Optimized State Diagrams

Last updated 7/18/23

Optimized State Diagrams

- Two formal approaches for optimization
  - Successive Partitions
  - Implication Chart

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State
A	0	B
A	1	C
B	0	D
B	1	F
C	0	F
C	1	E
D	0	B
D	1	G
E	0	F
E	1	C
F	0	E
F	1	D
G	0	F
G	1	G

State	Output
A	1
B	1
C	0
D	1
E	0
F	0
G	0

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Partitions	Next States	Action
P0	ABCDEFG 1101000	

Identify states and outputs

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Partitions	Next States	Action
P0	ABCDEFG 1101000	Separate ABD, CEFG

Partition into sets with same outputs

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Partitions	Next States		Action
P0	ABCDEFG 1101000		Separate ABD, CEFG
P1 In = 0 In = 1	ABD BDB CFG	CEFG FFEF ECDG	

Identify next states based on current state and inputs

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Partitions	Next States		Action
P0	ABCDEFG 1101000		Separate ABD, CEFG
P1 In = 0 In = 1	ABD BDB CFG	CEFG FFEF ECDG	Separate CEG and F

Identify any groups of next states that are not part of an existing partition

Separate those groups by current state

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Partitions	Next States			Action
P0	ABCDEFG 1101000			Separate ABD, CEFG
P1 In = 0 In = 1	ABD BDB CFG	CEFG FFEF ECDG		Separate CEG and F
P2 In = 0 In = 1	ABD BDB CFG	CEG FFF ECG	F E D	

Identify next states based on current state and inputs

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Partitions	Next States			Action
P0	ABCDEFG 1101000			Separate ABD, CEFG
P1 In = 0 In = 1	ABD BDB CFG	CEFG FFEF ECDG		Separate CEG and F
P2 In = 0 In = 1	ABD BDB CFG	CEG FFF ECG	F E D	Separate AD and B

Identify any groups of next states that are not part of an existing partition

Separate those groups by current state

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Partitions	Next States				Action
P0	ABCDEFG 1101000				Separate ABD, CEFG
P1 In = 0 In = 1	ABD BDB CFG		CEFG FFEF ECDG		Separate CEG and F
P2 In = 0 In = 1	ABD BDB CFG		CEG FFF ECG	F E D	Separate AD and B
P2 In = 0 In = 1	AD BB CG	B D F	CEG FFF ECG	F E D	

Identify next states based on current state and inputs

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Partitions	Next States				Action
P0	ABCDEFG 1101000				Separate ABD, CEFG
P1 In = 0 In = 1					eparate EG and F
P2 In = 0 In = 1	BDB CFG		FFF ECG	E D	Separate AD and B
P2 In = 0 In = 1	AD BB CG	B D F	CEG FFF ECG	F E D	No more reduction

Identify any groups of next states that are not part of an existing partition

Separate those groups by current state

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Partitions	Next States				Action
P0	ABCDEFG 1101000				Separate ABD, CEFG
P1 In = 0 In = 1	ABD BDB CFG		CEFG FFEF ECDG		Separate CEG and F
P2 In = 0 In = 1	ABD BDB CFG		CEG FFF ECG	F E D	Separate AD and B
P2 In = 0 In = 1	AD BB CG	B D F	CEG FFF ECG	F E D	No more reduction
Pfinal	AD	B	CEG	F	

Optimized State Diagrams

- Redundant / Equivalent States
  - Successive Partitions

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

7 states → 4 states

Partitions	Next States				Action
P0	ABCDEFG 1101000				Separate ABD, CEFG
P1 In = 0 In = 1	ABD BDB CFG		CEFG FFEF ECDG		Separate CEG and F
P2 In = 0 In = 1	ABD BDB CFG		CEG FFF ECG	F E D	Separate AD and B
P2 In = 0 In = 1	AD BB CG	B D F	CEG FFF ECG	F E D	No more reduction
Pfinal	AD	B	CEG	F	

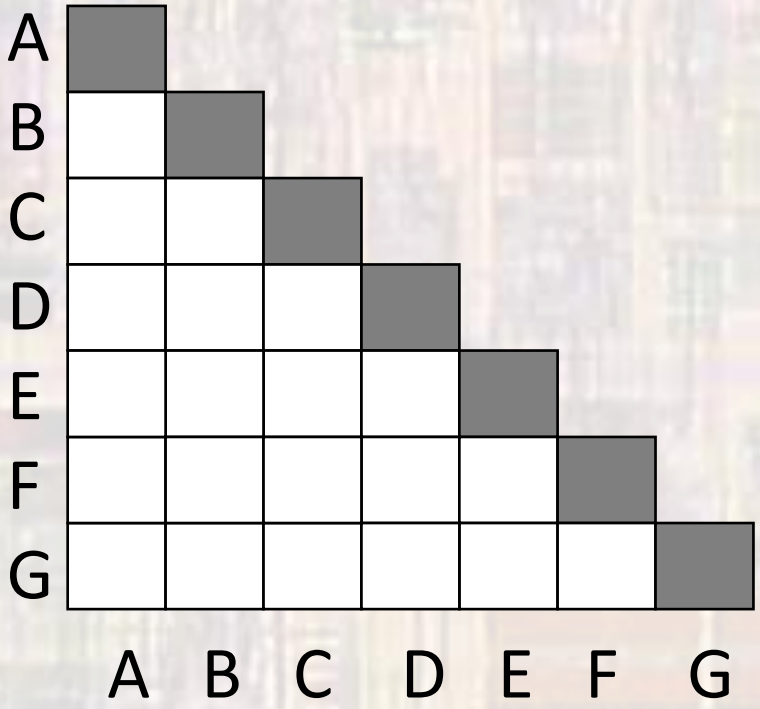
State	Input	Next State		Output
AD	0	B		1
AD	1	CEG		1
B	0	AD		1
B	1	F		1
CEG	0	F		0
CEG	1	CEG		0
F	0	CEG		0
F	1	AD		0

State	Input	Next State		Output
P	0	Q		1
P	1	R		1
Q	0	P		1
Q	1	S		1
R	0	S		0
R	1	R		0
S	0	R		0
S	1	P		0

Optimized State Diagrams

- Redundant / Equivalent States
  - Implication Chart

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

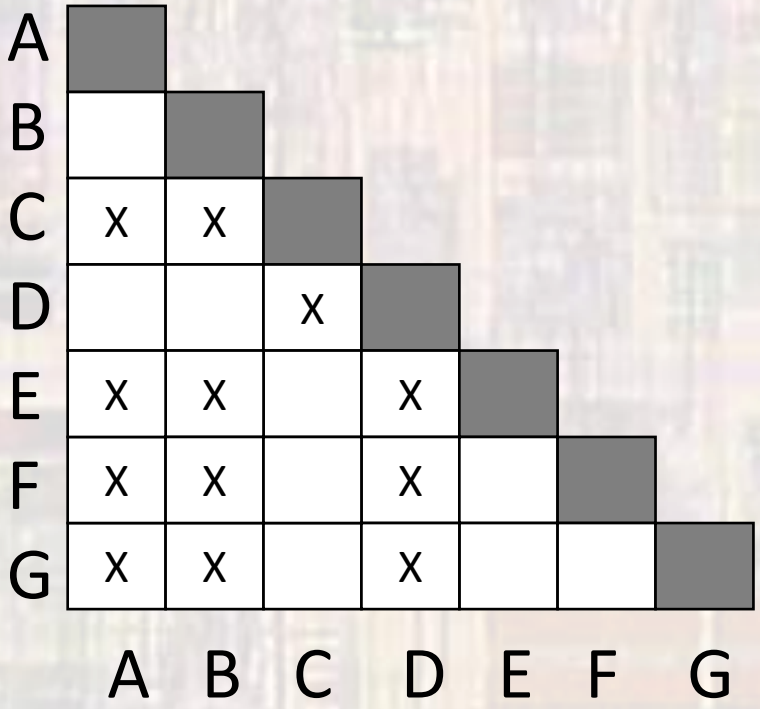


Optimized State Diagrams

- Redundant / Equivalent States
  - Implication Chart

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

compare pairs of states
If outputs are different X out box



Optimized State Diagrams

- Redundant / Equivalent States
  - Implication Chart

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

write in the implicants to all empty boxes

The two states would be the same IFF the implicants are the same

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

States A and B would be the same ONLY if B and D are the same and C and F are the same

Optimized State Diagrams

- Redundant / Equivalent States
  - Implication Chart

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Traverse the structure and X out any boxes whose implicants are already X'd out
This indicates the implicant is not true

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

Optimized State Diagrams

- Redundant / Equivalent States
  - Implication Chart

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Traverse the structure and X out any boxes whose implicants are already X'd out
This indicates the implicant is not true

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

Optimized State Diagrams

- Redundant / Equivalent States
  - Implication Chart

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

Traverse the structure and X out any boxes whose implicants are already X'd out
This indicates the implicant is not true

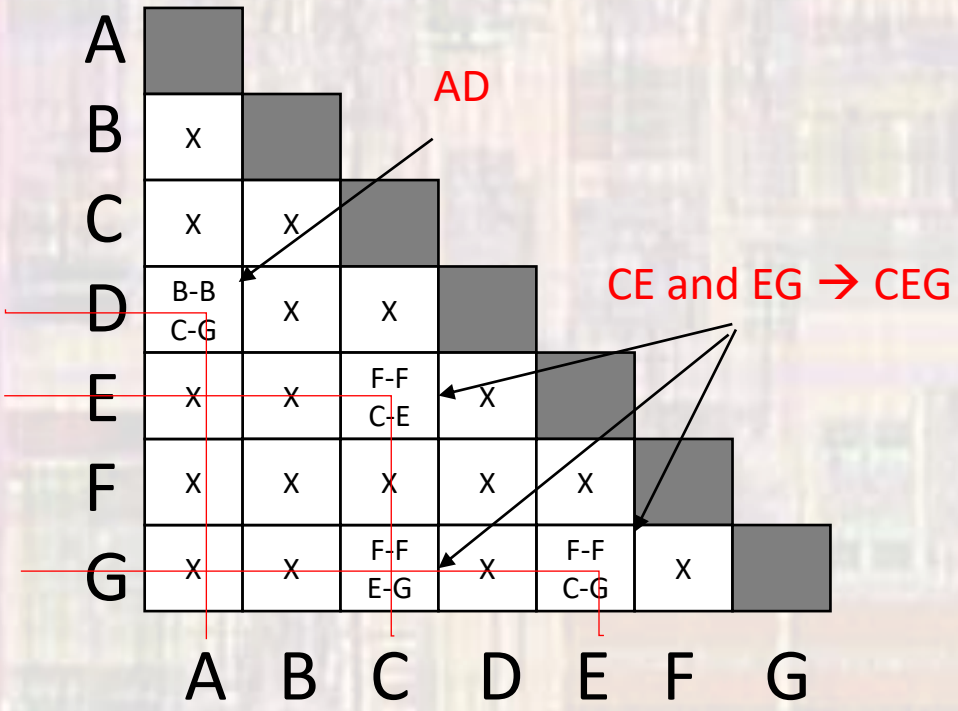
	A	B	C	D	E	F	G
A							
B	X						
C	X	X					
D	B-B C-G	X	X				
E	X	X	F-F C-E	X			
F	X	X	X	X	X		
G	X	X	F-F E-G	X	F-F C-G	X	

Optimized State Diagrams

- Redundant / Equivalent States
  - Implication Chart

Remaining un-X'd boxes indicate equivalent states

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

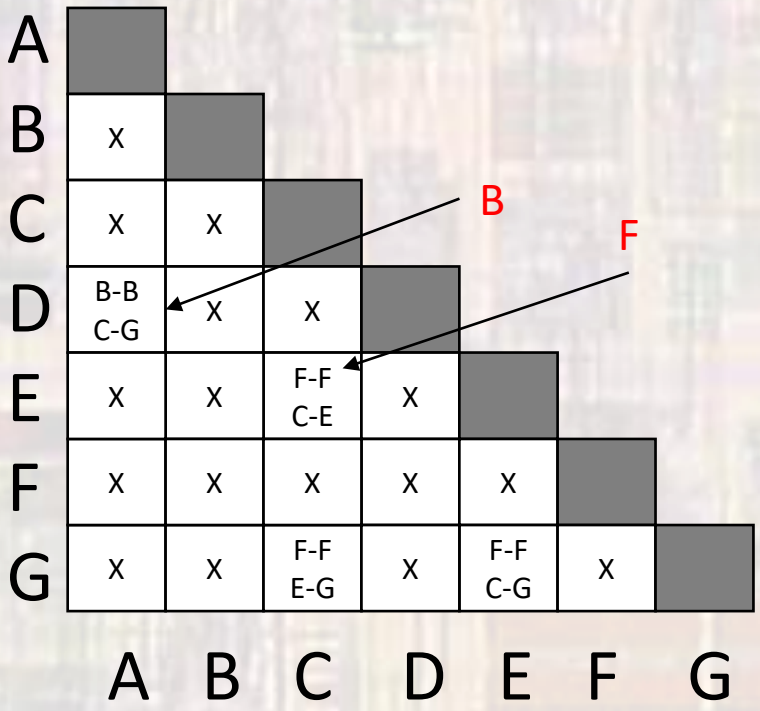


Optimized State Diagrams

- Redundant / Equivalent States
  - Implication Chart

Any duplicate entries indicate independent states

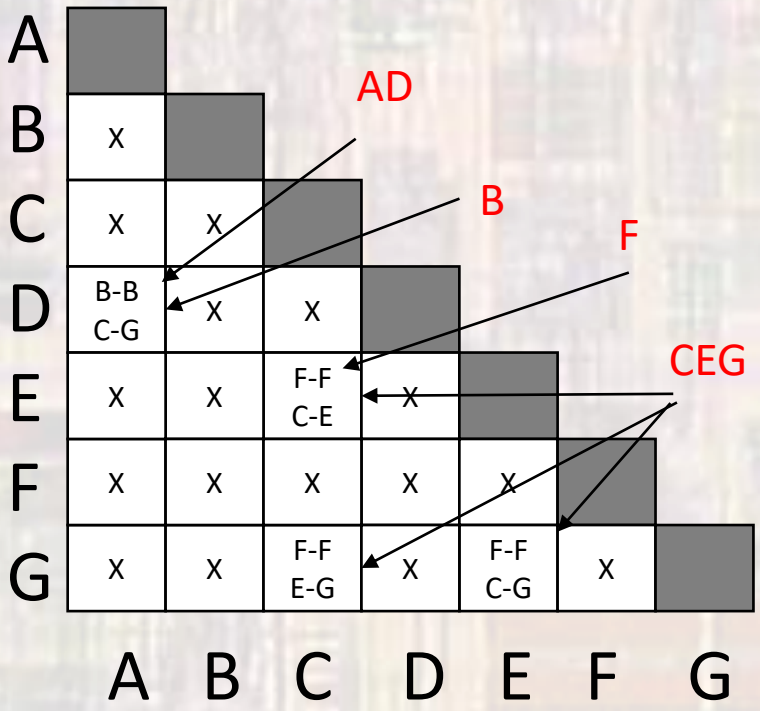
State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0



Optimized State Diagrams

- Redundant / Equivalent States
  - Implication Chart

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

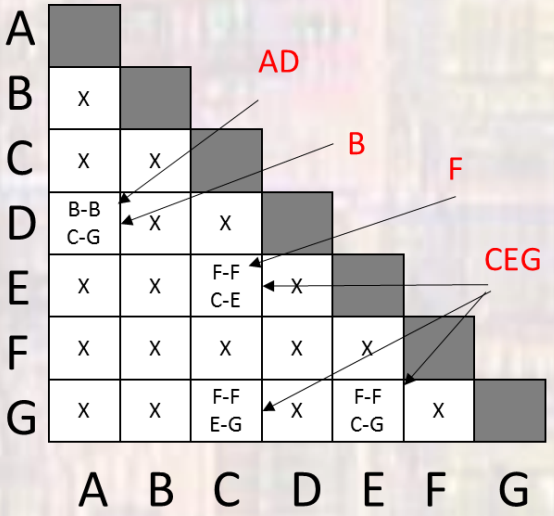


Optimized State Diagrams

- Redundant / Equivalent States
  - Implication Chart

State	Input	Next State		Output
A	0	B		1
A	1	C		1
B	0	D		1
B	1	F		1
C	0	F		0
C	1	E		0
D	0	B		1
D	1	G		1
E	0	F		0
E	1	C		0
F	0	E		0
F	1	D		0
G	0	F		0
G	1	G		0

7 states → 4 states



State	Input	Next State		Output
AD	0	B		1
AD	1	CEG		1
B	0	AD		1
B	1	F		1
CEG	0	F		0
CEG	1	CEG		0
F	0	CEG		0
F	1	AD		0

State	Input	Next State		Output
P	0	Q		1
P	1	R		1
Q	0	P		1
Q	1	S		1
R	0	S		0
R	1	R		0
S	0	R		0
S	1	P		0

Optimized FSMs

- Design Process Circuit Design
  1) Identify the states – collectively these make a state variable
  2) Identify the Inputs and Outputs
  3) Assign values for each input/output (encoding)
  4) Create a state transition diagram / table
  5) Optimize the state transition table
  6) Assign values for the state variable for each state (encoding)
  7) Create truth tables for the combinational logic blocks in the machine model: next state, output
  8) Minimize the next state and output equations using K-maps or Boolean Algebra techniques
  9) Draw the circuit schematic
  10) Verify the solution
  11) Build the physical circuit
  12) Test the physical circuit to ensure correct operation
- Design Process HDL
  1) Identify the states – collectively these make a state variable
  2) Identify the Inputs and Outputs
  3) Create a state transition diagram / table
  4) Optimize the state transition table
  5) Create the HDL to match the state transition table
  6) Choose an encoding scheme (or let the tool decide)
  7) Synthesize the design
  8) Verify the solution
  9) Build the physical circuit
  10) Test the physical circuit to ensure correct operation