Calibration-Based Differentiable Timing Optimization in Non-linear Global Placement

Wuxi Li, Yuji Kukimoto, Gregory Servel, Ismail Bustany, Mehrdad E. Dehkordi
AMD Inc.
San Jose, CA, USA

wuxi.li@amd.com

Introduction & Motivation

Timing-driven placement (TDP) is crucial for design closure.

Net weighting *[Martin+,DAC'19] [Liao+, DATE'22][Liang+, arXiv'22]*

Net-length constraining *[Sarrafzadeh+, DAC'97][Halpin+, DAC'01][Hur+, DAC'03]*

Mathematical programming *[Hamada+, DAC'93][Swartz+, DAC'95][Chowdhary, DAC'05]*

Differentiable timing optimization (DTO) *[Naylor+, US Patent][Guo+, DAC'22]*

DTO is considered the state-of-the-art, demonstrating excellent effectiveness with reasonable efficiency.

Limited discussion on supporting timing exceptions and advanced timing analysis features in DTO.

This hinders its applicability to real industry designs.

Timing Exceptions & Common Path Pessimism Removal (CPPR)

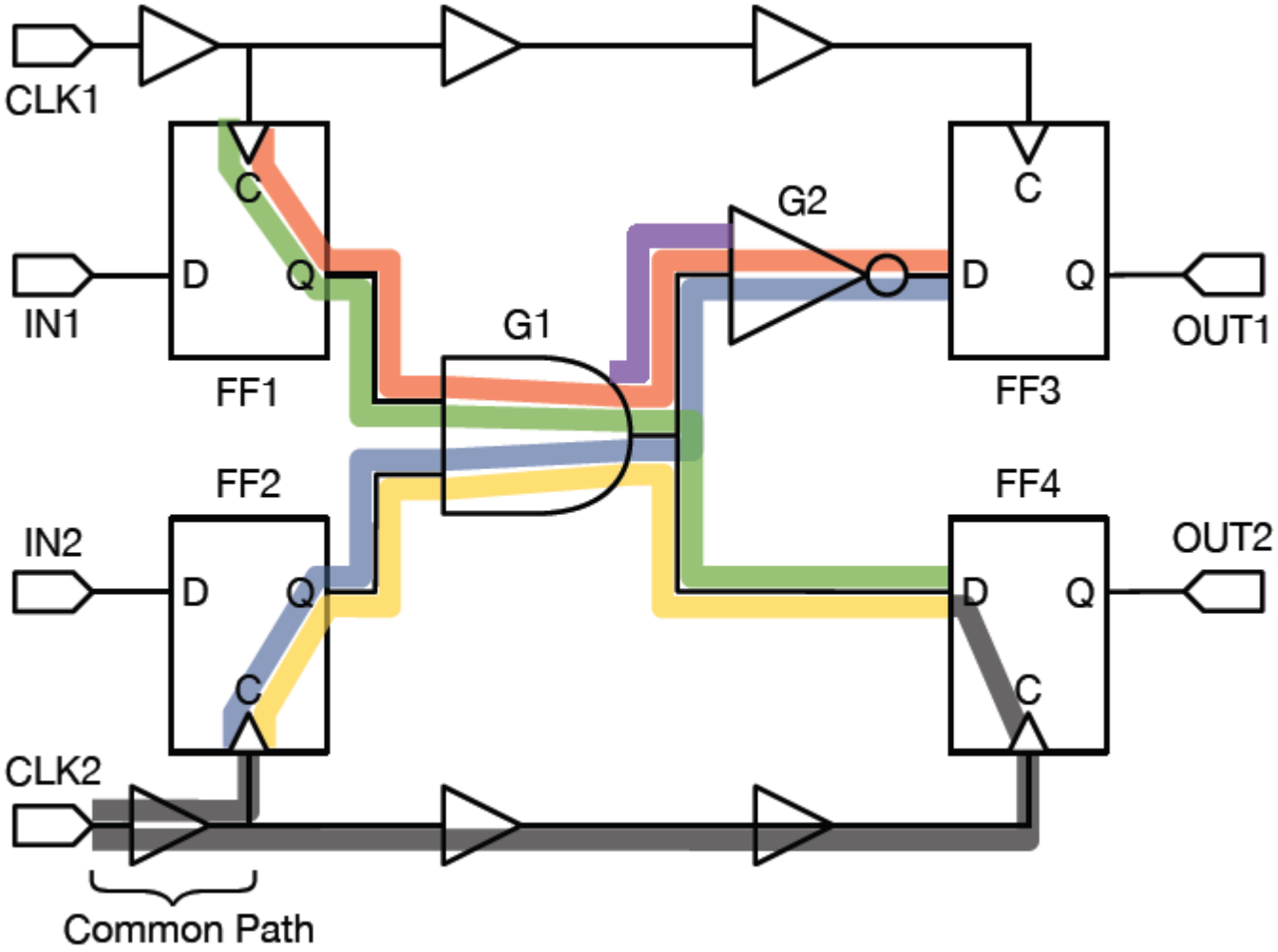
Clock Domain Crossing
FF1/C to FF4/D

Multi-Cycle Path
set_multicycle_path 2 -from FF1/C -to FF3/D

False Path
set_false_path -from FF2/C -to FF3/D

Path Segmentation
set_max_delay -from G1/O -to G2/I

Common Path Pessimism Removal (CPPR)
FF2/C to FF4/D



Tag-Based Static Timing Analysis (STA)

- Standard method to support timing exceptions and CPPR in STA.
- Each timing node maintains multiple (AAT, RAT, slack) tuples with different tags.
- Different tags represent different timing exceptions, CPPR, transition types, process corners, etc.
- Timing from different tags are propagated separately.

Tag Statistics of an Industry Design

Timing Exceptions	✗	✓	✓	✓	✓
CPPR	✗	✗	✓	✓	✓
Multi-Transition	✗	✗	✗	✓	✓
Multi-Corner	✗	✗	✗	✗	✓
#Tags / #Timing Nodes	1.00	2.22	2.51	4.24	7.86

One can potentially apply the same tagging mechanism to DTO.

Intractable development and maintenance efforts.

Unacceptable runtime and memory overhead.

Our Contributions

- We propose a timing calibration technique that can correlate a simple timer to a reference timer. The calibrated simple timer is ultrafast and timing exception/CPPR-aware.
- We implement a non-linear differentiable timing-driven global placement framework by extending the calibrated simple timer to a differentiable timing engine.
- We propose an angle-based cost-weighting scheme that dynamically balances various optimization objectives, leading to substantial improvement in solution quality and numeric convergence.
- We propose several techniques that notably reduce runtime with only minor or no compromise in solution quality.
- The proposed framework outperforms the latest Vivado in vital metrics, including maximum clock frequency, wirelength, routability, and back-end runtime, on a set of industry designs.

Proposed Overall Flow

Simple Timer (ST)

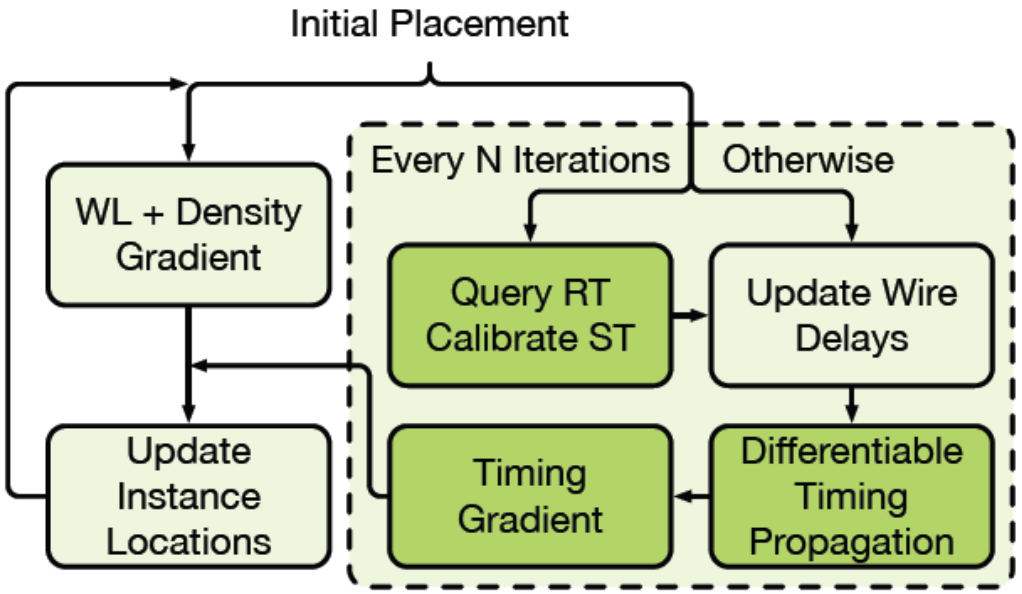
- Ultrafast
- Only support basic timing propagation

Reference Timer (RT)

- Accurate but slow (e.g., sign-off timer)
- Support advanced STA features like timing exceptions and CPPR

Key Idea

- Optimize timing using ST for runtime
- Calibrate ST using RT occasionally for accuracy



Timing Calibration Problem

Inputs:

A timing graph, a ST, and a RT.

DLY_j (delay) of all timing arcs in ST.

$SLK_i^\star$ (worst slack) of all timing nodes in RT.

Outputs:

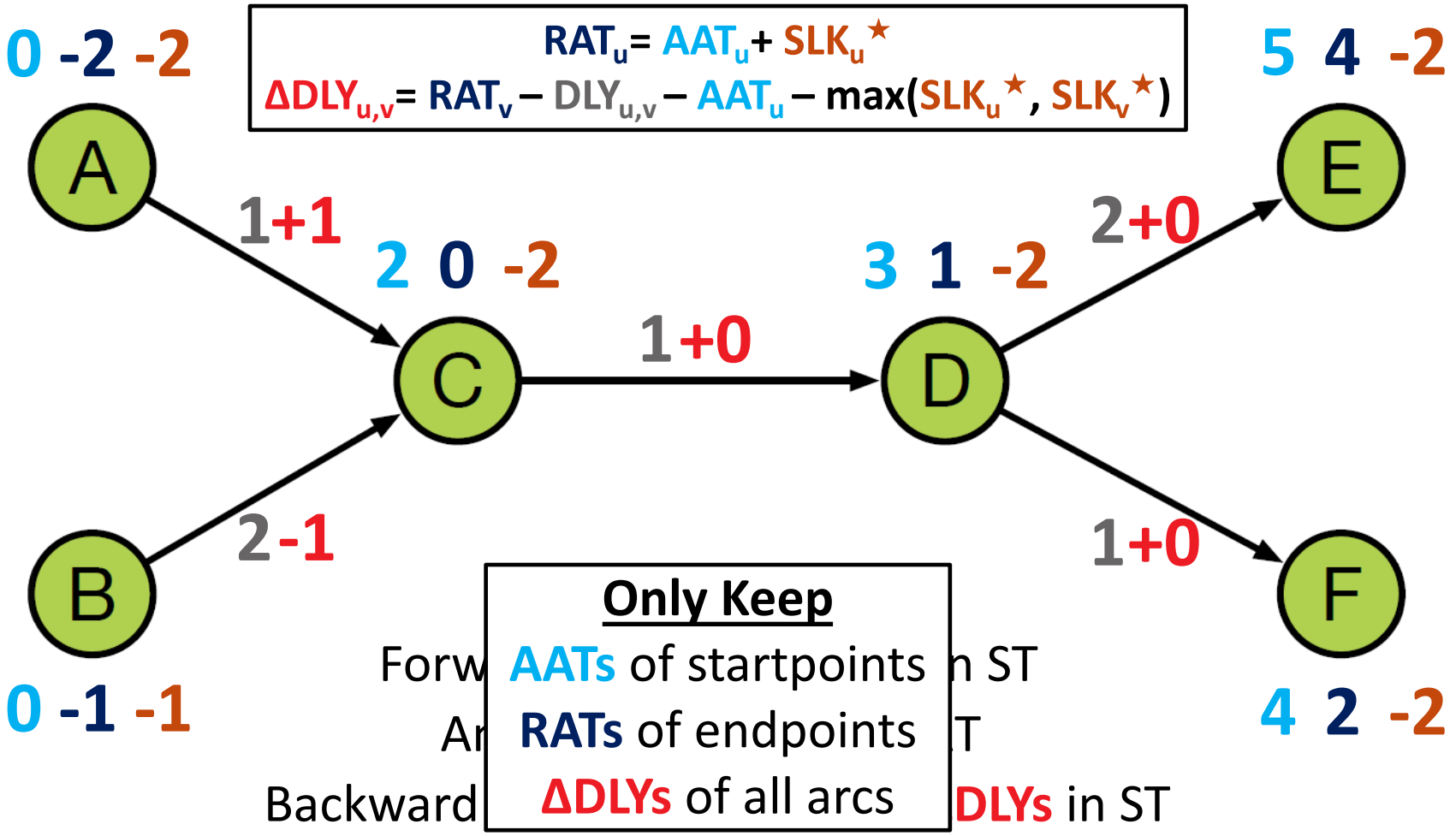
RAT_i (required arrival time) of all endpoints in ST.

ΔDLY_i (delay correction) of all timing arcs in ST.

Goal:

Exactly matching SLK_i (slack in ST) to $SLK_i^\star$ for all timing nodes when using $DLY_j + \Delta DLY_j$ instead of DLY_j in ST.

Timing Calibration Algorithm



Properties of the Calibrated ST

Property 1

The calibrated ST can exactly match the slacks from RT at the placement where the calibration is conducted.

Property 2

The slack discrepancy between the calibrated ST and RT at placement $\boldsymbol{x}$ (vector of instance locations) is bounded by $O(\|\boldsymbol{x} - \boldsymbol{x}^{\star}\|^{p})$, where $\boldsymbol{x}^{\star}$ denotes the placement where the calibration is conducted, and p is a constant dependent on the wire delay model.

Property 3

Many placement-invariant/insensitive timing factors can be captured through the calibration.

Properties 3 is the key of the lightweightness of the calibrated ST. Numerous sources of inaccuracy in ST are placement-invariant/insensitive, e.g., intra-instance delays, multi-cycle path, CPPR etc.

By virtue of Property 3, these placement-invariant/insensitive timing factors do not necessitate modeling in ST, as they call can be captured by the calibration.

Experimental Setup

Implementation

C++

elfPlace *[Li+, ICCAD'19]*

Cpp-Taskflow *[Huang+, IPDPS'19]*

Machine

AMD EPYC 7F52 16-Core 3.5 GHz CPUs

512 GB memory

Evaluation

AMD Vivado 2023.2

7nm Versal architecture

155 industry designs

Benchmark Statistics

Percentile	Min	25th	50th	75th	Max
#Instances (K)	8	181	314	635	1415
#Nets (K)	10	228	415	738	1589
#Pins (M)	0.05	1.14	1.99	3.71	7.61
#Timing Nodes (M)	0.06	1.44	2.71	4.11	9.29
#Timing Arcs (M)	0.08	2.26	5.14	8.35	28.69
#Clocks	1	2	6	31	173
#set_multicycle_path	0	0	0	16	2016
#set_false_path	0	0	13	261	6986
#set_max_delay	0	0	0	52	3998
#Tags per Timing Node	2.38	2.80	3.03	5.13	10.09

Main Results

Vivado 2023.2

Default timing-driven mode of the latest Vivado.

Ours Net-Wt

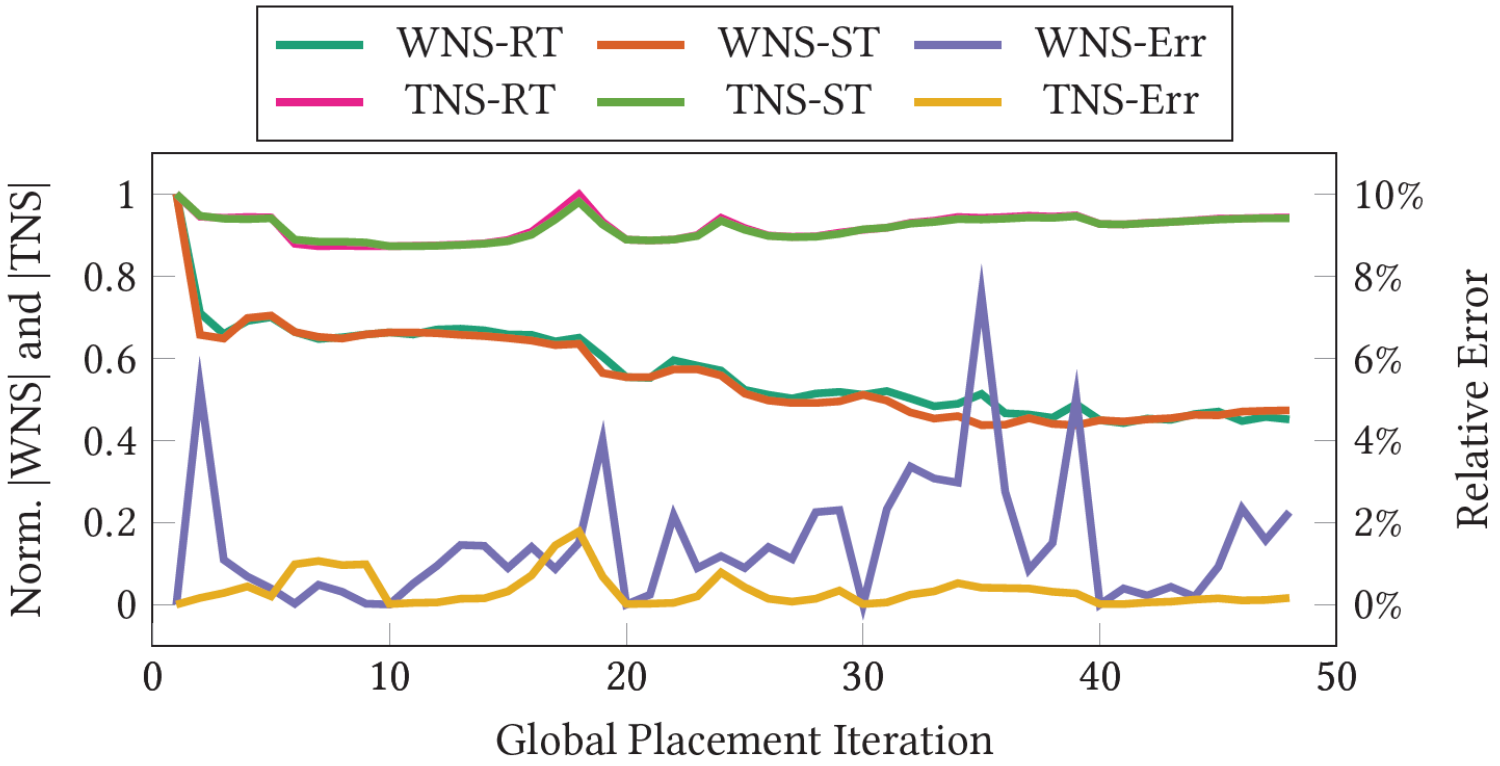
Our implementation of net weighting-based timing optimization, similar to *[Martin+,DAC'19],* **DREAMPlace 4.0** *[Liao+, DATE'22],* **AMF-Placer 2.0** *[Liang+, arXiv'22]*

Ours

Proposed calibration-based differentiable timing optimization。

Designs	Vivado 2023.2		Ours Net-Wt		Ours	
	All	Large	All	Large	All	Large
GP-Fmax	0.931	0.928	0.919	0.911	**1.000**	**1.000**
Place-Fmax	0.971	0.955	0.982	0.980	**1.000**	**1.000**
Route-Fmax	0.981	0.972	0.991	0.993	**1.000**	**1.000**
GP-WL	1.311	1.332	1.326	1.263	**1.000**	**1.000**
Place-WL	1.137	1.151	1.139	1.159	**1.000**	**1.000**
Route-WL	1.085	1.091	1.074	1.095	**1.000**	**1.000**
GP-RT	**0.946**	1.078	1.109	1.078	1.000	**1.000**
Place-RT	**0.976**	1.065	1.030	1.065	1.000	**1.000**
Route-RT	1.078	1.097	1.040	1.059	**1.000**	**1.000**
Total-RT	1.063	1.090	1.049	1.063	**1.000**	**1.000**
#Unroutes	10	9	8	7	**6**	**5**

Effectiveness of Timing Calibration



- Calibrate every 10 GP iterations
- **ZERO** slack error after calibration
- Max WNS error < 8%
- Max TNS error < 2%

Conclusion

- We propose a timing calibration technique that correlates a simple timer (ST) to a reference timer (RT).
- The ST can efficiently handle timing exceptions and CPPR by periodic calibration.
- We extend the calibrated ST to a differentiable timing optimization engine in global placement.
- The proposed framework outperforms the latest Vivado in vital metrics, including Fmax, wirelength, routability, and back-end runtime, on a set of industry designs.


AMD