KYBERNETIKA ČÍSLO 4, ROČNÍK 3/1967

Memory Elements and Partition Pairs in the Synthesis of Sequential Circuits

RUDOLF NOVANSKÝ

The article states some new results for the coding of internal states of sequential machines with whose realization memory elements of general types in the memory part of corresponding synchronous sequential circuit are used. In the investigation we apply the concept of partition pairs originally introduced by Stearns and Hartmanis [1]. On the basis of these results the algorithm is designed (programmable for a digital computer), which allows to establish the suitable coding of internal states for a given sequential machine. Furthermore, the process allows to determine which memory elements lead to the economical realization of the transient function of a given machine. The article is a continuation of the paper [6].

1. INTRODUCTION

The article deals with the synthesis of sequential machines by means of synchronous sequential circuits whose memory part contains memory elements of general types. It employs the so-called partition pairs, which — employed for coding of internal states — lead to the reduction of dependences in corresponding realization of a sequential machine. There is demonstrated the connection between the reduced dependences of inputs of general memory elements and the introduced partition pairs of type (D, R). The derived results are then employed for the creating of algorithm, by means of which there is formed for a given sequential machine — a set of partition pairs, which can singly lead to maximally reduced dependences of inputs of memory elements. From these partition pairs it is already possible to state the required types of memory elements. In the article we presuppose that the state reduction for the given sequential machine has been already carried out, and only the properties of its transient function are being investigated. At the same time we proceed from the presumption, that the most economical realization of the transient function of a given sequential machine can be obtained, if the dependences of the inputs of memory elements on the outputs from other memory elements are reduced at maximum. Even when we confine ourselves to the investigation of completely specified sequential machines, the stated results are also applicable for incompletely specified ones, too.

2. PROCESS OF SYNTHESIS OF SEQUENTIAL CIRCUITS

The elementary process of the synthesis of sequential machines has been described in [6]. Therefore we are going to repeat here only the most necessary terms and the symbolics that we shall employ further on. In the article we shall confine ourselves only to the synthesis of transient functions of sequential machines. Therefore it is sufficient when – further on – by a sequential machine $\mathscr{S}$ we shall mean a triad $\mathscr{S} = \langle J_{\mathscr{S}}, Q_{\mathscr{S}}, \delta_{\mathscr{S}} \rangle$ where $J_{\mathscr{S}}$ denotes a nonempty set of input symbols $\{I_1, I_2, \ldots, I_m\}$, $Q_{\mathscr{S}}$ a nonempty set of state symbols $\{q_1, q_2, \ldots, q_n\}$, and where m and n

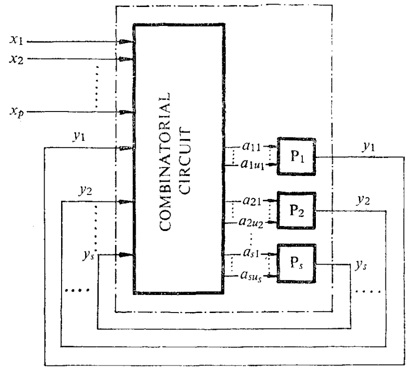

Fig. 1.

are finite integers denoting the numbers of input, and state symbols, respectively, $\delta_{\mathscr{S}}$ is a transient function of a sequential machine $\mathscr{S}$, which determines the mapping

$$Q_{\mathscr{S}} \times J_{\mathscr{S}} \xrightarrow{\delta_{\mathscr{S}}} Q_{\mathscr{S}} .$$

The transient function $\delta_{\mathscr{S}}$ prescribes the internal state of a machine $\mathscr{S}$ at the next time interval q^{t+1} according to the equation

$$q^{t+1} = \delta_{\mathscr{S}}(q^t, I^t) \tag{1}$$

if the internal state and the input of a machine $\mathscr{S}$ at present time interval is q^t, and I^t respectively.

The realization of a sequential machine $\mathscr{S}$ by a sequential circuit is presupposed in canonical form according to Fig. 1 (in which it is employed $s = \lceil \log_2 n \rceil$ of memory elements $P_1, P_2, \ldots, P_s$) to which we shall come by coding the input, state and output symbols of a sequential machine $\mathscr{S}$. Hereby the symbols of a machine $\mathscr{S}$ $I_1, I_2, \ldots, I_m$ are coded by means of $p = \lceil \log_2 m \rceil$ binary input variables $x_1, x_2, \ldots, x_p$, and $q_1, q_2, \ldots, q_n$ are coded by means of $s = \lceil \log_2 n \rceil$ binary state variables $y_1, y_2, \ldots, y_s$, where $\lceil a \rceil$ denotes the least integer, greater than or equal to a.

By coding input and state symbols we obtain – from the equation (1) of a sequential machine $\mathscr{S}$ the equations

$$(2) \qquad Y_i = F_i(y_1, y_2, \ldots, y_s, x_1, x_2, \ldots, x_p)\,, \quad i = 1, 2, \ldots, s\,,$$

and they are the dependences of binary state variables $Y_i = y_i(t + 1)$ $(i = 1, 2, \ldots \ldots, s)$ at time $t + 1$ on binary state variables $y_j = y_j(t)$ $(j = 1, 2, \ldots, s)$ and binary input variables $x_k = x_k(t)$ $(k = 1, 2, \ldots, p)$ at time t.

The equations (2) of a sequential circuit already prescribe in Fig. 1 the required behaviour of sequential circuit's part plotted by a line, and then also the outputs of particular memory elements $P_1, P_2, \ldots, P_s$. Therefore it is necessary to determine the types of particular memory elements P_i $(i = 1, 2, \ldots, s)$ and the dependence of excitation of their inputs a_{il} $(l = 1, \ldots, u_i)$.

$$(3) \qquad a_{il} = f_{il}(y_1, y_2, \ldots, y_s, x_1, x_2, \ldots, x_p),\ i = 1, 2, \ldots, s\,,$$

on input variables of a combinatorial circuit y_j $(j = 1, 2, \ldots, s)$ and x_k $(k = 1, 2, \ldots \ldots, p)$ so, that the prescribed equations (2) may be satisfied. The questions connected with the choice of memory elements P_i, and with the determination of their inputs a_{il} have been discussed in the sufficient manner in [6] (see also [7]), and will not be repeated here.

3. PARTITION PAIRS AND REDUCED DEPENDENCES

In this part we introduce the concept of partition pair originally defined by Stearns and Hartmanis [1]. Some of its properties will be investigated, and its connection with the reduced dependences of switching functions a_{il} in (3) on input variables $x_1, x_2, \ldots, x_p$ and on the subset of state variables $y_1, y_2, \ldots, y_s$ will be stated.

It was demonstrated in [2] and [6] that the so called partition with substitution property facilitate finding the coding with selfdependent subset in which the dependences (2) of binary state variables at time $t + 1$ on the state variables at time t are reduced as follows

$$(4) \qquad Y_i = F_i(y_1, y_2, \ldots, y_b, x_1, x_2, \ldots, x_p)\,, \quad i = 1, 2, \ldots, b\,, \quad b < s\,.$$

Then we look for inputs (3) of the corresponding memory elements also in the form

$$(5) \quad a_{il} = f_{il}(y_1, y_2, \ldots, y_b, x_1, x_2, \ldots, x_p)\,, \quad l = 1, 2, \ldots, u_i\,; \quad i = 1, 2, \ldots, b\,.$$

Not all sequential machines have partitions with S. P. which make it possible to find the coding with selfdependent subsets of binary state variables, and even when such partitions exist, they need not always lead to the greatest reduction of the dependences of all binary variables Y_i $(i = 1, 2, \ldots, s)$.

Therefore, Stearns and Hartmanis [1] introduced partition pairs (P. P.) which are the generalization of partitions with S. P. Partition pairs used for the coding of the states of a sequential machine lead to more general reductions of the dependences of binary state variables at time $t + 1$ on the state variables at time t. Let us show some properties of partition pairs that we are going to utilize.

Foremost we recapitulate a notion of a partition.

Definition 1. The partition π on a set Q is a collection of disjoint subsets Q_α the union of which is a set Q. The disjoint non-empty subsets Q_α contained in this collection will be called blocks of partition π.

It holds a series of relations for the computation with partitions. These relations have been indicated in [3] (see also [6]).

Further we shall understand that the set Q is the set of states of a sequential machine $\mathscr{S}$.

Definition 2. An ordered couple of partitions (π, π') on the set of states Q of a machine $\mathscr{S}$ be called the partition pair, if for any two states q_i and q_j from the same block $Q_\alpha \in \pi$, and for any input I_k, states $\delta(q_i, I_k)$ and $\delta(q_j, I_k)$ fall into the same block $Q_\beta \in \pi'$.

From this definition – as a special case – we obtain a partition with S. P., if it is true $\pi = \pi'$. We introduce for P. P. (partition pairs) partial ordering.

For two given partition pairs (π_1, π'_1) and (π_2, π'_2) on the set Q of a machine $\mathscr{S}$ we mark $(\pi_1, \pi'_1) \leqq (\pi_2, \pi'_2)$ if and only if it is $\pi_1 \leqq \pi_2$ and $\pi'_1 \leqq \pi'_2$.

The formation of new partition pairs from the partition pairs already known is enabled by the following theorem [1].

Theorem 1. *If (π_1, π'_1) and (π_2, π'_2) are two partition pairs on the set of states Q of a machine $\mathscr{S}$, then partition pairs are also couples of partitions $(\pi_1 . \pi_2, \pi'_1 . \pi'_2)$ and $(\pi_1 + \pi_2, \pi'_1 + \pi'_2)$.*

It is evident that couples of partitions which result from a partition pair (π, π') by the refinement of a partition π, or by the enlargement of a partition π' are partition pairs, too. For simplification it is advantageous to consider for a given partition π only the smallest partition π', – let us mark it $m(\pi)$ – so that $(\pi, m(\pi))$ is a partition pair. Similarly for a given partition π' it is to advantage to consider only the largest partition π – marked $M(\pi')$ – so that $(M(\pi'), \pi')$ is a partition pair. Then for a given partition π on the set Q of a machine $\mathscr{S}$, $m(\pi) = \prod_i \pi'_i$, where the product is over all such i that (π, π'_i) is a P. P. Similarly for a given partiton π' on the set of states Q of a machine $\mathscr{S}$, $M(\pi') = \sum_i \pi_i$, where the sum is over all such i that (π_i, π') is a partition pair (P. P.).

It is to advantage to establish for a given machine $\mathscr{S}$ the so-called Mm pairs because it is possible to determine from them all the partition pairs of a machine $\mathscr{S}$. The Mm pairs give us – in a brief form – the total information on the structural properties of a sequential machine $\mathscr{S}$.

Definition 3. A partition pair (π, π') on the set of states Q of a machine $\mathcal{S}$ will be called a Mm pair if and only if it is $\pi = M(\pi')$ and $\pi' = m(\pi)$.

For the computation with partition pairs, and esspecially with Mm pairs, Stearns and Hartmanis derived in [1] a series of relations, further on some of which we are going to utilize.

For a given partition pair (π, π') on the set of states Q of a machine $\mathcal{S}$ let $\beta(\pi)$, $\beta(\pi')$ be marked the corresponding numbers of partition blocks, and let $\gamma(\pi)$, $\gamma(\pi')$ be marked the numbers of states in the largest ones of blocks of π and π', respectively. The smallest number of binary state variables needed for the coding of the states of a machine $\mathcal{S}$ is $s = [\log_2 n]$. From the knowledge of the block of π in which there is included the present state of a machine S, and from that of input we can determine a block of π', which will contain the state of a machine S at the next time interval. That means that if we employ the first $b_2 = [\log_2 \beta(\pi')]$ variables $y_1, y_2, y_3, \ldots, y_{b_2}$ for the discrimination between blocks of π', and some $b_1 = [\log_2 \beta(\pi)]$ state variables $y_{j_1}, y_{j_2}, \ldots, y_{j_{b_1}}$ (among which there can be some of the set of variables $y_1, y_2, \ldots$ $\ldots, y_{b_2}$) for the discrimination between blocks of π, then the values of b_2 state variables $Y_1, Y_2, \ldots, Y_{b_2}$ at time $t + 1$ will be dependent only on the values of b_1 variables $y_{j_1}, y_{j_2}, \ldots, y_{j_{b_1}}$ and on inputs at time t. Then, we shall obtain the following dependences of binary state variables Y_i

$$Y_i = F_i(y_{j_1}, y_{j_2}, \ldots, y_{j_{b1}}, x_1, x_2, \ldots, x_p), \quad i = 1, 2, \ldots, b_2 . \tag{6}$$

If we want to obtain the reduced dependences of binary state variables and at the same time the number of those variables shall not exceed $s = [\log_2 n]$ for a n-state machine $\mathcal{S}$, then the general collection of partition pairs applicable to such a coding of states, must satisfy the rather complicated system of necessary conditions derived by Karp [5].

That is why it is advantageous to confine oneself further only to such partition pairs (π, π'), in which π' is non-trivial and it is not contained in any other non-trivial partition (i.e. it contains precisely two blocks) [4]. Such π' is called primary and is marked R.

Let the values of binary state variables $y_1, y_2, \ldots, y_s$ determine the individual states of a machine $\mathcal{S}$. Then every state variable y_i $(i = 1, 2, \ldots, s)$ qualifies a double block partition R_i on the set of the states Q of a machine S. A block of partition R_i involves the states of a machine $\mathcal{S}$ which have the same value of a variable y_i assigned. In [1] it was demonstrated that a necessary and sufficient condition for the goal that a variable Y_i at time $t + 1$ may dependent only on variables $\{y_j / j \in K_i\}$ and on inputs at time t is the condition

$$\prod_{j \in K_i} R_j \leqq M(R_i) , \tag{7}$$

where K_i is some subset of an index set $\{1, 2, \ldots, s\} = K_0$.

The relation (7) is called an inequality of the information flow. $M(R_i)$ is the largest partition (the smallest "amount of information") block of which contains the internal state of a machine $\mathscr{S}$ at time t. From the knowledge of this block and input I we can identify the partition block of R_i wherein the state of a machine will be contained at time $t + 1$. Inequality says that for the computation Y_i of the next state we must have at least as "much information" on a present state of machine $\mathscr{S}$ as much it is included in $M(R_i)$. Then, for a state variable Y_i we may write

(8) $$Y_i = F_i(y_{j_1}, y_{j_2}, \ldots, y_{j_{b_i}}, x_1, x_2, \ldots, x_p),$$

where

(9) $$K_i = \{j_1, j_2, \ldots, j_{b_i}\} \subseteq K_0$$

and

(10) $$b_i = \log_2 \beta\,(\prod_{j \in K_i} R_j)\,.$$

Let us note that $\prod_{j \in K_i} R_j$ in inequality (7) determines a partition $D_i = \prod_{j \in K_i} R_j$ on the set of states Q of a machine $\mathscr{S}$, so that (D_i, R_i) is a partition pair. (Note that (D_i, R_i) is a trivial partition pair, if D_i is a null partition $(D_i = 0)$.)

4. MEMORY ELEMENTS AND PARTITION PAIRS

Further on we shall show that the substantial relation exists between diminishing of dependences of inputs $a_{i1}, a_{i2}, \ldots, a_{iu_i}$ to a general memory element P_i and a non-trivial partition pair (D_i, R_i). We shall characterize this relation in some theorems which will state the necessary and sufficient conditions for the maximum reduction of dependences of inputs $a_{i1}, a_{i2}, \ldots, a_{iu_i}$ to an element P_i on state variables $y_1, y_2, \ldots \ldots, y_s$. On the basis of these theorems we are going to state an algorithm for the determination of the partition pairs, which will lead to the reductions demanded.

Theorem 2. *Let $\mathscr{S}$ be a sequential machine whose n-internal states are uniquely represented by the values of $s = \lceil \log_2 n \rceil$ binary state variables $y_1, y_2, \ldots, y_s$ (which are the values of the outputs of memory elements $\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_s$). Let us presume that the binary inputs $a_{i1}, a_{i2}, \ldots, a_{iu_i}$ of a memory element P_i are capable of being computed from inputs $x_1, x_2, \ldots, x_p$ and from the set $b_i < s$ of state variables $\{y_j | j \in K_i\}$. Then there exists a partition pair (D_i, R_i) so that for D_i it holds*

(11) $$D_i = B_i\,,$$

or

(12) $$D_i = B_i \,.\, R_i\,,$$

where B_i is a partition on the set of states Q of a machine $\mathscr{S}$, which identifies into one block all internal states differing only in the remaining $s - b_i$ state variables, where

$$\beta(B_i) \leqq 2^{b_i} \tag{13}$$

is the number of blocks of B_i.K_i, which has b_i elements, is some proper subset of index set $K_0 = \{1, 2, \ldots, s\}$, and R_i is the partition determined by the output y_i of a memory element P_i.

Proof. In accordance with the hypothesis, the inputs $a_{i1}, a_{i2}, \ldots, a_{iu_i}$ of an element P_i can be computed without the knowledge of $s - b_i$ state variables. Let us form the subsets of internal states of a machine $\mathscr{S}$ so that all the states, differing only in the given $s - b_i$ state variables, are placed in the same subset. Then, on the set of states Q of a machine $\mathscr{S}$ we obtain a partition that we call B_i. It is not difficult to show that for the number of partition blocks $\beta(B_i)$ there holds the relation (13). Likewise a partition R_i is to be obtained so that in the same subset there are placed all the states that differ in other state variables than y_i. Furthermore, from the characteristic equation of a memory element (1) in [6] it is to be seen that its output Y_i generally depends on binary inputs $a_{i1}, a_{i2}, \ldots, a_{iu_i}$, and on its internal state variable y_i. The inputs $a_{i1}, \ldots, a_{iu_i}$ of an element P_i are unambiguously determined by a partition B_i, and a state variable y_i is determined by a partiton R_i. Then we distinguish the following four cases:

1. This output Y_i of a memory element P_i does not depend on its internal variable y_i and $i \notin K_i$. Then a state variable Y_i is unambiguously determined by the input of a circuit I_k, $k = 1, 2, \ldots, m$, and by the collection of state variables $\{y_j/j \in K_i\}$. That means, that a partition B_i provides "sufficient information" also for computation Y_i, i.e. (B_i, R_i) is a partition pair. In this case $D_i = B_i$.

2. The output Y_i of a memory element P_i does not depend on its internal variable y_i, and $i \in K_i$. For the same reasons as in the case 1 (B_i, R_i) is a partition pair, too. That denotes that also in this case $D_i = B_i$.

3. The output Y_i of a memory element P_i is non-trivially dependent on its internal variable y_i and $i \notin K_i$. Then a state variable Y_i is unambiguously determined by the input of a circuit I_k $(k = 1, 2, \ldots, m)$ and by the set of state variables $\{y_j/j \in \{i\} \cup \cup K_i\}$. If we form the subsets of the states of a machine $\mathscr{S}$ so that in the same subset there are placed all the states that differ only in variables $y_j, j \notin \{i\} \cup K_i$, we obtain a partition on the set of states Q of a machine $\mathscr{S}$ which be called D_i. The partition D_i then provides "sufficient information" for a computation Y_i, i.e. (D_i, R_i) is a partition pair. On the other hand it is evident that in this case for a computation Y_i there is needed "information" which is contained both in a partition B_i, and in a partition R_i. That denotes that $D_i = B_i \,.\, R_i$.

4. The output of a memory element P_i is non-trivially dependent on its internal variable y_i and $i \in K_i$. Then the state variable Y_i is unambiguously determined by the input of a circuit I_k $(k = 1, 2, \ldots, m)$, and by the set of the state variables $\{y_j/j \in \in \{i\} \cup K_i\}$. But in this case $\{i\} \cup K_i = K_i$. That denotes that a partition B_i offers to us "sufficient information" for a computation Y_i, i.e., (B_i, R_i) is a partition pair. Then $D_i = B_i$. Q.E.D.

Let us go on discussing the four cases of the dependences of the output Y_i of a memory element P_i on the subsets of state variables above-mentioned in the proof of the theorem 2 regarding the theorem 4.1 in [6]:

1. The memory element P_i for Y_i is properly chosen. By choosing another element P'_i which would have a non-trivial internal feedback we should not achieve a further reduction of the dependences of binary inputs $a'_{i1}, \ldots, a'_{iu_i}$ on state variables $y_j, j \in K_i$. Let us comment that in this case there is $B_i . R_i < D_i$, i.e. the "information" contained in a partition $B_i . R_i$ is greater than the "information" needed for the computation of Y_i.

The two cases 2 and 4 can be investigated together. In both these cases the memory element P_i for Y_i is not properly chosen. According to the theorem 4.1 in [6] it can be replaced by another element P'_i, which has a non-trivial internal feedback so that binary inputs $a'_{i1}, \ldots, a'_{iu_i}$ may be independent on a state variable y_i. In this case there is $B_i = B_i . R_i = D_i$, i.e., in the assignment of a partition B_i we can exclude a variable y_i, when we keep the equality $B'_i . R_i = D_i$.

3. The memory element P_i for Y_i is properly chosen. By choosing another element P'_i we could only retain or enlarge the dependences of binary inputs $a'_{i1}, \ldots, a'_{iu_i}$ on state variables y_j. (The number of dependences would be enlarged by using an element without internal feedback.) In this case no other but a partition $B_i . R_i = D_i$ gives "sufficient information" for the computation of Y_i, because $B_i > D_i$.

In all these cases $(1-4)$ we were considering only such modifications of a partition B_i to a partition B'_i, which were leading to further reductions of the dependences of binary inputs $a'_{i1}, \ldots, a'_{iu_i}$ of an element P'_i on state variables, y_j, and which did not change a partition pair (D_i, R_i). These modifications having been performed, the cases 2 and 4 will turn into the case 3.

Now let us presuppose that the element P_i has been chosen so that the dependences of its binary inputs $a_{i1}, \ldots, a_{iu_i}$ on state variables y_j $(j = 1, 2, \ldots, s)$ are minimum. From the previous considerations it follows that there can occur only two cases of relationship between partitions B_i, R_i and D_i for a partition pair (D_i, R_i):

$$\text{(A)} \quad D_i < B_i \,, \quad D_i = B_i . R_i \quad \text{and} \quad b(B_i) = b(B_i . R_i) - 1 = b(D_i) - 1 \,,$$

$$\text{(B)} \quad D_i = B_i \,, \quad D_i > B_i . R_i \quad \text{and} \quad b(B_i) = b(B_i . R_i) - 1 = b(D_i) \,,$$

where $b(x) = [\log_2 \beta(x)]$ and $\beta(x)$ is the number of blocks of partition x.

The theorem 2 guarantees that the reduced dependences of binary inputs $a_{i1}, \ldots$ $\ldots, a_{iu_i}$ of any memory element P_i on state variables in a given machine $\mathcal{S}$ lead to the existence of a partition pair (D_i, R_i) on the set of states Q of this machine. There arises a question: "Whether and how does the existence of a partition pair (D_i, R_i) guarantee the reduced dependences of the inputs $a_{i1}, \ldots, a_{iu_i}$ of a memory element P_i for a given machine $\mathcal{S}$?" It is answered by the following two theorems:

Theorem 3. *Let $\mathcal{S}$ be a sequential machine that has n-internal states. Then there exists such a coding of the internal states of a machine $\mathcal{S}$ by the binary state variables $y_1, y_2, \ldots, y_s$ $(s = [\log_2 n])$, which permits the determination of the binary inputs $a_{i1}, \ldots, a_{iu_i}$ of a memory element P_i (for a state variable y_i) from inputs $x_1, x_2, \ldots, x_p$ and from the set $b(D_i) - 1$ of state variables $\{y_j/j \in K_i\}$, where $i \notin K_i$, in that case if there exists on the set of states Q of a machine $\mathcal{S}$ a partition pair (D_i, R_i) satisfying the following conditions*:

$$c(R_i) = s - 1\,, \tag{14}$$

$$b(D_i) = b(D_i \cdot R_i)\,, \tag{15}$$

$$b(D_i \cdot R_i) + c(D_i \cdot R_i) = s\,, \tag{16}$$

$$b(D_i \cdot R_i) = b(D_i \cdot R_i/R_i) + 1\,, \tag{17}$$

where K_i is a proper subset of an index set $K_0 = \{1, 2, \ldots, s\}$ which has the number of elements $b(D_i) - 1$; R_i is a double block partition on the set of states Q of a machine $\mathcal{S}$, $b(x) = [\log_2 \beta(x)]$ and $\beta(x)$ is the number of partition blocks of x; $c(x) = [\log_2 \gamma(x)]$ and $\gamma(x)$ stands for the number of states in the largest block of a partition x; $b(x/y) = [\log_2 \beta(x/y)]$ and $\beta(x/y)$ represents the greatest number of partition blocks of x involved in one partition block of y.

Proof. Let a partition pair (D_i, R_i) on the set of states Q of a machine $\mathcal{S}$ satisfy all the conditions of the theorem 3. Further on, let us consider a partition pair $(D_i \cdot R_i, R_i)$, that we obtain by the refinement of a partition D_i to a partition $D_i \cdot R_i$. To every internal state of a machine let be assigned the value of a binary state variable y_i in the following way: Let us consider $y_i = 0$ for every state belonging to the first block of R_i and $y_i = 1$ for every state belonging to the second block of R_i. The values of further $b(D_i) - 1$ state variables $\{y_j/j \in K_i\}$ will be assigned to the internal states of a machine $\mathcal{S}$ so that we assign a different configuration of the values of state variables $\{y_j/j \in K_i\}$ to each block of a partition $D_i \cdot R_i$ contained in the first or in the second block of partition R_i. The condition (17) guarantees that it may be done so. Then the values $b(D_i)$ of state variables $\{y_j/j \in \{i\} \cup K_i\}$ uniquely identify each block of partition $D_i \cdot R_i$. The values of the remaining $s - b(D_i)$ state variables $\{y_j/j \notin \{i\} \cup K_i\}$ will be assigned to states of a machine $\mathcal{S}$ so that to each state of one of the partition blocks of $D_i \cdot R_i$ we shall assign a different configuration of the values of these state variables. According to (16) we can do so. The values of these

$s - b(D_i)$ state variables uniquely determine every state inside the given block of the partiton $D_i . R_i$. The values of $c(R_i) = s - 1$ state variables, different from y_i, determine states inside the given partition block of R_i. According to the definition of a partition pair and the above-mentioned coding of states of a machine $\mathcal{S}$ for every input I_k $(k = 1, 2, \ldots, m)$ and for each value configuration $b(D_i)$ of state variables $\{y_j | j \in \{i\} \cup K_i\}$ (connected with a partition block of $D_i . R_i$) at time t there can be uniquely determined the value of a state variable Y_i at time $t + 1$, independently on the values of the remaining $s - b(D_i)$ state variables. Furthermore, from the aforesaid it follows that a state variable Y_i is nontrivially dependent on a state variable y_i. According to the theorem 4.1 in [6] for a variable Y_i we can find a memory element P_i so that its inputs $a_{i1}, a_{i2}, \ldots, a_{iu_i}$ would depend only on inputs $x_1, x_2, \ldots, x_p$, and on $b(D_i) - 1$ state variables $\{y_j | j \in K_i\}$. Q.E.D.

Note. The condition (15) of the theorem 3 guarantees that by investigating a partition pair $(D_i . R_i, R_i)$ instead of a partition pair (D_i, R_i) the number of the dependences of a state variable Y_i on state variables $\{y_j | j \in \{i\} \cup K_i\}$ will not increase. This number of the dependences of a variable Y_i is minimum if in the given partition pair (D_i, R_i) is $D_i = M(R_i)$.

In the previous theorem 3 there are investigated the partition pairs (D_i, R_i) for which $b(D_i . R_i) = b(D_i)$, i.e., only the partition pairs that will lead to the assignment of state variables in which a state variable Y_i depends non-trivially on a state variable y_i. From the discussion to the theorem 2 we know that it is only the case (A). With the dependence reduction of state variables, however, there can also arise a case (B), where $b(D_i . R_i) = b(D_i) + 1$. In this case the following theorem holds:

Theorem 4. *Let $\mathcal{S}$ be a sequential machine having n-internal states. Then there exists such a coding of the internal states of a machine $\mathcal{S}$ by $s = [\log_2 n]$ binary state variables $y_1, y_2, \ldots, y_s$ which permits the determination of the binary inputs $a_{i1}, a_{i2}, \ldots, a_{iu_i}$ of a memory element P_i (for a state variable y_i) from inputs $x_1, x_2, \ldots, x_p$ and from the set of $b(D_i)$ state variables $\{y_j | j \in K_i\}$, where $i \notin K_i$, in that case if on the set of states Q of a machine $\mathcal{S}$ there exists a nontrivial partition pair (D_i, R_i) satisfying the following conditions*:

$$c(R_i) = s - 1\,, \tag{14}$$

$$b(D_i . R_i) = b(D_i) + 1\,, \tag{18}$$

$$b(D_i) + c(D_i) = s\,, \tag{19}$$

$$c(D_i . R_i) = c(D_i) - 1\,, \tag{20}$$

where K_i is some proper subset of an index-set $K_0 = \{1, 2, \ldots, s\}$ which has the number of elements $b(D_i)$. The meaning of the other letters is the same as in the theorem 3.

Proof. Let (D_i, R_i) be a partition pair satisfying all the four conditions of the theorem 4. The values of binary state variable y_i be assigned to the states of a machine $\mathcal{S}$ in the same way as in the proof of the theorem 3. The values of further $b(D_i)$ state variables $\{y_j/j \in K_i\}$ be assigned to the internal states of a machine $\mathcal{S}$ in such a way that various configurations of the values of these $b(D_i)$ state variables correspond to various partition blocks of D_i. Then the values of $b(D_i) + 1$ already assigned state variables $\{y_j/j \in \{i\} \cup K_i\}$ uniquely determine partition blocks of $D_i . R_i$. For the differentiation of states inside the blocks of partition $D_i . R_i$ we assign to the states of a machine $\mathcal{S}$ lying in the same block of partition $D_i . R_i$ various configurations of the values of $c(D_i . R_i)$ state variables $\{y_i/j \notin \{i\} \cup K_i\}$. Then the number of all the state variables assigned by this process is given by $b(D_i) + 1 + c(D_i . R_i)$. With the statisfaction of the condition (20) se get a relation $b(D_i) + c(D_i) = s$ for this number, i.e., a condition (19). That means that a state variable y_i together with further $c(D_i) - 1$ state variables $\{y_j/j \notin \{i\} \cup K_i\}$ (which differ the states inside partition blocks of $D_i . R_i$) form the collection $c(D_i)$ of state variables $\{y_j/j \notin K_i\}$, whose values determine uniquely the states inside a given partition block of D_i. From the foregoin it is obvious that the state of a machine is uniquely determined by the values of $b(D_i) + c(D_i) = s$ binary state variables $\{y_j/j \in K_0\}$. Hereafter, the values $c(R_i) = s - 1$ of state variables, other than y_i, identify states inside given block of a partition R_i. In accordance with the definition of a partiton pair and the above-mentioned coding of the states of a machine $\mathcal{S}$ we can determine the value of state variable Y_i at time $t + 1$ uniquely for every input I_k $(k = 1, 2, \ldots, m)$ and for every value configuration of $b(D_i)$ state variables $\{y_j/j \in K_i\}$ (connected with the states contained in a block of D_i) independently on the values of the remaining $s - b(D_i)$ state variables. From the aforementioned it also follows that a state variable Y_i does not depend on a variable y_i. Then, according to the theorem 4.1 from [6] for a variable Y_i we are not able to find a memory element P_i whose inputs $a_{i1}, a_{i2}, \ldots, a_{iu_i}$ would depend on less that $b(D_i)$ state variables $\{y_j/j \in K_i\}$ (and on inputs $x_1, x_2, \ldots, x_p$). Q.E.D.

Note. If we employ for the coding of the states of a machine $\mathcal{S}$ that a partition pair (D_i, R_i) for which $b(D_i . R_i) = b(D_i) + 1$, then from the proof of the theorem 4 it is evident that for a state variable y_i it is most serviceable to use the memory element of type D that has only one input, and therefore for its input we construct only one Boolean function in a sequential circuit. In this case the results of the theorem 4 are the same as the results derived by Curtis [4].

Further, let us note that for a given machine S the existence of a non-trivial partition pair (D_i, R_i) having the properties mentioned in the theorems 3 or 4 is necessary and sufficient for a further dependence reduction of inputs to a memory element P_i except the reduction of input dependence referred to in [6] (see also [7]). The necessity is substantially stated by the proof of the theorem 2 in combination with the theorem 4.1 from [6]. The theorems 3 and 4 (for two possible cases of the dependence of a state variable Y_i on a variable y_i) prove that this existence is sufficient.

Theorems 3 and 4 yield useful criteria for the assignment of state variables which will lead to the reduced dependences of the inputs of memory elements if the partition pairs (D_i, R_i) for a given machine $\mathscr{S}$ are known. In this place then it is convenient to introduce an algorithm which for any machine $\mathscr{S}$ permits the determination of the set of all nontrivial partition pairs, that lead to the reductions required. (A partition pair is trivial when it leads to no reduced dependences of state variables.) This algorithm can have the following steps:

1. Let us form all two-block partitions R_i on the set of states Q of a machine $\mathscr{S}$ for which $c(R_i) = s - 1$.

2. To each partition R_i let us form all the maximum partitions D_i^l, such ones, that (D_i^l, R_i) is a partition pair.

3. Let us form a partition $D_i^l \,.\, R_i$ to each partiton pair (D_i^l, R_i).

4. To the partition pairs (D_i^l, R_i) satisfying the conditions of the theorem 3 let us select those partition pairs $(D_i^l \,.\, R_i, R_i)$ for which

$$b(D_i^l \,.\, R_i) \leqq s - 1 \,. \tag{21}$$

5. For the partition pairs (D_i^l, R_i), not satisfying the conditions (16) of the theorem 3, we search for all those sort of refinements D_i^{vl} of the maximum partition D_i^l that lead to partition pairs (D_i^{vl}, R_i), satisfying all the conditions of the theorem 3. To thus determined partition pairs we select all those partition pairs $(D_i^{vl} . R_i, R_i)$, for which there holds

$$b(D_i^l) = b(D_i^{vl} \,.\, R_i) \leqq s - 1 \,. \tag{22}$$

6. We select all the partition pairs (D_i^l, R_i) that satisfy the conditions of the theorem 4, and for which $b(D_i^l) \leqq s - 1$.

7. For partition pairs (D_i^l, R_i) not satisfying some of the conditions (19) and/or (20) of the theorem 4 we search for all such refinements D_i^{vl} of a partition D_i^l that lead to partition pairs (D_i^{vl}, R_i) satisfying all the conditions of the theorem 4. From thus determined partition pairs we select all those for which

$$b(D_i^l) = b(D_i^{vl}) \leqq s - 1 \,. \tag{23}$$

The algorithm for the selection of required partition pairs is formulated in such a way, that it may be applicable both for completely and for incompletely specified sequential machines. The difference between completely and incompletely specified machines occurs in realizing the step 2. For completely specified sequential machines there exists – to a given partition R_i – the only maximum partition $D_i = M(R_i)$

such, that (D_i, R_i) is a partition pair. By means of the theorem 1 it is possible to derive the relation

$$M(R_i) = \prod_{k=1}^{m} D_i^k , \tag{24}$$

where D_i^k is such a maximum partition whose blocks will be transformed themselves into the blocks of a partition R_i – under the influence of an input I_k.

Table 1.

n	w
2	1
3	3
4	3
5	15
6	25
7	35
8	35
9	255
10	501

Table 2.

$\mathscr{A}$ \ x_1x_2	00	01	10	11
1	2	2	1	1
2	1	1	4	2
3	4	2	3	3
4	3	1	2	4
5	5	2	4	2
6	6	2	3	4

For incompletely specified sequential machines the relation (24) is not applicable. To a given partition R_i the computation of maximum partitions D_i^l – such ones, that (D_i^l, R_i) are partition pairs – can be realized in accordance with the algorithm referred – to by Karp in [5] in connection with the first method for the computation of critical pairs. (There we put $Q = R_i$.) Further on, for the sake of simplicity, we are going to confine ourselves to completely specified sequential machines.

Note. Partition pairs (D_i^l, R_i) for which $b(D_i^l) = s$, do not lead to reduced dependences, and therefore we eliminate them from further considerations.

The number w of different two-block partitions R_i on the set Q of n-state sequential machine $\mathscr{S}$, for which it holds $c(R_i) = s - 1$, is determined by the relation

$$w = \tfrac{1}{2} \sum_{j=n-2^{s-1}}^{2^{s-1}} \binom{n}{j} ; \quad n \geqq 2 . \tag{25}$$

The corresponding values w for $n = 2 \div 10$ are listed in table 1.

Example 1. The transient function of a machine $\mathscr{A}$ is given by a transient table 2. We are to find the realization of this function by a sequential circuit wherein we make use of 3 memory elements which allow the maximum reduction of the dependences of their inputs on outputs from the other memory elements.

A machine $\mathscr{A}$ has 6 internal states. Then $s = [\log_2 6] = 3$, $w = 25$. Let us choose $R_3 = \{\overline{125}; \overline{346}\}$; $c(R_3) = [\log_2 3] = 2 = s - 1$. From the transient table 2 of a machine $\mathscr{A}$ we establish

$$D_3^{00} = \{\overline{125}, \overline{346}\}, \quad D_3^{01} = \{\overline{123456}\}, \quad D_3^{10} = \{\overline{14}, \overline{2356}\}, \quad D_3^{11} = \{\overline{125}, \overline{346}\}.$$

Table 3.

i	R_i	$D_i = M(R_i)$	$D_i . R_i$	$b(D_i)$	$b(D_i . R_i)$	$c(D_i . R_i)$	$b(D_i . R_i / R_i)$
1	$\{\overline{123}, \overline{456}\}$	0	0	3	3	0	2
2	$\{\overline{124}, \overline{356}\}$	$\{\overline{12}, \overline{3}, \overline{45}, \overline{6}\}$	$\{\overline{12}, \overline{3}, \overline{4}, \overline{5}, \overline{6}\}$	2	3	1	2
3	$\{\overline{125}, \overline{346}\}$	$\{\overline{1}, \overline{25}, \overline{36}, \overline{4}\}$	$\{\overline{1}, \overline{25}, \overline{36}, \overline{4}\}$	2	2	1	1
4	$\{\overline{126}, \overline{345}\}$	0	0	3	3	0	2
5	$\{\overline{134}, \overline{256}\}$	$\{\overline{16}, \overline{2}, \overline{3}, \overline{4}, \overline{5}\}$	0	3	3	0	2
6	$\{\overline{135}, \overline{246}\}$	$\{\overline{13}, \overline{24}, \overline{5}, \overline{6}\}$	$\{\overline{13}, \overline{24}, \overline{5}, \overline{6}\}$	2	2	1	1
7	$\{\overline{136}, \overline{245}\}$	$\{\overline{13}, \overline{24}, \overline{5}, \overline{6}\}$	$\{\overline{13}, \overline{24}, \overline{5}, \overline{6}\}$	2	2	1	1
8	$\{\overline{145}, \overline{236}\}$	0	0	3	3	0	2
9	$\{\overline{146}, \overline{235}\}$	0	0	3	3	0	2
10	$\{\overline{156}, \overline{234}\}$	$\{\overline{1}, \overline{2}, \overline{3}, \overline{4}, \overline{56}\}$	$\{\overline{1}, \overline{2}, \overline{3}, \overline{4}, \overline{56}\}$	3	3	1	2
11	$\{\overline{1234}, \overline{56}\}$	$\{\overline{1234}, \overline{56}\}$	$\{\overline{1234}, \overline{56}\}$	1	1	2	0
12	$\{\overline{1235}, \overline{46}\}$	$\{\overline{1}, \overline{25}, \overline{3}, \overline{4}, \overline{6}\}$	$\{\overline{1}, \overline{25}, \overline{3}, \overline{4}, \overline{6}\}$	3	3	1	2
13	$\{\overline{1236}, \overline{45}\}$	$\{\overline{1}, \overline{25}, \overline{3}, \overline{4}, \overline{6}\}$	0	3	3	0	2
14	$\{\overline{1245}, \overline{36}\}$	$\{\overline{125}, \overline{3}, \overline{4}, \overline{6}\}$	$\{\overline{125}, \overline{3}, \overline{4}, \overline{6}\}$	2	2	2	1
15	$\{\overline{1246}, \overline{35}\}$	$\{\overline{12}, \overline{3}, \overline{45}, \overline{6}\}$	$\{\overline{12}, \overline{3}, \overline{4}, \overline{5}, \overline{6}\}$	2	3	1	2
16	$\{\overline{1256}, \overline{34}\}$	$\{\overline{1}, \overline{25}, \overline{3}, \overline{4}, \overline{6}\}$	$\{\overline{1}, \overline{25}, \overline{3}, \overline{4}, \overline{6}\}$	3	3	1	2
17	$\{\overline{1345}, \overline{26}\}$	$\{\overline{16}, \overline{2}, \overline{3}, \overline{4}, \overline{5}\}$	0	3	3	0	2
18	$\{\overline{1346}, \overline{25}\}$	$\{\overline{1}, \overline{2}, \overline{36}, \overline{4}, \overline{5}\}$	$\{\overline{1}, \overline{2}, \overline{36}, \overline{4}, \overline{5}\}$	3	3	1	2
19	$\{\overline{1356}, \overline{24}\}$	$\{\overline{13}, \overline{24}, \overline{5}, \overline{6}\}$	$\{\overline{13}, \overline{24}, \overline{5}, \overline{6}\}$	2	2	1	2
20	$\{\overline{1456}, \overline{23}\}$	0	0	3	3	0	2
21	$\{\overline{16}, \overline{2345}\}$	$\{\overline{1}, \overline{2}, \overline{35}, \overline{4}, \overline{6}\}$	$\{\overline{1}, \overline{2}, \overline{35}, \overline{4}, \overline{6}\}$	3	3	1	2
22	$\{\overline{15}, \overline{2346}\}$	$\{\overline{1}, \overline{2}, \overline{36}, \overline{4}, \overline{5}\}$	$\{\overline{1}, \overline{2}, \overline{36}, \overline{4}, \overline{5}\}$	3	3	1	2
23	$\{\overline{14}, \overline{2356}\}$	0	0	3	3	0	2
24	$\{\overline{13}, \overline{2456}\}$	$\{\overline{13}, \overline{24}, \overline{5}, \overline{6}\}$	$\{\overline{13}, \overline{24}, \overline{5}, \overline{6}\}$	2	2	1	2
25	$\{\overline{12}, \overline{3456}\}$	$\{\overline{1}, \overline{2}, \overline{36}, \overline{4}, \overline{5}\}$	$\{\overline{1}, \overline{2}, \overline{36}, \overline{4}, \overline{5}\}$	3	3	1	2

Then

$$M(R_3) = D_3 = D_3^{00} . D_3^{01} . D_3^{10} . D_3^{11} = \{\overline{1}, \overline{25}, \overline{36}, \overline{4}\}.$$

Further

$$D_3 . R_3 = D_3 = \{\overline{1}, \overline{25}, \overline{36}, \overline{4}\}.$$

Now we investigate whether the conditions of the theorem 3 are satisfied:

$$b(D_3) = b(D_3 \,.\, R_3) = [\log_2 4] = 2 \,,$$
$$c(D_3 \,.\, R_3) = [\log_2 2] = 1 \,,$$
$$b(D_3 \,.\, R_3) + c(D_3 \,.\, R_3) = 2 + 1 = 3 = s \,,$$
$$b(D_3 \,.\, R_3/R_3) = [\log_2 2] = 1 \,,$$
$$b(D_3 \,.\, R_3/R_3) + 1 = 1 + 1 = b(D_3 \,.\, R_3) = 2 \,.$$

We find that the partition pair $(\{\overline{1}, \overline{25}, \overline{36}, \overline{4}\}; \{\overline{125}, \overline{346}\})$ facilitates coding of the states of a machine $\mathscr{A}$ in such a way that the binary inputs of some momory element P_3 for a state variable Y_3 are merely the functions of inputs x_1, x_2 and of the only further state variable. Then the

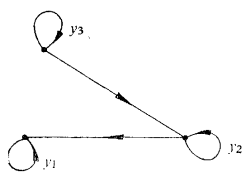

Fig. 2.

partition pair (D_3, R_3) is a candidate for a further choice of the assignment of s-state variables in which we intend to reach the maximum reduction of the dependences of inputs to all memory elements on outputs from the remaining elements. In an analoguous fashion computations have to be accomplished for all 25 partitions R_i $(i = 1, 2, \ldots, 25)$. The results of the corresponding computations are arranged in table 3. From this table it is easy to establish the i-indexes of partition pairs (D_i, R_i) that we have to eliminate because they do not satisfy some of the conditions of the theorem 3:

$i = 2, 15$ . .	do not satisfy the condition	(15) ,
$i = 2, 10, 12, 14, 15, 16, 18, 21, 22, 25$	do not satisfy the condition	(16) ,
$i = 19, 24$	do not satisfy the condition	(17) ,

From the table 3 by the assistance of the first four steps of the above-stated algorithm one can select only four nontrivial partition pairs having the indexes $i = 3, 6, 7$ and 11. It is evident that the application of the fifth algorithm step gives no contribution to the partition pairs already selected. (Steps 6 and 7 refering to forming the partition pairs leading to the reductions of the second type may be omitted, here.) From the mentioned four partition pairs it is not difficult to select a triad permitting the realization with the maximim reduction of input dependences of not one but all of these memory elements. This triad of partition pairs is identified by a triad of indexes (3, 7, 11). From the table 3 we can establish without difficulty that

$$D_3 \;= R_3 \,.\, R_7 \;= \{\overline{1}, \overline{25}, \overline{36}, \overline{4}\} \,,$$
$$D_7 \;= R_7 \,.\, R_{11} = \{\overline{13}, \overline{24}, \overline{5}, \overline{6}\} \,,$$
$$D_{11} = R_{11} \qquad = \{\overline{1234}, \overline{56}\} \,,$$

where $R_3 = \{\overline{125}, \overline{346}\}$ and $R_7 = \{\overline{136}, \overline{245}\}$.

If we assign state variables Y_1, Y_2, Y_3 in correspondence with partitions R_3, R_7, R_{11} respectively, the dependences of state variables Y_i $(i = 1, 2, 3)$ on state variables y_j $(j = 1, 2, 3)$ can be demonstrated synoptically in an auxiliary graph on Fig. 2. In fact every node of the graph represents a logical block that computes the value of a relevant state variable with delay 1 from the inputs of a sequential circuit x_1 and x_2 (which are not distinguished in the graph), and from the led — in state variables.

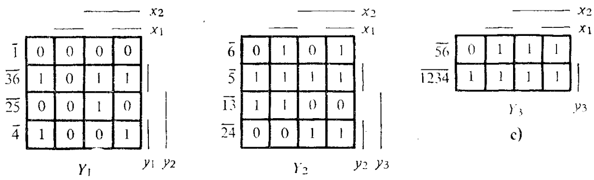

Fig. 3.

In the determining of a sequential circuit that realizes the transient table 2 of a machine $\mathscr{A}$, we proceed in the following way: For every partition block of R_3, R_7 and R_{11}let there be chosen values of corresponding binary state variables which we write above the relevant blocks:

$$R_3 = \{\overset{Y_1 = 0}{\overline{125}},\ \overset{Y_1 = 1}{\overline{346}}\}, \quad R_7 = \{\overset{Y_2 = 0}{\overline{136}},\ \overset{Y_2 = 1}{\overline{245}}\}, \quad R_{11} = \{\overset{Y_3 = 1}{\overline{1234}},\ \overset{Y_3 = 0}{\overline{56}}\}.$$

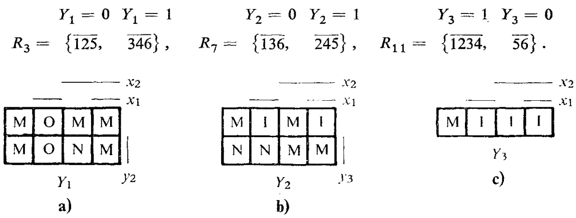

Fig. 4.

Then to every partition block D_3 and D_7 there correspond the following combinations of the values of state variables:

$$\begin{array}{rl} y_2 = & \quad 0 \qquad 1 \\ y_1 = & \overline{0\ 1}\ \overline{0\ 1} \\ D_3 = & \{\overline{1}, \overline{36}, \overline{25}, \overline{4}\}, \end{array} \qquad \begin{array}{rl} y_3 = & \quad 0 \qquad 1 \\ y_2 = & \overline{0\ 1}\ \overline{0\ 1} \\ D_7 = & \{\overline{6}, \overline{5}, \overline{13}, \overline{24}\}. \end{array}$$

From the transient table 2 by the aid of corresponding partition pairs we can obtain (without any difficulty) maps of Fig. 3a), b), c) for individual state variables Y_1, Y_2 and Y_3 respectively. The maps from Fig. 3 can be re-drawn in the corresponding maps in Fig. 4 where columns of values from Fig. 3 are substituted for their designation, introduced in [6].

On the basis of the considerations from the [6] we can already establish the suitable types of memory elements P_i for individual variables Y_i $(i = 1, 2, 3)$, for which inputs $a_{i1}, \ldots, a_{iu_i}$ will not be dependent on a variable y_i. A characteristic map of an element P_1 must contain columns M, O and N (for designations see [6]). Analogically the characteristic map of an element P_2 must contain M, I and N, and the characteristic map of an element P_3 has contain the columns M and I. Therefore, the chosen memory elements are *J-K*-flip-flop for P_1 and P_2, and *R-S*-flip-flop for P_3. The characteristic maps for both types of elements are shown in Fig. 5 a) and b). The excitation of memory elements can be obtained so that in maps in Fig. 4 individual letters I, M, N and O are substituted for the input values of corresponding memory elements which lead to the columns of the same designation. The resulting maps for the excitation of memory elements P_1,

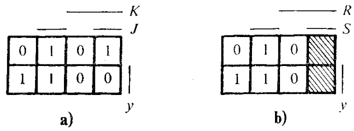

Fig. 5.

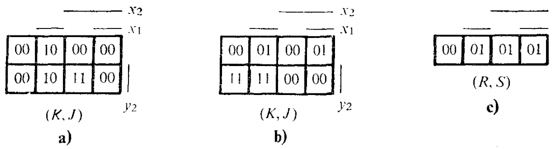

Fig. 6.

P_2 and P_3 resp. are on Fig. 6a), b) and c); from them the input equations of elements P_1, P_2 and P_3 have been already derived. The sequential circuit realizing the transient function of a primarily specified sequential machine $\mathscr{A}$ by the aid of 15-diodes, 3-invertors and 3-memory elements

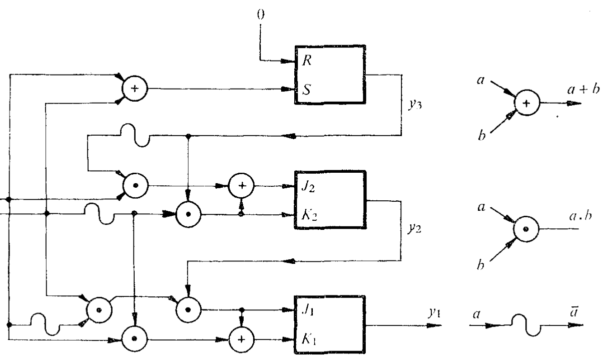

Fig. 7.

in diode logic, is shown in Fig. 7, where the meaning of particular designations are also established.

Example 2. Let the transient function of a machine $\mathscr{B}$ be established in the table 4 (see also [4] machine $\mathscr{B}$). We are to realize this function through a sequential circuit so that the dependences of inputs of applied memory elements on the outputs from other elements may be reduced in a maximum way.

The table 5 shows the reduced set of partiton pairs (D_i, R_i) for which $b(D_i) \leqq s - 1 = 2$. When employing the first five steps of algorithm for our example we obtain only one partition pair with an index $i = 4$. Through the application of the 6th step of algorithm we obtain partition pairs with indexes $i = 2, 3, 5, 6$. On the basis of a step 7 we obtain a partition pair (D_1^1, R_1) where the partition $D_1^1 = \{\overline{0}, \overline{2}, \overline{15}, \overline{34}\}$ has arisen by the refinement of a partition D_1. The partitions D_i of partition pairs with indexes $i = 7, 8, 9, 10, 11$ are not possible to be refined so that we may satisfy the required conditions.

Table 4.

$\mathscr{B}$ \ x_1x_2	00	01	11	10
0	3	4	4	1
1	0	3	2	4
2	5	2	4	1
3	2	4	2	0
4	1	2	4	4
5	0	5	2	2

From partition pairs with $i = 1^1, 2, 3, 4, 5, 6$ we can already select such three pairs that will lead to realization with dependences which are reduced in a maximum way. A suitable triad of partition pairs is given by a triad of indexes (2, 3, 4). If we assign state variables y_1, y_2, y_3 conformably to partitions 2, 3, 4 respectively, the corresponding graph of dependeces of state variables will be on Fig. 8. Now we are proceeding analogically with example 1.

$$\begin{array}{lll}
Y_1 = \quad 0 \quad 1 & Y_2 = \quad 0 \quad 1 & Y_3 = \quad 0 \quad 1 \;, \\
R_2 = \{\overline{023}, \overline{145}\}\,, & R_2 = \{\overline{2345}, \overline{01}\}\,, & R_4 = \{\overline{024}, \overline{135}\}\,,
\end{array}$$

$$\begin{array}{ll}
y_2 = \quad 0 \qquad 1 & y_1 = \quad 0 \qquad 1 \\
y_3 = \overline{0 \quad 1}\ \overline{0 \quad 1} & y_3 = \overline{0 \quad 1}\ \overline{0 \quad 1} \\
D_1 = \{\overline{24}, \overline{35}, \overline{0}, \overline{1}\}\,, & D_4 = \{\overline{02}, \overline{3}, \overline{4}, \overline{15}\}\,.
\end{array}$$

Table 5.

i	R_i	$D_i = M(R_i)$	$D_i . R_i$	$b(D_i)$	$b(D_i . R_i)$	$c(D_i)$	$c(D_i . R_i)$	$b(D_i . R_i/R_i)$
1	$\{\overline{0125}, \overline{35}\}$	$\{\overline{02}, \overline{15}, \overline{34}\}$	$\{\overline{02}, \overline{1}, \overline{3}, \overline{5}, \overline{4}\}$	2	3	1	1	2
2	$\{\overline{023}, \overline{145}\}$	$\{\overline{0}, \overline{1}, \overline{24}, \overline{35}\}$	0	2	3	1	0	2
3	$\{\overline{01}, \overline{2345}\}$	$\{\overline{023}, \overline{145}\}$	$\{\overline{0}, \overline{1}, \overline{23}, \overline{45}\}$	1	2	2	1	1
4	$\{\overline{024}, \overline{135}\}$	$\{\overline{02}, \overline{15}, \overline{3}, \overline{4}\}$	$\{\overline{02}, \overline{15}, \overline{3}, \overline{4}\}$	2	2	1	1	1
5	$\{\overline{035}, \overline{124}\}$	$\{\overline{02}, \overline{15}, \overline{3}, \overline{4}\}$	0	2	3	1	0	2
6	$\{\overline{015}, \overline{234}\}$	$\{\overline{03}, \overline{14}, \overline{2}, \overline{5}\}$	0	2	3	1	0	2
7	$\{\overline{05}, \overline{1234}\}$	$\{\overline{04}, \overline{12}, \overline{3}, \overline{5}\}$	$\{\overline{0}, \overline{12}, \overline{3}, \overline{4}, \overline{5}\}$	2	3	1	1	2
8	$\{\overline{03}, \overline{1245}\}$	$\{\overline{05}, \overline{24}, \overline{1}, \overline{3}\}$	$\{\overline{0}, \overline{1}, \overline{24}, \overline{3}, \overline{5}\}$	2	3	1	1	2
9	$\{\overline{013}, \overline{245}\}$	$\{\overline{0}, \overline{1}, \overline{23}, \overline{45}\}$	$\{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{45}\}$	2	3	1	1	2
10	$\{\overline{0135}, \overline{24}\}$	$\{\overline{02}, \overline{15}, \overline{3}, \overline{4}\}$	$\{\overline{0}, \overline{15}, \overline{2}, \overline{3}, \overline{4}\}$	2	3	1	1	2
11	$\{\overline{02}, \overline{1345}\}$	$\{\overline{0}, \overline{1}, \overline{24}, \overline{35}\}$	$\{\overline{0}, \overline{1}, \overline{2}, \overline{35}, \overline{4}\}$	2	3	1	1	2

Maps for particular state variables Y_1, Y_2, Y_3 which are represented on Fig. 9 together with their algebraic expressions can be obtained with the same process as for a machine $\mathscr{A}$. From Fig. 8 it evident that it is advantageous to employ a more general memory element for a variable y_3, while for variables y_1 and y_2 it is favourable to employ a memory element of D type. On Fig. 10 there are the equations for the excitation of inputs J-K flip-flop if we employ it for a vari-

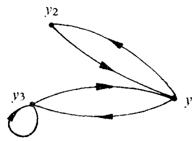

Fig. 8.

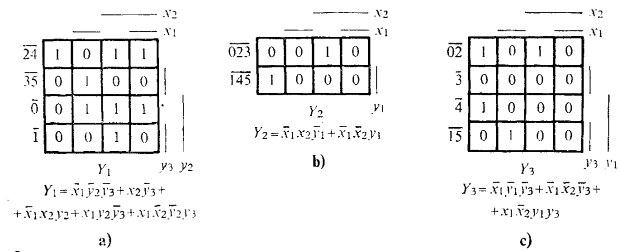

Fig. 9.

able y_3. In this case there is the saving of four diodes with respect to a realization by means of an element of D type on Fig. 9c).

Another possible realization of the transient function of a machine $\mathscr{B}$ can be achieved by the aid of a triad of partition pairs with indexes (2, 4, 5). The corresponding diagram with the dependences of variables connected with relevant partitions is on Fig. 11. It has been found that this realization puts out of service one diode more than the realization mentioned before.

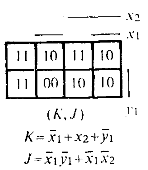

Fig. 10.

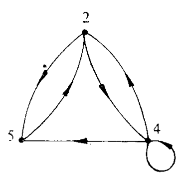

Fig. 11.

When selecting partition pairs according to the mentioned algorithm – in a general case – the selected set of partition pairs need not contain such a subset of s-partition pairs which will lead to an assignment with the maximum total reduction of dependeces in the realization of a sequential machine. Then it can be advantageous to refine some or also all partition pairs in such a way that instead of very restrictive conditions (22) and (23) looser conditions may hold:

$$(22') \qquad b(D_i^{v_1} . R_i) \leqq s - 1$$

or

$$(23') \qquad b(D_i^{v_1}) \leqq s - 1 .$$

The final selection of s-partition pairs will be carried out from the enlarged set of partition pairs satisfying the algorithm with thus changed conditions. If we refine all partition pairs (also those which have already satisfied some conditions of the theorem 3 or of the theorem 4) and if we do the final selection of s-partition pairs systematically from this enlarged set, we may obtain an assignment with a maximum total reduction of the dependences of inputs to memory elements. But the set of partition pairs from which we choose, in this case, is growing rapidly, so that selection is becoming more difficult.

For some machines need not exists such a coding of internal states by $s = [\log_2 n]$ binary state variables, which would enable reduction of dependences of inputs on state variables, with all memory elements (with exception of a reduction noted in the theorem 4.1 from [6]). Then it is necessary for coding admit also a selection of such subsets s-partition pairs, some of which are also trivial (i.e. $D_i = 0$). That means that the set of P. P. from which we choose s-elements requires a further extension.

Let us note on conclusion that the mentioned algorithm for determination of (D, R)-partition pairs is programmed for digital computer Epos.

(Received October 21st, 1966.)

REFERENCES

[1] Stearns R. E., Hartmanis J.: On the State Assignment Problem for Sequential Machines II. IRE Trans. *EC-10* (1961), 4, 593—603.

[2] Hartmanis J.: On the State Assignment Problem for Sequential Machines I. IRE Trans. *EC-10* (1961), 2, 157—165.

[3] Hartmanis J.: Symbolic Analysis of a Decomposition of Information Processing Machines. Inf. and Contr. *3* (1960), 154—178.

[4] Curtis H. A.: Multiple Reduction of Variable Dependency of Sequential Machines. J. ACM *9* (1962), 324—344.

[5] Karp R. M.: Some Techniques of State Assignment for Synchronous Sequential Machines. IEEE Trans. *EC-13* (1964), 5, 507—518.

[6] Novanský R.: Syntéza sekvenčních obvodů s využitím obecných paměťových elementů. Kybernetika *3* (1967), 3, 234—252.

[7] Васильева Н. П., Гашковец И. С.: Простейшие последовательностные элементы (Триггеры) и реализация запоминающей части последовательностных функций на этих элементах. Автоматика и телемеханика *25* (1964), 6, 1004—1014.

Paměťové elementy a rozkladové páry při syntéze sekvenčních obvodů

Rudolf Novanský

Článek uvádí některé nové výsledky pro kódování vnitřních stavů sekvenčních strojů, při jejichž realizaci jsou v paměťové části odpovídajícího sekvenčního obvodu využívány paměťové elementy obecných typů. Při vyšetřování se využívá konceptu rozkladových párů původně zavedeného Stearnsem a Hartmanisem [1]. Na základě těchto výsledků je sestaven algoritmus (programovatelný pro samočinný počítač), který dovoluje pro daný sekvenční stroj určit vhodné kódování vnitřních stavů. Postup dále dovoluje určit, jaké paměťové elementy vedou k (v uvedeném smyslu) nejekonomičtější realizaci vnitřní funkce daného stroje. Článek je pokračováním práce [6].

Ing. Rudolf Novanský, Výzkumný ústav matematických strojů, Loretánské nám. 3, Praha 1.