Course Name: Digital Hardware Design
Course Code: 17B1NEC741

Finite State Machine-5

Dr. Arti Noor
Dean, Academic Affairs
Electronics and Communication Engineering,
Jaypee Institute of Information Technology, Noida

State Minimization in FSM

- It is the process of finding and eliminating the redundant states.
- Redundant states are also referred to as equivalent states.
- Eliminating equivalent states reduces the number of flip-flops.
- Simplify combinational logic of the finite-state machine.

Approaches:

1. Row Equivalence Method
2. Implication Table Approach
3. Partitioning Approach

State Minimization in FSM

Row Equivalence Method

- Check that rows of a state table which are equivalent.
- Condition for equivalence:
  - Same O/P for both the states.
  - K successors must be same for all the input conditions.

Present State	Next State		Output
	$x = 0$	$x = 1$	
S_0	S_1	S_2	0
S_1	S_3	S_4	0
S_2	S_5	S_6	0
S_3	S_3	S_4	1
S_4	S_5	S_6	0
S_5	S_3	S_4	0
S_6	S_5	S_6	0

State Minimization in FSM
Row Equivalence Method

- S1 & S5 are same
- Combine S1 and S5. Define as S1*

Reduced Table

Present State	Next State		Output
	x=0	x=1	
S0	S1*	S2	0
S1*	S3	S4	0
S2	S5(S1*)	S6	0
S3	S3	S4	1
S4	S5(S1*)	S6	0
S6	S5(S1*)	S6	0

Present State	Next State		Output
	$x = 0$	$x = 1$	
S_0	S_1	S_2	0
S_1	S_3	S_4	0
S_2	S_5	S_6	0
S_3	S_3	S_4	1
S_4	S_5	S_6	0
S_5	S_3	S_4	0
S_6	S_5	S_6	0

State Minimization in FSM
Row Equivalence Method

- S2,S4 and S6 are same.
- Combine all Define as S2*

Present State	Next State		Output
	x=0	x=1	
S0	S1*	S2	0
S1*	S3	S4	0
S2	S5(S1*)	S6	0
S3	S3	S4	1
S4	S5(S1*)	S6	0
S6	S5(S1*)	S6	0

Reduced Table

Present State	Next State		Output
	x=0	x=1	
S0	S1*	S2*	0
S1*	S3	S4(S2*)	0
S2*	S5(S1*)	S4(S2*)	0
S3	S3	S4(S2*)	1

State Minimization in FSM
Row Equivalence Method

Final Reduced Table

Present State	Next State		Output
	x=0	x=1	
S0	S1*	S2*	0
S1*	S3	S4(S2*)	0
S2*	S5(S1*)	S4(S2*)	0
S3	S3	S4(S2*)	1

Present State	Next State		Output
	$x = 0$	$x = 1$	
S_0	S_1^*	S_2^*	0
S_1^*	S_3	S_2^*	0
S_2^*	S_1^*	S_2^*	0
S_3	S_3	S_2^*	1

State Minimization
Implication Table Approach- Example 1

- A chart is prepared to find the equivalent steps.
- A square Xij contains the equivalent states between Xi and Xj.
- It can be modified as Xij=Xji and thus the triangle above the diagonal can be removed.
- Also, the diagonal can be removed.

	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
S_0								
S_1						$X_{5,1}$		
S_2								
S_3								
S_4								
S_5		$X_{1,5}$						
S_6								
S_7								

	S_0	S_1	S_2	S_3	S_4	S_5	S_6
S_1							
S_2							
S_3							
S_4							
S_5							
S_6							
S_7							

State Minimization
Implication Table Approach- Example 1

Minimize the following state table:

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

State Minimization
Implication Table Approach- Example 1

A								
B						Xij		
C								
D								
E								
F		Xij						
G								
H								
	A	B	C	D	E	F	G	H

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

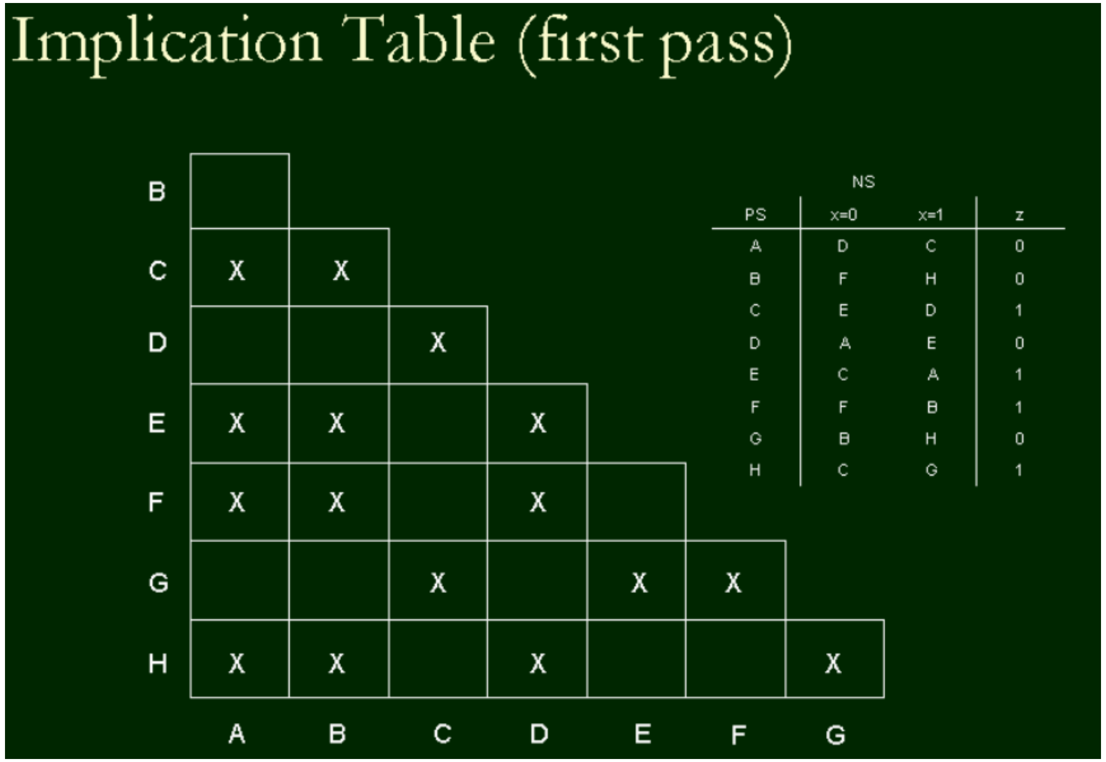
State Minimization
Implication Table Approach- Example 1

	A	B	C	D	E	F	G	H
A								
B						Xij		
C								
D								
E								
F		Xij						
G								
H								

	A	B	C	D	E	F	G
B							
C							
D							
E							
F							
G							
H							

State Minimization
Implication Table Approach- Example 1

Fill the State table in the Chart.
First step



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

State Minimization
Implication Table Approach- Example 1

Fill the State table in the Chart.

First step

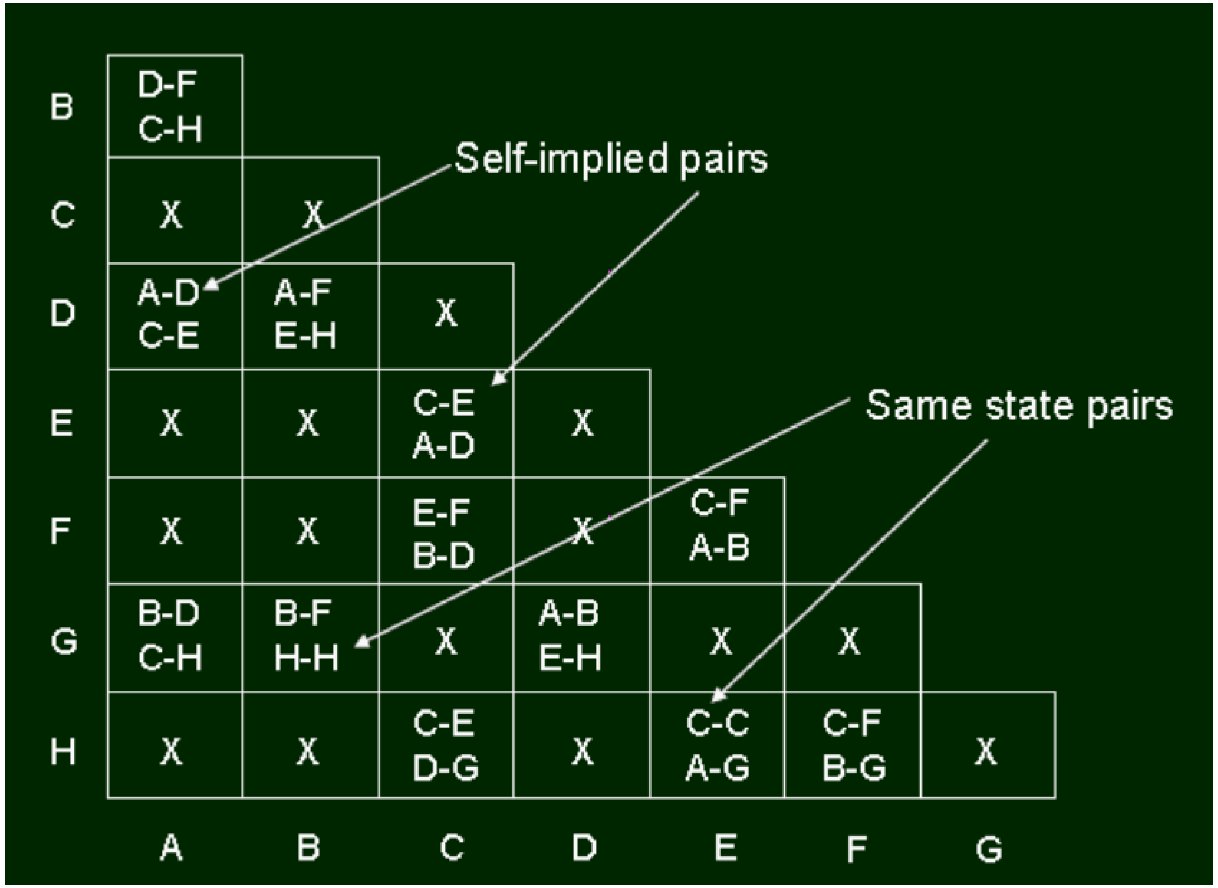
	A	B	C	D	E	F	G
B	D-F C-H						
C	X	X					
D	A-D C-E	A-F E-H	X				
E	X	X	C-E A-D	X			
F	X	X	E-F B-D	X	C-F A-B		
G	B-D C-H	B-F H-H	X	A-B E-H	X	X	
H	X	X	C-E D-G	X	C-C A-G	C-F B-G	X

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

State Minimization
Implication Table Approach- Example 1

Second step

- Remove self implied pairs
  - A-D in cell A-D
  - C-E in cell C-E
- Remove same state pairs
  - H-H in cell B-G
  - C-C in cell H-E



State Minimization
Implication Table Approach- Example 1

Second step

	A	B	C	D	E	F	G
B	D-F C-H						
C	X	X					
D	A-D C-E	A-F E-H	X				
E	X	X	C-E A-D	X			
F	X	X	E-F B-D	X	C-F A-B		
G	B-D C-H	B-F H-H	X	A-B E-H	X	X	
H	X	X	C-E D-G	X	C-C A-G	C-F B-G	X

Self-implied pairs

Same state pairs

	A	B	C	D	E	F	G
B	D-F C-H						
C	X	X					
D	C-E	A-F E-H	X				
E	X	X	A-D	X			
F	X	X	E-F B-D	X	C-F A-B		
G	B-D C-H	B-F	X	A-B E-H	X	X	
H	X	X	C-E D-G	X	A-G	C-F B-G	X

Self-implied pairs

Same state pairs

State Minimization
Implication Table Approach- Example 1

Third step

One column (or row) at a time, eliminate implied pairs

Searching for equivalent states:

- **Start top to the bottom starting from the left side of the chart.**
- **Check the possibility of combining the state.**
- **Rrepeat the step until no further combination is possible.**

B	~~D-F C-H~~						
C	X	X					
D	C-E	~~A-F E-H~~	X				
E	X	X	A-D	X			
F	X	X	~~E-F B-D~~	X	~~C-F A-B~~		
G	B-D C-H	~~B-F~~	X	~~A-B E-H~~	X	X	
H	X	X	C-E D-G	X	A-G	~~C-F B-G~~	X
	A	B	C	D	E	F	G

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

State Minimization
Implication Table Approach- Example 1

Combine equivalent states (based on coordinates of cells, not contents)

- A ≡ D, C ≡ E in example
  - Equivalence is pairwise
    - A ≡ B, B ≡ C implies A ≡ C (transitive)

Construct reduced state table

B	~~D-F C-H~~						
C	X	X					
D	C-E	~~A-F E-H~~	X				
E	X	X	A-D	X			
F	X	X	~~B-F B-D~~	X	~~C-F A-B~~		
G	~~B-D C-H~~	~~B-F~~	X	~~A-B E-H~~	X	X	
H	X	X	~~C-E D-G~~	X	~~A-G~~	~~C-F B-G~~	X
	A	B	C	D	E	F	G

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

State Minimization
Implication Table Approach- Example 1

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

PS	NS x=0	NS x=1	z
A	A*	C	0
B	F	H	0
C	C*	A*	1
F	F	B	1
G	B	H	0
H	C	G	1

State Minimization
Partitioning Approach

State Minimization through Partitioning:

- Form an initial partition (P1) that includes all states.
- Form a second partition (P2) by separating the states into two blocks based upon their output values.
- Form a third partition (P3) by separating the states into blocks corresponding to the next state values.
- Continue partitioning until two successive partitions are the same (i.e. $P_{N-1} = P_N$).
- All states in any one block are equivalent.

  Equivalent states can be combined into a single state.

State Minimization
Partitioning Approach- Example 1

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

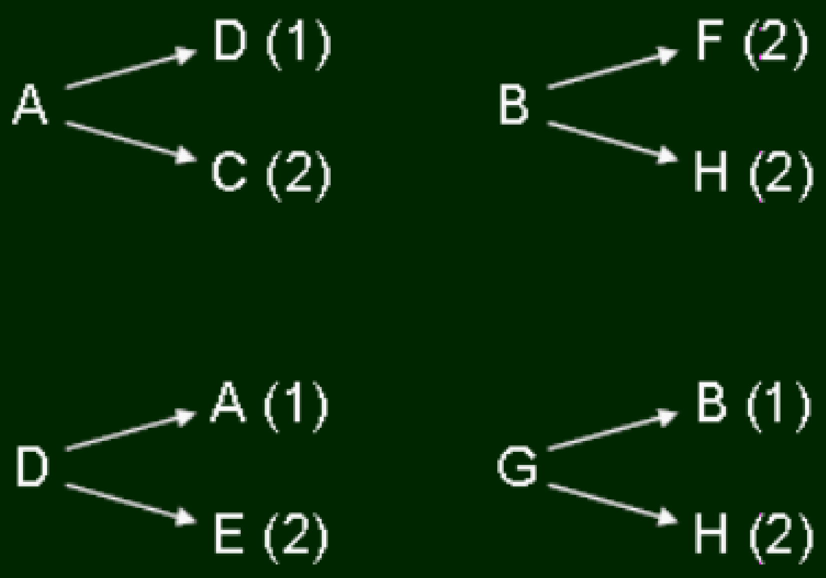
State Minimization
Partitioning Approach- Example 1

- P_0 = (ABCDEFGH)
- P_1 is obtained by splitting states having different outputs
  - P_1 =(ABDG)(CEFH)
    - Block 1 = ABDG, Block 2 = CEFH

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

State Minimization
Partitioning Approach- Example 1

- Obtain P_2
  - Block 1 = ABDG, Block 2 = CEFH



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

State Minimization
Partitioning Approach- Example 1

- Obtain P_2 (cont)
  - Block 1 = ABDG, Block 2 = CEFH

$C \rightarrow E\ (2)$, $C \rightarrow D\ (1)$

$E \rightarrow C\ (2)$, $E \rightarrow A\ (1)$

$F \rightarrow F\ (2)$, $F \rightarrow B\ (1)$

$H \rightarrow C\ (2)$, $H \rightarrow G\ (1)$

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

State Minimization
Partitioning Approach- Example 1

- $P_2 = (ADG)(B)(CEFH)$
  - Block 1 = ADG
  - Block 2 = B
  - Block 3 = CEFH

State Minimization
Partitioning Approach- Example 1

- Obtain P_3
  - $P_2 = (ADG)(B)(CEFH)$

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

A → D (1), C (3)

D → A (1), E (3)

G → B (2), H (3)

C → E (3), D (1)

E → C (3), A (1)

F → F (3), B (2)

H → C (3), G (1)

State Minimization
Partitioning Approach- Example 1

- $P_3 = (AD)(G)(B)(CEH)(F)$
  - Block 1 = AD, block 2 = G, block 3 = B, block 4 = CEH and block 5 = F

State Minimization
Partitioning Approach- Example 1

- Obtain P_4
  - $P_3 = (AD)(G)(B)(CEH)(F)$

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

$A \to D\,(1)$, $A \to C\,(4)$

$D \to A\,(1)$, $D \to E\,(4)$

$C \to E\,(4)$, $C \to D\,(1)$

$E \to C\,(4)$, $E \to A\,(1)$

$H \to C\,(4)$, $H \to G\,(2)$

State Minimization
Partitioning Approach- Example 1

- $P_4 = (AD)(G)(B)(CE)(H)(F)$
  - Block 1 = AD, block 2 = G, block 3 = B, block 4 = CEH, block 5 = H and block 6 = F

State Minimization
Partitioning Approach- Example 1

- Obtain P_5
  - $P_4 = (AD)(G)(B)(CE)(H)(F)$

$A \to D\ (1)$, $A \to C\ (4)$

$D \to A\ (1)$, $D \to E\ (4)$

$C \to E\ (4)$, $C \to D\ (1)$

$E \to C\ (4)$, $E \to A\ (1)$

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

- Obtain P_5 (cont)
  - No blocks split from P_5

State Minimization
Partitioning Approach- Example 1

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

States A and D; C and E will be combined.

	NS		
PS	x=0	x=1	z
A	A*	C	0
B	F	H	0
C	C*	A*	1
F	F	B	1
G	B	H	0
H	C	G	1