ECE 27000 S22 P 16

Decomposition and State Assignment

One approach to state assignment is based on reducing the dependence among the state variables.

In general, $Y_i = f_i(y_1, \cdots, y_k, x_1, \cdots, x_m)$.
If we can ensure that Y_i will depend only on a subset of $y_1, \cdots, y_k$, we are likely to reduce the number of literals in its implementation.

Example:

PS	NS, z $x = 0$	$x = 1$
A	$A, 0$	$D, 1$
B	$A, 0$	$C, 0$
C	$C, 0$	$B, 0$
D	$C, 0$	$A, 1$

State assignment:

	PS	NS, z $x = 0$	$x = 1$
A	00	00, 0	11, 1
B	01	00, 0	10, 0
C	10	10, 0	01, 0
D	11	10, 0	00, 1

$Y_1 = x'y_1 + xy_1' = f_1(x, y_1)$
$Y_2 = xy_2' = f_2(x, y_2)$
$z = xy_1'y_2' + xy_1y_2 = f_0(x, y_1, y_2)$

Y_1 is independent of y_2.
Y_2 is independent of y_1.

Achieving reduced dependence among state variables is based on decomposition of the machine into two or more components.

Decomposition is based on partitioning of the machine states into subsets with certain properties.

	$y_1 y_2$
A	00
B	01
C	10
D	11

In the example, the state assignment defines two partitions on the states of the machine:

The partition corresponding to y_1 divides the states into states for which $y_1 = 0$, and states for which $y_1 = 1$. We obtain $\pi_1 = \{\overline{A, B}; \overline{C, D}\}$.

The partition corresponding to y_2 divides the states into states for which $y_2 = 0$, and states for which $y_2 = 1$. We obtain $\pi_2 = \{\overline{A, C}; \overline{B, D}\}$.

Note that every state has a unique code. Therefore, no two states are in the same block of both partitions.

Partitions

A partition π on a set S is a collection of disjoint subsets of S whose union is S, i.e.,
$\pi = \{B_\alpha\}$ such that
$B_\alpha \cap B_\beta = \phi$ for $\alpha \neq \beta$, and
$\bigcup \{B_\alpha\} = S$.

Each subset is called a *block* of the partition.

We denote partitions in the following way:
$S = \{A, B, C, D, E, F\}$
$\pi = \{\overline{A, B}; \overline{C, D, E}; \bar{F}\}$

We use $s \equiv t(\pi)$ to indicate that s and t are in the same block of π.

We define two partitions, $\pi(0)$ and $\pi(1)$, as follows.
$S = \{A, B, C, D, E, F\}$
$\pi(0) = \{\bar{A}; \bar{B}; \bar{C}; \bar{D}; \bar{E}; \bar{F}\}$
$\pi(1) = \{\overline{A, B, C, D, E, F}\}$

For π_1 and π_2 on S, we say that $\pi_1 \leq \pi_2$ iff every block of π_1 is contained in a block of π_2.

Example:
$\{\overline{A, B}; \overline{C, D, E}; \bar{F}\} \leq \{\overline{A, B}; \overline{C, D, E, F}\}$.

$\leq$ is a partial order.

If π_1 and π_2 are partitions on S, then $\pi_1 \cdot \pi_2$ is the partition on S such that $s \equiv t(\pi_1 \cdot \pi_2)$ iff $s \equiv t(\pi_1)$ and $s \equiv t(\pi_2)$.

Example:

$S = \{A, B, C, D, E, F\}$

$\pi_1 = \{\overline{A, B}; \overline{C, D, E}; \bar{F}\}$

$\pi_2 = \{\overline{A, B, C}; \overline{D, E, F}\}$

$\pi_1 \cdot \pi_2 = \{\overline{A, B}; \bar{C}; \overline{D, E}; \bar{F}\}$.

$\pi_1 + \pi_2$ is the partition on S such that $s \equiv t(\pi_1 + \pi_2)$ iff there exists a sequence in S, $s = s_0, s_1, \cdots, s_n = t$, such that $s_i \equiv s_{i+1}(\pi_1)$ or $s_i \equiv s_{i+1}(\pi_2)$.

Example:

$S = \{A, B, C, D, E, F\}$

$\pi_1 = \{\overline{A, B}; \overline{C, D, E}; \bar{F}\}$

$\pi_2 = \{\overline{A, B, C}; \overline{D, E, F}\}$

$\pi_1 + \pi_2 = \{\overline{A, B, C, D, E, F}\}$

$\pi_1 = \{\overline{A, B}; \overline{C, D}; \bar{E}; \bar{F}\}$

$\pi_2 = \{\overline{A, B, C}; \bar{D}; \overline{E, F}\}$

$\pi_1 + \pi_2 = \{\overline{A, B, C, D}; \overline{E, F}\}$

These two operations are meet and join for partitions. Partitions on a set form a lattice.

In our view of state assignment as a partitioning problem, each state variable y_i defines a partition π_i on the set of states of the machine.
Two states are in the same block of π_i iff they are assigned the same value of y_i.

If the assignment is such that each state is assigned a unique code, then we must have $\pi_1 \cdot \pi_2 \cdot \cdots \cdot \pi_k = \pi(0)$.

In the first example we have seen:
$\pi_1 = \{\overline{A, B}; \overline{C, D}\}$.
$\pi_2 = \{\overline{A, C}; \overline{B, D}\}$.
$\pi_1 \cdot \pi_2 = \pi(0)$.

Closed Partitions

Also called partitions with substitution property.

Definition: A partition π on the set of states of a machine M is said to be *closed* if, for every two states S_i and S_j in the same block of π and any input symbol I_k of the machine, the I_k-successors of S_i and S_j are in the same block of π.

Example:

PS	NS, z $x = 0$	$x = 1$
A	$A, 0$	$D, 1$
B	$A, 0$	$C, 0$
C	$C, 0$	$B, 0$
D	$C, 0$	$A, 1$

$\pi_1 = \{\overline{A, B}; \overline{C, D}\}$ is closed:

$AB \xrightarrow{0} AA$

$AB \xrightarrow{1} CD$

$CD \xrightarrow{0} CC$

$CD \xrightarrow{1} AB$

Same example:

PS	NS, z	
	$x = 0$	$x = 1$
A	$A, 0$	$D, 1$
B	$A, 0$	$C, 0$
C	$C, 0$	$B, 0$
D	$C, 0$	$A, 1$

$\pi_2 = \{\overline{A, C}; \overline{B, D}\}$ is closed:

$AC \xrightarrow{0} AC$

$AC \xrightarrow{1} BD$

$BD \xrightarrow{0} AC$

$BD \xrightarrow{1} AC$

The importance of closed partitions:

Consider $\pi_1 = \{\overline{A, B}; \overline{C, D}\}$.
If we use a state assignment where $y_1 = 0$ for AB and $y_1 = 1$ for CD, we will be able to determine the value of Y_1 from x and y_1.

	NS, z	
PS	$x = 0$	$x = 1$
AB	AA	CD
CD	CC	AB

		Y_1	
	y_1	$x = 0$	$x = 1$
AB	0	0	1
CD	1	1	0

$Y_1 = x'y_1 + xy_1'$

In contrast, $\tau = \{\overline{A, D}; \overline{B, C}\}$ is not a closed partition.

$AD \xrightarrow{0} AC$, and AC is not a block of the partition.

Suppose that we assign y_1 based on τ and y_2 based on another partition.

In order to determine Y_1 for AD and $x = 0$, we will need to know whether the present state is A or D.

For this, we will need to know y_1 and y_2.

We will obtain $Y_1 = f_1(x, y_1, y_2)$.

In general, let M be a machine with k state variables $y_1, y_2, \cdots, y_k$. If there exists a closed partition π on the states of M, and if r state variables are assigned to the blocks of π so that all the states contained in each block are assigned the same values of $y_1, y_2, \cdots, y_r$, then the next state variables $Y_1, Y_2, \cdots, Y_r$ are independent of the remaining $k - r$ variables.

In the example, consider $\pi_2 = \{\overline{A, C}; \overline{B, D}\}$.
If we use a state assignment where $y_2 = 0$ for AC and $y_2 = 1$ for BD, we will be able to determine the value of Y_2 from x and y_2.

	NS, z	
PS	$x = 0$	$x = 1$
AC	AC	BD
BD	AC	AC

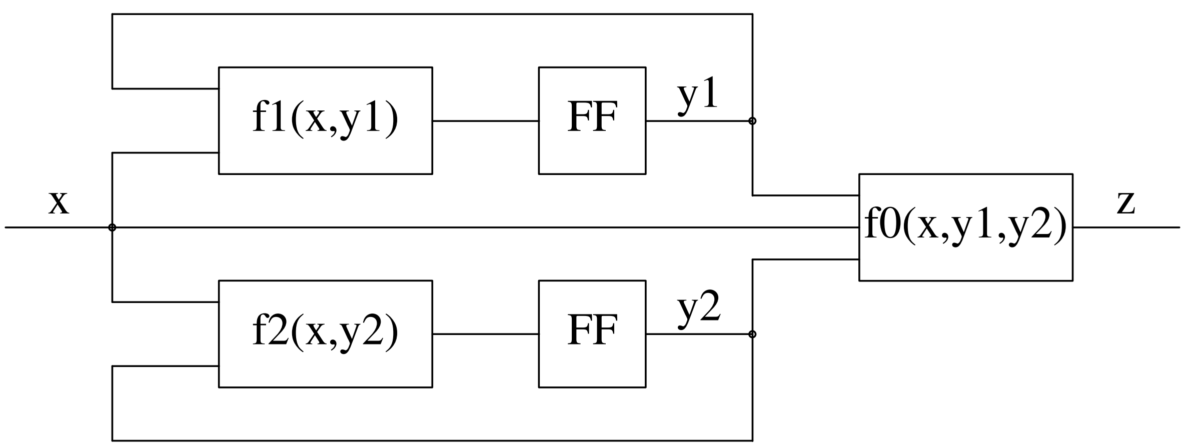
		Y_2	
	y_2	$x = 0$	$x = 1$
AC	0	0	1
BD	1	0	0

$Y_2 = x y_2'$

Since $\pi_1 \cdot \pi_2 = \pi(0)$, y_1 and y_2 provide a unique code to every state.

	$y_1 y_2$
A	00
B	01
C	10
D	11

We obtain a decomposition of the machine into two parallel machines
(a parallel implementation).



A parallel decomposition is obtained if there exist two nontrivial closed partitions π_1 and π_2 such that $\pi_1 \cdot \pi_2 = \pi(0)$.

If there is only one closed partition, we can use it to obtain a serial decomposition.

Example:

	NS, z	
PS	$x = 0$	$x = 1$
A	B	C
B	A	D
C	D	B
D	C	A

This machine has only one closed partition,
$\pi = \{\overline{A, B}; \overline{C, D}\}$.

If we assign y_1 according to π, we will obtain an independent component.

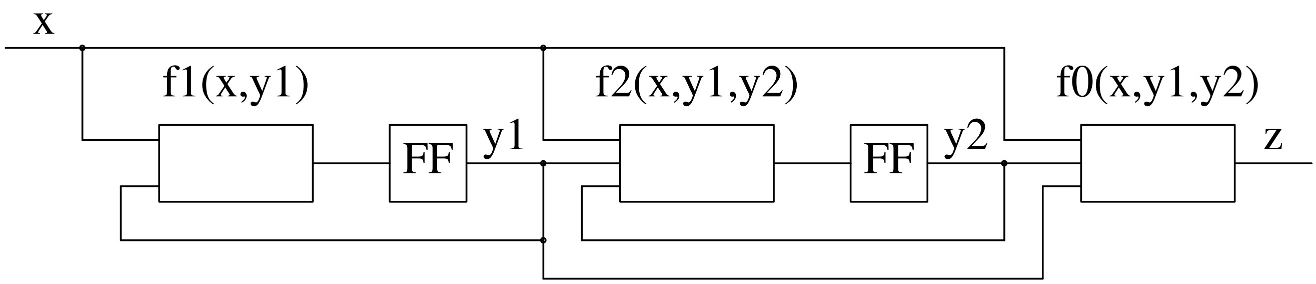
We need y_2 that will distinguish the states with the same code according to y_1.
A possible partition τ corresponding to y_2 is
$\tau = \{\overline{A, C}; \overline{B, D}\}$ or
$\tau = \{\overline{A, D}; \overline{B, C}\}$.

Let us use $\tau = \{\overline{A, C}; \overline{B, D}\}$.

We have $\pi \cdot \tau = \pi(0)$.
The state assignment:

	$y_1 y_2$
A	00
B	01
C	10
D	11

The general form of the decomposition:



If a partition has m blocks, it requires $\lceil \log_2 m \rceil$ state variables.

Example:

	NS, z	
PS	$x = 0$	$x = 1$
A	A	D
B	C	E
C	B	D
D	C	A
E	B	B

This machine has only one closed partition,
$\pi = \{\overline{A, B, C}; \overline{D, E}\}$.

If we assign y_1 according to π, we will obtain an independent component.

We need y_2 and y_3 that will distinguish the states with the same code according to y_1.
A possible partition τ corresponding to y_2 and y_3 is
$\tau = \{\overline{A, D}; \overline{B, E}; \bar{C}; \}$ or
$\tau = \{\overline{A, D}; \bar{B}; \overline{C, E}; \}$ or
$\tau = \{\bar{A}; \overline{B, D}; \overline{C, E}; \}$ or
$\tau = \{\overline{A, E}; \overline{B, D}; \bar{C}; \}$, and so on.

Let us use $\tau = \{\overline{A, D}; \overline{B, E}; \bar{C}; \}$ with
$y_2 y_3 = 00$ for $\overline{A, D}$,
$y_2 y_3 = 01$ for $\overline{B, E}$, and
$y_2 y_3 = 10$ for $\bar{C}$.

This results in the state assignment

	$y_1 y_2 y_3$
A	000
B	001
C	010
D	100
E	101

The general form of the decomposition:
$Y_1 = f_1(x, y_1)$.
$Y_2 = f_1(x, y_1, y_2, y_3)$.
$Y_3 = f_1(x, y_1, y_2, y_3)$.
$z = f_0(x, y_1, y_2, y_3)$.

It is possible to use more than two partitions to obtain a state assignment.

Example:

	NS, z	
PS	$x = 0$	$x = 1$
A	H	B
B	F	A
C	G	D
D	E	C
E	A	C
F	C	D
G	B	A
H	D	B

It is not possible to find two closed partitions such that $\pi_1 \cdot \pi_2 = \pi(0)$.

The machine has the following closed partitions.
$\pi_1 = \{\overline{A, B, C, D}; \overline{E, F, G, H}\}.$
$\pi_2 = \{\overline{A, D, E, H}; \overline{B, C, F, G}\}.$
$\pi_3 = \{\overline{A, D}; \overline{B, C}; \overline{E, H}; \overline{F, G}\}.$

Note that $\pi_3 = \pi_1 \cdot \pi_2$.

If we assign y_1 according to π_1 and y_2 according to π_2, we will obtain two parallel components.

Since $\pi_1 \cdot \pi_2 \neq \pi(0)$, y_1 and y_2 will not give a unique code to every state.
We need y_3 that will distinguish the states with the same code according to y_1 and y_2.

A possible partition τ corresponding to y_3 is
$\tau = \{\overline{A, B, E, F}; \overline{C, D, G, H}\}$, or
$\tau = \{\overline{A, B, E, G}; \overline{C, D, F, H}\}$, or
$\tau = \{\overline{A, B, F, H}; \overline{C, D, E, G}\}$, and so on.
Let us use $\tau = \{\overline{A, B, G, H}; \overline{C, D, E, F}\}$.

The state assignment:

	$y_1 y_2 y_3$
A	000
B	010
C	011
D	001
E	101
F	111
G	110
H	100

$$Y_1 = x' y_1'$$
$$Y_2 = x' y_2 + x y_2'$$
$$Y_3 = f_3(x, y_1, y_2, y_3)$$

Computing the Closed Partitions

The closed partitions form a lattice based on the following property.

Theorem: The product $\pi_1 \cdot \pi_2$ and the sum $\pi_1 + \pi_2$ of two closed partitions on the set of states of M are also closed.

Outline of the proof for $\pi_1 \cdot \pi_2$:
Consider $s_i \equiv s_j(\pi_1 \cdot \pi_2)$.
We have that $s_i \equiv s_j(\pi_1)$ and $s_i \equiv s_j(\pi_2)$.
Consider any I-successor s_p, s_q of s_i, s_j.
Since $s_i \equiv s_j(\pi_1)$, $s_p \equiv s_q(\pi_1)$.
Since $s_i \equiv s_j(\pi_2)$, $s_p \equiv s_q(\pi_2)$.
Therefore, $s_p \equiv s_q(\pi_1 \cdot \pi_2)$.

We will generate the minimal nontrivial closed partitions, and form the rest of the closed partitioned by taking all possible sums.

The minimal closed partitions are obtained by requiring that a pair of states $S_i S_j$ would be placed in the same block, for every S_i and S_j.

Example:

PS	NS, z $x = 0$	$x = 1$
A	E	B
B	E	A
C	D	A
D	C	F
E	F	C
F	E	C

$AB \rightarrow AB$

$\pi_1 = \{\overline{A, B}; \bar{C}; \bar{D}; \bar{E}; \bar{F}\}.$

$AC \rightarrow DE, AB$

$ABC, DE \rightarrow DE, AB, CF$

$ABCF, DE \rightarrow DE, ABC$

$\pi_2 = \{\overline{A, B, C, F}; \overline{D, E}\} \geq \pi_1.$

$AD \to CE, BF \to DF, AC$
$ABCDEF$
$\pi(1)$

$AE \to EF, BC$
$AEF, BC \to EF, BC, DE$
$ADEF, BC \to CEF, BCF, DE$
$ABCDEF$
$\pi(1)$

$\ldots$

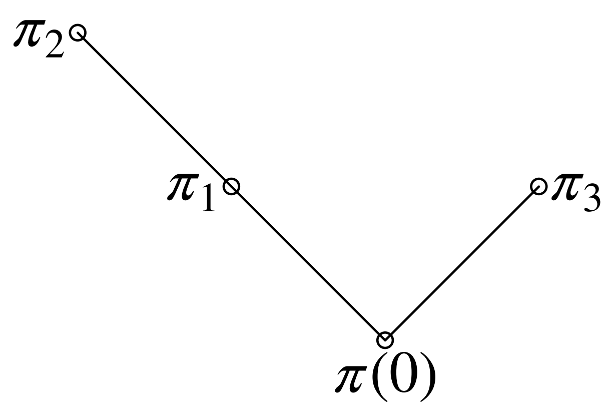
$EF \to EF.$
$\pi_3 = \{\bar{A}; \bar{B}; \bar{C}; \bar{D}; \overline{E, F}\}.$

The lattice:

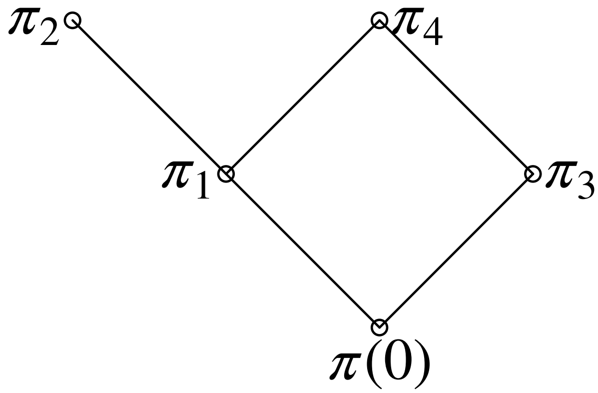
$\pi_1 = \{\overline{A, B}; \bar{C}; \bar{D}; \bar{E}; \bar{F}\}.$

$\pi_2 = \{\overline{A, B, C, F}; \overline{D, E}\}.$

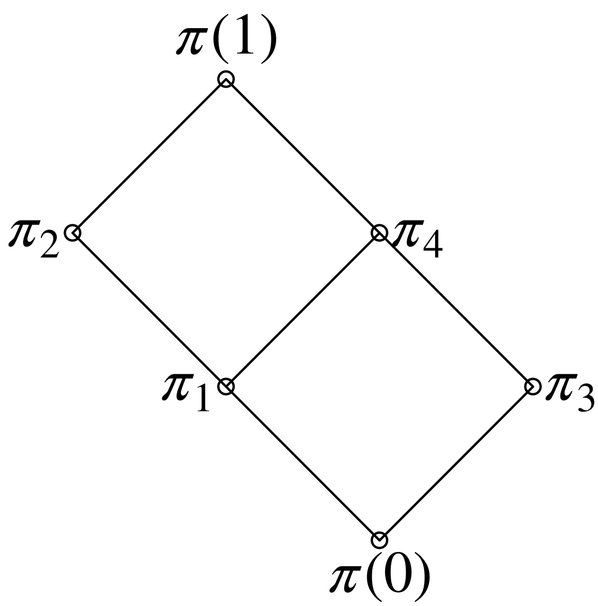
$\pi_3 = \{\bar{A}; \bar{B}; \bar{C}; \bar{D}; \overline{E, F}\}.$



$\pi_1 + \pi_3 = \{\overline{A, B}; \bar{C}; \bar{D}; \overline{E, F}\} = \pi_4.$



$\pi_2 + \pi_4 = \pi(1)$.



A state assignment:

Option 1:
Use $\pi_2 = \{\overline{A, B, C, F}; \overline{D, E}\}$ for y_1.
Use $\tau = \{\overline{A, D}; \overline{B, E}; \bar{C}; \bar{F}\}$ for y_2 and y_3.
$Y_1 = Y_1(x, y_1)$.
$Y_2 = Y_2(x, y_1, y_2, y_3)$.
$Y_3 = Y_3(x, y_1, y_2, y_3)$.

option 2:
Use $\pi_4 = \{\overline{A, B}; \bar{C}; \bar{D}; \overline{E, F}\}$ for y_1 and y_2.
Use $\tau = \{\overline{A, E, C}; \overline{B, F, D}\}$.
$Y_1 = Y_1(x, y_1, y_2)$.
$Y_2 = Y_2(x, y_1, y_2)$.
$Y_3 = Y_3(x, y_1, y_2, y_3)$.

Note that π_1 and π_3 are not useful in this case since they each require three state variables that will depend on all three state variables.

Example of serial decomposition:

	NS, z	
PS	$x = 0$	$x = 1$
A	$G, 1$	$D, 1$
B	$H, 0$	$C, 0$
C	$F, 1$	$G, 1$
D	$E, 0$	$G, 0$
E	$C, 1$	$B, 1$
F	$C, 0$	$A, 0$
G	$A, 1$	$E, 1$
H	$B, 0$	$F, 0$

$AB \rightarrow GH, CD \rightarrow EF$

$\pi_2 = \{\overline{A, B, G, H}; \overline{C, D, E, F}\}$.

$AG \rightarrow DE \rightarrow CE, BG$

$ABG, CDE \rightarrow AGH, CEF$

$ABGH, CDEF$

$\pi_1 = \{\overline{A, B, G, H}; \overline{C, D, E, F}\}$.

The lattice of closed partitions:

$$\pi(1)$$

$$\pi_1 = \{\ \overline{A, B, G, H}; \overline{C, D, E, F}\}$$

$$\pi_2 = \{\ \overline{A, B}; \overline{C, D}; \overline{E, F}; \overline{G, H}\}$$

$$\pi(0)$$

Assign y_1 to π_1.

Use π_2 by selecting a partition τ_1 such that $\pi_2 = \pi_1 \cdot \tau_1$.
Use $\tau_1 = \{\overline{A, B, C, D}; \overline{E, F, G, H}\}$.
Assign y_2 based on τ_1.
We will have $Y_2 = f_2(x, y_1, y_2)$ (independent of y_3).

Select a partition τ_2 such that $\pi_2 \cdot \tau_2 = \pi(0)$.
Use $\tau_2 = \lambda_o = \{\overline{A, C, E, G}; \overline{B, D, F, H}\}$.
Assign y_3 based on τ_2.

The state assignment yields the following equations.
$Y_1 = f_1(x, y_1)$.
$Y_2 = f_2(x, y_1, y_2)$.
$Y_3 = f_3(x, y_1, y_2, y_3)$.
$z = f_0(x, y_3)$.

Reducing the Output Dependency

Definition: A partition λ_o on the states of a machine M is said to be *output − consistent* if, for every block of λ_0 and every input, all the states contained in the same block have the same outputs.

Example:

	NS, z	
PS	$x = 0$	$x = 1$
A	$B, 1$	$D, 0$
B	$A, 0$	$C, 1$
C	$D, 0$	$A, 1$
D	$C, 1$	$B, 0$

$\lambda_o = \{\overline{A, D}; \overline{B, C}\}$.

If we assign y based on λ_o, we can write the output z as a function of x and y:

PS	z: $x = 0$	z: $x = 1$
AD $y = 0$	1	0
BC $y = 1$	0	1

This machine has a closed partition
$\pi = \{\overline{A, B}; \overline{C, D}\}$.
$\pi \cdot \lambda_o = \pi(0)$.

A state assignment based on π and λ_o:

	$y_1 y_2$	$Y_1 Y_2, z$ $x = 0$	$x = 1$
A	00	01, 1	10, 0
B	01	00, 0	11, 1
C	11	10, 0	00, 1
D	10	11, 1	01, 0

$Y_1 = x' y_1 + x y_1'$
$Y_2 = f_2(x, y_1, y_2)$.
$z = x' y_2' + x y_2$.

Note that we had another choice in this case: Instead of λ_o, we could have used $\tau = \{\overline{A, C}; \overline{B, D}\}$. However, this would have resulted in the output depending on both y_1 and y_2.

A closed partition $\pi \le \lambda_0$ is also output-consistent.

Reducing the Input Dependency

Definition: A partition λ_i on the states of a machine M is said to be input-consistent if, for every state S_i of M and all input combinations $I_1, I_2, \cdots, I_p$, the next states $I_1 S_i, I_2 S_i, \cdots, I_p S_i$ are in the same block of λ_i.

Example:

	NS, z	
PS	$x = 0$	$x = 1$
A	$D, 0$	$C, 1$
B	$C, 0$	$D, 0$
C	$E, 0$	$F, 1$
D	$F, 0$	$F, 0$
E	$B, 0$	$A, 1$
F	$A, 0$	$B, 0$

An input-consistent partition:

$A \rightarrow CD$

$B \rightarrow CD$

$C \rightarrow EF$

$D \rightarrow FF$

$E \rightarrow AB$

$F \rightarrow AB$

$\lambda_i = \{\overline{AB}; \overline{CD}; \overline{EF}\}.$

If we assign state variables y_1 and y_2 to λ_i, Y_1 and Y_2 will be independent of the inputs.

If we have a closed partition $\pi \geq \lambda_i$, then the state variables we assign to π will be independent of the inputs and the other state variables.

For the example above:
$\lambda_i = \{\overline{AB}; \overline{CD}; \overline{EF}\}$ is a closed partition.
We have $\lambda_o = \{\overline{ACE}; \overline{BDF}\}$.
$\lambda_i \cdot \lambda_o = \pi(0)$.
We can use λ_i and λ_o to find a state assignment:

		$Y_1Y_2Y_3, z$	
	$y_1y_2y_3$	$x=0$	$x=1$
A	000	011, 0	010, 1
B	001	010, 0	011, 0
C	010	100, 0	101, 1
D	011	101, 0	101, 0
E	100	001, 0	000, 1
F	101	000, 0	001, 0

The resulting equations:

$Y_1 = y_2$

$Y_2 = y_1' y_2'$

$Y_3 = f_3(x, y_1, y_2, y_3)$

$z = x y_3'$

General Decomposition and State Assignment

Many machines do not have closed partitions, or do not have enough closed partitions to define each state variable based on a closed partition.

In such cases, it is possible to use partition pairs.

Definition: A *partition pair* (τ, τ') on the states of a machine M is an ordered pair of partitions such that if S_i and S_j are in the same block of τ, then for every input symbol I_k, the I_k-successors of S_i and S_j are in the same block of τ'.

If $\tau = \tau'$, then τ is a closed partition.

Otherwise, if we assign y_1 to τ and y_2 to τ', then $Y_2 = f_2(x, y_1)$.

Property: Suppose that (τ, τ') is a partition pair. If $\tau_p \leq \tau$, then (τ_p, τ') is also a partition pair, and if $\tau'_q \geq \tau'$, then (τ, τ'_q) is also a partition pair.

Outline of proof:
If $\tau_p \leq \tau$, then the successor sets of τ_p are not larger than the successor sets of τ. Therefore, they are contained in τ'.

If $\tau'_q \geq \tau'$, then the successor sets of τ, which are contained in τ', are also contained in τ'_q.

Example:

PS	NS, z			
	$x_1x_2 = 00$	01	11	10
A	C	A	D	B
B	E	C	B	D
C	C	D	C	E
D	E	A	D	B
E	E	D	C	E

Partition pairs:

$(\tau_1, \tau'_1) = (\{\overline{A, B, D}; \overline{C, E}\}, \{\overline{A, C, E}; \overline{B, D}\})$

$(\tau_2, \tau'_2) = (\{\overline{A, D}; \bar{B}; \overline{C, E}\}, \{\overline{A, B, D}; \overline{C, E}\})$

$(\tau_3, \tau'_3) = (\{\overline{A, C}; \bar{B}; \overline{D, E}\}, \{\overline{A, C, D}; \overline{B, E}\})$

Let us assign:

y_1 based on τ'_1,

y_2 based on τ'_2, and

y_3 based on τ'_3.

Since $\tau_1 = \tau'_2$, we have
$Y_1 = f_1(x_1, x_2, y_2)$.

We have $\tau'_2 \cdot \tau'_3 = \{\overline{A, D}; \bar{B}; \bar{C}; \bar{E}\}$.
This implies that $\tau'_2 \cdot \tau'_3 \leq \tau_2$.
Therefore, given y_2 and y_3 (or the block of $\tau'_2 \cdot \tau'_3$), we can find Y_2.
$Y_2 = f_2(x_1, x_2, y_2, y_3)$.

We have $\tau'_1 \cdot \tau'_3 = \{\overline{A, C}; \bar{B}; \bar{D}; \bar{E}\}$.
This implies that $\tau'_1 \cdot \tau'_3 \leq \tau_3$.
Therefore, given y_1 and y_3 (or the block of $\tau'_1 \cdot \tau'_3$), we can find Y_3.
$Y_3 = f_3(x_1, x_2, y_1, y_3)$.