CS221: Digital Design

FSM Optimization : Implication Chart Method

A. Sahu
Dept of Comp. Sc. & Engg.
Indian Institute of Technology Guwahati

Outline

- FSM State Optimization
  - Row Matching Method (Last Class)
  - Partitioning Method (Last Class)
  - Implication Chart Method
- FSM State Encoding
  - Binary, gray, One-hot
  - Heuristic Based

FSM State Minimization

- Minimizing number of state reduce
  - Requirement of bigger size state register
  - Possibly reduce the CCC

Some Definitions

- **State Equivalence**: S1 and S2 are equivalent if for every input sequence applied to machine goes to same NS and Output
  - If $S1(t+1)=S2(t+1)$ and $Z1=Z2$ then $S1=S2$

- **Distinguishable States**: Two states S1 and S2 are Distinguishable iff there exist at least one finite input sequence which produce different outputs from S1 and S2

Methods

- Row Matching Method or Partitioning Method
- Implication Chart Method

Implication Chart Methods

FSM Reduction: Implication Chart Method

Problem:

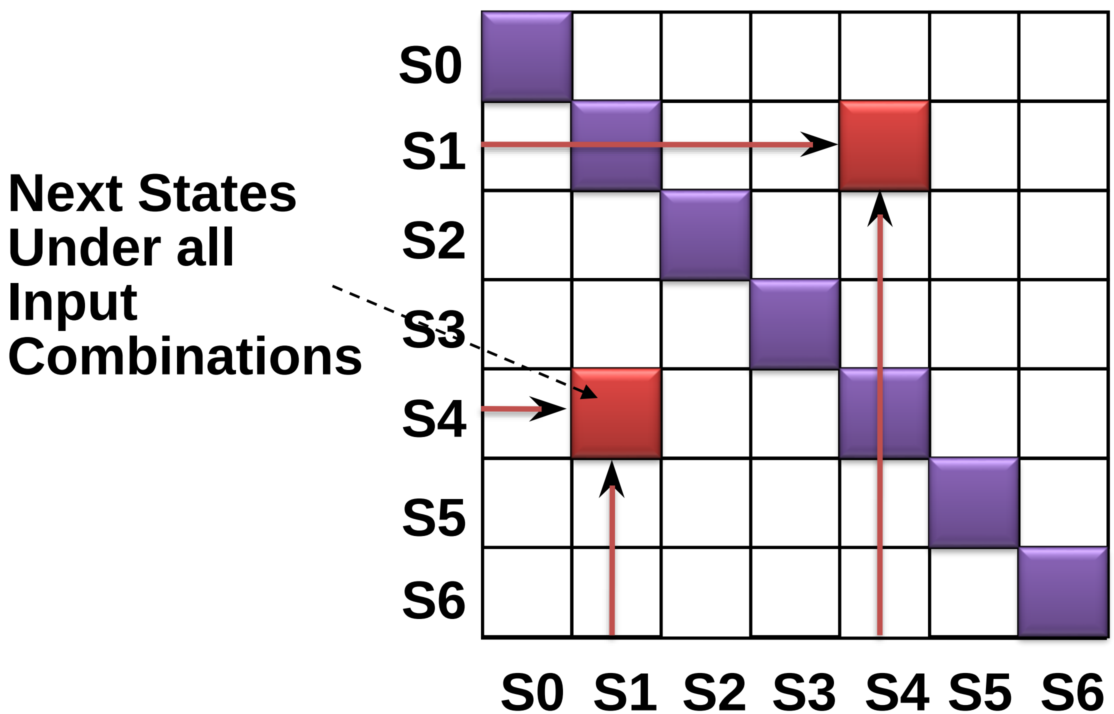
Single input X, Single output Z

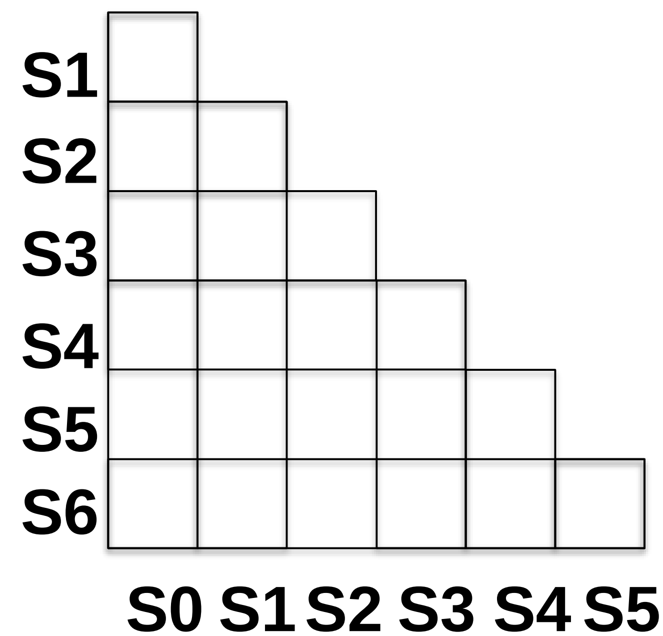
Output a 1 whenever the serial sequence 010 or 110 has been observed at the inputs

Input Sequence	Present State	Next State X=0	Next State X=1	Output X=0	Output X=1
Reset	S_0	S_1	S_2	0	0
0	S_1	S_3	S_4	0	0
1	S_2	S_5	S_6	0	0
00	S_3	S_0	S_0	0	0
01	S_4	S_0	S_0	1	0
10	S_5	S_0	S_0	0	0
11	S_6	S_0	S_0	1	0

Implication Chart Method

Enumerate all possible combinations of states taken two at a time





Implication Chart

Naive Data Structure:
X_{ij} will be the same as X_{ji}
Also, can eliminate the diagonal

Implication Chart Method

Filling in the Implication Chart

Entry $\mathbf{X}_{\mathbf{ij}}$ — Row is $\mathbf{S}_{\mathbf{i}}$, Column is $\mathbf{S}_{\mathbf{j}}$

$\mathbf{S}_{\mathbf{i}}$ is equivalent to $\mathbf{S}_{\mathbf{j}}$ if outputs are the same and next states are equivalent

$\mathbf{X}_{\mathbf{ij}}$ contains the next states of $\mathbf{S}_{\mathbf{i}}$, $\mathbf{S}_{\mathbf{j}}$ which must be equivalent if S_i and S_j are equivalent

If $\mathbf{S}_{\mathbf{i}}$, $\mathbf{S}_{\mathbf{j}}$ have different output behavior, then $\mathbf{X}_{\mathbf{ij}}$ is crossed out

Implication Chart Method

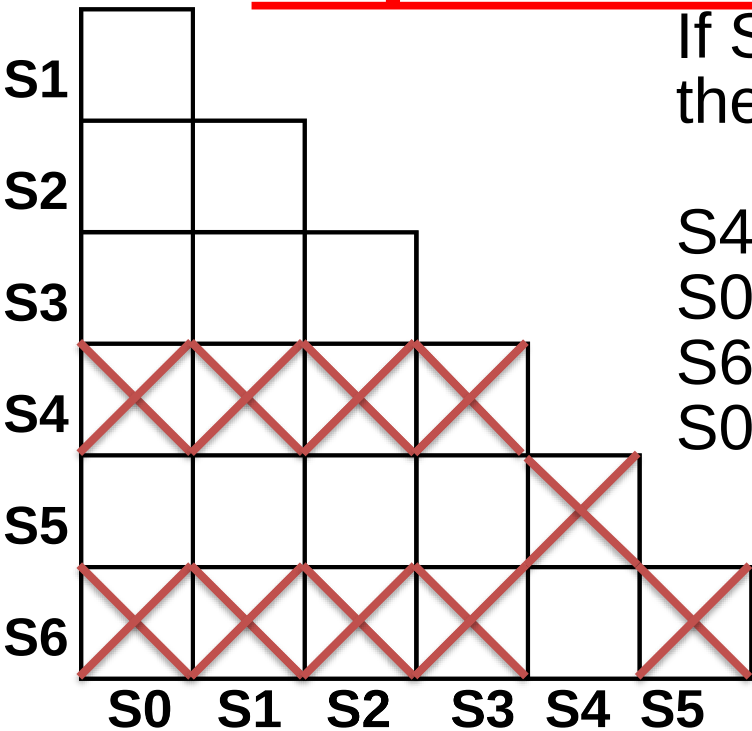
If S_i, S_j have different output behavior, then X_{ij} is crossed out

	S0	S1	S2	S3	S4	S5
S1						
S2						
S3						
S4						
S5						
S6						

Starting Implication Chart

Input	P State	NS X=0	NS X=1	Output X=0	Output X=1
Reset	S_0	S_1	S_2	0	0
0	S_1	S_3	S_4	0	0
1	S_2	S_5	S_6	0	0
00	S_3	S_0	S_0	0	0
01	S_4	S_0	S_0	1	0
10	S_5	S_0	S_0	0	0
11	S_6	S_0	S_0	1	0

Implication Chart Method



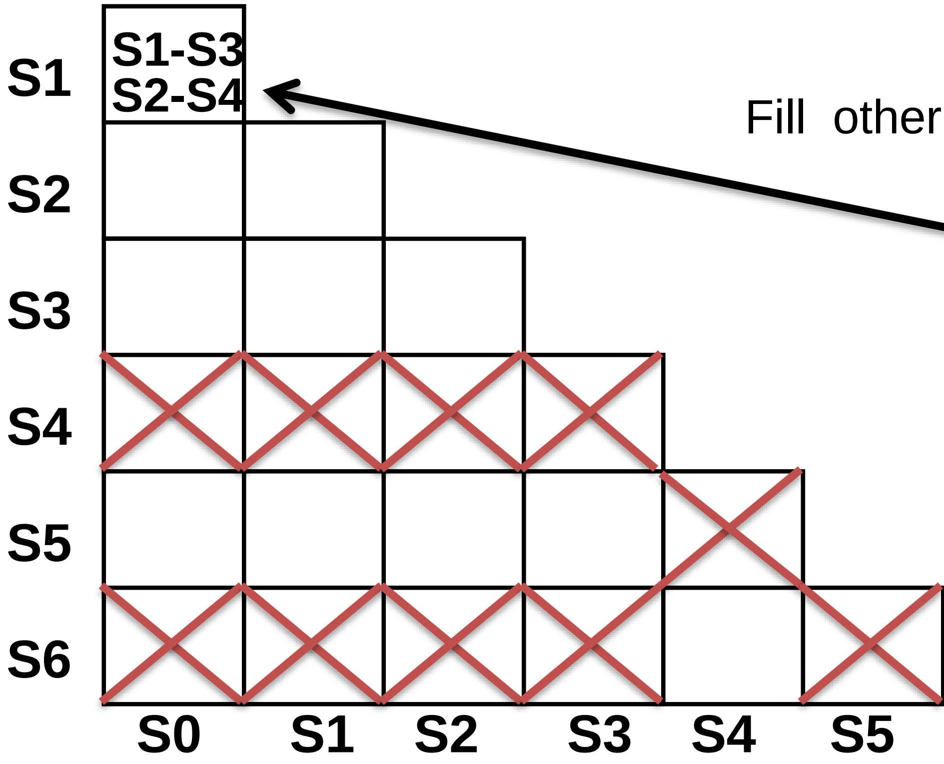
Starting Implication Chart

If Si, Sj have different output behavior, then Xij is crossed out

S4 have different out put behavior with S0, S1, S 2, S3, S5
S6 have different out put behavior with S0, S1, S 2, S3, S5

Input	P State	NS X=0	NS X=1	Output X=0	Output X=1
Reset	S_0	S_1	S_2	0	0
0	S_1	S_3	S_4	0	0
1	S_2	S_5	S_6	0	0
00	S_3	S_0	S_0	0	0
01	S_4	S_0	S_0	1	0
10	S_5	S_0	S_0	0	0
11	S_6	S_0	S_0	1	0

Implication Chart Method



Starting Implication Chart

Fill other Sij Based on Next State Transaction

S0 : S1, S2 when X=0, 1

S1 : S3, S4 when X=0,1

Input	P State	NS X=0	NS X=1	Output X=0	Output X=1
Reset	S_0	S_1	S_2	0	0
0	S_1	S_3	S_4	0	0
1	S_2	S_5	S_6	0	0
00	S_3	S_0	S_0	0	0
01	S_4	S_0	S_0	1	0
10	S_5	S_0	S_0	0	0
11	S_6	S_0	S_0	1	0

Implication Chart Method

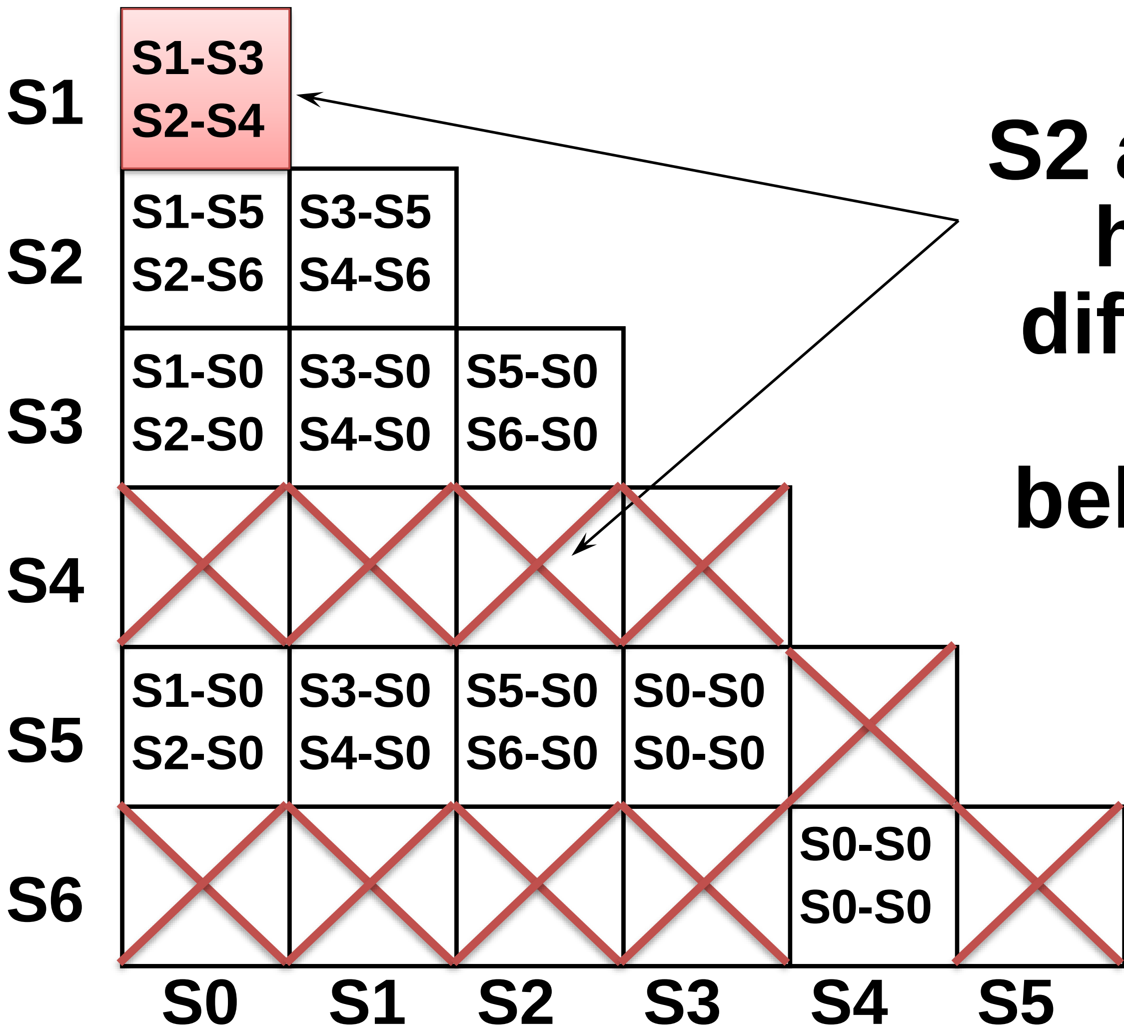
Fill others Sij Based on Next State Transaction

	S0	S1	S2	S3	S4	S5
S1	S1-S3 S2-S4					
S2	S1-S5 S2-S6	S3-S5 S4-S6				
S3	S1-S0 S2-S0	S3-S0 S4-S0	S5-S0 S6-S0			
S4						
S5	S1-S0 S2-S0	S3-S0 S4-S0	S5-S0 S6-S0	S0-S0 S0-S0		
S6					S0-S0 S0-S0	

Starting Implication Chart

Input	P State	NS X=0	NS X=1	Output X=0	Output X=1
Reset	S_0	S_1	S_2	0	0
0	S_1	S_3	S_4	0	0
1	S_2	S_5	S_6	0	0
00	S_3	S_0	S_0	0	0
01	S_4	S_0	S_0	1	0
10	S_5	S_0	S_0	0	0
11	S_6	S_0	S_0	1	0

Implication Chart Method

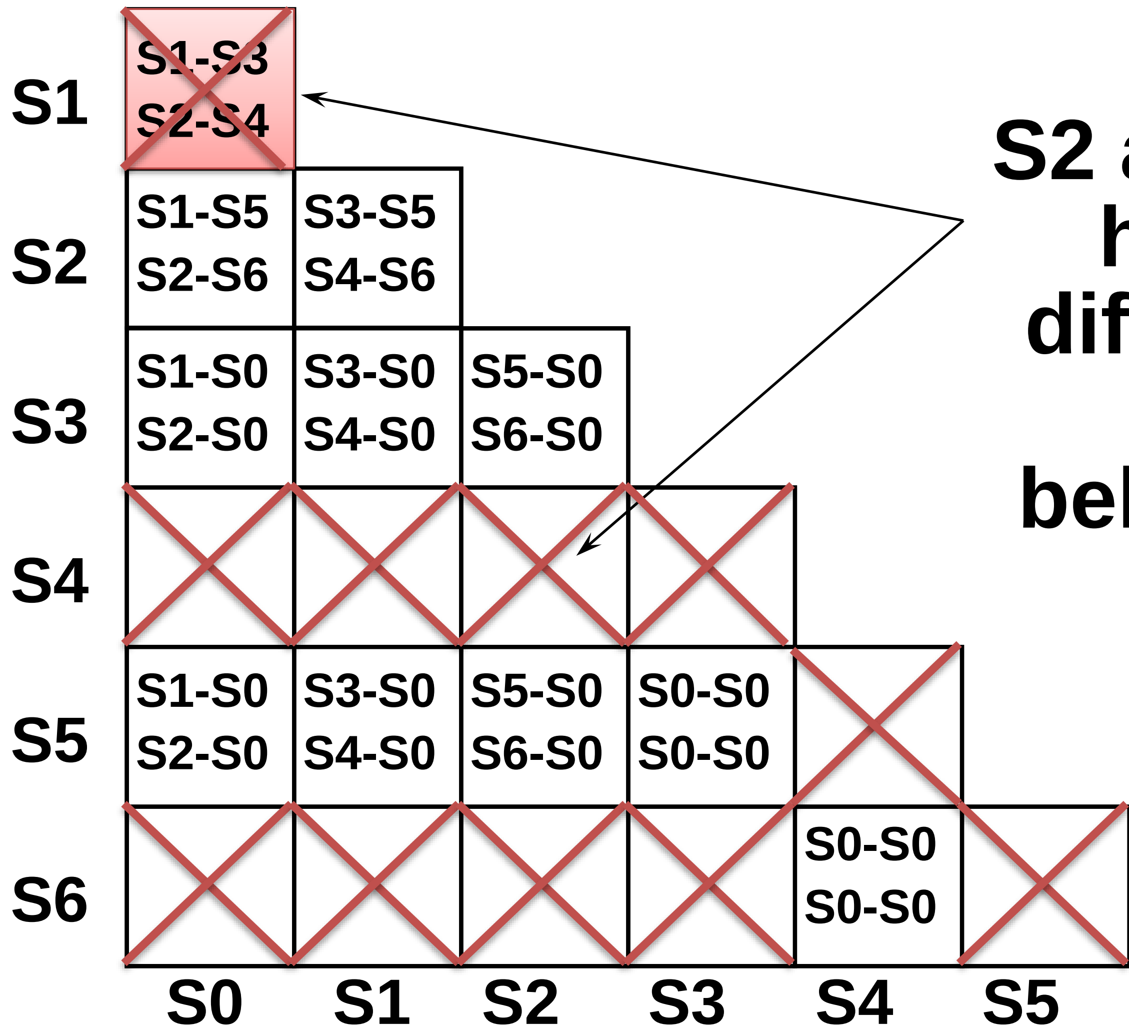


S2 and S4 have different I/O behavior

This implies that S1 and S0 cannot be combined

Marked with Cross

Implication Chart Method

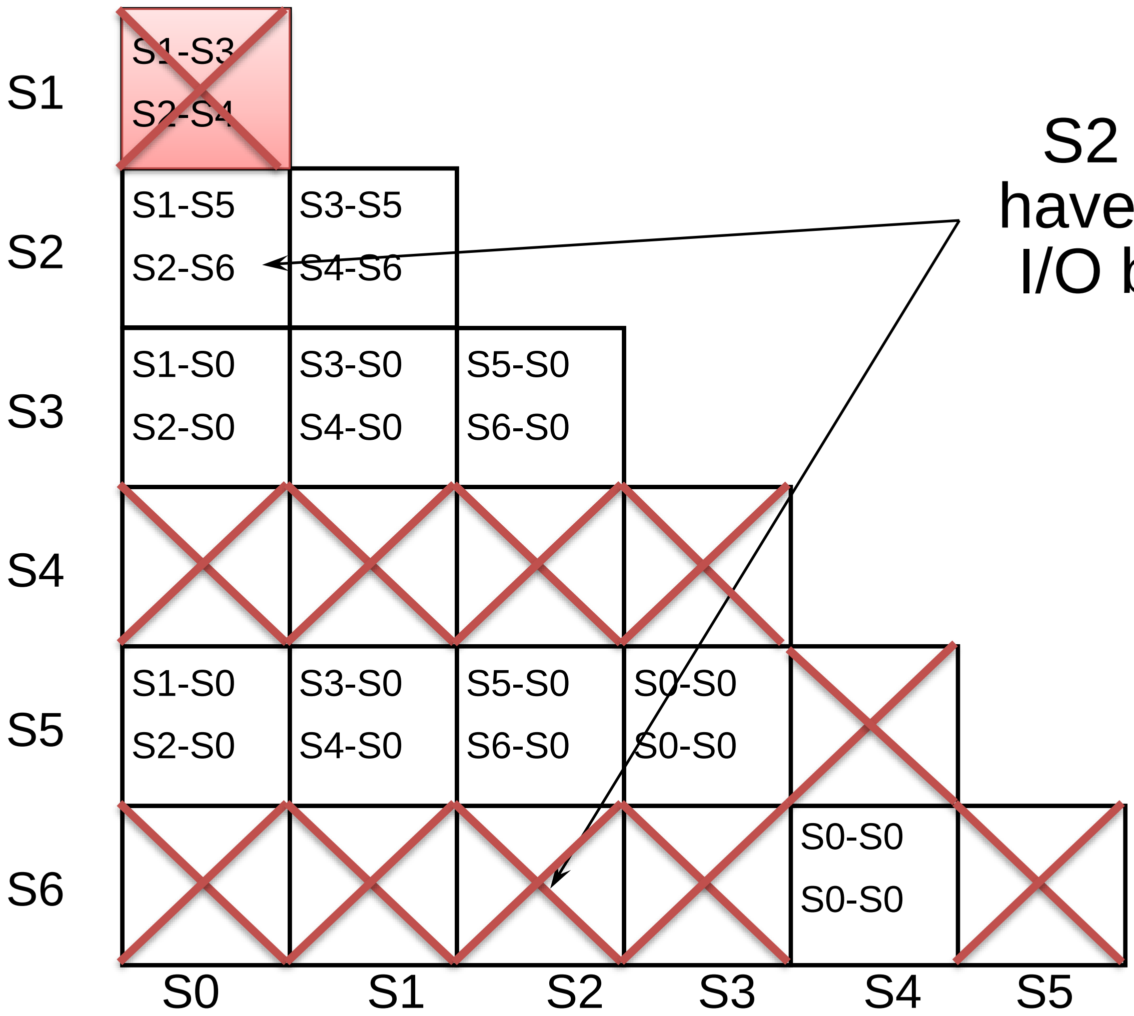


S2 and S4 have different I/O behavior

This implies that S1 and S0 cannot be combined

Marked with Cross

Implication Chart Method

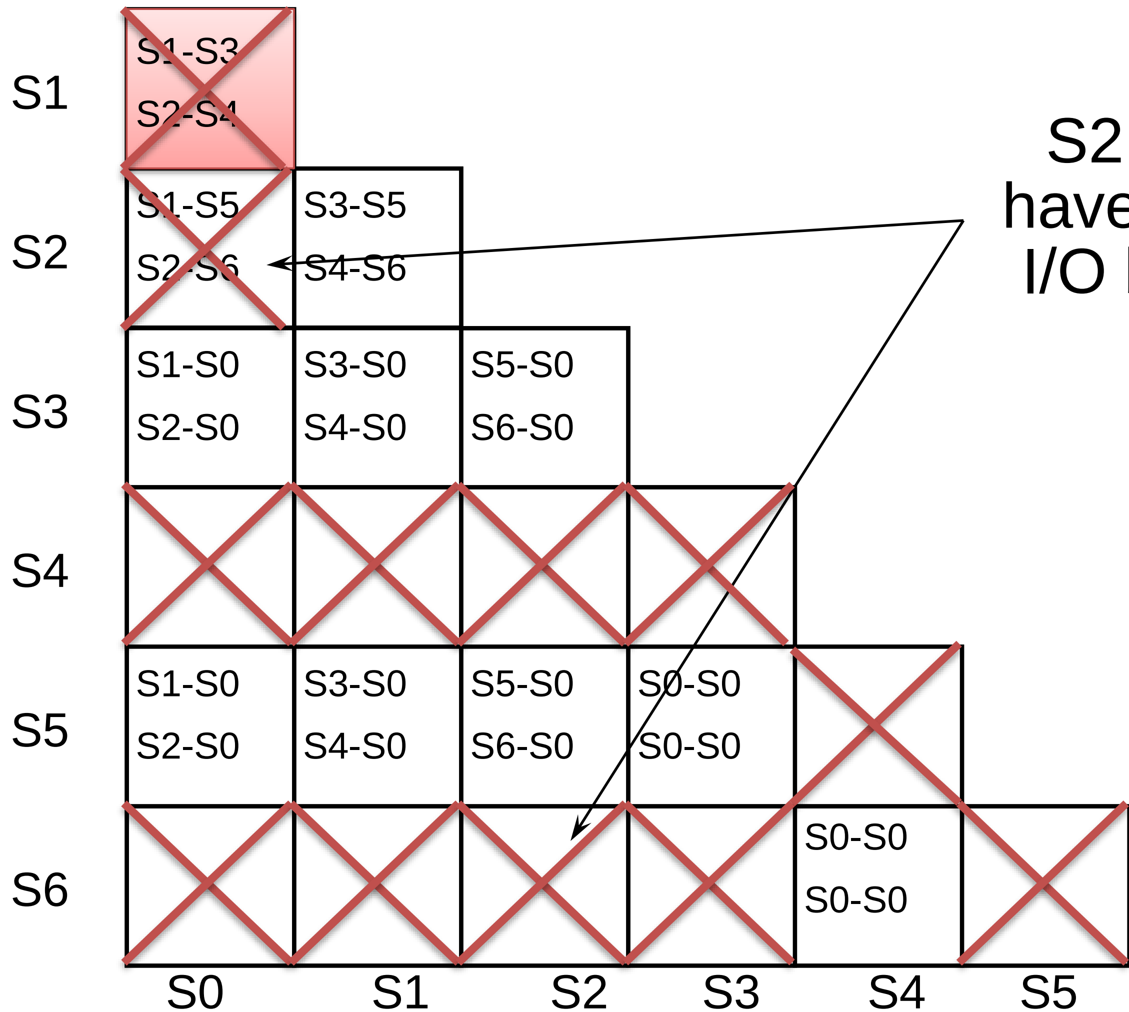


S2 and S6 have different I/O behavior

This implies that S1 and S0 cannot be combined

Marked with Cross

Implication Chart Method

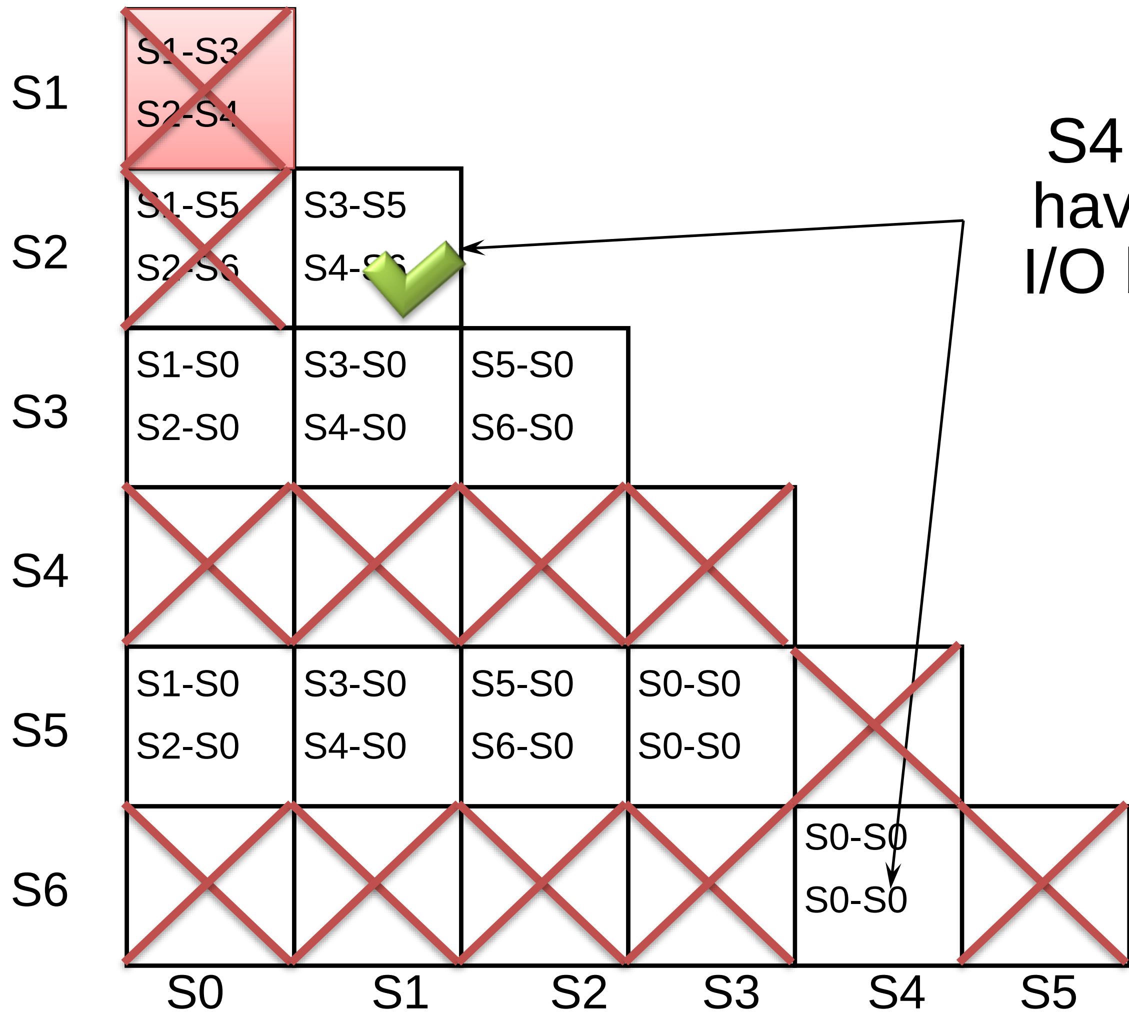


S2 and S6 have different I/O behavior

This implies that S2 and S0 cannot be combined

Marked with Cross

Implication Chart Method



S4 and S6 have **Same** I/O behavior

Implication Chart Method

	S0	S1	S2	S3	S4	S5
S1	~~S1-S3~~ ~~S2-S4~~					
S2	~~S1-S5~~ ~~S2-S6~~	S3-S5 S4-S6				
S3	~~S1-S0~~ ~~S2-S0~~	S3-S0 S4-S0	S5-S0 S6-S0			
S4						
S5	S1-S0 S2-S0	S3-S0 S4-S0	S5-S0 S6-S0	S0-S0 S0-S0		
S6					S0-S0 S0-S0	

S1 S0 is already crossed

So S3 S0

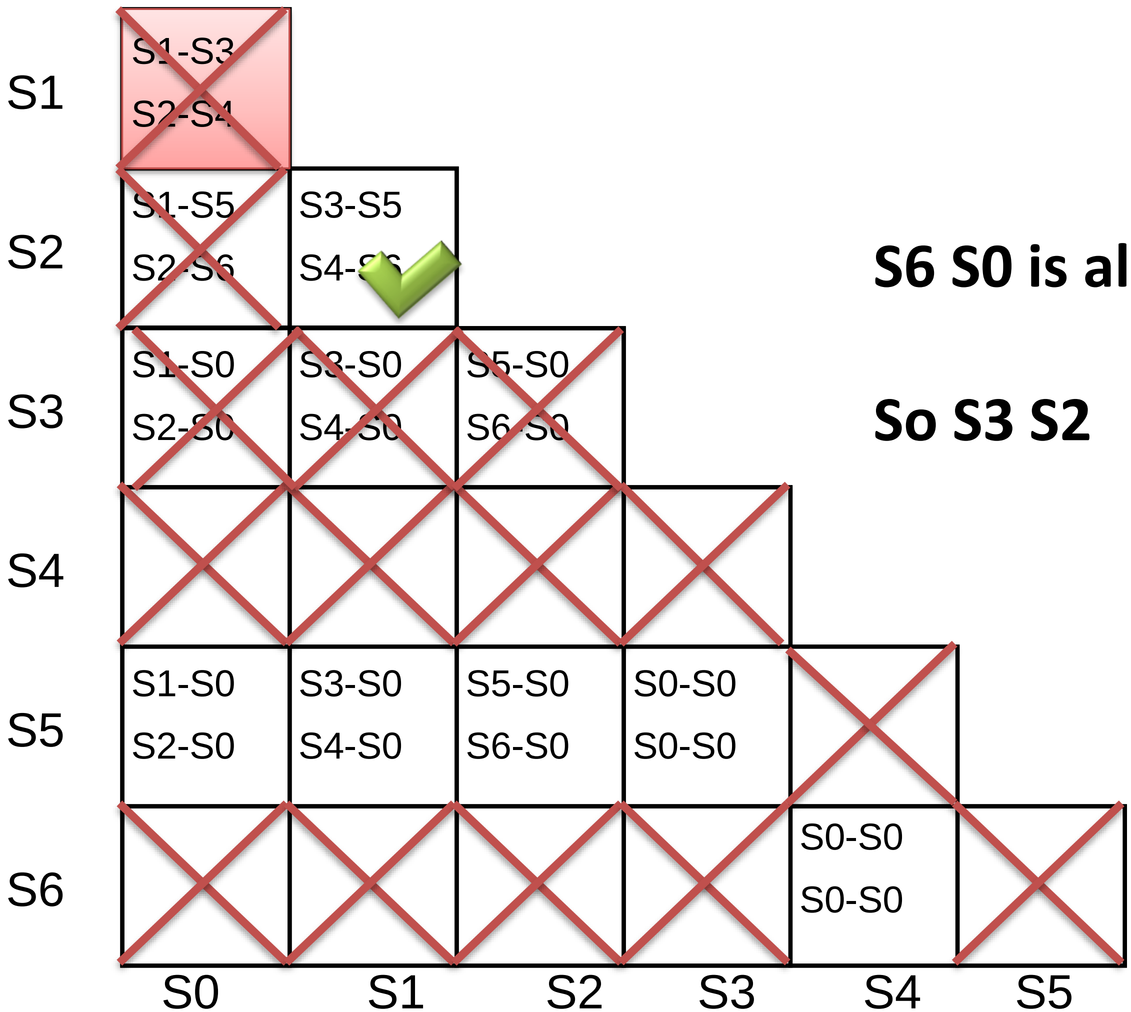
Implication Chart Method

S1	~~S1-S3~~ ~~S2-S4~~					
S2	~~S1-S5~~ ~~S2-S6~~	S3-S5 S4-S6				
S3	~~S1-S0~~ ~~S2-S0~~	~~S3-S0~~ ~~S4-S0~~	S5-S0 S6-S0			
S4						
S5	S1-S0 S2-S0	S3-S0 S4-S0	S5-S0 S6-S0	S0-S0 S0-S0		
S6					S0-S0 S0-S0	
	S0	S1	S2	S3	S4	S5

S4 S0 is already crossed

So S3 S1

Implication Chart Method



S6 S0 is already crossed

So S3 S2

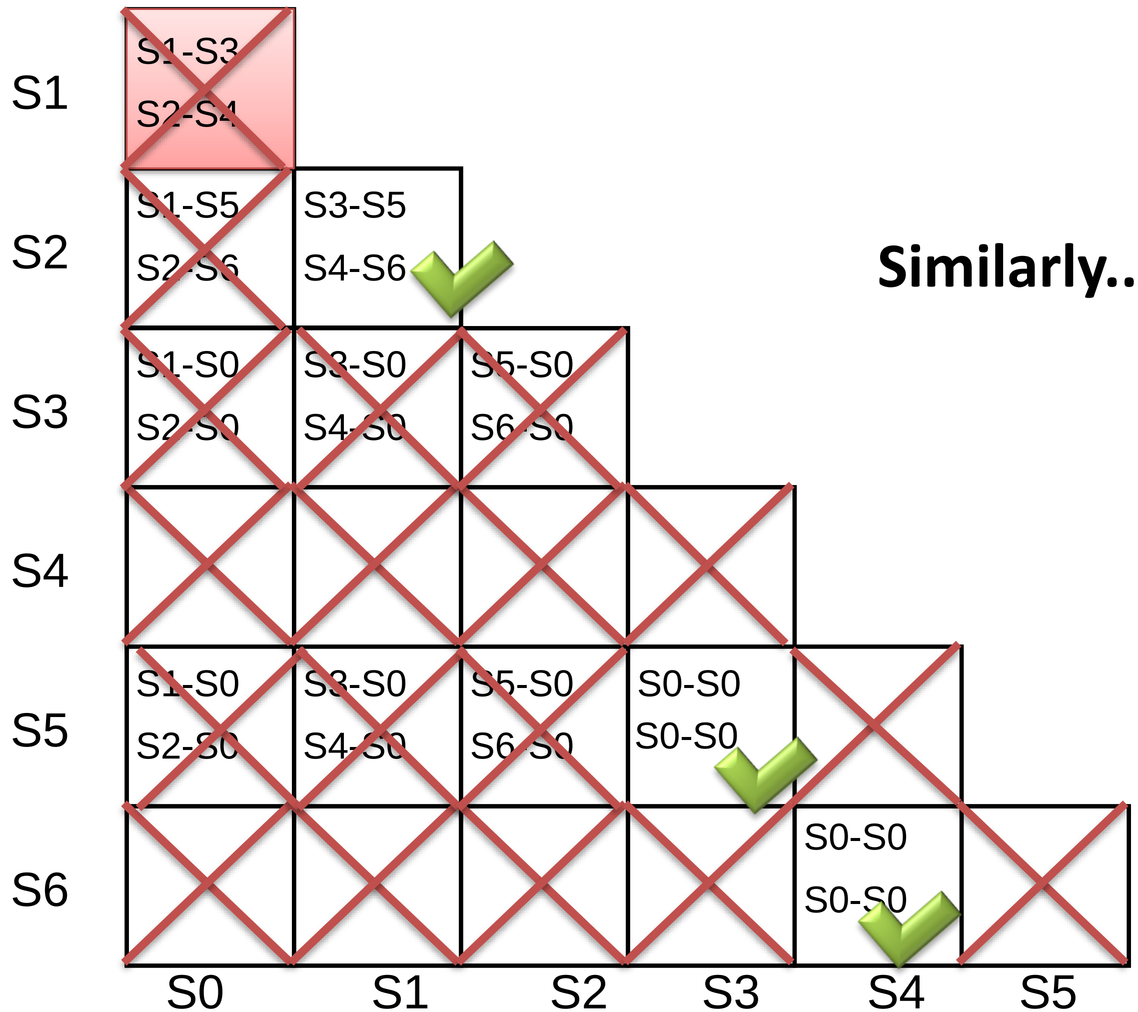
Implication Chart Method

	S0	S1	S2	S3	S4	S5
S1	~~S1-S3~~ ~~S2-S4~~					
S2	~~S1-S5~~ ~~S2-S6~~	S3-S5 S4-S6				
S3	~~S1-S0~~ ~~S2-S0~~	~~S3-S0~~ ~~S4-S0~~	~~S5-S0~~ ~~S6-S0~~			
S4						
S5	~~S1-S0~~ ~~S2-S0~~	S3-S0 S4-S0	S5-S0 S6-S0	S0-S0 S0-S0		
S6					S0-S0 S0-S0	

S2 S0 is already crossed

So S5 S0

Implication Chart Method



Similarly..

Second Pass Adds No New Information

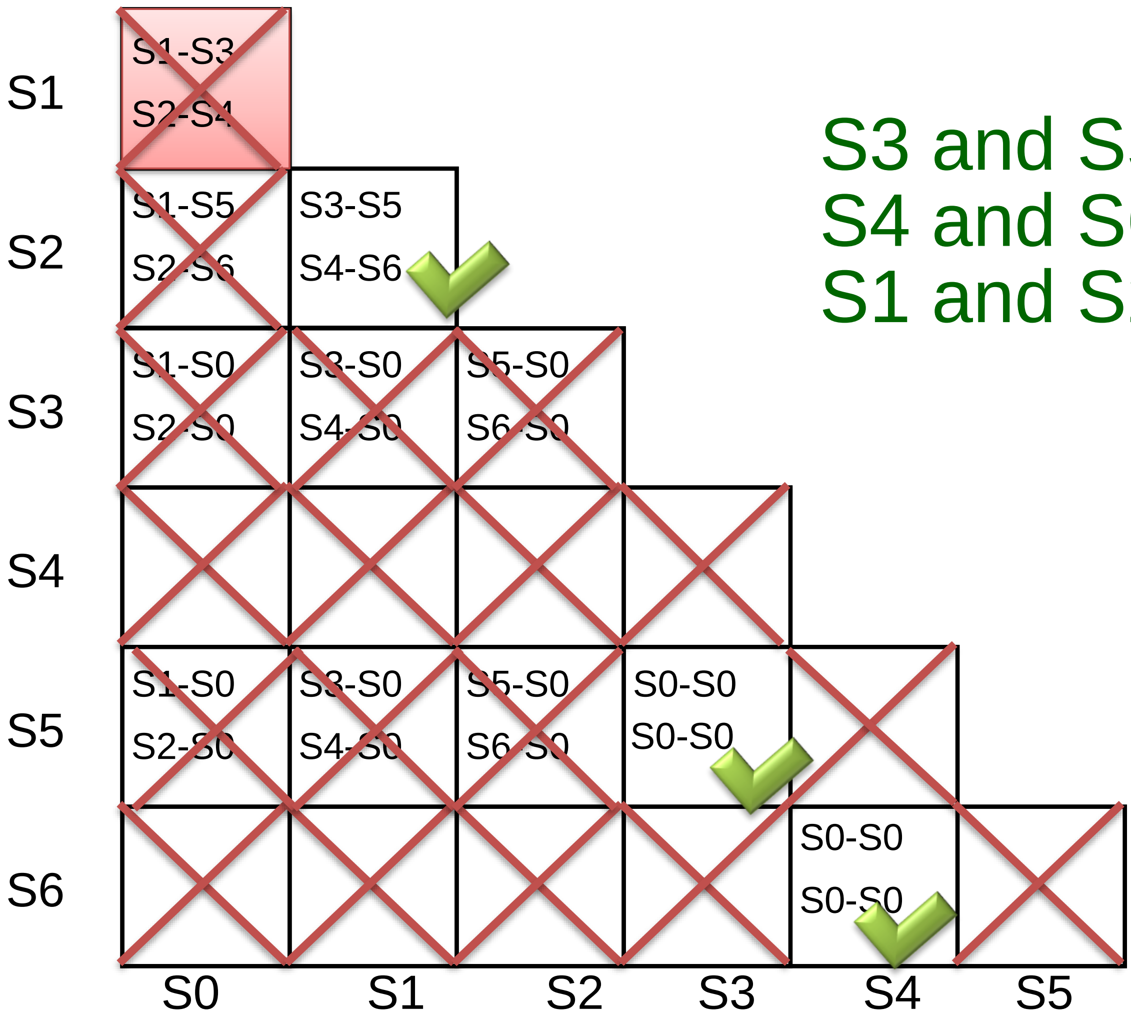
Implication Chart Method

	S0	S1	S2	S3	S4	S5
S1	~~S1-S3 S2-S4~~					
S2	~~S1-S5 S2-S6~~	S3-S5 S4-S6				
S3	~~S1-S0 S2-S0~~	~~S3-S0 S4-S0~~	~~S5-S0 S6-S0~~			
S4						
S5	~~S1-S0 S2-S0~~	~~S3-S0 S4-S0~~	~~S5-S0 S6-S0~~	S0-S0 S0-S0		
S6					S0-S0 S0-S0	

Similarly..

Second Pass Adds No New Information

Implication Chart Method



S3 and S5 are equivalent
S4 and S6 are equivalent
S1 and S2 are equivalent

Second Pass Adds No New Information

Final : Reduce State Tabe

- Reduces State table

Input	P State	NS X=0	NS X=1	Output X=0	Output X=1
Reset	S_0	S_1	S_2	0	0
0	S_1	S_3	S_4	0	0
1	S_2	S_5	S_6	0	0
00	S_3	S_0	S_0	0	0
01	S_4	S_0	S_0	1	0
10	S_5	S_0	S_0	0	0
11	S_6	S_0	S_0	1	0

Input	PS	NS		Output	
		X=0	X=1	X=0	X=1
Reset	S0	S1’	S1’	0	0
0 or 1	S1’	S3’	S4’	0	0
00 or 10	S3’	S0	S0	0	0
01 or 11	S4’	S0	S0	1	0