Clustering-based signal merging in STA

Anton Belov, Adrian Wrixon, Maurice Keller, Himanshu Dadheech

Synopsys Inc.

TAU 2019

Monterey, CA, USA

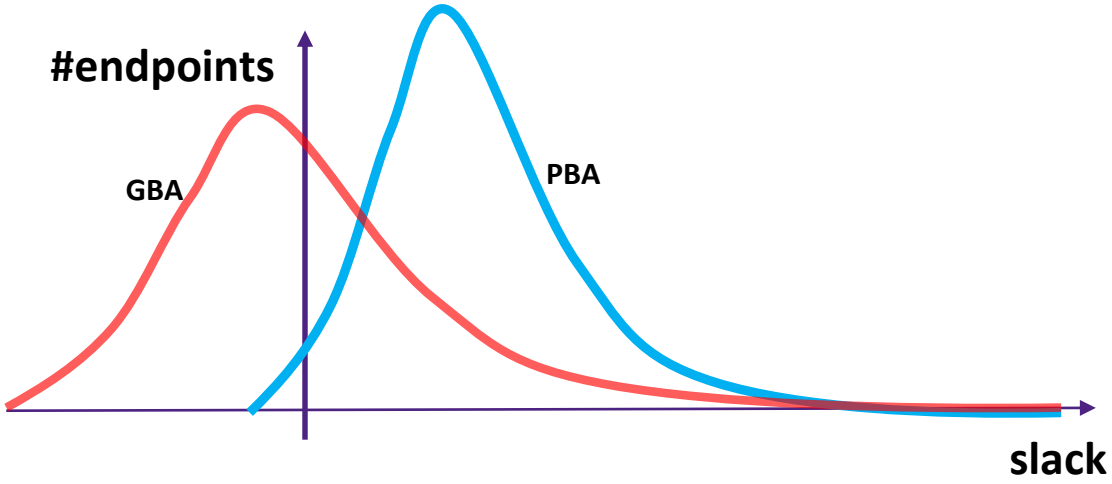
SYNOPSYS®

Silicon to Software™

Introduction: GBA-PBA accuracy gap (1/2)

- GBA (Graph-Based Analysis)
  - Timing values for the entire circuit are computed in a BFS sweep => *runtime/memory are linear.*
  - Timing properties are worst-cased (“merged”) at points of convergence => *slacks are pessimistic*.

- PBA (Path-Based Analysis)
  - Timing values are computed one path at a time => *runtime/memory can be exponential*.
  - No convergence => no merging => *slacks are accurate*.

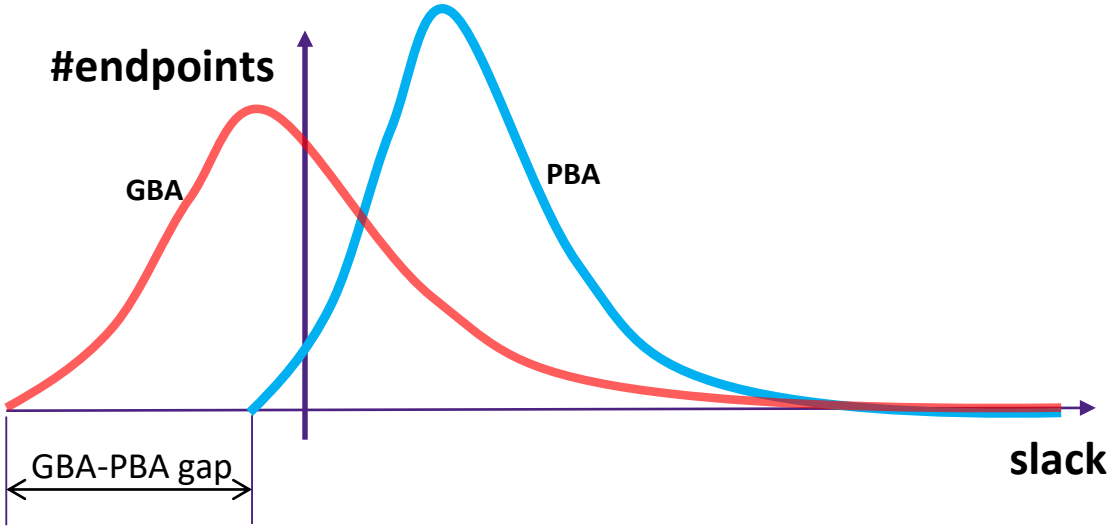


Introduction: GBA-PBA accuracy gap (1/2)

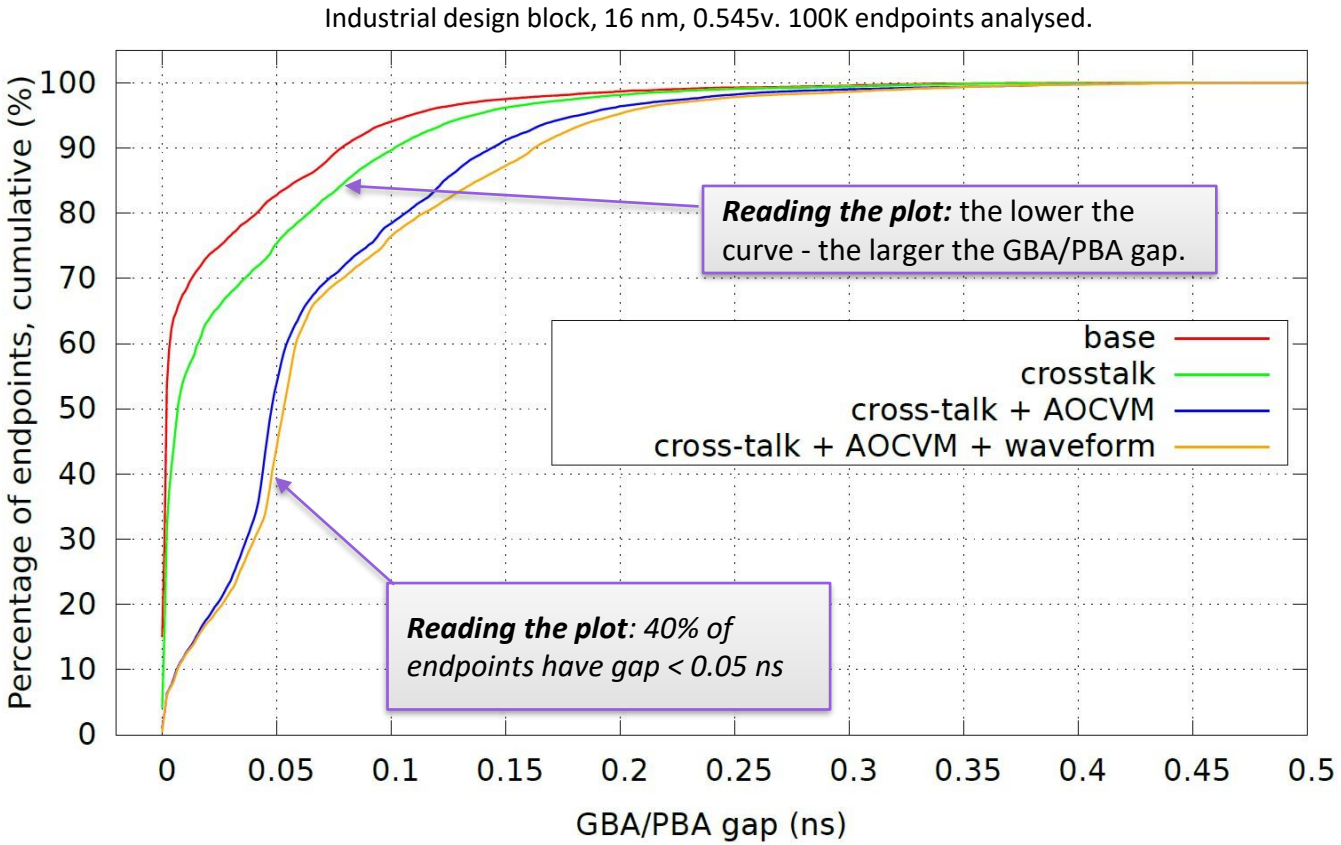
- GBA (Graph-Based Analysis)
  - Timing values for the entire circuit are computed in a BFS sweep => *runtime/memory are linear.*
  - Timing properties are worst-cased ("merged") at points of convergence => *slacks are pessimistic*.

- PBA (Path-Based Analysis)
  - Timing values are computed one path at a time => *runtime/memory can be exponential*.
  - No convergence => no merging => *slacks are accurate*.

- GBA-PBA gap
  - Difference between GBA and PBA slacks
  - Large GBA-PBA gap is a **problem:**
    - Slower and more memory intensive PBA-based signoff
    - Slower and less optimal ECO
  - GBA-PBA gap is getting worse …

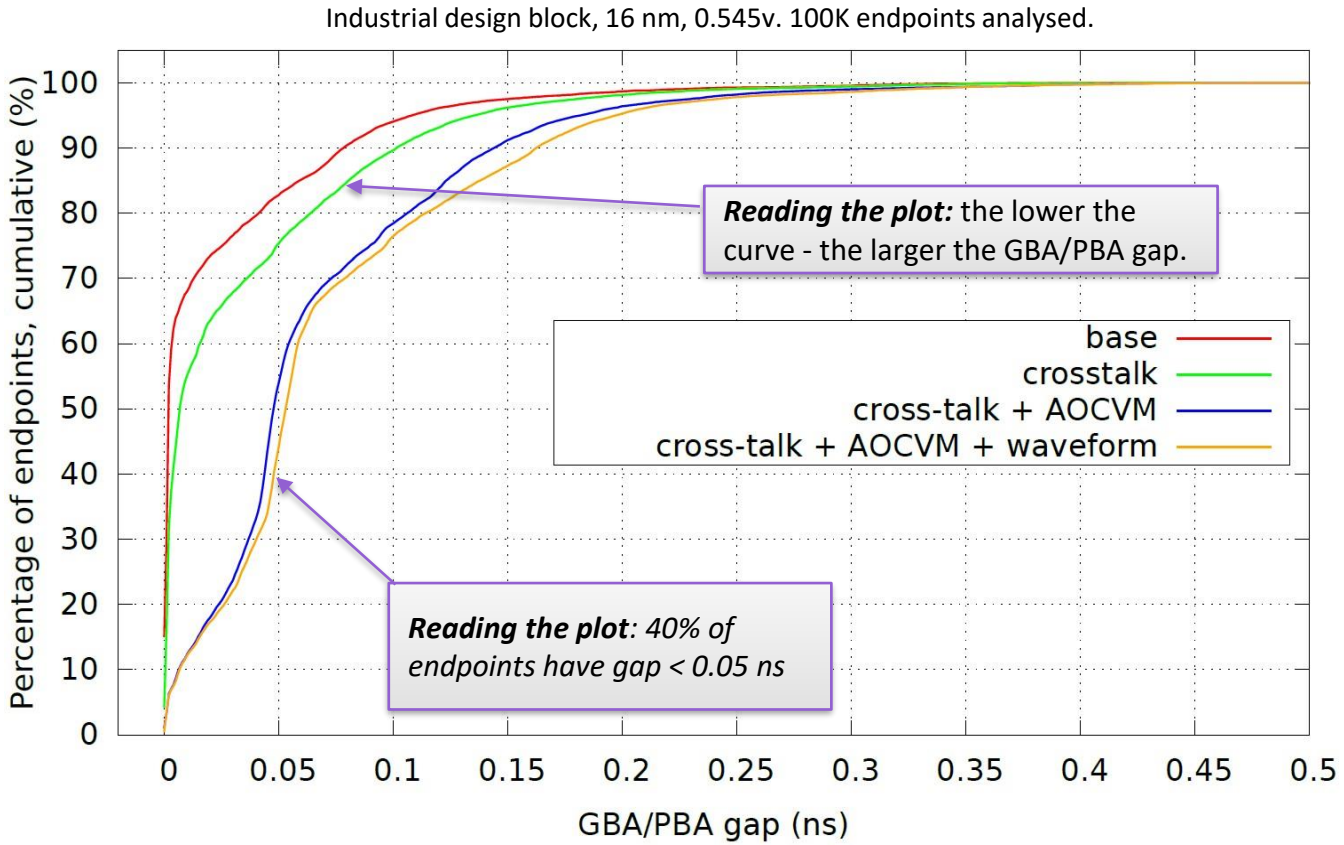


Introduction: GBA-PBA accuracy gap (2/2)



- GBA-PBA gap increases with the number of signal dimensions
  - Each dimension contributes merge pessimism

Introduction: GBA-PBA accuracy gap (2/2)



- GBA-PBA gap increases with the number of signal dimensions
  - Each dimension contributes merge pessimism

➢ **Problem:** how to improve accuracy of GBA ?
  - *Accuracy is lost in merging*
  - Approach 1: improve quality of merging
  - Approach 2: do less merging

Multiple-Signal Propagation

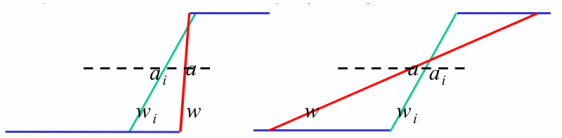
- Proposed in early 2000s (Blaauw, et al ICCAD 2000; Lee, et al ICCAD 2001)
  - Dominance: $S1$ dominates $S2$ at node n, if $at(S1) \geq at(S2)$ everywhere in fanout of n.
  - In some cases it is possible to detect dominance
  - In some cases it is possible to construct an accurate bounding signal
  - When neither is possible => propagate multiple signals.

Multiple-Signal Propagation

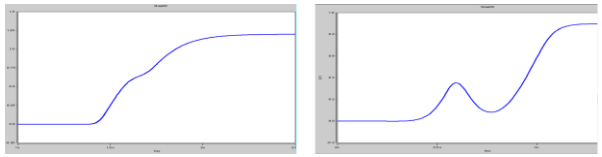
- Proposed in early 2000s (Blaauw, et al ICCAD 2000; Lee, et al ICCAD 2001)
  - Dominance: $S1$ dominates $S2$ at node n, if $at(S1) \geq at(S2)$ everywhere in fanout of n.
  - In some cases it is possible to detect dominance
  - In some cases it is possible to construct an accurate bounding signal
  - When neither is possible => propagate multiple signals.

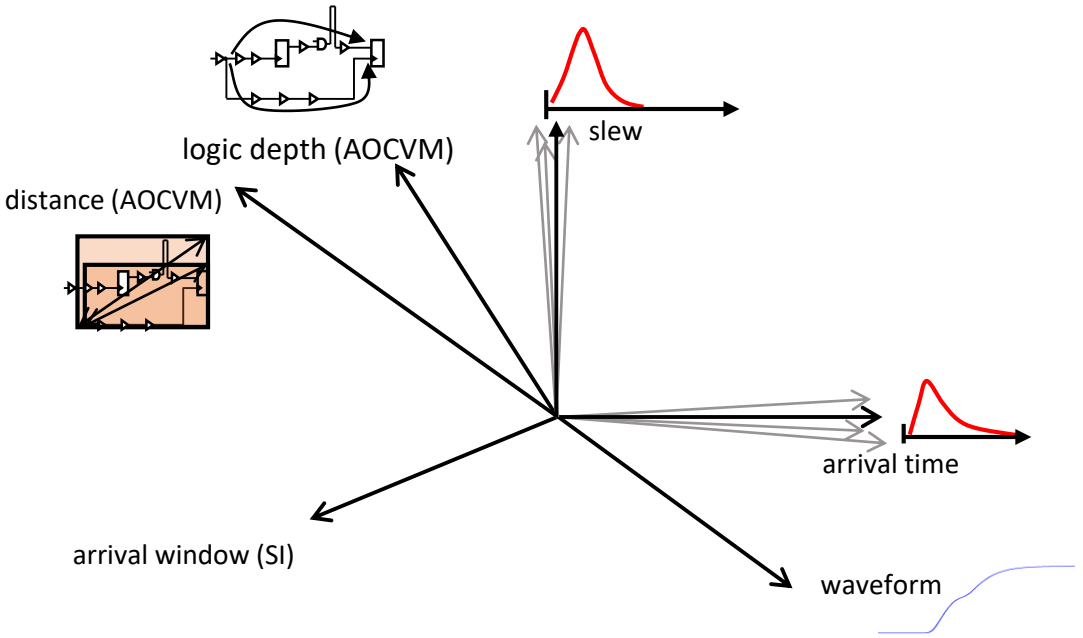
➢ **Problem:** old techniques do not translate

  - Signals were assumed to be 2-D: arrival time and slew



  - In modern STA signals are k-D



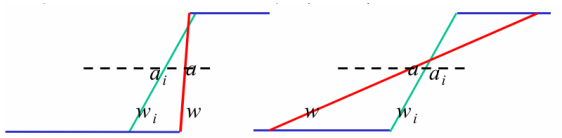


Multiple-Signal Propagation

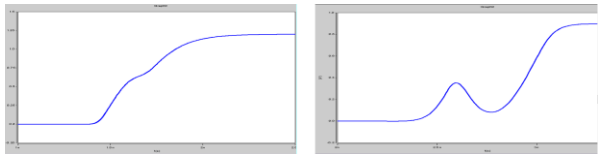
- Proposed in early 2000s (Blaauw, et al ICCAD 2000; Lee, et al ICCAD 2001)
  - Dominance: $S1$ dominates $S2$ at node n, if $at(S1) \geq at(S2)$ everywhere in fanout of n.
  - In some cases it is possible to detect dominance
  - In some cases it is possible to construct an accurate bounding signal
  - When neither is possible => propagate multiple signals.

➢ **Problem:** old techniques do not translate

- Signals were assumed to be 2-D: arrival time and slew

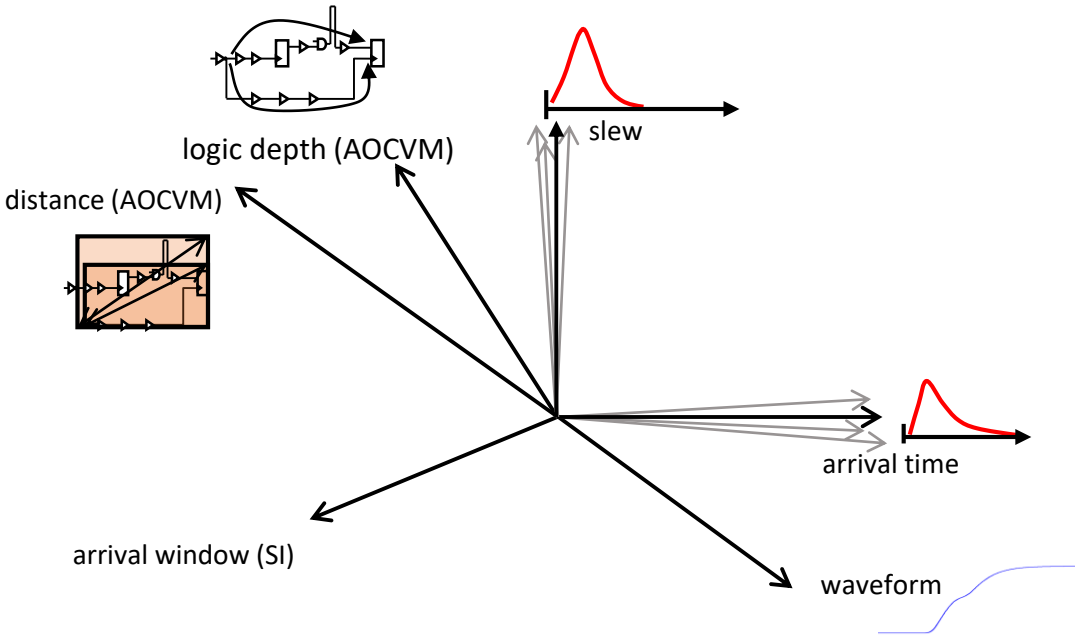


- In modern STA signals are k-D



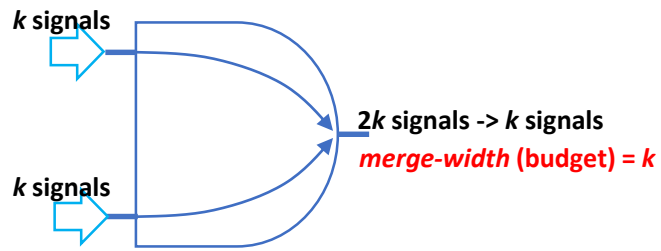
In this paper: focus on multiple-signal propagation

- How to maximize accuracy with a given runtime/memory budget ?

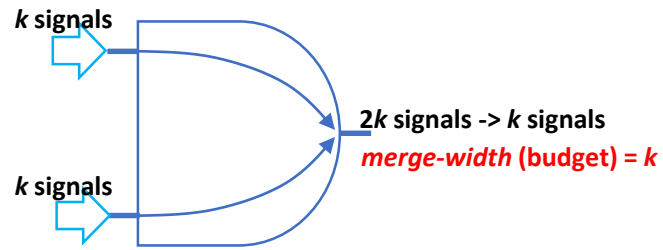


Clustering-based Signal Merging

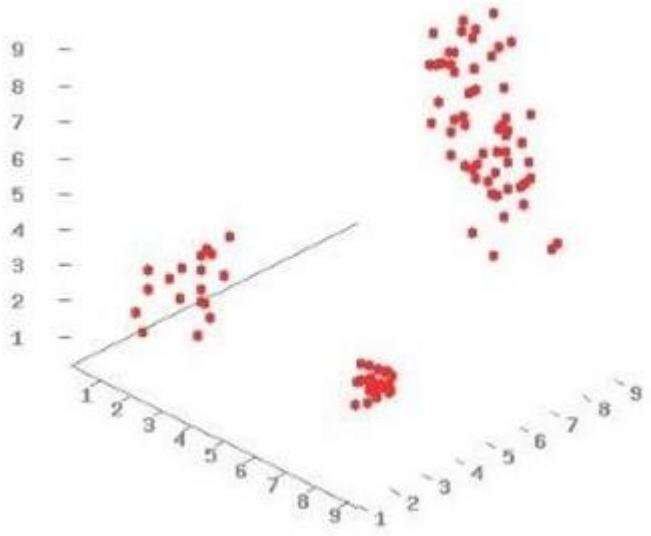
- Propagate multiple signals, but control resources
  - *merge-width (mw)* = the maximum number of unmerged signals per node

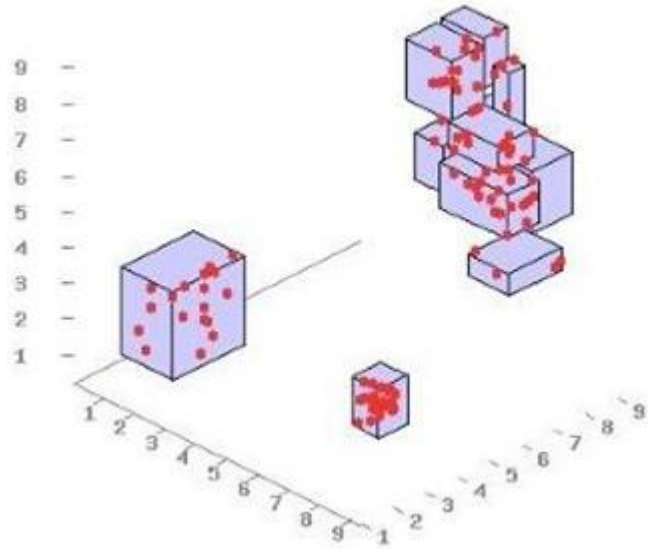


Clustering-based Signal Merging

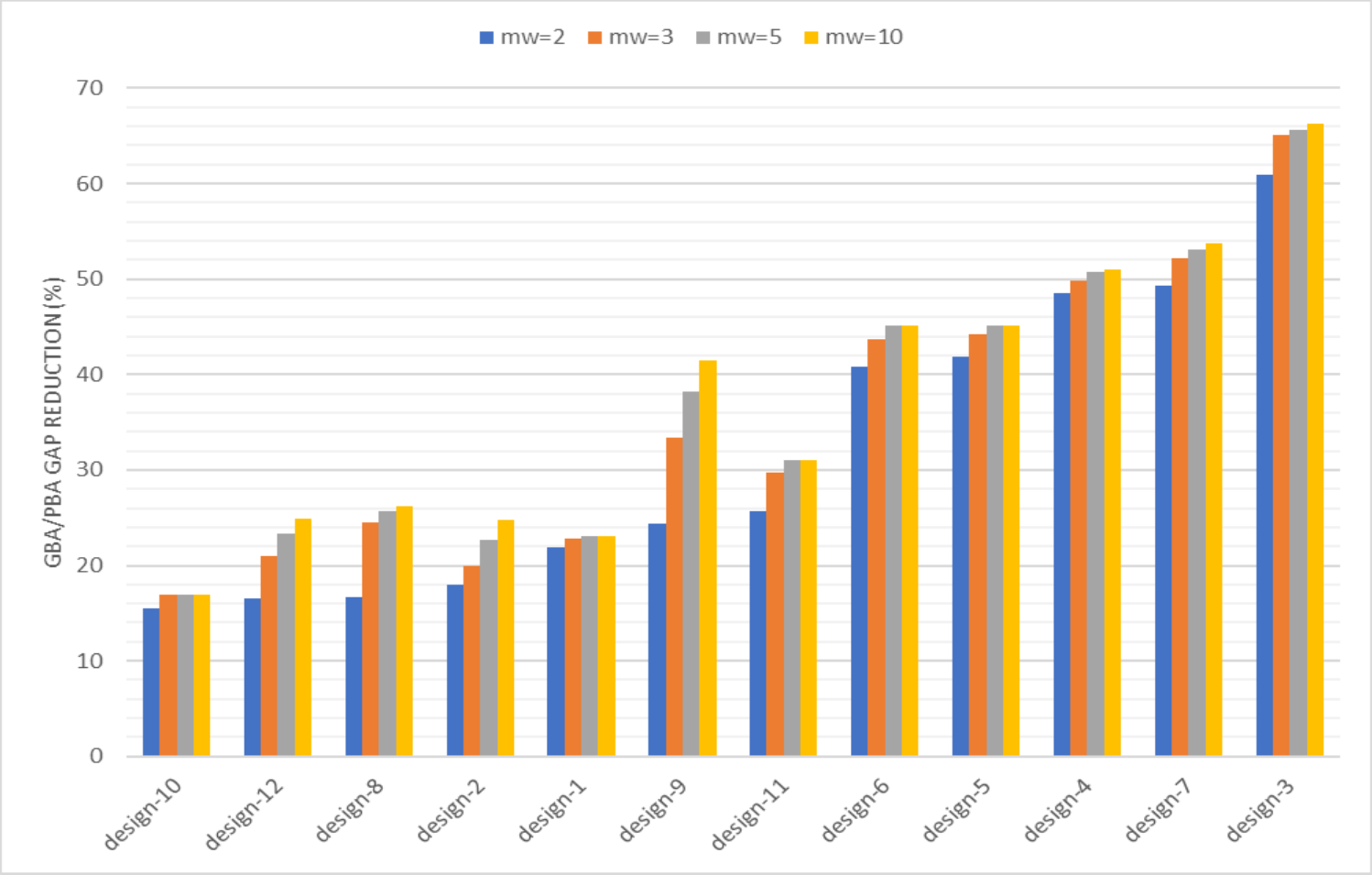


- Propagate multiple signals, but control resources
  - *merge-width (mw)* = the maximum number of unmerged signals per node
- When forced to merge – partition signals into *mw clusters* (subsets), but control accuracy:
  - Metric: *accuracy-loss* = endpoint arrival time difference unmerged vs merged
    - Infeasible to compute exactly, but can estimate heuristically
  - Translate all dimensions into arrival times; sensitivity is important
  - Find partition that minimizes overall accuracy-loss
  - Each cluster is merged pessimistically (safe)





Experimental results: GBA-PBA gap closure



- 12 blocks
  - 1M-3M instances
  - 7nm-20nm CCS
  - SI, Waveform, POCV and AOCV

➢ **Observations:**

- 20-60% gap closure (~35% avg)
- Sensitive to *merge-width,* but not always

Experimental results: PBA-based signoff

- PBA-based signoff requires computation of the worst PBA path for each violating endpoint
  - "exhaustive PBA"
- GBA-PBA accuracy gap has large impact on performance of exhaustive PBA

Experimental results: PBA-based signoff

- PBA-based signoff requires computation of the worst PBA path for each violating endpoint
  - "exhaustive PBA"
- GBA-PBA accuracy gap has large impact on performance of exhaustive PBA

Averages across all designs

	Runtime *x*-factor	Memory penalty
mw = 2	3.08*x*	17.6 %
mw = 3	3.21*x*	29.9 %
mw = 5	3.37*x*	47.9 %
mw = 10	3.32*x*	81.2 %

➢ **Observations**

- Highlight: 11.70x speedup, 11.8% memory penalty
- 3 designs with 4-5x speedup, under 20% memory penalty
- Lowlight: 1.06x speedup, 30.6% memory penalty
- Optimal merge width and benefits are design/technology dependent

Summary

- GBA-PBA accuracy gap is a (growing) problem
  - Signoff and ECO are impacted
- Possible solution: multiple-signal propagation with clustering-based merging
  - Experimental results are encouraging
  - Ripe for heuristics
  - Ripe for Machine Learning

Summary

- GBA-PBA accuracy gap is a (growing) problem
  - Signoff and ECO are impacted
- Possible solution: multiple-signal propagation with clustering-based merging
  - Experimental results are encouraging
  - Ripe for heuristics
  - Ripe for Machine Learning

Thank you !

SYNOPSYS®
Silicon to Software™