A Robust Cell-level Crosstalk Delay Change analysis

Igor Keller, Ken Tseng, Nishath Verghese

Cadence Design Systems

ikeller, kentseng, nashv@cadence.com
Some Technology Trends

- Interconnect is a key player
  - $C_c/C_{tot}$, $R_{line}/R_{drv}$ grow
- Timing closure is harder to reach
  - Clock cycle smaller
  - Tighter margins
  - Bigger uncertainty
- Waveform effects important
  - Resistive interconnect
  - X-Talk
  - Skewed gates
Modern SI-STA problems

- Biggest inaccuracy factors:
  - Modeling switching aggressors (linear, ramp? ☹)
  - Aggressor selection
  - Alignment
    - Linear Superposition of noiseless transition and glitch ☹
  - Irregular waveform
    - Measurement (slew & delay) ☹ ☹
    - Downstream effect ☹
  - Soft-ground effects ☹ ☹
  - Linear(ized) driver model:
    - Ceff:
      - Line resistance and x-talk

- Non-robustness
  - Amplifies small inaccuracy ☹
  - Discontinuities in results ☹
Today SI-STA must be

- **Accurate:**
  - Reference: WC realizable setup + spice
  - Conservative
  - Slightly conservative (~10%)

- **Efficient:**
  - Big designs overnight
  - Run time ~linear with number of nets, gates

- **Robust with respect to variations of**
  - Waveforms on aggressors/inputs
  - Alignment
  - Voltage supply, load on receiver
Building blocks of new SI-STA

- Driver model
- Interconnect
- Waveform
- Measurement
- Alignment
Delay Change: problem formulation

Stage

Arrival time

Delay Change

ROP

Receiver Output Probing

\[ T = J \int V(t) \, dt \]

\[ J_{\text{ref}} = \int_0^\infty t \delta(V_{\text{ref}}) V(t) \, dt, \quad V_{\text{ref}} = \frac{1}{2} V_{dd} \]

\[ \Delta = J[V_{ns}(t)] - J[V_{\text{nom}}(t)] \]

\[ \Delta \to \max_{\tau \in S} \]

\[ J_{\text{rop}}[V(t)] = J_{\text{ref}}[W(t)] \]
Driver Current Model

- Current is a function of voltages
- Like spice but for the whole cell
\[ I = I(V_i, V_o, \dot{V}_i, \ddot{V}_o) \]

\[ I = I(V_i, V_o) + C_M (\dot{V}_i - \dot{V}_o) + C_g \ddot{V}_o \]
Standard Cell Library

- ViVo models
  - Input and output CCCIs
  - Miller and ground caps

- Propagation of slews
Interconnect reduction

- RC-network
  - Distributed victim & aggressors
  - Soft-grounded
- Block-Arnoldi
  - Proven passivity
Driver output noiseless response

- Truncate $Z(s)$ to PI-Load
- 2 ODEs solved efficiently
  - 2nd order implicit numerical scheme (Crank-Nicholson)
    - Newton iterations employ predictor-corrector idea
Fast PWL transition

- BFS traverse of whole design
- One-stage cells:
  - Driver is modeled using Vivo model
- Multi-stage cells
  - Last CCC is modeled using Vivo model
  - Slew propagation table till last CCC
Voltage and current responses from each aggressor

\[ i_0 = \sum_k i_{0k}(t-t_k) \]

\[ \sum_k v_{jk}(t-t_k) \]

\[ v_{jk} = H_{jk} \cdot v_k \]

\[ Y_{0k} \cdot v_k \]

Short circuit current

Open circuit voltage

\[ Y(s) \]

\[ H(s) \]

\[ i_{0k} = Y_{0k} \cdot v_k \]

\[ \sum_k v_{jk}(t-t_k) \]

\[ Y_{0k} \cdot v_k \text{ and } H_{jk} \cdot v_k \text{ are found once for all alignments!} \]
Driver output response with X-Talk

\[ i_0 = \sum_{k} i_{0k}(t-t_k) \]

\[ V_i \quad V_o \quad I(t) \quad Z(s) \]

\[ Y(s) \quad H(s) \]

\[ \int I(t) \]
Propagate driver response across interconnect

Voltage Substitution

\[ v_{j0} = H_{j0} \cdot v_0 \]

\[ v_{j0} = H_{j0} \cdot v_0 \text{ is found once for each alignment!} \]

Linear superposition:

\[ v_j = v_{j0} + \sum v_{jk}(t-t_k) \]
ViVoSim

Driver output response
nonlinear solver

+ Receiver responses
linear solver

ViVoSim

ViVo

\[ V_i \]

\[ V_o \]

\[ Z(s) \]

\[ Y(s) \]

\[ H(s) \]

\[ V_{r1} \]

\[ V_{r2} \]
ViVoSim accuracy vs. Spice

VivoSim vs Spice
comparison of noiseless transitions

Input

Miller Effect

Drv

Rcv

r=200 Ohm
x=c=20 fF

1.1u/1.5u
ViVoSim accuracy on path

[Diagram of electrical circuit with graphs showing voltage over time]
Flow Diagram

- Fast PWL propagation
- Driver output response
- Receiver input response
- Receiver output response
- Aggressor responses
- Select alignment
- Driver output response
- Receiver input response
- Receiver output response

Noise less

noisy

Alignments, receivers

Delay changes
Alignment search

- Nonlinear optimization under constraints

- 2-step process
  - Aggressor cluster aligned with victim
    - Weighted average of individual alignments
  - Refinement: per each aggressor
    - Rarely needed
Results: 40K, 1.3um, 200sec Linux 2Gh

<table>
<thead>
<tr>
<th>late rise</th>
<th>late fall</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg err = 1.6%</td>
<td>avg err = 2.5%</td>
</tr>
<tr>
<td>stdev err = 2.4%</td>
<td>stdev err = 3.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>early rise</th>
<th>early fall</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg err = 3.2%</td>
<td>avg err = 2.8%</td>
</tr>
<tr>
<td>stdev err = 3.4%</td>
<td>stdev err = 3.4%</td>
</tr>
</tbody>
</table>
Pessimism reduction with ROP

- Path delay without noise:
  - 4ns

- Path delay with noise:
  - Rcv Input Probing (RIP):
    - 8.5ns
  - Rcv Output Probing (RIP):
    - 6ns

- Pessimism reduction
  - 2.5ns!
ROP vs RIP slacks

Worst RIP slack = -1.56ns
Worst ROP slack = -1.33ns

Missing noisy slew annotation?
Conclusions

- A new delay change analysis is based on:
  - Current model
  - Efficient linear and nonlinear solvers
  - PWL waves propagation
  - ROP
  - Nonlinear constrained optimization for alignment

- Accurate:
  - 2-3% off linear sweep in Spice

- Robust, due to:
  - Optimization
  - ROP

- Reduces pessimism, due to:
  - ROP
  - PWL on aggressors

- Fast:
  - Few hours for 1M gates design

- Implemented in CeltIC