Exploiting Craig Interpolants in Unbounded Model Checking of Hardware Designs

FMCAD 2015

September 28th, 2015
Austin (TX), USA

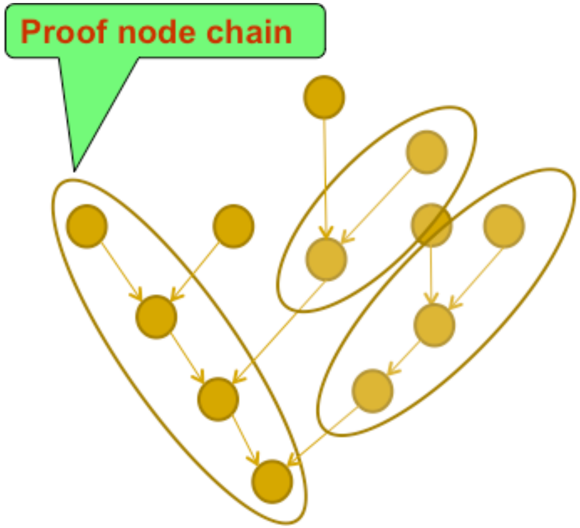
Danilo Vendraminetto

Formal Methods Group, Politecnico di Torino, Italy.

Outline

- Background
  - Verification of hardware designs
  - Craig Interpolants in UMC
- *Contributions*
  - Redundancy removal and reduction in UNSAT proofs and ITPs
    - Heuristic procedure for scalable *ITP compaction*
  - Abstraction and refinement techniques for ITPs
    - Heuristic procedure for abstracting *without resorting to resolution proofs*
- Experimental results & Conclusions

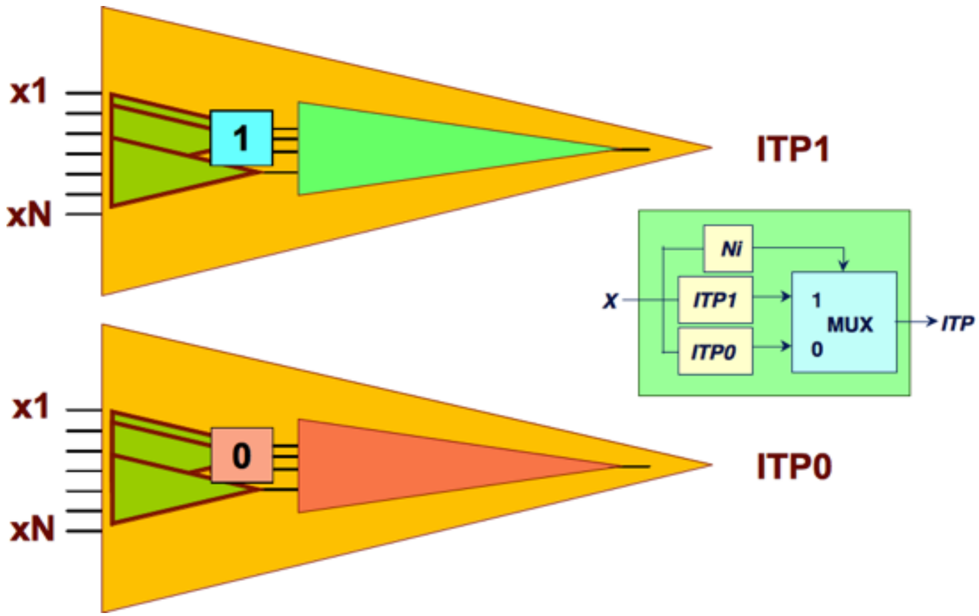
ITP Proof Compaction



- Proof reduction
  - Recycle-pivots [Bar-Inal & al. HVC08]
    - Exploiting proof topology: *proof node chains*
- Logic synthesis manipulations on the proof
  - Constant propagation
  - BDD-based sweeping (for equivalences)
  - Observability Don't Care (lightweight)

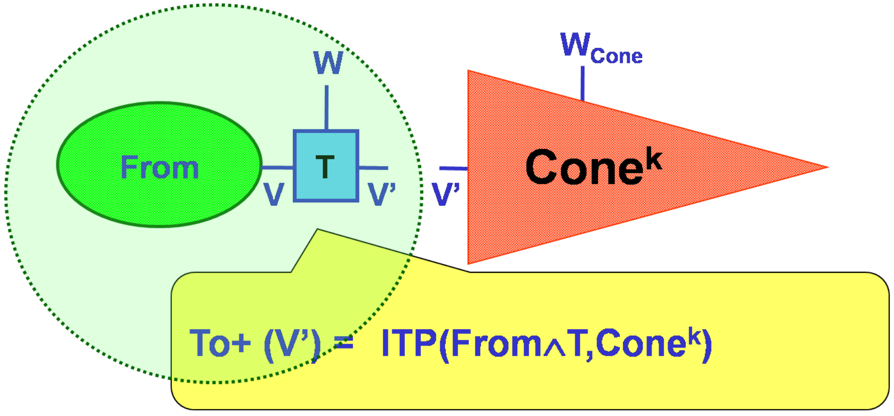
ITP Circuit Compaction

- Proof into AIG
  - ODC (structural)
  - Logic synthesis
    - rewrite / refactor
      - using ABC tool
    - AIG balance
  - ITE-based decomposition
    - iff necessary



Interpolant Abstraction

- ITP+: take on improved Craig's interpolation
- Incremental computation of interpolants using alternative techniques
  - Equivalence classes, mutual implications of state variables
  - Cube-based over-approximation, based on the detection of those state variables that are stuck at constant values



ITP$^+$ - Abst. by iterative refinement

Loop through candidates

Find atomic abstraction

...until adequate

...or return Craig's ITP

$IMG+_{Adq}$ (From, T, Cone^k)

 To+ = Full_state_space

 Foreach Class ∈ Abstraction _classes

  Select abstraction

  $To+_{Class}$ = IMG+ using abstraction

  To+ = To+ ∧ $To+_{Class}$

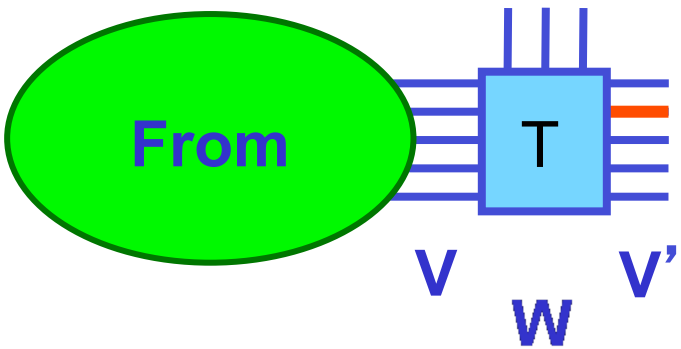
  if UNSAT(To+ ∧ Cone^k) return To+

 return (To+ ∧ ITP(From ∧T, Cone^k))

ITP$^+$ - Abstraction classes

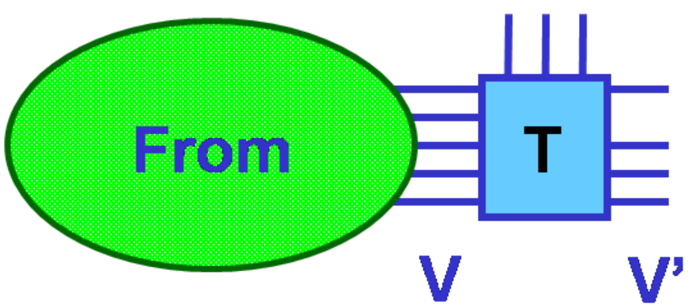
- *Tightening* abstraction classes
  - Equivalent state variables
  - Constant state variables
  - SAT-based enumeration
- *Loosening* abstraction classes
  - Localization abstraction
  - Ternary abstraction

ITP^{+} - Constant state variables



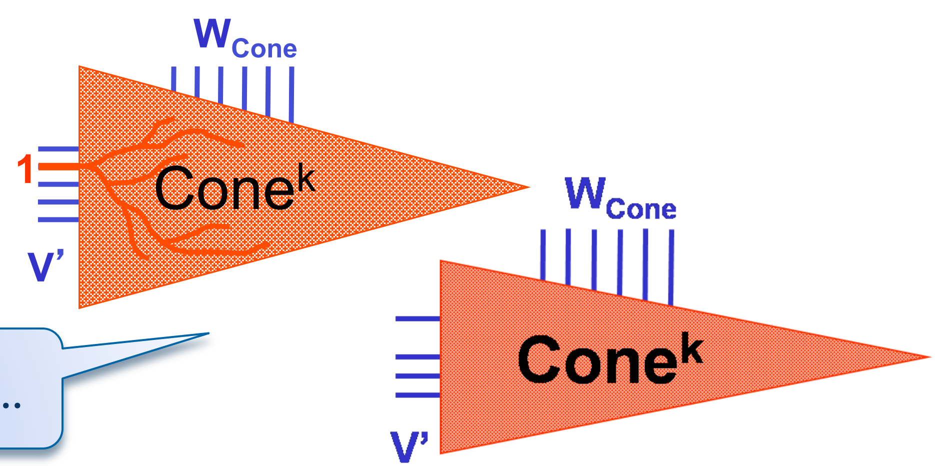
$V_i' = 1$?

If yes, simplify…



Refine: litearal invariant

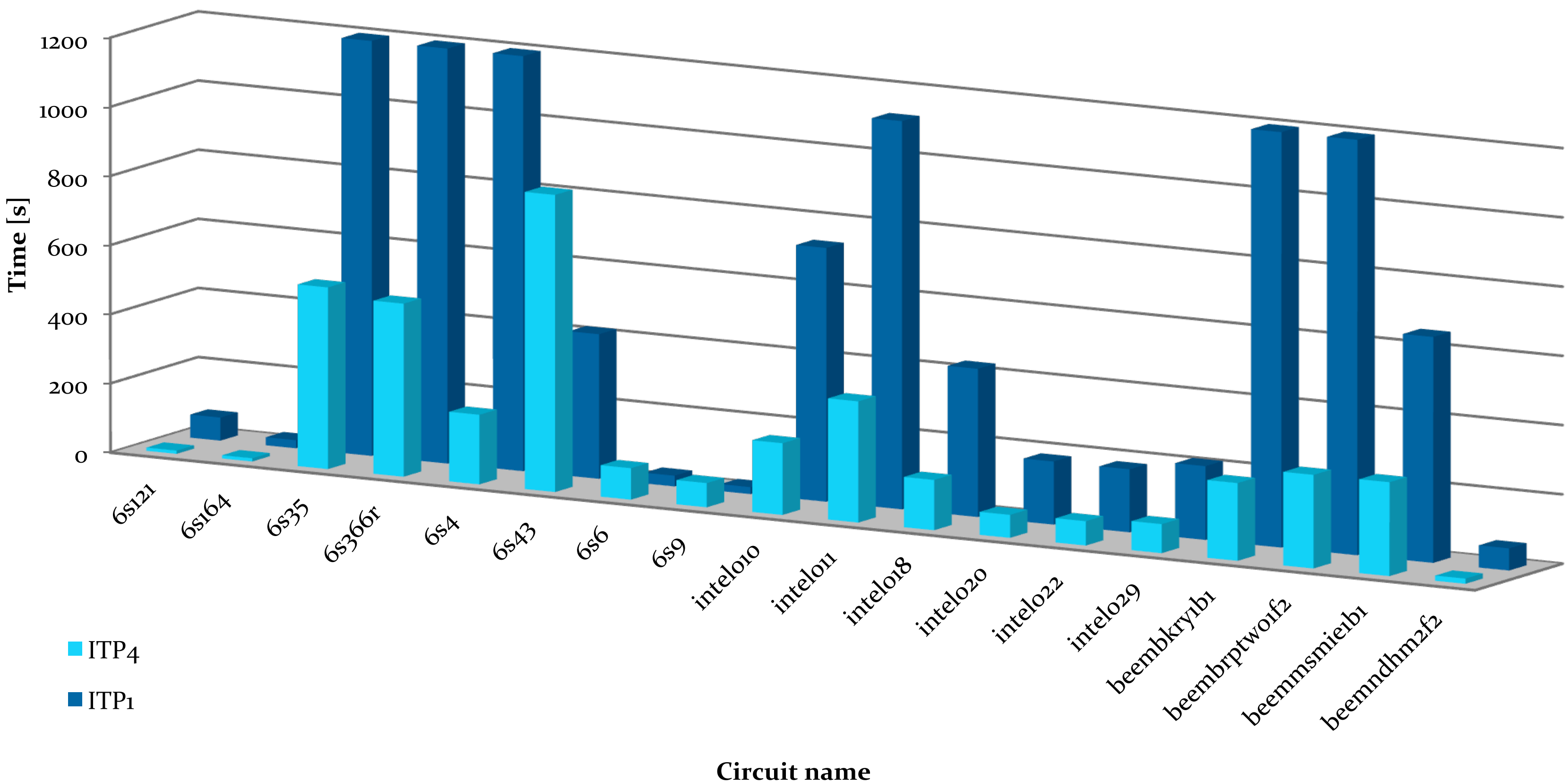
$To+ = To+ \wedge v'_i$



Simplify cone as well (constant prop.)…

Some experimental results…

- Interpolant abstraction

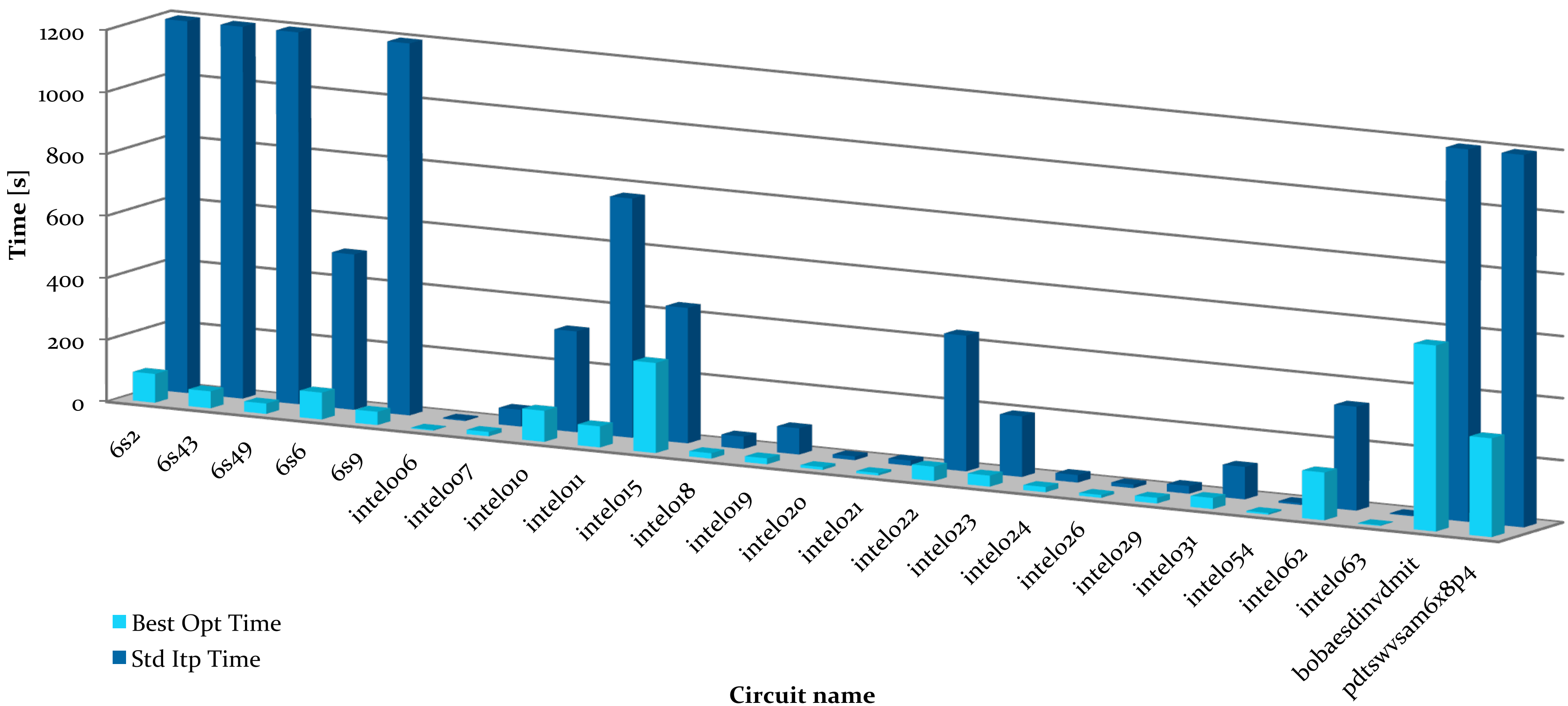


Come to my poster presentation for more details and experimental results

Thanks for your attention

Some experimental results…

- Interpolant compaction



Conclusions

- ITP-based MC heavily relies on scalability
- We developed effective techniques to improve standard ITPs.
  - *Scalable techniques, applied incrementally*
- Best suited as a second engine
  - Hard-to-prove properties (hard for IC3)
  - Explosion of standard interpolation
  - Can afford extra time (for memory)