Timing Analysis with Clock Skew

David Harris, Mark Horowitz[1], & Dean Liu[1]

David_Harris@hmc.edu, {horowitz, dliu}@vlsi.stanford.edu

March, 1999

Harvey Mudd College

Claremont, CA

[1] (with Stanford University, Stanford, CA)

Outline

- Introduction
- Timing Analysis Formulation
- Timing Analysis with Clock Skew
- Timing Verification Algorithm
- Results
- Conclusion

Introduction

Clock skew, as a fraction of the cycle time, is a growing problem for fast chips

- Fewer gate delays per cycle
- Poor transistor length, threshold tolerances
- Larger clock loads
- Bigger dice

Introduction

Clock skew, as a fraction of the cycle time, is a growing problem for fast chips

- Fewer gate delays per cycle
- Poor transistor length, threshold tolerances
- Larger clock loads
- Bigger dice

The designer may:

- Reduce skew

  Very hard; clock networks are already well optimized

Introduction

Clock skew, as a fraction of the cycle time, is a growing problem for fast chips

- Fewer gate delays per cycle
- Poor transistor length, threshold tolerances
- Larger clock loads
- Bigger dice

The designer may:

- Reduce skew

  Very hard; clock networks are already well optimized

- Tolerate skew

  Flip-flops and traditional domino circuits reduce cycle time by skew
  Latches and skew-tolerant domino can hide modest amounts of skew

Introduction

Clock skew, as a fraction of the cycle time, is a growing problem for fast chips

- Fewer gate delays per cycle
- Poor transistor length, threshold tolerances
- Larger clock loads
- Bigger dice

The designer may:

- Reduce skew

  Very hard; clock networks are already well optimized

- Tolerate skew

  Flip-flops and traditional domino circuits reduce cycle time by skew
  Latches and skew-tolerant domino can hide modest amounts of skew

- Only budget necessary skews

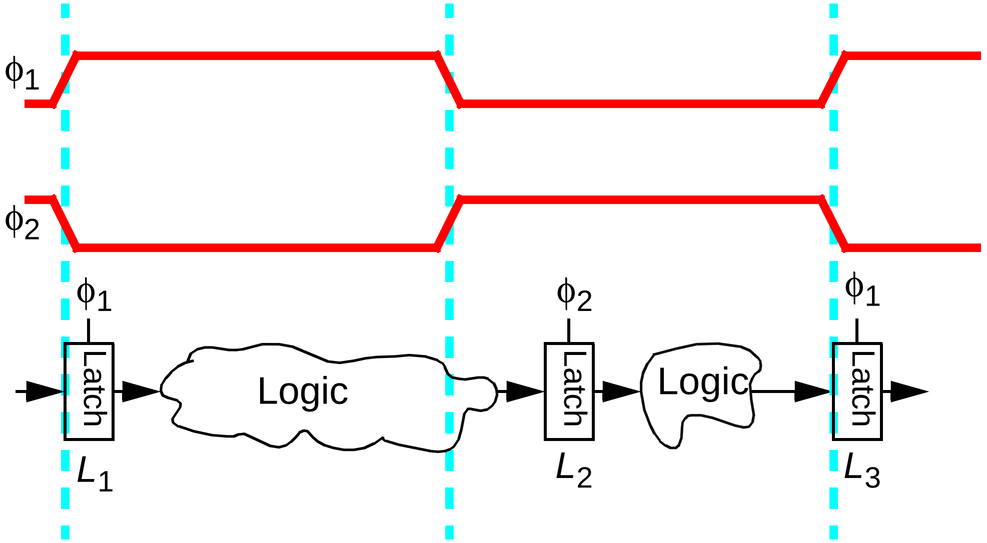
  Skew between nearby latches is often much less than skew across die
  Need better timing analysis for different skews between different latches

Timing Analysis Formulation

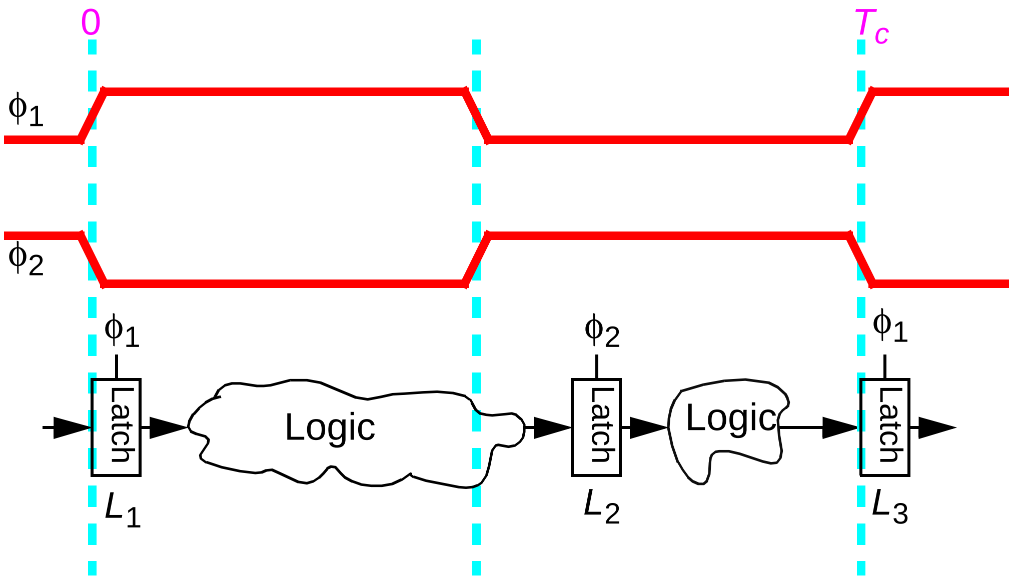
Build on Sakallah, Mudge, Olukotun (SMO) analysis of latch-based systems.

System contains:

- k clocks $C = \{\phi_1, \phi_2, ..., \phi_k)$
- l latches $L = \{L_1, L_2, ..., L_l\}$

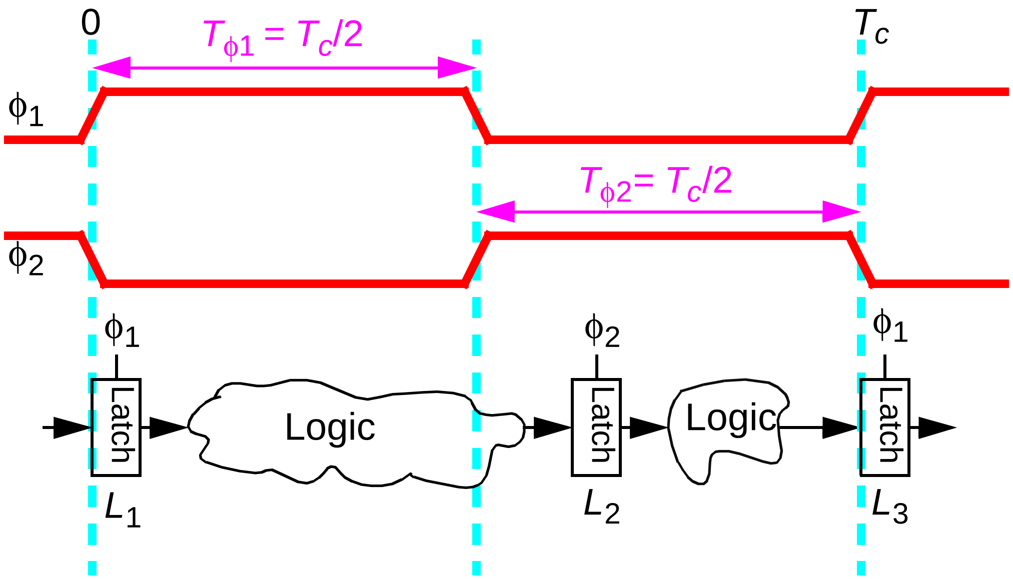


Clock Waveforms



T_c: cycle time

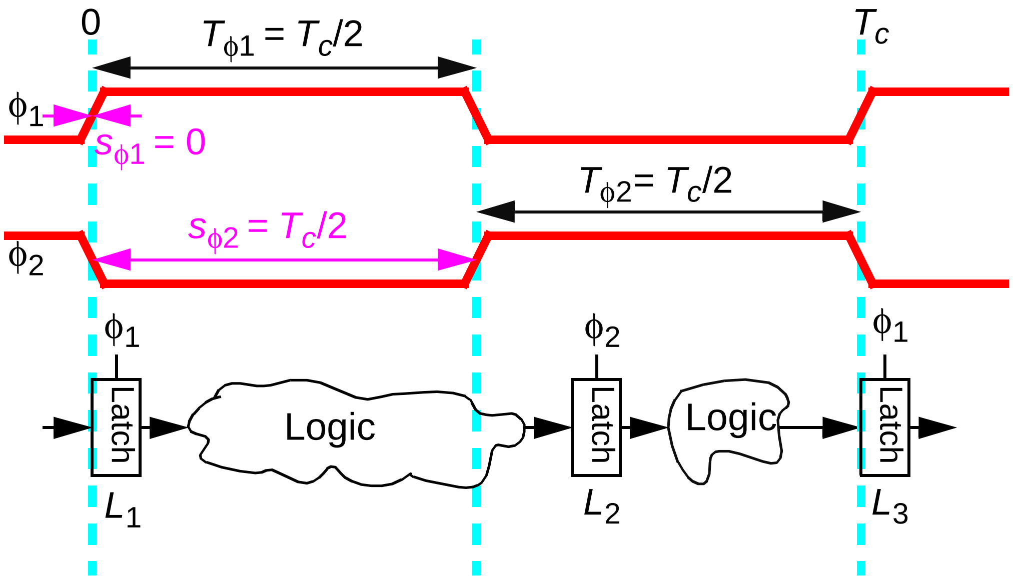
Clock Waveforms



T_c: cycle time

T_{ϕ_i}: duration for which ϕ_i is high

Clock Waveforms

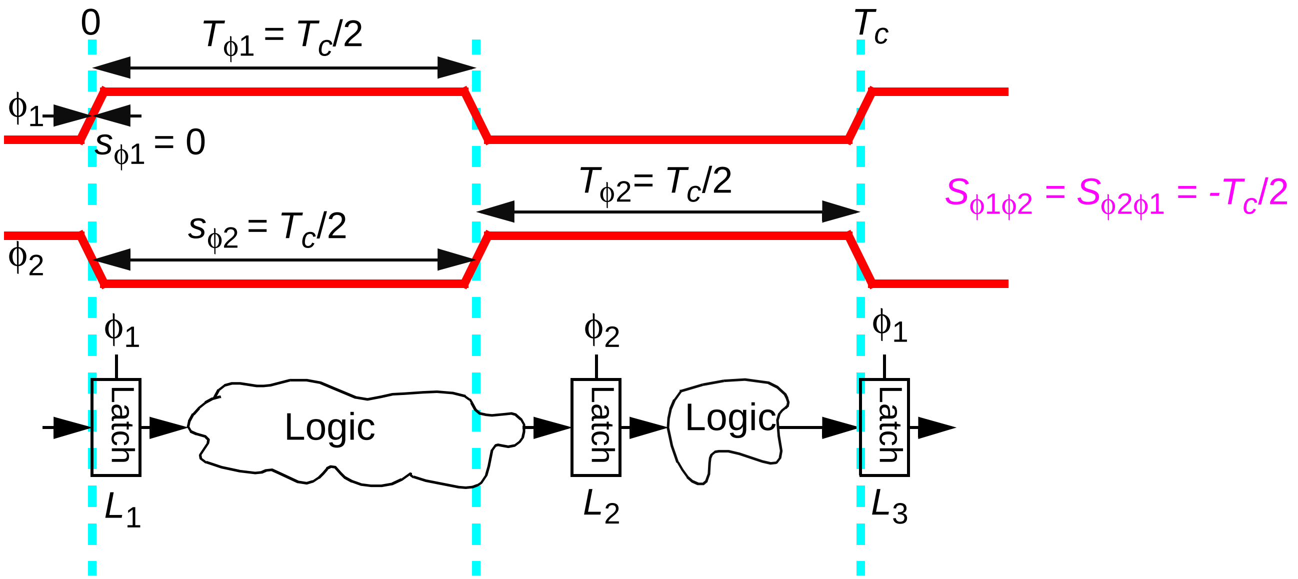


T_c: cycle time

T_{ϕ_i}: duration for which ϕ_i is high

s_{ϕ_i}: start time, relative to beginning of common clock, of ϕ_i being high

Clock Waveforms



T_c: cycle time

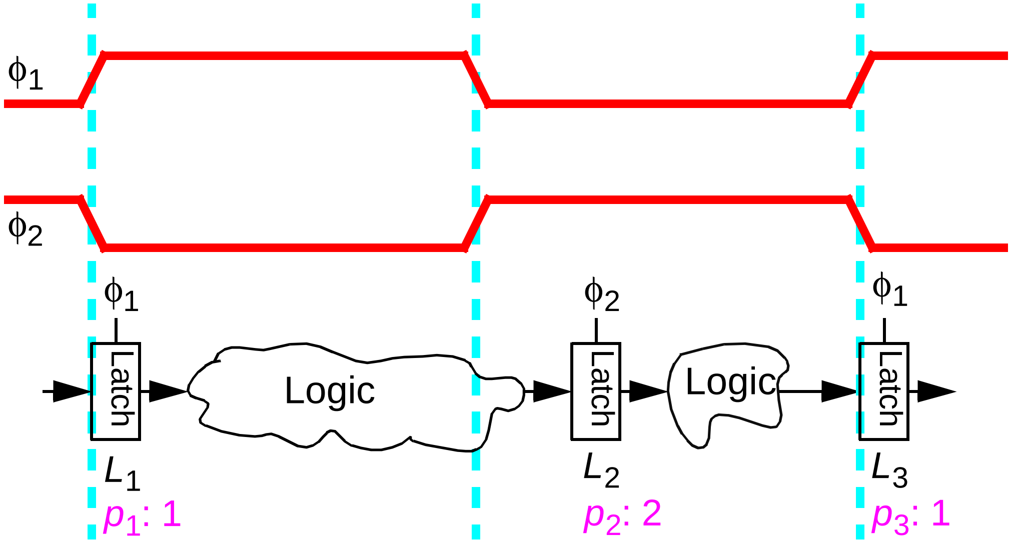
T_{ϕ_i}: duration for which ϕ_i is high

s_{ϕ_i}: start time, relative to beginning of common clock, of ϕ_i being high

$S_{\phi_i \phi_j}$: phase shift from ϕ_i to next occurrence of ϕ_j. Used to translate times relative to particular clock phases.

Latch Variables

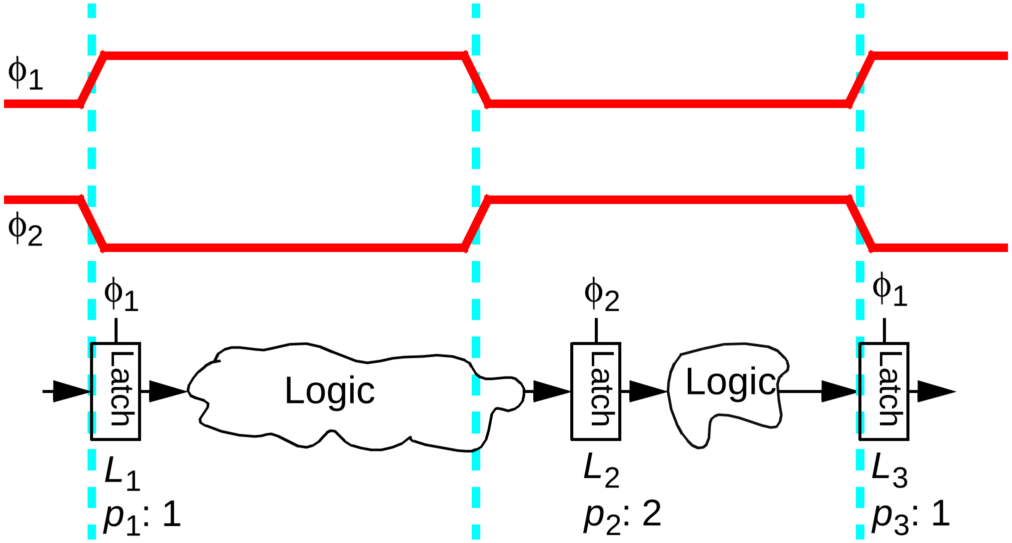
p_i: clock phase controlling latch *i*



Latch Variables

p_i: clock phase controlling latch *i*

Δ_{DC_i}: setup time for latch *i*

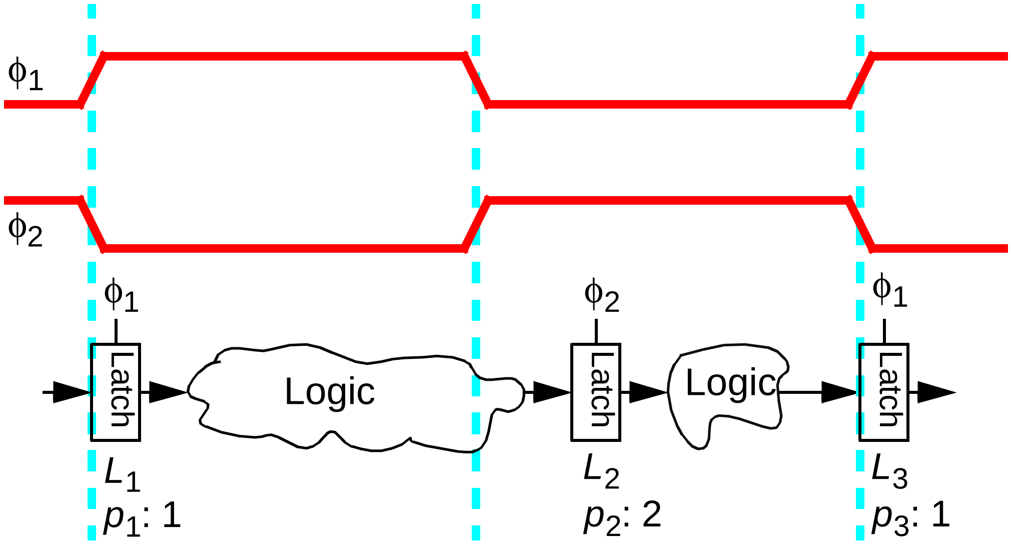


Latch Variables

p_i: clock phase controlling latch *i*

Δ_{DC_i}: setup time for latch *i*

Δ_{DQ_i}: propagation delay through latch *i*

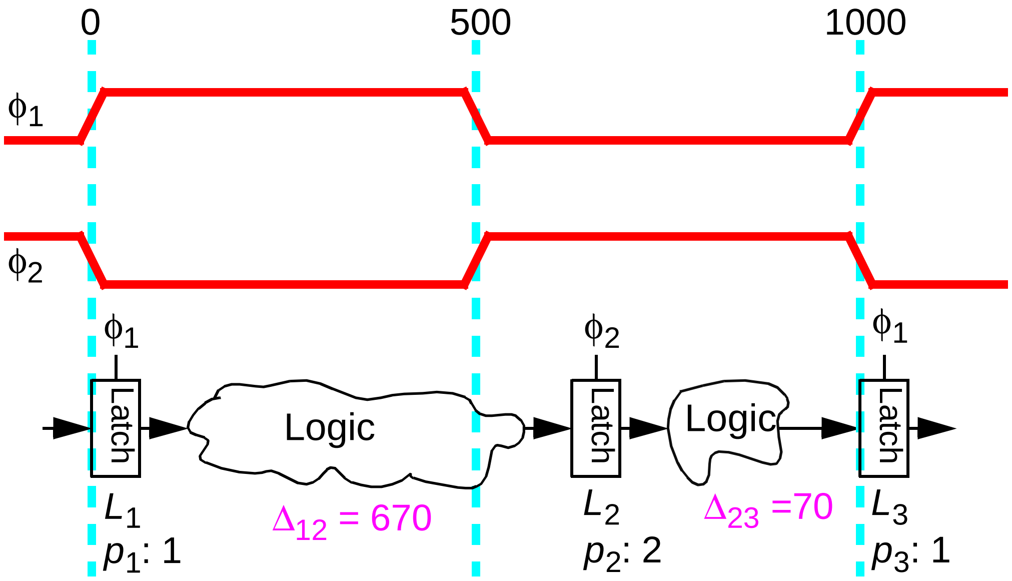


Latch Variables

p_i: clock phase controlling latch *i*

Δ_{DC_i}: setup time for latch *i*

Δ_{DQ_i}: propagation delay through latch *i*



Assume:

T_c = 1000 ps

Δ_{DQ} = 80 ps

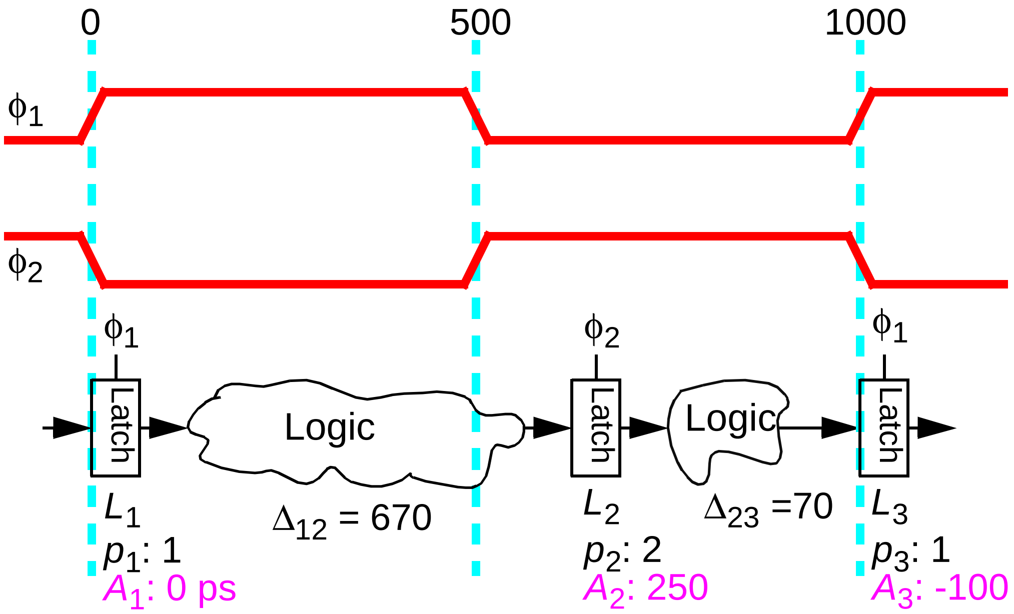
Δ_{ij}: propagation delay through logic between latches *i* and *j*

Latch Variables

p_i: clock phase controlling latch i

Δ_{DC_i}: setup time for latch i

Δ_{DQ_i}: propagation delay through latch i



Assume:

T_c = 1000 ps

Δ_{DQ} = 80 ps

Δ_{ij}: propagation delay through logic between latches i and j

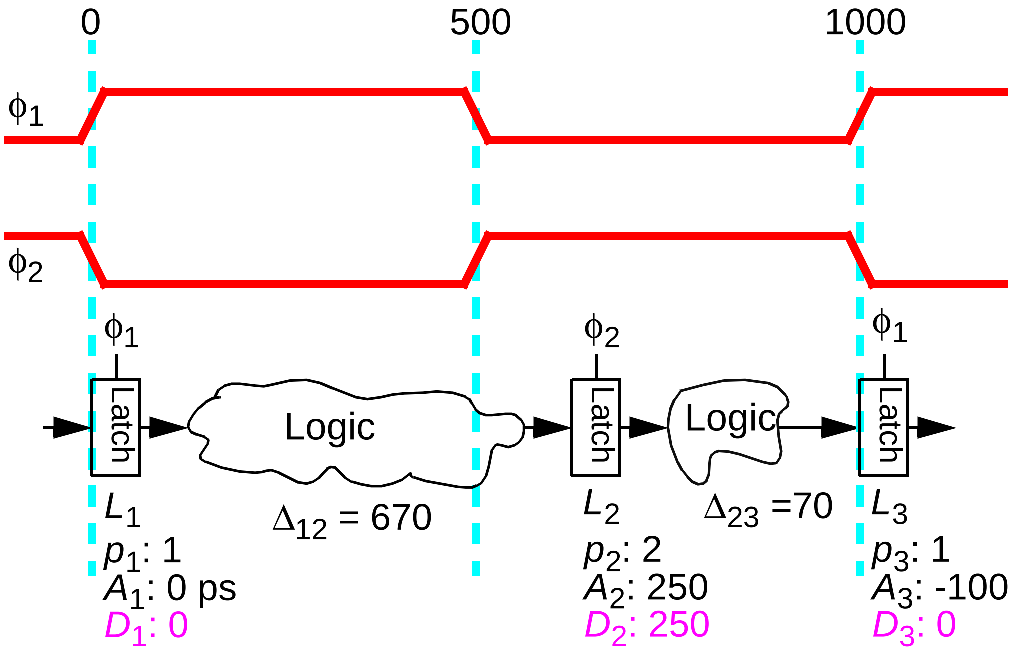
A_i: arrival time at latch i, relative to start of p_i

Latch Variables

p_i: clock phase controlling latch *i*

Δ_{DC_i}: setup time for latch *i*

Δ_{DQ_i}: propagation delay through latch *i*



p_1: 1
A_1: 0 ps
D_1: 0

p_2: 2
A_2: 250
D_2: 250

p_3: 1
A_3: -100
D_3: 0

Assume:

T_c = 1000 ps

Δ_{DQ} = 80 ps

Δ_{ij}: propagation delay through logic between latches *i* and *j*

A_i: arrival time at latch *i*, relative to start of p_i

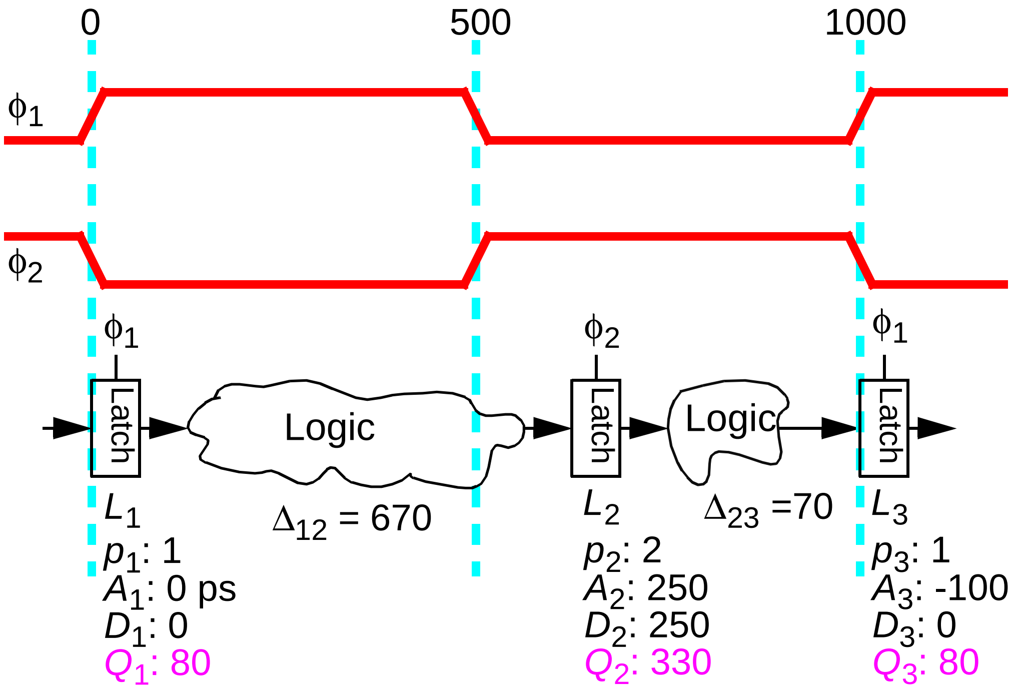
D_i: departure time from latch *i*

Latch Variables

p_i: clock phase controlling latch *i*

Δ_{DC_i}: setup time for latch *i*

Δ_{DQ_i}: propagation delay through latch *i*



0 500 1000

ϕ_1

ϕ_2

ϕ_1 Latch Logic ϕ_2 Latch Logic ϕ_1 Latch

L_1
p_1: 1
A_1: 0 ps
D_1: 0
Q_1: 80

$\Delta_{12} = 670$

L_2
p_2: 2
A_2: 250
D_2: 250
Q_2: 330

$\Delta_{23} = 70$

L_3
p_3: 1
A_3: -100
D_3: 0
Q_3: 80

Assume:

$T_c =$ 1000 ps

$\Delta_{DQ} =$ 80 ps

Δ_{ij}: propagation delay through logic between latches *i* and *j*

A_i: arrival time at latch *i*, relative to start of p_i

D_i: departure time from latch *i*

Q_i: output time of latch *i*

Timing Constraints

Latch Departure:

$$\forall i \in L \qquad D_i = max(0, A_i)$$

Timing Constraints

Latch Departure:

$$\forall i \in L \qquad D_i = max(0, A_i)$$

Latch Output:

$$\forall i \in L \qquad Q_i = D_i + \Delta_{DQ_i}$$

Timing Constraints

Latch Departure:

$$\forall i \in L \qquad D_i = max(0, A_i)$$

Latch Output:

$$\forall i \in L \qquad Q_i = D_i + \Delta_{DQ_i}$$

Latch Arrival:

$$\forall i, j \in L \qquad A_i = max(Q_j + \Delta_{ji} + S_{p_j p_i})$$

Timing Constraints

Latch Departure:

$$\forall i \in L \qquad D_i = max(0, A_i)$$

Latch Output:

$$\forall i \in L \qquad Q_i = D_i + \Delta_{DQ_i}$$

Latch Arrival:

$$\forall i, j \in L \qquad A_i = max(Q_j + \Delta_{ji} + S_{p_j p_i})$$

Propagation Constraints:

$$\forall i, j \in L \qquad D_i = max(0, max(D_j + \Delta_{DQ_j} + \Delta_{ji} + S_{p_j p_i}))$$

Timing Constraints

Latch Departure:

$$\forall i \in L \qquad D_i = max(0, A_i)$$

Latch Output:

$$\forall i \in L \qquad Q_i = D_i + \Delta_{DQ_i}$$

Latch Arrival:

$$\forall i, j \in L \qquad A_i = max(Q_j + \Delta_{ji} + S_{p_j p_i})$$

Propagation Constraints:

$$\forall i, j \in L \qquad D_i = max(0, max(D_j + \Delta_{DQ_j} + \Delta_{ji} + S_{p_j p_i}))$$

Setup Constraints:

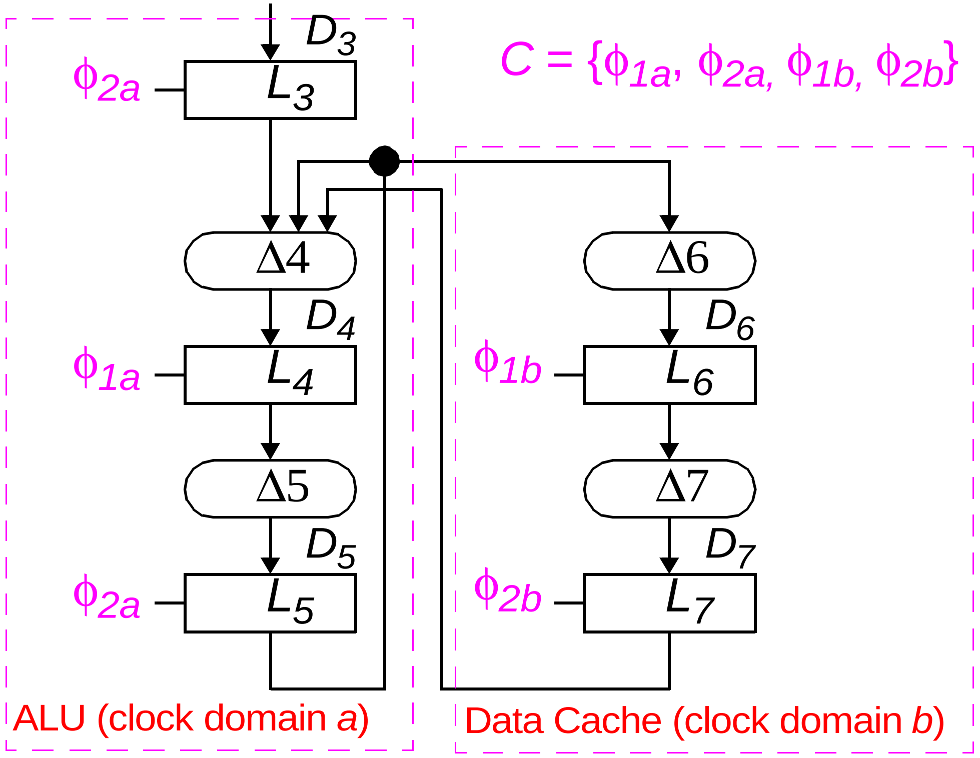
$$\forall i \in L \qquad D_i + \Delta_{DC_i} \leq T_{p_i}$$

Timing Analysis with Clock Skew

Clock skew is the difference between nominal and actual interarrival times of a pair of clocks.

Enlarge set of physical clocks *C* to model skew between nominally identical clocks.

Example:



$C = \{\phi_{1a}, \phi_{2a}, \phi_{1b}, \phi_{2b}\}$

t_{skew}^{local} within domains

t_{skew}^{global} between domains

Single Skew Formulation

Easy and conservative to budget global skew everywhere

Effectively increases setup time at each latch

Setup Constraints:

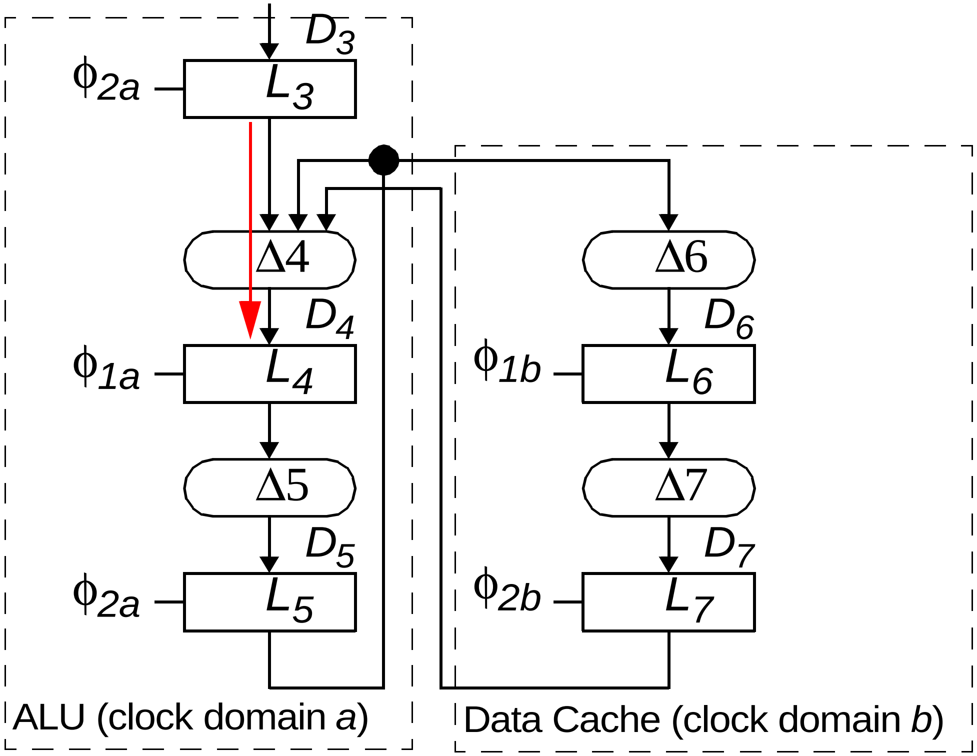
$$\forall i \in L \qquad D_i + \Delta_{DC_i} + t_{skew}^{global} \le T_{p_i}$$

Too conservative for high-speed designs with big global skews

Exact Skew Budgets

How much skew must be budgeted?

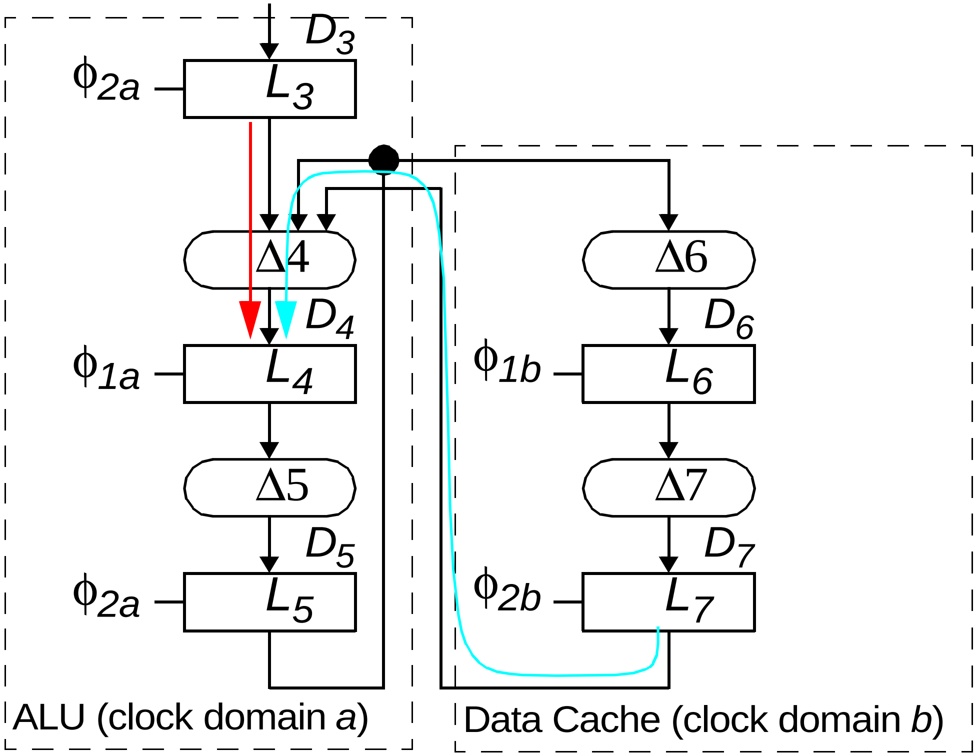
- L_3 to L_4: local skew



Exact Skew Budgets

How much skew must be budgeted?

- L_3 to L_4: local skew
- L_7 to L_4: global skew

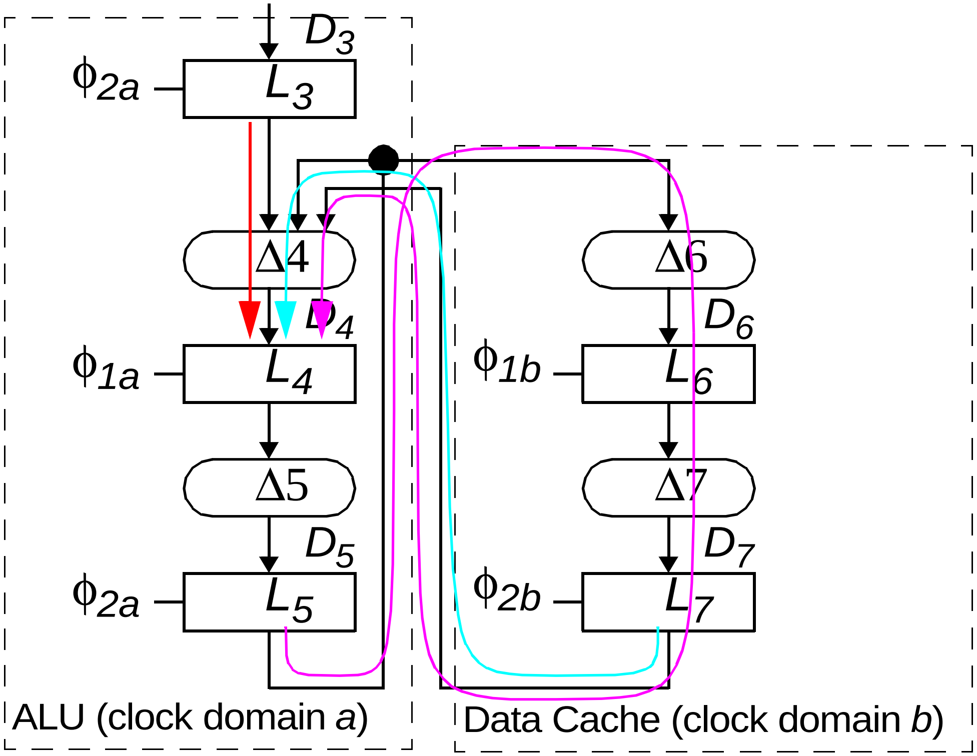


Exact Skew Budgets

How much skew must be budgeted?

- L_3 to L_4: local skew
- L_7 to L_4: global skew
- L_5 to L_4 through transparent L_6, L_7: local skew

Must track launching clock to determine skew budget



Exact Skew Formulation

Define arrival and departure times with respect to launching clocks:

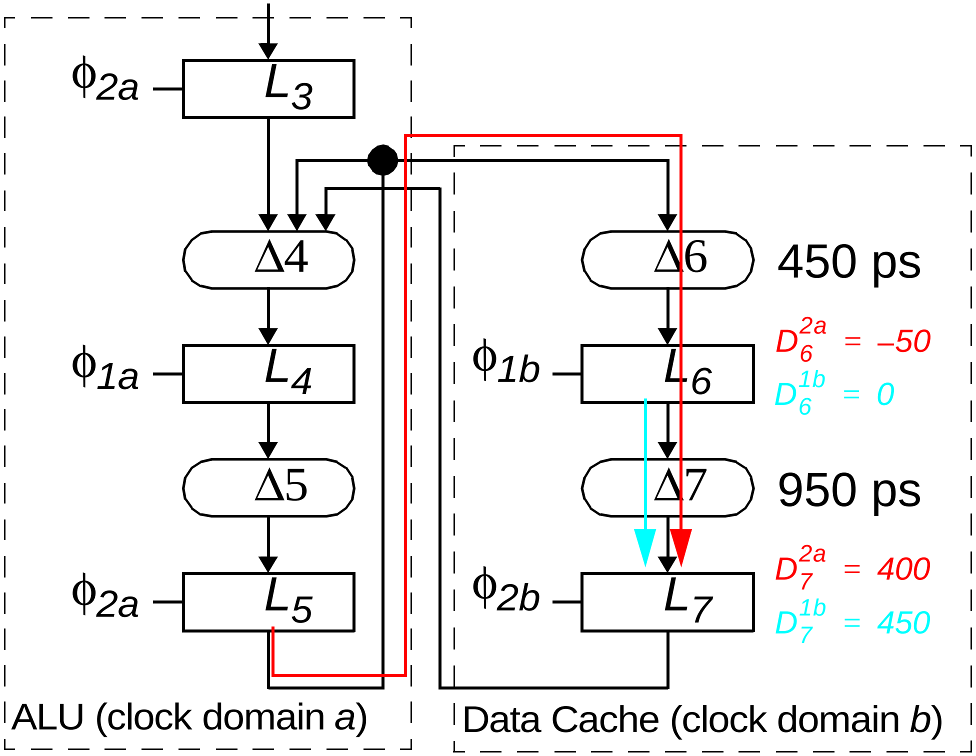
- A_i^c: arrival time at latch *i* for path launched by clock *c*
- D_i^c: departure time from latch *i* for path launched by clock *c*
- $t_{skew}^{\phi_i, \phi_j}$: skew between clocks ϕ_i, ϕ_j

Negative Departure Times

Must now allow negative departure times with respect to other clocks:

- Path from L_5 to L_7 is earlier than L_6 to L_7, but sees more skew, miss setup
- Reaches L_6 at -50 ps, but L_6 may be transparent by then because of skew

Departure times w.r.t. latch’s own clock still must be nonnegative



$T_c = 1000$ ps

$$t_{skew}^{2a,\,2b} = 150 \text{ ps}$$

$$t_{skew}^{1b,\,2b} = 25 \text{ ps}$$

$$\Delta_{DC} = 0$$

$$\Delta_{DQ} = 0$$

Exact Constraints with Skew:

Propagation Constraints (single skew):

$$\forall i, j \in L \qquad D_i = max(0, max(D_j + \Delta_{DQ_j} + \Delta_{ji} + S_{p_j p_i}))$$

Setup Constraints (single skew):

$$\forall i \in L \qquad D_i + \Delta_{DC_i} + t_{skew}^{global} \le T_{p_i}$$

Propagation Constraints (exact skew):

$$\begin{aligned} \forall i, j \in L, c \in C \qquad & if\ c = p_i \\ & then\ D_i^c = max(0, max(D_j^c + \Delta_{DQ_j} + \Delta_{ji} + S_{p_j p_i})) \\ & else\ D_i^c = max(D_j^c + \Delta_{DQ_j} + \Delta_{ji} + S_{p_j p_i}) \end{aligned}$$

Setup Constraints (exact skew):

$$\forall i \in L, c \in C \qquad D_i^c + \Delta_{DC_i} + t_{skew}^{c, p_i} \le T_{p_i}$$

Other Timing Constraints

Flip-flops:

- No transparency, easier than latches
- Still budget skew between launching and receiving clocks

Min-delay:

- Only requires checks between consecutive pairs of clocked elements
- Standard verification algorithms work if proper skew is used

Verification Algorithm

Check constraints with generalized Szymanski-Shenoy relaxation algorithm

```
For each latch $i$:
  $D_i^{p_i} = 0$; $D_i^{max} = 0$; $c_i^{max} = p_i$          // initialize departure times
  Enqueue $D_i^{p_i}$
While queue is not empty
  Dequeue $D_j^c$
  For each latch $i$ in fanout of $j$
    $A = D_j^c + \Delta_{DQj} + \Delta_{ji} + S_{p_j p_i}$          // calculate arrival time
    If $(A > D_i^c)$ AND $\left(A + t_{skew}^{c_i^{max}, c} > D_i^{max}\right)$          // is it possibly critical?
      If $(A + \Delta_{DCi} + t_{skew}^{c, p_i} > T_{p_i})$          // does it violate setup time?
        Report setup time violation
      Else
        $D_i^c = A$; Enqueue $D_i^c$          // keep following path
        If $(A > D_i^{max})$ $D_i^{max} = A$; $c_i^{max} = c$
```

Results

Analyzed MAGIC: Memory & General Interconnect Controller of FLASH supercomputer

Assume $t_{skew}^{local} = 250ps$ $t_{skew}^{global} = 500ps$

Model A:

- As designed, from MAGIC .sdf database

Model B:

- Flops converted to latch pairs, logic balanced between pairs

Results

Analyzed MAGIC: Memory & General Interconnect Controller of FLASH supercomputer

Assume $t_{skew}^{local} = 250ps$ $t_{skew}^{global} = 500ps$

Model A:

- As designed, from MAGIC .sdf database

Model B:

- Flops converted to latch pairs, logic balanced between pairs

		Model A	Model B
# Flip-Flops		10559	0
# Latches		1819	22937
Single Skew	T_c	9.43 ns	8.05 ns
	# Latch Departures Checked	3866	24995
Exact Skew	T_c	9.38	7.96
	# Latch Departures Checked	4009	25328

CPU time < 1 second in all cases

Conclusions

Global skews will be too large for GHz + systems

- Use skew-tolerant circuit techniques such as latches
- Take advantage of smaller local skews where possible

Requires support of timing analyzer

- Budget appropriate skew at each receiver
- Track departure times with respect to launching clocks
- Allow negative departure times with respect to other clocks

Leads to explosion in number of timing constraints. However...

- Most are not tight because most critical paths do not borrow time across many latches
- Relaxation algorithm automatically prunes loose constraints
- Very small increase in runtime

Expect synchronous systems well beyond 1 GHz