Qualifying Relative Timing Constraints for Asynchronous Circuits

Jotham Vaddaboina Manoranjan Kenneth S. Stevens
University of Utah
Salt Lake City

International Symposium on Asynchronous Circuits and Systems

May, 2016

Abstract Statement

Employ signal path analysis to rank sets of relative timing constraints.
Given a Solution Set $set0, set1, ..., setn$ for a Circuit ckt_i
$set0 = rtc_0^0, rtc_0^1, ..., rtc_0^n$
$set1 = rtc_1^0, rtc_1^1, ..., rtc_1^n$
... $setn = rtc_n^0, rtc_n^1, ..., rtc_n^n$
Find the best solution set!

"Allow me to re-introduce myself" - Relative Timing

- Relative timing is a formalism that explicitly represents timing requirements
- RT constraints are used to:
  - make hazards unreachable
  - guarantee functional correctness of circuit
- Modular design capability
- Successfully applied to ASIC and FPGA based designs
- Process generation of advantage

Formalism

$pod \mapsto poc_0 \prec poc_1$

The above constraints specifies that after the point-of-divergence pod, point-of-convergence poc_0 must occur before poc_1.

maximum delay (pod to poc_0) $<$ minimum delay (pod to poc_1)

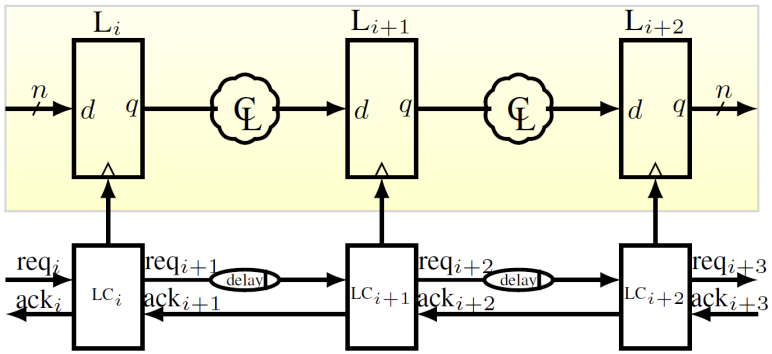


Figure : Implementation with Delays

Applying the formalism

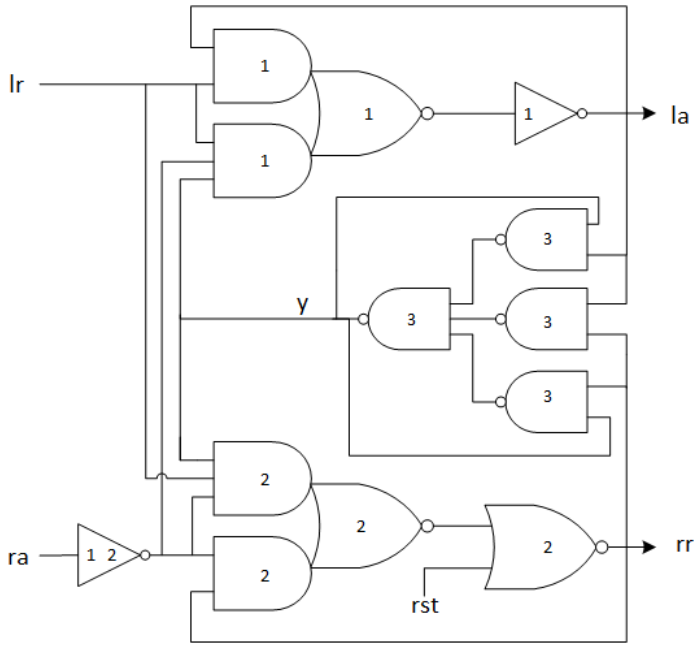


Figure : A Burst-mode controller

$$
\begin{aligned}
\text{LEFTI} &= \text{rst.LEFT} \\
\text{LEFT} &= \text{lr.c1.'la. c2.lr.'la.LEFT} \\
\text{RIGHT} &= \text{'c1.'rr.'c2.ra.'rr.ra.RIGHT} \\
\text{SPEC} &= (\text{LEFTI} \mid \text{LEFT} \mid \text{RIGHT}) \setminus \{ \text{c1,c2} \}
\end{aligned}
$$

Figure : CCS specification of the burst-mode controller

Circuit and Model

- Speed Independent vs Delay Insensitive
  - The circuit analysis employs delay insensitive models
  - Most accurate models
- ASICs vs FPGAs
  - Profound Impact on FPGA
  - FPGAs cannot employ Speed Independent models
- Signal delays are comparable to the logic delays on an FPGA

Controller Implementation on an FPGA

An FPGA based controller implementation is utilized to present the methodology

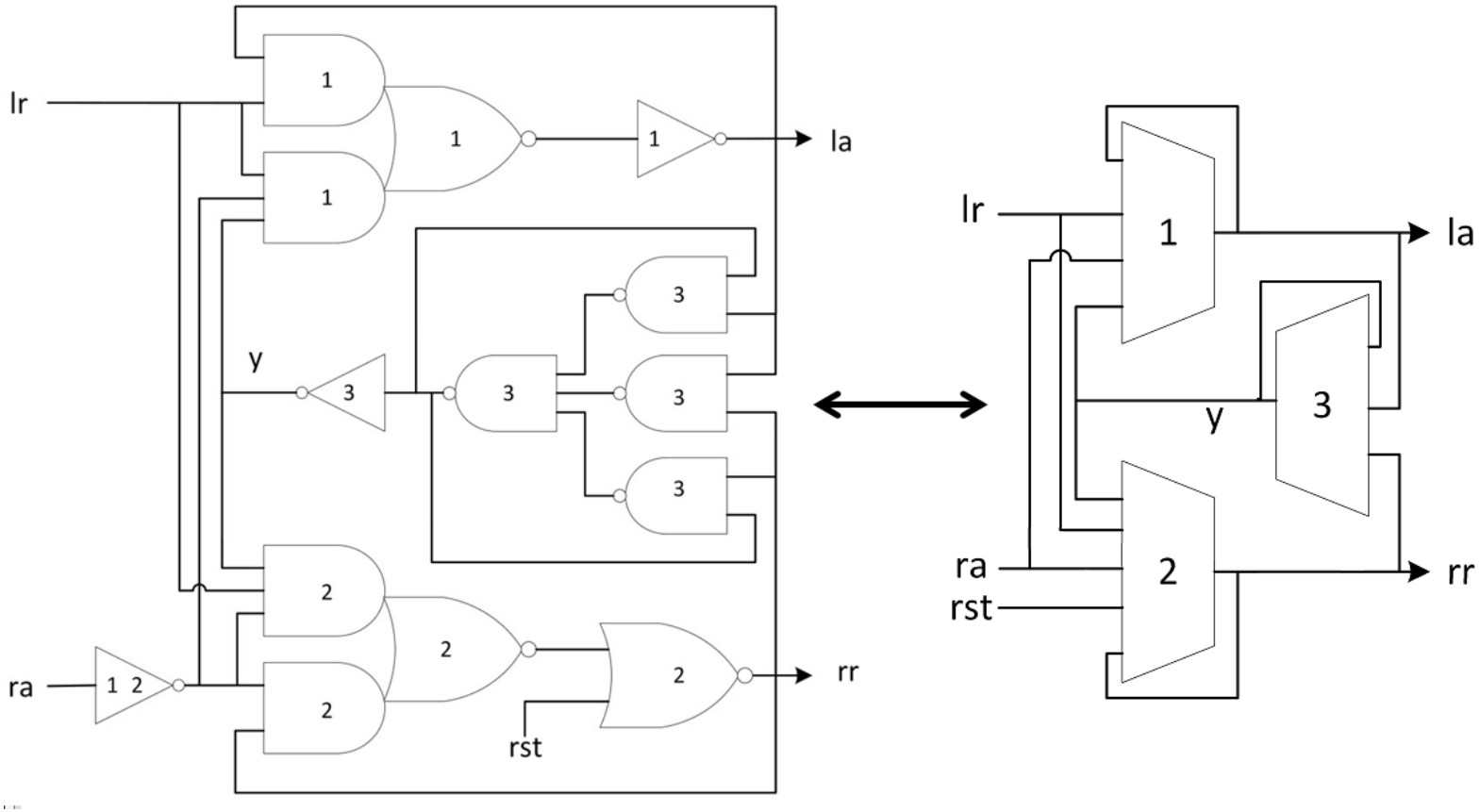


Figure : Look-up-table based controller implementation

Fully Expressing the Controller

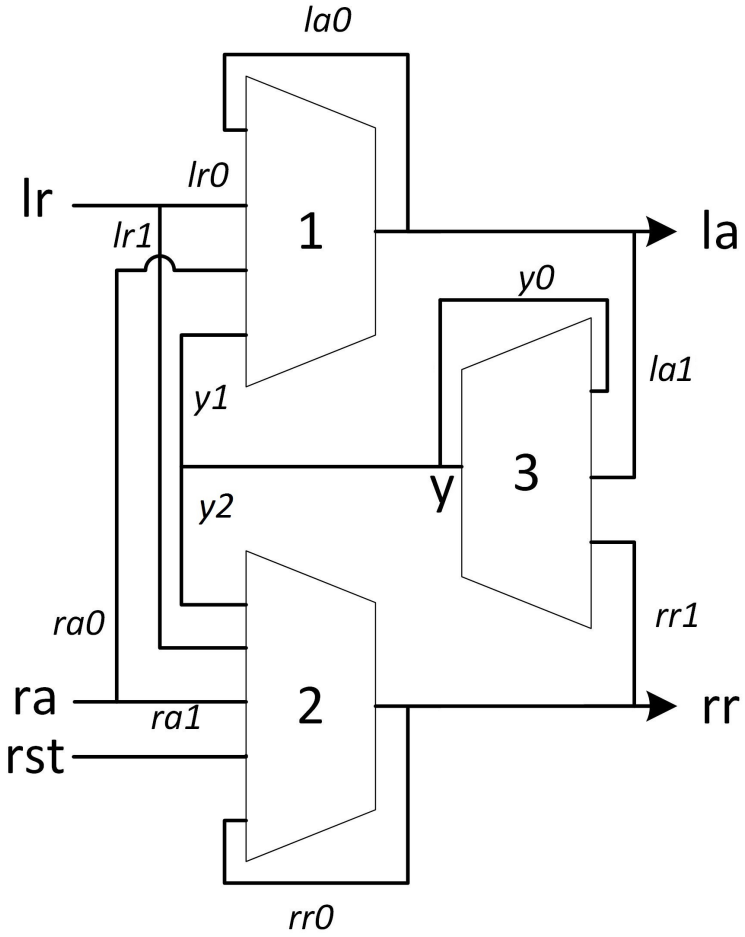


Figure : Controller with modeled forks

Given the circuit, what does and RT constraint look like?

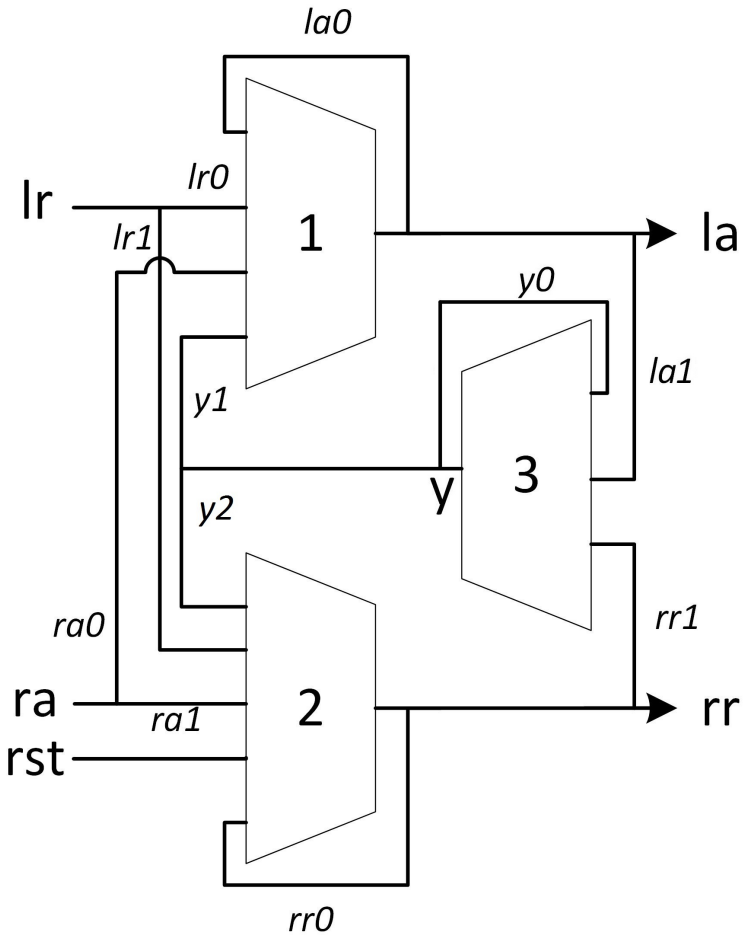


Figure : Implementation with Delays

$$lr\uparrow \ \mapsto \ rr0\uparrow + m \ \prec y2\downarrow \tag{1}$$

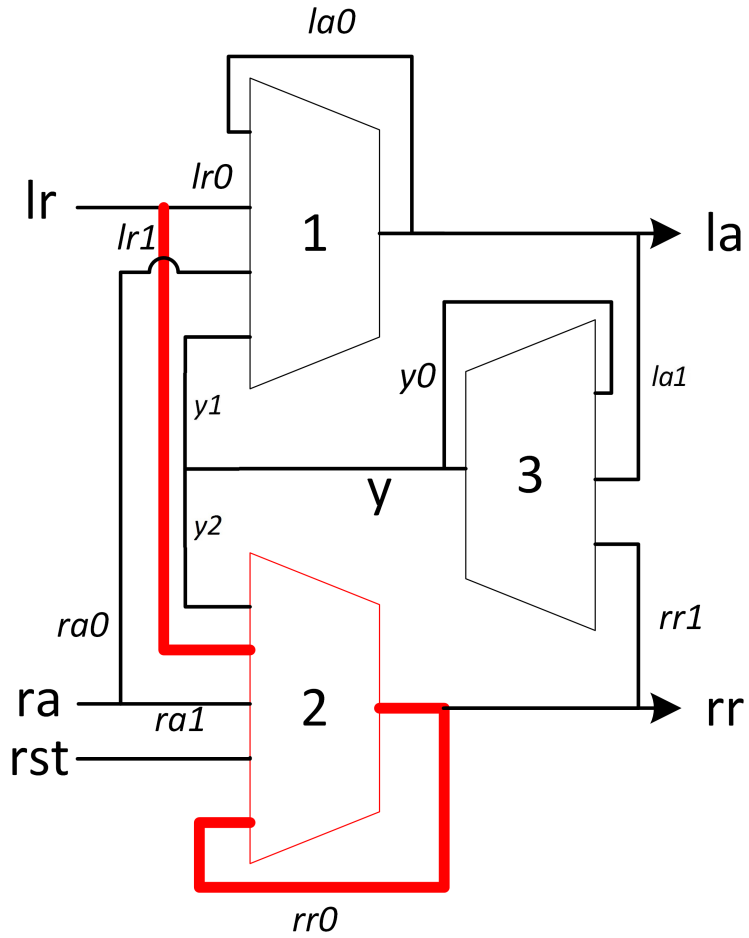


Figure : Max Constraint Path

$$lr\uparrow \mapsto rr0\uparrow + m \prec y2\downarrow \quad \text{max-path:} \quad lr\uparrow \mapsto rr0\uparrow \tag{2}$$

This is the early arrival path

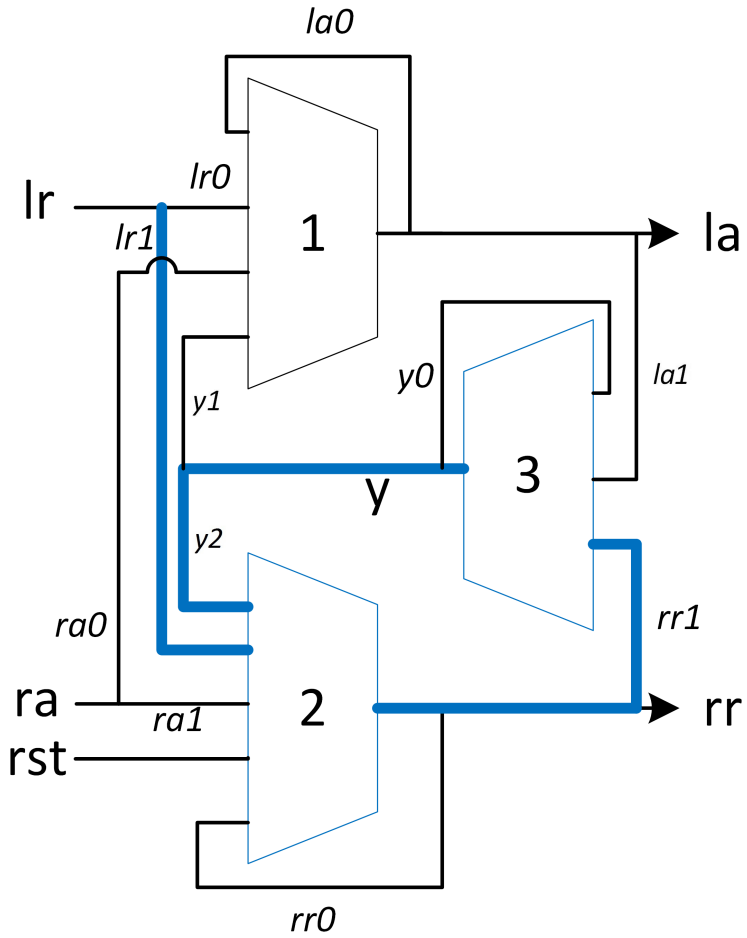


Figure : Min Constraint Path

$$lr\uparrow \;\mapsto\; rr0\uparrow + m \;\prec y2\downarrow;; \quad \text{min-path:} \quad lr\uparrow \;\mapsto\; y2\downarrow \tag{3}$$

This is the late arrival path

That was just one constraint!

RTC0: $pod0 \mapsto poc0_0 \prec poc0_1$

That's more like it!

RTC0: $pod0 \mapsto poc0_0 \prec poc0_1$
RTC1: $pod1 \mapsto poc1_0 \prec poc1_1$
RTC2: $pod2 \mapsto poc2_0 \prec poc2_1$
RTC3: $pod3 \mapsto poc3_0 \prec poc3_1$
RTC4: $pod4 \mapsto poc4_0 \prec poc4_1$
.....
.....
RTCn: $podn \mapsto pocn_0 \prec pocn_1$

Each circuit usually require multiple RT constraints to be satisfied. Together these are called a Set of RT constraints or an RTC set.

- Each circuit can have more than one RTC set associated with it
- Each of these sets, when faithfully implemented guarantee functional correctness
- These sets are heuristically created using ARTIST
  - Given a circuit implementation and a formal design specification, ARTIST automatically generates sets of relative timing constraints
  - Constraints are heuristically selected based on a set of internal rules
  - A large set of constraints can be found based on the search algorithm and heuristics employed

Set0 Set1
Set 2 Set3
.....
Setm Setn

Set0 Set1
Set 2 Set3
.....
Setm Setn

ARE ALL RELATIVE TIMING CONSTRAINT SETS BORN EQUAL?

ARE ALL RELATIVE TIMING CONSTRAINT SETS BORN EQUAL?

No!

- Circuits usually have more than one possible constraint sets - numerous "easy to implement constraints" would be preferable over a set with a few "hard to implement constraints"
- Each sets can have many constraints that interact with each other - conflicting timing requirements
- Constraint sets impact circuit performance - delays may be needed to force conformance

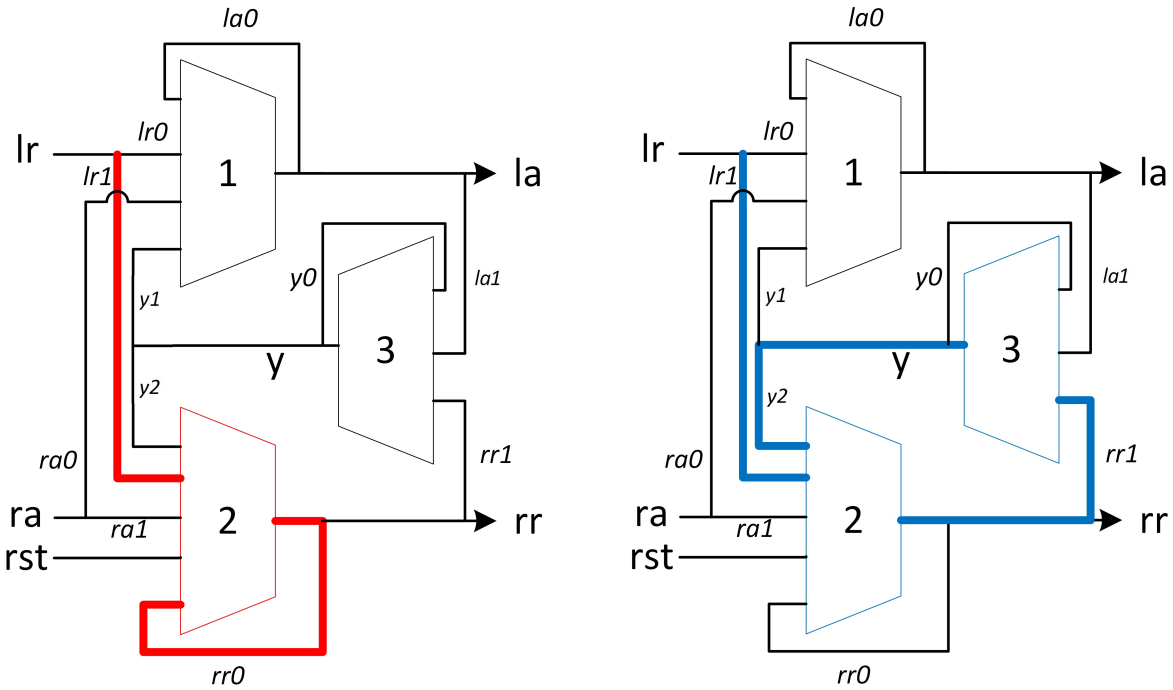
Qualitative analysis of constraint sets can enable better circuit design choices

Key parameters:

- Robustness - Larger margin = better constraint! - Shorter early path and longer late path
- Timing conflicts - Two sided timing constraints
  - RTC0: $pod \mapsto poc_0 \prec poc_1$
  - RTC1: $pod \mapsto poc_1 \prec poc_0$
  - Conflicting requirements

Robustness of Constraints

$lr1 \mapsto rr0 \prec y2$



RT inequality:

$lr1 + d2 + rr0 + m \leq lr1 + d2 + rr1 + d3 + y2$

A margin m is incorporated

$lr1 + d2 + rr1 + m \leq lr1 + d2 + rr1 + d3 + y2$

$m = (lr1 + d2 + rr1 + d3 + y2) - (lr1 + d2 + rr1)$

$m = (D_S + D_L + D_S + D_L + D_S) - (D_S + D_L + D_S)$

$m = D_S + D_L$

m represents inherent robustness

- higher the value to m the better the robustness of the constraint
- Each constraint in the set is analyzed and the value of m
- worst-case value of m in a set are used to qualify the constraint set

Possible competing timing requirements:

RTCA: $x \mapsto y \prec z$
RTCB: $x \mapsto w \prec y$

RTCA may be attempting to reduce delay in the path $x \mapsto y$ while *RTCB* is trying to increase the path delay.

Interacting paths:

Early Path: $[A, X, Y, X, Y, Z]$
Late Path: $[B, X, Y, Z]$

- paths are viewed as sets
- sets can either be *equivalent*, *disjoint* or *overlapping*
- When they overlap, one can be the subset of another, or they can have common elements in a way that the intersection is not equivalent to either set.

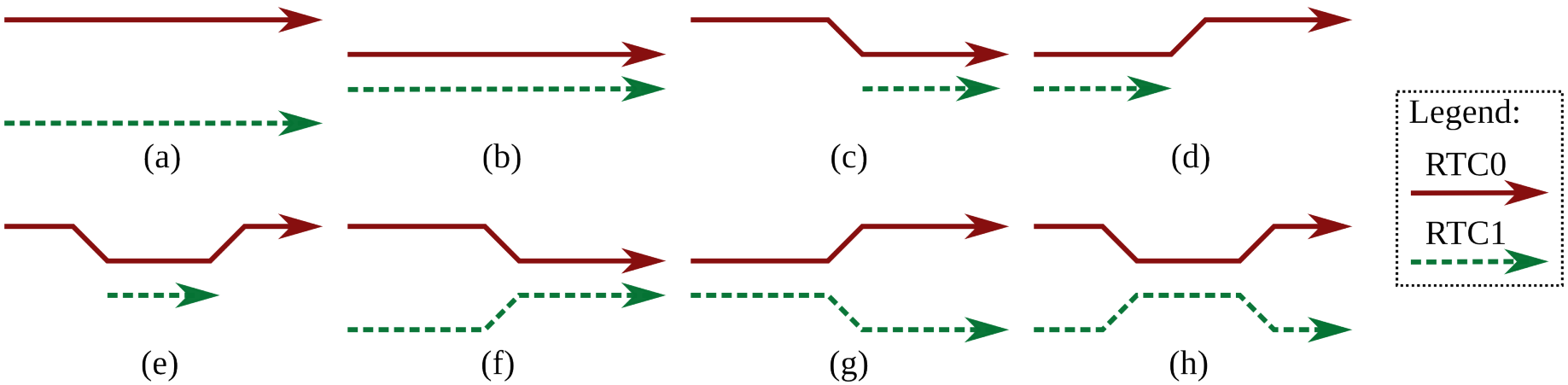


Figure : Examples of various ways in which paths from different RTCs can overlap.

Strong conflicts can create competing timing requirements that cannot be met.
Weak conflicts may also create competing timing requirements, however due to non overlapping portions, they can always be met.

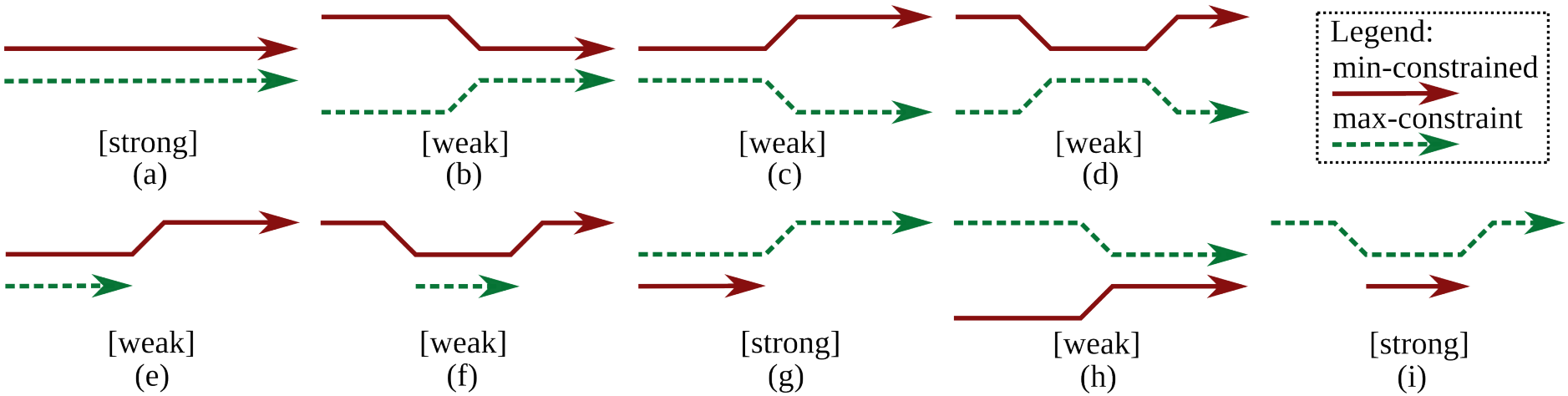


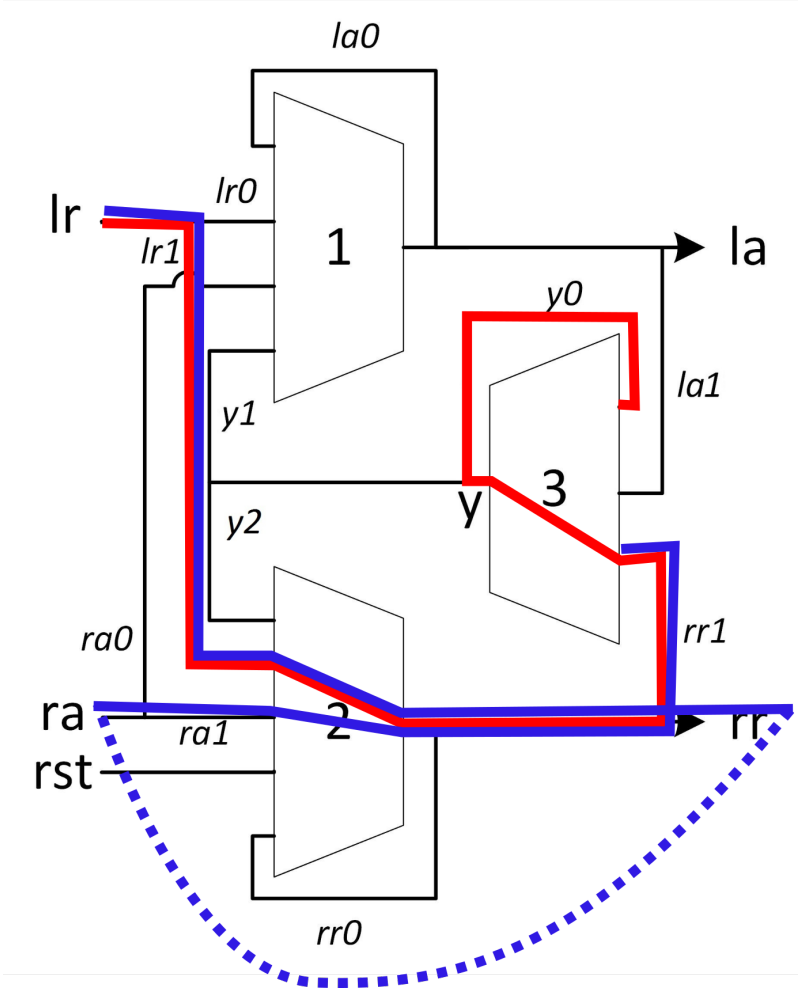
Figure : Various possible conflicts

A directed graph $G = (V, E)$ is utilized to represent two-sided timing constraints

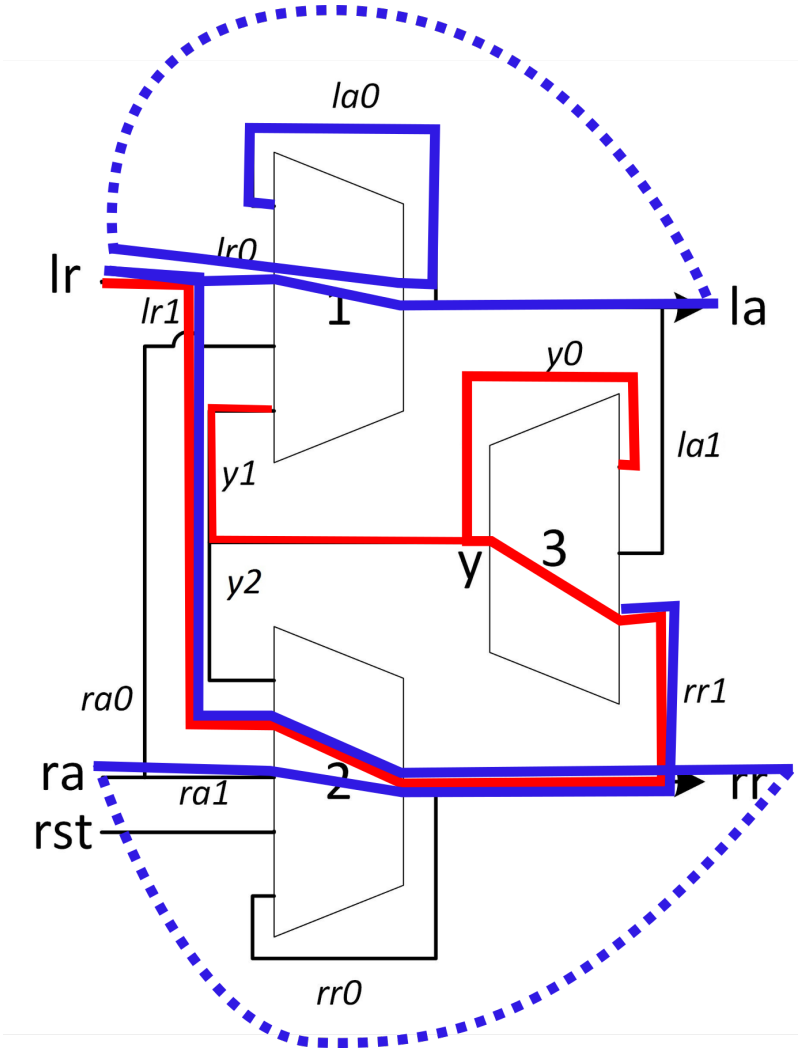
V is the set of vertices. Each individual RTC is represented by a Vertex

A conflict between two RTCs is represented by a directed edge connecting the two
RTC0: $lr \mapsto y0 \prec rr1$
RTC1: $lr \mapsto y1 \prec la0$

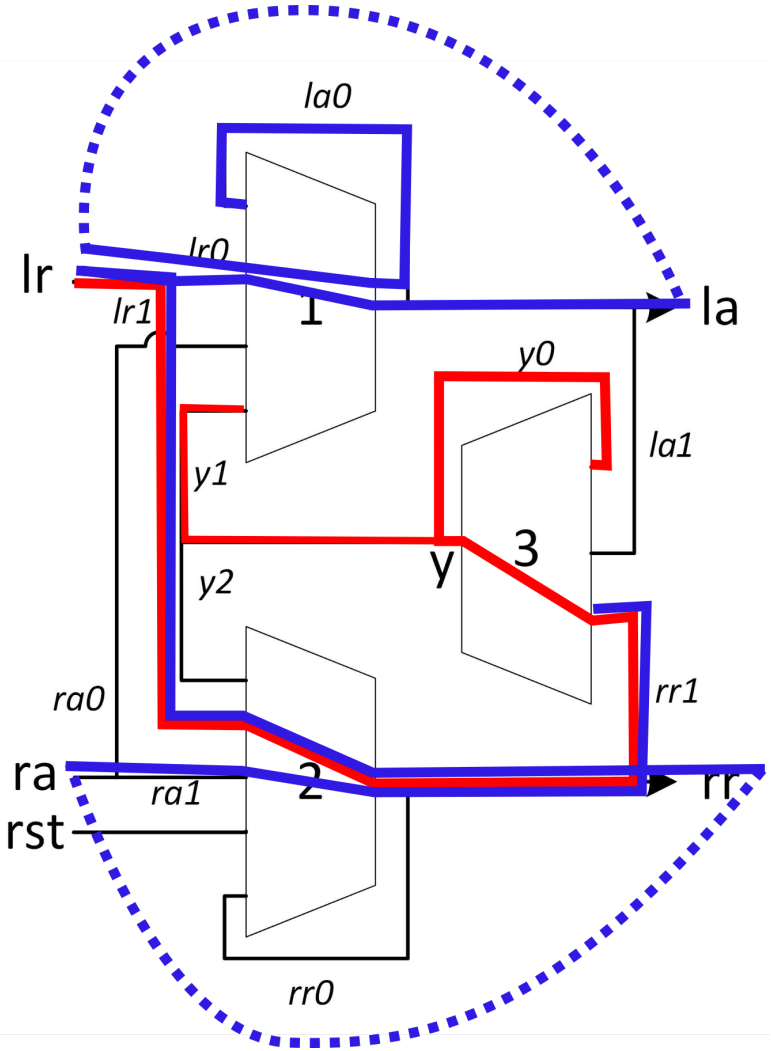
RTC0: $lr1 \mapsto y0 \prec rr1$



RTC0: $lr \mapsto y0 \prec rr1$
RTC1: $lr \mapsto y1 \prec la0$

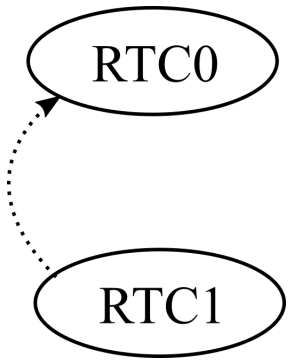


RTC0 places a min delay requirement between $lr1$ and $rr1$
RTC1 places a max delay between $lr1$ and $y1$
These constraints have overlapping paths

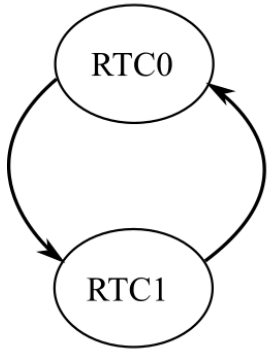


RTC0 places a min delay requirement between $lr1$ and $rr1$
RTC1 places a max delay between $lr1$ and $y1$
These constraints have overlapping paths

RTC0 places a min delay requirement between $lr1$ and $rr1$
RTC1 places a max delay between $lr1$ and $y1$
These constraints have overlapping paths



RTC0: $pod \mapsto poc_0 \prec poc_1$
RTC1: $pod \mapsto poc_1 \prec poc_0$
Two sided constraints on both paths that can never be simultaneously resolved



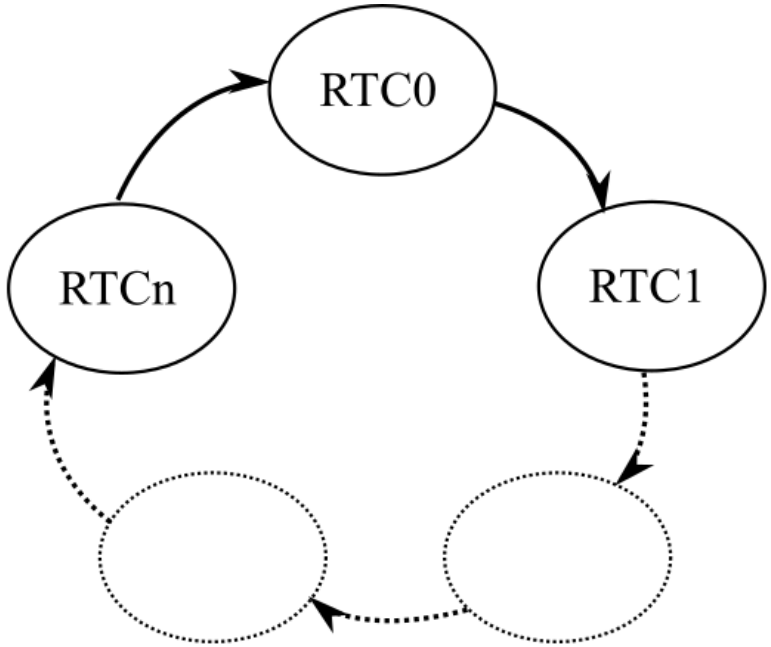


Figure : Cyclical Conflicts

Strong Conflicts are represented by solid lines, and weak by dotted lines

Given a RTC set from a circuit:

- The minimum delay constrained path (early path) of an RTC may conflict with the maximum delay constrained (late path) of the other RT constraints within a set.
- The minimum delay constrained path (early path) of an RTC may conflict with the maximum delay constrained (late path) of the other RT constraints within a set.

Hence each RTC can be analyzed for cyclical dependencies between constraints.

Each constraint set is analyzed for Strong Cyclical Conflicts and Weak Cyclical Conflicts

Constraint set metrics

With our analysis of constraint sets, we now have the following information for each set:

1. Inherent robustness (m)
2. Weak conflicts (wc)
3. Strong conflicts (sc)
4. Weak cyclical conflicts (wcc)
5. Strong cyclical conflicts (scc)

Constraint set metrics

With our analysis of constraint sets, we now have the following information for each set:

1. **Inherent robustness** (m)
2. Weak conflicts (wc)
3. **Strong conflicts** (sc)
4. **Weak cyclical conflicts** (wcc): With at least one strong conflict
5. **Strong cyclical conflicts** (scc)

What else can we use this for?

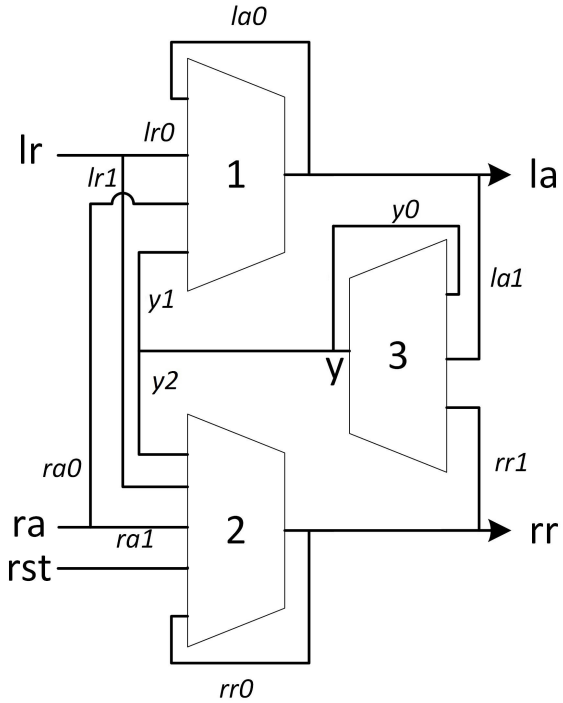


Figure : Controller with modeled forks

- A SI based circuit is first modeled
- Extracted constraints sets are evaluated and the best set is picked
- The best constraint set is utilized and one single fork is added
- The extraction of RT constraints is rerun

RTC set0

$rtc0 : lr \ \mapsto \ y0 \ \prec la1;$
$(lr1 + d2 + rr1 + d3 + y0 + m \leq lr0 + d1 + la + D_E + lr0 + d1 + la1)$
$rtc1 : lr1 \ \mapsto \ y0 \ \prec rr1;$
$(lr1 + d2 + rr1 + d3 + y0 + m \leq lr1 + d2 + rr + D_E + ra + d3 + rr1)$
$rtc2 : lr1 \ \mapsto \ y2 \ \prec y;$
$(lr1 + d2 + rr1 + d3 + y2 + m \leq lr1 + d2 + rr + D_E + ra + d2 + rr1 + d2)$
$rtc3 : lr \ \mapsto \ y2 \ \prec lr1;$
$(lr1 + d2 + rr1 + d3 + y2 + m \leq lr0 + d1 + la + D_E + lr0 + d1 + la + D_E + lr1)$

Figure : Relative timing constraint set: set0

Another Set

$rtc0$ - $rtc2$ - same as $rtc2$ from Set0

$rtc3 : lr \mapsto y1 \prec lr1;$

$(lr1 + d2 + rr1 + d3 + y1 + m \leq lr0 + d1 + la + D_E + lr0 + d1 + la + D_E + lr1)$

$rtc4 : lr \mapsto lr1 \prec y2;$

$(lr1 + d1 + la + D_E + lr1 + d1 + la + D_E + lr1 + m \leq lr1 + d2 + rr1 + d3 + y2)$

$rtc5 : lr \mapsto y0 \prec rr1;$

$(lr1 + d2 + rr + D_E + ra1 + d2 + rr1 + d3 + y0 + m \leq$
$lr0 + d1 + la + D_E + lr0 + d1 + la + D_E + lr1 + d2 + rr1)$

$rtc6 : rr1 \mapsto y2 \prec y;$

$(rr1 + d3 + y2 + m \leq rr1 + d3 + y1 + d1 + la1 + d3)$

$rtc7 : lr \mapsto ra1 \prec lr1;$

$(lr0 + d1 + la + D_E + lr0 + d1 + D_E + lr1 + d2 + rr + D_E + ra1 + m \leq$
$lr1 + d2 + rr + D_E + ra1 + D_E + rr1 + d3 + y1 + d1 + la + D_E + lr1)$

$rtc8 : lr \mapsto lr1 \prec ra1;$

$(lr1 + d2 + rr + D_E + ra1 + D_E + rr1 + d3 + y1 + d1 + la + D_E + lr1 + m \leq$
$lr0 + d1 + la + D_E + lr0 + d1 + D_E + lr1 + d2 + rr + D_E + ra1)$

Figure : Relative timing constraint set: set2

Table : RTC set0 analysis

Constraint#	m
rtc0	$1 * D_S + 1 * D_L$
rtc1	$1 * D_S + 1 * D_L$
rtc2	$2 * D_S + 1 * D_L$
rtc3	$2 * D_S + 2 * D_L$
Worst-case m	$1 * D_S + 1 * D_L$
Average m	$1.5 * D_S + 1.3 * D_L$

Number of Strong Conflicts: 0
Number of Strong Cycles: 0
Number of Weak Cycles: 0

Analysis was automated

Order of precedence or priority to rank the constraint sets:

1. *Strong cyclical conflicts*: Sets with the lower number of strong cyclical conflicts are ranked higher
2. *Worst case m*: Sets with higher worst case m are ranked higher
3. *Strong conflicts*: Sets with lower number of strong conflicts are ranked higher
4. *Weak cyclical conflicts*: Sets with lower number of weak cyclical conflicts are ranked higher
5. *Average m*: Sets with a higher average m are ranked higher. Usually, this metric is only used to break ties between sets.

Table : Controller implementation results on the FPGA

	Worst-case m	Average m	Strong Conflicts	Strong Cyclic Conflct	Weak Cyclic Conflct	Max margin (ps)	Min margin (ps)	Ave margin (ps)	Constr. met?
set0	$1 * D_S + 1 * D_L$	$1.5 * D_S + 1.3 * D_L$	0	0	0	661	212	410	Yes
set1	$1 * D_S + 1 * D_L$	$1.5 * D_S + 1.3 * D_L$	0	0	0	438	317	358	Yes
set2	$-3 * D_S + -3 * D_L$	$0.7 * D_S + 0.4 * D_L$	11	1	9	334	-248	148	No
set3	$-3 * D_S + -3 * D_L$	$0.9 * D_S + 0.7 * D_L$	9	0	9	474	-173	116	No
set4	$-3 * D_S + -3 * D_L$	$0.9 * D_S + 1 * D_L$	7	0	7	532	-173	196	No
set5	$-3 * D_S + -3 * D_L$	$0.9 * D_S + 0.9 * D_L$	7	0	6	532	-173	179	No
set6	$-3 * D_S + -3 * D_L$	$1.1 * D_S + 1 * D_L$	7	0	7	535	-173	267	No
set7	$-3 * D_S + -3 * D_L$	$1.2 * D_S + 1 * D_L$	7	0	7	769	-173	222	No
set8	$-3 * D_S + -3 * D_L$	$1.1 * D_S + 1 * D_L$	7	0	6	769	-173	267	No
set9	$-3 * D_S + -3 * D_L$	$1.3 * D_S + 1.1 * D_L$	7	0	7	665	-173	208	No

Table : RT generation runtime reduction using developed tool for FPGA controller implementation

	RT generation runtime (s)	Analysis tool runtime (s)	Total runtime (s)
Direct DI model	7139.52	–	7139.52
Iterative model	8.01	5.85	13.86
Total runtime reduction	–	–	99.8%

Table : RT generation runtime reduction for asynchronous controllers using developed tool

Controller	Reference	Traditional runtime (s)	Iterative tool runtime (s)	Reduction
LC_BM	[1]	7,139.52	13.86	99.8%
PCHB	[2]	3.56	0.53	85.1%
BRF1,LH1	[3]	10,342.71	297.62	97.2%

[1] K. S. Stevens, Y. Xu, and V. Vij, "Characterization of Asynchronous Templates for Integration into Clocked CAD Flows," in *15th International Symposium on Asynchronous Circuits and Systems*. IEEE, May 2009, pp. 151161.
[2] M. D. Riedel and J. Bruck, "Timing Analysis of Cyclic Combinational Circuits," in *International Workshop on Logic and Synthesis*. IEEE, 2004, pp. 6977.
[3] S. B. Furber, "A Small Compendium of 4-Phase Macropipeline Latch Control Circuits," University of Manchester, Dept. of Computer Science, Technical Report v0.3, 17/01/99, 1999.

Conclusion

- A methodology to evaluate and rank constraint sets has been presented
- RT constraint robustness and timing conflicts are analyzed
- A methodology to identify cyclical timing conflicts is presented
- The methodology is used to optimized RTs constraint for DI models

Key References

K. S. Stevens, R. Ginosar, and S. Rotem, "Relative Timing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 1, no. 11, pp. 129140, Feb. 2003.

W. Lee, V. S. Vij, A. R. Thatcher, and K. S. Stevens, "Design of Low Energy, High Performance Synchronous and Asynchronous 64-Point FFT," in *Design, Automation and Test in Europe (DATE)*. IEEE, Mar 2013, pp. 242247.

P. A. Beerel, R. O. Ozdag, and M. Ferretti, A Designers Guide to Asynchronous VLSI. Cambridge University Press, 2010, ISBN 0521872448,9780521872447

W. Chuang, S. S. Sapatnekar, and I. N. Hajj, "Delay and Area Optimization for Discrete Gate Sizes under Double-Sided Timing Constraints," in Custom Integrated Cicruits Conference. IEEE, May 1993, pp. 9.4.1 9.4.4.

V. S. Vij, R. P. Gudla, and K. S. Stevens, "Interfacing Synchronous and Asynchronous Domains for Open Core Protocol," in *International Conference on VLSI Design*, Jan 2014, pp. 282287.

K. S. Stevens, Y. Xu, and V. Vij, "Characterization of Asynchronous Templates for Integration into Clocked CAD Flows," in *15th International Symposium on Asynchronous Circuits and Systems*. IEEE, May 2009, pp. 151161.

The End