56 DESIGN AUTOMATION CONFERENCE
FROM CHIPS TO SYSTEMS – LEARN TODAY, CREATE TOMORROW

Accurate and Correct-by-Construct Hold Margin Methodology for Standard cells

Gaurav Varshney, Rashmi Sachan, Tanvi Sharma

Texas Instruments

Sequential Cell Hold Margin

- Setup-time and Hold-time are two critical timing parameters for sequential cells (Latches and Flops) in standard cell library and are modelled as constraints arcs in Liberty (.lib) files.
- Both setup-time and hold-time are measured by running binary search based spice simulations and observing either delay push-out or by observing glitch at the output.
- It is important to account for random-variation (local mismatch) for hold-time to prevent silicon failure for a timing closed design. This is typically achieved by applying margin to the hold numbers in Library.

Introduction / Background

- Known methods of hold margining can be summed up as following empirical formula:
  - Hold margin = *a*(Inverter delay) + b*(CLK slew) + c*(Data slew) + d*(CLK→Q delay of the flop) + e*
  - Where a, b, c, d and e are weight factors obtained by running mismatch simulations on sample cells and fitting the equation.
- Same equation is applied across the corners and Clock & Data slew combinations and different flop types in a library.

Prior Art / Existing Methodology

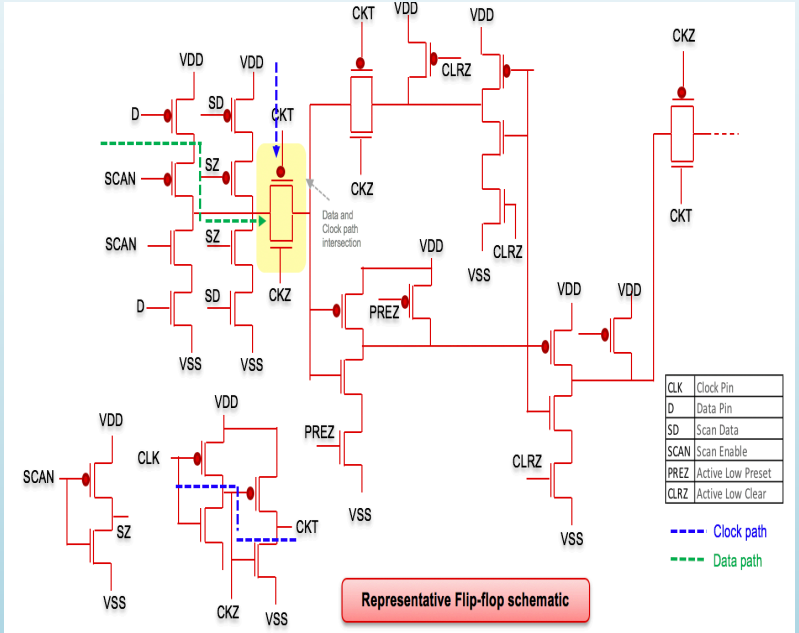
- **Limitations:**
  - Both inverter delay and CLK→Q delay components in the equation are approximations for the actual electrical effect.
  - Assumes all flops can be uniformly margined across functionalities, topologies and architectures.
  - Equation does not account for mixed-VT transistors (given prior assumption).
  - Assumes all flops can be uniformly margined across process/temp/voltage corners.
  - Risk of both over-margining and under-margining.
  - Manually applied and error prone.
- Alternatively, doing full blown mismatch simulations for hold margining would result into very high runtime.

Limitations / Challenges

Proposed Solution

- Hold-time of a flop is directly dependent on the Data path delay and clock path delay to a node where they intersect each other.
- Local mismatch variation seen on these paths will directly contribute to the variation in hold-time numbers.
- It is not necessary to have mismatch simulations of the hold measurement to find impact of local variation which is extremely compute intensive.
  - No of sims per timing arc = (number of data slews * number of clock slews * binary search iterations * Monte-Carlo runs)
- Instead, we measure local mismatch simulations on these paths to arrive at margin numbers for hold margining (Running simulations on partitioned/simple paths is much faster than running on full flop).
- These margin numbers are integrated in characterization environment, eliminating manual intervention.

Solution Overview



Data Path and Clock Path intersection for Hold Time

Solution Details

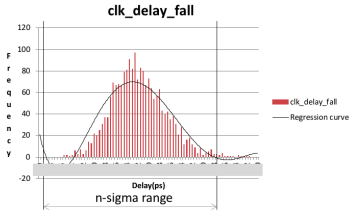


Figure: Clock to clock-node delay values calculated through MC sims

$$clk_{factor} = \frac{n * \text{standard deviation of } clk_{delay}}{\text{Mean of } clk_{delay}}$$

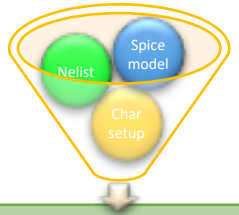
$$data_{factor} = \frac{n * \text{standard deviation of } data_{delay}}{\text{Mean of } data_{delay}}$$

$$data\ slew_{factor} = \text{derived using linear regression}$$

$$clock\ slew_{factor} = \text{derived using linear regression}$$

$$additional\ margin = fixed\ number\ (e.g\ 15ps)$$

Final Hold Margin Equation



Input Netlist

Process input netlist to find data path and clock path and intersection of data and clock paths which influence hold time measurement

Run Monte-Carlo simulations to find ***n-sigma*** standard deviation of identified clock and data path delays

Calculate weightages of clock path and data path based on ***n-sigma*** standard deviation of path delay. Fit results to get slew factor for data and clock slews

Create a TCL file consisting of internal node names for data path and clock path with their weight factors

TCL plug-in for internal path measurement, margin calculation and .lib margining

Characterization Tool

Data path and clock path

Monte-Carlo simulation results

Weight factors for data & clock paths including slews

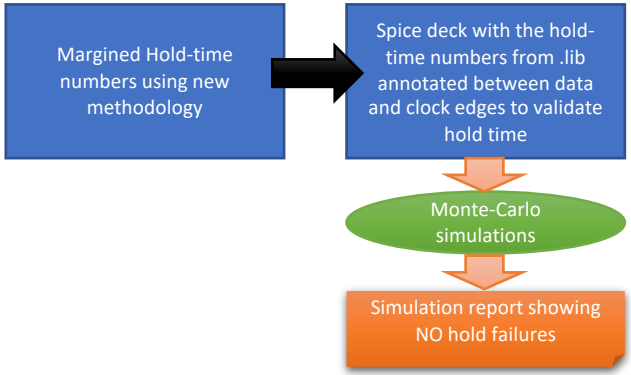
Hold margining TCL file

Output library with margined hold numbers with new methodology

$$Final\ hold\ margin = \boldsymbol{clk_{factor}} * clk_{delay} + \boldsymbol{data_{factor}} * data_{delay} + \boldsymbol{data\ slew_{factor}} * data_{slew} + \boldsymbol{clk\ slew_{factor}} * clk_{slew} + \boldsymbol{constant}\ (= \boldsymbol{15ps})$$

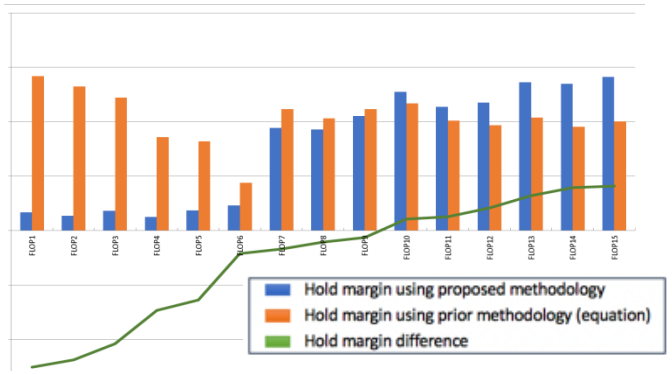
Note: Inverter delay and CLK→Q delay components have been removed, and components for Data and Clock path delay are added in the proposed equation. These components are measured for every corner and slews hence no approximation in terms of flop architecture/topology. This also ensures handling of mixed-VT transistors on data/clock path.

Evidence



- Proposed methodology has been validated by annotating hold-time numbers in a spice simulation decks and running Monte-Carlo mismatch simulations.
  - No hold failures detected in mismatch simulations.
  - Confirming margin added to hold is sufficient to account for local/random mismatch.
- All sequential cells and all hold arcs across all the corners have been validated.

Verification of proposed methodology



- Comparative study with proposed solution:
  - Negative number differences depicts the data points which are over margined with existing methodology and hence will lead to additional hold buffering in the design → area impact and cycle time impact.
  - Positive number differences depicts data points which are under margined and hence might cause silicon failure.

Results

Summary

- Paper presents an automated and correct-by-construct methodology for hold-margin calculation to account for random variation (local mismatch)
  - Method minimizes computational complexity without trading off accuracy by running mismatch simulations on data and clock paths.
  - Methodology can be easily integrated in native characterization tools as final formulated equation is simple enough to be coded as TCL plug-in.

- Proposed method should prevent any potential Silicon risk due to insufficiency of hold margins in the library.

- It should also result in better area and schedule improvement for the cases of over-margining (hold margined pessimistically) with the prior methodology.