Some Dangers in State Reduction of Sequential Machines

J. Hartmanis and R. E. Stearns

General Electric Research Laboratory, Schenectady, New York

In this paper we investigate the negative effects of state reduction on the realization of a sequential machine. It is shown that state reduction can destroy realizations of a given sequential machine from sets of smaller machines and thus lead to a sequential machine that is harder to realize.

To understand some of the reasons why and when state reduction should be carried out, several results are obtained which describe the changes of the structure of a sequential machine under state reduction. It is seen that the undesirable effects of state reduction are closely associated with the failure of certain distributive laws to hold between the partitions used for state reduction and the partitions used in the realization of the unreduced machine.

INTRODUCTION

In this paper we shall study the effects of state reduction (Moore, 1956; Hartmanis, 1961) on the realization of a sequential machine and show that state reduction may lead to a machine that is harder to realize than an unreduced machine. We limit ourselves here to synchronous, completely specified Moore type sequential machines (Moore, 1956), although what we say will usually apply to a wider class of machines. After explaining the danger of state reduction in general, we shall investigate why and when state reduction destroys the loop-free and reduced dependence realizations of sequential machines as analyzed by Hartmanis (1961, 1962a) and Stearns and Hartmanis (1961).

First, we wish to illustrate and define two types of realizations (see also (Huffman, 1954) in which the importance of these two types of realization is discussed for asynchronous sequential machines). Consider the machine $A$ of Fig. 1. We shall synthesize this machine in the conventional way. The steps of this synthesis are shown in Fig. 2.

The first step is to assign a two-variable code name to each state.
The code of Fig. 2 is the one we picked. There is one code name (10) left over and we call it state "d." Next we write the table in binary form with "don't care" conditions. These "don't care" conditions are filled to keep the equations simple. This leads to a four-state machine $A'$ of which $A$ is a submachine. For each state of $A$, there is a corresponding state in $A'$, and the transitions are always from corresponding states into corresponding states. We now make a formal definition.

**Definition 1.** Let $M'$ be a sequential machine with the input set $I'$ and state set $S'$. Then two nonvoid subsets $I$ of $I'$ and $S$ of $S'$ define a submachine $M$ of $M'$ if and only if any input $I_5$ in $I$ maps all the states of the subset $S$ into $S$.

Most of the approaches to the state assignment problem and the problem of machine decomposition that have been published are directed toward finding a realization of $M$ as a submachine of some larger machine $M'$. This is certainly true of the authors' own work (Hartmanis, 1961, 1962a; Stearns and Hartmanis, 1961). We must not forget, however, that the essence of a realization is the input-output relation and not the association of a single name to a single state. Thus machine $A^*$ of Fig. 3 is also a realization of machine $A$.

The reader will notice that machine $A$ is obtained from $A^*$ by state reduction. We say that $A$ is obtained from $A^*$ by the partition $\{a_1, a_2; \overline{b}; \overline{c}\}$. From Hartmanis (1961), we know that a partition reduces a
machine if and only if the partition is output consistent and has the substitution property (S.P.). We make the following definition.

**Definition 2.** We shall say that the machine $M$ is an image of the machine $M^*$ under $\pi$ if the partition $\pi$ on the set of states of $M^*$ has S.P., and if $M$ is obtained from $M^*$ by the state reduction defined by $\pi$.

Thus we have found two realizations which are essentially different in character, even though they may be considered special cases of the most general case—that of realizing a machine as the image of a sub-machine. The important point of this discussion is that, working with machine $A$ in the usual manner, one will overlook the image type of realization. If one were given the machine $A$, it is difficult to say what larger machines he should analyze. If, however, one is given the unreduced machine $A^*$, we contend that he should consider the possibilities for coding the machine without reduction; for if he blindly reduces the machine before analysis, he may eliminate the possibility of finding the simplest realization. This is the danger of state reduction. In a subsequent example, we shall show that the cost of state reduction can be significant.

**Effects of State Reduction on Machine Structure**

We shall now illustrate and discuss the effect of state reduction on the partition with S.P. analysis of Hartmanis (1961, 1962a).

Consider the machine $B^*$ and its reduced form $B$ given in Fig. 4.

---

**Fig. 3.** Machine $A^*$ with code and equations

```
<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01</td>
<td>10</td>
</tr>
</tbody>
</table>

$\sigma_1$: $c$ $c$ $c$ $c$ $0$ $\sigma_2$: $c$ $c$ $c$ $c$ $0$

$b$: $a_1$, $b$, $c$, $a_2$, $b$, $b$, $b$, $c$ $1$

$c$: $b$, $b$, $b$, $c$ $1$

$\sigma_1 \rightarrow 00$, $\sigma_2 \rightarrow 10$, $b \rightarrow 01$, $c \rightarrow 11$

$Y_1 = \bar{Y}_2 + X_1$, $Y_2 = Y_1 + \bar{Y}_2 + X_2$

---

**Fig. 4.** Machines $B^*$ and $B$ with their lattices of S.P. partitions
To determine whether the machine $B^*$ can be reduced and whether it can be realized by a loop-free combination of smaller machines, we compute the partitions with $S.P.$ on this machine (Hartmanis, 1961, 1962a). These partitions are:

$$
\begin{align*}
0 &= \{0; 1; 2; 3; 4; 5; 6; 7\}, \\
\pi_1 &= \{0, 1; 2, 3; 4, 5; 6, 7\}, \\
\pi_2 &= \{0, 1, 2; 3, 4, 5, 6, 7\},  \\
\pi_3 &= \{0; 1; 2, 5; 3; 4; 6; 7\}, \\
\pi_4 &= \{0, 1; 2, 3, 4, 5; 6, 7\}, \\
\pi_5 &= \{0; 1; 2, 5; 3, 4; 6; 7\}, \\
\pi_6 &= \{0; 1, 2, 3, 4, 5, 6, 7\}, \\
I &= \{0, 1, 2, 3, 4, 5, 6, 7\}.
\end{align*}
$$

To reduce states of a sequential machine, we must have a nontrivial partition with $S.P.$ such that all states contained in one block have the same output (Hartmanis, 1961). For $B^*$, $\pi_2$ is such an output consistent partition and reduces $B^*$ to $B$ by merging the states "3" and "4". Note that the states of $B$ are the blocks of $\pi_3$.

We now return to machine $B^*$ and observe that

$$
\pi_2 > \pi_1 > 0.
$$

This implies that $B^*$ can be realized from three two-state machines connected in series (Hartmanis, 1962a).

In Fig. 5 we have given the internal state assignment and a schematic representation of the realization corresponding to the three partitions.

The logical equations for this realization are given below and they

![Fig. 5. State assignment for machine $B^*$ and the corresponding realization](image-url)
require 19 diodes for this two-layer "and/or" gate realization of the state behavior of machine $B^*$:

\[
\begin{align*}
Y_1' &= \bar{Y}_1X = f_1(Y_1, X) \\
Y_2' &= \bar{X}\bar{Y}_2 + Y_1Y_2X = f_2(Y_1, Y_2, X) \\
Y_3' &= \bar{X}\bar{Y}_3 + \bar{Y}_2\bar{Y}_3 + Y_2Y_3\bar{X} = f_3(Y_2, Y_3, X) \\
Z &= Y_1Y_2\bar{Y}_3.
\end{align*}
\]

On the other hand, it can be computed that the only nontrivial partitions with $S.P.$ for the reduced machine $B$ are

\[
\pi_4 = \{0, 1; 2, 3', 5; 6, 7\} \quad \text{and} \quad \pi_6 = \{0; 1; 2, 5; 3'; 6; 7\}.
\]

Thus we see that $B$ can be realized only from two three-state machines connected in series, and this would require a four-variable-state assignment. If we insist on using only a three-binary variable-state assignment, then we cannot realize $B$ from smaller component machines. The best known assignment for machine $B$ requires 22 diodes to realize its state behavior (using shared logic) and it was discovered by the reviewer of this paper: $0 \rightarrow 001$, $1 \rightarrow 011$, $2 \rightarrow 010$, $3' \rightarrow 000$, $5 \rightarrow 110$, $6 \rightarrow 111$, $7 \rightarrow 101$. Another assignment requiring 24 diodes was supplied by D. B. Armstrong of Bell Telephone Laboratories. Thus in this particular case state reduction has led us to a machine which is harder to realize than the unreduced machine. Also in this case the reason is easily seen. In reducing $B^*$ to $B$, the clean flow of information in $B^*$ was "smeared," and the partitions with $S.P.$ that determined a simple loop-free realization were destroyed.

To get a better understanding as to how state reduction "smears" information, let us investigate the relation between the $S.P.$ lattice for an unreduced machine $M^*$ and the $S.P.$ lattice for machine $M$ obtained from $M^*$ by an (output-consistent) partition $\pi$ with $S.P.$ Now the states of $M$ may be thought of as blocks of $\pi$, and a partition on the states of $M$ as blocks of blocks of states of $M^*$. It is often convenient to think of such a partition $\tau$ for $M$ as a partition on the states of $M^*$ such that $\tau \geq \pi$. A little reflection shows the following.

**Lemma 1.** If $L^*$ is the set of partitions with $S.P.$ on machine $M^*$ and $M^*$ reduces to $M$ by $\pi_R$ then $L = \{\pi \in L^* \mid \pi \geq \pi_R\}$ is the set of partitions with $S.P.$ for $M$. In other words, the state reduction of $M^*$ by $\pi_R$ changes the partition $\pi$ on $M^*$ into $\pi + \pi_R$ on $M$. 
Observe that \( \pi_R \) is the zero partition for \( M \). Note also that, except for the trivial case \( \pi_R = 0 \), \( M \) has fewer partitions with S.P. than \( M^* \). This loss of structure is usually compensated in part or in full by the fact that the blocks of partitions in \( L \) have fewer elements than their counterparts in \( L^* \), for here we must remember that \( L \) partitions are really blocks of blocks.

We shall now use Lemma 1 to obtain necessary and sufficient conditions that a parallel decomposition of a machine \( M \) is preserved by state reduction. From Hartmanis (1962a), we know that the state behavior of \( M \) can be realized by two smaller machines \( M_1 \) and \( M_2 \) operating in parallel if and only if there exist two nontrivial partitions with S.P., \( \pi_1 \) and \( \pi_2 \), on the set state of \( M \) such that \( \pi_1 \cdot \pi_2 = 0 \). The machines \( M_1 = M_{\pi_1} \) and \( M_2 = M_{\pi_2} \) compute the block of \( \pi_1 \) and \( \pi_2 \), respectively, in which the state of \( M \) is contained. If \( \pi \) is a partition with S.P. on \( M \) which reduces it to \( M_\pi \), then it reduces \( M_{\pi_1} \) to \( M_{\pi_1+\pi} \) and \( M_{\pi_2} \) to \( M_{\pi_2+\pi} \), because the partitions \( \pi_1 \) and \( \pi_2 \) are changed to \( \pi_1 + \pi \) and \( \pi_2 + \pi \) by the state reduction. Since \( \pi \) is the "zero" partition for \( M_\pi \), we have obtained the following result.

**Theorem 1.** If \( M \) is a completely specified sequential machine and \( \pi, \pi_1, \) and \( \pi_2 \) are partitions with S.P. such that the parallel machines \( M_{\pi_1} \) and \( M_{\pi_2} \) realize the state behavior of \( M \), then the reduced parallel machines \( M_{\pi_1+\pi} \) and \( M_{\pi_2+\pi} \) realize the state behavior of \( M_\pi \) if and only if \( (\pi_1 + \pi) \cdot (\pi_2 + \pi) = \pi \).

**Example.** Machine \( C \) of Fig. 6 has many partitions with S.P. including:

\[
\begin{align*}
\pi_1 &= \{1, 6; 2, 5; 3, 8; 4, 7\}, \\
\pi_2 &= \{1, 2, 3, 4; 5, 6, 7, 8\}, \\
\pi_3 &= \{1, 3, 5, 7; 2, 4, 6, 8\}, \\
\pi &= \{1, 2; 3, 4; 5, 6; 7, 8\}.
\end{align*}
\]

Notice that \( \pi \) is the largest output-consistent partition with S.P. Furthermore the partition \( \pi_1 \) leads to a four-state machine \( C_{\pi_1} \) which can be used in parallel with either \( C_{\pi_2} \) or \( C_{\pi_3} \) (both two-state machines) to realize the state behavior of \( C \). Now \( (\pi_1 + \pi) \cdot (\pi_2 + \pi) = \pi \) but \( (\pi_1 + \pi) \cdot (\pi_3 + \pi) > \pi \). Thus the parallel connection of the two machines \( C_{\pi_1+\pi} \) and \( C_{\pi_2+\pi} \) realize the state behavior of \( C_\pi \), whereas the machines \( C_{\pi_1+\pi} \) and \( C_{\pi_3+\pi} \) do not.

The S.P. lattice for the reduced machine \( C_\pi \) contains only four par-
Fig. 6. Machine C

<table>
<thead>
<tr>
<th></th>
<th>I₁</th>
<th>I₂</th>
<th>I₃</th>
<th>I₄</th>
<th>I₅</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Considerations: $I(= \pi₃ + \pi), \pi₂(= \pi₂ + \pi), \pi₁ + \pi,$ and $\pi.$ Thus, although $C_{\pi₁}$ and $C_{\pi₃}$ realize $C_{\pi}$ as an image, there is no trace of this on the reduced lattice. On the other hand, the realization of $C$ by $C_{\pi₁+\pi}$ and $C_{\pi₂+\pi}$ is evident from the unreduced lattice if one marks the output-consistent partitions, for then it is seen at once that $(\pi₁ + \pi) \cdot (\pi₂ + \pi)$ is output-consistent.

A lattice $L$ is said to be distributive if and only if for any three elements $X, Y,$ and $Z$ in $L,$ $X(Y + Z) = XY + XZ.$ If the $S.P.$ lattice of a machine is distributive, then $(\pi + \pi₁) \cdot (\pi + \pi₂) = (\pi + \pi₁) \cdot \pi + (\pi + \pi₁) \cdot \pi₂ = \pi \cdot \pi + \pi \cdot \pi₁ + \pi \cdot \pi₂ + \pi₁ \cdot \pi₂.$ Now if $\pi₁ \cdot \pi₂ = 0,$ this final expression is equal to $\pi$ since $\pi \cdot \pi = \pi,$ $\pi₁ \cdot \pi ≤ \pi,$ and $\pi₂ \cdot \pi ≤ \pi.$ We have proved the corollary.

**Corollary.** If the $S.P.$ lattice for machine $M$ is distributive, then the condition of Theorem 1 must hold, and all parallel decompositions of $M$ are preserved under state reduction.

It is interesting to recall that the holding of the distributive law in the lattice $L^*$ of a machine is also related to the uniqueness of parallel decompositions of sequential machines (Hartmanis, 1962b). Thus we can see that, in a number of different aspects, the distributive law guarantees “nice” properties of sequential machines that are not possessed by other machines. It is the conviction of the authors that the distributive law and other algebraic laws on the lattice of partitions with $S.P.$ lead to a natural classification of machines and should be further investigated.
We have looked at a simple kind of decomposition and have seen that even these parallel decompositions are sometimes lost under state reduction. It is not surprising, therefore, that the more general reduced dependence and loop-free realizations of Stearns and Hartmanis (1961) and Hartmanis (1962a) are even more changed by state reduction. This is because the component machines have inputs from other component machines and the partition which reduces the entire machine need not be consistent with the information exchanged by these component machines. In the terminology of Stearns and Hartmanis (1961) the information flow inequalities are not necessarily preserved under state reduction. Thus there are many conditions that have to hold before component-wise reduction is possible. It can be shown, however, that if the partitions in question distribute, then the realization is preserved. It is important to observe that the partitions involved in the more general realization do not all have $S.P.$; so even if the $S.P.$ lattice is distributive, the realization may be destroyed by state reduction.

CONCLUSION

In this paper we have shown that state reduction can destroy the simplest realizations of a sequential machine and thus should be treated with caution. We showed this by studying the effects which state reduction has on the structure of a sequential machine. We have found that merging states sometimes disrupted the systematic information flow in the unreduced machines and destroyed simple realizations from component machines that could not be detected by the $S.P.$ lattice or partition pair lattice for the reduced machine.

These results indicate that, in general, state reduction should not be carried out before a sequential machine is analyzed for state assignments by means of structural properties or by other methods. It also indicates that the state assignment problem (and the general decomposition problem) must be generalized by permitting multiple codes for the internal states of a given sequential machine. Thus, ideally, one should, in the analyses of a sequential machine $M$, study the class of machines of which $M$ is a submachine, an image machine or a sub-image machine and select from these sets the machine which is most easily realized.

ACKNOWLEDGMENT

The authors owe a debt of gratitude to E. H. Farr of the General Engineering Laboratory of the General Electric Co. for supplying the first example of a se-
quential machine which is harder to realize after state reduction. Farr's sequential machine came from an actual design problem, and this paper is a direct product of the discussions and analyses of sequential machines that were started by this striking example.

Received: April 20, 1962.

References


Hartmanis, J. (1962a), Loop-free structure of sequential machines, Information and Control 5, No. 1, 25–43.


