Design Verification – *An Introduction*

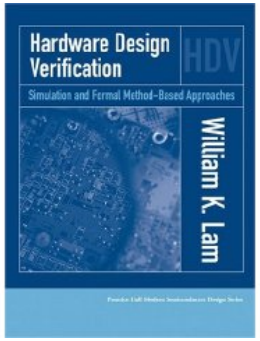
Testing & Verification

Dept. of Computer Science & Engg, IIT Kharagpur

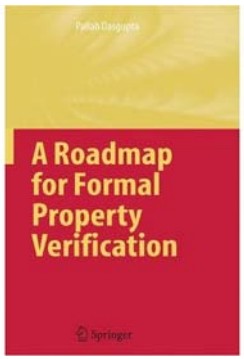
Pallab Dasgupta

Professor, Dept. of Computer Science & Engg.,
Professor-in-charge, AVLSI Design Lab,
Indian Institute of Technology Kharagpur

Main References



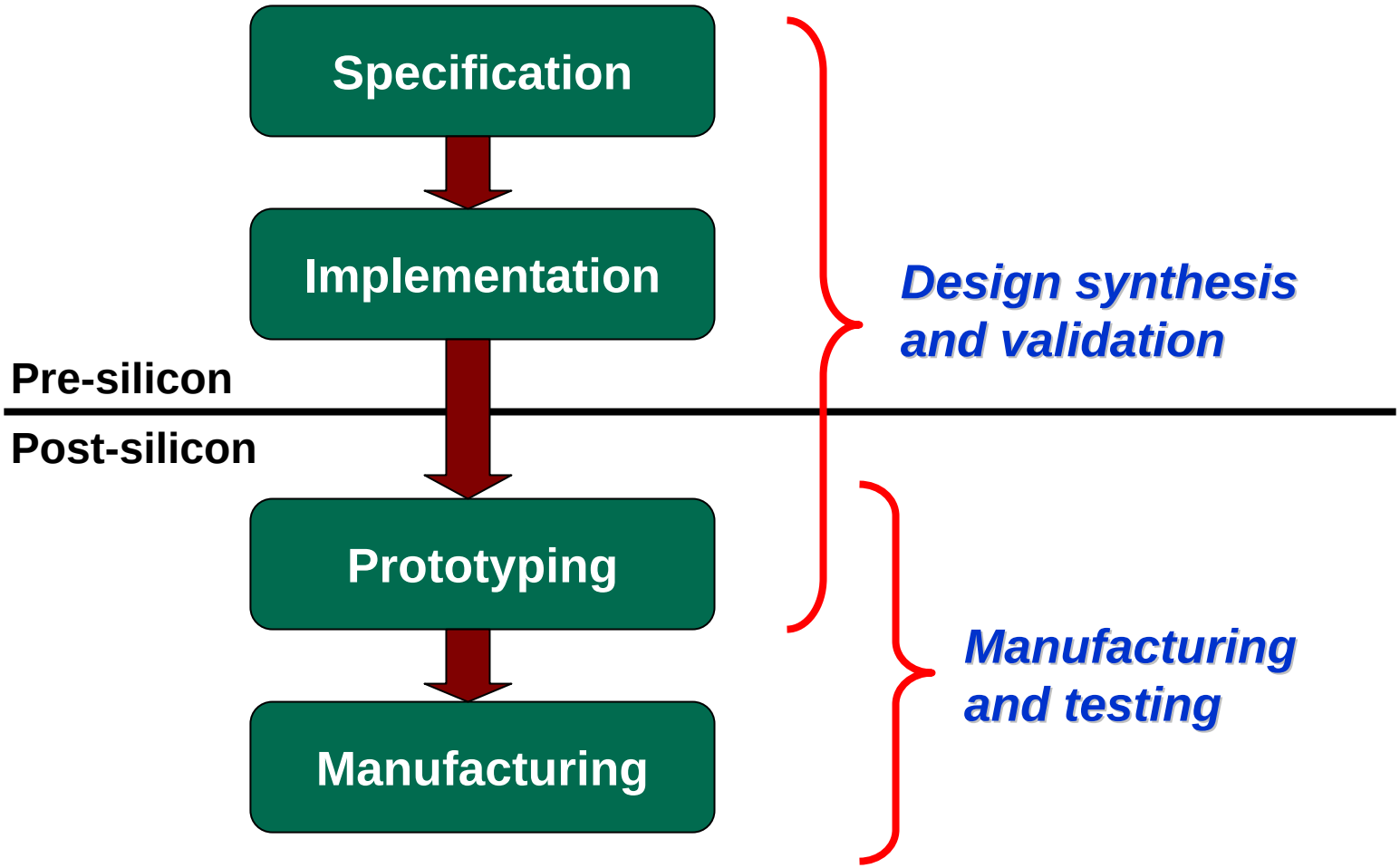
Hardware Design Verification: *Simulation and Formal Method-Based Approaches*
William K Lam
Prentice Hall Modern Semiconductor Design Series



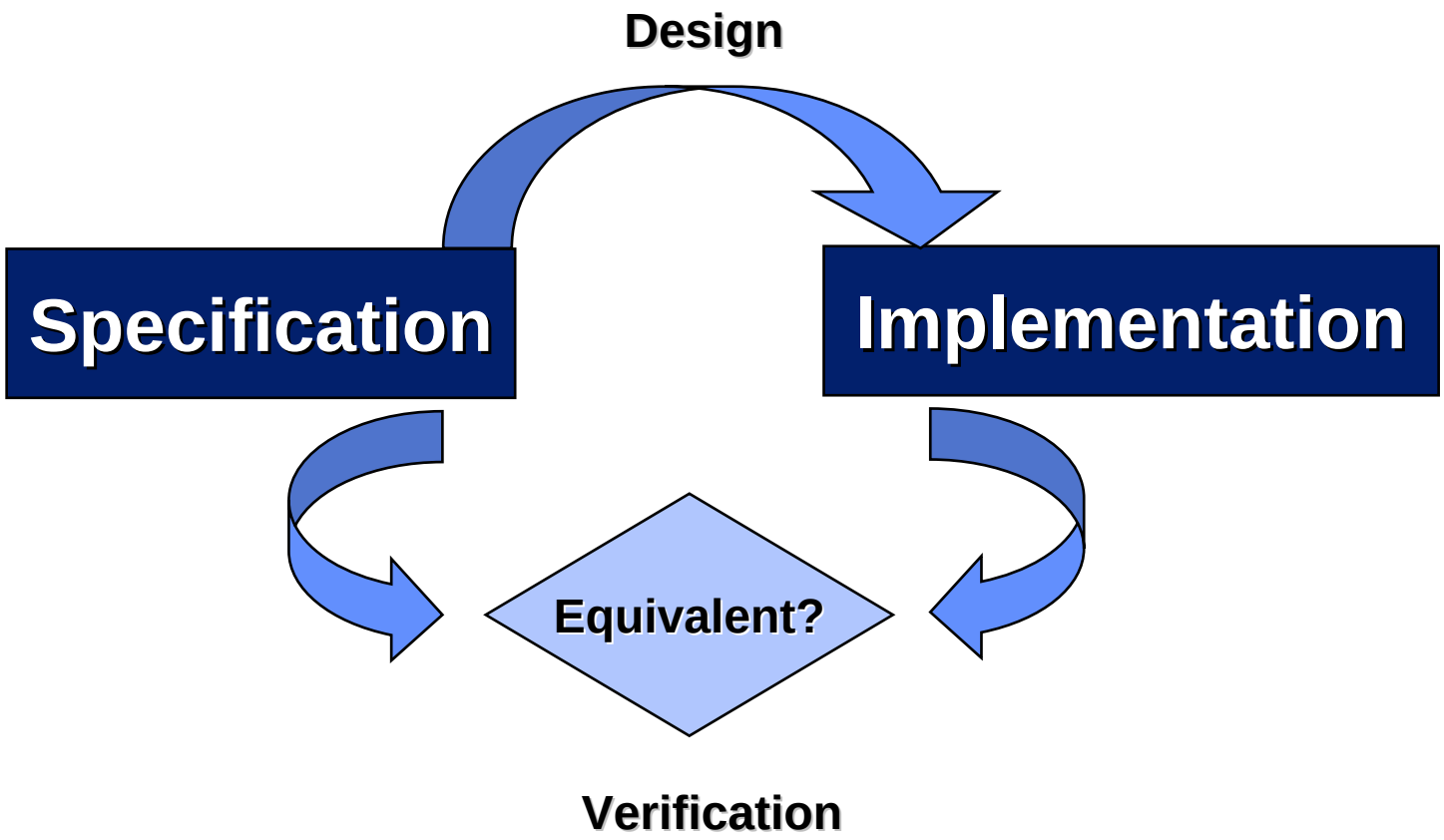
A Roadmap for Formal Property Verification
Pallab Dasgupts
Springer

Course Web: http://www.facweb.iitkgp.ernet.in/~pallab and follow link to courses

Design, Validation and Testing



Design and Verification



Design Challenges

Suppose we have to design a pacemaker

Resource optimization → **Reduce Cost**

Test automation and Design-for-Test → **Improve Testability**

Algorithms for better yield → **Improve Reliability**

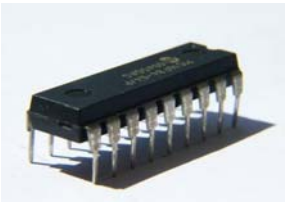
Improve Verifiability → ***Verification reliability and coverage***

Minimize Power → ***Power optimization during synthesis***

Minimize Area → ***Area optimization during synthesis***

Minimize Delay → ***Delay optimization during synthesis***

Digital Design: Abstraction Levels



Exponential growth in circuit size (Moore's Law)

Formalisms introduced at the Entry-Level

Register Transfer Level

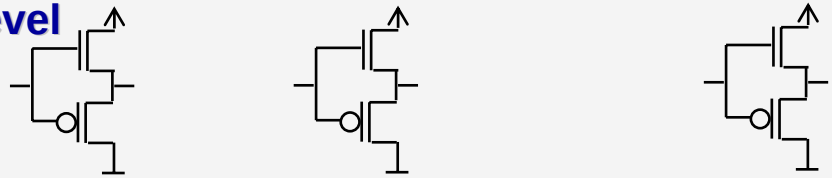
```
always @( posedge clk )
begin
 if (!rst) begin a1 <= a2;
      a2 <=  ~a1; end;
end
```

Restricted semantics of Programming Languages, Communicating Concurrent State Machines (CSM)

Gate Level

Boolean Logic
Finite State Machines

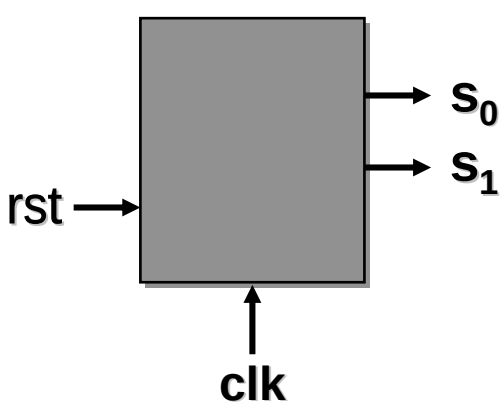
Transistor Level

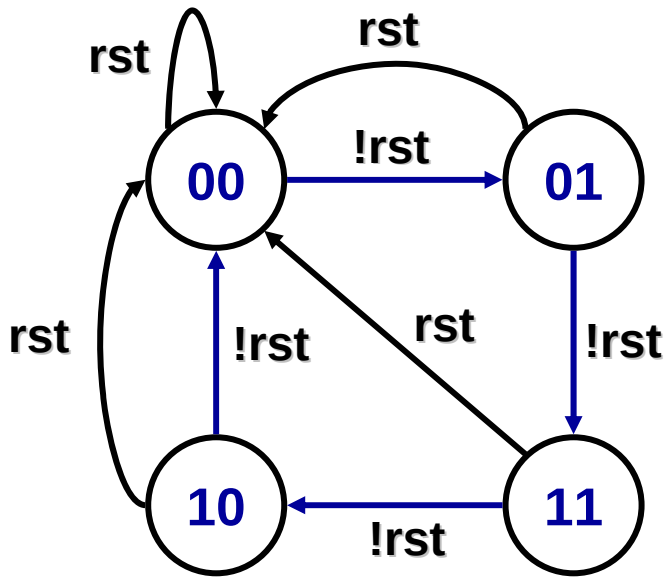


Schematic

Design Example: 2-bit Gray Counter

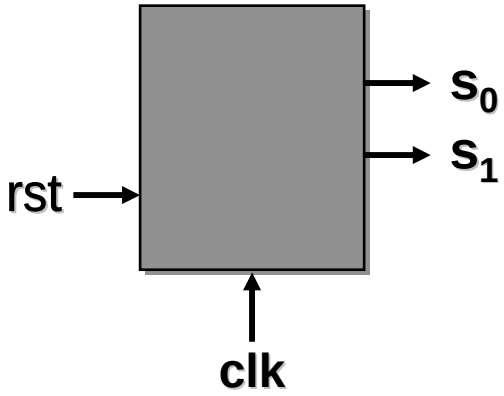
Gray Counter: Successive values should differ only in one bit. Reset signal resets the counter to zero.

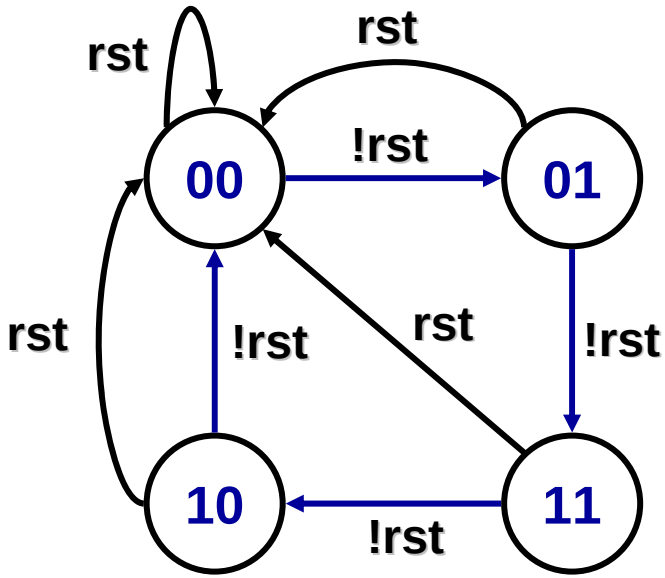




State m/c Representation

Design Example: 2-bit Gray Counter





State m/c Representation

State Transition Table

(s_0s_1)	rst	(n_0n_1)
00	0	01
00	1	00
01	0	11
01	1	00
10	0	00
10	1	00
11	0	10
11	1	00

Design Example: 2-bit Gray Counter

State Transition Table

$(s_0 s_1)$	rst	$(n_0 n_1)$
00	0	01
00	1	00
01	0	11
01	1	00
10	0	00
10	1	00
11	0	10
11	1	00

State Transition Functions:

$$n_0 = s_0' s_1 r' + s_0 s_1 r'$$
$$n_1 = s_0' s_1' r' + s_0' s_1 r'$$

After Logic Minimization:

$$n_0 = s_1 r'$$
$$n_1 = s_0' r'$$

Design Example: 2-bit Gray Counter

State Transition Functions:

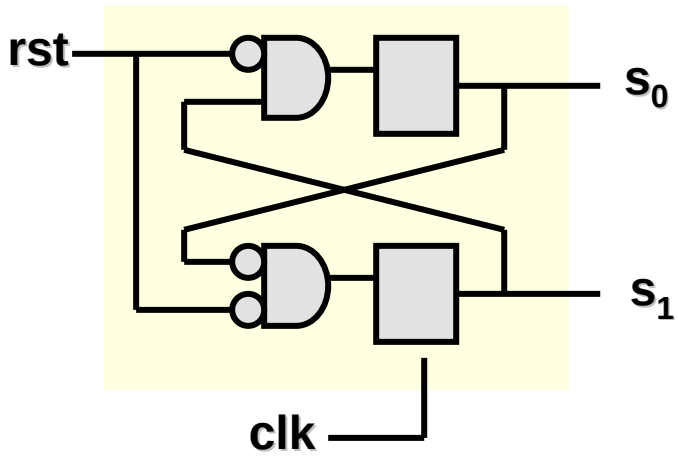
$$n_0 = s_1 r'$$
$$n_1 = s_0' r'$$

Verilog Code (RTL):

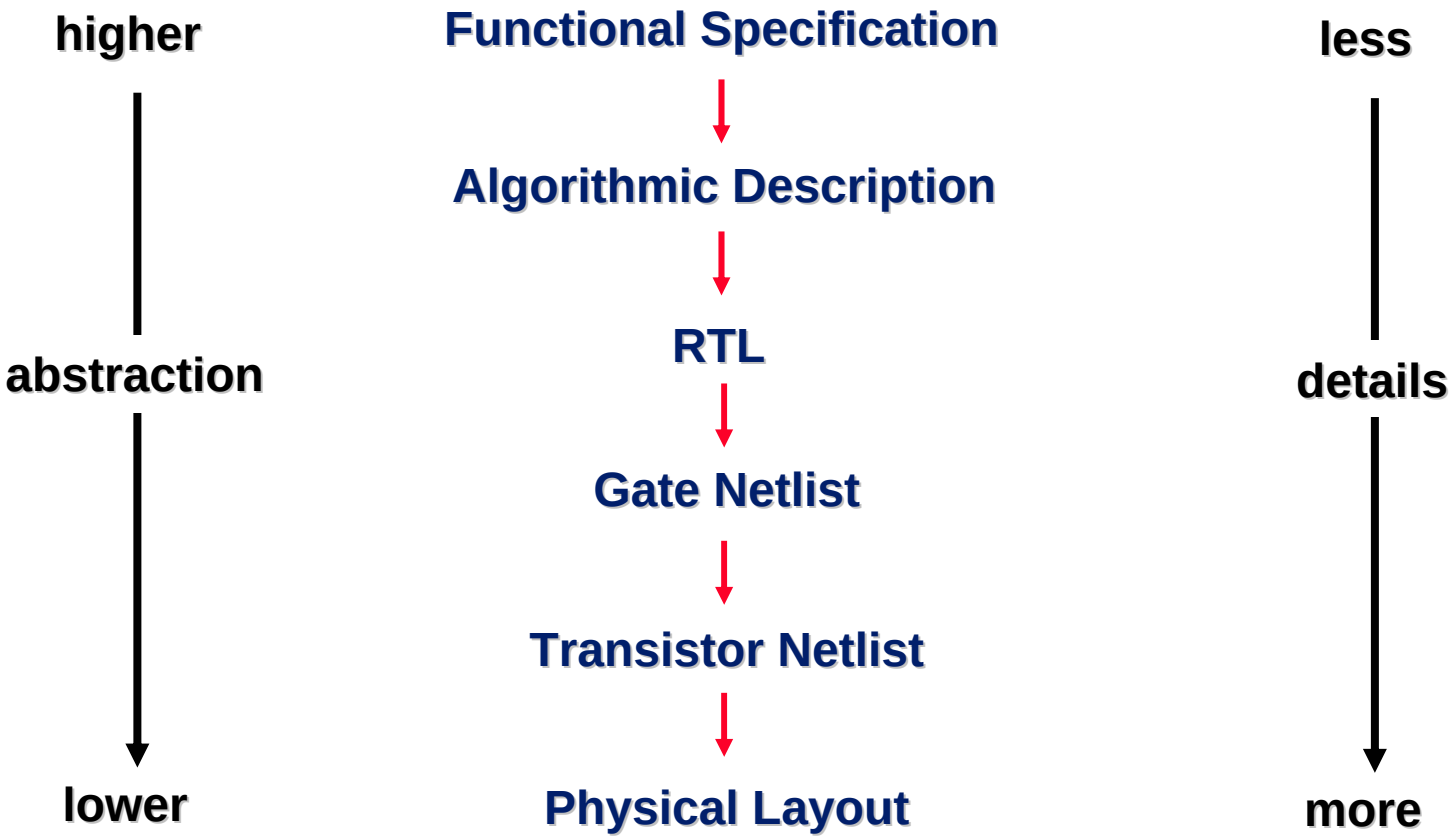
```
module GrayCounter(s0, s1, rst)
input rst;
reg s0, s1;

always @ (posedge clk)
begin
    s0 <= s1 & ~rst;
    s1 <= ~s0 & ~rst;
end
endmodule
```

Synthesis



Abstractions in Design Flow



Design and Verification