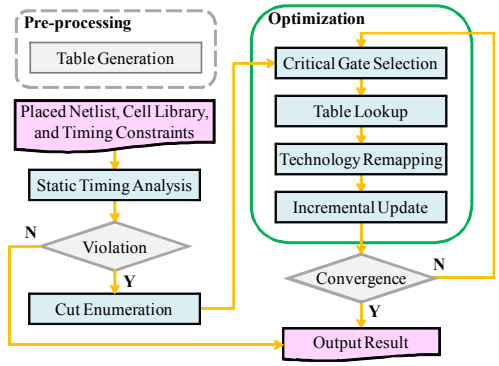
TRECO: Dynamic Technology Remapping for Timing Engineering Change Orders

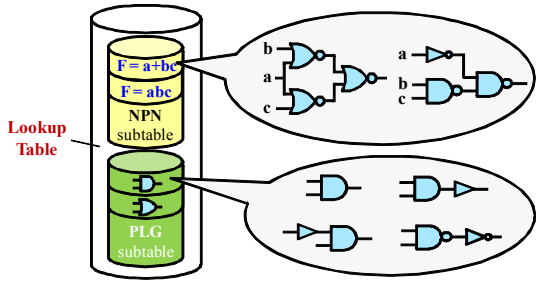
Kuan-Hsien Ho, Jie-Hong R. Jiang, and
Yao-Wen Chang

Graduate Institute of Electronics Engineering
National Taiwan University
Taipei, Taiwan

Outline

- Engineering Change Orders
- Problem Formulation
- Design Flow & Algorithms
  - Table Generation
  - Cut Enumeration
  - Table Lookup
  - Technology Remapping
- Timing-Aware Functional ECO
- Experimental Results
- Conclusions





Benchmarks	Gate count	# Spare Cells	Clock Period (ns)	# ECO Paths	# Critical Gates	WNS (ns)	TNS (ns)
Design_1	19342	569	41	16	9938	4.43	31.99
Design_2	31154	1049	68	10	1073	7.24	35.83
Design_3	10657	624	22	96	3675	1.87	98.82
Design_4	77297	2919	30	32	11979	2.75	58.83
Design_5	30825	839	14	38	1984	0.55	8.37

Engineering Change Orders (ECOs)

- ECOs are for incremental changes of a design
- In view of optimization stages:
  - Pre-mask ECOs
    - Done before chip-mask being manufactured
  - Post-mask ECOs
    - Done after chip-mask being manufactured
- In view of optimization goals
  - Functional ECOs
    - Applied for functional rectification
  - Timing ECOs
    - Applied for timing repair

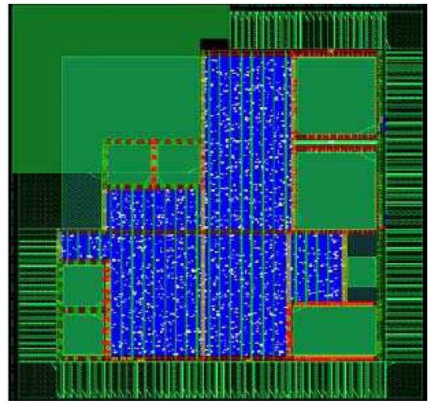
Engineering Change Orders (ECOs)

Engineering Change Orders (ECOs)

- ECOs are for incremental changes of a design
- Generally, ECOs can be divided into two categories,
- Pre-mask ECOs
  - Done before chip-mask being manufactured
  - Usually for design-time saving
- Post-mask ECOs
  - Done after chip-mask being manufactured
  - Usually for design-cost saving
- Spare cells are redundant cells reserved in a chip design and mainly used for post-mask ECOs

More on Post-Mask ECOs

- Since masks for transistors are much more expensive than those for metal layers,
  - Metal-only ECO, which performs design changes on metal layers, is the mainstream
- To enable metal-only ECO, spare cells are spread over the circuit layout after placement



Source: Faraday Technology Cooperation

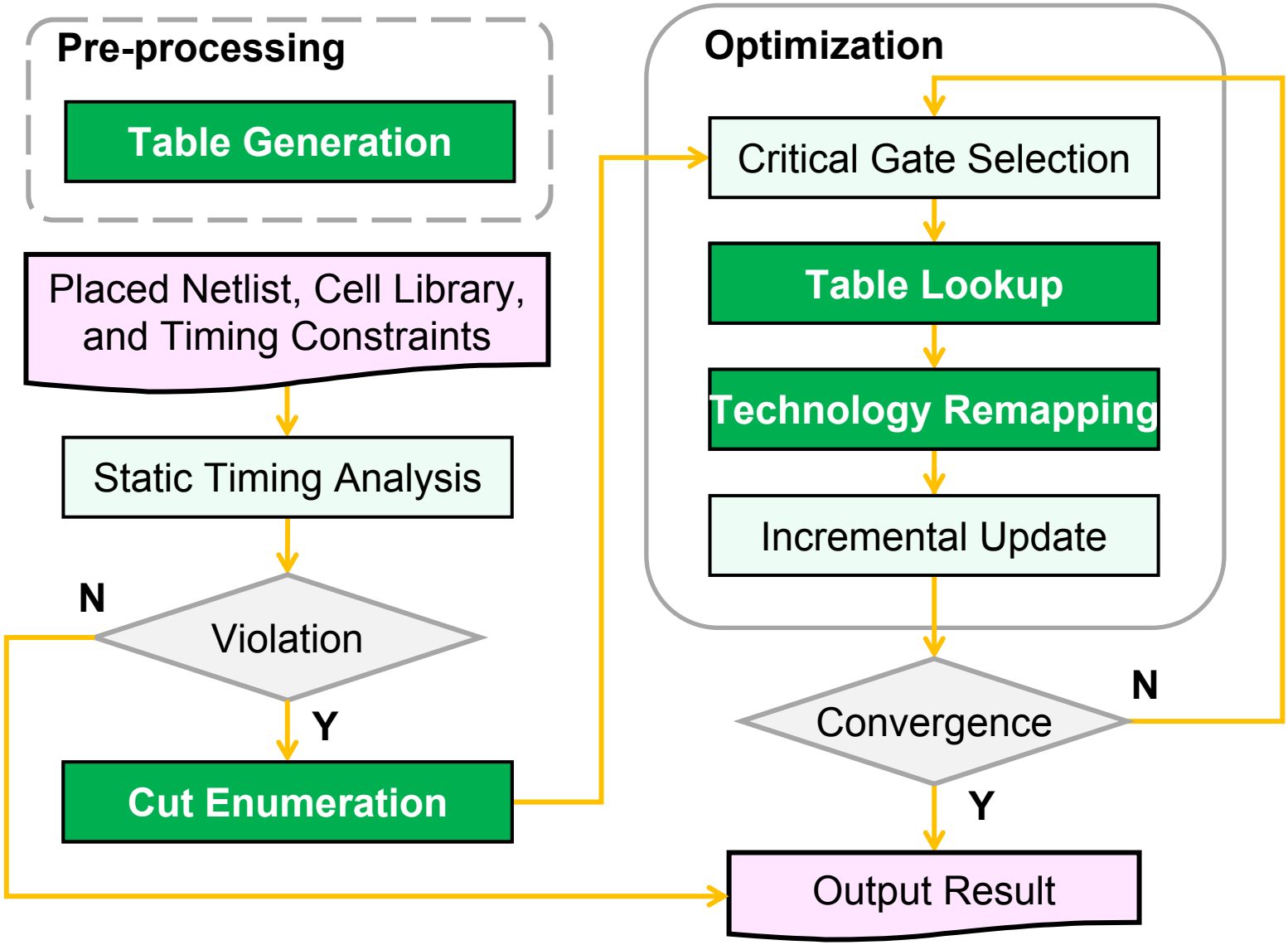
Previous Work

- Functional ECOs
  - Technology remapping considering wirelength minimziation
- Timing ECOs
  - Gate sizing, buffer insertion, and/or technology remapping

Timing-Aware Functional ECO Problem

- Given
  - A placed and routed circuit netlist
  - A set of placed spare cells
  - A cell library
  - A timing constraint
  - A set of correction logic netlists for functional rectification
- Our goal is to rewire the design such that
  - The whole circuit functionality is corrected
  - The timing constraint is satisfied
- When functional rectification is not needed, the problem reduces to timing ECO problem

Design Flow


Pre-processing
Table Generation
Placed Netlist, Cell Library, and Timing Constraints
Static Timing Analysis
N
Violation
Y
Cut Enumeration
Optimization
Critical Gate Selection
Table Lookup
Technology Remapping
Incremental Update
Convergence
N
Y
Output Result

Design Flow

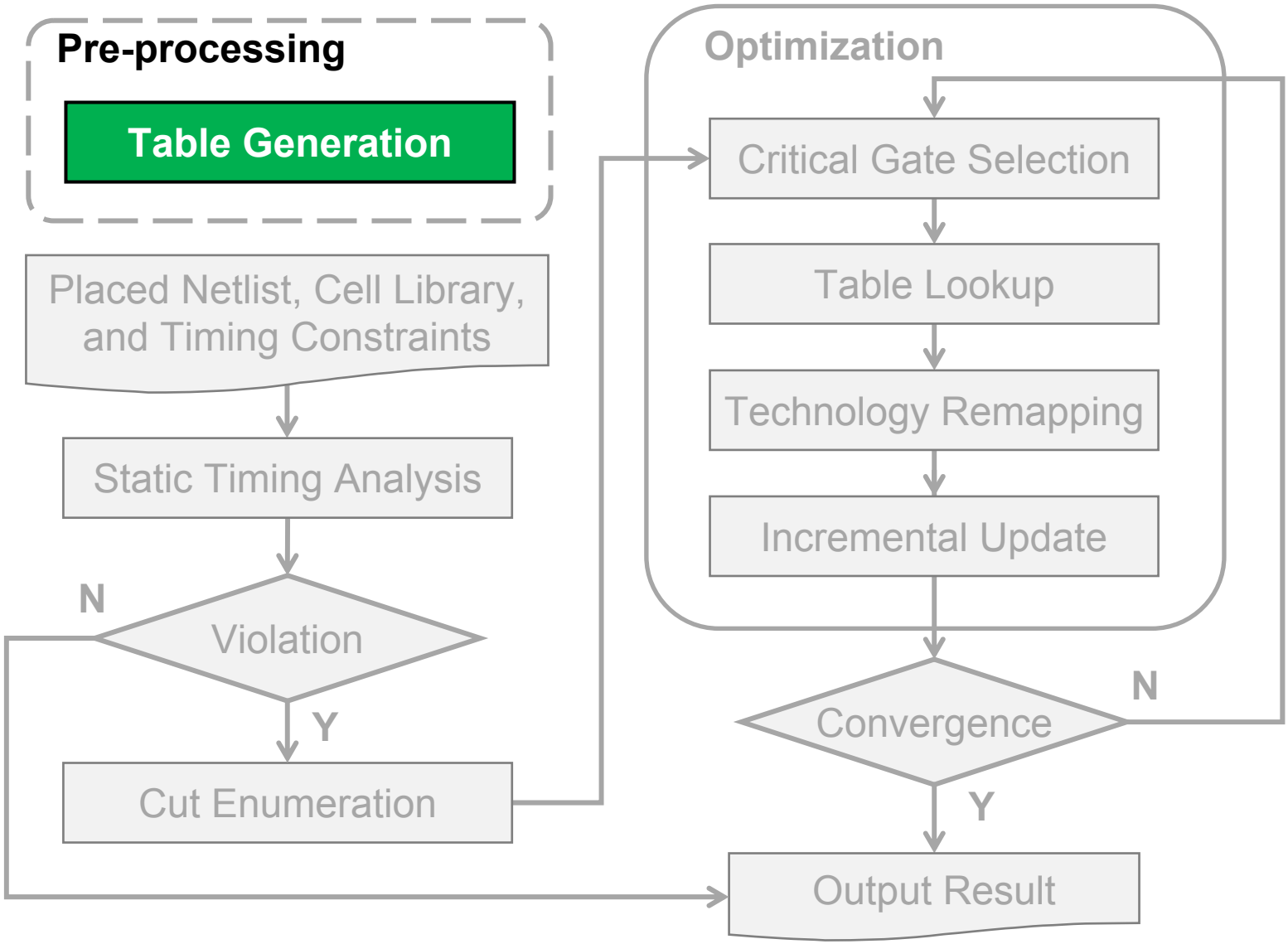

Pre-processing
Table Generation
Placed Netlist, Cell Library, and Timing Constraints
Static Timing Analysis
N
Violation
Y
Cut Enumeration
Optimization
Critical Gate Selection
Table Lookup
Technology Remapping
Incremental Update
N
Convergence
Y
Output Result

Table Generation: NPN-Subtable

- Two functions are in the same **NPN equivalence class** if they can be equivalent by
  - **N**egation of any variable(s)
  - **P**ermutation of variables and/or
  - **N**egation of the function

ex. $F=a+bc$
$G=b+ac$

F and G are **NPN-equivalent**

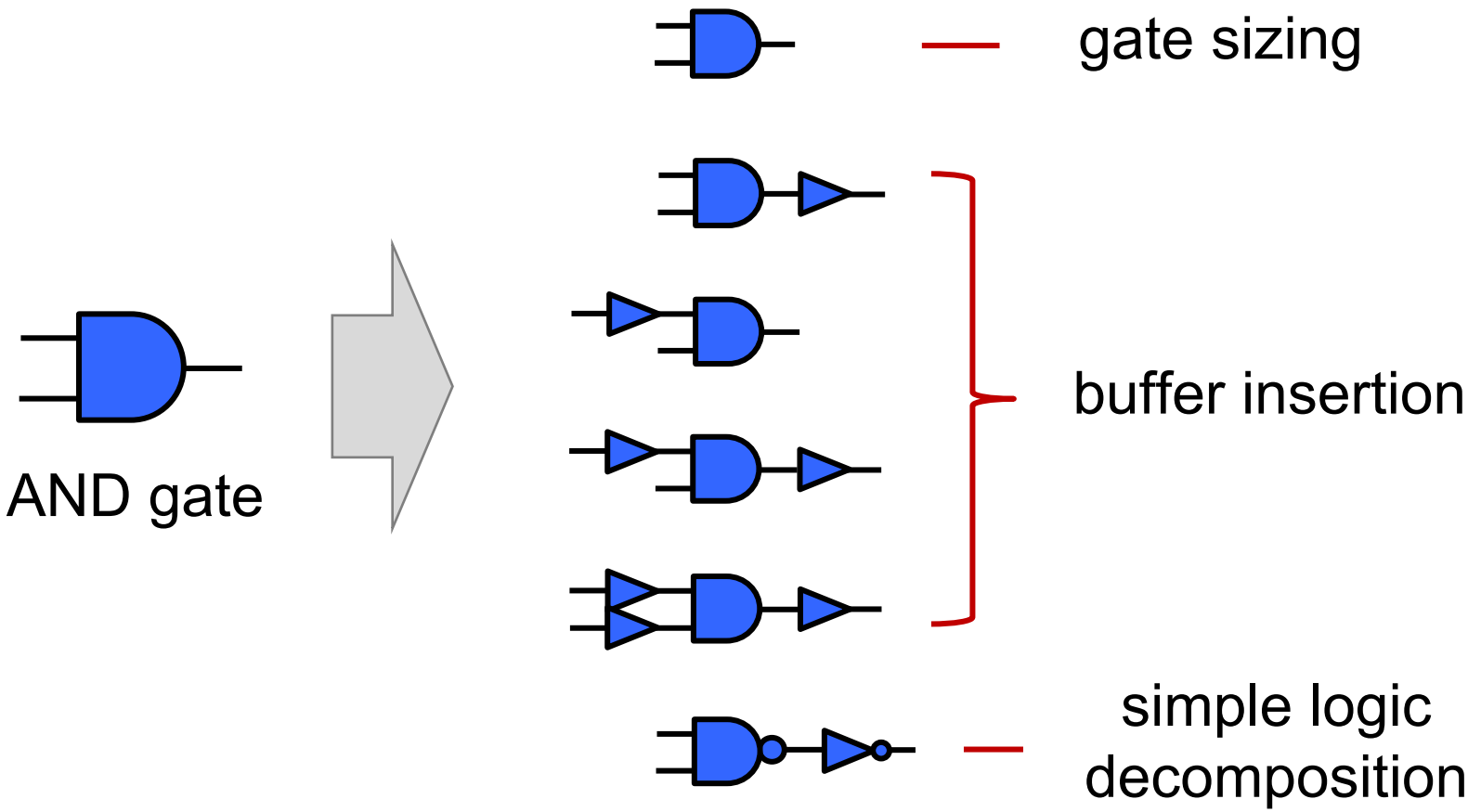
# inputs	# Functions	# NPN classes
1	4	2
2	16	4
3	256	14
4	**65,536**	**222**
5	4,294,967,296	616,126

222 pattern graphs are enough to form all 4-input Boolean functions

Our pattern graphs are composed of NAND, NOR, and inverters

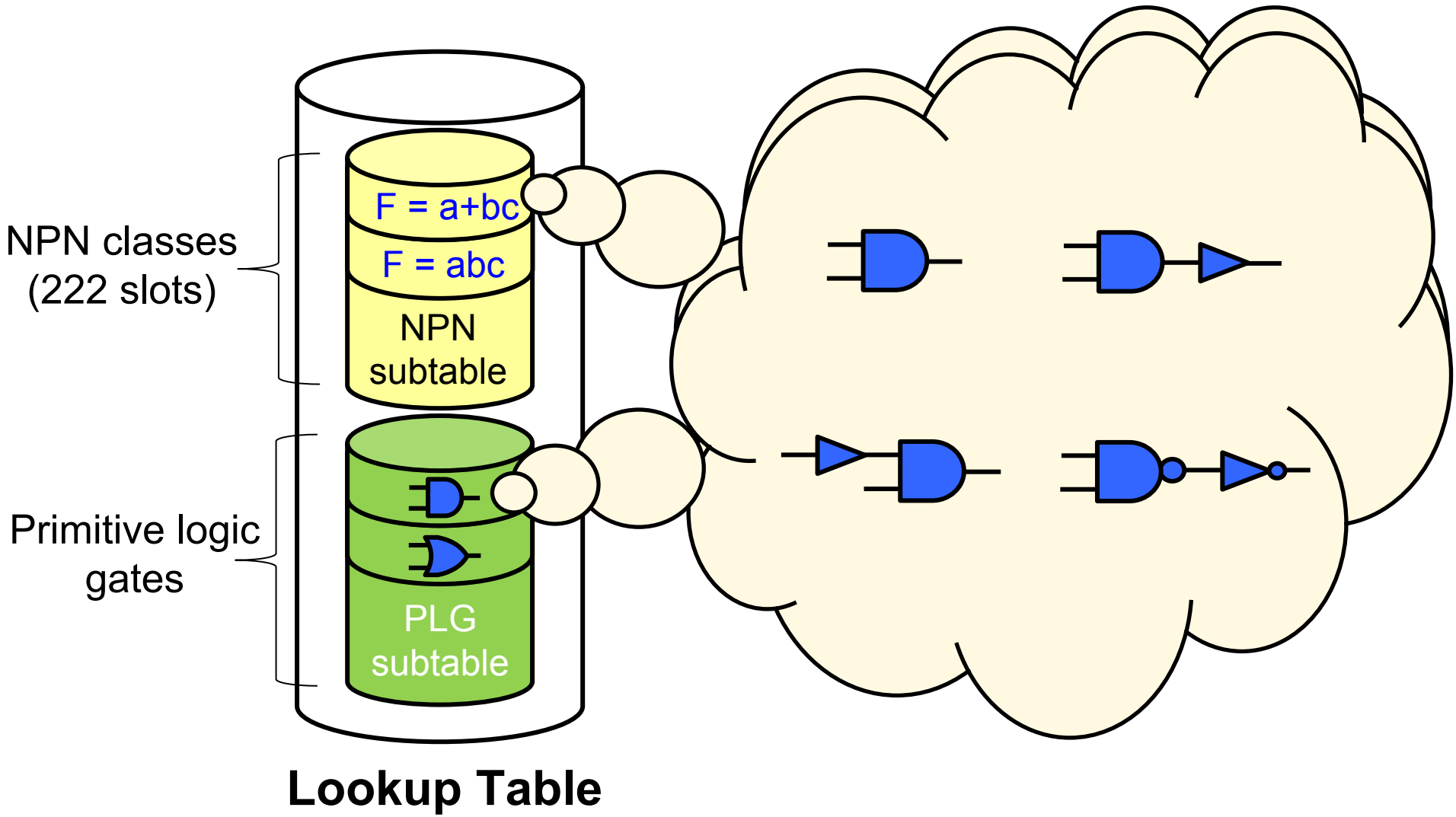
Table Generation: PLG-Subtable

- To enhance the diversity of pattern graphs, we further construct a table called PLG-subtable

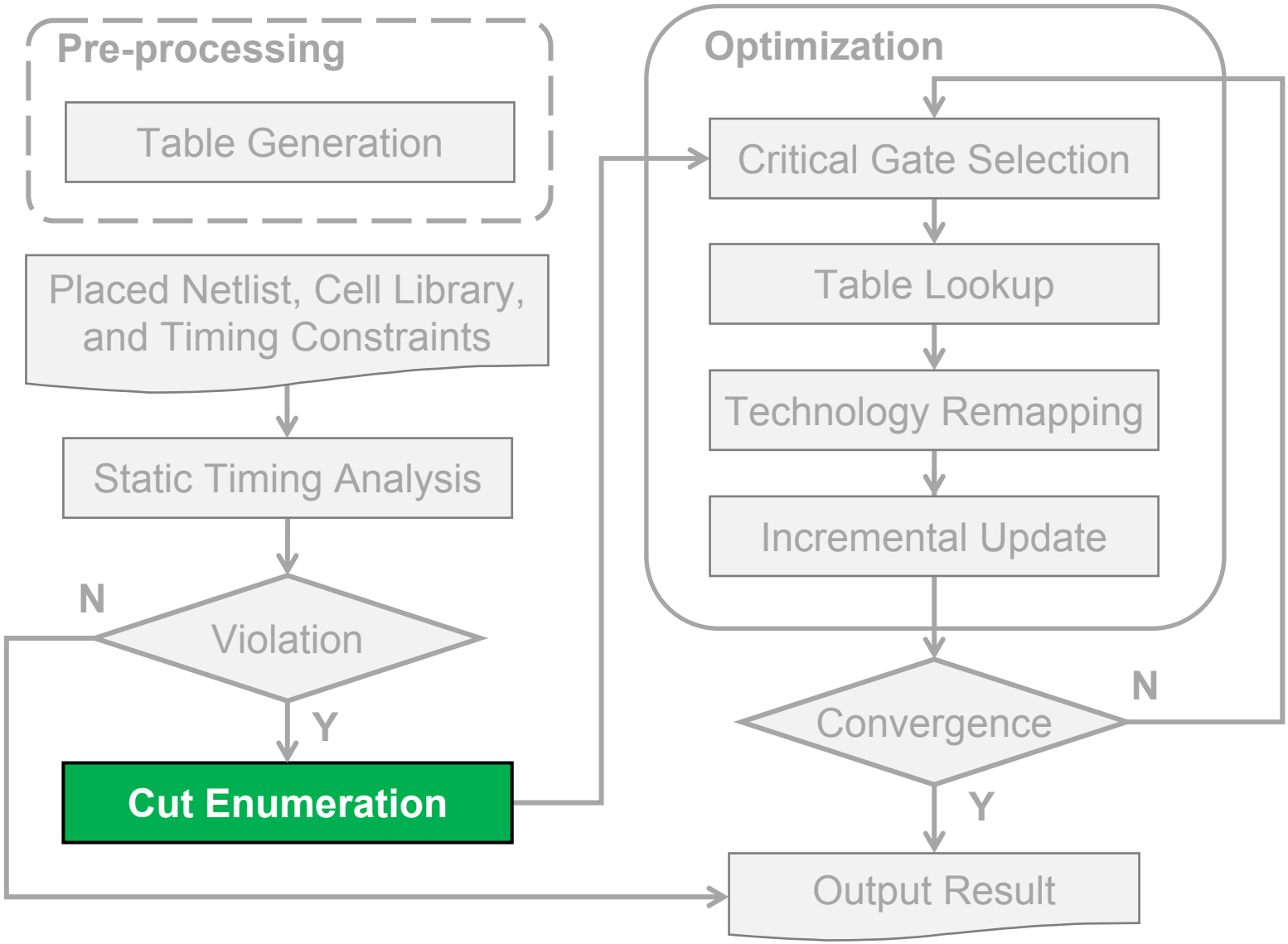


All gate types in the given cell library are considered

Lookup Table Overview

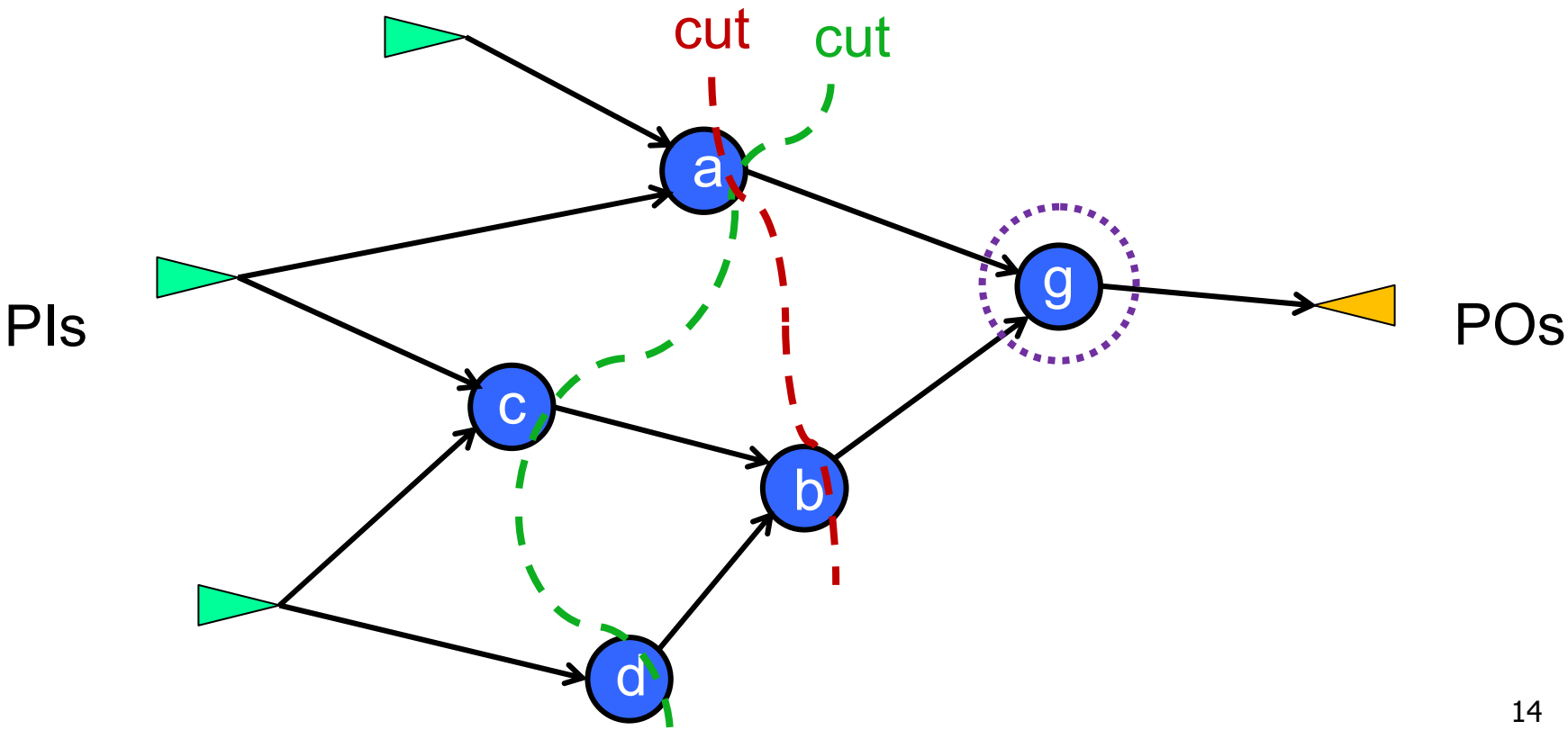


Design Flow


Pre-processing
Table Generation
Placed Netlist, Cell Library, and Timing Constraints
Static Timing Analysis
N
Violation
Y
Cut Enumeration
Optimization
Critical Gate Selection
Table Lookup
Technology Remapping
Incremental Update
Convergence
N
Y
Output Result

Cut Enumeration

- A set of gates ***C*** is a cut for a gate ***g*** if
  - Each path from primary inputs (PIs) to ***g*** must go through at least one node of ***C***
- A cut is called *k*-feasible if
  - # of gates in ***C*** does not exceed the bound, *k*



Design Flow

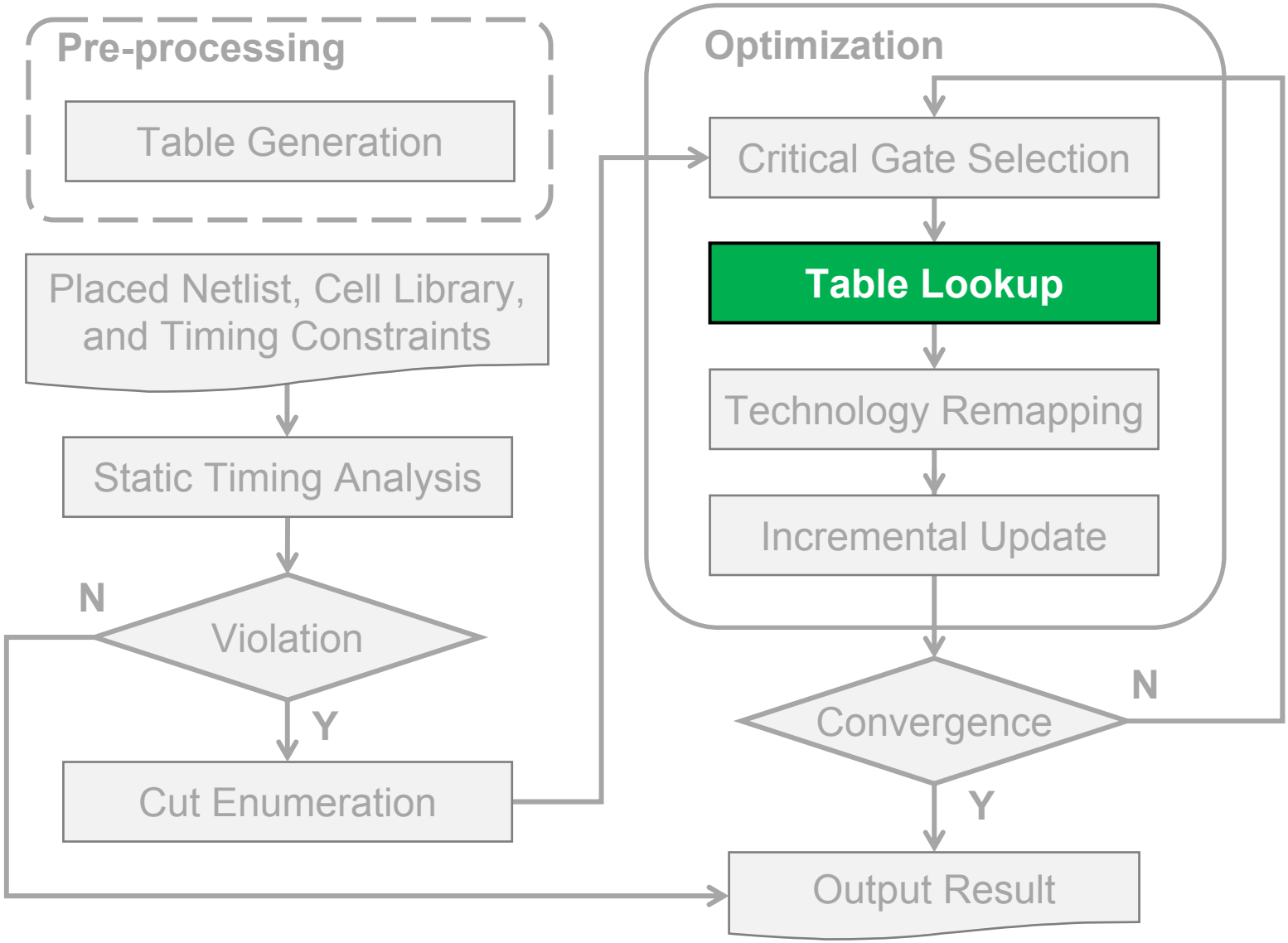
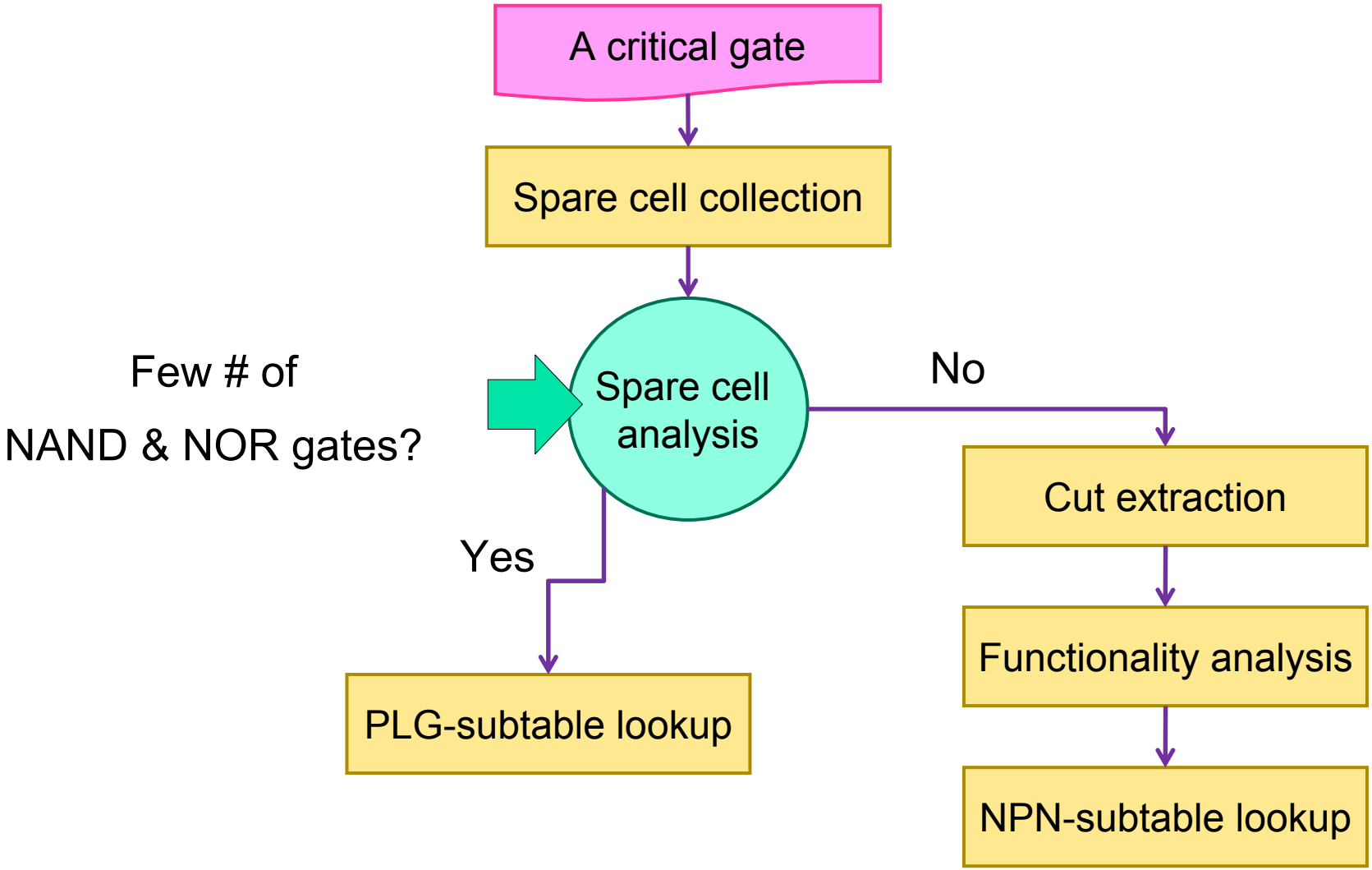
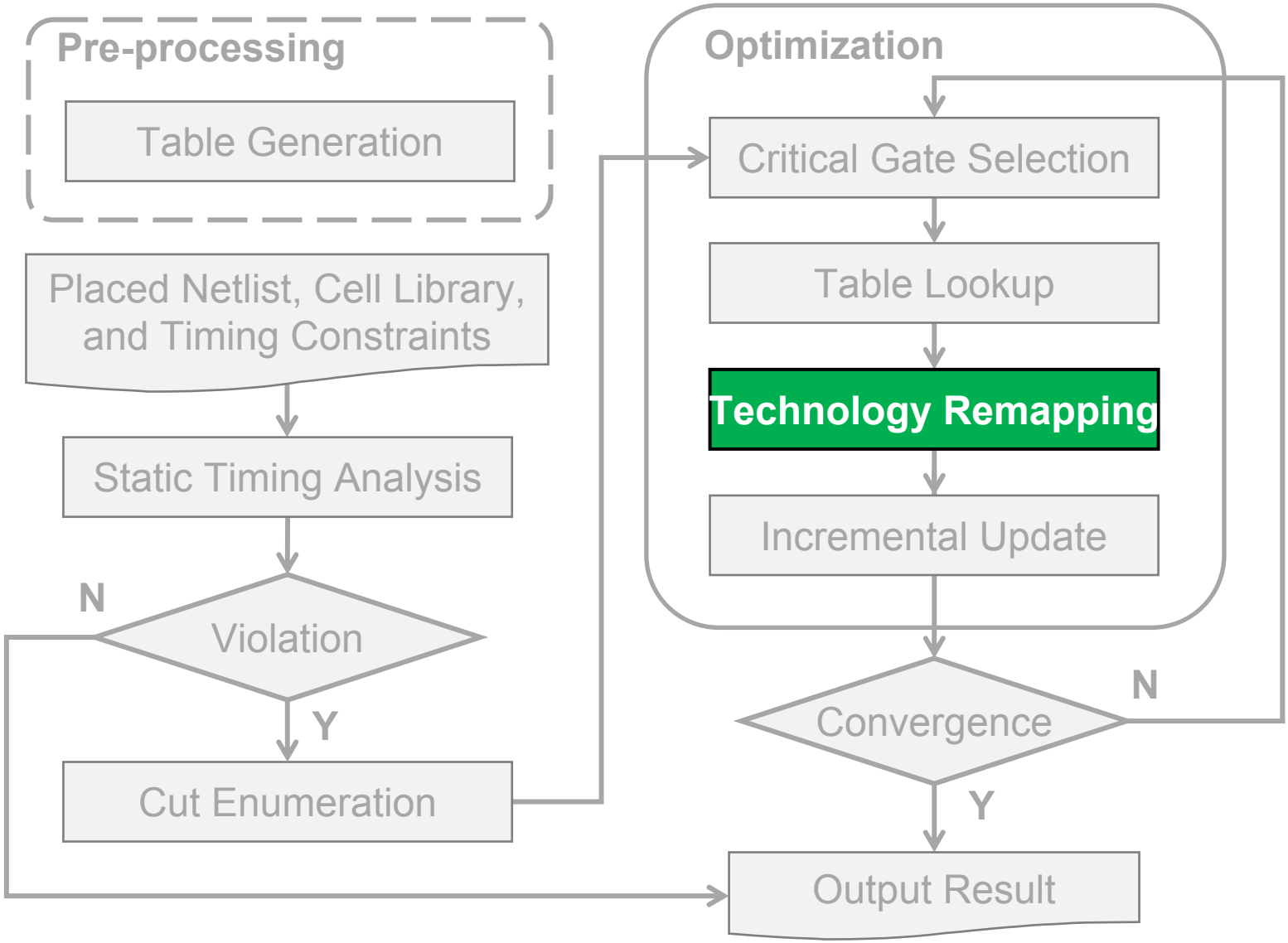

Pre-processing
Table Generation
Placed Netlist, Cell Library, and Timing Constraints
Static Timing Analysis
N
Violation
Y
Cut Enumeration
Optimization
Critical Gate Selection
Table Lookup
Technology Remapping
Incremental Update
Convergence
N
Y
Output Result

Table Lookup

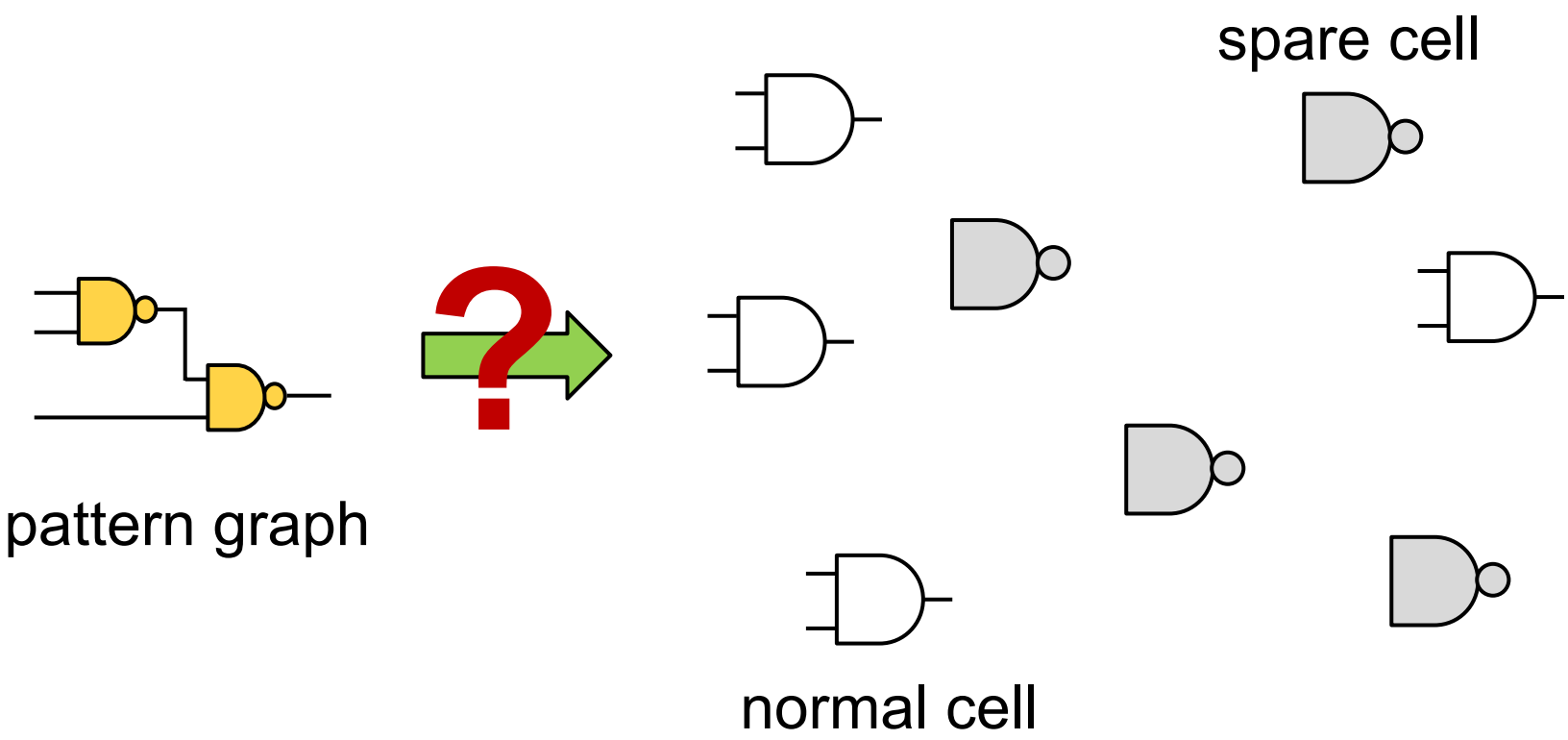


Design Flow


Pre-processing
Table Generation
Placed Netlist, Cell Library, and Timing Constraints
Static Timing Analysis
N
Violation
Y
Cut Enumeration
Optimization
Critical Gate Selection
Table Lookup
Technology Remapping
Incremental Update
Convergence
N
Y
Output Result

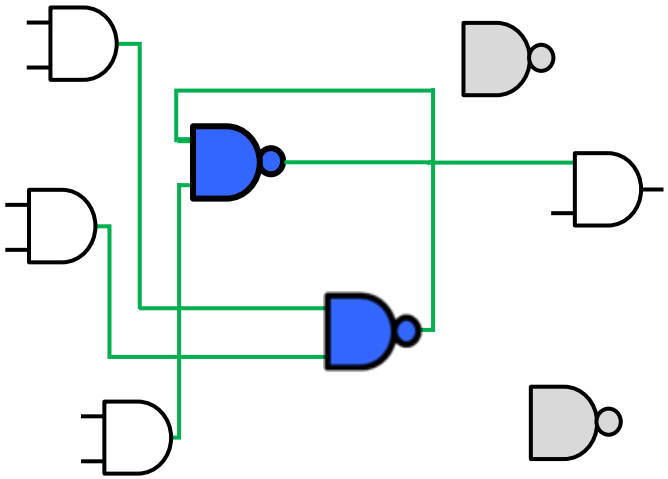
Challenges of Remapping (1/2)

- Spare-cell selection problem
- In the following example
  - C_2^4 choices
  - Complexity: $O(n^2)$ if there are n spare cells

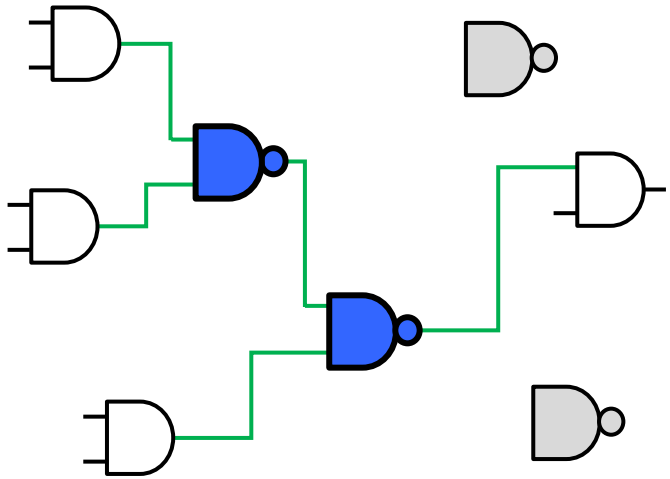


Challenges of Remapping (2/2)

- Even if the same spare cells are chosen…



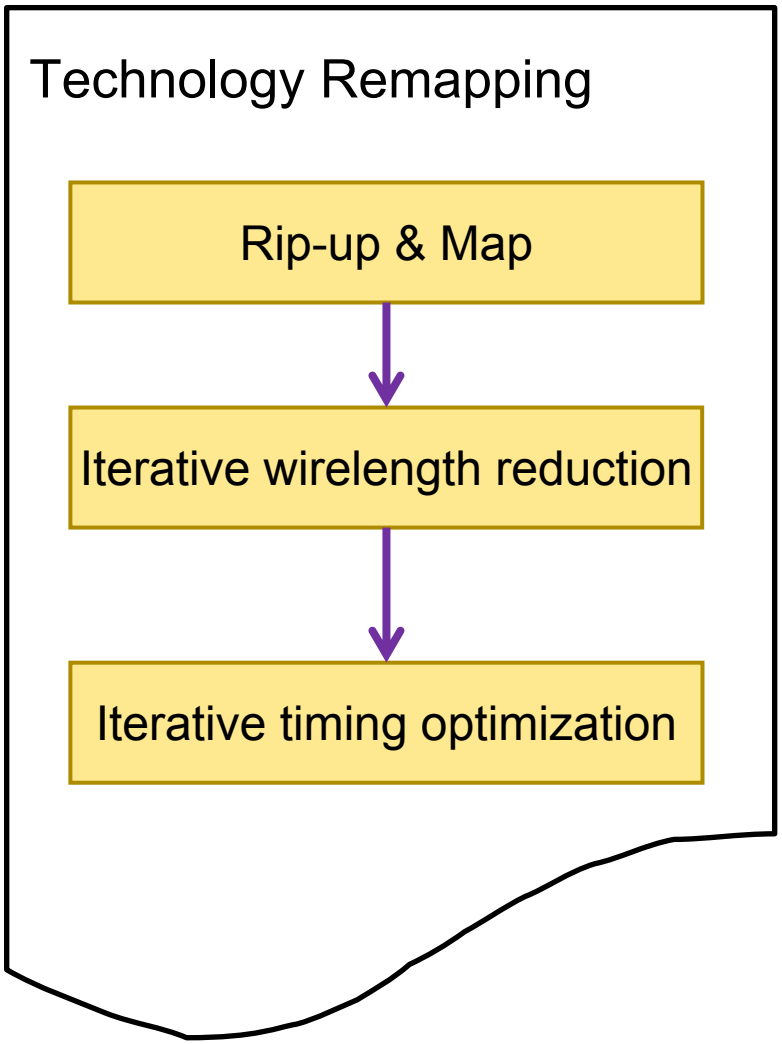
Larger wiring cost



Smaller wiring cost

The wiring cost might be quite different!

Algorithm of Technology Remapping



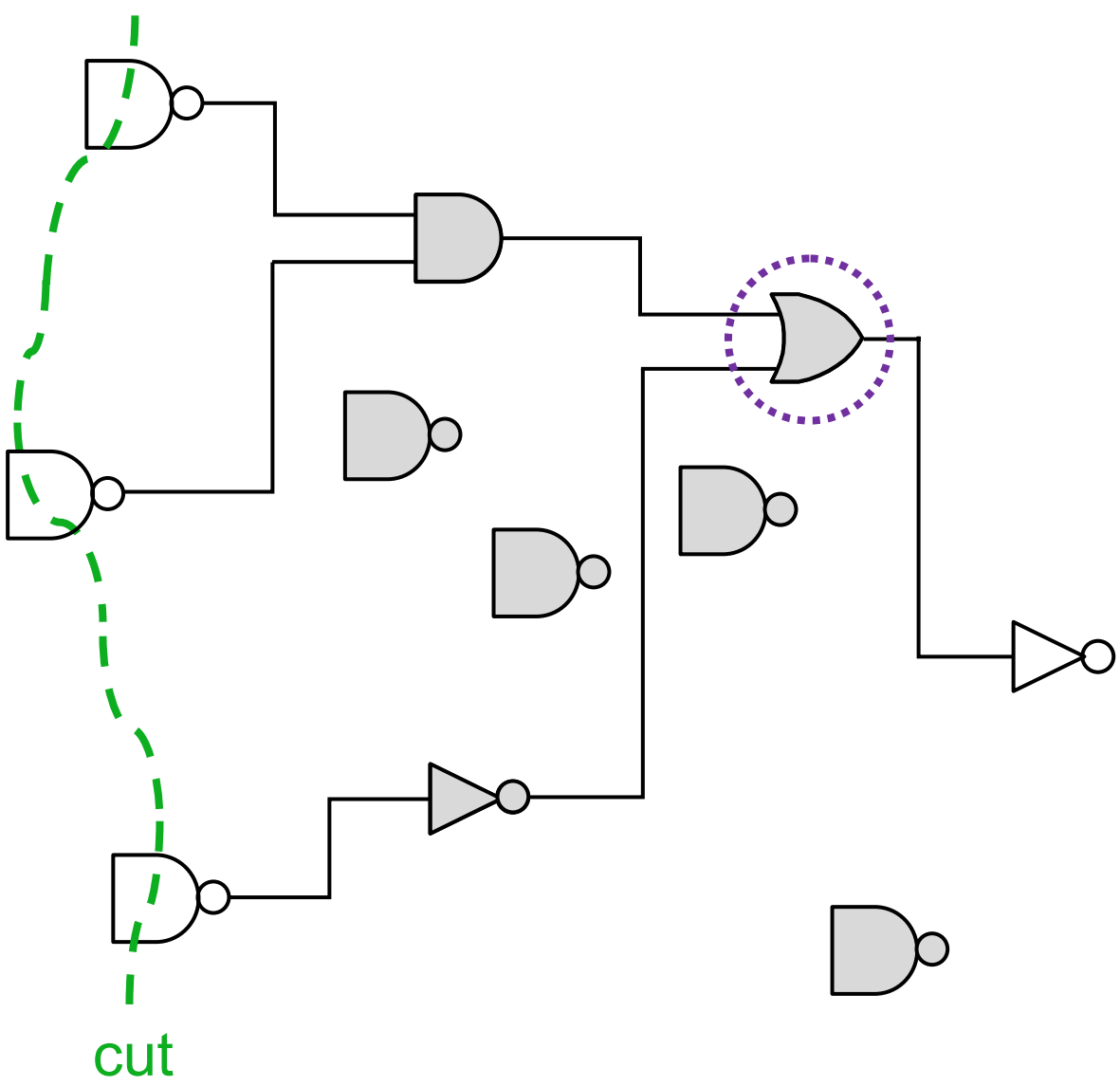
Rip-up the target sub-circuit

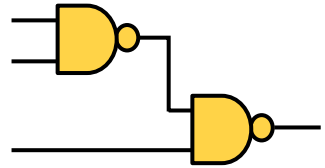
Map to a new circuit with a given pattern graph randomly

Reduce total wirelength

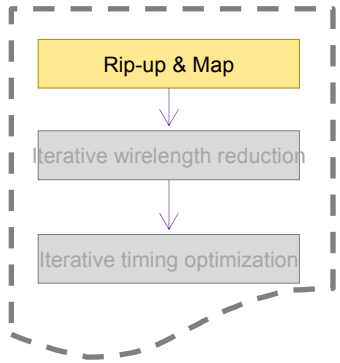
Optimize timing of the new circuit

Remapping – Rip-Up & Map

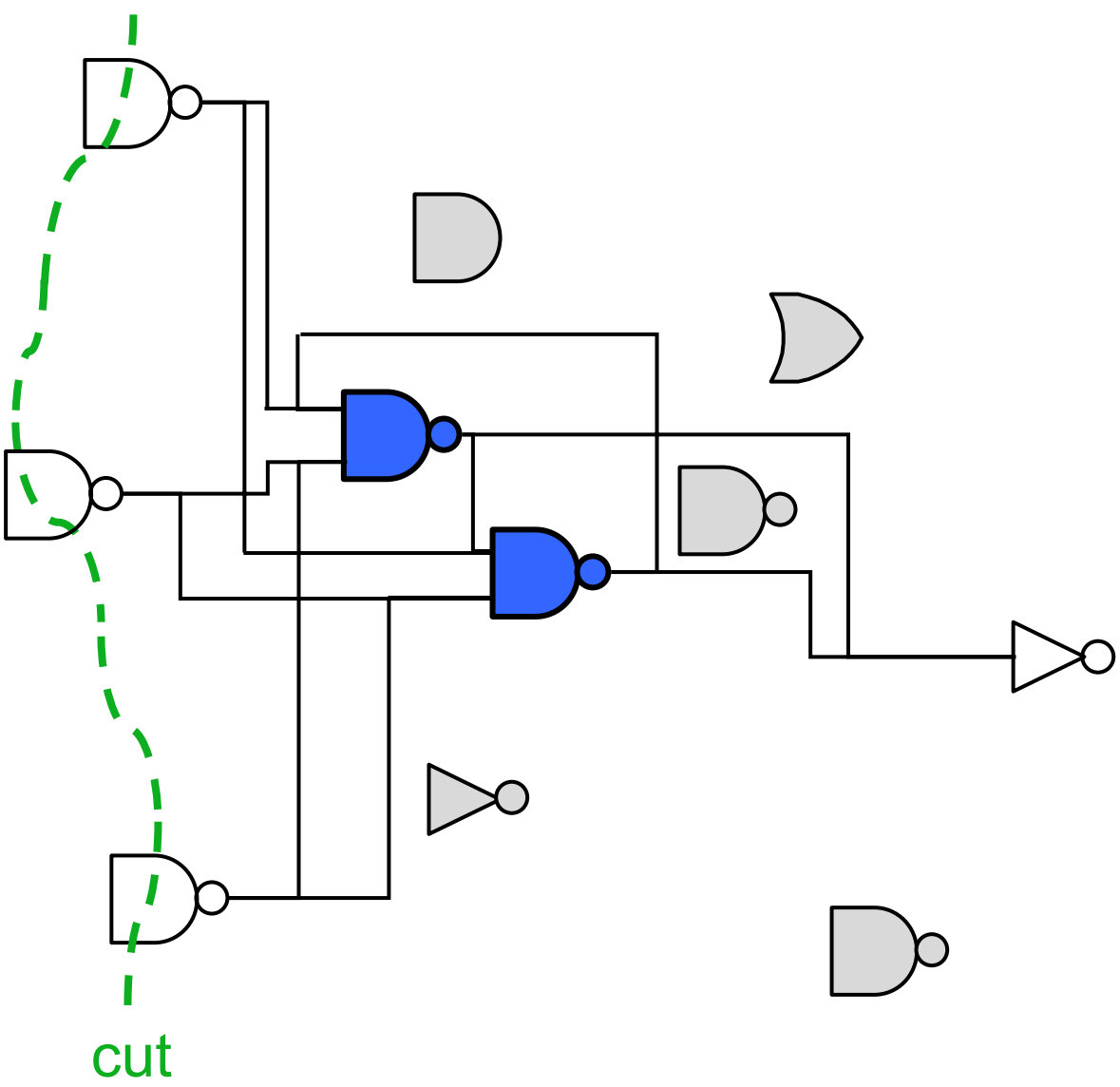


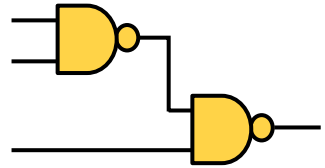


pattern graph

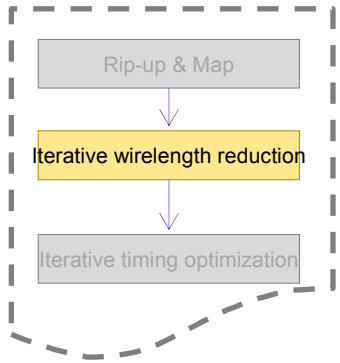


Remapping – Wirelength Reduction

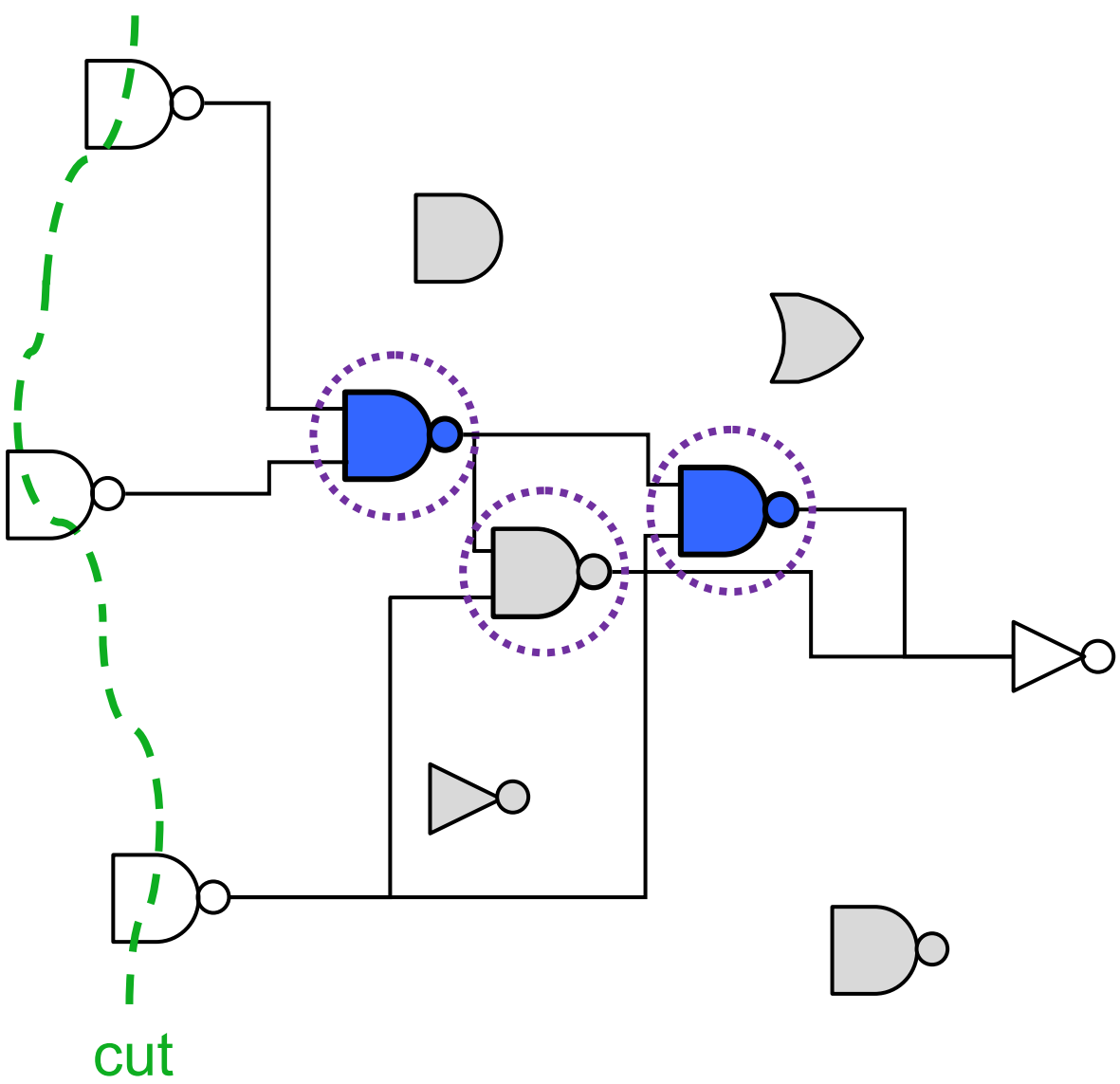


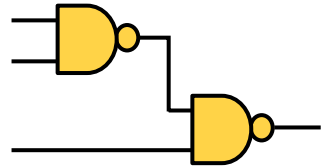


pattern graph

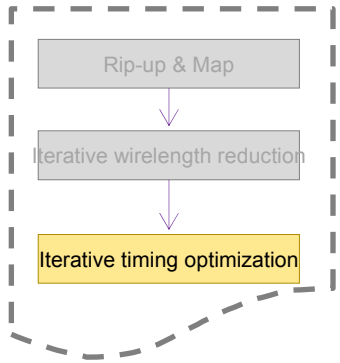


Remapping – Timing Optimization



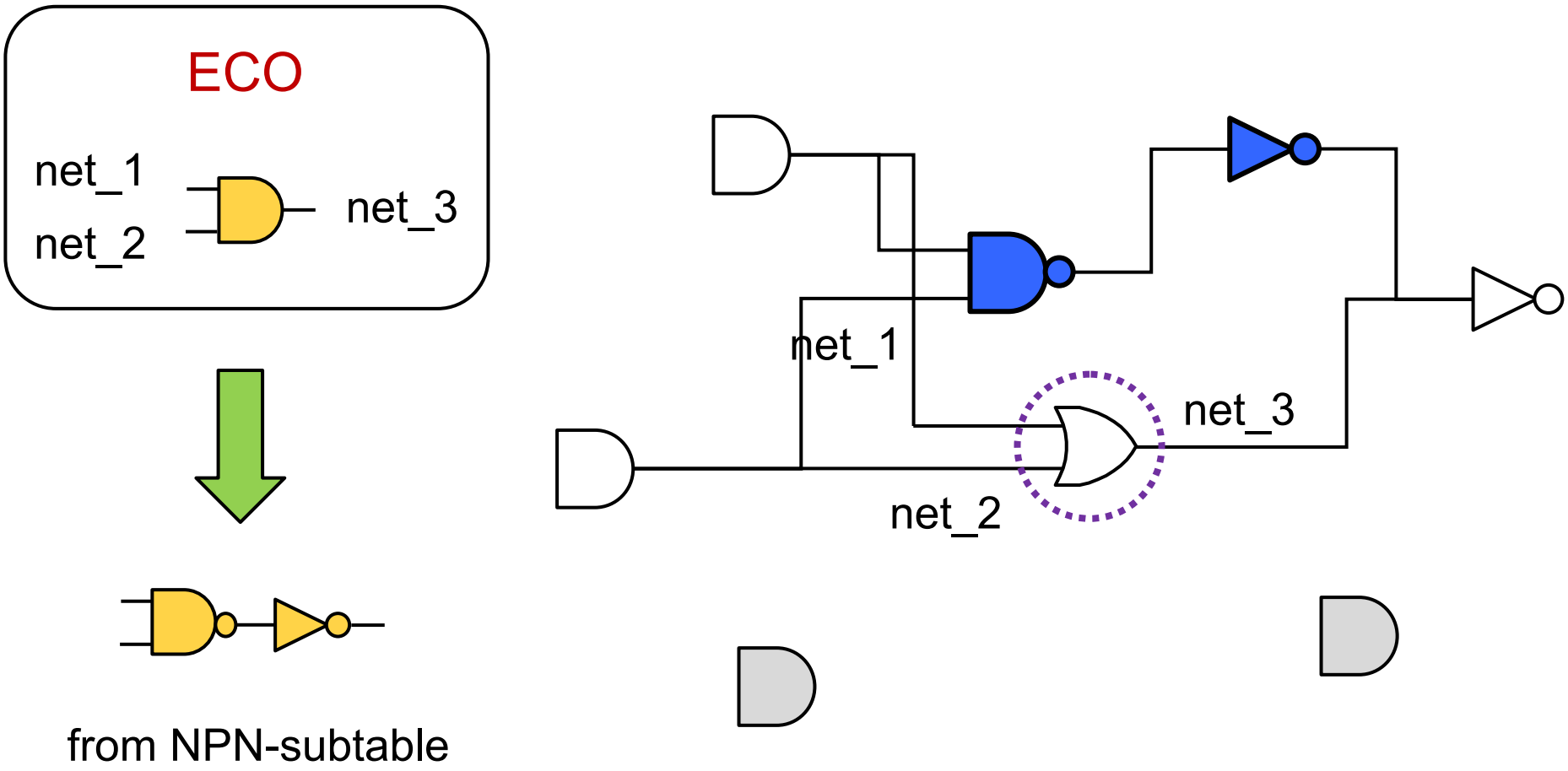


pattern graph



Extension: Timing-Aware Functional ECO

- By NPN-subtable and the remapping techniques,



Experimental Setup

- C++ programming language
- Linux workstation
  - 2.4 GHz CPU and 8 GB memory
- Five industrial designs using 180nm technology

Benchmarks	Gate count	# Spare Cells	Clock Period (ns)	# ECO Paths	# Critical Gates	WNS (ns)	TNS (ns)
Design_1	19342	569	41	16	9938	4.43	31.99
Design_2	31154	1049	68	10	1073	7.24	35.83
Design_3	10657	624	22	96	3675	1.87	98.82
Design_4	77297	2919	30	32	11979	2.75	58.83
Design_5	30825	839	14	38	1984	0.55	8.37

- The spare-cell types considered were AND, OR, NAND, NOR, buffer, and inverter

Experimental Results

- The execution time of TRECO depends on several factors, such as
  - # of critical gates,
  - Cut size and cut volume of each critical gate,
  - # of subgraphs being considered,
  - Success rate and effectiveness of remapping on a gate,
  - Spare-cell density, etc.

	Initial			Technology Remapping Part of [4]						TRECO					
Benchmarks	# ECO Paths	WNS (ns)	TNS (ns)	# ECO Paths	WNS (ns)	Imp. Ratio	TNS (ns)	Imp. Ratio	Time (s)	# ECO Paths	WNS (ns)	Imp. Ratio	TNS (ns)	Imp. Ratio	Time (s)
Design_1	16	4.43	31.99	11	2.99	32.55%	20.03	37.40%	543	9	2.69	39.35%	12.82	59.94%	192
Design_2	10	7.24	35.83	4	3.21	55.60%	6.54	81.73%	221	3	2.42	66.58%	3.92	89.07%	121
Design_3	96	1.87	98.82	80	1.34	28.48%	61.06	38.21%	78	78	1.16	38.07%	55.60	43.74%	16
Design_4	32	2.75	58.83	32	2.57	6.53%	54.63	7.13%	1071	32	2.20	20.07%	49.96	15.08%	759
Design_5	38	0.55	8.37	12	0.10	81.57%	0.51	93.90%	124	9	0.07	86.96%	0.33	96.03%	19
Avg.						**0.73x**		**0.77x**	**3.51x**			**1**		**1**	**1**

[4] Y.-P. Chen, "ECO Timing Optimization using Spare Cells and Technology Remapping," *M.S. Thesis, National Taiwan University, July* 2006.
(Thesis version of [Chen et al., ICCAD'2007])

Conclusions

- We have presented a post-mask ECO technique, called TRECO, which supports timing as well as functional ECOs.
- Compared with prior work, it offers a more effective and efficient solution on timing optimization using technology remapping.
- Experimental results based on five industrial designs show that TRECO can alleviate timing violations under reasonable CPU times

Thank You!

Kuan-Hsien Ho

khho@eda.ee.ntu.edu.tw