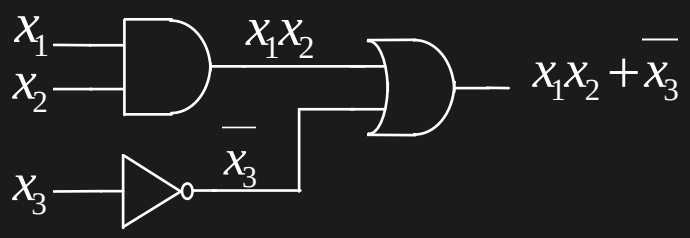
Analysis of Combinational Cycles in Synchronous Circuits

Tom Shiple
Advanced Technology Group
Synopsys, Inc.

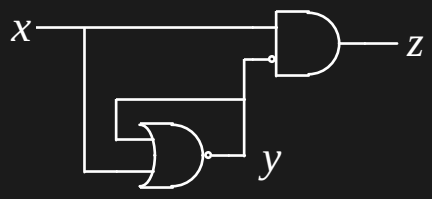
(Joint work with Gerard Berry, Robert Brayton, Alberto Sangiovanni-Vincentelli, Vigyan Singhal)

UCB CAD Seminar, 19 March 1997

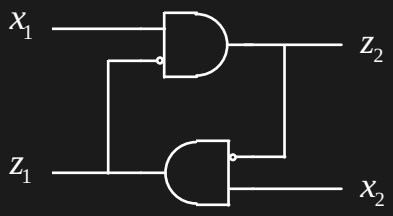
Acyclic Circuits



Cyclic Circuits



output-stable

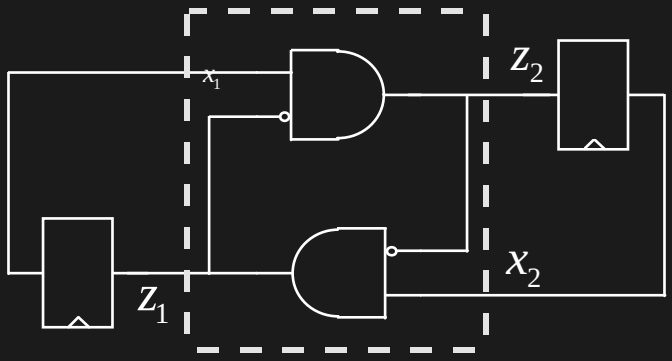


not output-stable

TRS 3/97

3

Cyclic Circuits with Flip-flops



TRS 3/97

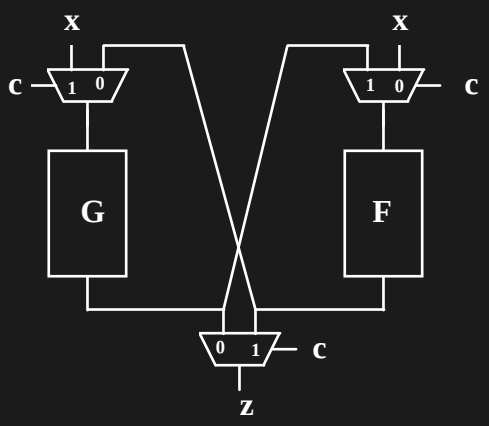
4

Who Cares?

Where do you ever see combinational cycles in the first place?

Resource Sharing

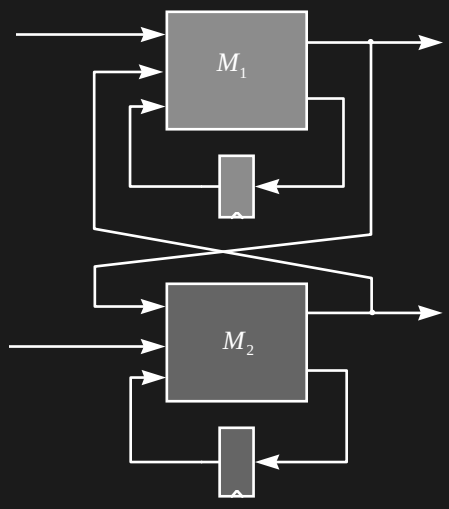
z = if(c) then F(G(x)) else G(F(x))



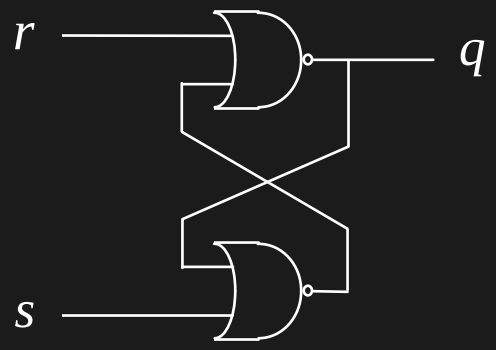
Behavioral-level Synthesis

```
S1   c = a + b;  1
     d = c + e;  2
-------------------
S2   f =  g + e; 3
     i =  f  + b; 4
```


Composition of Mealy FSMs



State-holding Circuits



TRS 3/97

9

General Statement of the Problem

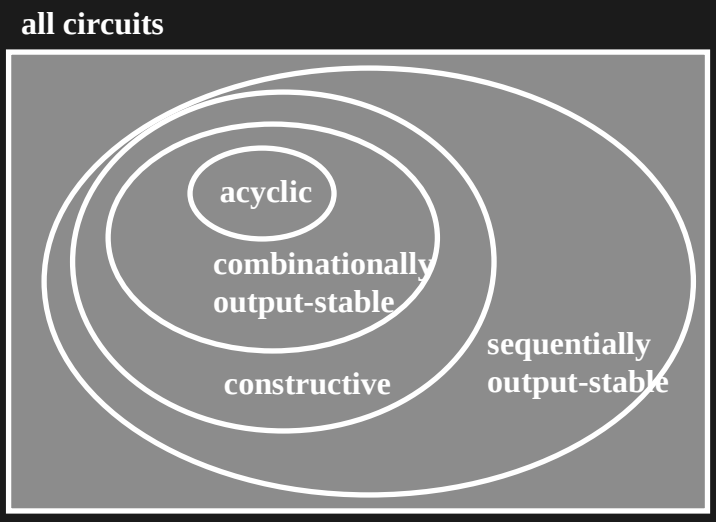
Given: an arbitrary interconnection of logic gates and flip-flops

Question: for each input sequence, does there exist a unique, stable output sequence?

TRS 3/97

10

Classes of Output-stable Circuits



TRS 3/97

11

Distinguishing Assumptions

Assumption	combinational out-stability	constructive	sequential out-stability
comb. wires forget their values	YES	YES	NO
flip-flop inputs must be stable	YES	YES	NO
all states reachable	YES	NO	NO

conservative ⟷ permissive

TRS 3/97

12

Algorithms

- Decision procedure for each class
- If circuit is in class
  - generate acyclic circuit with same I/O behavior
  - important as preprocessor for some CAD tools (e.g., cycle simulation, formal verification)
- If circuit not in class
  - generate an error trace
- Check membership in class in presence of a user-provided environment

Outline

- Related work
- Delay model
- Combinationally output-stable circuits
- Constructive circuits
- Sequentially output-stable circuits
- Conclusion

Related Work

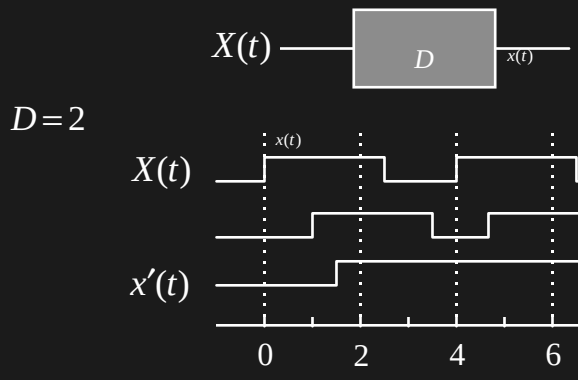
- Malik ‘93
  - introduced combinationally output-stable class
  - suggested ternary simulation as analysis tool
  - did not formalize delay model
- Halbwachs and Maraninchi ‘95
  - notion of stability is Boolean consistency
  - conservative in some cases, permissive in others

Related Work (cont.)

- FSM extraction from transistor netlists
  - Kam ‘95, Pandey ‘95, Singh ‘95
  - class of stable circuits not formalized
    - » classification implicit in the result of their algorithms
    - » no proof that extracted cct has same behavior as original
  - conservative in some cases, permissive in others

Delay Model

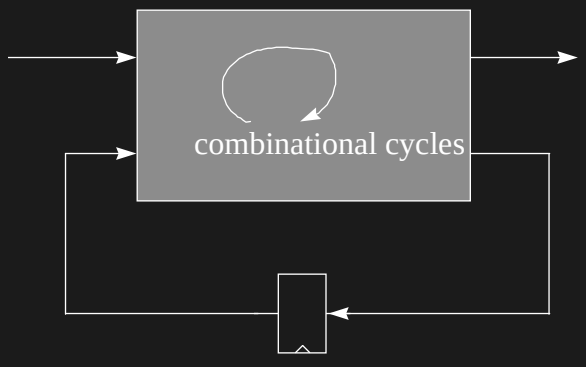
Up-bounded inertial delay
[Brzozowski and Seger, '90]



TRS 3/97

17

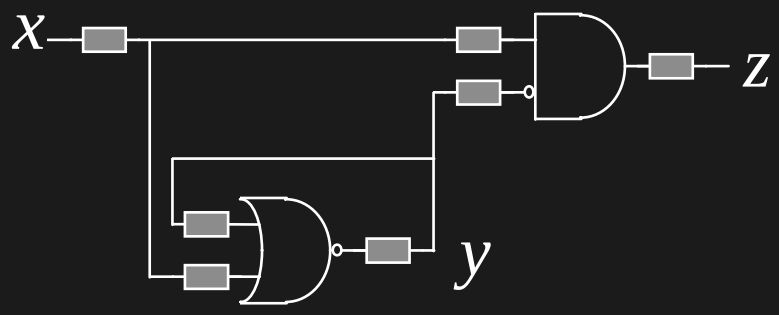
Circuit Model



TRS 3/97

18

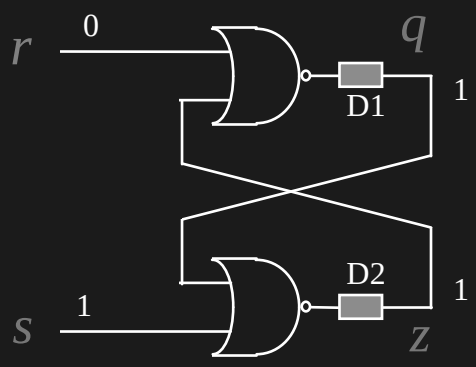
Circuit Model



TRS 3/97

19

Circuit Behavior Over Time



TRS 3/97

20

Non-transient States

If the circuit output eventually stabilizes, it will do so within (2^m - 2)D time of input change

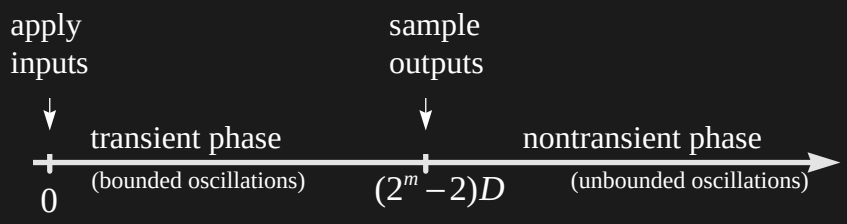
Outline

- Related work
- Delay model
- Combinationally output-stable circuits
- Constructive circuits
- Sequentially output-stable circuits
- Conclusion

Combinational Output-Stability

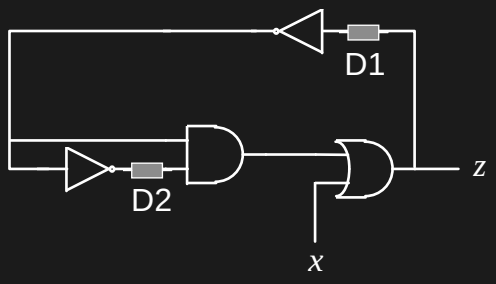
- Treat flip-flop I/O as primary I/O
  - implies stability required on flip-flop inputs
  - implies all states treated as reachable
- Assume comb. wires cannot hold state
  - implies initial value of comb. wires unknown

Definition of Comb. Output-Stability



- For every input assignment, the output is unique in the nontransient phase
  - (regardless of initial values on internal wires)

Dependence on Placement and Value of Delays



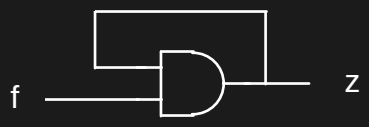
- dependence on placement of delays
  - put them everywhere to be conservative
- no dependence on upper delay bound

TRS 3/97

25

Problem is co-NP-complete

NP-hard: reduction from satisfiability



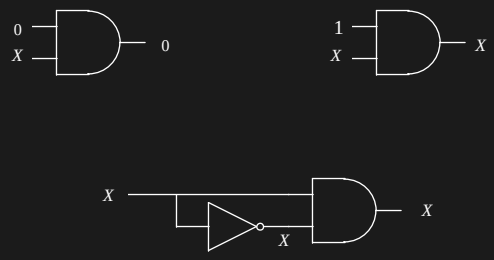
NP: use ternary simulation to verify guess of a “bad” input

TRS 3/97

26

Ternary Simulation

- *X* - uncertain or changing value

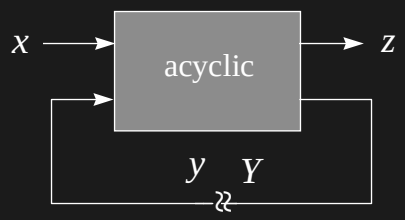


TRS 3/97

27

Alg. for Testing Comb. Out-Stability

- Break feedback
- Initialize feedback wires to *X*



TRS 3/97

28

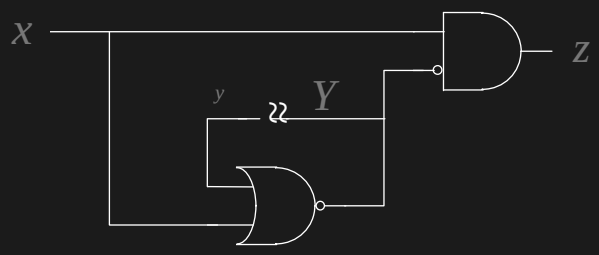
Alg. for Testing Comb. Out-Stability

- Apply ternary simulation to convergence
  - corresponds to propagation of facts
  - ternary extension of a Boolean fn is monotonic
    - »guarantees convergence within k passes
  - implemented using a pair of BDDs for each gate output
- Circuit is comb. out-stable iff there are no X's at outputs when convergence reached

TRS 3/97

29

Example 1

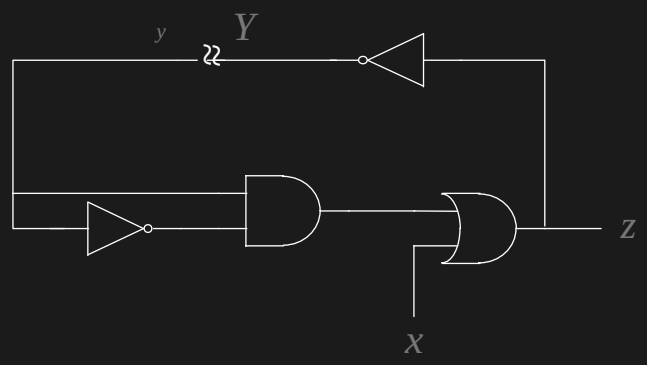


- Combinationally output-stable
- Halbwachs: not "consistent"

TRS 3/97

30

Example 2



- Not combinationally output-stable
- Halbwachs: consistent

TRS 3/97

31

Outline

- Related work
- Delay model
- Combinationally output-stable circuits
- Constructive circuits
- Sequentially output-stable circuits
- Conclusion

TRS 3/97

32

Sequential Circuit Operation

- Clock period sufficient to allow outputs to settle
- All flip-flops enabled by same clock
- Inputs change at each clock tick
- Outputs sampled at clock ticks

TRS 3/97

33

Definition of Constructivity

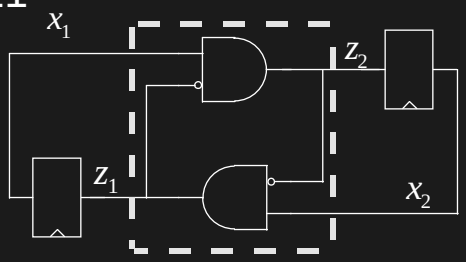
- For each input *sequence*, the output and next state sequences are stable and uniquely determined
  - under the assumption that comb. wires do not hold state across clock ticks (and hence have arbitrary values)
- Same as comb. out-stability, except stability need hold only for reachable states
- PSPACE-hard

TRS 3/97

34

Example

- Combinational part of circuit not constructive for $x_1x_2 = 11$
- However, this state is not reachable when initial state is anything other than 11



TRS 3/97

35

Algorithm for Testing Constructivity

- Perform ternary simulation on combinational part to determine present state values that induce an X at an output or next-state line
- Perform reachability to determine if any of these “bad” states are reachable

TRS 3/97

36

Symbolic Reachability Analysis

- Symbolic ternary simulation yields a pair of Boolean functions for each output F
  - F^1 = inputs for which F is 1
  - F^0 = inputs for which F is 0
  - F^X = $\overline{F^1 + F^0}$ = inputs for which *F* is *X*
- Example

$$F^1_{z_1} = \overline{x_1} x_2$$

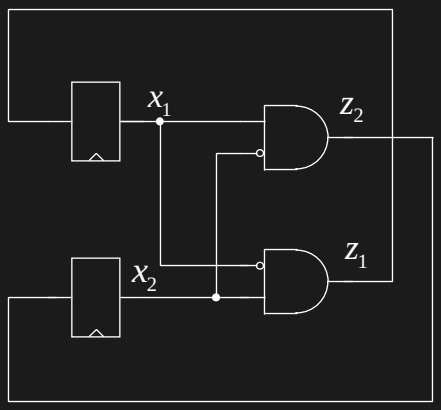
$$F^0_{z_1} = \overline{x_2}$$

$$F^X_{z_1} = x_1 x_2$$

Symbolic Reachability Analysis

- For each next state signal *F*, use F^1 as next state function, under assumption that circuit is constructive
  - check each new set of reached states for “bad” states
- If “bad” state is reached => not constructive
  - generate error trace working backwards
- If fixed point reached => constructive
  - generate equivalent acyclic circuit, using functions F^1

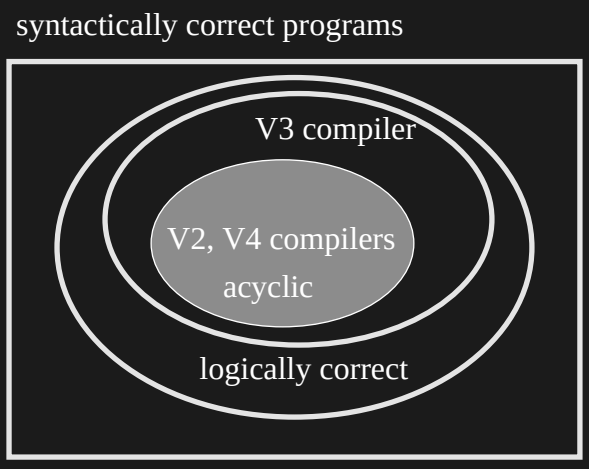
Example



Application To Esterel

- Esterel: synchronous language for reactive systems
- Syntactically correct programs can have combinational cycles
- Problem: which programs are “well-behaved”?
- Previous semantics: logical correctness
  - non-intuitive: doesn’t correspond to information propagation
  - NP-complete for a single input

Logical Correctness Semantics

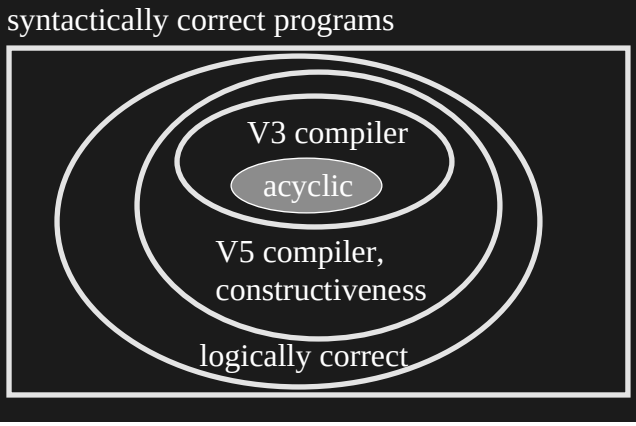


TRS 3/97

41

New Semantics: Constructiveness

- Corresponds to information propagation
- Test is linear for a single input



TRS 3/97

42

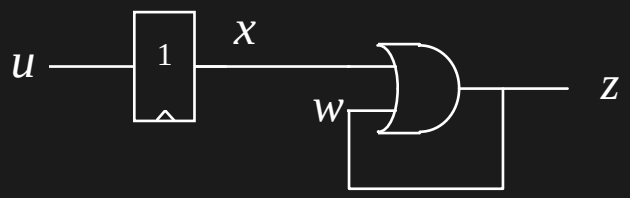
Full Abstraction Theorem

- A program is constructive *if and only if* the translated circuit is constructive
- V5 compiler uses constructiveness algorithm
- Example
  - program of 1043 lines
  - 3989 wires, 157 flip-flops
  - 11.5 minutes (DEC AlphaServer2000)

Outline

- Related work
- Delay model
- Combinationally output-stable circuits
- Constructive circuits
- Sequentially output-stable circuits
- Conclusion

Example



- Constructivity ignores storage at z
- Also, constructivity requires stability at flip-flop inputs

Defn of Sequential Output-Stability

A circuit is sequentially output-stable if

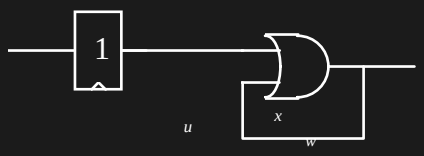
- for each input sequence,
- there exists a unique, stable output sequence

Problem is PSPACE-hard.

Outline of Algorithm

- Create the clock-level state transition diagram
  - over flip-flop/combinational wire values,
  - where transitions are between non-transient states
- Compute the bisimulation equivalence relation over states, wrt the output values
- Form the quotient machine, wrt bisimulation
- Check the quotient machine for next-state determinacy

Example 1



input: 1 0 0 1 0 ...
output: 1 1 1 1 1 ...

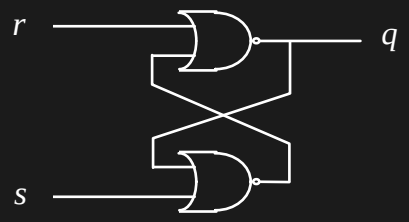
⇓

sequentially output-stable

⇓

generate equivalent acyclic circuit

Example 2



input: 11 01 11 00 10 ...
output1: 0 1 0 **0** 0 …
output2: 0 1 0 **1** 0 ...

⇓

not sequentially output-stable

⇓

generate error trace

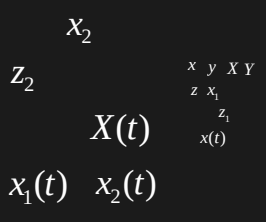
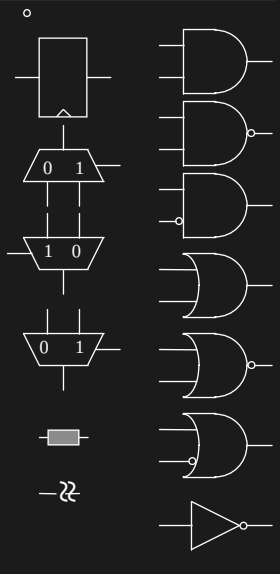
Environment for RS-latch

Another Example

Future Work

- Allow transistors, tri-state devices, multi-phase clocks
- Develop implicit procedure for testing sequential out-stability
- For stable ccts, derive acyclic implementations that preserve more structure
- Develop theory for “n-delay output-stability”

Conclusions

- Combinational cycles arise in practice
- Formalized Malik's notion of combinational output-stability
- Generalized to
  - constructivity: correspondence to well-behaved Esterel programs
  - sequential output-stability
- Can produce equivalent acyclic circuit when output-stable

Gates