Inductive Equivalence Checking under Retiming and Resynthesis

J.-H. Roland Jiang and Wei-Lun Hung

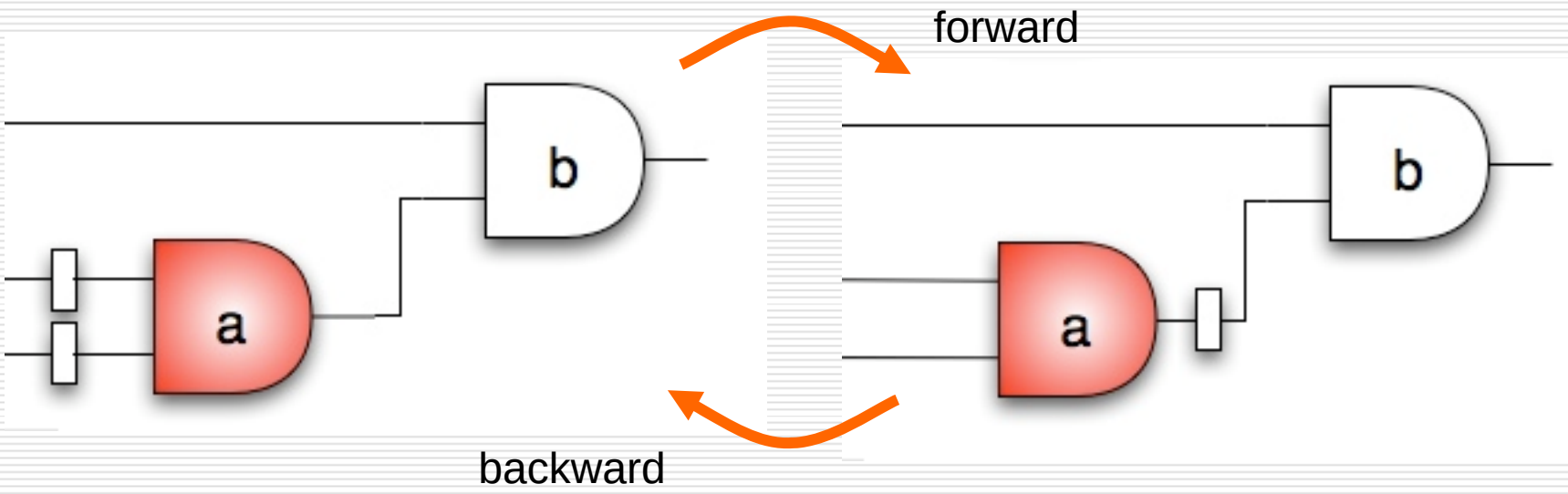
National Taiwan University

Outline

- Introduction and motivation
- Prior work
- Our approaches
- Experimental results
- Conclusions

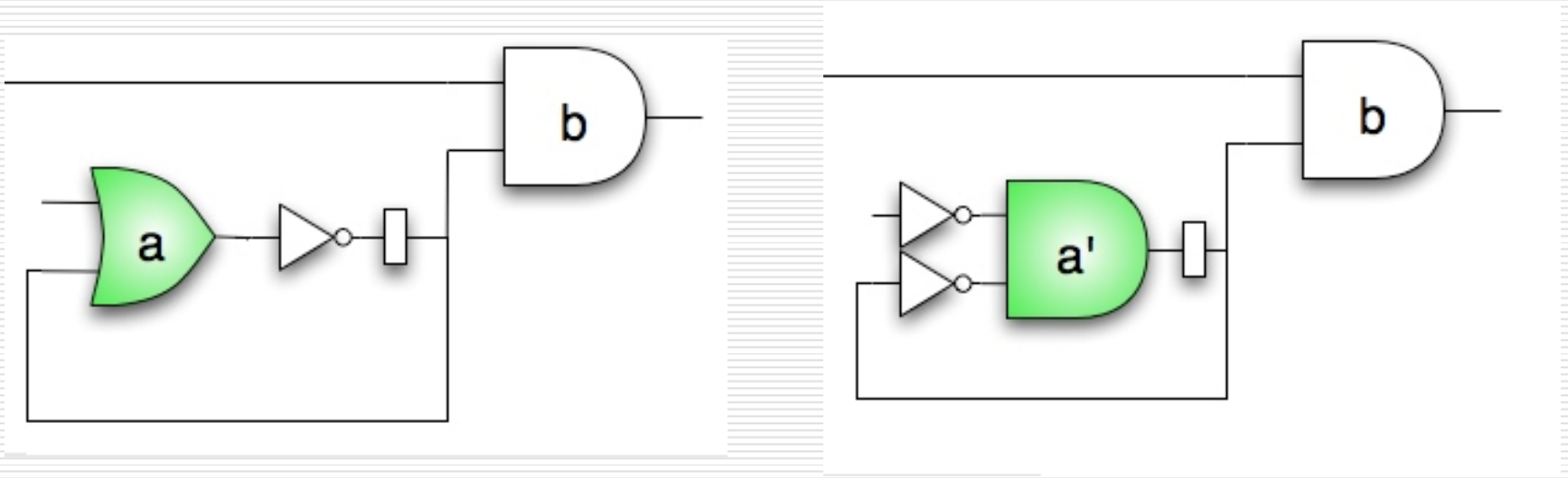
Retiming

- Registers relocated for optimization
  - Transition functions and combinational boundaries changed
  - Circuit structures preserved



Resynthesis

- Structural rewriting for optimization
  - Functionality unchanged



Motivation

- Retiming & resynthesis (T&S) are the most important and practical techniques in sequential circuit optimization
- Not widely used due to its verification complexity
  - The more iterations of retiming and resynthesis are applied, the harder the equivalence can be verified
  - PSPACE-complete for unlimited T&S iterations
- Important to explore effective verification methods under restricted T&S
  - Explore the specialty of retiming & resynthesis

Prior work

- Retiming verification
  - Circuit isomorphism checking [SSB92]
  - Induction
    - Retiming invariant [MS03] (w/ known matching)
    - Inductive signal correspondence [vE00,BS98] (w/o known matching)

  Only work for retiming verification, but not retiming combined with resynthesis

- SAT-based safety property checking
  - Temporal induction [BC00,SSS00,ES03]
  - Reduction techniques [MBPK05,LC06]

  Specialty of T&S equivalence checking not exploited

Inductive signal correspondence

- A *signal correspondence*, denoted as $\approx$, of a circuit *C* with transition function δ and initial states I is an equivalence relation over signals of *C* such that every signal pair $(f_i, f_j) \in \approx$, i.e., $\approx \Rightarrow (f_i \equiv f_j)$, satisfies

  init-state condition:

  $$I(s) \Rightarrow (f_i(x, s) \equiv f_j(x, s))$$

  next-state condition:

  $$\approx(x, s) \Rightarrow (f_i(x', \delta(x, s)) \equiv f_j(x', \delta(x, s)))$$

  for all input vars x, x' and state vars s

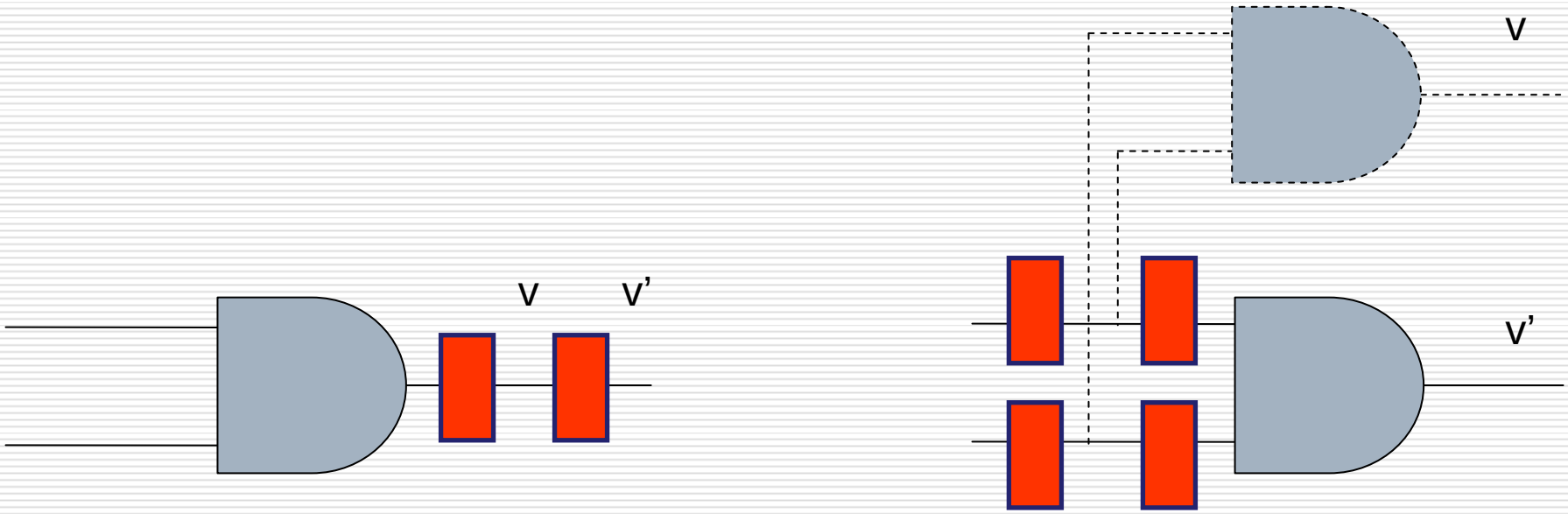
- Two circuits are equivalent if $\approx(x, s) \Rightarrow \lambda_{\times}(x, s)$, where $\lambda_{\times}$ is the output function of the product machine

Inductive signal correspondence

- Complete for retiming equivalence checking
  - virtual forward retiming [vE00] is needed
- Incomplete for retiming+resynthesis equivalence checking
  - virtual backward retiming is needed
    - Unavailable however due to initial-state consistency problem
- Inductive signal correspondence can be made compete for up to retiming+resynthesis+retiming

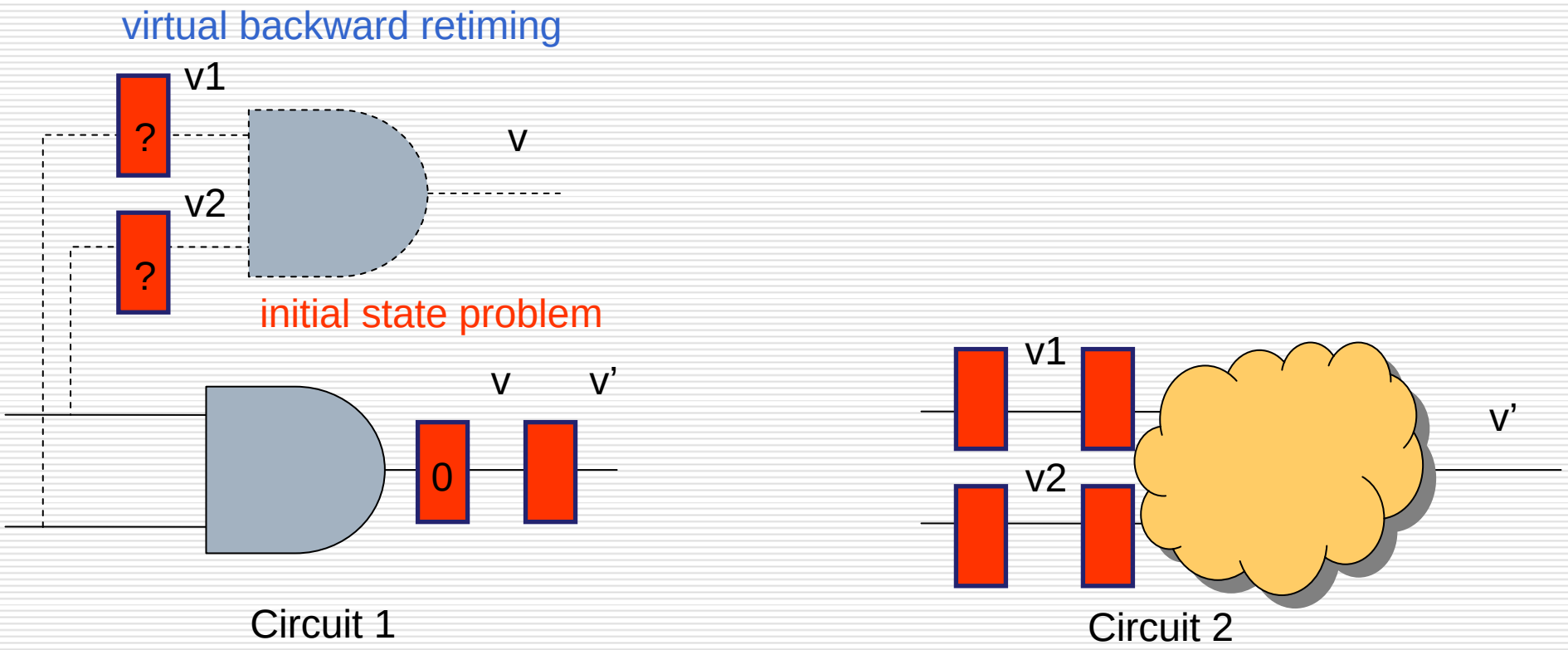
Virtual retiming

virtual forward retiming



Virtual forward retiming bridges missing the links between transition updates

Virtual retiming



Virtual backward retiming is needed if resynthesis is involved

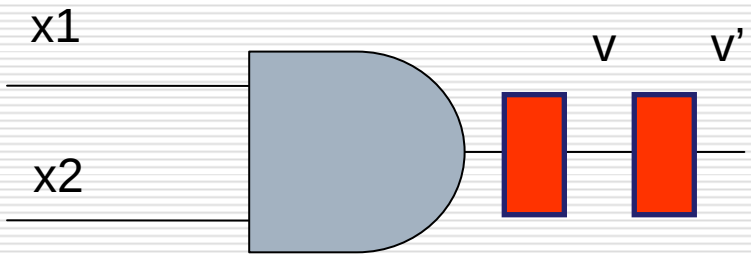
Enabling techniques

- In theory
  - Characterize minimal set of equivalent signal pairs for complete equivalence checking
  - Perform timeframe expansion to overcome the nonexistence of virtual backward retiming problem
    - More systematic than virtual retiming
  - Identify signal correspondence among different timeframes

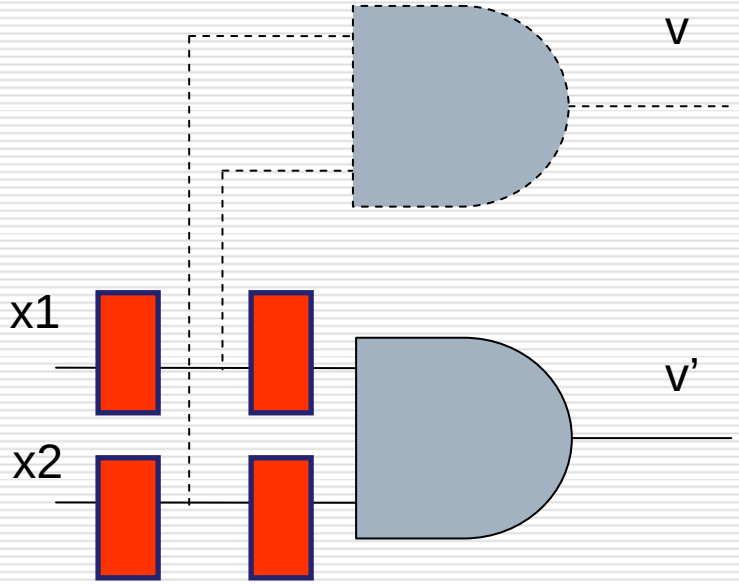
- In practice
  - SAT-based solving
  - Signal pair reduction
  - Multi-timeframe and intelligent simulation

Minimal set of equivalent signal pairs

- Inductive equivalence checking is complete if the equivalences of transition updates continue



Circuit 1



Circuit 2

Relation {(x1,x1),(x2,x2),(v',v')} does not continue

Relation {(x1,x1),(x2,x2),(v,v),(v',v')} continues

Timeframe expansion

- Any signal can be expressed in terms of timed Boolean formula over cut signals (feedback vertex set in cyclic circuit)

- If signal pairs $(f_i^1, f_j^2) \in \approx$ form cuts in the two circuits under verification, then $\approx$ yields continuous transition updates after long-enough timeframe expansion
  - Complete for equivalence checking
  - #Timeframes is up-bounded by the register depth (max #registers along any path in the induced circuit after removing cut signals)

Timeframe expansion

- k-step signal correspondence $\approx_k$

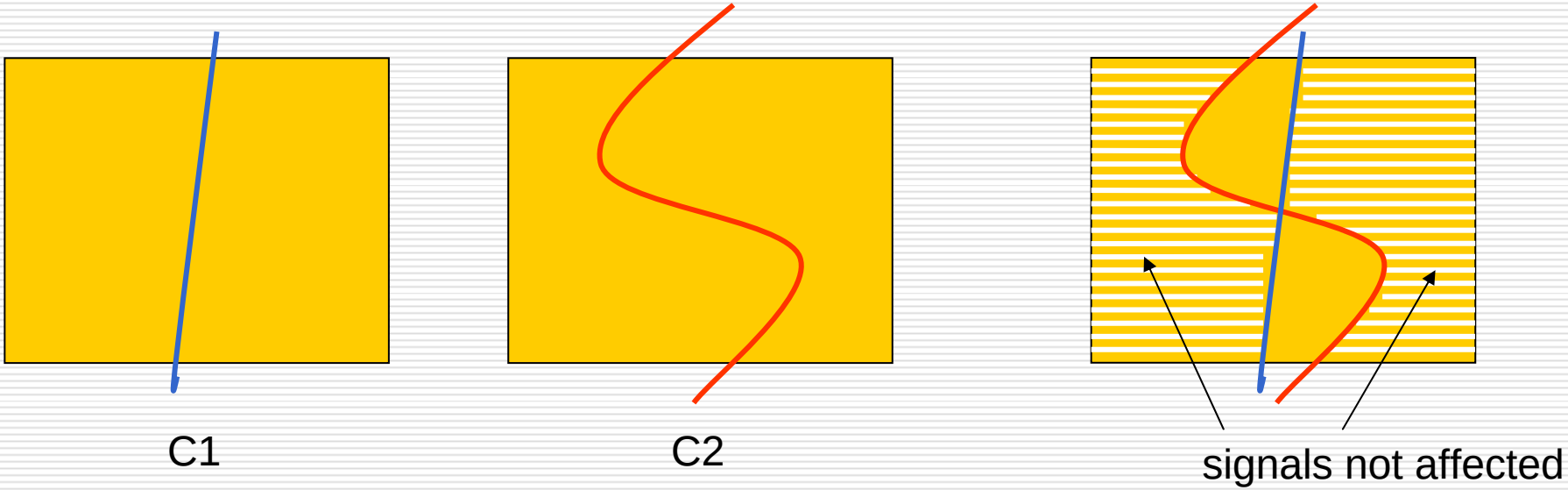
- Let $f^k(x^k,\ldots, x^1,s) = f(x^k, \delta(x^{k-1},\ldots, \delta(x^1,s)))$

  $I(s) \Rightarrow \wedge_{t=1}^{k} (f_i^t(x,s) \equiv f_j^t(x,s))$ and
  $\wedge_{t=1}^{k} \approx_k^t (x.s) \Rightarrow (f_i^{k+1}(x', \delta(x,s)), f_j^{k+1}(x', \delta(x,s)))$

  for all input vars x, x' and state vars s

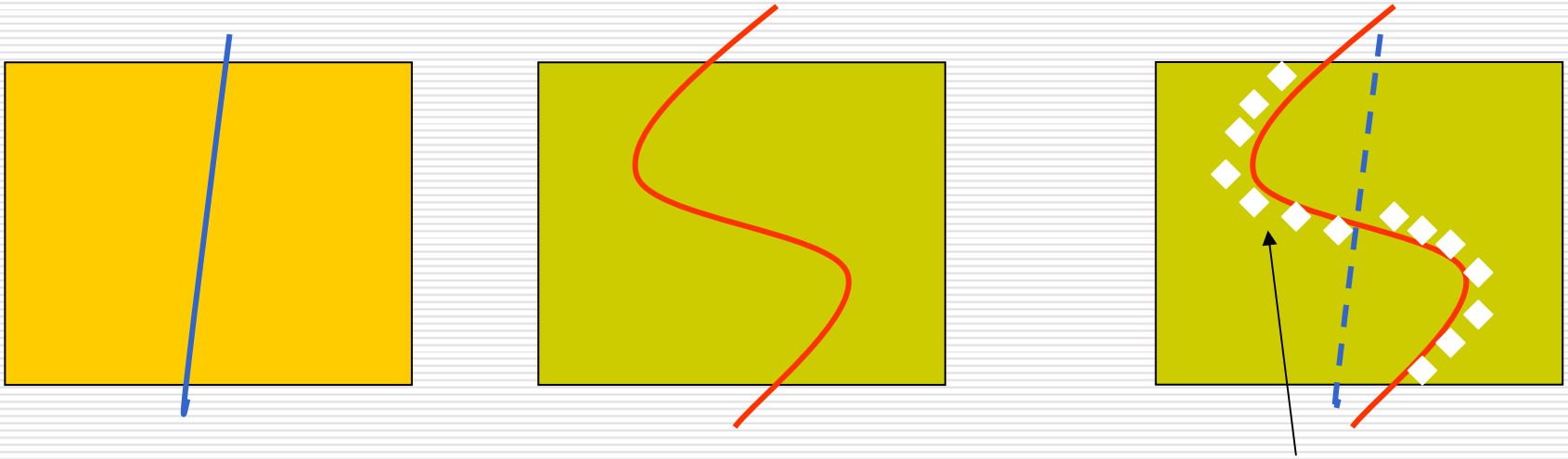
Equivalence checking for T

- Timeframe expansion gives a simple and systematic replacement of virtual forward retiming
  - It makes inductive signal correspondence complete for retiming (T) equivalence checking

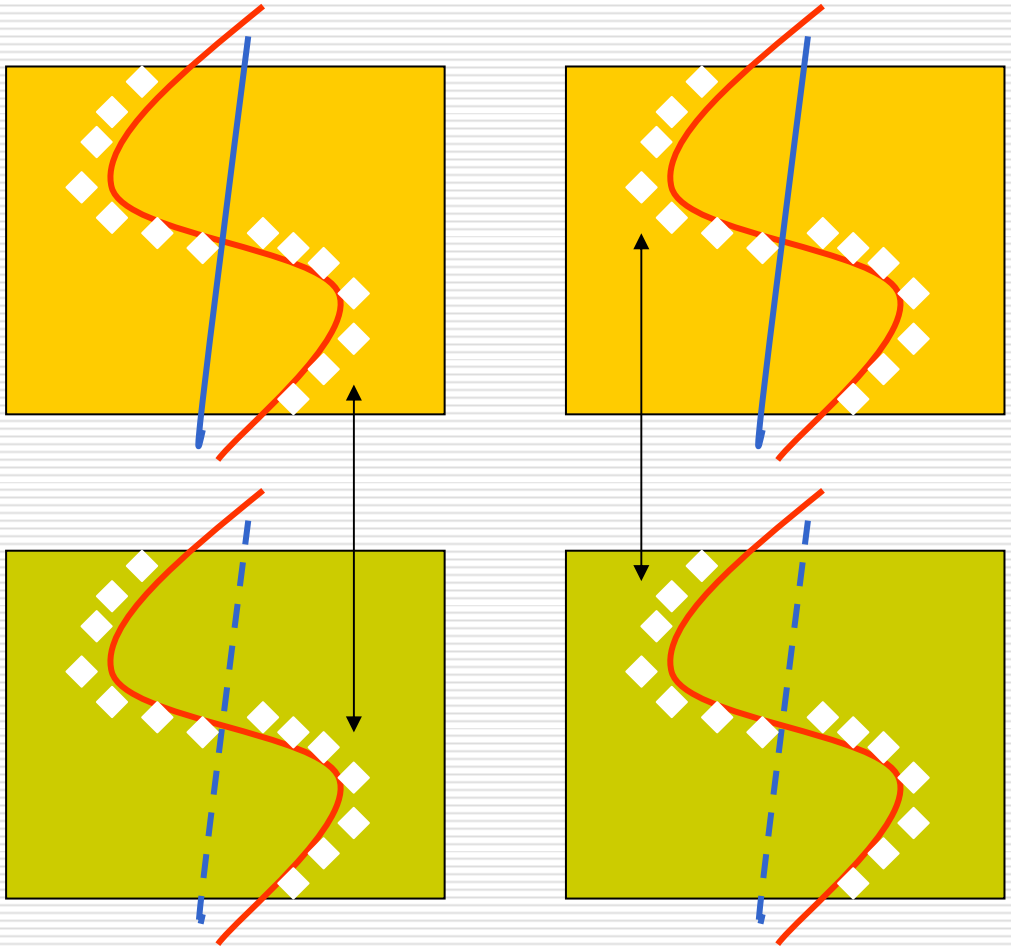


Equivalence checking for TS / ST

- Timeframe expansion bridges missing links in transition updates
  - It makes inductive signal correspondence complete for retiming+resynthesis (TS) and resynthesis+retiming (ST) equivalence checking



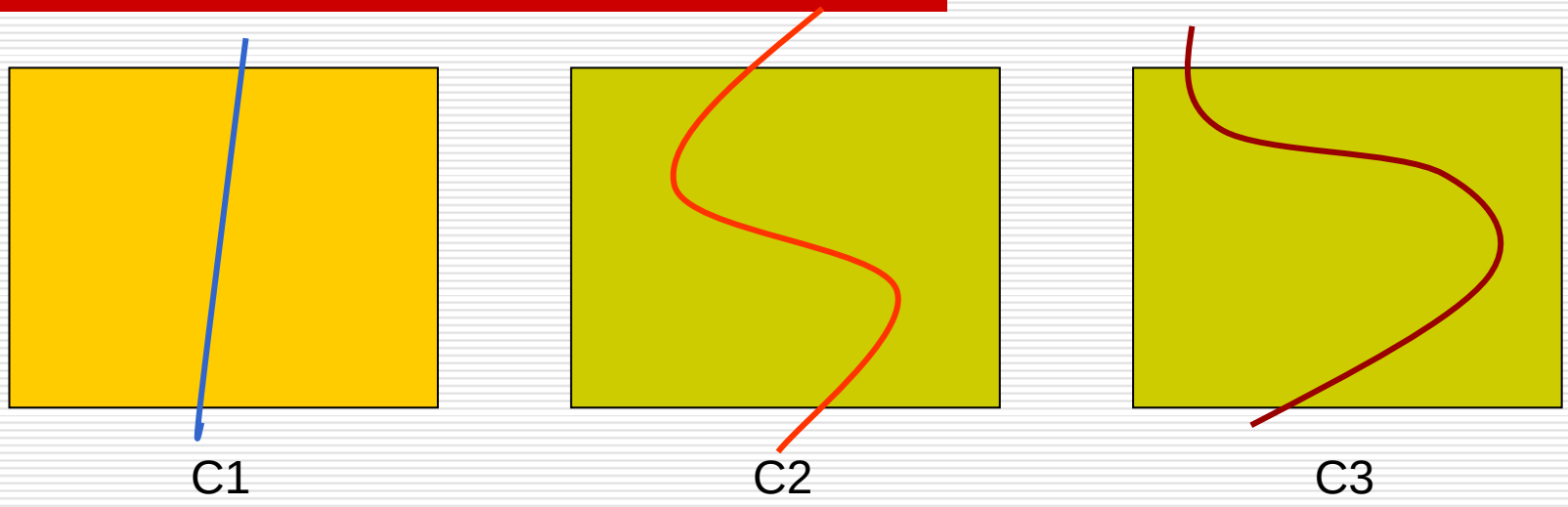
Equivalence checking for TS / ST

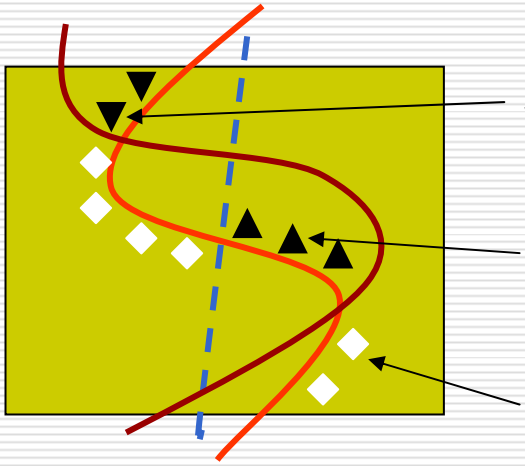


Still a cut with variables in two timeframes

Only need to explore signal corr. in the same timeframe

Equivalence checking for TST





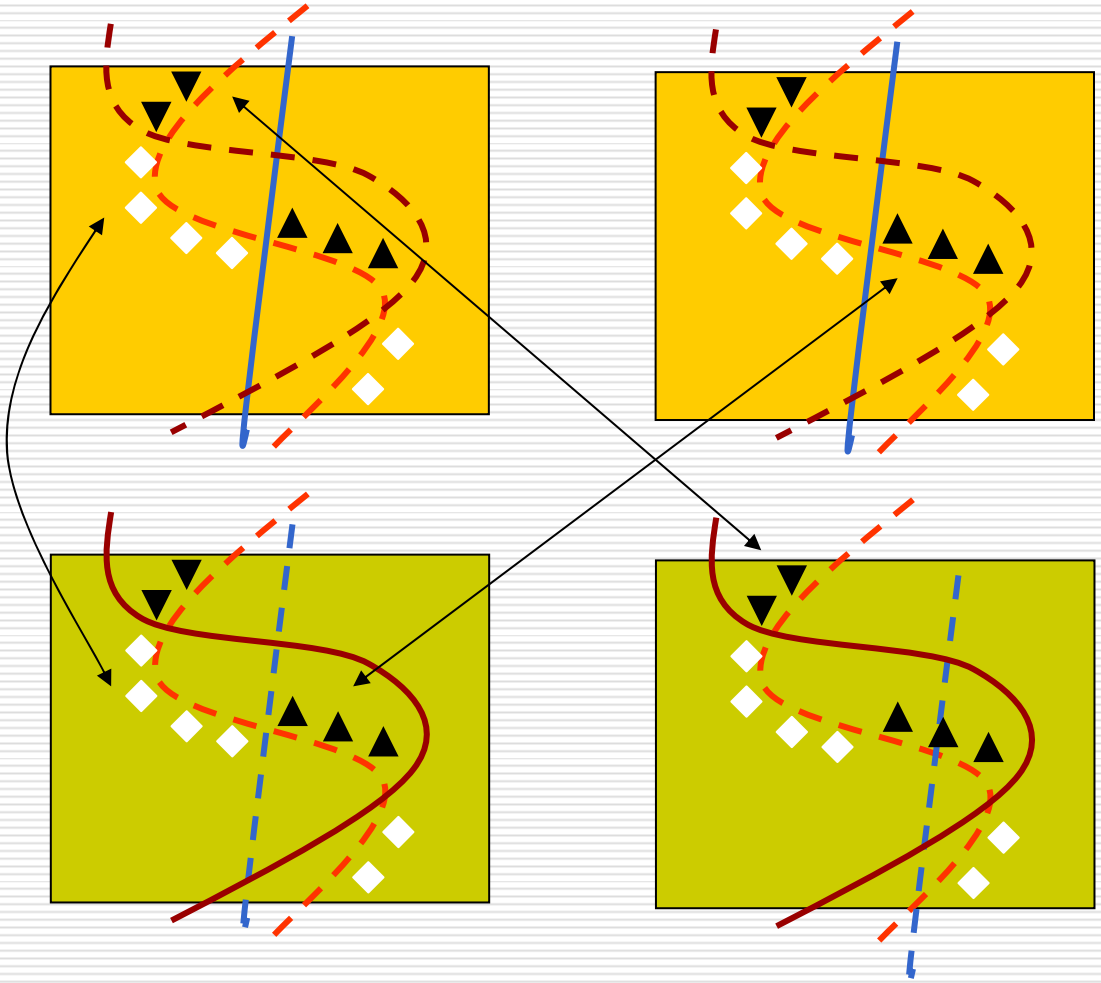
signals delayed 1-cycle

signals advanced 1-cycle

signals not affected

(still a cut in a 3-frame time span)

Equivalence checking for TST



Still a cut with variables in three timeframes

Need to explore signal corr. across different timeframes

TF expansion vs. SC across multi-TF

- Timeframe expansion
  - To make transition update continuous
- SC across two adjacent timeframes
  - To make TST checking complete

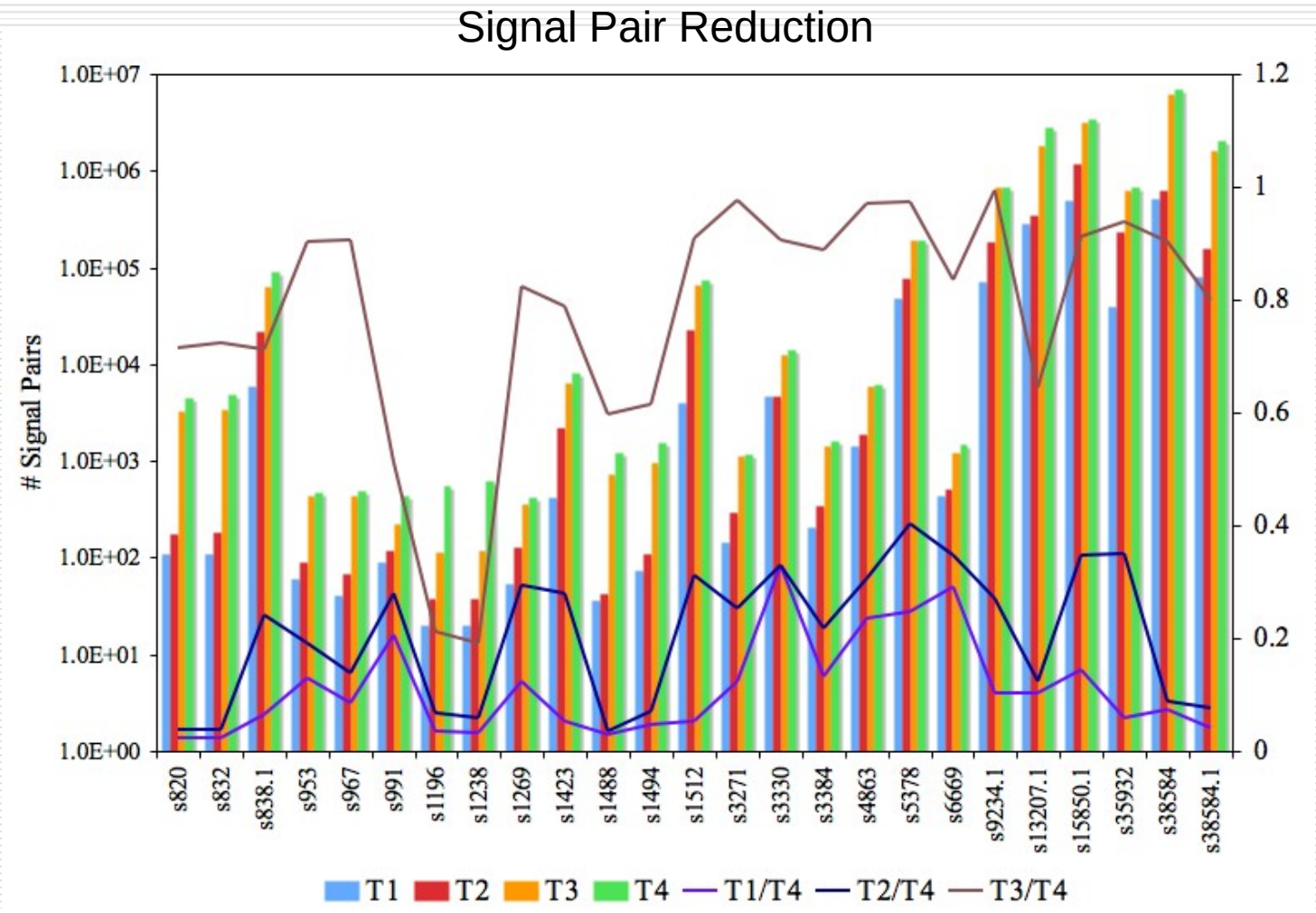
Signal pair reduction

- T
  - Output boundary vs. forward retime region
  - Input-output boundary vs. forward retime region
- TS
  - Input-output boundary vs. retime region
- TST
  - Retime region vs. retime region

Experimental Results

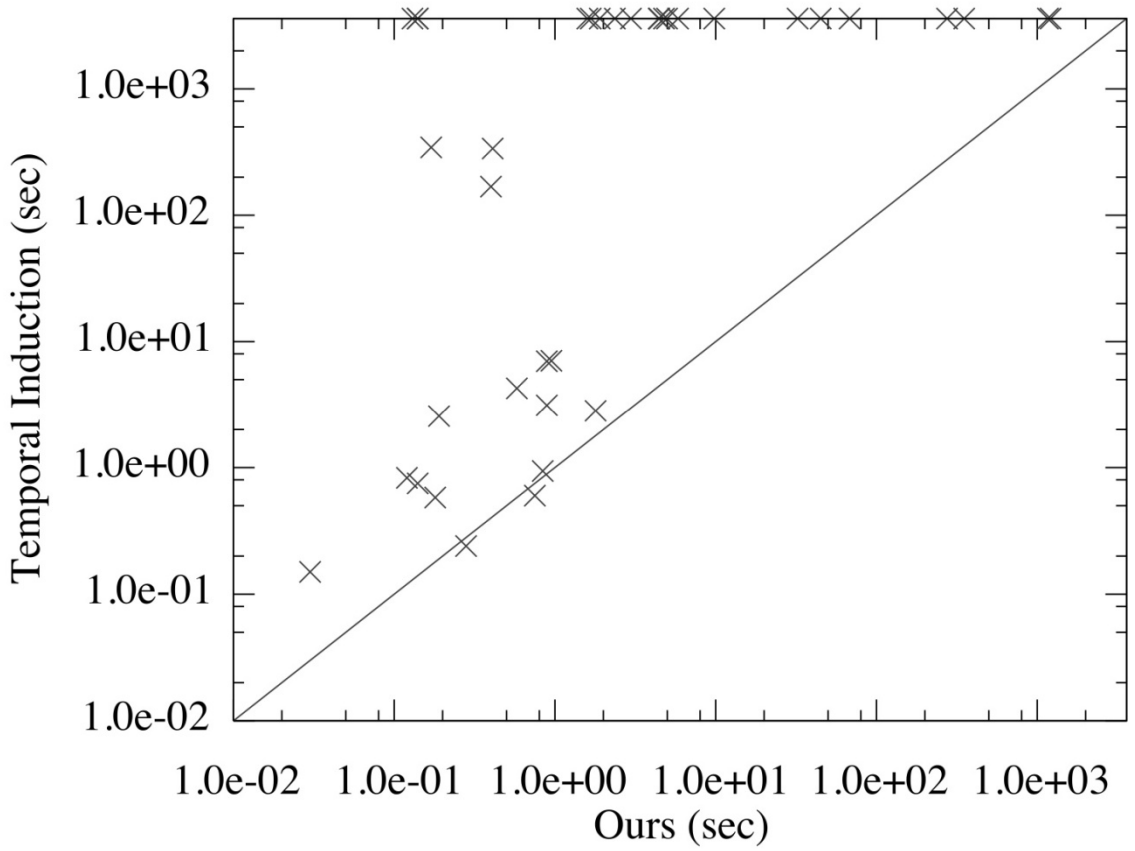
Circuit	reg #	vE00	T		T + S		T + S + T	
			Mem.	Time	Mem.	Time	Mem.	Time
s1423	187	676	15.0	2.22	14.3	5.44	14.6	19.04
s1512	247	34.7	14.4	5.61	14.7	33.45	16.1	261.89
s3271	351	808	15.7	3.55	15.7	4.47	16.2	11.82
s3330	438	1316	15.8	4.83	15.7	5.15	15.3	5.22
s3384	339	-	17.3	110.18	17.1	180.15	16.4	50.89
s4863	467	55.6	16.7	11.8	16.2	12.58	17.4	25.02
s6669	970	-	22.4	56.89	23.2	251.86	20.8	124.39
s9234.1	670	-	19.8	722.87	25.3	653.37	34.3	3482.47
s13207.1	1265	-	26.1	10190	27.4	16853	-	>10hr
s35932	3754	-	39.7	1707	41.6	17393	-	>10hr
s38584	2909	-	49.9	2807	49.3	5339	-	>10hr
s38584.1	2853	-	43.9	6692	40.5	7919	-	>10hr

Experimental Results (cont'd)



Experimental Results (cont'd)

Ours vs. Temporal Induction



Conclusions and future work

- Complete equivalence checking for up to retime+resynthesis+retime
- Signal pair reduction extends the capability and capacity of equivalence checking for retiming & resynthesis
- STS equivalence checking?