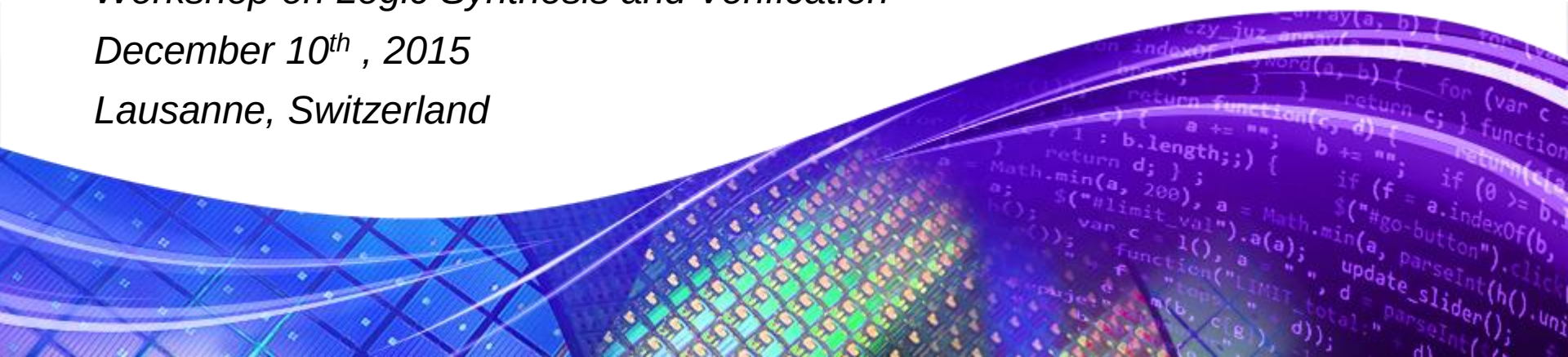
SYNOPSYS®
Silicon to Software™

Fast Synthesis

DC Explorer Perspective

Jovanka Ciric Vujkovic, Ph.D.
Senior Manager R&D, Design Group, Synopsys Inc.

Workshop on Logic Synthesis and Verification
December 10th , 2015
Lausanne, Switzerland



Outline

Fast Synthesis in Today's Designs

Fast Synthesis in DC Explorer

Synthesis Flow With Fast Gate Sizing

Conclusion and Future Work

Fast Synthesis in Today's Designs

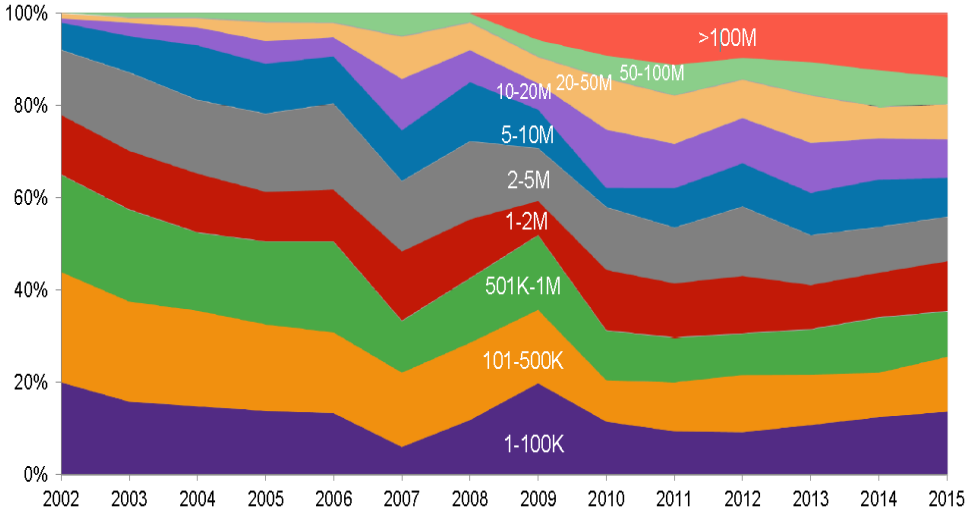
SYNOPSYS®

Growing Complexity Requires Rethinking of Design Strategy

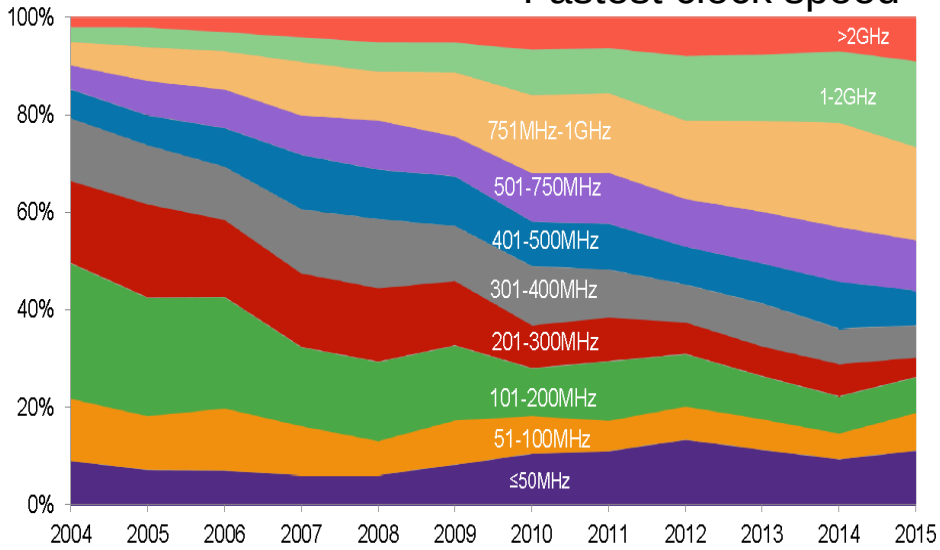
Today's Challenges

- In North America, about 20% of the designs exceed 100M gates
- Higher clock speed trend
- Multiple voltage domains
- Complex SDC constraints
- Large number of IPs
- Schedules of 15 months or less

Gate count of IC designs



Fastest clock speed

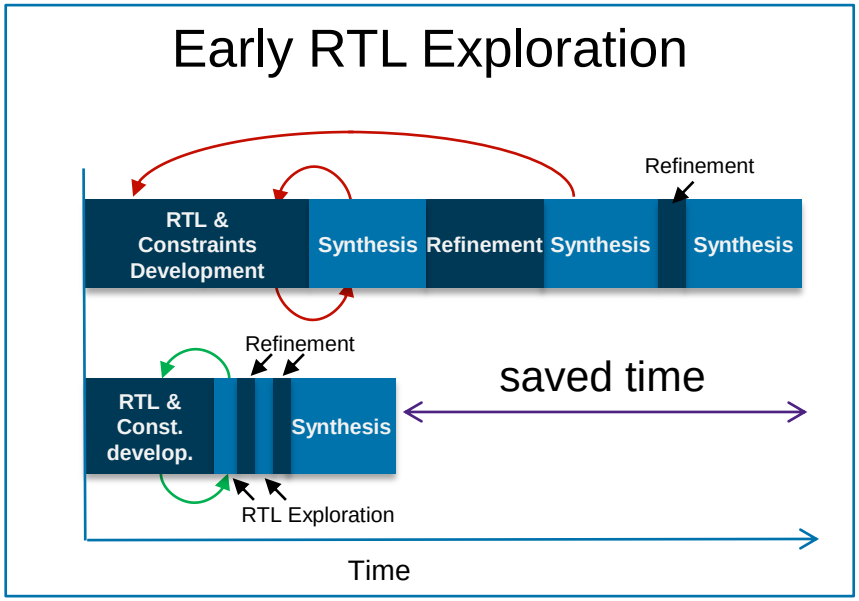


Source: Synopsys' Global User 2015 Survey

SYNOPSYS®

Growing Design Complexity Requires Fast and Early Exploration

- During early design stages
  - RTL and constraints are incomplete
  - Many blocks and 3rd-party IPs are incomplete or unavailable
  - Floorplan is unavailable or preliminary



- Need an efficient way to:
  - Resolve data inconsistencies
  - Debug timing constraints
  - Improve design data

- Fast synthesis requires high-quality netlists and reduces design schedules

SYNOPSYS®

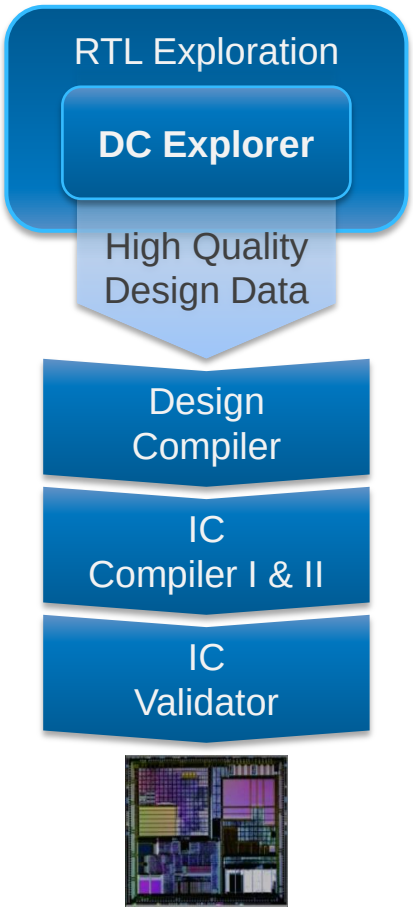
Fast Synthesis in DC Explorer

SYNOPSYS®

DC Explorer in Design Cycle

Better Starting Point For RTL Synthesis

- Tolerance for incomplete data
  - Faster RTL and constraints development
  - Pre-Synthesis floorplanning
- 5-10X faster runtime compared to final RTL synthesis
  - Quick what-if analyses
- Physical implementation
  - Reading floorplans
  - Congestion-driven placement
  - Physically aware optimizations
- 8% timing and area correlations
  - Early visibility into synthesis results

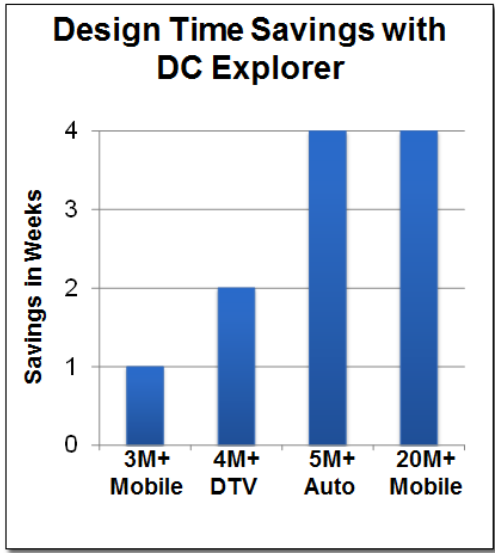


SYNOPSYS®

DC Explorer

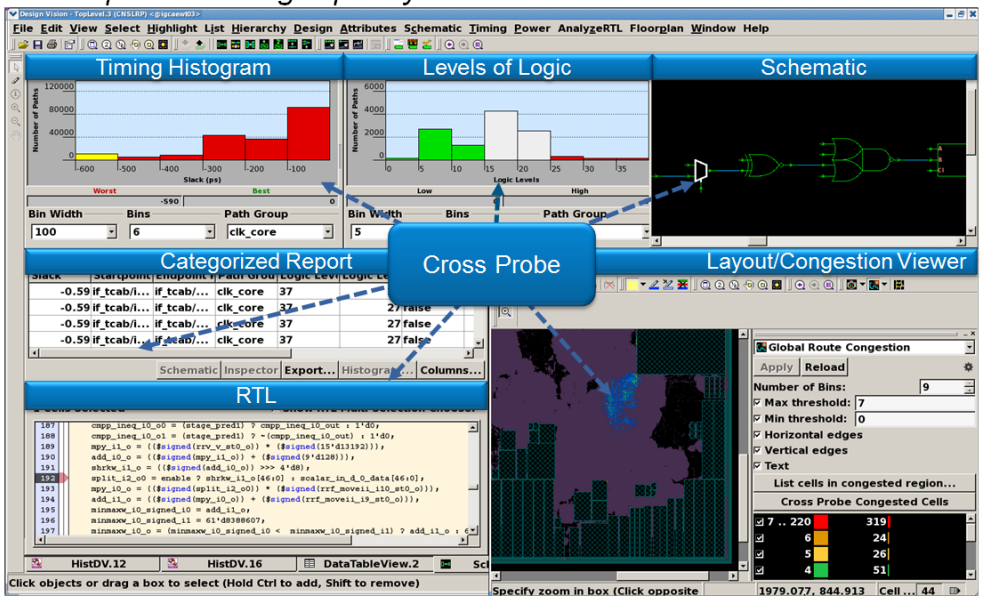
Early Design Exploration

- Up to one month faster schedule
- Early visibility
  - Tolerance for incomplete data
  - Low-power support
  - Floorplan exploration
- Debug and RTL cross-probing
  - Timing analysis
  - Logic-level analysis
  - Congestion analysis



RTL Debug & Cross Probing

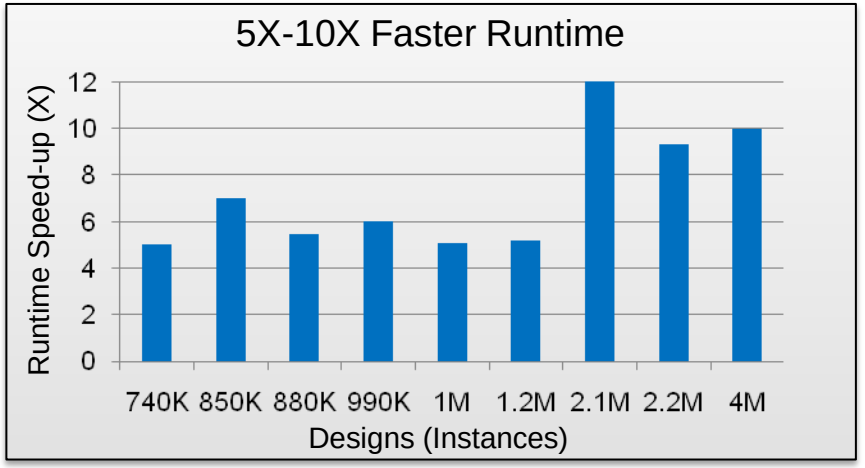
Helps create high quality RTL

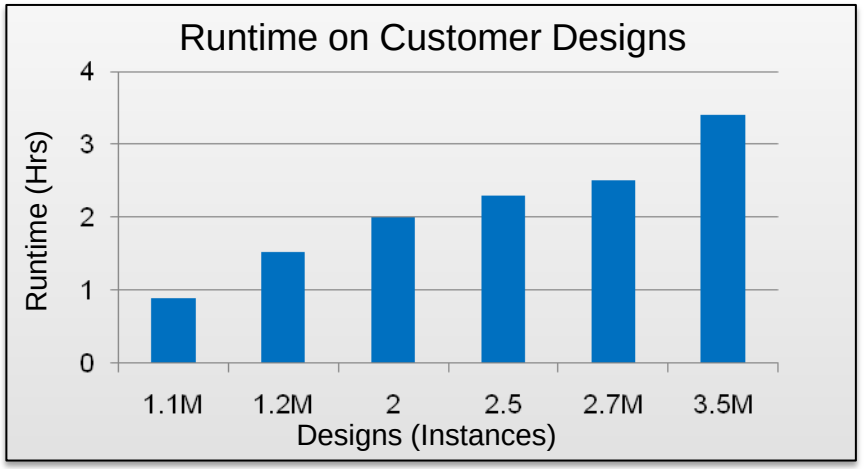


SYNOPSYS®

5-10X Faster Runtime Compared to Final RTL Synthesis

- New fast optimization technology
  - Unaffected by the quality of constraints
  - Multicore support delivers additional 2X faster runtime on 4 cores
- Enabling rapid what-if explorations of design configurations

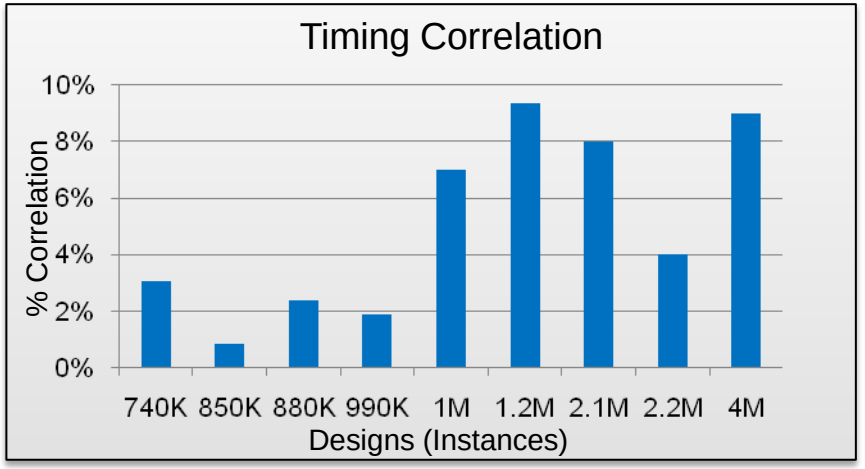


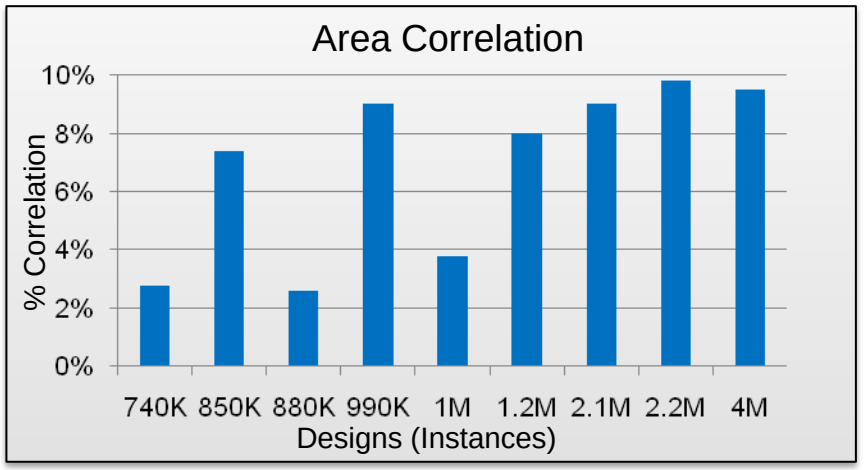


SYNOPSYS®

8% Correlation with Design Compiler

- Assess the likelihood of meeting design targets
- Support power and test
  - Clock gating, %LVT leakage optimizations, scan insertion, and test DRC checks
- Identify potential improvements before implementation
  - Datapath architecture





SYNOPSYS®

How to Design a Fast Synthesis Flow

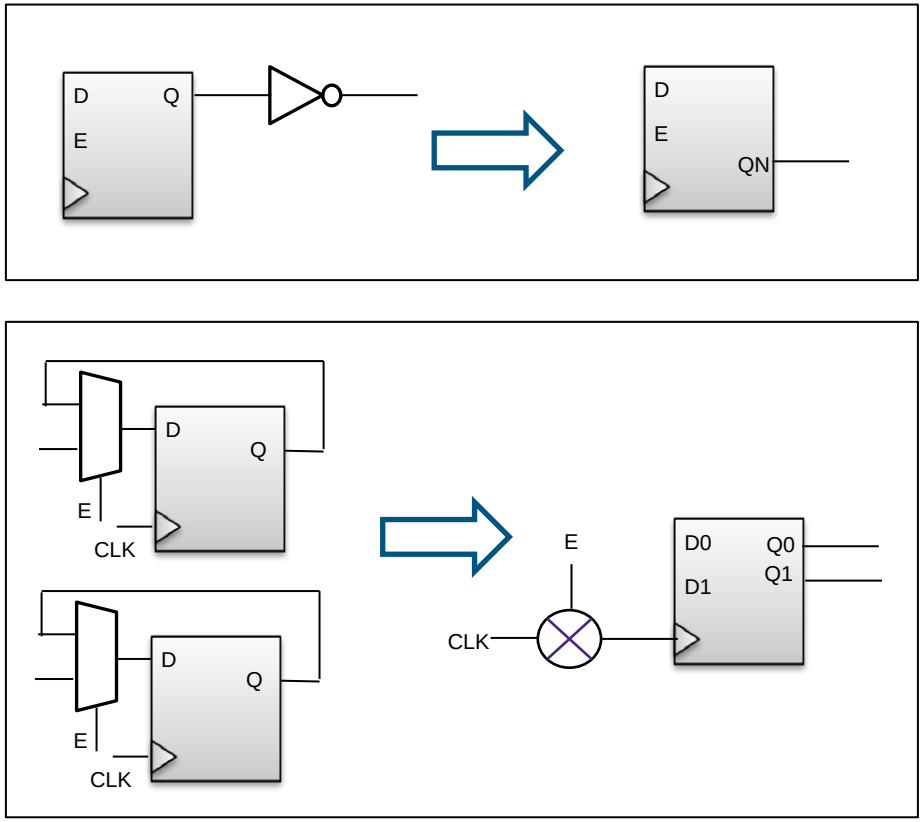
- Principles for achieving faster runtime
  - Re-think old and devise new faster algorithms
  - Create a convergent flow
  - Approximate only when QoR impact is minimal
  - Reduce effort for iterative algorithms
  - Exploit design characteristics

- Have state-of-the-art Design Compiler Graphical reference flow
  - Important to achieve tight correlation with final synthesis
  - No missing functionalities

No Compromise on Features

Fast synthesis in DC Explorer supports all major optimizations and engines

- Combinational Optimizations for timing and area
  - Boundary optimization
  - Constant propagation
  - Datapath extraction and optimization
- Sequential optimizations
  - Sequential output inversion
  - Unloaded and constant register removal
  - Register merging
  - Retiming
- Clock gating
- Combinational and sequential gate sizing
- High-fanout buffering
- Congestion-driven placement
- Multicorner multimode
- Multibit register mapping



SYNOPSYS®

Synthesis Flow with Fast Gate Sizing

SYNOPSYS®

Gate Sizing

- Gate sizing optimization assigns gate sizes to all cells in the design using the technology library model for each gate to meet timing constraints with minimal area and power

- Prior work in gate sizing:
  - Continuous methods
    - Convex nonlinear optimization (numerical formulation, Lagrangian relaxation)
    - Linear programming and network flow
    - Slew budgeting
  - Discrete methods
    - Sensitivity-based iterative methods
    - Dynamic programming
    - Branch and bound

- Gate sizing algorithm in DC Explorer is based on numerical synthesis

Patented Technology for Gate Sizing

(12) **United States Patent**
Mottaez et al.

(10) **Patent No.: US 9,171,122 B2**
(45) **Date of Patent: Oct. 27, 2015**

(54) **EFFICIENT TIMING CALCULATIONS IN NUMERICAL SEQUENTIAL CELL SIZING AND INCREMENTAL SLACK MARGIN PROPAGATION**

(71) Applicant: **Synopsys, Inc.**, Mountain View, CA (US)

(72) Inventors: **Amir H. Mottaez**, Los Altos, CA (US); **Mahesh A. Iyer**, Fremont, CA (US)

(73) Assignee: **SYNOPSYS, INC.**, Mountain View, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **13/691,480**

(22) Filed: **Nov. 30, 2012**

(65) **Prior Publication Data**

US 2013/0145339 A1 Jun. 6, 2013

Related U.S. Application Data

(60) Provisional application No. 61/566,464, filed on Dec. 2, 2011.

(51) **Int. Cl.**
G06F 9/455 (2006.01)
G06F 17/50 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 17/5081*** (2013.01); ***G06F 17/50*** (2013.01); ***G06F 17/505*** (2013.01); *G06F 2217/84* (2013.01)

(58) **Field of Classification Search**
USPC 716/108, 113, 134
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,526,543	B1 *	2/2003	Rosser	716/104
7,844,931	B2 *	11/2010	Hutzl et al.	716/123
8,060,844	B2 *	11/2011	Gidon et al.	716/101
8,539,413	B1 *	9/2013	Singla et al.	716/113
2001/0029600	A1 *	10/2001	Lee et al.	716/17
2002/0116685	A1 *	8/2002	van Ginneken	716/1
2006/0095879	A1 *	5/2006	Brahme et al.	716/6
2008/0216042	A1 *	9/2008	Hutzl et al.	716/18
2009/0089721	A1 *	4/2009	Alpert et al.	716/2
2012/0284680	A1 *	11/2012	Iyer et al.	716/113
2013/0043923	A1 *	2/2013	Gasper et al.	327/281
2013/0145339	A1 *	6/2013	Mottaez et al.	716/134

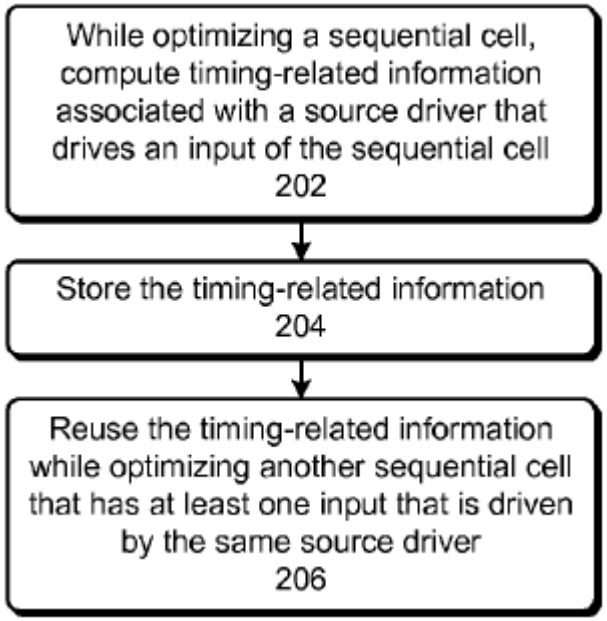
* cited by examiner

Primary Examiner — Brian Ngo
(74) *Attorney, Agent, or Firm* — Park Vaughan Fleming & Dowler LLP; Laxman Sahasrabuddhe

(57) **ABSTRACT**

Techniques and systems are described for improving the efficiency of timing calculations in numerical sequential cell sizing and for improving the efficiency of incremental slack margin propagation. Some embodiments cache timing-related information associated with a source driver that drives an input of a sequential cell that is being sized, and/or timing-related information for each output of the sequential cell that is being sized. The cached timing-related information for the source driver can be reused when sizing a different sequential cell. The cached timing-related information for the outputs of the sequential cell can be reused when evaluating alternatives for replacing the sequential cell. Some embodiments incrementally propagate slack margins in a lazy fashion (i.e., only when it is necessary to do so for correctness or accuracy reasons) while sizing gates in the circuit design in a reverse-levelized processing order.

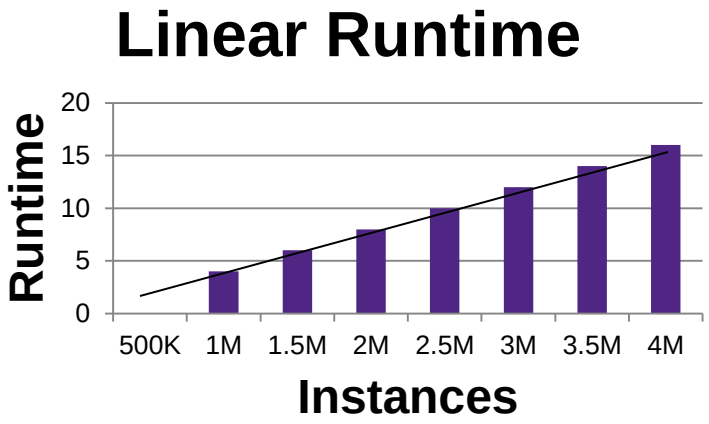
15 Claims, 3 Drawing Sheets

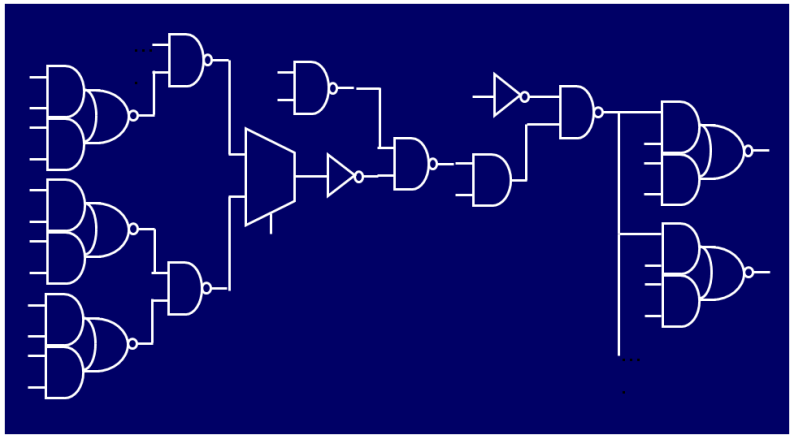


SYNOPSYS®

Numerical Synthesis

- Constraint-invariant synthesis
  - Linear runtime in the size of the design
  - Works on entire design instead of few critical paths
- Numerical synthesis
  - Advanced size-independent library modeling enables numerical formulation
  - Optimal solution using state-of-the-art numerical solvers
  - Works for sequential and combinational logic





Numerical Formulation = f(library cell timing, library pin cap, path stages, endpoint loading, startpoint cap, and so on)

SYNOPSYS®

Numerical Delay Modeling Basics

- Theory of Logical Effort by Sutherland *et al.* 1999
  - *g : logical effort*
  - *h : electrical effort*
  - *p : parasitic delay of gate*

$$d = g \times h + p$$

- Can be rewritten as:

$$d = (R \times C_i) \times \left(\frac{C_o}{C_i}\right) + p$$

- Derivation of *g* and *p* for a library cell:
  - Not perfectly linear
  - Different delays for rise and fall times
  - Variance between different timing arcs
  - Slope variance for different input transitions

- Library analysis requires clustering and handling outliers for *g* and *p* derivation

Gate Sizing Algorithm

(12) **United States Patent**
Iyer et al.

(10) **Patent No.: US 8,966,430 B1**
(45) **Date of Patent: Feb. 24, 2015**

(54) **ROBUST NUMERICAL OPTIMIZATION FOR OPTIMIZING DELAY, AREA, AND LEAKAGE POWER**

(71) Applicant: **Synopsys, Inc.**, Mountain View, CA (US)

(72) Inventors: **Mahesh A. Iyer**, Fremont, CA (US); **Amir H. Mottaez**, Los Altos, CA (US)

(73) Assignee: **Synopsys, Inc.**, Mountain View, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 14 days.

(21) Appl. No.: **13/954,923**

(22) Filed: **Jul. 30, 2013**

(51) **Int. Cl.**
G06F 17/50 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 17/5072*** (2013.01)
USPC **716/133**

(58) **Field of Classification Search**
CPC G06F 17/30; G06F 17/50
USPC 716/133
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2010/0169846	A1 *	7/2010	Gupta et al.	716/2
2012/0221995	A1 *	8/2012	Zahn et al.	716/134
2013/0254734	A1 *	9/2013	Gupta et al.	716/133
2014/0040845	A1 *	2/2014	Zahn et al.	716/114

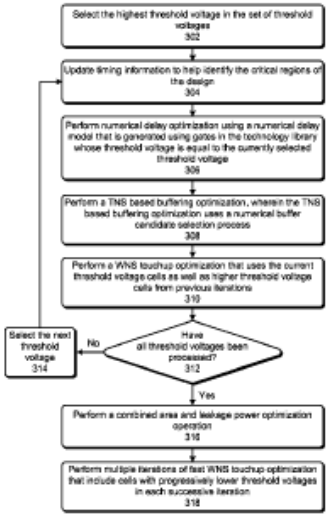
* cited by examiner

Primary Examiner — Jack Chiang
Assistant Examiner — Mohammed Alam
(74) *Attorney, Agent, or Firm* — Park, Vaughan, Fleming & Dowler LLP; Laxman Sahasrabuddhe

(57) **ABSTRACT**

Systems and techniques are described for performing numerical delay, area, and leakage power optimization on a circuit design. During operation, an embodiment can iteratively perform at least the following set of operations in a loop, wherein in each iteration a current threshold voltage value is progressively decreased: (a) perform numerical delay optimization on the circuit design using a numerical delay model that is generated using gates in a technology library whose threshold voltages are equal to the current threshold voltage; (b) perform a total negative slack based buffering optimization on the circuit design; and (c) perform a worst negative slack touchup optimization on the circuit design that uses gates whose threshold voltages are greater than or equal to the current threshold voltage. Next, the embodiment can perform combined area and leakage power optimization on the circuit design. The embodiment can then perform multiple iterations of worst negative slack touchup optimization.

15 Claims, 4 Drawing Sheets



(12) **United States Patent**
Iyer et al.

(10) **Patent No.: US 8,977,999 B2**
(45) **Date of Patent: Mar. 10, 2015**

(54) **NUMERICAL DELAY MODEL FOR A TECHNOLOGY LIBRARY CELL TYPE**

(71) Applicant: **Synopsys, Inc.**, Mountain View, CA (US)

(72) Inventors: **Mahesh A. Iyer**, Fremont, CA (US); **Amir H. Mottaez**, Los Altos, CA (US)

(73) Assignee: **Synopsys, Inc.**, Mountain View, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/246,857**

(22) Filed: **Apr. 7, 2014**

(65) **Prior Publication Data**

US 2014/0223400 A1 Aug. 7, 2014

Related U.S. Application Data

(62) Division of application No. 13/450,178, filed on Apr. 18, 2012, now Pat. No. 8,762,905.

(51) **Int. Cl.**
G06F 17/50 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 17/5031*** (2013.01); ***G06F 17/5036*** (2013.01)
USPC **716/108**; 716/100; 716/101; 716/104; 716/110; 716/113; 716/132; 716/134

(58) **Field of Classification Search**
USPC 716/100–101, 104, 108, 110–111, 113, 716/132, 134
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

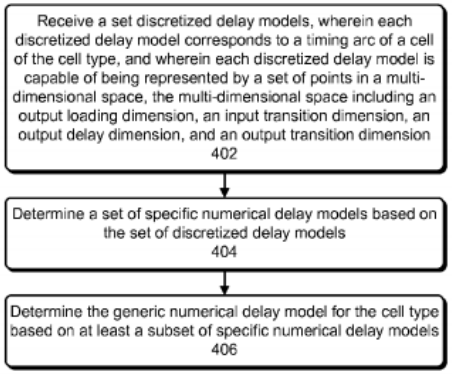
4,924,430	A	5/1990	Zasio et al.	
7,222,311	B2	5/2007	Kaufman et al.	
7,222,318	B2	5/2007	Srinivasan	
7,483,819	B2	1/2009	Gurumurthy et al.	
7,921,392	B2	4/2011	Furnish	
7,930,674	B2	4/2011	Parker et al.	
7,958,484	B2	6/2011	Kalafala et al.	
8,196,077	B2	6/2012	Yoda	
8,332,793	B2	12/2012	Bose	
8,527,257	B2	9/2013	Reddy et al.	
8,607,176	B2	12/2013	Soreff et al.	
2005/0229142	A1	10/2005	Boppana	
2010/0313176	A1	12/2010	Takahashi et al.	
2011/0204918	A1	8/2011	Nojiri	

Primary Examiner — Stacy Whitmore
(74) *Attorney, Agent, or Firm* — Park, Vaughan, Fleming & Dowler LLP; Laxman Sahasrabuddhe

(57) **ABSTRACT**

Methods and systems for determining a numerical delay model based on one or more discretized delay models are described. A discretized delay model is a delay model in which the delay behavior is represented using a set of discrete data points of delay behavior. A numerical delay model is a delay model that can be used by a numerical solver to optimize a cost function. In general, computing delay using a numerical delay model is significantly faster than computing delay using discretized delay models. This performance improvement is important when optimizing a design for various metrics like timing, area and leakage power, because repeated delay computations are required in circuit optimization approaches.

15 Claims, 3 Drawing Sheets



SYNOPSYS®

Summary and Future Work

SYNOPSYS®

Summary and Future Work

- Designing fast synthesis with good QoR and must-have optimizations for today's large designs is complex
- Tight correlation with final synthesis is a must
- Fast runtime in synthesis continues to be a major objective

- Modeling additional physical effects in smaller geometries to maintain correlation with increased design complexities
- New technologies are developed to speed up synthesis without QoR degradation, such as area, timing, and power

SYNOPSYS®
Silicon to Software™

Thank You