Induced Hierarchical Verification of Speed-Independent Circuits

Vida Vakilotojar
Advisor: Peter A. Beerel
Department of EE-Systems
University of Southern California

TECHCON'98, Sept. 9-11

Talk Outline

Background

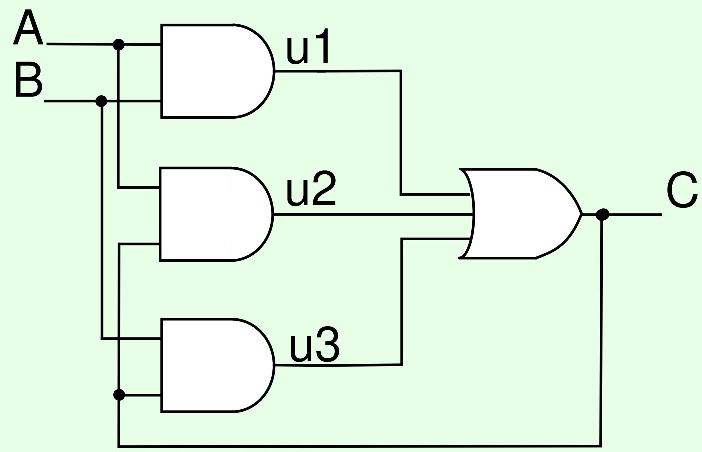
- Speed-independent (SI) circuits
- Associated verification tasks

Hierarchical verification of SI circuits

- Previous work
- Overview of our approach

Preliminary results and summary

Speed-Independent Circuits



Sum-of-Products Muller C-element

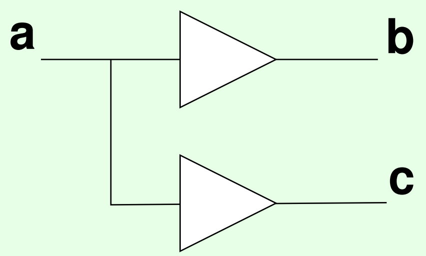
Asynchronous circuits

- Sequential digital circuits with no clock

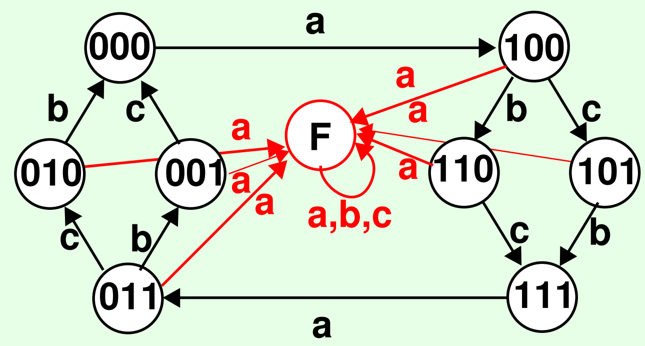
Speed-Independent circuits

- Should “work correctly” regardless of gate delays

Behavioral Model of a Circuit [Dill'89]



Two buffers with
a common input



Behavioral model
of the two buffers

Failure state

- Entered if gate output is disabled (exhibits a hazard)

Interleaving semantics

- No simultaneous transitions

Represents speed-independence assumption

- Arbitrary gate delay + zero wire delay

Verification Tasks

Failure-freedom

- A closed circuit (no primary inputs) is failure-free iff the failure state is not reachable.
  - A failure-free circuit is hazard-free

Trace Conformance [Dill '89]

- The spec. can be safely substituted by the circuit
- Conformance holds iff the circuit composed with the mirror of the specification is failure-free

The point

- Verifying conformance and hazard-freedom can be reduced to verifying failure-freedom

Verification of SI Circuits

Dill '89

- Supports hierarchical verification
- Key limitation: explicit state technique
  - Can't handle large flat circuits

McMillan/Burch et al '94

- Implicit state techniques (BDDs)
- Good experimental results on some circuits
- Same theoretical complexity as explicit state technique

Roig et al '95

- Improved BDD methods using a mixture of breadth-first and depth-first search

Induced Hierarchical Verification of SI Circuits

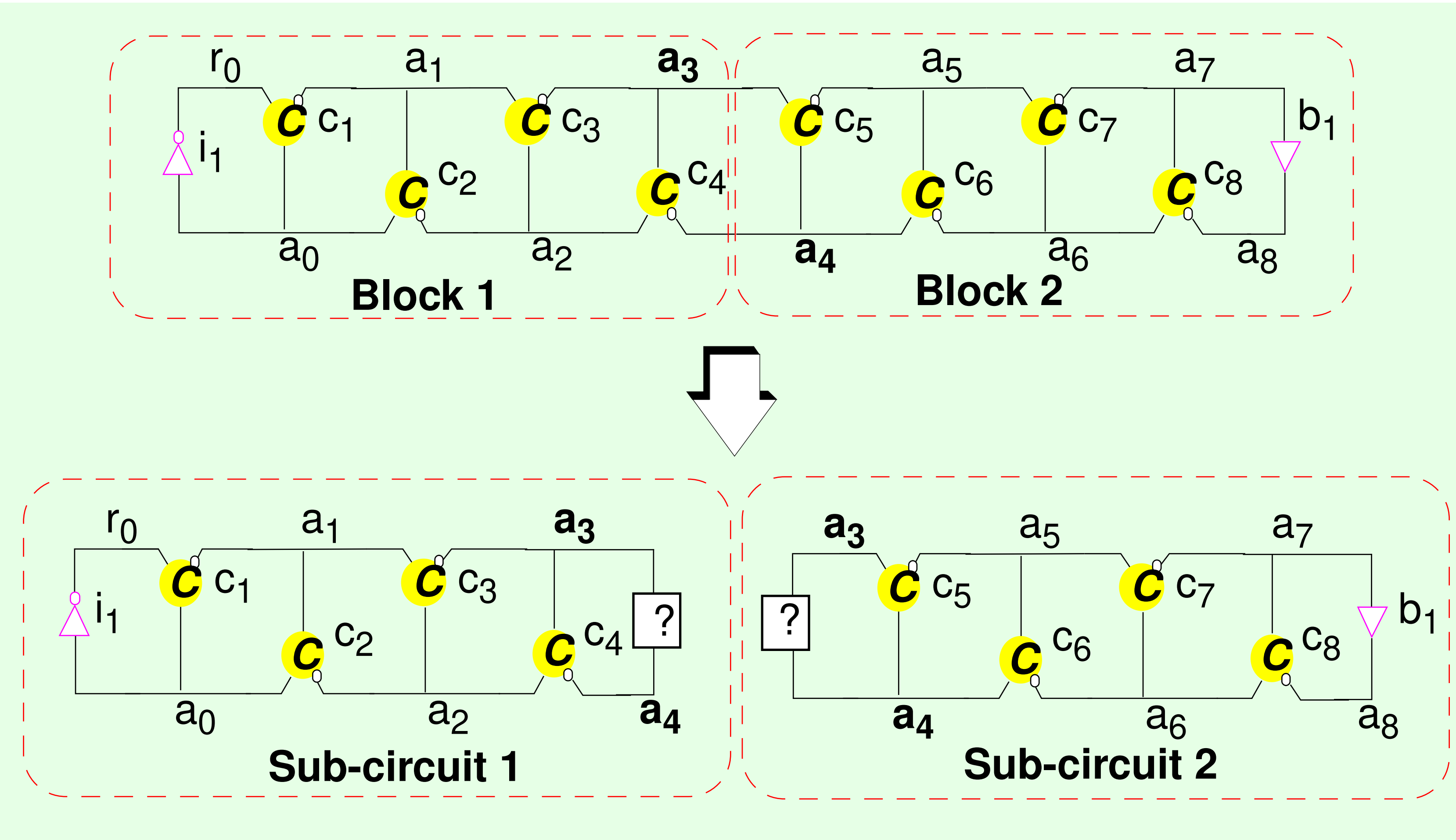
Goal

- Attack state explosion problem by creating hierarchy in an otherwise flat circuit

Approach

- Partition the circuit into blocks
- Efficiently derive specification for each block
- Verify conformance of blocks hierarchically

Illustrating The Approach



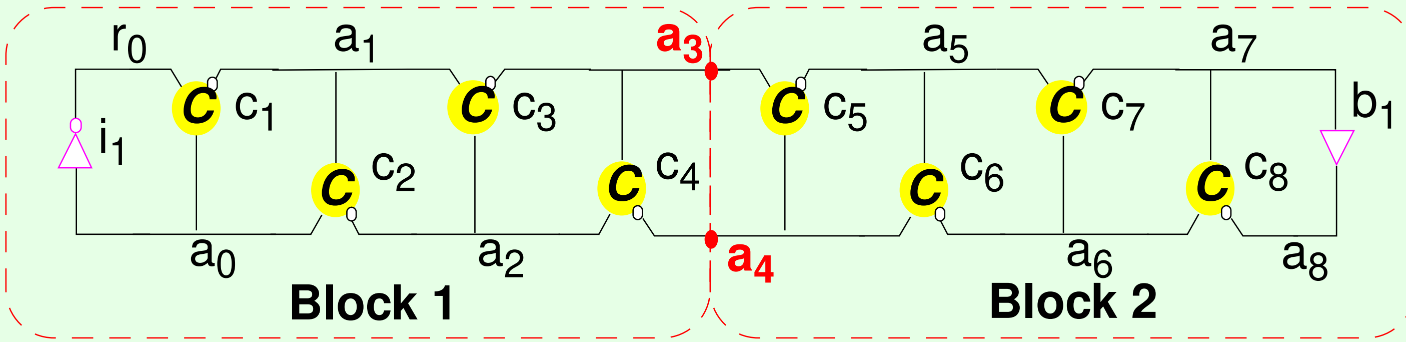
Leads to exponential reduction in complexity if deriving compact bloc k specifications is c heap

Safe Abstraction

Definition

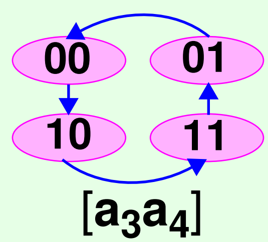
- A non-empty behavior (set of traces starting with the initial state) over a set of signals *E* is a safe abstraction of the actual behavior over *E* if the behavior
  - is exact when the circuit is failure-free
  - is an under-approximation when the circuit has a failure

Example of a Safe Abstraction



$E = \{\mathbf{a_3}, \mathbf{a_4}\}$

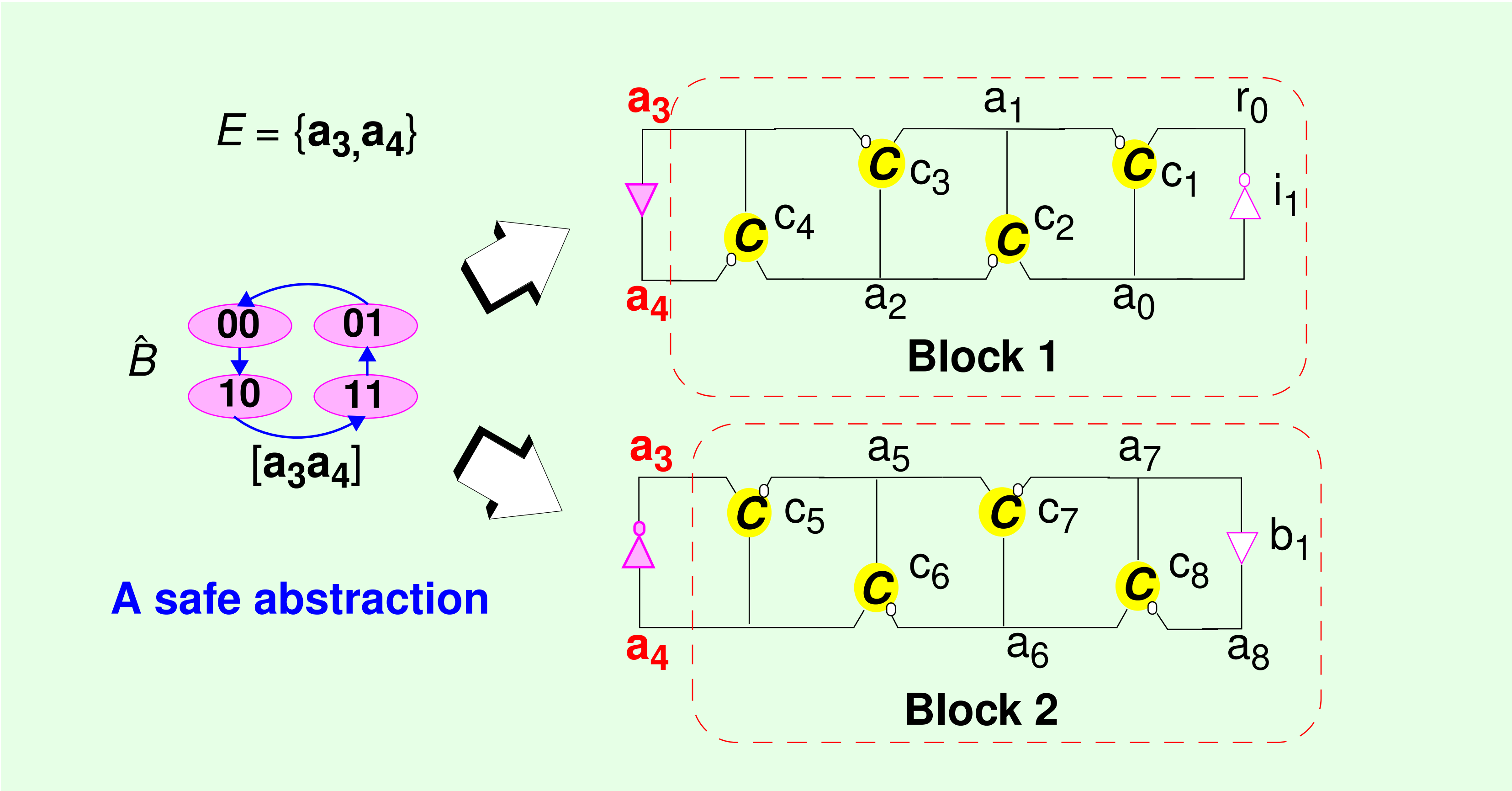
$\hat{B}$



The circuit is speed-independent (failure-free) and $\hat{B}$ exactly represents the actual behavior of the external signals, $E = \{\mathbf{a_3}, \mathbf{a_4}\}$.

$\hat{B}$ is a safe abstraction of the behavior of E.

Deriving Block Specifications



Conformance of each block to its specification (i.e., failure-freedom of each sub-circuit) is verified hierarchically.

The Key Result

Claim

- Given a safe abstraction $\hat{B}$ of the behavior of a set of external signals E, a circuit is failure-free *iff* all of its sub-circuits are failure-free.

Intuition

- Under-approximation means that the output of some-sub-circuit is under-estimated.
- When verifying this sub-circuit it will produce an illegal output.

Our hierarchical verification is exact; i.e., yields no false positive or false negative results.

Finding Safe Abstractions

Assumption

- Assume that the circuit is speed-independent (i.e., failure-free)

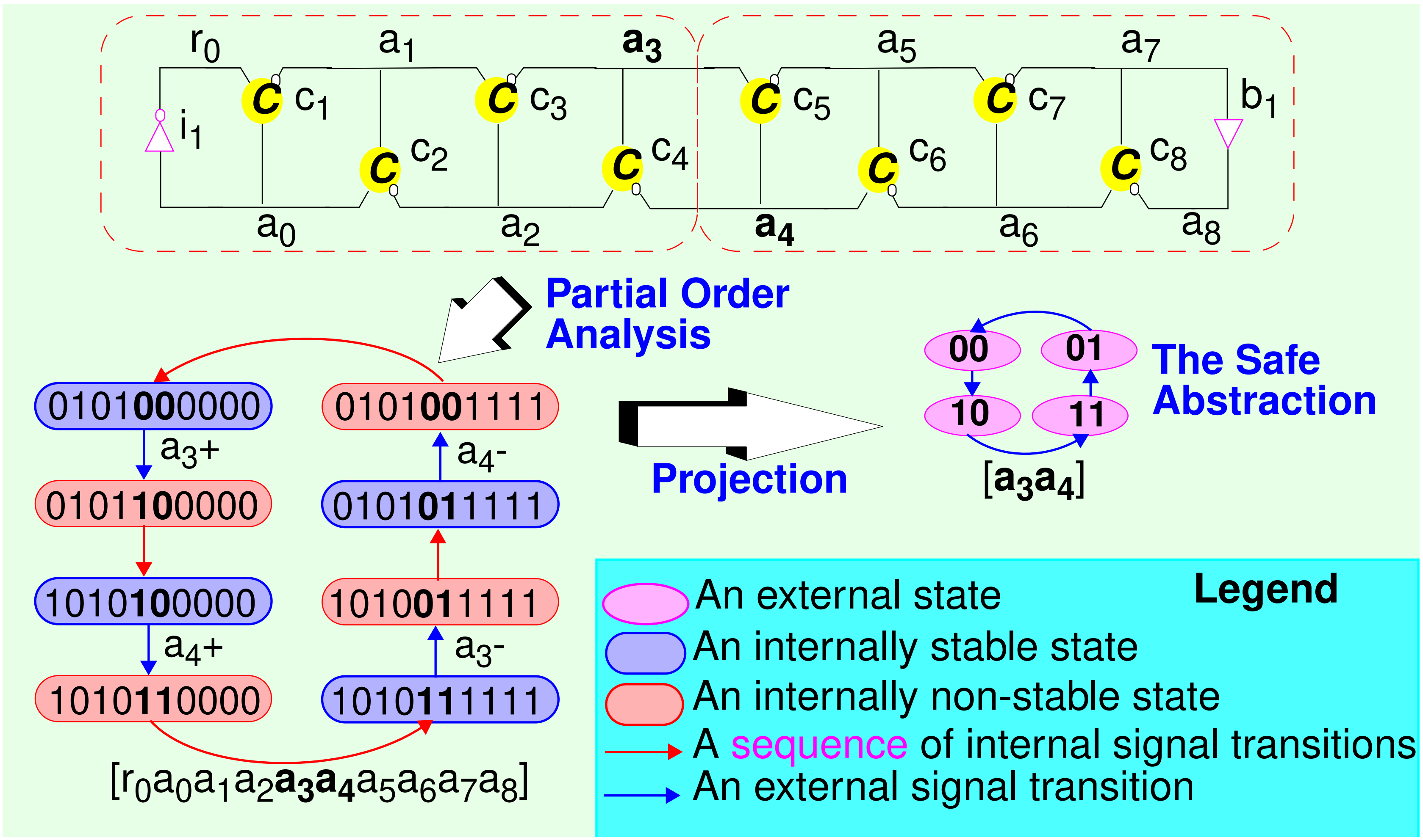
Key implication

- Can efficiently derive a safe abstraction of the behavior of *E* using an efficient partial order technique [similar to Weih & Greenstreet '96]

The Justification

- If the assumption does not hold, the partial order technique may lead to an under-approximation.
- It can never lead to an over-approximation.

Illustrating Our Partial Order Approach



Always let internal signals settle before firing any external transition

Implementation Considerations

Goal

- Want to use symbolic techniques (BDD’s)
- Represent safe abstractions using TRs over only external signals

Issue

- Not all safe abstractions can be modeled by a TR
- Must satisfy USC property
  - Identical external states must have the same next states
  - Limits possible set of external signals
  - We implemented algorithms to check for USC

Relationship to Previous Work

Previous induced hierarchical verification efforts

- Collapse combinational gates
  - Explicit state [Beerel et al ’93]
  - Implicit state [Roig et al ’94]

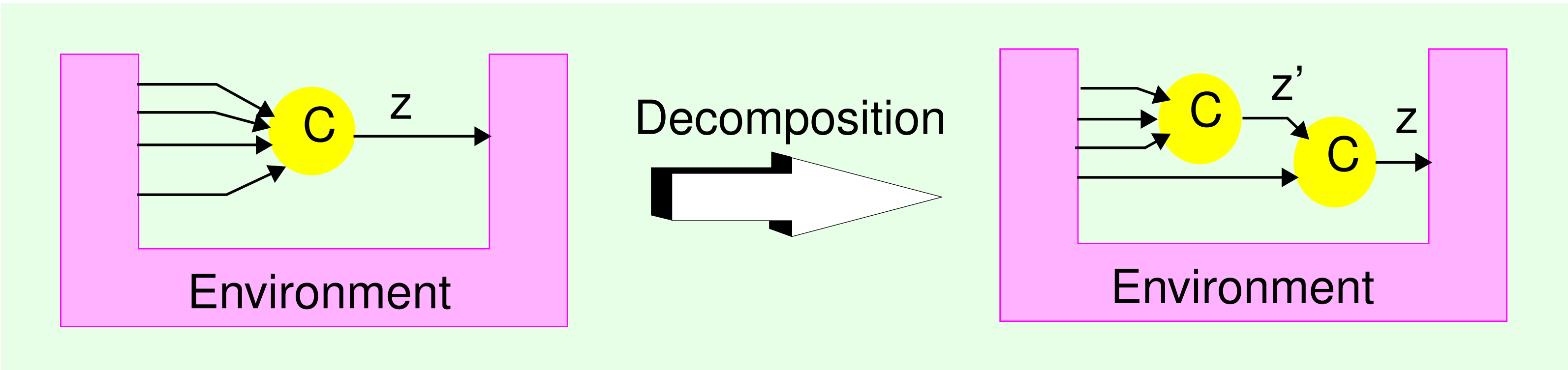
Key limitations

- Memory element outputs must be external
  - This significantly limits their applicability

Our contribution

- Generalized approach that allows *some* memory elements to be hidden
- Generalized notion of cyclic combinational circuits [Malik ’93, Wolf 93, Shiple ’96]

Application to SI Technology Mapping



Recent SI technology mapping techniques

- Often use sequential decomposition
  - Burns '96, Cortadella et al '97

Our contribution: faster verification of these circuits

- Newly introduced internal memory elements can be hidden

Preliminary Results

Status of symbolic prototype tool

- External signals are user defined
- One-level of hierarchy implemented

Preliminary tool run-times

Circuit	# of blocks	Flat-Circuit Run-Time (ms)	One-Level Hierarchy Run-Time (ms)	Run-time Ratio	Peak BDD Node # Flat	Peak BDD Node # Hierarchical	BDD Ratio
FIFO 4	2	90	110	~1	999	539	~2
FIFO 8	2	690	320	~2	4,959	2,410	~2
FIFO 16	2	33,510	2,090	~16	28,516	8,106	~3.5
DMEring 2	2	4,280	1,850	~2	12,395	15,496	~1
DMEring 3	3	105,630	3,560	~30	80,675	21,466	~4
DMEring 4	4	627,830	7,000	~90	406,103	34,261	~12

Summary

Introduced hierarchical verification of SI circuits

- Combines assume-guarantee paradigm and partial-order techniques
- Able to hide *some* memory elements
- Has an efficient symbolic implementation

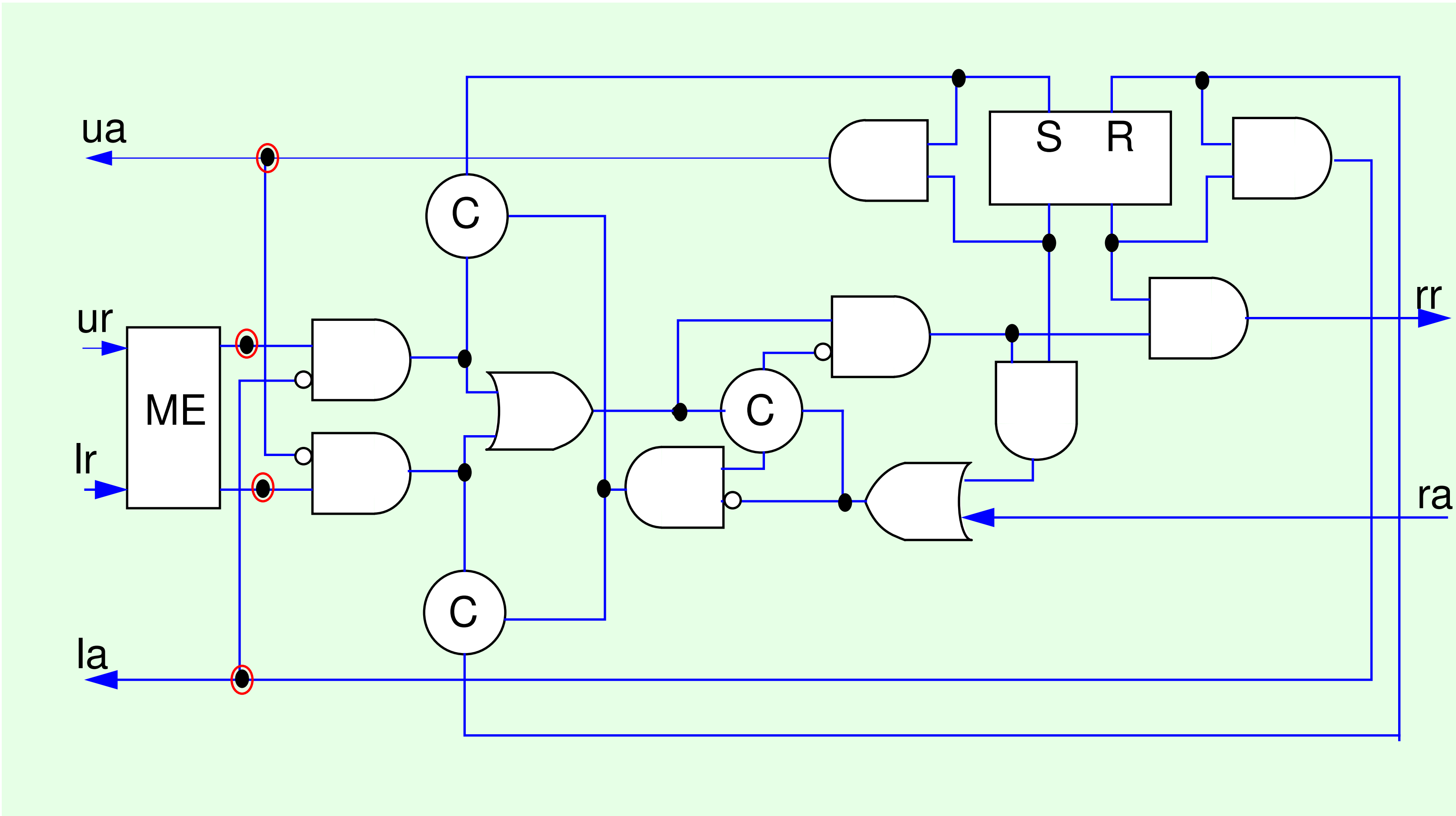
Related this work to

- Previous SI circuit verification work
- Other traditional reduction techniques [see paper]

Future work

- Explore relationship with traditional partial order techniques and application to timed circuits

DME-ring Example (DME Cell)



DME-ring