Symbolic Trajectory Evaluation

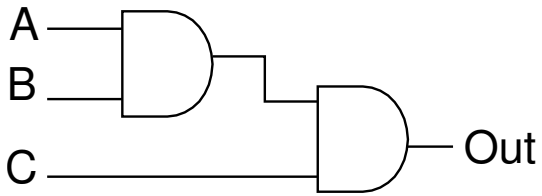
Sérgio Campos, A. Jain

Background — Model Checking

- Verify that the system satisfies some properties
  - Build the transition relation for the system
  - Properties expressed in temporal logic
- Model checking
  - Expressive
  - But... computationally expensive
- Alternative — Symbolic Trajectory Evaluation (STE)
  - Simulation based verification
  - Tradeoff between expressiveness and performance

Logic Simulation

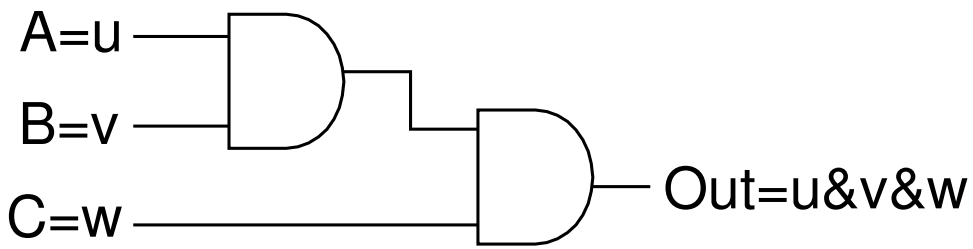
- Input patterns — Set inputs to logic value 0, 1
- Simulate — Propagate logic values
- Display logic values at output
- Example: n-input AND gate
  - 2^n input patterns



A	B	C	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

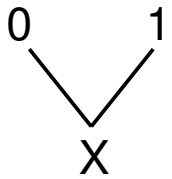
Symbolic Simulation

- Input patterns — Set inputs to variables
- Symbolic simulate — Propagate boolean expressions
  - BDDs for storing boolean expressions
- Display output expression
- Example n-input AND gate
  - 1 symbolic pattern with n variables.



Ternary Simulation

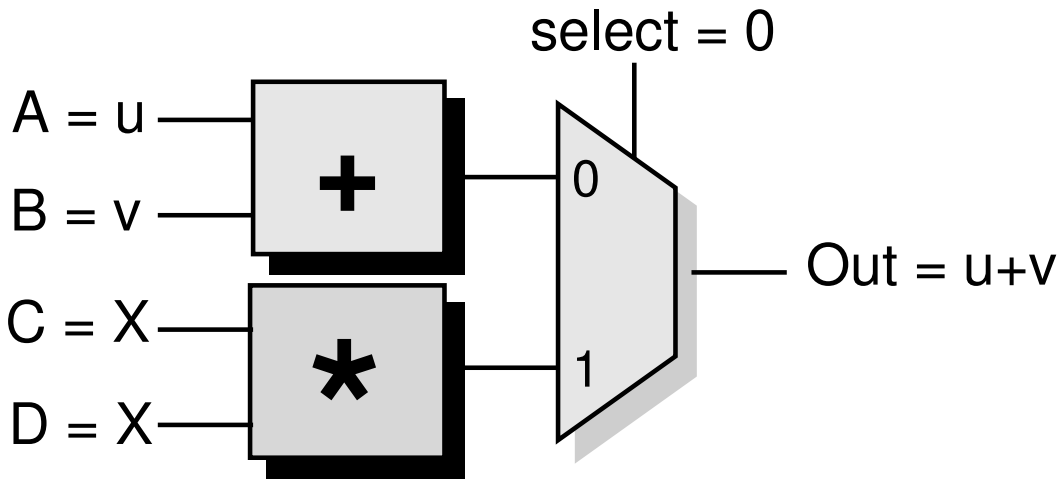
- Add X to logic value set
- Input patterns — Set inputs to 0,1,X
- Truth tables for gates extended to 0,1,X
  - Monotonic: $F(X) = 0 \rightarrow F(1) = 0 \wedge F(0) = 0$
- Example n-input gate
  - $n + 1$ input patterns



A	B	C	Out
0	X	X	0
X	0	X	0
X	X	0	0
1	1	1	1

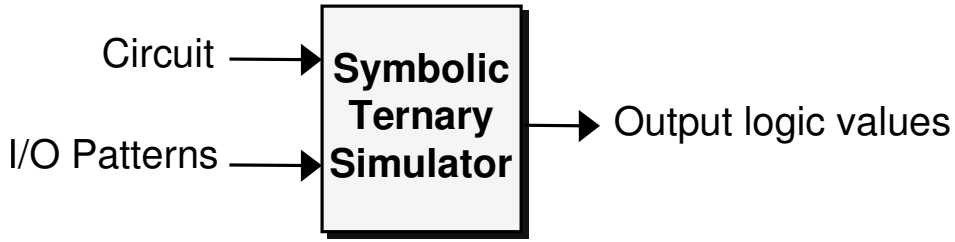
Symbolic Ternary Simulation

- Combine ternary and symbolic simulation
  - Expressive power of symbolic values
  - Computational Efficiency of ternary values

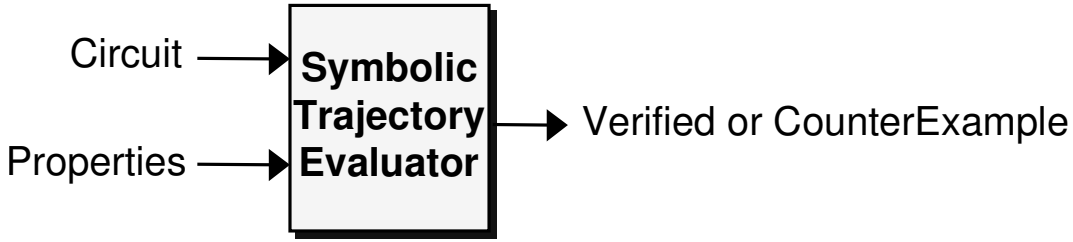


Symbolic Trajectory Evaluation

- Symbolic Ternary Simulation



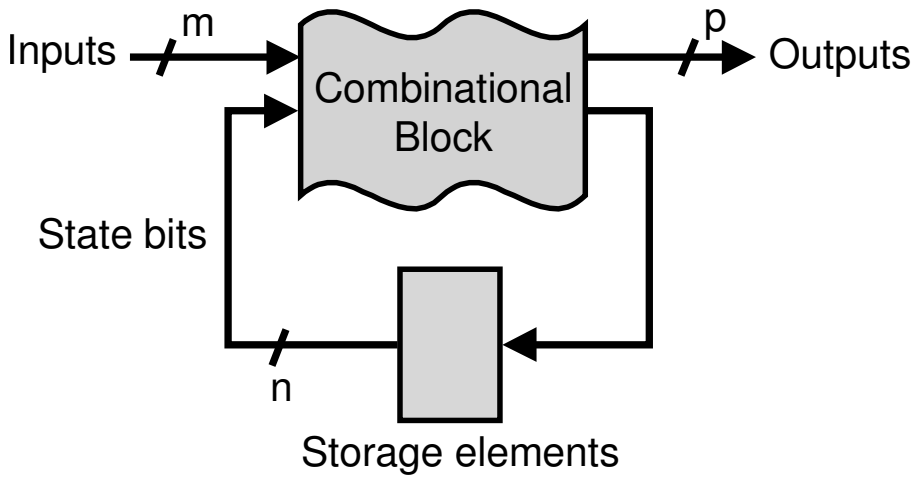
- Symbolic Trajectory Evaluation



  - Properties are a concrete means of specifying desired behavior

Circuit

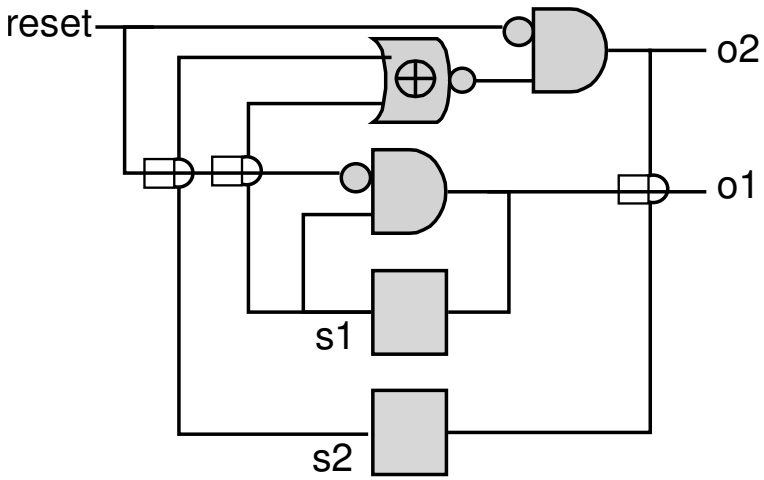
- Synchronous digital system
  - Zero delay combinational logic
  - Storage elements updated every positive clock edge

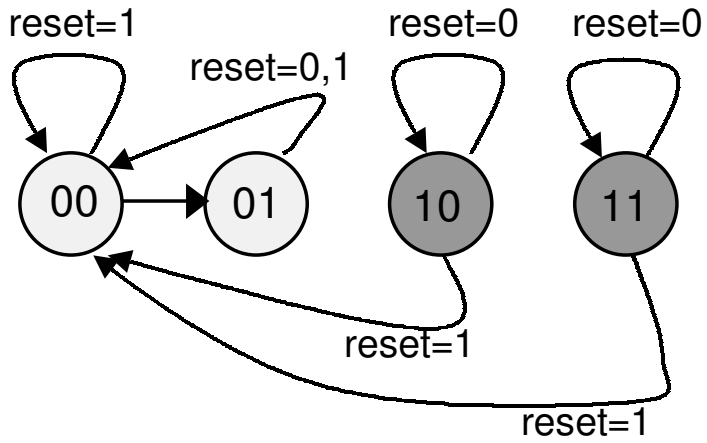


Example Circuit

Modulo 2 Counter

- Start state: 00
- Valid states: 00, 01
- Invalid states: 10, 11

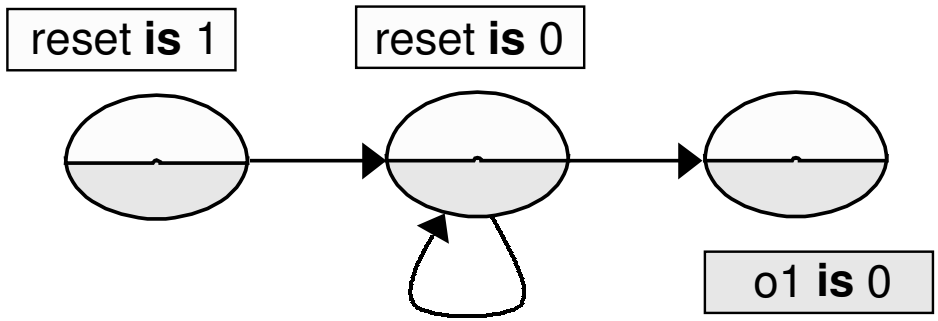




Properties

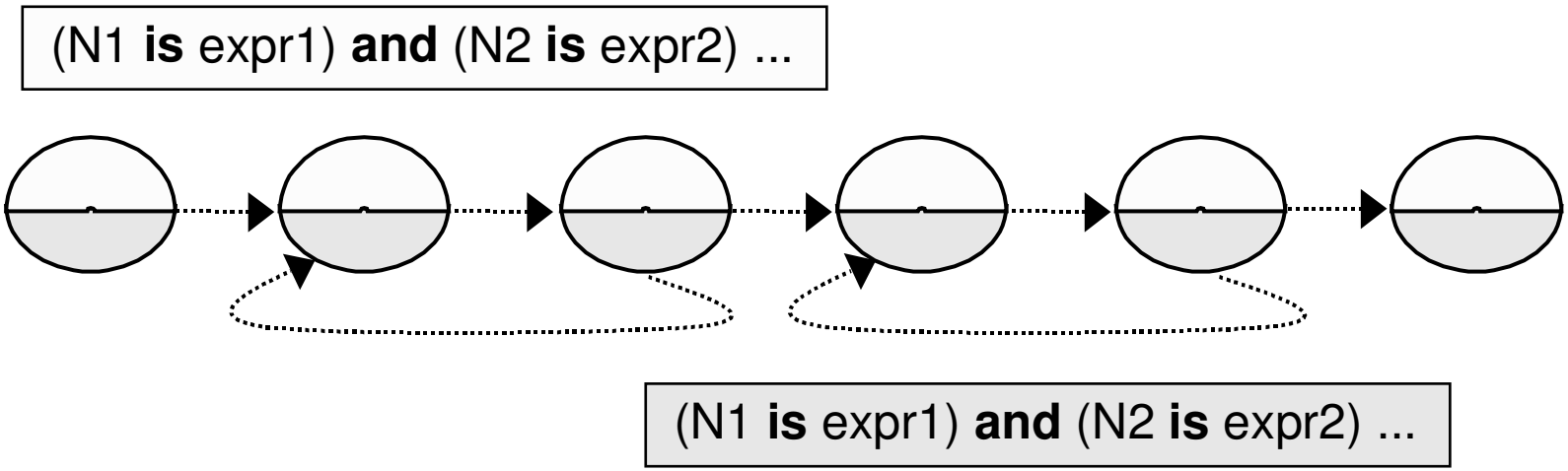
Example for Modulo 2 counter

- IF
  - Reset for 1 cycle and
  - Modulo 2 counting for an arbitrary number of cycles
- THEN
  - Next cycle output o1 should be 0 (valid states).

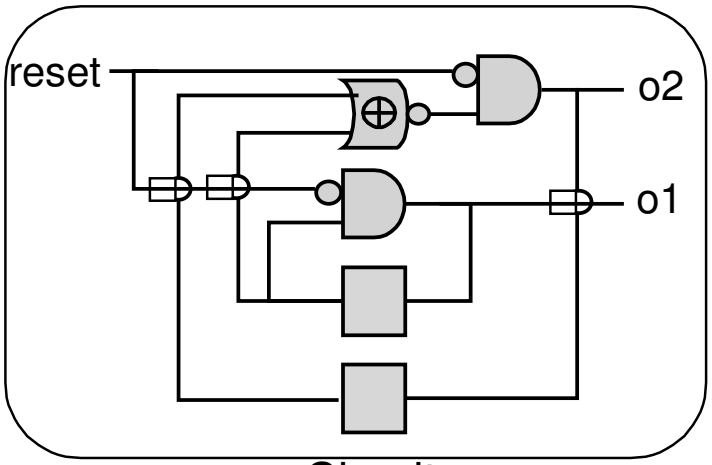


General Form of Properties

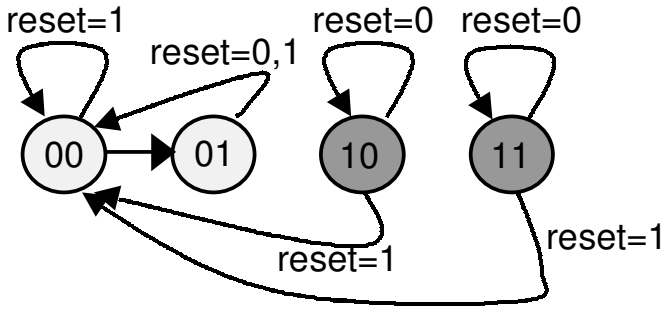
- Sequence of vertices — next-cycle operator
- Simple loops — arbitrary number of cycles
- Annotations
  - Upper half: *Asserts*, constraints on inputs and states
  - Lower half: *Checks*, desired behavior for outputs and states

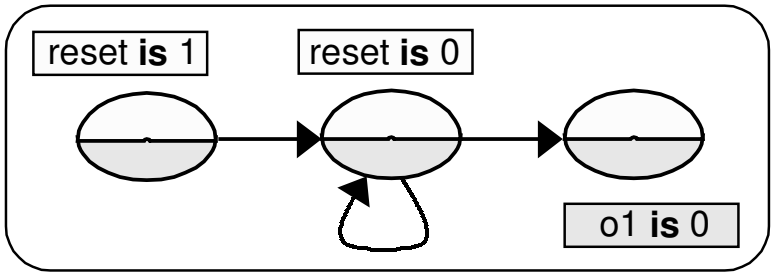


The Picture so far...



Circuit





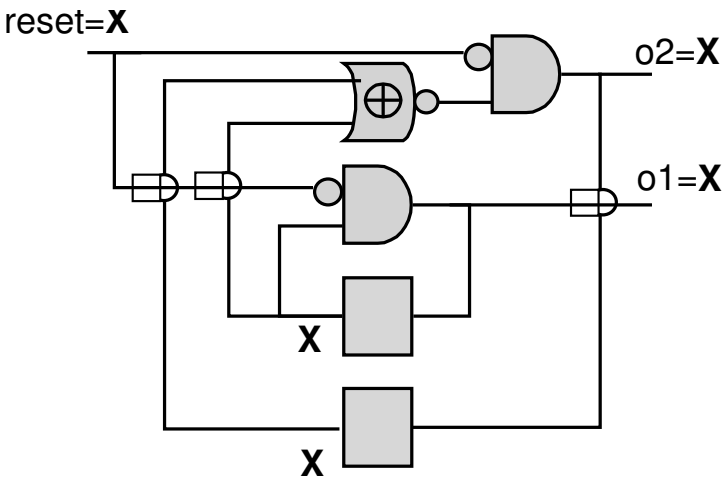
Property

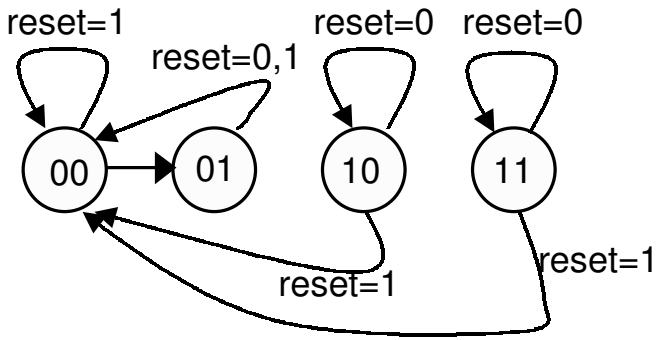
Symbolic Trajectory Evaluator

Does property hold for this circuit?

Trajectory Evaluation — Initialization

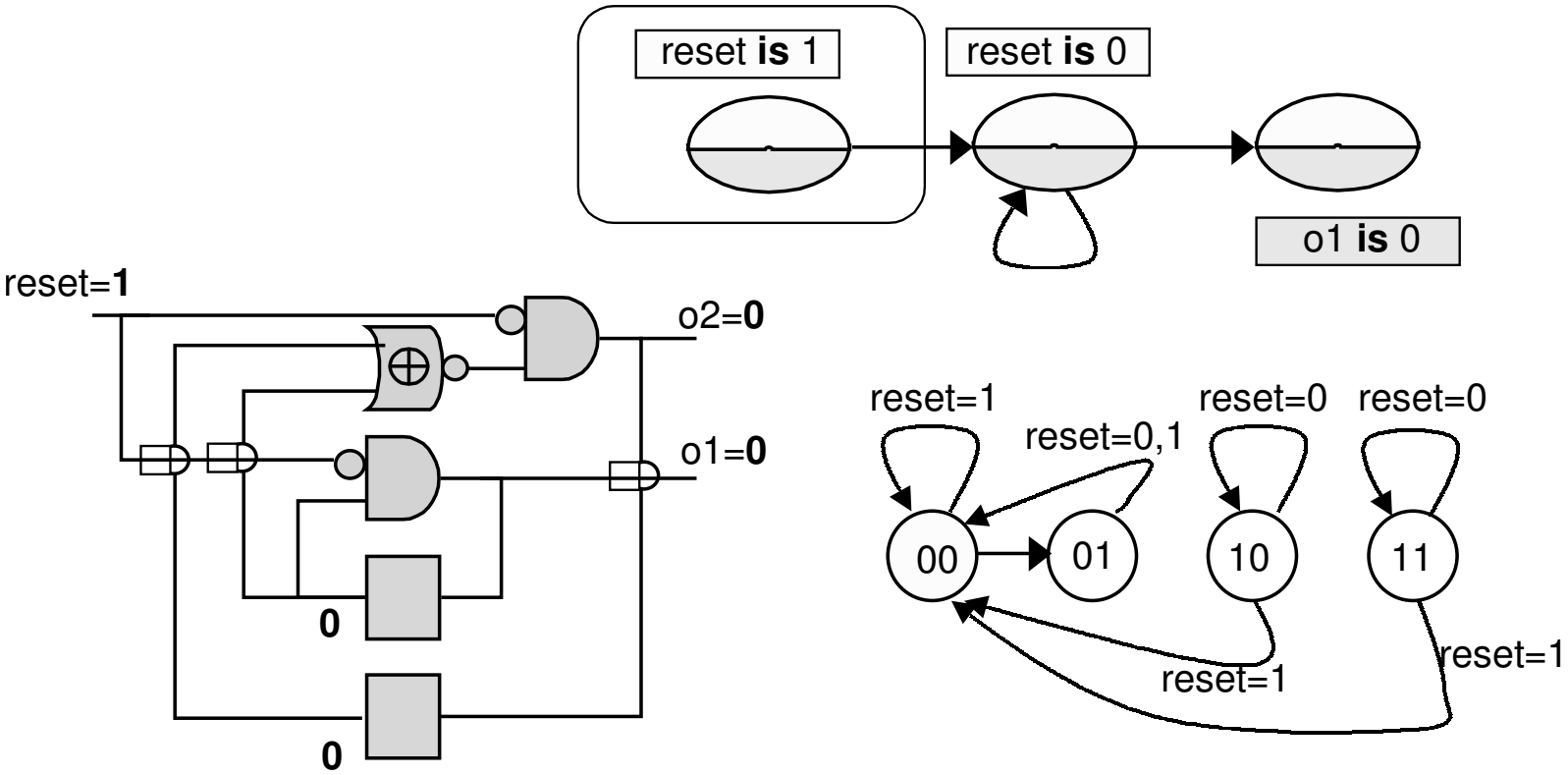
- All nodes to X
- XX represents any possible state: 00,01,10,11



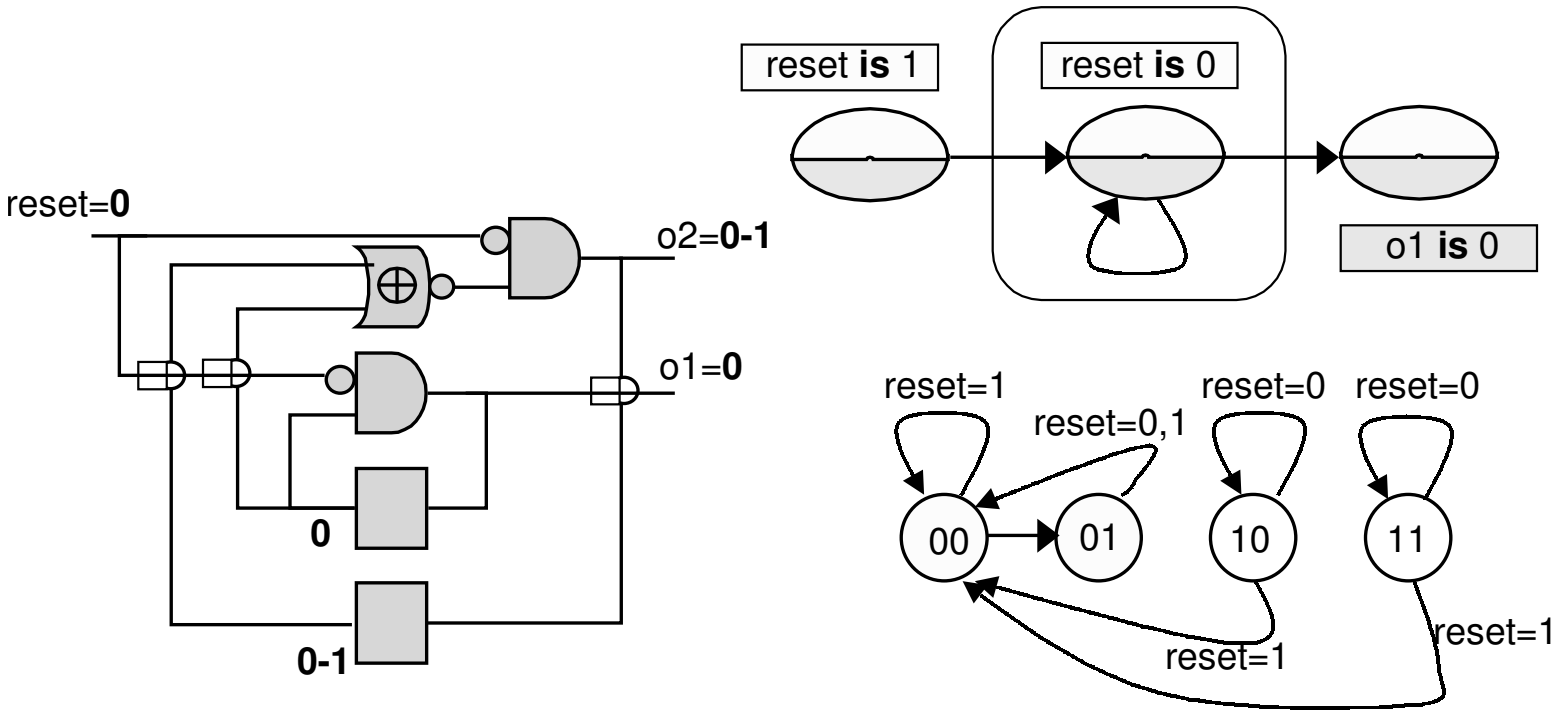


Trajectory Evaluation — Assert

- Property drives the evaluator



Trajectory Evaluation — Loops

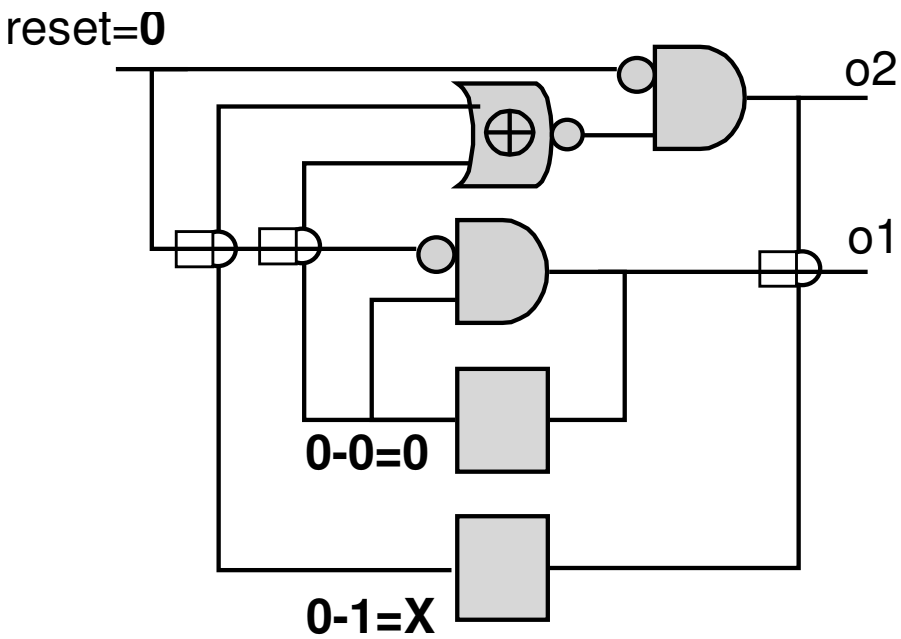


- Problem: Infinite loops!
  - Solution: reachable state set (from model checking!)

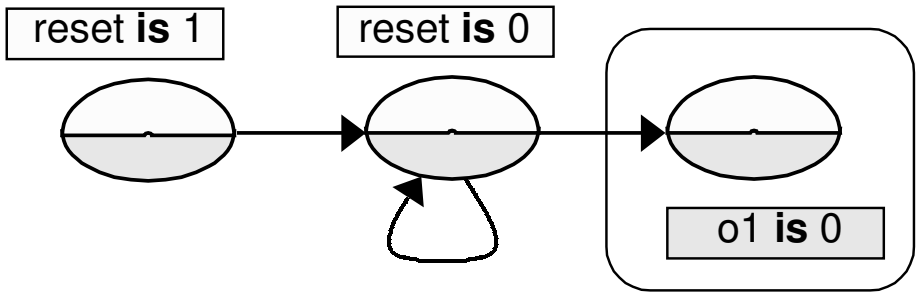
Trajectory Evaluation — Fix Point

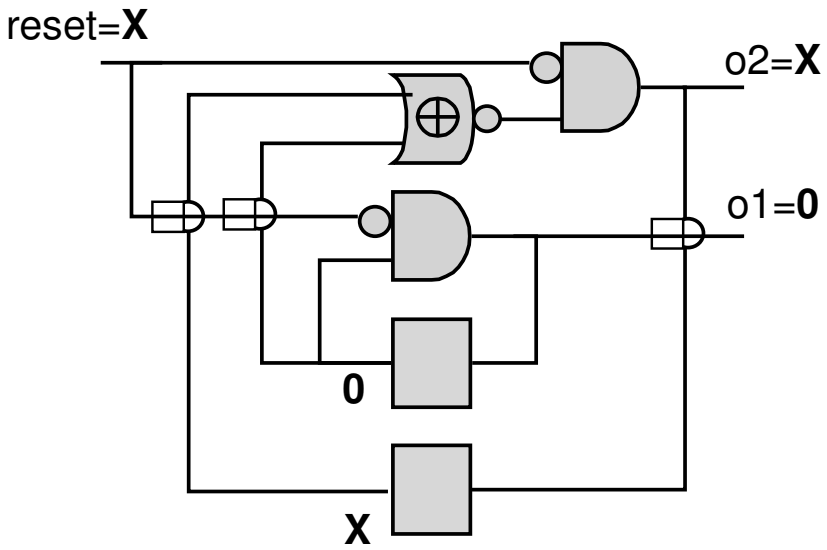
Fix point computation

- Keep previous state of circuit - 00
- Resolve new with previous state - Resolve(00,01) = 0X
- Continue until fix point - 0X
  - 0X represents {00,01}



Trajectory Evaluation — Verified!





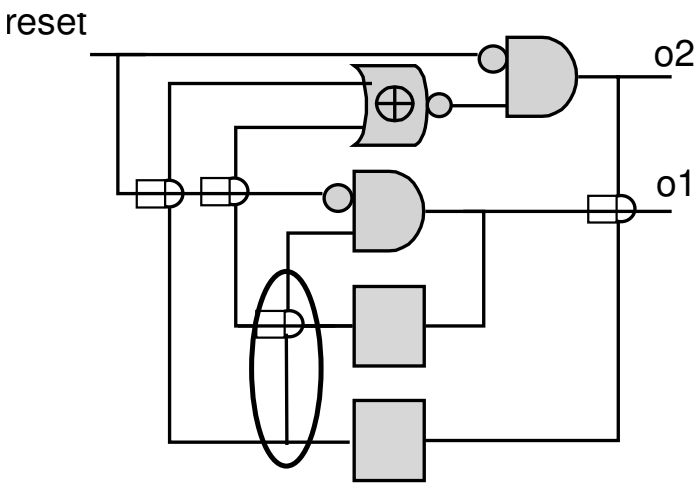
Perform check

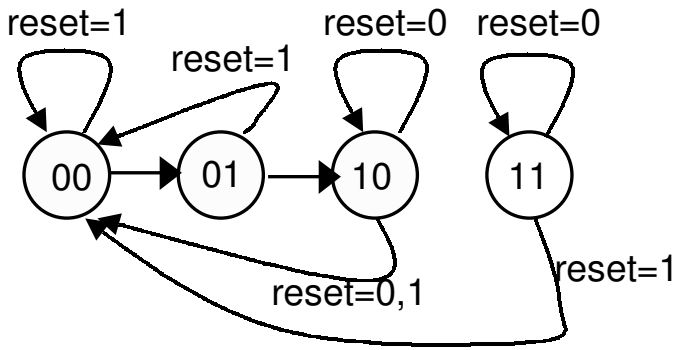
- Property Verified!

What if There is an Error in the Circuit ?

An error!

- Modulo 3 counter
- Reachable set 00, 01, 10 — Fix point XX
- Output o1=X. Property does not hold

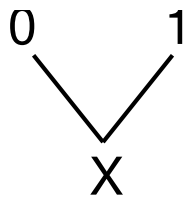




Theoretical Basis — Lattice Theory

- Partial order $\sqsubseteq$
  - $X \sqsubseteq 1$
  - $X \sqsubseteq 0$
  - No relation between 0 and 1

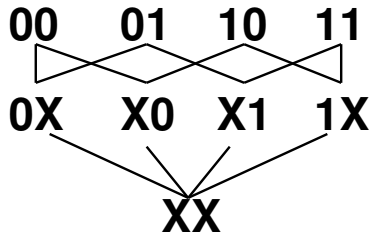
- Properties
  - Reflexive: $a \sqsubseteq a$
  - Anty-symmetric: $a \sqsubseteq b \wedge b \sqsubseteq a \rightarrow a = b$
  - Transitive: $a \sqsubseteq b \wedge b \sqsubseteq c \rightarrow a \sqsubseteq c$



Partially Ordered Sets — POSETS

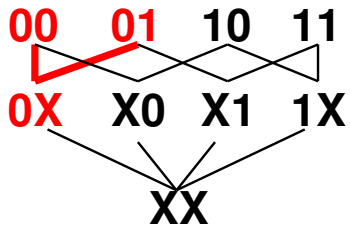
Poset $(S, \sqsubseteq)$

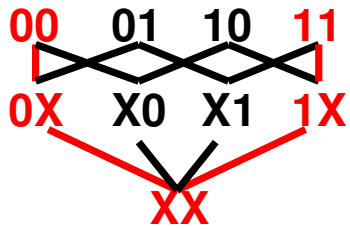
- S = {0,1,X}
- S = {00,01,10,11,0X,X0,X1,1X,XX}



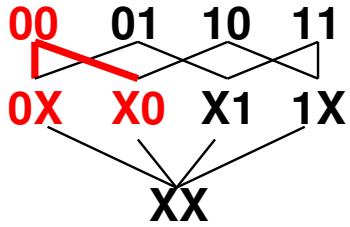
Operations

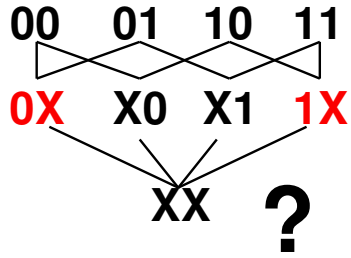
- Greatest lower bound





- Least upper bound





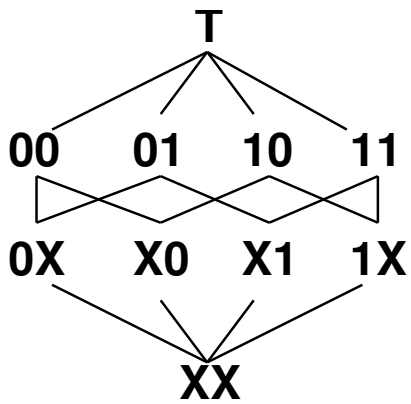
Complete Lattice

A POSET $(S, \sqsubseteq)$ is a complete lattice if

- Greates lower bound exists for every subset of S.
- Least upper bound exists for every subset of S.

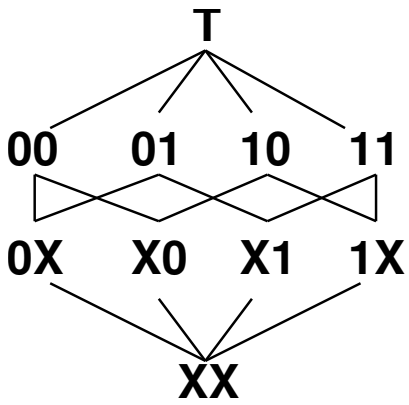
Introduce top element T.

- Universal lower bound: XX
- Universal upper bound: T



STE and Lattices

- Set S describes the ternary circuit state
  - s1 = 0 and s2 = X, circuit state = 0X
- The circuit state defines a complete lattice

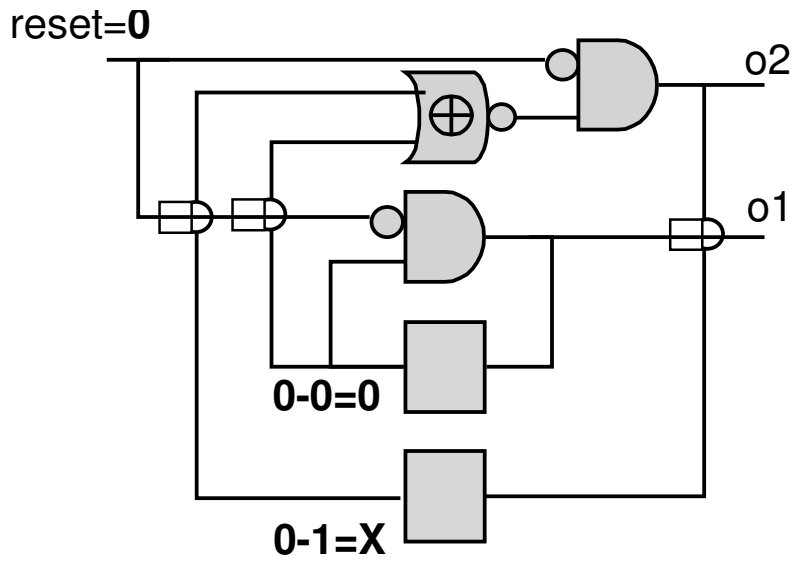


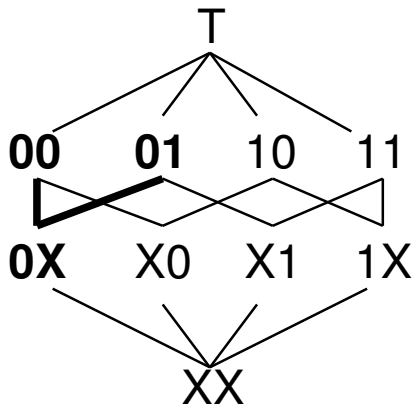
- STE can be explained in terms of lattice operations

STE and Lattice Theory

Example — fix point computation

- Greatest lower bound computation
- Guaranteed to converge
  - Universal lower bound XX





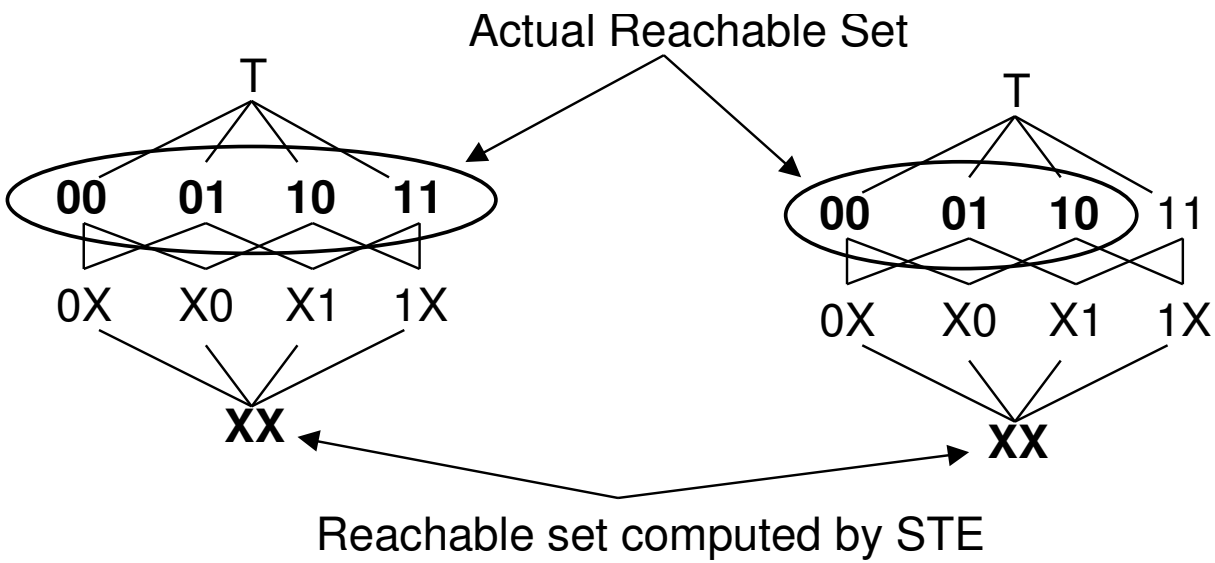
STE X Model Checking

- Advantages
  - Computation complexity
    - Does not build transition relation
    - Computes next-state on-the-fly as required by specification
  - Ternary modeling
    - Sets represented by single lattice element
    - Set (00, 01) represented by 0X
    - Set operations replaced by lattice operations
  - Operates directly on circuit
    - Allow complex timing models
- Disadvantages
  - Symbolic assertions less expressive than CTL
    - Until operator
    - No disjunction — either this or that property
  - Simulation based — only forward traversal
  - Ternary modeling
    - Pessimistic verification
    - Can lead to false negatives

Ternary Modeling — Pessimistic Verification

Can over-estimate the reachable state set

- Set $\{00, 01, 10\}$ is approximated by XX



Application

- Motorola and Intel
  - Verify data-intensive circuits — RAM, stack, register files, ALU
- HECTOR microprocessor
  - DAC 1994
  - Similar to PDP-11
  - Switch level circuit verified against instruction set architecture
- Part of PowerPC processor
  - DAC 1997
  - Fixed point unit in IBM's AS/400 Advanced 36 computer
  - 28,000 multi-input gates, ¿ 1000 latches
  - 50 hours of CPU time
  - 185 Mbytes of memory