Efficient Analysis of Pipeline Models for WCET Computation

Stephan Wilhelm (sw@absint.com)
AbsInt GmbH and Saarland University

Outline

1. aiT's pipeline analysis

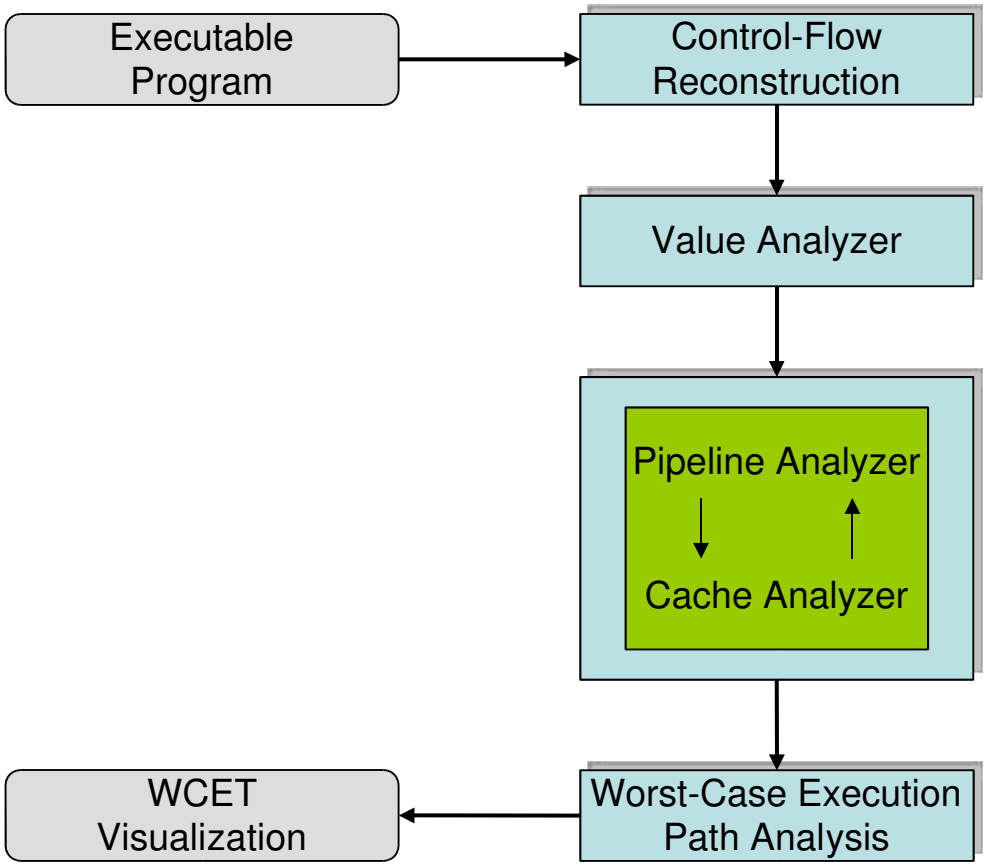
3. Why improve its efficiency?

5. BDD based pipeline analysis

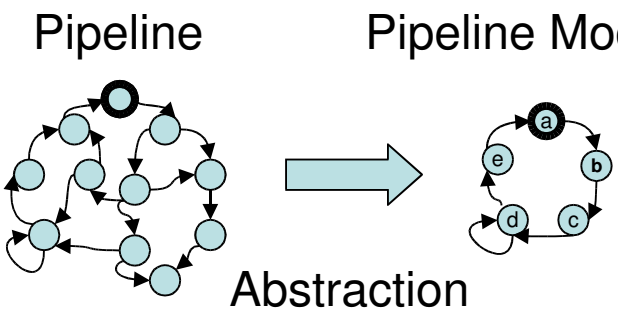
7. Generating pipeline analyzers

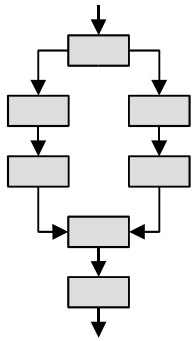
9. Status and prospects

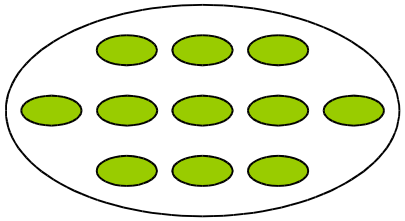
The aiT-Framework

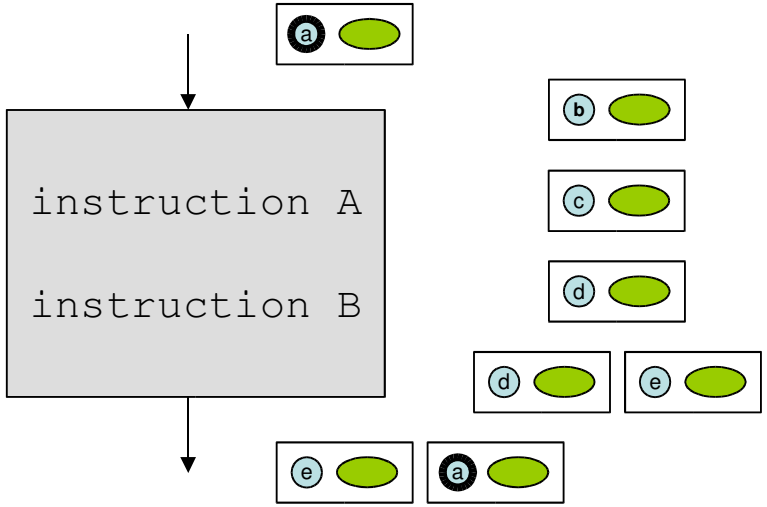


Pipeline Analysis









Timing Anomalies!

State explosion.

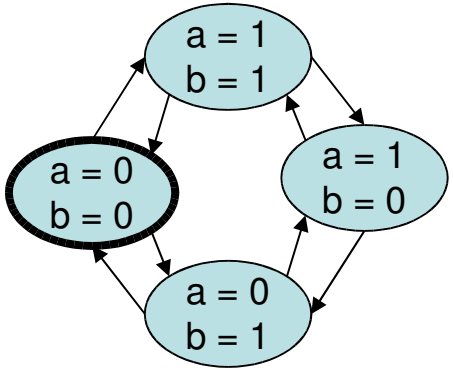
Timing Anomalies

- local worst case vs. global worst case
  - cache hit on global worst-case path (Lundquist/Stenström)
  - “unintuitive behavior”
  - no upper bound
- Caused by features like: caches, out-of-order execution, branch prediction, ...

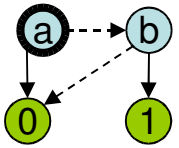
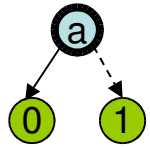
No local worst-case decisions.

State explosion.

Symbolic Representation of FSM State Sets

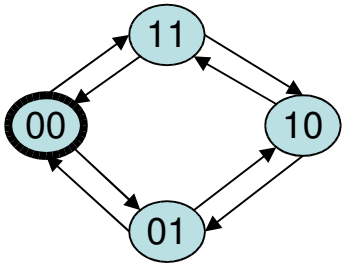


A 2-bit automaton

FSM state sets	characteristic functions	BDD's
01	$\neg a \wedge b$	
01 00	$\neg a$	
01 00 10 11	1	

- BDD size depends on variable ordering
- worst case: exponential
- in most cases better orderings are possible

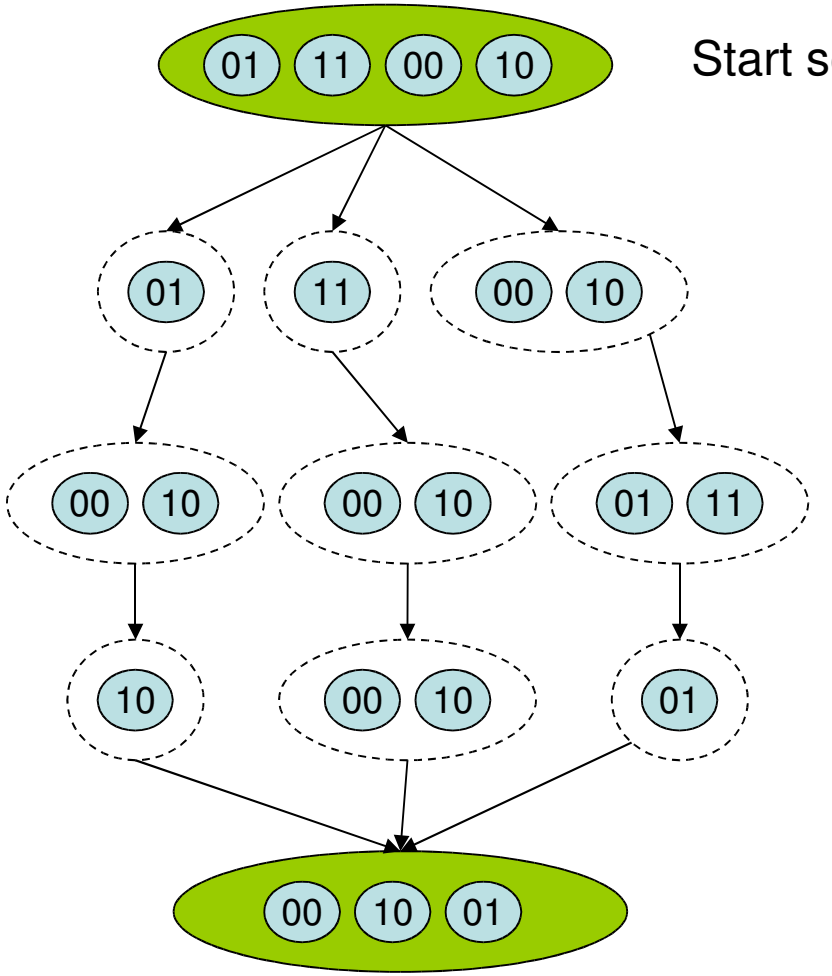
BDD based Pipeline Analysis



A 2-bit automaton

```
00: 01
01: 10
10: 01
```

The program



Start set

1. Partition the set
2. Compute images
3. Apply inputs
4. Compute union

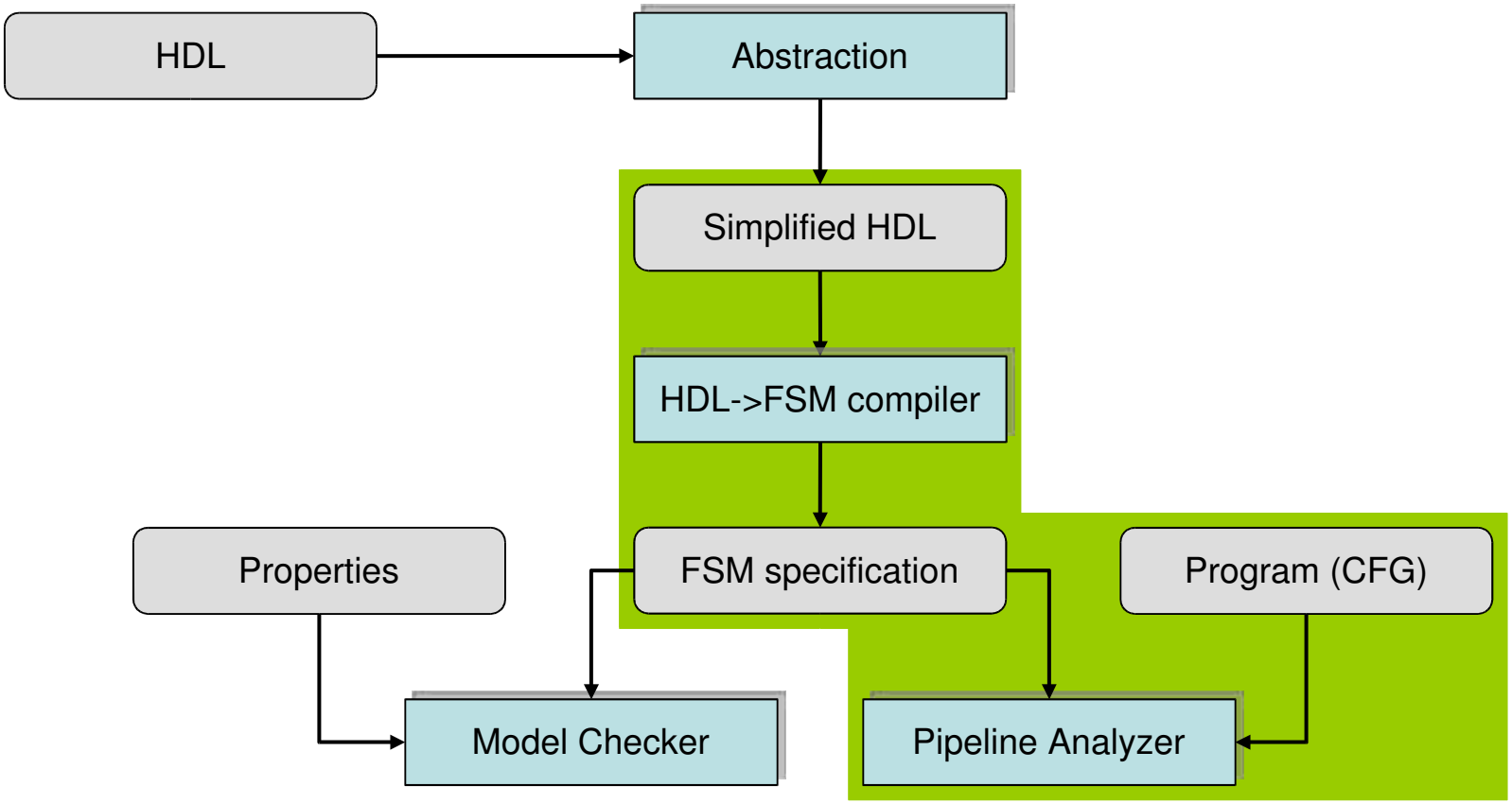
Model Specification & Validation

- Documentation is often incorrect/incomplete.
- Model specification is error-prone.
- Trace validation is difficult.

Generate analysis from HDL specification.

- Verilog and VHDL can be compiled into FSM's (Cheng, 1994).
  - Implemented in VIS model checker.

Generating Pipeline Analyzers



Status and Prospects

1. Implementation of proof-of-concept.
   - For simple examples.
   - Only one basic block.
2. Integration with DFA framework (PAG). ← today
3. Integration with CFG representation.
4. Specification of a real pipeline.
   - Infineon Tricore 2
5. Compare against “classic aiT”.

past

future