INFORMATION AND CONTROL 12, 538–548 (1968)

Decomposition of a Synchronous Machine into an Asynchronous Submachine driving a Synchronous One

G. B. GERACE AND G. GESTRI

Centro Studi Calcolatrici Elettroniche del C.N.R. presso l'Università di Pisa, Pisa, Italy

To complete the study of the serial and parallel decompositions of a synchronous sequential machine into an asynchronous submachine and a synchronous one, discussed in *Information and Control*, 1967, Vol. 11, pp. 568–591, we consider the serial decomposition where the asynchronous submachine drives the synchronous one.

I. INTRODUCTION

In a previous paper (Gerace and Gestri, 1967) we studied the serial and parallel decompositions of a synchronous sequential machine into a synchronous submachine and an asynchronous one. In the serial decomposition the synchronous submachine drove the asynchronous one. In this paper, to complete the study of the serial decomposition, we consider the case in which the asynchronous submachine drives the synchronous one.

Three types of sequential machines will be discussed, namely, machines of the Moore type, machines of the Mealy type, and anticipated machines defined in Gerace and Gestri (1967). Only machines without "don't care" conditions will be considered, and the possibility of state splitting (Hartmanis and Stearns, 1962) will not be taken into account. Arguments developed in Hartmanis (1961, 1962) and in Gerace and Gestri (1967) are assumed to be known.

II. PHYSICAL REALIZATION

Let I and S_h be an input state and an internal state, respectively, of a sequential machine M. Denote by $N(I, S_h)$ the next-state of M determined by I and S_h. A fundamental state machine is a state machine such that, for any I and S_h, if $N(I, S_h) = S_k$, then $N(I, S_k) = S_k$.

Let us assume that a synchronous machine M is decomposed into two state machines serially connected, M_F and M_S, and into a combinational network C_0 (Hartmanis, 1962). Let us assume further that the first machine M_F is a fundamental state machine. We shall denote by $M_F \rightarrow M_S$ this decomposition.

The physical realization of M corresponding to this decomposition is shown in Fig. 1, where both M_F and M_S are realized by synchronous devices. Assuming that a finite time d is necessary to the combinational parts of M_F and M_S for the computation of the corresponding next states, the delay element Δ_1 which stores the internal state must be such that $\Delta_1 + d = \Delta$, where Δ is the time interval of the input sequence of M.

To determine the conditions for M_F to be realizable by an asynchronous device, let us see what happens when the storage element Δ_1 is removed from the device realizing M_F (Fig. 2). Since M_F is a fundamental machine, its output sequence, identical to the present state sequence, remains unchanged. However, denoted by t_n any time when the input state changes, the present state (outputs) of M_F changes at time $t_n + d$,

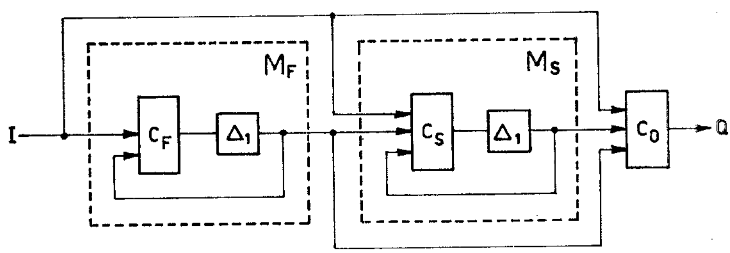

FIG. 1. Serial decomposition of a synchronous machine M into two synchronous submachines, M_F and M_S.

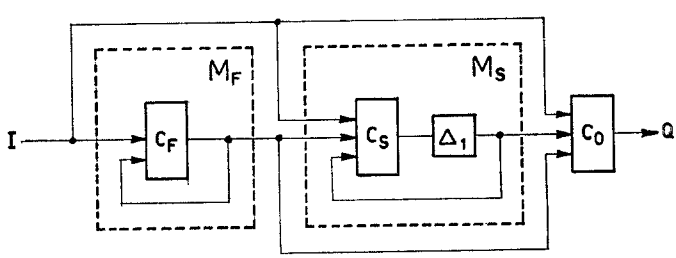

FIG. 2. Serial decomposition of M into an asynchronous submachine driving a synchronous one.

whereas it changes at time $t_n + \Delta = t_{n+1}$ when Δ_1 is present in the device realizing M_F. Therefore, to realize M_F by an asynchronous device without changing the external behavior of M, we must ensure that the state behavior of M_S and the output of C_0 are independent of any change of the present state (outputs) of M_F at time $t_n + d$.

For any state **a** of M, let A and K be the corresponding states of M_F and M_S, respectively; for any input I of M, let B be the next-state of M_F for present state A and input I, $N_S(I,A,K)$ and $N_S(I,B,K)$ be the next states of M_S for present state K and input states I,A and I,B, respectively, and $C_0(I,A,K)$ and $C_0(I,B,K)$ be the output states of the combinational network C_0 for inputs I,A,K and I,B,K, respectively. The external behavior of M remains unchanged after the removal of Δ_1 from the device realizing M_F, if for any I and **a** of M we have:

(i) $N_S(I,A,K) = N_S(I,B,K)$

(ii) $C_0(I,A,K) = C_0(I,B,K)$

In fact, for any change at time $t_n + d$ of the state of M_F, if condition (i) holds, the state behavior of M_S remains unchanged, and if conditions (i) and (ii) are satisfied, the output of C_0 does not change.

In the following section we shall put in terms of partitions the conditions for the existence of an $M_F \rightarrow M_S$ decomposition satisfying conditions (i) and (ii).

III. DECOMPOSITION USING A FUNDAMENTAL SUBSTITUTION PROPERTY PARTITION

Let π be a partition with the substitution property (S.P.) of a sequential machine M.

DEFINITION. We shall say that π is fundamental if and only if for any state **a** and any input I of M, denoted by B the block of π containing $N(I, \mathbf{a})$, we have $N(I, \mathbf{b}) \in B$ for any state **b** of M contained in B.

Trivially a $M_F \rightarrow M_S$ decomposition exists for a sequential machine M if and only if a fundamental S.P. partition π exists for M. The following theorem gives in terms of partitions the conditions of existence of an $M_F \rightarrow M_S$ decomposition such that M_F can be realized by an asynchronous device.

THEOREM 1. *For a sequential machine M a $M_F \rightarrow M_S$ decomposition such that M_F can be realized by an asynchronous device exists if a pair of partitions π,π^* exists which satisfy the following conditions:*

(a) $\pi \cdot \pi^* = 0$

(b) *π is a fundamental S.P. partition*

(c) *any two states of M in a common block of π^* which have the next state in the same block of π for some input I, have for this I the same next state.*

(d) *any two states of M in a common block of π^* which have the next-state in the same block of π for some input I, have for this I the same output state.*

Proof. Conditions (a) and (b) are the necessary and sufficient conditions for the existence of a serial decomposition $M_F \to M_S$. We have only to show that this decomposition satisfies condition (i) of the previous section if and only if condition (c) is satisfied, and satisfies condition (ii) if and only if condition (d) is met.

Assume first that the decomposition $M_F \to M_S$ corresponding to the partitions π and π^* satisfies condition (i). Let $\mathbf{a}$ and $\mathbf{b}$ be any two states in a common block of π^*, say, K. Denote by A and B the blocks of π containing $\mathbf{a}$ and $\mathbf{b}$, respectively, and let us assume that $N(I, \mathbf{a})$ and $N(I, \mathbf{b})$ belong to the same block of π, say, C. When M is in the state $\mathbf{a}$ and input I is applied, M_F is in the state A and makes a transition to the state C. For condition (i) we have $N_S(I,A,K) = N_S(I,C,K)$. When M is in state $\mathbf{b}$ and input I is applied, M_F is in state B and makes a transition to state C. For condition (i) we have $N_S(I,B,K) = N_S(I,C,K)$. Therefore, $N_S(I,A,K) = N_S(I,B,K)$; since the next state of M_S is in the block of π^* containing the next state of M, this means that $N(I, \mathbf{a})$ and $N(I, \mathbf{b})$ are in the same block of π^*. Then $N(I, \mathbf{a}) = N(I, \mathbf{b})$, since $N(I, \mathbf{a})$ and $N(I, \mathbf{b})$ are in a common block of π and in a common block of π^*, and $\pi \cdot \pi^* = 0$. We have shown that if $M_F \to M_S$ satisfies condition (i), π and π^* satisfy condition (c).

Conversely, let $\mathbf{a}$ be an internal state and I an input state of M, and A and K be the blocks of π and π^*, respectively, containing $\mathbf{a}$, and B be the block of π containing $N(I, \mathbf{a})$. If $B \cap K \neq \emptyset$, let $\mathbf{b}$ be the state of M defined by $B \cap K$. Trivially, $\mathbf{a}$ and $\mathbf{b}$ belong to the same block of π^*. Furthermore, since π is fundamental, $N(I, \mathbf{b})$ is contained in the block B of π; then $N(I, \mathbf{a})$ and $N(I, \mathbf{b})$ belong to the same block of π, and we have $N(I, \mathbf{a}) = N(I, \mathbf{b})$, because of condition (c). Therefore $N_S(I,A,K) = N_S(I,B,K)$. If $B \cap K = \emptyset$, we have a "don't care" condition in the state table of M_S for $N_S(I,B,K)$; to show that this condition can always be filled so that $N_S(I,B,K) = N_S(I,A,K)$ we have only to show that no other state of M which requires a different filling can exist. In fact, assume that a state $\mathbf{b}$ of M belonging to the block K of π^* and such that $N(I, \mathbf{b})$ belongs to B exists; let us denote by C the block of π containing

b. Trivially, $N_S(I,C,K) = N_S(I,A,K)$, because of condition (c). Therefore, if condition (c) holds, condition (i) is satisfied, or it can be satisfied with careful filling of the "don't care" conditions in M_S.

The proof that condition (d) is satisfied if and only if condition (ii) holds is easily obtained by the substitution of the output state of M to the next state of M in the proof of condition (c) given above.

Note that for an anticipated machine, a pair of partitions which satisfy condition (c) also satisfy condition (d), as it is easily seen from the definition of this type of machine (Gerace and Gestri, 1967). Therefore for these machines it is not necessary to take under consideration condition (d). Furthermore, in the same paper it was shown that if we don't care to anticipate one time unit the output sequence, any Moore machine can be transformed into an anticipated machine with the same state table. Then, given a Moore machine having a pair π,π^* which does not satisfy condition (d), but which satisfies the other conditions of Theorem 1, if we don't care to anticipate one unit the output sequence, it is possible to transform the Moore machine into an anticipated machine for which the pair π,π^* satisfies Theorem 1. Therefore, this machine can be realized by an asynchronous device driving a synchronous device.

IV. EXAMPLES

Let us consider the Mealy machine P described in Fig. 3. The S.P. partition

$$\pi = \{\overline{1,2,3};\ \overline{4,5};\ \overline{6,7,8}\} \equiv \{A,B,C\}$$

			I_1	I_2	I_3	I_1	I_2	I_3
A	α	1	8	4	3	000	101	101
	β	2	7	5	2	010	110	111
	γ	3	6	4	1	100	111	000
B	β	4	7	5	5	010	110	011
	α	5	8	4	4	000	101	001
C	β	6	7	8	2	010	011	111
	α	7	8	6	3	000	010	101
	γ	8	6	7	1	100	001	000

FIG. 3. Machine P

is fundamental, as we can verify from the state table of machine P. The partition

$$\pi^* = \{\overline{1,5,7}; \overline{2,4,6}; \overline{3,8}\} \equiv \{\alpha,\beta,\gamma\}$$

satisfies, with π, Theorem 1. In fact, let us verify that the pair π,π^* satisfies condition (c). For the block α of π^* and input I_1, we see that states "1," "5," "7" have the next states in the same block, C, of π; condition (c) is satisfied since these states have the same next state. For the block α and input I_2, states "1" and "5" have the next states in the same block, B, of π; condition (c) is satisfied since these states have the same next state. For the block α and input I_3, states "1" and "7" have the next states in the same block, A, of π; condition (c) is satisfied since these states have the same next state. In the same way condition (c) can be verified for the states in the blocks β and γ of π^*.

The test for condition (d) of Theorem 1 follows the same line. For example, states "1," "5," and "7," which have for input I_1 the next-states in the same block, C, of π, must have the same output state for input I_1. Since π and π^* satisfy Theorem 1, we know that in the corresponding $M_F \rightarrow M_S$ decomposition of P, M_F can be realized by an asynchronous device (Fig. 2). The state tables of machines M_F and M_S are obtained as explained by Hartmanis (1962) and are shown in Fig. 4a and 4b, respectively. Since $\gamma \cap B = \emptyset$, we have don't care conditions in the state table of M_S, circled in Fig. 4b; as it was seen in the proof of Theorem 1, these don't-care conditions must be filled carefully. This is easily obtained. In fact, from the state table of M_F, we see that for input I_2 we have a transition from state A to state B; therefore, in the state table of M_S the unspecified cell in column (I_2, B) must be filled as the cell in the same row and in column (I_2, A), as shown in Fig. 4b. For inputs I_1 and I_3 we have no transition to the state B, so the don't-care conditions for state γ and inputs (I_1, B) and (I_3, B) need not be filled. In the same way we obtain the filling of the don't-care conditions in the output functions of C_0, shown in Fig. 4c. In the example given above, the pair π,π^* was given. We shall now give an example showing how, given a fundamental S.P. partition, the π^* with the least number of blocks can be obtained.

From the state table of machine R, shown in Fig. 5, we see that the S.P. partition

$$\pi = \{\overline{1,2}; \overline{3,4}; \overline{5,6}\} \equiv \{A,B,C\}$$

is fundamental. To obtain the partition π^*, let us denote by the symbols

	I_1	I_2	I_3
A	C	B	A
B	C	B	B
C	C	C	A

a)

	I_1A	I_1B	I_1C	I_2A	I_2B	I_2C	I_3A	I_3B	I_3C
α	γ	γ	γ	β	β	β	γ	β	γ
β	α	α	α	α	α	γ	β	α	β
γ	β	(−)	β	β	(β)	α	α	(−)	α

b)

	I_1A	I_1B	I_1C	I_2A	I_2B	I_2C	I_3A	I_3B	I_3C
α	000	000	000	101	101	010	101	001	101
β	010	010	010	110	110	011	111	011	111
γ	100	(−)	100	111	(111)	001	000	(−)	000

c)

FIG. 4. Serial decomposition of machine P. (a) State table of M_P; (b) state table of M_S; (c) output table of C_0.

		I_1	I_2	I_3	I_1	I_2	I_3
A	1	2	3	1	00	11	10
	2	1	4	2	01	00	01
B	3	4	4	5	10	00	11
	4	3	3	6	11	11	00
C	5	1	6	5	01	10	11
	6	1	5	5	10	01	10

FIG. 5. Machine R

α, β, etc. the blocks of π^*. The states of R in the first block, A, of π are coded in an arbitrary manner, for example,

$$\text{"1"} \to \alpha, \qquad \text{"2"} \to \beta.$$

Then the second block of π is considered. Since for input I_2 the blocks A and B are mapped in the same block, B, of π, we can give to one state of B the same symbol used for one state of the block A only if this pair of states has for input I_2 the same next state and the same output state, because of Theorem 1. From the state and output tables of R we see that it is possible to code both the states in B without using extra symbols, in the following way

$$\text{“3”} \rightarrow \beta, \qquad \text{“4”} \rightarrow \alpha.$$

For the states in the block C, we see that input I_1 maps A and C, and input I_3 maps B and C, in the same block of π. From the state and output tables of R we see that symbol β can be used for state “5,” which has for input I_1 the same next and output states as state “2,” and has for input I_3 the same next and output state as state “3,” whereas for state “6” an extra symbol is necessary. Then we give the code

$$\text{“5”} \rightarrow \beta, \qquad \text{“6”} \rightarrow \gamma,$$

and we obtain

$$\pi^* = \{\overline{1,4}; \overline{2,3,5}; 6\} \equiv \{\alpha,\beta,\gamma\}.$$

Note that in this example, the π^* with the least number of blocks has more than $n(\pi)$ blocks, where $n(\pi)$ denotes the number of states in the largest block of π.

Figure 6 shows the state tables of M_F and M_S, and the output table of C_0 which result from the pair π,π^*. The don't-care conditions are

	I_1	I_2	I_3
A	A	B	A
B	B	B	C
C	A	C	C

a)

	I_1A	I_1B	I_1C	I_2A	I_2B	I_2C	I_3A	I_3B	I_3C
α	β	β	(—)	β	β	(—)	α	γ	(γ)
β	α	α	α	α	α	γ	β	β	β
γ	(α)	(—)	α	(—)	(—)	β	—	(—)	β

b)

	I_1A	I_1B	I_1C	I_2A	I_2B	I_2C	I_3A	I_3B	I_3C
α	00	11	(—)	11	11	(—)	10	00	00
β	01	10	01	00	00	10	01	11	11
γ	(10)	(—)	10	(—)	(—)	01	(—)	(—)	10

c)

FIG. 6. Serial decomposition of machine R. (a) State table of M_F; (b) state table of M_S; (c) output table of C_0.

circled in the figure and have been filled, according to the proof of Theorem 1, in such a way that conditions (i) and (ii) are satisfied.

Finally, let us consider two Moore machines. For the machine T_1 in Fig. 7, the S.P. partition

$$\pi = \{\overline{1,2}; \overline{3,4}\} \equiv \{A,B\}$$

is fundamental. The π^* with the least number of blocks is easily obtained, and is given by

$$\pi^* = \{\overline{1,3}; \overline{2,4}\} \equiv \{\alpha,\beta\}.$$

The corresponding decomposition is shown in Fig. 8; we see that the output of C_0 depends only on the state of M_S. It is not difficult to recognize that this is true for any strongly connected Moore machine having an $M_F \rightarrow M_S$ decomposition such that M_F can be realized by an asynchronous device.

	I_1	I_2	I_3	Q
1	2	4	1	0
2	1	3	2	1
3	3	4	1	0
4	4	3	2	1

FIG. 7. Machine T_1

	I_1	I_2	I_3
A	A	B	A
B	B	B	A

M_F

	I_1A	I_1B	I_1C	I_2A	I_2B	I_2C
α	β	α	β	β	α	α
β	α	β	α	α	β	β

M_S

α	0
β	1

C_0

FIG. 8. Serial decomposition of machine T_1

s(t)	Q(t)
1	00
2	01
3	10
4	11

FIG. 9. Output table of machine T_2

		Q(t)
S(t+1)	1	00
	2	01
	3	10
	4	11

a)

	α	β
A	00	01
B	10	11

b)

FIG. 10. (a) Output table of the anticipated machine U; (b) Output table of C_0.

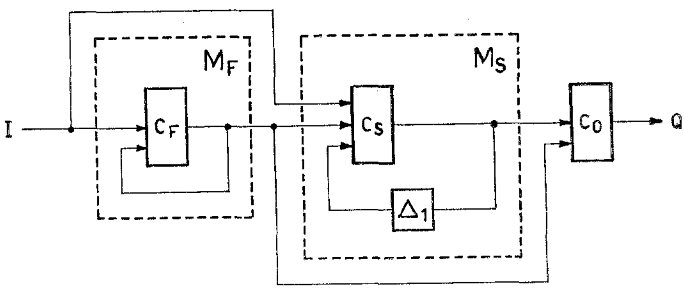

FIG. 11. Realization of machine U

For the Moore machine T_2, defined by the state table of machine T_1 in Fig. 7 and by the output table in Fig. 9, the pair π,π^* given above does not satisfy condition (d) of Theorem 1. However, if we don't care to anticipate one unit the output sequence, T_2 can be transformed into the anticipated machine U, which has the same state table as T_2 and the output table shown in Fig. 10a; the pair π,π^* satisfies Theorem 1 for machine U, and therefore in the corresponding $M_F \rightarrow M_S$ decomposition of U, M_F can be realized by an asynchronous device. The state tables of M_F and M_S have been already obtained; the output functions of C_0, which depend on the next states of M_F and M_S, are shown in Fig. 10b. The corresponding realization of machine U is shown in Fig. 11.

CONCLUSIONS

In this paper the serial decomposition of a synchronous sequential machine into an asynchronous machine driving a synchronous one has been studied. We have assumed that the asynchronous machine is described by a fundamental state table. Aside from the practical interest in reducing the number of delay elements in sequential machines, we feel that this study may contribute to clarify the conditions under which sequential networks of different types (synchronous and asynchronous) can be connected in a system of sequential networks.

RECEIVED: March 26, 1968

REFERENCES

GERACE, G. B., AND GESTRI, G. (1967). Decomposition of synchronous sequential machines into synchronous and asynchronous submachines. *Information and Control* **11**, 568–591.

HARTMANIS, J. (1961). On the state assignment problem for sequential machines, I. *IRE Trans. Electronic Computers* **EC-10**, 157–165.

HARTMANIS, J. (1962). Loop-free structure of sequential machines. *Information and Control* **5**, 25–43.

HARTMANIS, J., AND STEARNS, R. E. (1962). Some dangers in state reduction of sequential machines. *Information and Control* **5**, 252–260.