i, S_{i+1}, \ldots, S_{i+(n-1)}, S_i$. S_{i+1} also has an equivalent state in M_p, S_{i+1}^*. After progressing along the successor states of S_i exactly n-1 transitions, the state $S_{i+(n-1)}$ is reached. In order for M_p to be equivalent to M, it must have equivalent states for each of the n-1 transitions mentioned above, ending in a state equivalent to $S_{i+(n-1)}$, $S_{i+(n-1)}^*$. Since the next transition of M results in a transition from $S_{i+(n-1)}$ to S_i, completing the loop of period n, M_p must have a transition from $S_{i+(n-1)}^*$ to an S_i^*, but this S_i^* cannot be in the same state set as the first S_i^*, by the hypothesis that

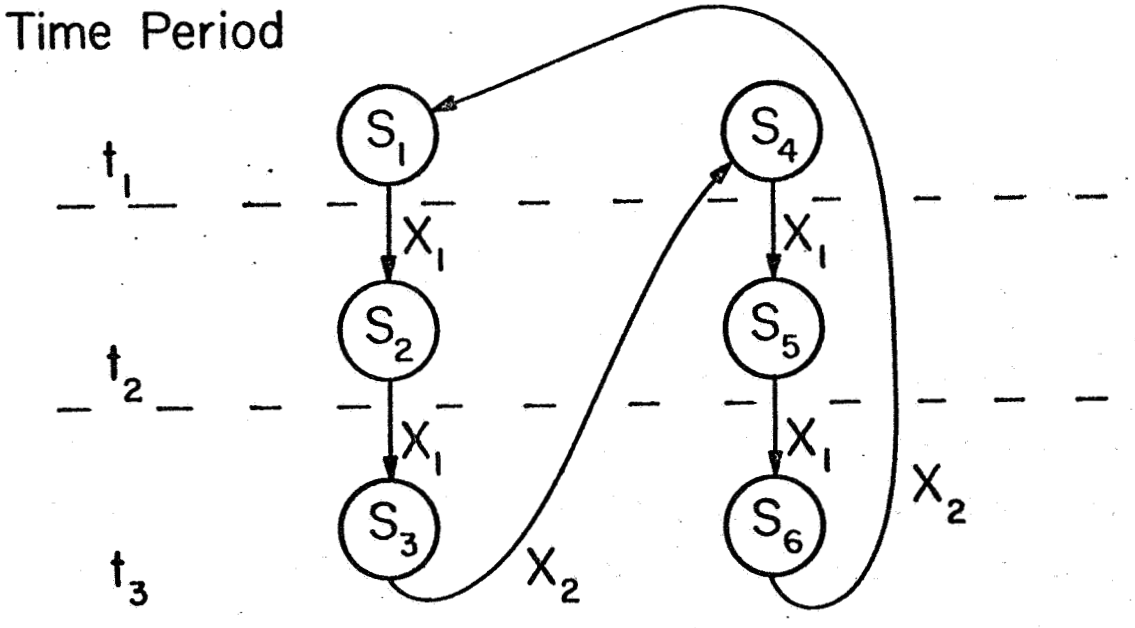
M is not of a period divisible by one of the prime factors of n. We thus are forced to have two states S_i^* in M_p, in different state sets. Denote these by S_{i1}^* for the beginning S_i^* and S_{i2}^* for the next S_i^*, etc. We again traverse the loop of period n in M, beginning with M_p in state S_{i2}^*. At the end of n transitions, another S_i^* must be chosen, S_{i3}^*, since if the latest S_i^* were in the same state set as S_{i1}^* or S_{i2}^*, T and n would be related by a common prime factor, which would be a contradiction. Traversal of the loop continues in the above manner, until an S_i^* is assigned to every state set. Since there are T S_i^* states, the machine is trivial, by Lemma 1. A contradiction is thus reached, proving the Theorem.

Theorem 1 provides information regarding the periods which may be used for non-trivial periodizations, but provides no information regarding the relative merits of two or more periodizations which satisfy Theorem 1. An added constraint, that T divide n, provides this added information. The maximum saving in memory required accrues when T is the greatest divisor of n [2].

Example: Consider the fixed machine M shown in Figure 9a. Examination of the machine reveals that the loop is of period 6. By Theorem 1, any non-trivial periodic machine M_p must have a period T such that T and n have a common prime factor. Since 6 has prime factors 3 and 2, T may thus be any multiple of those numbers. If we choose T = 3, M_p is as shown in Figure 9b. We thus require only 2 bits of memory at any given time, as opposed to 6 in the fixed machine M.



a) Fixed Machine



b) Periodic Machine

Figure 9: Demonstration of Theorem 1

If the additional restriction that T divide n is not followed, a machine results which, although not trivial, is inferior to machines satisfying the rule, inasmuch as no fewer storage states in any state set are required, and the number of transitions is larger. The presence in the periodic machine of more than one state equivalent to a state of the fixed machine also complicates the output logic.

Example: Consider the machine with loop of period 8 shown in Figure 10. The periodicized machine with period T = 4 is shown in Figure 11a, and the machine with period T = 6 is shown in Figure 11b. The latter machine is clearly inferior to the T = 4 machine, but is not the trivial representation.

We next examine a machine M with loops n_i (i = 1, 2,. . ., r) as shown in Figure 12, and its non-trivial periodic representation, M_p.

Corollary 1: M_p is of period T, where T is a multiple of one of the common prime factors of $\{n_i\}$.

Proof: Machine M can be re-drawn as shown in Figure 13, and consists of r machines with common states. Any periodic representation must satisfy individually the loops; hence, by Theorem 1, for any loop n_i (i = 1, 2,. . ., r) the equivalent non-trivial periodic machine M_p must have period T, where T is divided by one of the prime factors of the n_i (i = 1, 2, . . ., r). Hence, T must be divided by all the prime factors of the $\{n_i\}$.

In addition, T must be divided by the summation of any set of the $\{n_i\}$, since a loop can consist of the sequence . . . S_{k+1}, S_{k+2}, S_{k+1}, S_{k+2}, . . . Thus, the total restriction on T is that it not only be

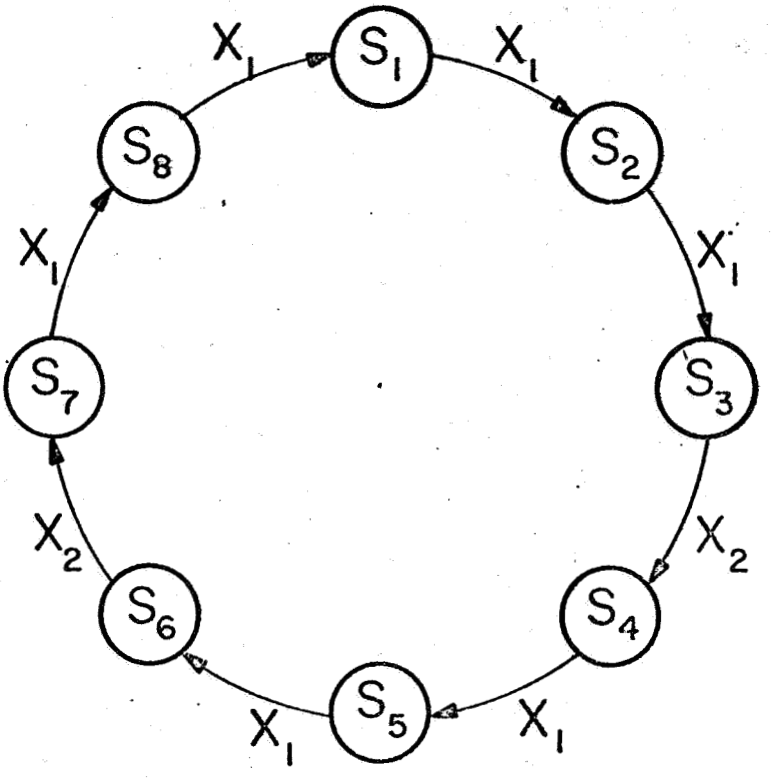
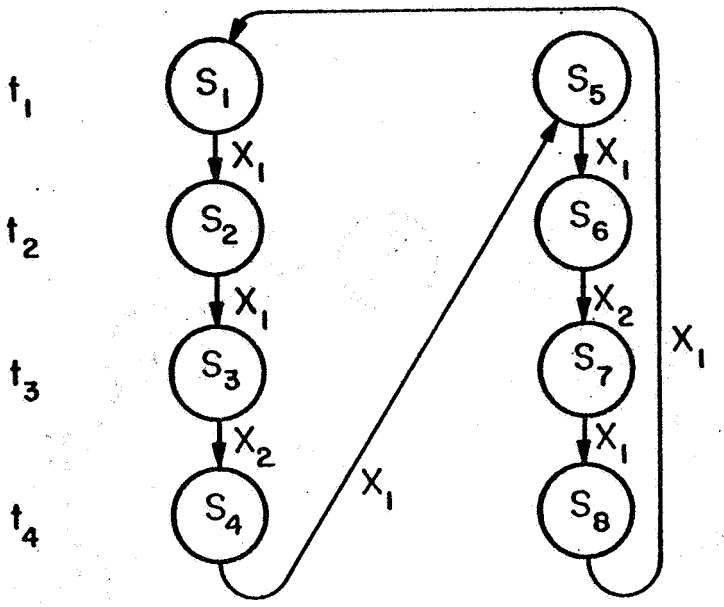
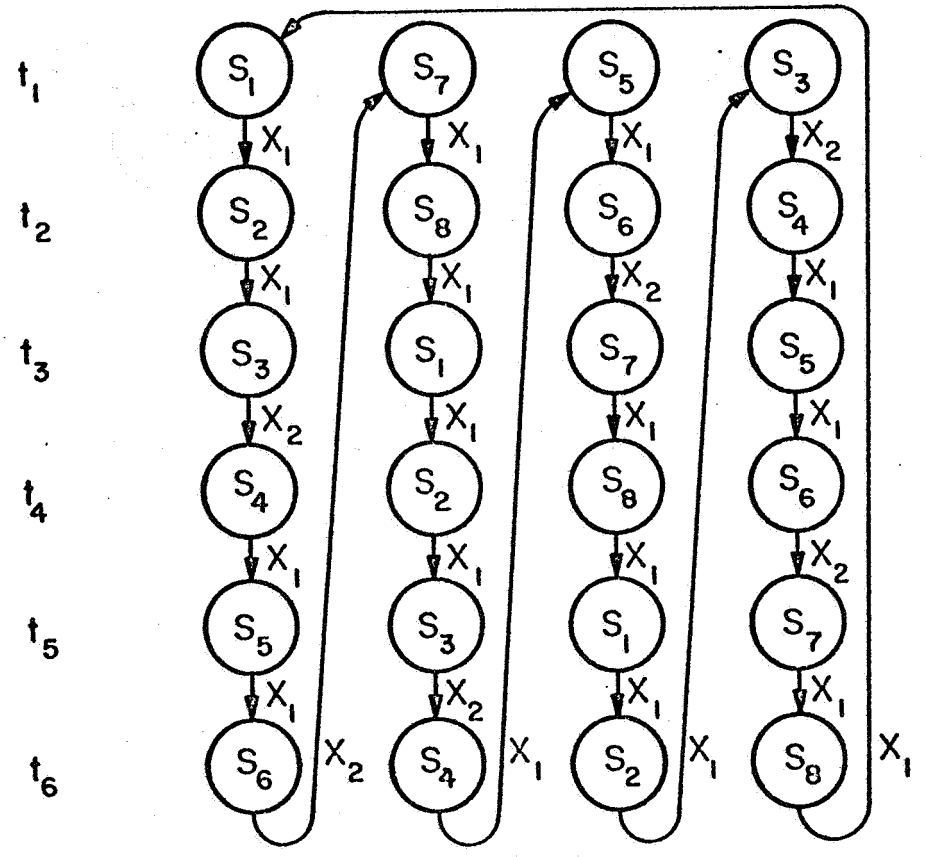


Figure 10: Eight-State Machine



a) Implementation with T = 4



b) Implementation with T = 6

Figure 11: Periodization Examples

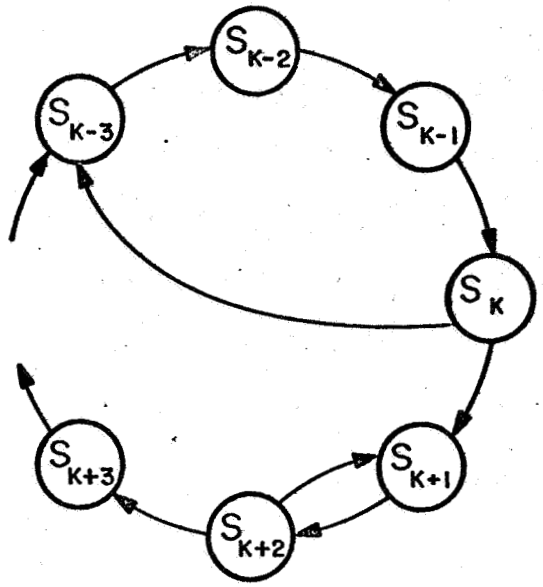


Figure 12: Multi-Loop Machine

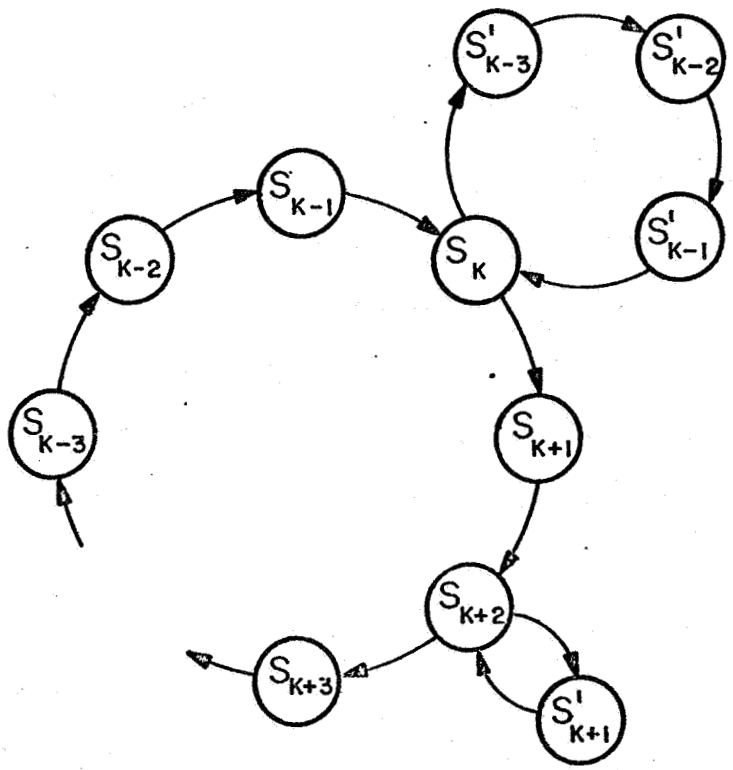


Figure 13: Multi-Machine Equivalent to Multi-Loop Machine

divided by the prime factors of the $\{n_i\}$, but it must also be divided by their summation. In order for this to be possible, the $\{n_i\}$ must have a common factor, which implies a common prime divisor.

Again, since the $\{n_i\}$ must be related by a common prime factor, the corollary below follows immediately:

Corollary 2: If two or more loops are relatively prime, no non-trivial periodic representation exists.

Example: Let us choose a machine M as shown in Figure 14. M has $n_1 = 3$ and $n_2 = 2$. Since these are relatively prime, the corollary predicts that no non-trivial periodic implementation exists. Figure 15 depicts the various phases of periodization with $T = 3$.

In Figure 15a, the n_1 loop has been periodicized. Observe, however, that there must be a S_2 to S_1 transition. In Figure 15b, a state S_1 has been added, but it must be added in state set 3, and corresponding states added to complete the other possible transitions to and from S_2. Again, however, we must include a S_2 to S_1 transition, this time with S_1 in state set 2. This inevitably leads to the accompanying states as shown in Figure 15c, which can be recognized as the trivial periodic representation, as predicted by the corollary.

The following corollary also is a direct consequence of the preceding theorem and corollaries.

Corollary 3: The only periodic representation of a machine with one or more loops of period 1 is the trivial representation.

Example: To demonstrate why this is necessarily a consequence, consider the machine in Figure 16. The loops n_1 and n_2 are of period 1 and 4. Choosing $T = 4$, Figure 17a depicts the periodic

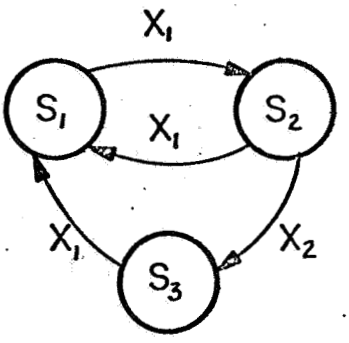


Figure 14: Machine with Loops of Period 2 and 3

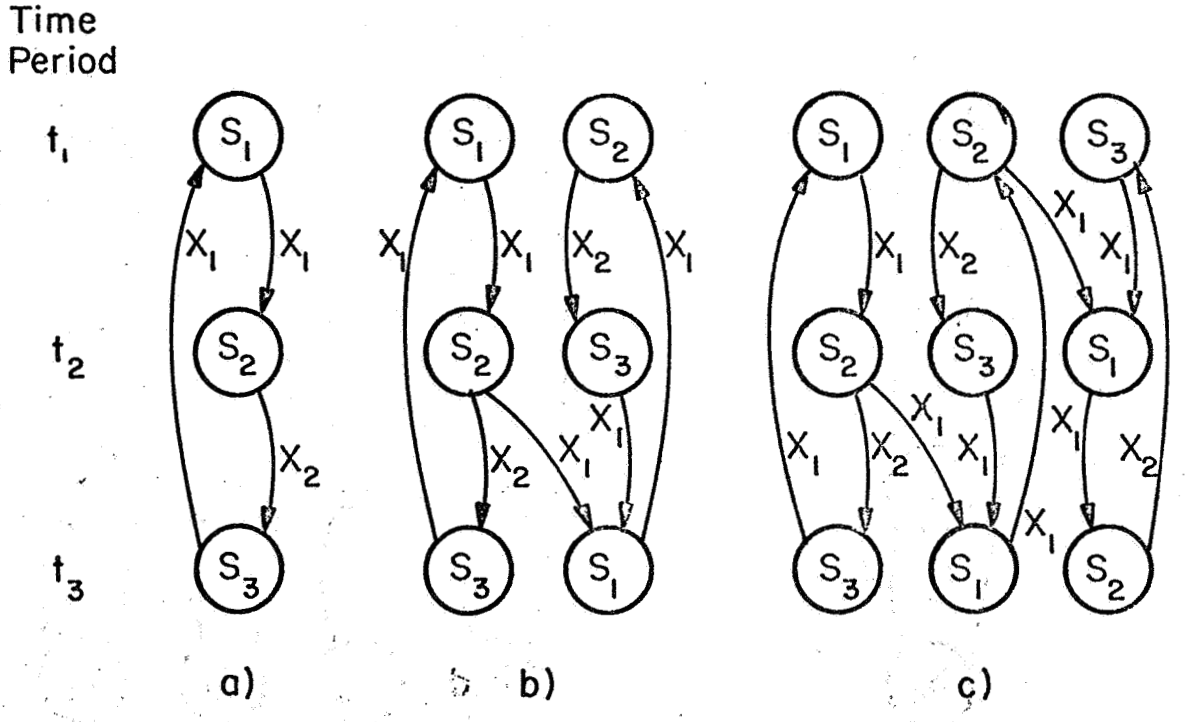


Figure 15: Demonstration of Corollary 2

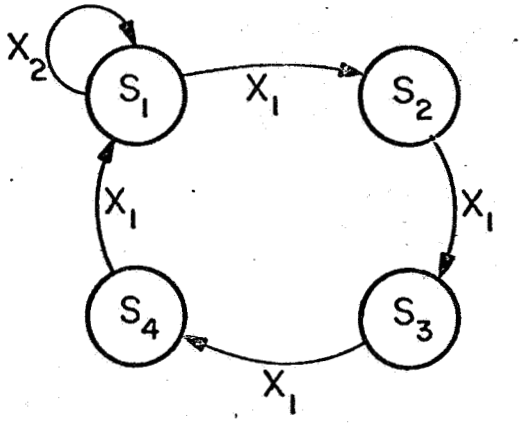


Figure 16: Machine with Loops of Period 1 and 4

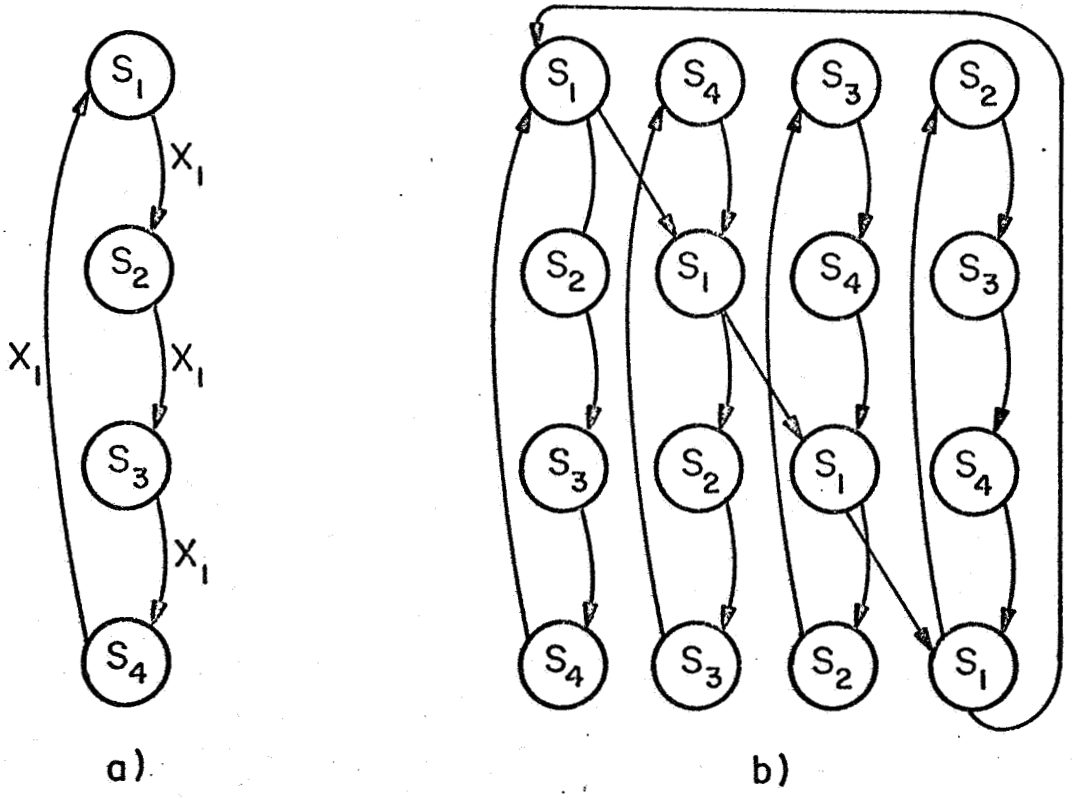


Figure 17: Demonstration of Corollary 3

representation with loop n_2 periodicized. Note that, if the loop of period 1 did not exist, the periodization would be complete. The existence of the loop of period 1 requires that a transition S_1 to S_1 exist between any state set but this in turn, implies that S_2, S_3 and S_4 must also be in every state set, which is the trivial periodic representation. The process is illustrated in Figure 17b.

Utilizing the preceding theorem and its corollaries, we can conclude under what conditions the periodization of a given machine is apt to be feasible, e.g., result in a saving of memory capacity. The conditions are dominated by the loops of the fixed machine, and hence the procedure developed in the next section will utilize this dominance.

D. Synthesis of Periodic Machines

With the fundamental principles as contained in Theorem 1 and its corollaries, a systematic procedure for periodizing a given fixed machine can be developed. Such a procedure involves the tabulation of the loop lengths and calculation of the common prime factor. If the period of the periodic machine has been fixed by other considerations (e.g. available clock mechanism, etc.), then the procedure will quickly specify the allowable periods. If the period has not been previously specified, then the period may be selected; in general, the greatest common divisor (gcd) should be chosen, to minimize the memory capacity required at any given time. The procedure described above is tabulated in Table I, which forms a convenient aid in periodizing machines of up to T = 8 and containing fewer than 6

I CALCULATION OF PERIOD

1.	LOOP PERIODS							
2.	COMMON DIVISORS							
3.	COMMON PRIME FACTORS							
4.	ALLOWABLE PERIODS							

II SYNTHESIS

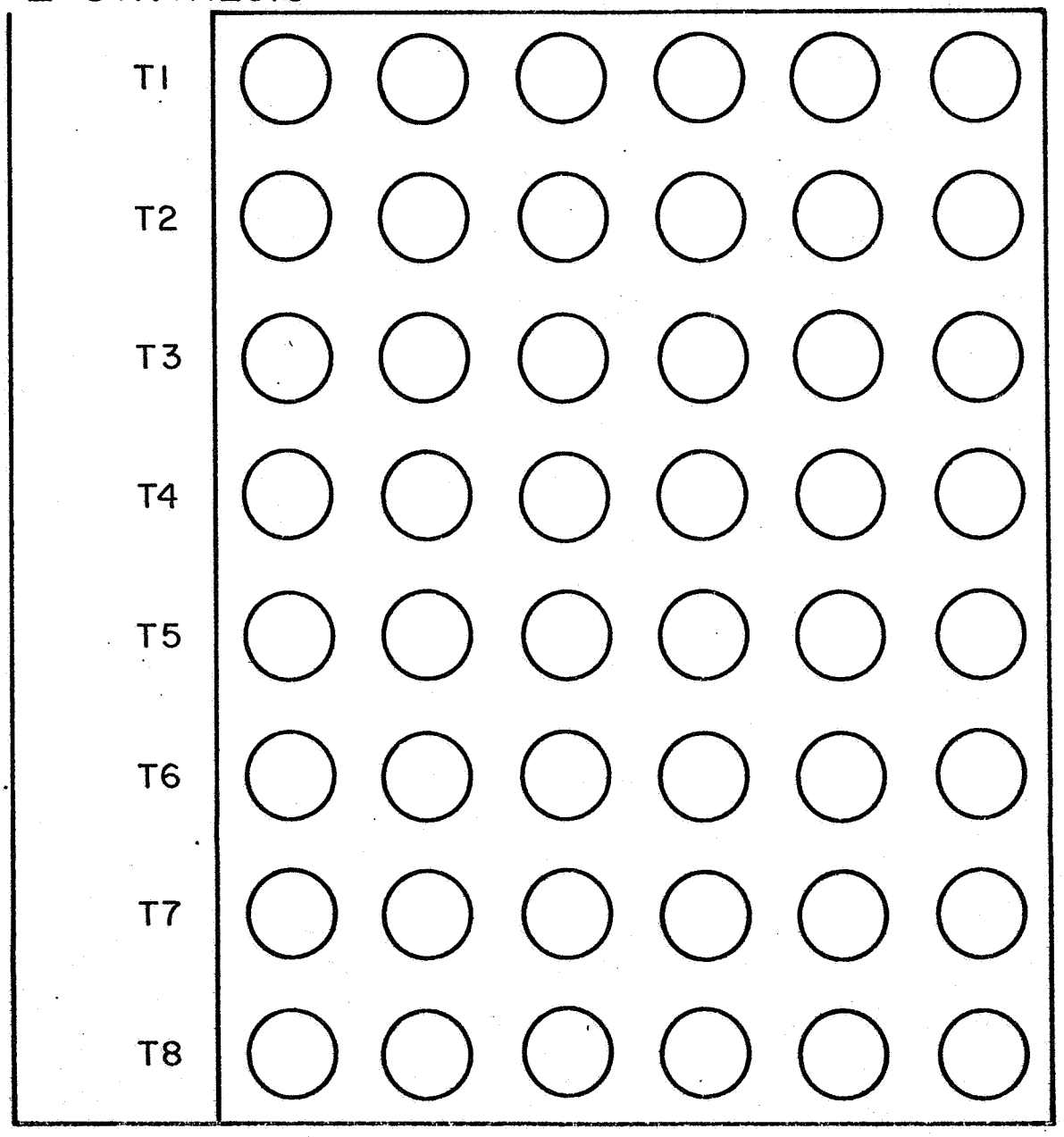


Table 1: Periodic Machine Synthesis

states per state set.

Example: Suppose we are given the problem of designing a pattern detector which behaves as follows:

Sequence	Output, $Z_1Z_2Z_3$
$X_1X_1X_1X_1$	000
$X_1X_1X_1X_2$	110
$X_1X_1X_2X_1$	100
$X_1X_2X_1X_1$	010
$X_2X_1X_1X_1$	001

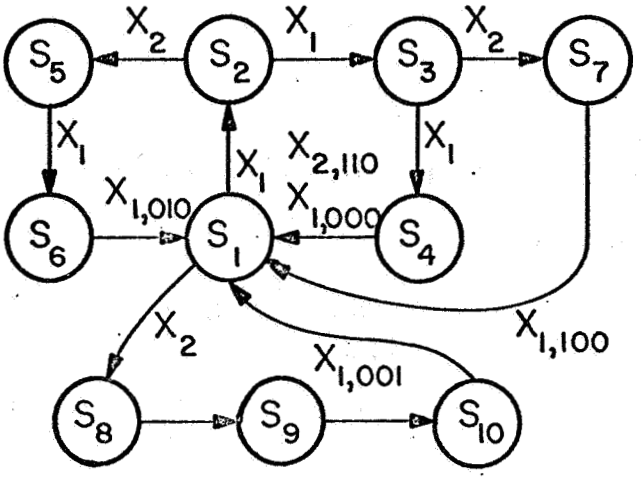
The Z outputs are pulses which occur during the last clock period of the sequence; X_1 and X_2 are pulses such that either one or the other occurs, the $X_1 \cdot X_2$ combination is prohibited.

The State Diagram for such a machine is shown in Figure 18. Analysis by standard techniques [4] reveals that no states are redundant. Four R-S Flip-flops plus supporting logic as shown in Figure 20, are required for implementation.

For the periodization of the machine, all loops are of period 4, so the period T of the equivalent periodic machine is chosen as 4. The periodic machine is shown in Figure 19; its implementation requires but two flip-flops, and is shown in Figure 21. Again, it is assumed for the periodic machine that a four phase clock is available in the system; if one must be generated, much of the savings in elements is lost in the generation of the multi-phase clock.

E. Extension to Non-Strongly Connected Machines

In order to extend the procedures described in an earlier



NOTE: OUTPUT = $Z_1 Z_2 Z_3$ IS 000 UNLESS OTHERWISE NOTED.

Figure 18: Pattern Detector State Diagram

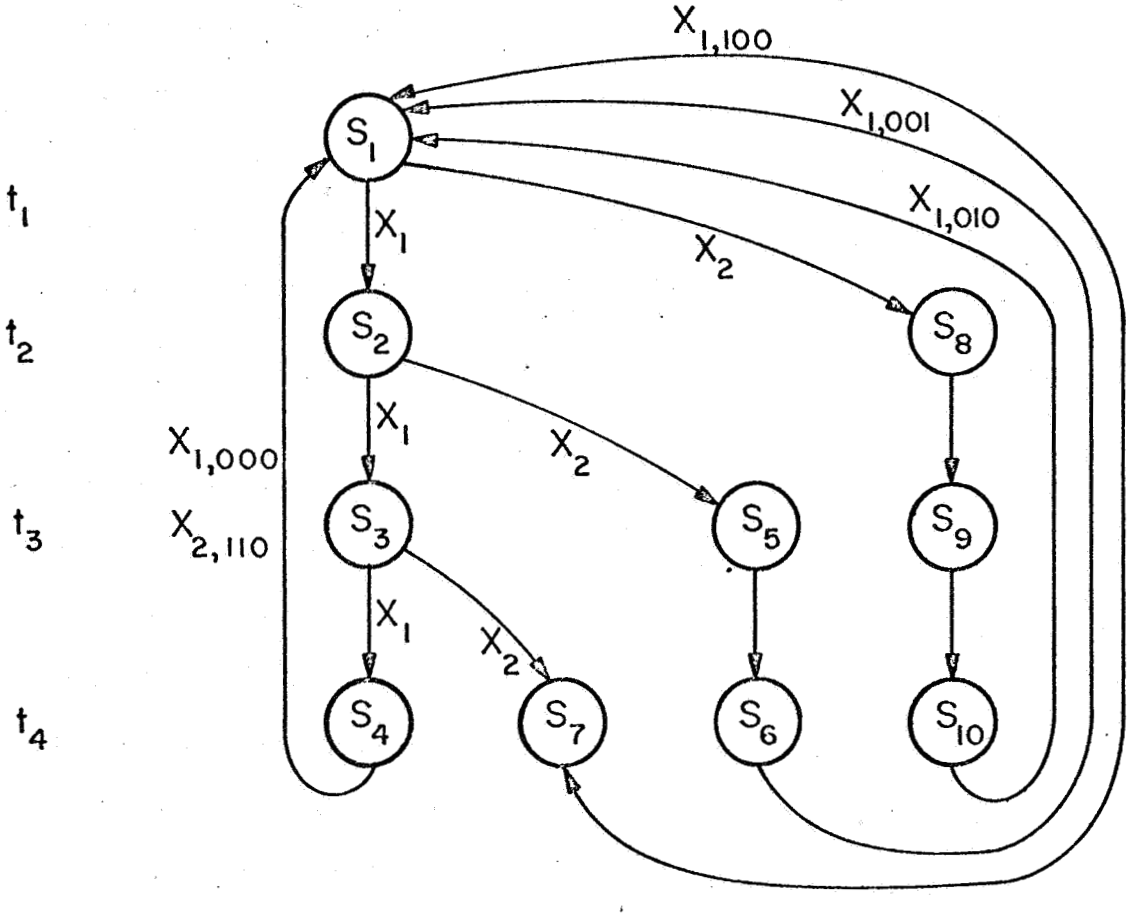


Figure 19: Periodic Machine for Pattern Detector

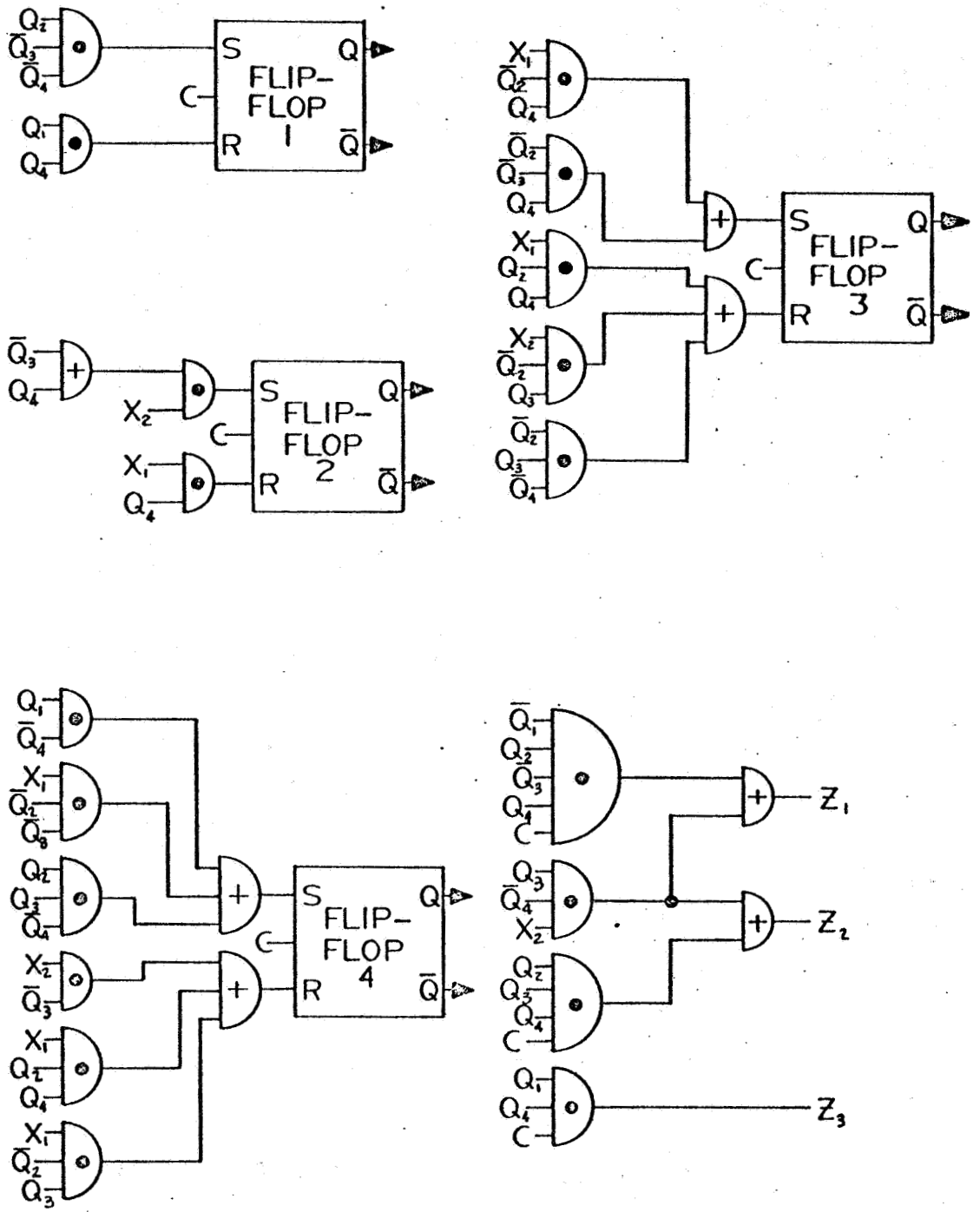


Figure 20: Conventional Machine Implementation

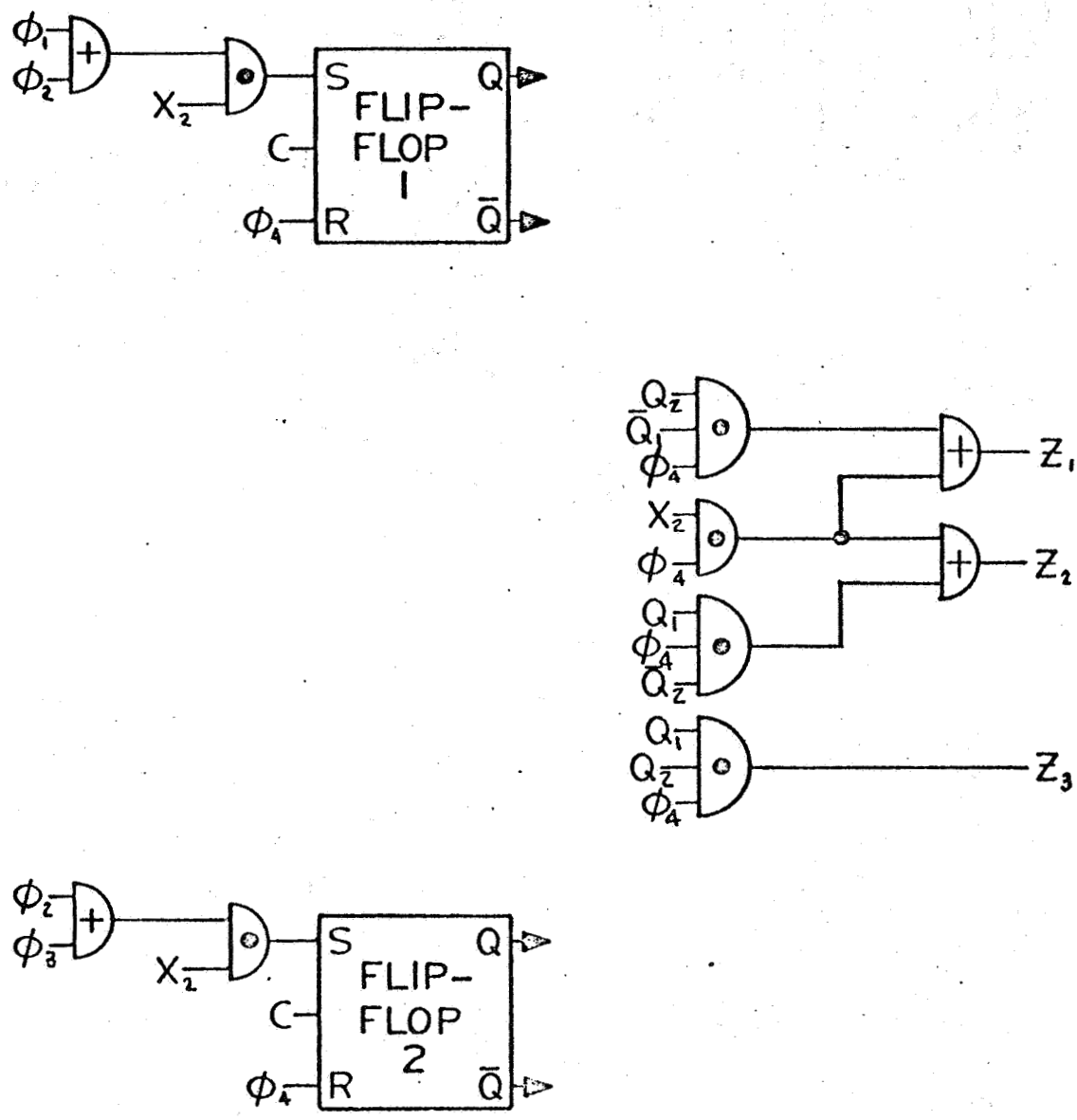


Figure 21: Periodic Machine Implementation

section of this thesis for application to non-strongly connected machines, changes must be made to compensate for the difference in structure of the machine.

The typical non-strongly connected machine as shown in Figure 22 functions as though it were several machines, with linkages between machines, some of which are unilateral. When a machine of this type is to be periodicized, an attempt to apply the procedure developed earlier should be made. If, upon application of the procedure, the conclusion that the machine is only trivially periodicizeable is reached, then the machine must be re-examined. If it consists of several machines, weakly connected, some of which are periodicizeable, then the periodization can be accomplished using the common prime factor of some of the machines. The periodization will be trivial for the other machines.

Example: When the machine of Figure 22 is to be periodicized, tabulation of the loops reveals two relatively prime factors, 3 and 2. Since the bulk of the machine consists of loops of period 3 or multiples thereof, periodization is attempted with T = 3, and is shown in Figure 23. The portion representing states S_4 and S_5 of the fixed machine can be seen to occur in every state set, but the periodization is non-trivial in the larger sense, inasmuch as the other states do not occur in every state set.

Hence, we see that the concepts evolved in the body of this Thesis still apply, with modification, to non-trivial connected machines.

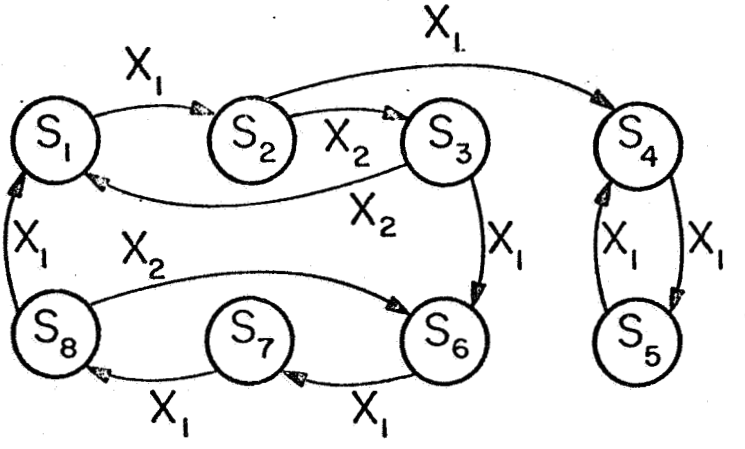


Figure 22: Non-Strongly Connected Machine

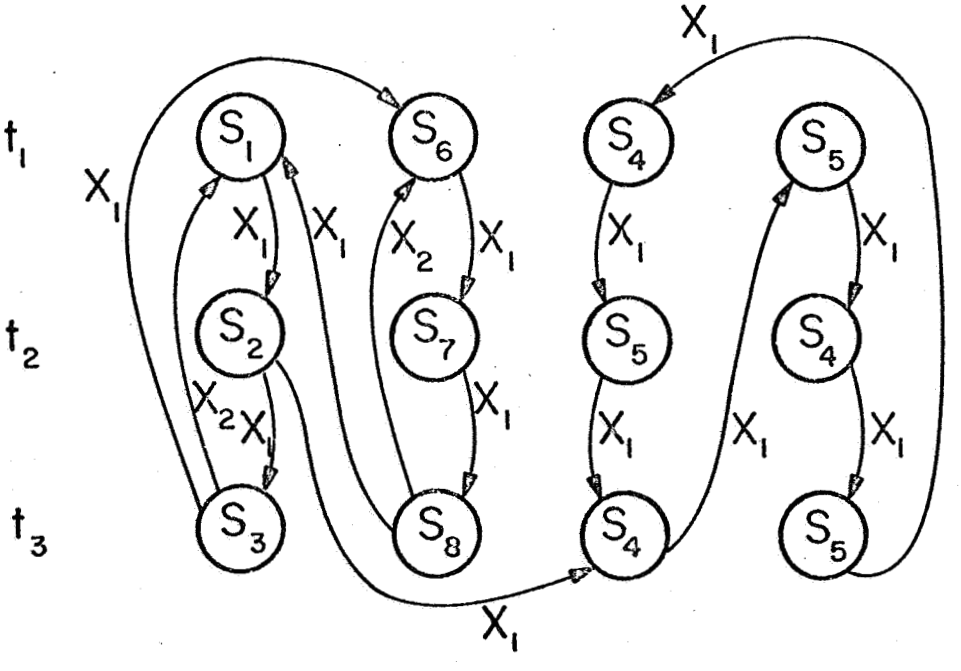


Figure 23: Periodization of Non-Strongly Connected Machine

F. Extension to Machines Having a Transient Set of States

Inasmuch as machines having a transient set of states can be considered a class of non-strongly connected machines, it is expected that the same modified procedures used to extend the periodization of strongly connected machines to the non-strongly connected case could be utilized. This is indeed the case; we periodicize the strongly-connected portion, and then include states equivalent to the transient states. Such states are added in state sets such that the transitions are compatible with the rest of the periodic machine. If the transient states cannot be assigned to any particular state set due to other system constraints, then the periodization will result in a trivial periodic machine, since the transient state must then appear in each state set.

Example: The machine in Figure 24 has a group of transient states, S_1, S_2 and S_3. The remainder of the machine can be periodicized with T = 3, as shown in Figure 25. When the transient states are added, S_1 and S_2 are added in appropriate state sets. S_2 must be in a state set succeeding the state set to which S_1 is assigned, but the state set succeeding that of S_2 must contain a state equivalent to S_4. This necessitates the addition of another $\{S_4 - S_9\}$ to acccommodate S_2.

Hence, not all machines with transient states can be periodicized in a meaningful manner, since the required transitions from the transient states to the remainder of the machine may force additional sets of states to be added such that a trivial or near-trivial machine results. A good "rule of thumb" is to count mod T, the

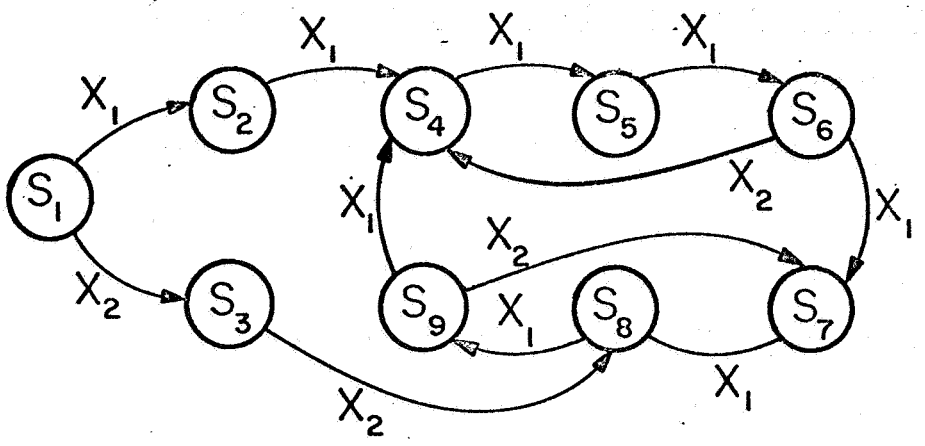


Figure 24: Machine with Transient States

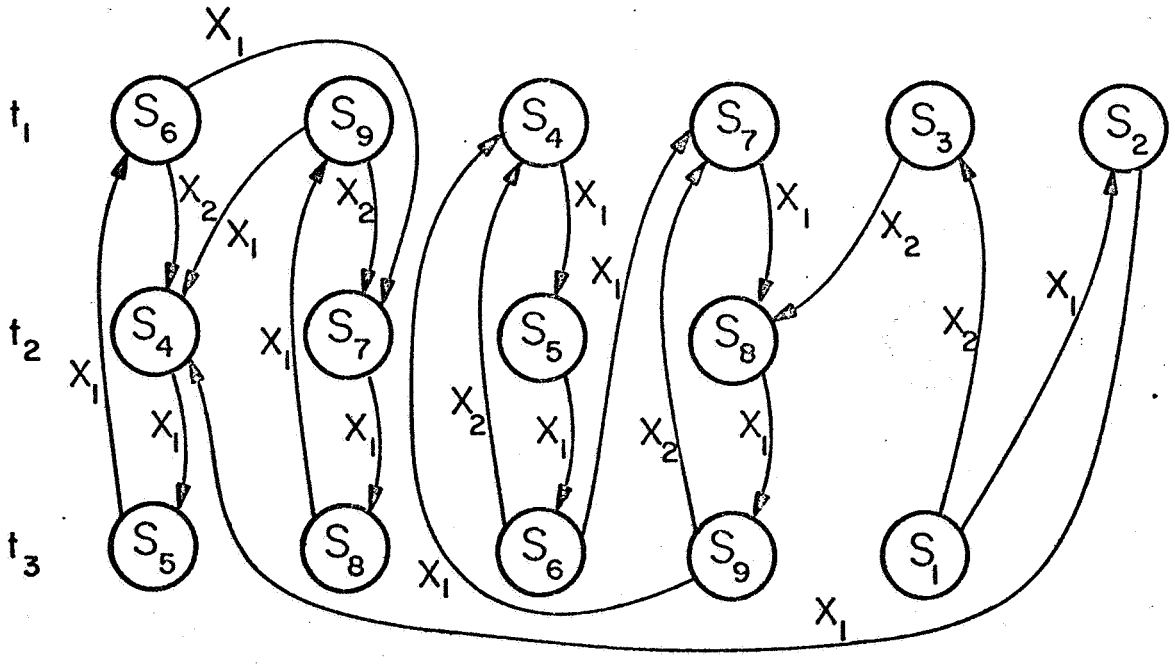


Figure 25: Periodization of Machine with Transient States

number of transitions from the beginning transient state to some arbitrary state of the strongly connected portion of the machine. If the count differs along the various transient paths, additional groups of states will be necessary when the machine is periodicized.

G. Extension to Non-Minimal Machines

When the extension of the synthesis techniques described in the previous sections to non-minimal machines is investigated, the nature of the minimization process must be considered. In order for two states S_i and S_j of a machine to be equivalent, equivalent sequences of states and corresponding outputs must result from any permitted sequence of inputs applied to the machine when starting in either S_i or S_j. Viewed in the terminology developed in this Thesis, this equivalence corresponds to the requirement that the loop lengths of loops containing equivalent states be identical, and have common start and finish points.

Example: The machine of Figure 26 contains two states, S_1 and S_3, which are equivalent. The sequence of states S_2, S_1, S_4 has an equivalent sequence including S_3, i.e. S_2, S_3, S_4. The minimized four state machine has only one path between S_2 and S_4.

Since equivalence of states thus can be said to imply equivalence of loop lengths, the theorem and corollaries developed previously can be extended to non-minimal machines. A loop of length n in a non-minimal machine must still exist in the minimized version, and thus the restrictions imposed upon the periodization process by the existence of that loop is independent of the condition of minimality.

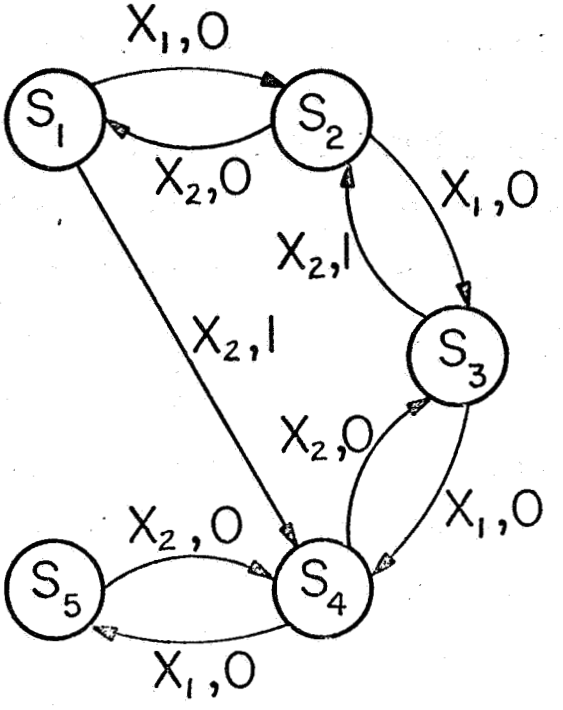


Figure 26: Non-Minimal Machine

Thus, when a non-minimal machine is considered for periodization, the same procedures may be utilized. The number of states in the periodic machine which results may be non-minimal, however.

III. CONCLUSIONS

The foundations upon which practical design of periodic machines depends have been investigated, and theorems derived which quickly and accurately predict the outcome from periodization of a fixed machine.

Machines with one or more loops of period 1 were found to be poor candidates for periodization, inasmuch as the only periodization is the trivial one. Machines where the periods of the loops are relatively prime are likewise poor candidates.

In summary, the prerequisites for useful periodization are:

1) Availability of timing source

2) Loops of machine compatible with available timing

3) No loops of period 1

4) No multiple loops having relatively prime factors

Under these circumstances, a substantial saving of logical elements can be achieved.

It appears that further research is warranted into minimization techniques suitable for the periodization process. For example, if the state diagram shown in Figure 27 had been chosen for the periodization of the pattern detector discussed previously, a more complicated implementation would have resulted. This is left as a future research topic.

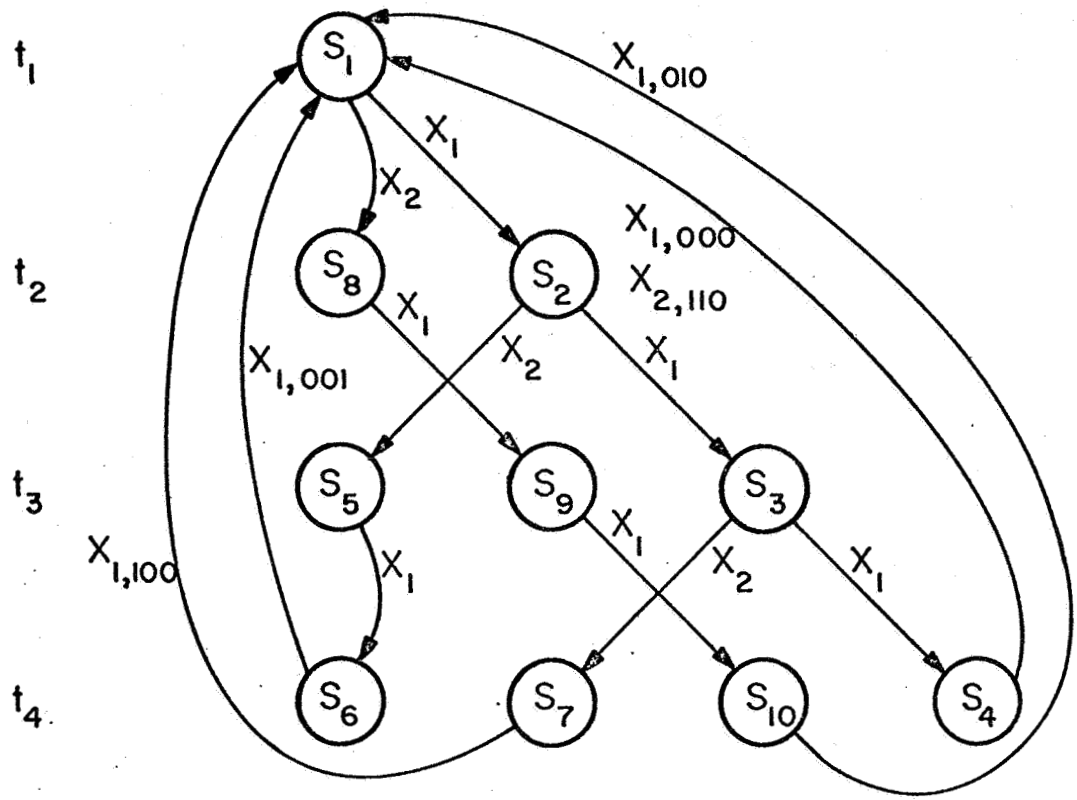


Figure 27: Alternate Pattern Detector State Diagram

SELECTED BIBLIOGRAPHY

1. Gill, A., "Time Varying Sequential Machines," Journal of the Franklin Institute, Vol. 247, No. 6, December 1963.

2. Gill and Flexer, "Periodic Decomposition of Sequential Machines," Journal of the ACM, Vol. 14, No. 4, October 1967.

3. Meisel, W. S., "Variable-Threshold Threshold Elements," Ph.D. Dissertation, University of Southern California, June 1967, pp. 137-142.

4. McCluskey, E. J., "Introduction to the Theory of Switching Circuits," McGraw-Hill, 1965, pp. 248-261.

5. Ware, Willis H., "Digital Computer Technology and Design," Vol. II, John Wiley and Sons, 1963, p. 13.11.

6. Birkhoff and McLane, "A Survey of Modern Algebra," MacMillan Company, 1963.