EE 709:Testing & Verification of VLSI Circuits

Introduction

Virendra Singh

Associate Professor

Computer Architecture and Dependable Systems Lab

Dept. of Electrical Engineering

Indian Institute of Technology Bombay, Mumbai

viren@ee.iitb.ac.in

Course Outline

- VLSI Testing
- VLSI System Design Verification
- Post Silicon Debug
- Fault Diagnosis

Course Schedule

Class Hours

- Monday (10:30 am to 11:30 am) – VLSI Testing
- Tuesday (11:30 am to 12:30 pm) – Design Verification
- Thursday (8:30 am to 9:30 am) – Debug/Diagnosis/Test/Verification

Office Hours

- Thursday (3:00 pm to 4:00 pm)

Course Evaluation

- Mid Term Exam (15%)
  - Open Book/Notes Exam
- Final Exam (40%)
  - Open Book/Notes Exam
- Assignments (15%)
  - Set of assignments will be given periodically
- Course Projects (20%)
  - 2 projects each (Test and Verification)
  - Group (Max size 2)
- Continuous Evaluations (10%)
  - Bi (Tri) weekly tests (70% best will be counted)

Books

- Essential of Electronic Testing for Digital, Memory, and Mixed Signal VLSI Circuits
  - Michel. L. Bushnell and Vishwani D. Agrawal
  - Springer 2005
- Logic Testing and Design for Testability
  - Hideo Fujiwara
  - MIT Press 1985
- Digital System Testing and Testable Design
  - M. Abramovici, M. Breuer, A. Friedman
  - IEEE Press 1994 (now available in Jayco Publication)
- VLSI Test Principals and Architectures
  - L.W. Wang, C.W. Wu, and W. Xiaoqing
  - Academic Press 2006
- Current Literature (IEEE TC/TCAD/TVLSI, JETTA)

Books

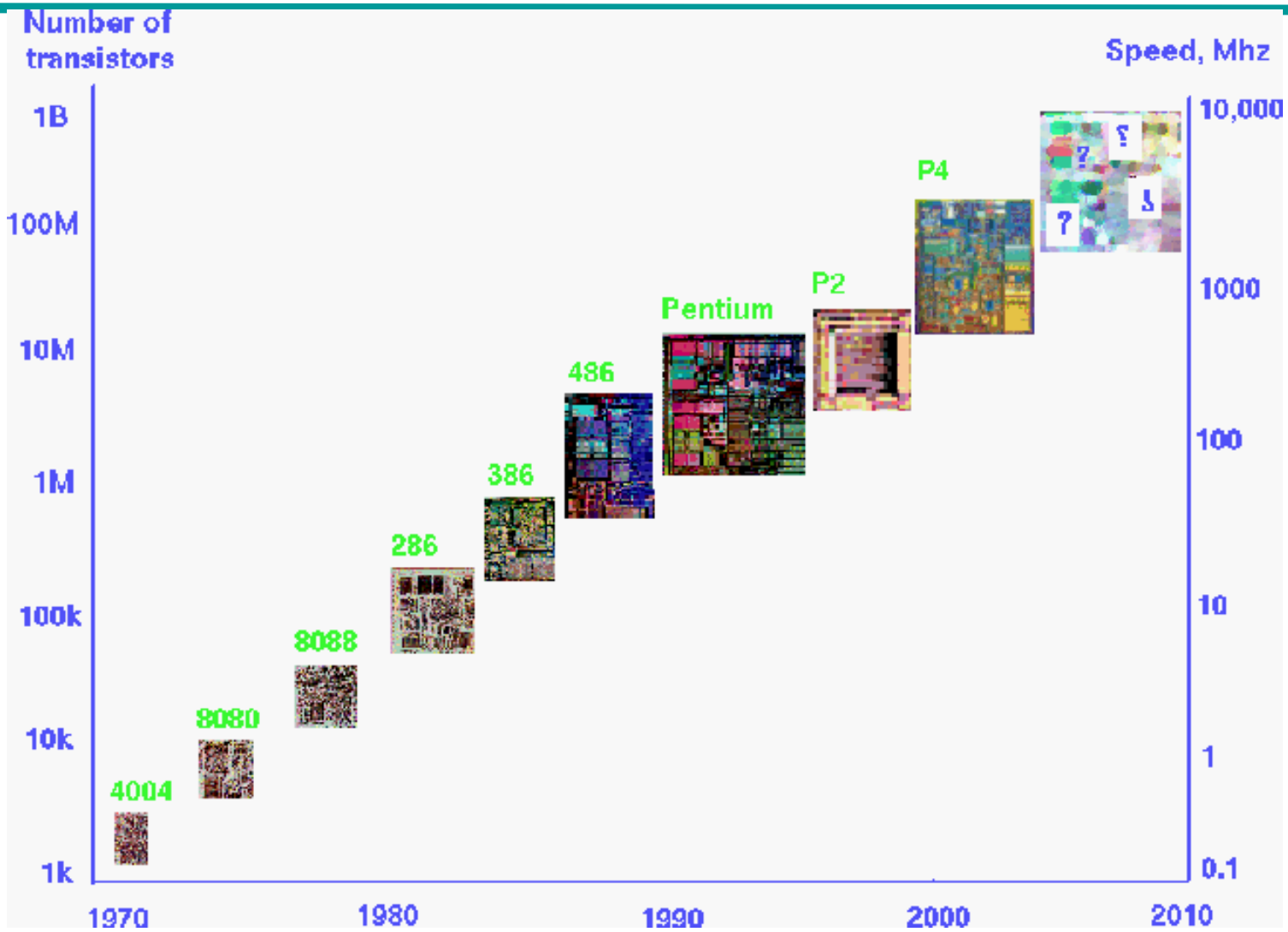
- Logic in Computer Science
  - M. Huth and M.Ryan
  - Cambridge Univ. Press, 2004
- Introduction to Formal Hardware Verification
  - Thomas Kropf
  - Springer
- Hardware Design Verification
  - William K. Lam
  - Prentice Hall

- Current Literature (IEEE TC/TCAD/TVLSI)

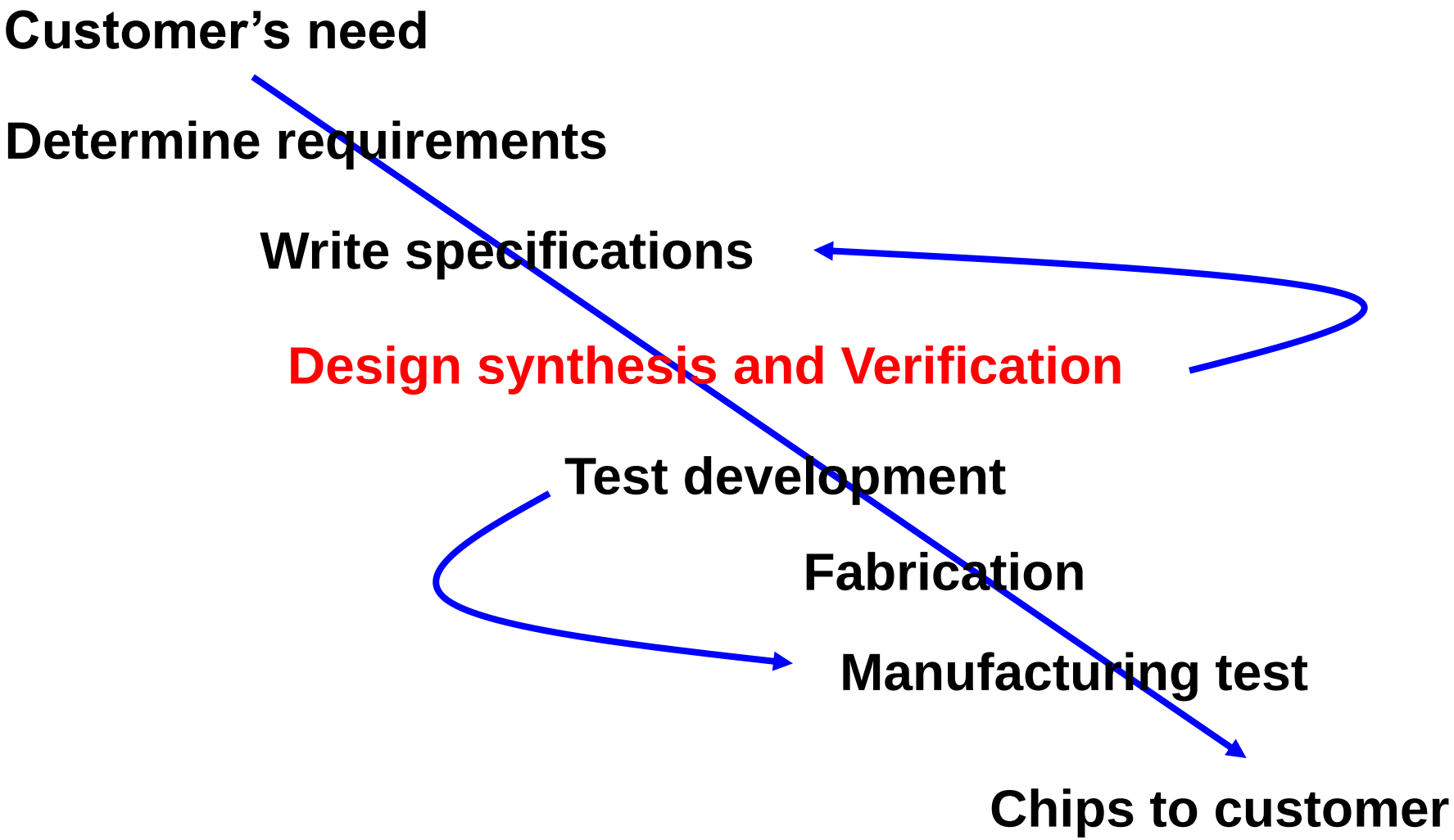
Acknowledgement

- Prof. Hideo Fujiwara, NAIST, Japan
- Prof. Kewal Saluja, Univ. of Wisconsin-Madison
- Prof. Masahiro Fujita, Tokyo University
- Prof. Jacob Abraham, UT Austin
- Prof. Vishwani Agrawal, Auburn Univ.
- Prof. Adit Singh, Auburn Univ.
- Prof. Samiha Mourad, Santa Clara Univ.
- Prof. Michiko Inoue, NAIST
- Prof. Erik Larsson, Linkoping Univ.
- Dr. Subir Roy, Texas Instruments, India
- Dr. Rubin Parekhji, TI, India

Design Complexity



VLSI Realization Process



Definitions

- *Design synthesis:* Given an I/O function, develop a procedure to manufacture a device using known materials and processes.

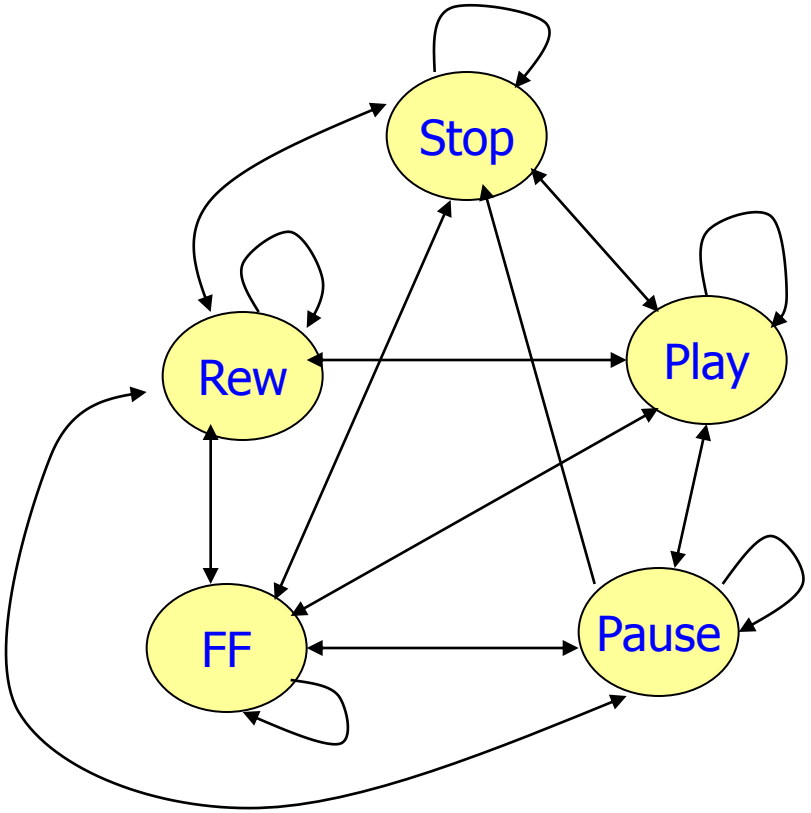
- *Verification:* Predictive analysis to ensure that the synthesized design, when manufactured, will perform the given I/O function.

- *Test:* A manufacturing step that ensures that the physical device, manufactured from the synthesized design, has no manufacturing defect.

Verification of DVD Player

DVD Player

- 6 inputs
  - Nothing is pressed
  - Play, Pause, Stop
  - FF, Rew
- Internal 5 States
  - Stopped, Paused
  - Play at normal speed
  - Forward at 2X speed
  - Rewind at 2X speed



Verification of DVD Player

- Assume 1024 x 786 pixels
- True colour (32 bits)
- Number of discrete states = $(2^{32})^{(1024x786)}$
- Combination of current states to next states $[(2^{32})^{(1024x786)}]^2$
- Pixels are independent
- Bounded number of total states: No. of pixels x number of possible colours x number of internal state machines
- $1024x786x2^{32}x5 = 16,888,498,602,639,360$
- All transitions from current state to next states are considered

Verification of DVD Player

- Number of possible next states: No. of pixels x number of possible colours x number of possible inputs
- $1024 \times 786 \times 2^{32} \times 6 = 20,266,198,323,167,232$
- Possible current state to possible next states are to be verified
- $16,888,498,602,639,360 \times 20,266,198,323,167,232 = 3.4 \times 10^{32}$
- Assume a simulation engine can verify 1,000,000 transitions per second

It needs 10,853,172,947,159,498,300 Years to verify

Importance of Formal Verification

Simulation

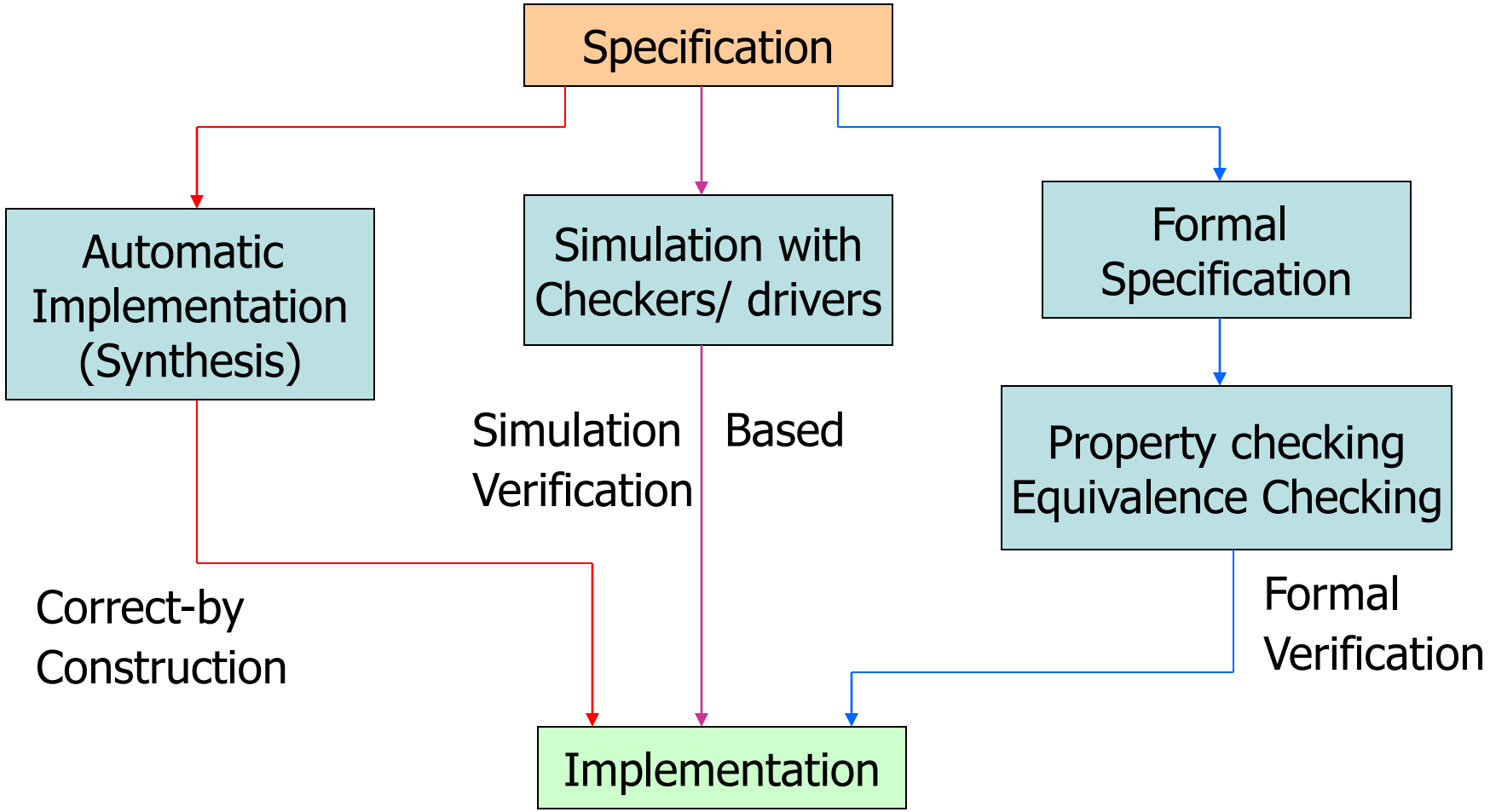
- Can be applied in any design level
- But quality of verification fully depends on Simulation Patterns
  - Corner cases may be missed
  - Random is just random and does not cover corner cases

Emulation

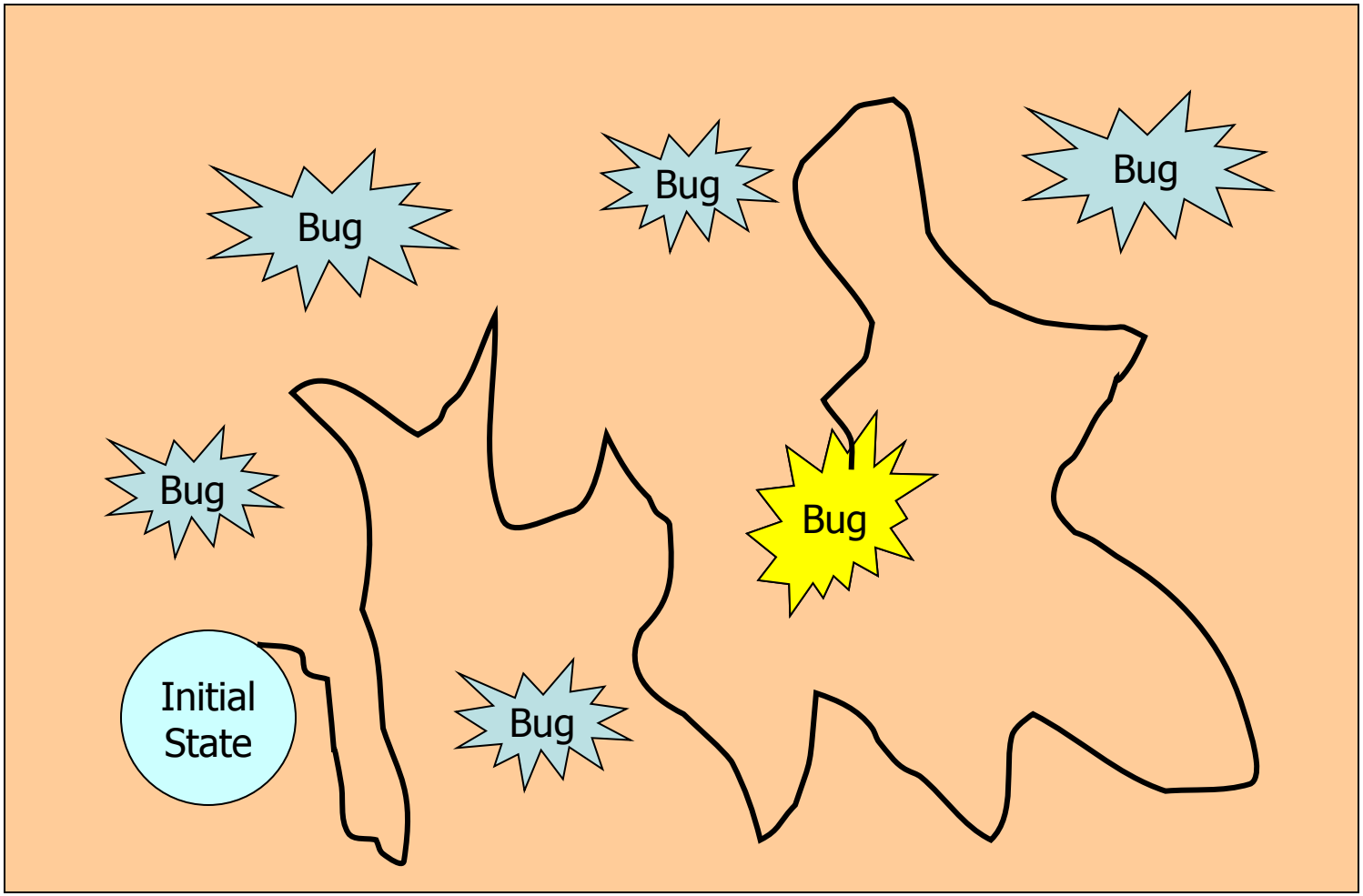
- Implement on FPGA or other programmable device – need lot of preparation
- Still verification quality fully depends on Simulation Patterns – corner cases problem remains

Famous bug: Pentium Floating point bug - $500 m

Design Verification



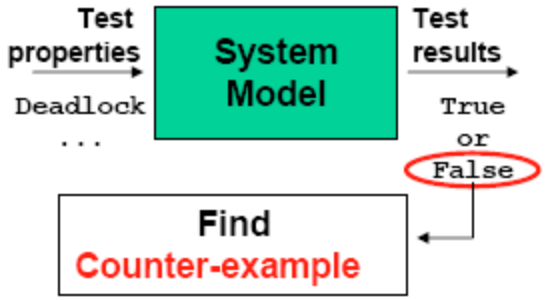
Simulation-Based Verification



Simulation Vs Formal Verification

- Simulation/emulation

- Formal Verification



- Cannot cover all cases
- Corner cases may be missed
- Essential method and good for initially debugging

- ✓ Equivalent to all case simulation
- ✓ No corner case w.r.t given property

Simulation vs Formal Verification

- Program testing can be used to show the presence of the bugs, but never to show the absence!

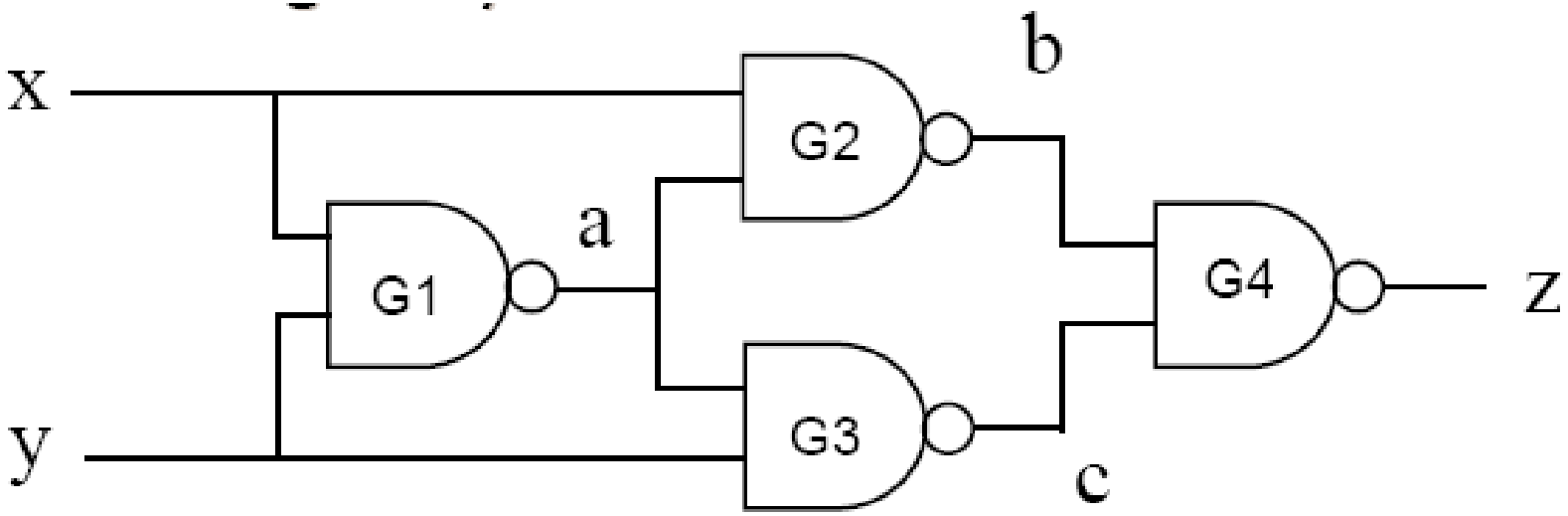
(E.W. Dijkstra)

Simulation Vs Formal Verification

Example:

- Exclusive-OR circuit
- z = (~x & y) + (x & ~y)

x	y	z	~x&y+x&~y
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0



Simulation Vs Formal Verification

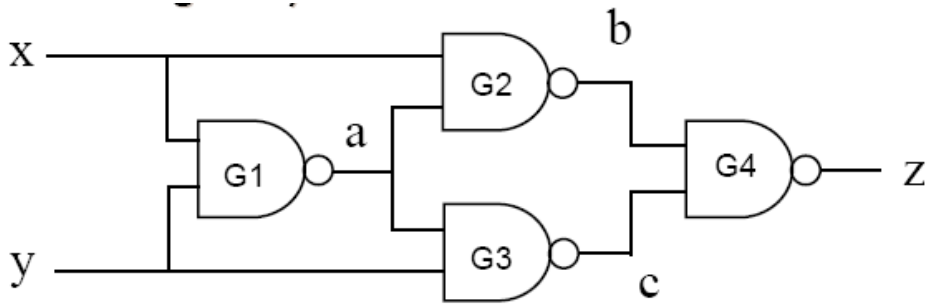
- Transform the formulae for circuit to the one for specification by mathematical reasoning

z = ~b + ~c

b = ~x + ~a

c = ~a + ~y

a = ~x + ~y



z = ~b + ~c

= ~(~x + ~a) + ~(~x + ~y)

= a & x + a & y

= (~x + ~y) & x + (~x + ~y)& y

= x & ~y + ~x & y

- All transformation are based on axioms and theorems
- Mathematical proof of correctness of design

Formal Verification

Techniques

- Deductive Verification (Theorem proving)
  - Uses axioms, rules to prove system correctness
  - Difficult and time consuming
- Model Checking
  - Automatic technique to prove correctness of concurrent systems
  - Symbolic algorithms (using BDD)
- Equivalence Checking
  - Check if two circuits are equivalent

Thank You