Interdependent Latch Setup/Hold Time Characterization via Euler-Newton Curve Tracing on State-Transition Equations

Shweta Srivastava, Jaijeet Roychowdhury

Dept of ECE, University of Minnesota, Twin Cities

shwetas@umn.edu
Setup/Hold Times in Timing Analysis

Hold Time constraint for short path

\[ \text{hold-time}_{R3} + \delta < \text{clk-to-Q-delay}_{R2} + \text{short-path-delay} \]

\[ T > \text{clock-to-delay}_{R1} + \text{long-path-delay} + \text{setup-time}_{R3} - \delta \]
Finding setup/hold times
- Crucial component of library characterization
- **Accuracy all-important**
  - detailed ckt level simulation, best models
- Takes **months** for a cell library
  - Intel, IBM, AMD, ...

Finding setup/hold times is important but expensive
Setup and Hold Times

Clock-to-Q delay

Q output

Failed Transition

Q output waveforms

Clock Edge

Is There Only **ONE** Setup and Hold Time?

- **Assumption** made today for STA:
  - ✯ setup/hold times *unique*

**Assumption NOT TRUE!**

There can be many pairs of setup/hold times for a latch/register
Setup/Hold Time Tradeoff Curves


Each point on the curve represents a valid pair of setup/hold times

Impact

Reducing pessimism in timing analysis
Interdependent Setup and Hold Times

Clock-to-Q delay

Same hold delay for different setup and hold delays

Same Q output waveforms

How to Exploit Setup/Hold Interdependence in Timing Analysis?

Exploit Setup/Hold Time Trade-off

hold violation is removed
at the expense of larger setup time

\[ \text{hold-time}_{R3} + \delta < \text{clk-to-Q-delay}_{R2} + \text{short-path-delay} \]

hold time is large

hold time is small

may not be critical

Finding Tradeoff Curve is Very Important!

Therefore:

- Finding setup/hold tradeoff curve is very valuable
- But: Very computationally expensive
Why Expensive?

Q value vs setup and hold delays

- **Problem:** finding the full Q surface
  - Large number of transient simulations
  - i.e., infeasible in practice
Our Contribution: Find the Curve Quickly

- Contribution of this work
  - New setup/hold trade-off curve finding technique
  - much faster than prior brute-force technique

- Key idea: trace the curve “directly”
  - avoid looking at points far from curve
Our Formulation of the Problem

A **scalar** equation with **two** unknowns (setup & hold time)

- One equation, two unknowns => many solutions (the curve)
- Solve numerically using Newton-Raphson method
  - **Rapid convergence**

Formulation:

Differential equations for register output waveform

\[ \frac{d}{dt} \bar{q}(\bar{x}) + \bar{f}(\bar{x}) + \bar{b}u(t) = 0 \]

Target clock-to-Q delay

Voltage threshold defining clock-to-Q delay

Q output waveform

setup time and hold time (unknowns)

Voltage threshold defining clock-to-Q delay

Contribution: Problem Formulation

Q output waveform

\[ Q_{tf}(\tau_s, \tau_h) = r \]

\[ h(\tau_s, \tau_h) \equiv Q_{tf}(\tau_s, \tau_h) - r = 0 \]

- Evaluation of \( h(\tau_s, \tau_h) \) is a transient simulation
- Solution of ONE point
  - special type of Newton-Raphson (NR) method
    - suitable for underdetermined equations
    - Moore-Penrose Pseudo Inverse NR (MPPI-NR)
- Finding the entire curve
  - Euler-Newton curve tracing method
  - Uses MPPI-NR for each point on curve

Intuition Behind Euler-Newton Method

Start with initial guess: \( A_E = (T_{s0}, T_{h0}) \)

Corrector step
Run Newton-Raphson

Predictor step
Compute Euler step

Predict a new point along the tangent of the curve

Constant clock-to-q delay Curve

Setup time \( T_s \)

Hold time \( T_h \)
Solving $h(\tau_s, \tau_h) = 0$ by Euler-Newton

Start with initial guess $(\tau_{s0}, \tau_{h0})$

evaluate $h(\tau_s, \tau_h)$

converged? yes

No

evaluate

$H(\vec{\tau}) = \begin{bmatrix} \frac{dh}{d\tau_s} & \frac{dh}{d\tau_h} \end{bmatrix}$

Moore-Penrose pseudo inverse

$H(\vec{\tau})^+ = H(\vec{\tau})^t (H(\vec{\tau})H(\vec{\tau})^t)^{-1}$

update

$\begin{bmatrix} \tau_s \\ \tau_h \end{bmatrix} = \begin{bmatrix} \tau_s \\ \tau_h \end{bmatrix} + h(\ldots) H(\vec{\tau})^+$

Found a point on the curve

Compute unit tangent vector

$T(H(\vec{\tau})) = \left( \begin{bmatrix} -\frac{dh}{d\tau_h} \\ \frac{dh}{d\tau_s} \end{bmatrix} \right) \frac{1}{length}$

Predict a new point along tangent

$\begin{bmatrix} \tau_{s0} \\ \tau_{h0} \end{bmatrix} = \begin{bmatrix} \tau_s \\ \tau_h \end{bmatrix} + \alpha T(H(\vec{\tau}))$
Connections with “RF” Simulation

- New Algorithm:
  - very similar to shooting method
  - used in “RF” simulation

- Implementation easy in RF simulators
  - Eg, SPECTRE-RF (Cadence), MICA (Freescale)

- “RF” simulation capabilities important for:
  - core characterization of digital circuits!
Validation
Validation on TSPC register

Positive-edge triggered

True Single-Phased Clocked register

Points on constant clock-to-q delay curve obtained by Euler-Newton method

Curve represents the pairs of setup and hold times – each point on curve results in 10% increase in nominal clock-to-Q delay
TSPC register validation

Speedup: ~12.5x (30min vs 6h 40min)

Q output surface as a function of setup and hold delays

Curve extracted from Q output surface: Brute-force method

Points obtained by Euler-Newton method

A plane at height of 1.25 V
C$^2$MOS register: Validation

Speedup: ~10.5x (16min vs 2h 48min)

Positive-edge triggered

Curve extracted from Q output surface: Brute-force method

Points obtained by Euler-Newton method
Summary

- New technique for fast setup/hold tradeoff curve characterization
  - Adapts ideas from “RF” simulation (shooting)

- **Importance/Impact:**
  - “free” elimination of violations/slack in timing analysis
  - reduces unnecessary optimism or pessimism

- Validated on TSPC and C2MOS registers
  - Speedups of an order of magnitude

*Key advance* in making setup/hold tradeoff exploitation PRACTICAL
Acknowledgments

- Chandramouli Kashyap and Chirayu Amin,
  Intel Strategic CAD Labs

- Sani Nassif,
  IBM Austin Research Laboratory