On Sequential Machine Decompositions for Reducing the Number of Delay Elements

P. R. Menon

Bell Telephone Laboratories, Incorporated, Murray Hill, New Jersey

In a recent paper, Gerace and Gestri (1967b) showed that the number of delay elements required in the realization of a synchronous sequential machine may be reduced by decomposing it into a state synchronous machine $M_s$, a state asynchronous machine $M_f$, and a combinational output circuit. This paper discusses some problems associated with the realization of $M_f$ not considered in the above-mentioned paper, and presents an alternative procedure for obtaining the decomposition. It is shown that any $m$-block partition with S.P. can be used to obtain such a decomposition in which the synchronous machine $M_s$ has at most $2m$ states. This leads to the result that any machine can be decomposed into a two-state autonomous machine realizable with a trigger flip-flop and a single delay, and an asynchronous machine with at most twice as many states as the original machine.

INTRODUCTION

In a recent paper, Gerace and Gestri (1967b) considered the problem of decomposing a synchronous sequential machine $M$ into a state synchronous machine $M_s$, a state asynchronous machine $M_f$, and a combinational output circuit $C_0$. The purpose of the decomposition was to minimize the number of delay elements required in the realization, since $M_f$ could be realized without any delay elements.

In this paper, we first discuss some problems associated with the delay-free realization of $M_f$, not considered by Gerace and Gestri. An alternative procedure for decomposing a synchronous machine into synchronous and asynchronous machines is presented. We also show that any $m$-block partition with substitution property\(^1\) (S.P.) (Hartmanis, 1961) can

\(^1\) A partition $\pi$ on a set of states $S$ is a grouping of all the members of $S$ into disjoint subsets called blocks. A partition $\pi$ has the substitution property (S.P.) if for any two states belonging to the same block of $\pi$ and every input $I_k$, the states to which the machine goes when the input $I_k$ is applied are contained in a common block.
be used to obtain a decomposition such that the state synchronous machine $M_s$ can be realized with at most $\lceil \log_2 m \rceil + 1$ delays, where $\lceil x \rceil$ is the smallest integer greater than or equal to $x$. Since every machine has a trivial one-block partition $I$ with S.P., it follows that a single delay is sufficient for realizing any synchronous machine.

We shall assume that the machine $M$ to be realized is specified by a flow table, and that the flow table is reduced. In this paper, we restrict our attention to the state behavior of machines, since the output circuit can be realized in the conventional manner. For any state $S_i$ and input $I_k$, the flow table specifies the next state of the machine, which we shall denote by $N(I_k, S_i)$. A normal mode flow table (also referred to as fundamental state table) is a flow table which has the following property: If for any $S_i$ and input $I_k$, $N(I_k, S_i) = S_j$, then $N(I_k, S_j) = S_j$. The $S_j$ is referred to as a stable state and is usually circled in the flow table.

**REALIZATION OF $M_F$**

The inputs to the asynchronous machine $M_F$ consist of the external inputs and the state variables of the synchronous machine $M_s$. As a result, more than one input to $M_F$ may change during any transition and $M_F$ should be designed to operate correctly independent of the order of these changes. In addition, $M_F$ can be realized without delay elements if and only if it contains no essential hazards (Unger, 1959). However, if the clock signal is applied to $M_F$ also, as shown in Fig. 1, a delay-free realization of $M_F$ allowing multiple-input changes is possible.

In Fig. 1, the external inputs are allowed to change only when the clock signal is 0. Let $W$ be the width of the clock pulse. For the combinational circuit $C_s$, let $d_s$ be the minimum duration of input signals required for correct response of the circuit and let $d_s'$ be the minimum time for the effect of an input change to propagate to any of its outputs. Let $d_F$ and $d_F'$ be defined similarly for the circuit, $C_F$. Clearly, we require $W \geq d_s$ and $W \geq d_F$. In order to prevent both machines from undergoing more than one state transition for any input transition, we also require $W < D + d_s'$, where $D$ is the magnitude of inserted delays. The time between successive clock pulses should be sufficient to allow both circuits to become stable.

The machine $M_F$ should be realized so that it remains stable whenever

---

2 A flow table contains an essential hazard if there exist a stable total state $(S_0, I_1)$ and a second input $I_2$ which can follow $I_1$ such that, starting with the machine in $(S_0, I_1)$ the input sequence $I_2I_1I_2$ leads to a state different from that led to by $I_2$. 
the clock signal is 0. This can be accomplished by realizing an augmented flow table \( M_p' \) obtained from \( M_p \) by treating the clock as an additional input variable. The columns of \( M_p' \) representing input states with clock signal equal to 0 will contain only stable states, and the other columns of \( M_p' \) are identical to the corresponding columns of \( M_p \). Note that \( M_p' \) will be a PF table (Gerace and Gestri, 1967a). This table contains no essential hazards (Gerace, 1966), because all transitions are to or from columns which contain only stable states. The \( M_p' \) can be realized without inserted delay elements, provided a suitable state assignment is used.

The state assignment for \( M_p \) should be free of critical races. In the absence of delays, races cannot be resolved by fixing the order of state variable changes. It is therefore necessary to use state assignments in which only one variable changes in a transition or all variables which change during a transition do so without critical races. Both these types of assignments will be referred to as single transition time (STT) assign-
ments. Liu (1963) and Tracey (1966) have given procedures for obtaining such assignments for normal mode flow tables. These assignments often require more than \([\log_2 n]\) variables for realizing an \(n\)-state table. The state assignment technique for minimizing the number of delay elements presented in an earlier paper (Gerace and Gestri, 1967a) may be viewed as a decomposition of the given machine into a synchronous submachine and a number of two-state normal mode asynchronous submachines, each of which required only a single state variable for its realization.

**DECOMPOSITION PROCEDURE**

Before presenting an alternative procedure for decomposition, we restate a theorem by Gerace and Gestri (1967b).

**THEOREM**

i. A sequential machine \(M\) can be decomposed into two serially connected machines \(M_s\) and \(M_f\), where \(M_f\) is a normal mode asynchronous machine if and only if there exist two nontrivial partitions \(\pi\) and \(\pi^*\) on the states of \(M\) satisfying the following conditions:

(a) \(\pi \cdot \pi^* = 0\)
(b) \(\pi\) has S.P.
(c) For any state \(S_i\) and any input \(I\) of \(M\), let \(A\) and \(K\) be the blocks of \(\pi^*\) and \(\pi\), respectively containing \(S_i\), and let \(B\) be the block of \(\pi^*\) containing \(N(I, S_i)\). If \(B \cap K \neq \emptyset\), then \(S_j = B \cap K\) is such that \(N(I, S_j) = N(I, S_i)\).

Since the number of delays required in the realization depends only on the number of states in \(M_s\), decompositions for minimizing the number of delays can be obtained even if \(\pi^* = 0\). The following procedure which is an extension of a method for decomposition of normal mode asynchronous machines (Tan et al., 1968), takes advantage of this fact.

For a given machine \(M\), the synchronous submachine \(M_s\) is obtained in the usual manner by associating a state of \(M_s\) to every block of a partition \(\pi\) with S.P. on the set of states of \(M\) (Hartmanis, 1961, 1962). We then construct a flow table \(M'\) which has a row corresponding to every row of \(M\) and a column for every combination of input and internal states of \(M_s\). For any state \(S_i\) of \(M\), let \(B_i\) be the block of \(\pi\) containing \(S_i\). Thus we can identify every internal state of \(M_s\) with a block of \(\pi\).

We fill the entries of \(M'\) as follows: If \(N(I_k, S_i) = S_j\) as specified by \(M\), we enter the state \(S_j\) in row \(S_i\), column \((I_k, B_i)\) of \(M'\). This procedure is repeated for all entries of \(M\). The flow table \(M'\) will contain unspecified
entries. We attempt to make $M'$ normal mode by suitably specifying these entries. For every unstable entry $S_j$ in a column $(I_k, B_i)$ of $M'$, the stable state $S_j$ is entered in row $S_j$, column $(I_k, B_i)$, if that entry is still unspecified. The flow table so obtained will be a normal mode table if and only if every column contains a stable entry corresponding to every unstable entry in it. The flow table of $M_F$ is obtained by applying any state reduction technique (Paull and Unger, 1959; Grasselli and Luccio, 1965) to $M'$, but not merging any pair of states contained in the same block of $\pi$. Compatible columns of the reduced table may also be merged to reduce the dependence of $M_F$ on the inputs or the states of $M_S$. If $M_F$ can be made independent of the states of $M_S$, the decomposition is a parallel decomposition.

For a given flow table $M$ and a submachine $M_S$ defined by the partition $\pi$ with S.P. on the set of states of $M$, a normal mode flow table $M'$ can be constructed using the algorithm given above if and only if the following condition is satisfied: If $N(I_k, S_i) = S_j$ and $S_j \in B_i$, then $N(I_k, S_i) = S_j$ for all states $S_i$ and inputs $I_k$. If $S_j \in B_i$, then the entry in row $S_j$, column $(I_k, B_i)$ of $M'$ will be initially unspecified and may be specified to be $S_j$. However if $S_j \in B_i$, the next state entry in row $S_j$, column $(I_k, B_i)$ will be specified to be $N(I_k, S_j)$ as in $M$. Note that the condition given above for $M'$ to be normal mode is the same as condition (c) of Theorem 1 if $\pi^* = 0$. Since $\pi$ has S.P. and $\pi \cdot \pi^* = 0$ trivially, the series connection of $M_S$ and $M'$ will realize $M$, and $M'$ is normal mode. The state reduction of $M'$ makes $\pi^*$ nontrivial and the realization more economical without affecting the number of delays required. The same procedure may be used to obtain decompositions in which $M_S$ is defined by a set system$^3$ (Hartmanis and Stearns, 1964) instead of a partition with S.P.

The following example taken from Gerace and Gestri (1967b) will demonstrate the decomposition procedure discussed above.

In this example, $M_S$ is obtained by using the partition $\pi = (123, 456, 789)$. The $M'$ is obtained by the construction discussed earlier and $M_F$ is obtained by reducing the number of states of $M'$. An STT state assignment is shown to the right of $M_F$. It can be verified that none of the three variables can be eliminated without introducing critical races. For instance, if $y_5$ is eliminated, the machine may reach the state $A$ instead of $C$ if $\gamma I_5$ is applied when the machine is in state $B$.

$^3$ A set system on the set of states $S$ is a grouping of all members of $S$ into subsets called blocks, $B_i$, such that $B_i \subseteq B_i$ implies $i = j$. 
The method presented in this section is also applicable to serial decompositions in which a normal mode asynchronous machine $M_F$ drives a synchronous machine $M_s$, discussed in a more recent paper by Gerace and Gestri (1968). The asynchronous machine $M_F$ is obtained as discussed in the above-mentioned paper. To obtain $M_s$, we construct a flow table $M'$ whose rows and columns are labelled as discussed earlier in this section. If in the original machine $M$, $N(I_k, S_i) = S_j \in B_j$ and $S_i \in B_i$, where $B_i$ and $B_j$ are blocks of the partition $\pi$ defining $M_F$. 
then in $M'$ we make

$$N((I_k, B_i), S_i) = N((I_k, B_j), S_i) = S_i.$$  

Similarly, the outputs of $M'$ in row $S_i$ and columns $(I_k, B_i)$ and $(I_k, B_j)$ are made the same. That is, the next state and output of $M'$ are the same for the initial and final states of any transition in $M'$. The $M_s$ is obtained by minimizing the number of states of $M'$ without merging states contained in the same block of $\pi$. 

<table>
<thead>
<tr>
<th>$\alpha I_1$</th>
<th>$\alpha I_2$</th>
<th>$\alpha I_3$</th>
<th>$\beta I_1$</th>
<th>$\beta I_2$</th>
<th>$\beta I_3$</th>
<th>$\gamma I_1$</th>
<th>$\gamma I_2$</th>
<th>$\gamma I_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>9</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>9</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>-</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>-</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

$M'$

**Example 1C**

<table>
<thead>
<tr>
<th>$\alpha I_1$</th>
<th>$\alpha I_2$</th>
<th>$\alpha I_3$</th>
<th>$\beta I_1$</th>
<th>$\beta I_2$</th>
<th>$\beta I_3$</th>
<th>$\gamma I_1$</th>
<th>$\gamma I_2$</th>
<th>$\gamma I_3$</th>
<th>$\gamma_1$</th>
<th>$\gamma_2$</th>
<th>$\gamma_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(147) A</td>
<td>A</td>
<td>C</td>
<td>C</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(258) B</td>
<td>B</td>
<td>B</td>
<td>C</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(369) C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>A</td>
<td>C</td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
ON SEQUENTIAL MACHINE DECOMPOSITIONS

<table>
<thead>
<tr>
<th></th>
<th>I₁</th>
<th>I₂</th>
<th>I₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

**M**

**EXAMPLE 2A**

<table>
<thead>
<tr>
<th></th>
<th>I₁</th>
<th>I₂</th>
<th>I₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>(123) α</td>
<td>β</td>
<td>γ</td>
<td>γ</td>
</tr>
<tr>
<td>(45) β</td>
<td>a</td>
<td>a</td>
<td>β</td>
</tr>
<tr>
<td>(1'2'3')γ</td>
<td>β</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

**Mₛ**

**EXAMPLE 2B**

<table>
<thead>
<tr>
<th></th>
<th>I₁</th>
<th>I₂</th>
<th>I₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3'</td>
<td>1'</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>2'</td>
<td>3'</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>2'</td>
<td>2'</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>1'</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2'</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3'</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**Mₐ**

**EXAMPLE 2C**

**EXAMPLE 2.**
DECOMPOSITIONS WITH STATE-SPLITTING

Any \( m \)-block partition with S.P. on the states of a machine \( M \) may be used for decomposing it into a synchronous machine \( M_s \) and a normal mode asynchronous machine \( M_f \), by splitting the states of \( M \) whenever required. In such a decomposition, the machine \( M_s \) may have at most \( 2^m \) states requiring \( \lceil \log_2 m \rceil + 1 \) delays. The machine \( M_f \) may have twice as many states as \( M \), but requires no delays.

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|c|}
\hline
& \alpha I_1 & \alpha I_2 & \alpha I_3 & \beta I_1 & \beta I_2 & \beta I_3 & \gamma I_1 & \gamma I_2 & \gamma I_3 \\
\hline
1 & 4 & 3' & 1' & - & - & - & - & - & 1 \\
2 & 5 & 2' & 3' & 2 & 2 & - & - & 2 & 2 \\
3 & 5 & 2' & 2' & 3 & 3 & - & - & 3 & 3 \\
4 & 4 & - & - & 2 & 2 & 4 & 4 & - & - \\
5 & 5 & - & - & 3 & 3 & 5 & 5 & - & - \\
1' & - & - & 1' & - & - & - & 4 & 3 & 1 \\
2' & - & 2' & 2' & - & - & - & 5 & 2 & 3 \\
3' & - & 3' & 3' & - & - & - & 5 & 2 & 2 \\
\hline
\end{array}
\]

\( M_A' \)

EXAMPLE 2D

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|c|}
\hline
& \alpha I_1 & \alpha I_2 & \alpha I_3 & \beta I_1 & \beta I_2 & \beta I_3 & \gamma I_1 & \gamma I_2 & \gamma I_3 \\
\hline
(35)B & & B & D & D & B & B & B & B & B \\
(2)C & B & D & E & C & C & - & - & C & C \\
(2')D & - & D & D & - & - & - & B & C & B \\
(3)E & - & E & E & - & - & - & B & C & C \\
\hline
\end{array}
\]

\( M_F \)

EXAMPLE 2E
Let $S_i$ be a state of $M$ contained in a block $B_i$ of the partition $\pi$ with S.P. If $N(I_k, S_i) = S_j \neq S_i$ for some input $I_k$ and $S_j \in B_i$, then all states $S_i$ contained in the block $B_i$ are split into $S_i$ and $S_i'$. A new table $M_A$ is constructed as follows: If $N(I_k, S_i) = S_j$ in $M$, then $N(I_k, S_i) = S_j'$ and $N(I_k, S_i') = S_j$ in $M_A$. All other next state entries in the primed state are made identical to the entries in the corresponding unprimed state. The $M_A$ is equivalent to $M$ and has a partition $\pi_A$ with S.P. The blocks of $\pi_A$ consist of the blocks of $\pi$ and additional blocks $B_i'$ corresponding to blocks $B_i$ of $\pi$, whose members were split. Thus $\pi_A$ can have at most $2m$ blocks. The partition $\pi_A$ also satisfies condition (c) of Theorem 1 with respect to $\pi^* = 0$ and may be used for obtaining a decomposition with the method of the preceding section.

The machine $M$ has a two-block partition $\pi = (123, 45)$ with S.P. It can be verified that this partition cannot be used to obtain a two-state machine $M_S$ and a normal mode machine $M_F$. However, the machine $M_A$ obtained by splitting states 1, 2 and 3 has a partition $\pi' = (123, 45, 1'2'3')$ with S.P., which yields the decomposition of $M_A$ into a three-state machine $M_S$ and a five-state machine $M_F$. Two delays are sufficient in the realization of $M_S$.

Since every machine has a trivial one-block partition $I$ with S.P., it follows from the above result that any machine can be decomposed into a two-state machine $M_S$ and a normal mode machine $M_F$. All states of $M$ have to be split and the resulting machine has a two block

---

![Diagram](image.png)

**Fig. 2.**
\[ M_{\alpha} \]

**Example 3A**

\[
\begin{array}{cc}
\alpha & \beta \\
\beta & \alpha \\
\end{array}
\]

**Example 3B**

\[
\begin{array}{ccc}
I_1 & I_2 & I_3 \\
1 & 4' & 3' & 1' \\
2 & 5' & 2' & 3' \\
3 & 5' & 2' & 2' \\
4 & 2' & 2' & 4' \\
5 & 3' & 3' & 5' \\
1' & 4 & 3 & 1 \\
2' & 5 & 2 & 3 \\
3' & 5 & 2 & 2 \\
4' & 2 & 2 & 4 \\
5' & 3 & 3 & 5 \\
\end{array}
\]

**Example 3C**

**Example 3.**
### Example 3D

<table>
<thead>
<tr>
<th>$aI_1$</th>
<th>$aI_2$</th>
<th>$aI_3$</th>
<th>$\beta I_1$</th>
<th>$\beta I_2$</th>
<th>$\beta I_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4'</td>
<td>3'</td>
<td>1'</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>5'</td>
<td>2'</td>
<td>3'</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>5'</td>
<td>2'</td>
<td>2'</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2'</td>
<td>4'</td>
<td>4'</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>3'</td>
<td>5'</td>
<td>5'</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>1'</td>
<td>-</td>
<td>-</td>
<td>1'</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>2'</td>
<td>2'</td>
<td>2'</td>
<td>2'</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>3'</td>
<td>3'</td>
<td>3'</td>
<td>3'</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>4'</td>
<td>4'</td>
<td>-</td>
<td>4'</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5'</td>
<td>5'</td>
<td>-</td>
<td>5'</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

$M_A'$

### Example 3E

<table>
<thead>
<tr>
<th>$aI_1$</th>
<th>$aI_2$</th>
<th>$aI_3$</th>
<th>$\beta I_1$</th>
<th>$\beta I_2$</th>
<th>$\beta I_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(11')</td>
<td>4'</td>
<td>3'</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>5'</td>
<td>2'</td>
<td>3'</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>5'</td>
<td>2'</td>
<td>2'</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2'</td>
<td>4'</td>
<td>4'</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>3'</td>
<td>5'</td>
<td>5'</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>2'</td>
<td>2'</td>
<td>2'</td>
<td>2'</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>3'</td>
<td>3'</td>
<td>3'</td>
<td>3'</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>4'</td>
<td>4'</td>
<td>-</td>
<td>4'</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5'</td>
<td>5'</td>
<td>-</td>
<td>5'</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

$M_F$

### Example 3E
partition $\pi_A = (\alpha, \beta)$ with S.P., one containing all the unprimed states and the other containing all the primed states. Since transitions in $M_A$ will always be from an unprimed state to a primed state or vice versa, the machine $M_S$ defined by $\pi_A = (\alpha, \beta)$ will have identical columns which can be merged into a single column as shown below:

\[
\begin{array}{c|c}
\alpha & \beta \\
\beta & \alpha \\
\end{array}
\]

The two-state machine $M_S$ can be realized by a trigger flip-flop which changes state when its input is 1 and remains in its previous state when the input is 0.

Figure 2 shows the single-delay realization of any sequential machine using the decomposition discussed above.

The trigger flip-flop should be such that its output changes exactly once during a clock pulse. The magnitude of the delay $D$ and the width of the clock pulse should be such that the clock signal becomes 0 before the output of the delay changes, thus preventing $M_F$ from changing more than once during a transition. Alternatively, the delay may be included in the realization of the trigger flip-flop, as in the model of Fig. 1.

A decomposition of the machine $M$ of Example 2, using a two-state machine $M_S$ is shown above. Though a single-delay realization of $M$ is possible, $M_F$ now has nine states and requires more state variables than in Example 2 for its realization.

CONCLUSION

We have shown that any synchronous sequential machine can be decomposed into a series connection of a synchronous machine $M_S$ and a normal mode asynchronous machine $M_F$. The latter can be realized without delay elements if the clock signal is supplied to it and a single transition time state assignment is used. Using state-splitting, a machine which has an $m$-block partition with S.P. can be realized with $\lceil \log_2 m \rceil + 1$ delay elements. Since every machine has the trivial partition $I$ with S.P., a decomposition leading to a single-delay realization can be obtained at the expense of increasing the number of state variables required for the realization of the asynchronous submachine.

Received: April 15, 1969
REFERENCES


TAN, C. J., MENON, P. R., AND FRIEDMAN, A. D. (1968), Structural simplification and decomposition of asynchronous sequential circuits, Conference Record, Ninth Annual Symposium on Switching and Automata Theory, 7-19.
