Alexey Glebov

Series-Parallel BDD: Theory and Applications

Outline

- Introduction
- Several words about general BDD
- SP-BDD: motivation
- SP-BDD: basic definitions and theorems
- SP-BDD: basic algorithms
- SP-BDD: basic applications
- SP-BDD: application in detail – fast delay calculation
- SP-BDD: application in detail – resynthesis
- Conclusion

Outline

- Introduction
- Several words about general BDD
- SP-BDD: motivation
- SP-BDD: basic definitions and theorems
- SP-BDD: basic algorithms
- SP-BDD: basic applications
- SP-BDD: application in detail – fast delay calculation
- SP-BDD: application in detail – resynthesis
- Conclusion

BDD - R.Bryant, 1986
Boolean functions

SP-BDD - A.Glebov, D.Blaauw, L.Jones, 1995
Boolean functions and CMOS circuit topology

Outline

- Introduction
- Several words about general BDD
- SP-BDD: motivation
- SP-BDD: basic definitions and theorems
- SP-BDD: basic algorithms
- SP-BDD: basic applications
- SP-BDD: application in detail – fast delay calculation
- SP-BDD: application in detail – resynthesis
- Conclusion

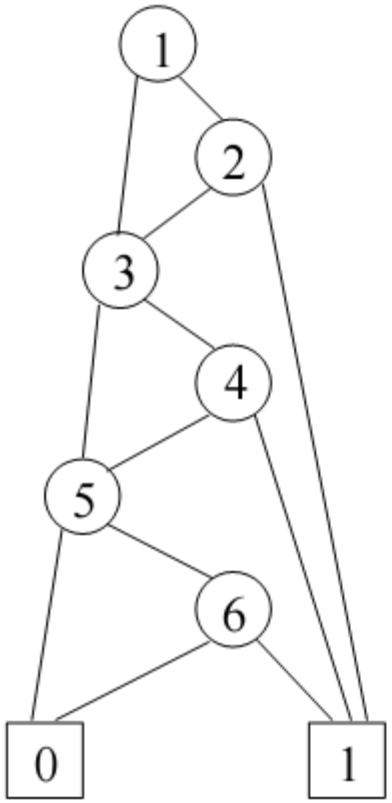
Binary Decision Diagram (BDD) –

binary directed acyclic graph with root, two sinks (terminals),

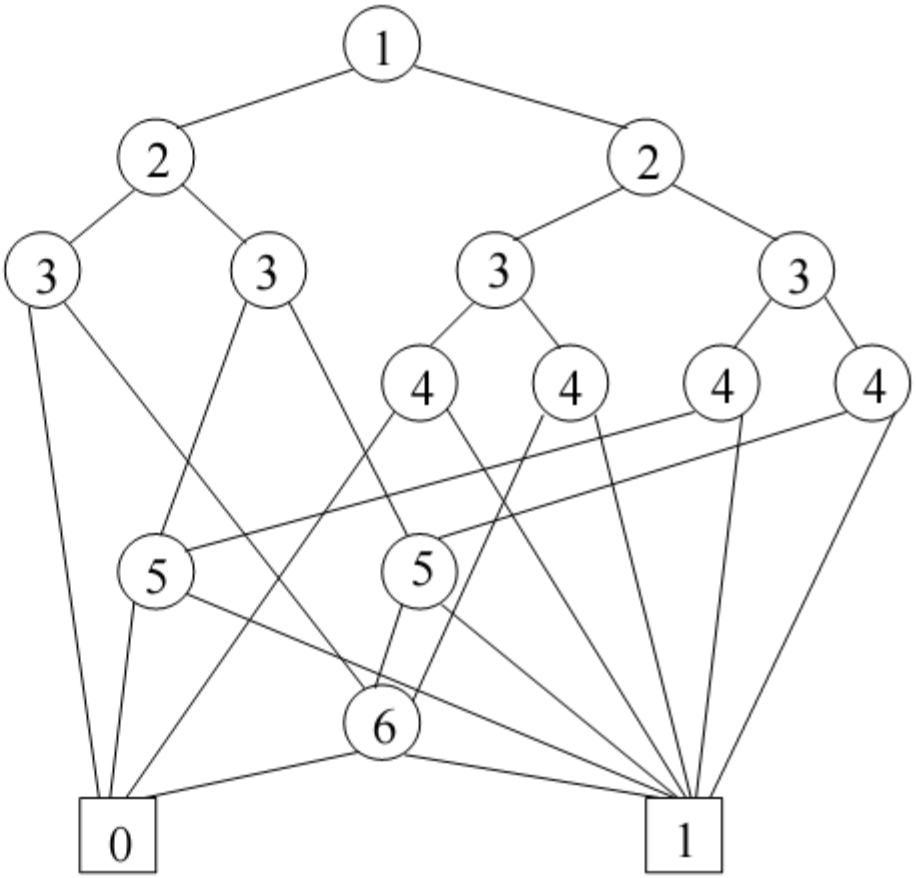
Boolean variable assigned to every non-terminal vertex,

and Boolean constants (0 and 1) assigned to terminals

$x_1 \cdot x_2 + x_3 \cdot x_4 + x_5 \cdot x_6$



$x_1 \cdot x_4 + x_2 \cdot x_5 + x_3 \cdot x_6$



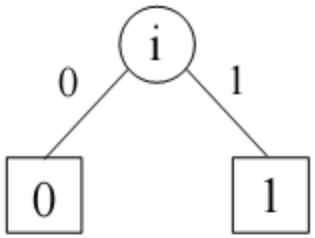
left = low, right = high

Size depends on variable order, Essentially complex functions

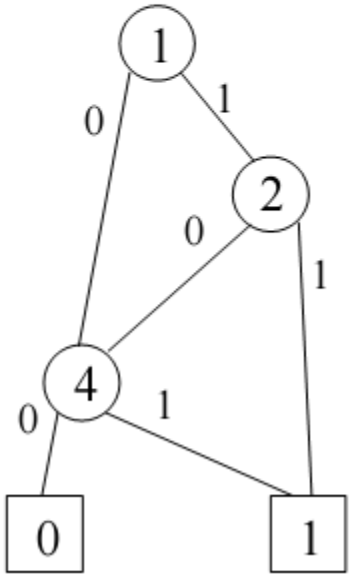
ROBDD - Reduced Ordered BDD

For specified variable order -
canonical representation of Boolean function

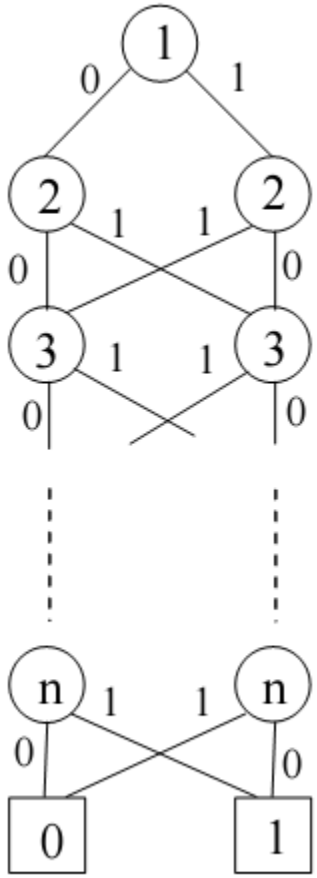
More examples of ROBDD



$x_1 \cdot x_2 + x_4$



Imparity check



Основные приложения BDD:

- **Формальная верификация цифровых схем**
- **Генерация тестов**
- **Логический синтез цифровых схем**

Outline

- Introduction
- Several words about general BDD
- SP-BDD: motivation
- SP-BDD: basic definitions and theorems
- SP-BDD: basic algorithms
- SP-BDD: basic applications
- SP-BDD: application in detail – fast delay calculation
- SP-BDD: application in detail – resynthesis
- Conclusion

Series-Parallel BDD (SP-BDD)

Motivation: search for convenient data structure

for algorithm of transistor reordering in CMOS gate

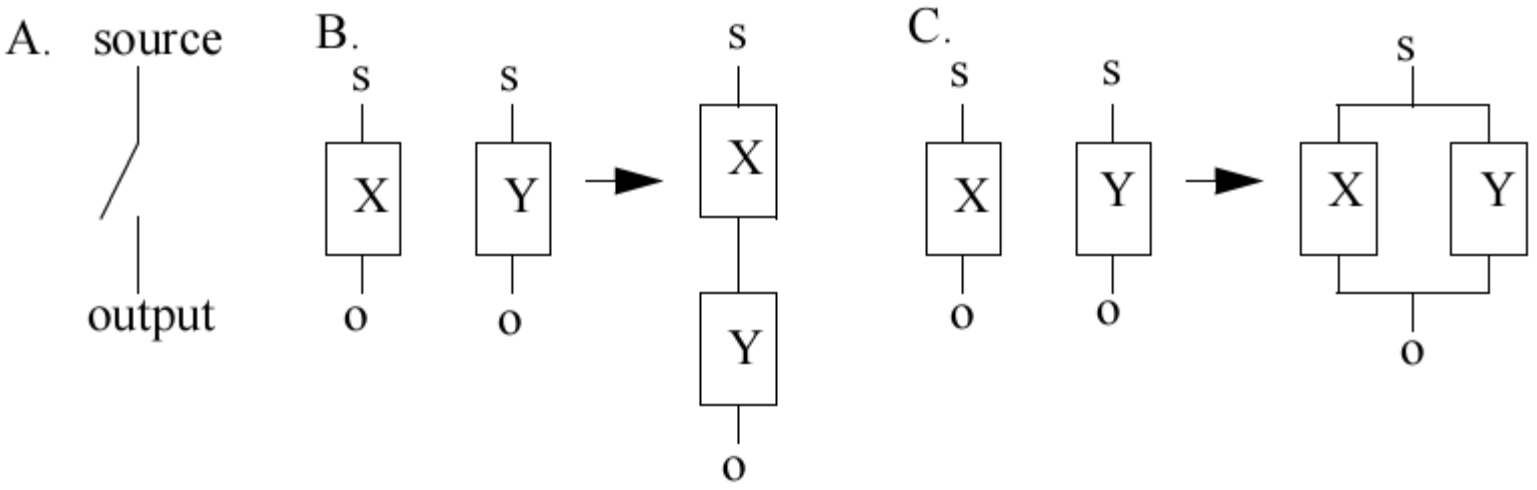
(CMOS gate = CCMOS gate)

Outline

- Introduction
- Several words about general BDD
- SP-BDD: motivation
- SP-BDD: basic definitions and theorems
- SP-BDD: basic algorithms
- SP-BDD: basic applications
- SP-BDD: application in detail – fast delay calculation
- SP-BDD: application in detail – resynthesis
- Conclusion

Definition 1. ***SP-network*** **is one of the following objects:**

- **a switch,**
- **series connection of two SP-networks,**
- **parallel connection of two SP-networks.**



According to Definition 1, SP-network is a non-directed graph with vertices for nodes and edges for switches.

Definition 2. ***SP-function*** **is a Boolean function that can be associated with certain SP-net-work.**

Definition 3. ***Partial order associated with SP-network*** **is a binary relation "<" on the set of SP-network switches, defined as follows. Let a,b be switches of SP-network. Then a<b ("a precedes b") if there is non-self-intersecting path from source terminal to output termi-nal, containing a and b, with a preceding b in the path.**

It can be easily shown that:

- the relation "<" is really a partial order;

- SP-network is completely specified if and only if the list of switches and associated partial order on them are specified.

Definition 4. ***Linear order associated with SP-network*** **is a binary relation "<<" on the set of SP-network switches, that satisfies the following conditions:**

- **If a<<b then a<b (i.e. linear order contains partial order).**
- **If SP-network contains SP-networks X and Y connected in parallel, then**

  either a<<b for every a from X and every b from Y,

  or b<<a for every a from X and every b from Y.

Definition 5. ***SP-BDD associated with SP-network*** **is a ROBDD (reduced ordered BDD, i.e. reduced function graph in terms of [1]) for Boolean function associated with this SP- network, if its order of variables is a linear order associated with this SP-network.**

Theorem 1. Let F be SP-network with associated Boolean function f. Then SP-BDD associated with F has minimal size among all ROBDDs for f. More precisely, SP-BDD assosiated with F has exactly one non-terminal vertex for each argument of f, and besides it has two terminal vertices with values 0 and 1.

Corollary 1. For the terminal vertices of SP-BDD the following is correct:

- t0 can be low-son and cannot be high-son of other vertex;
- t1 can be high-son and cannot be low-son of other vertex.

Corollary 2. A vertex of SP-BDD cannot be low-son of some vertex and at the same time high-son of other vertex.

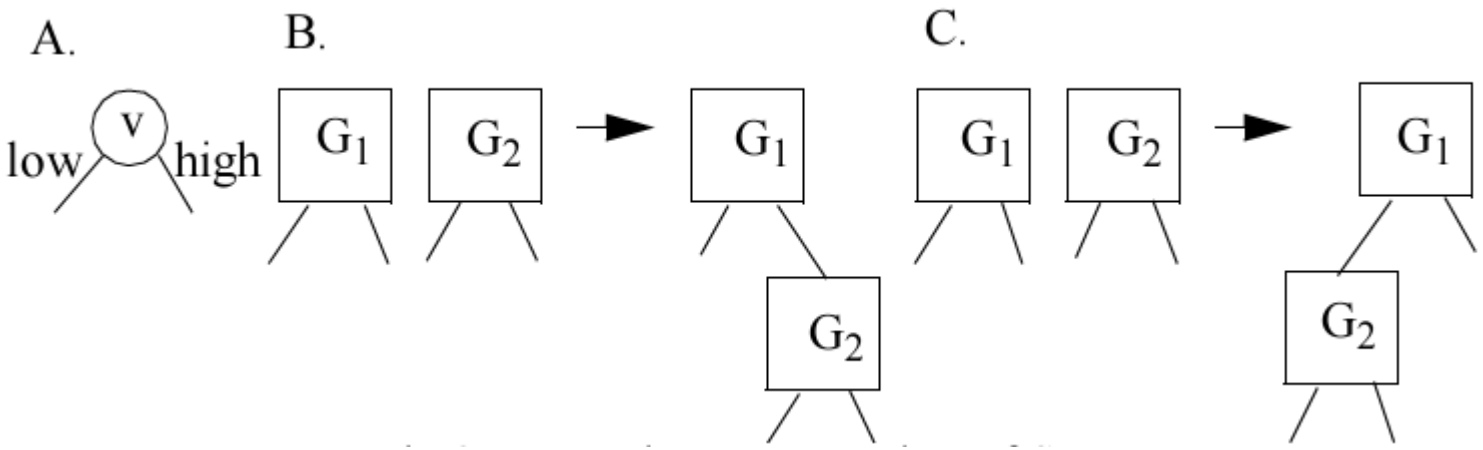
Corollary 3. For any two neighbouring (in linear order) vertices v,w of SP-BDD (i.e. such that index(v)+1=index(w)) we have either low(v)=w or high(v)=w.

Recursive construction of SP-BDD:

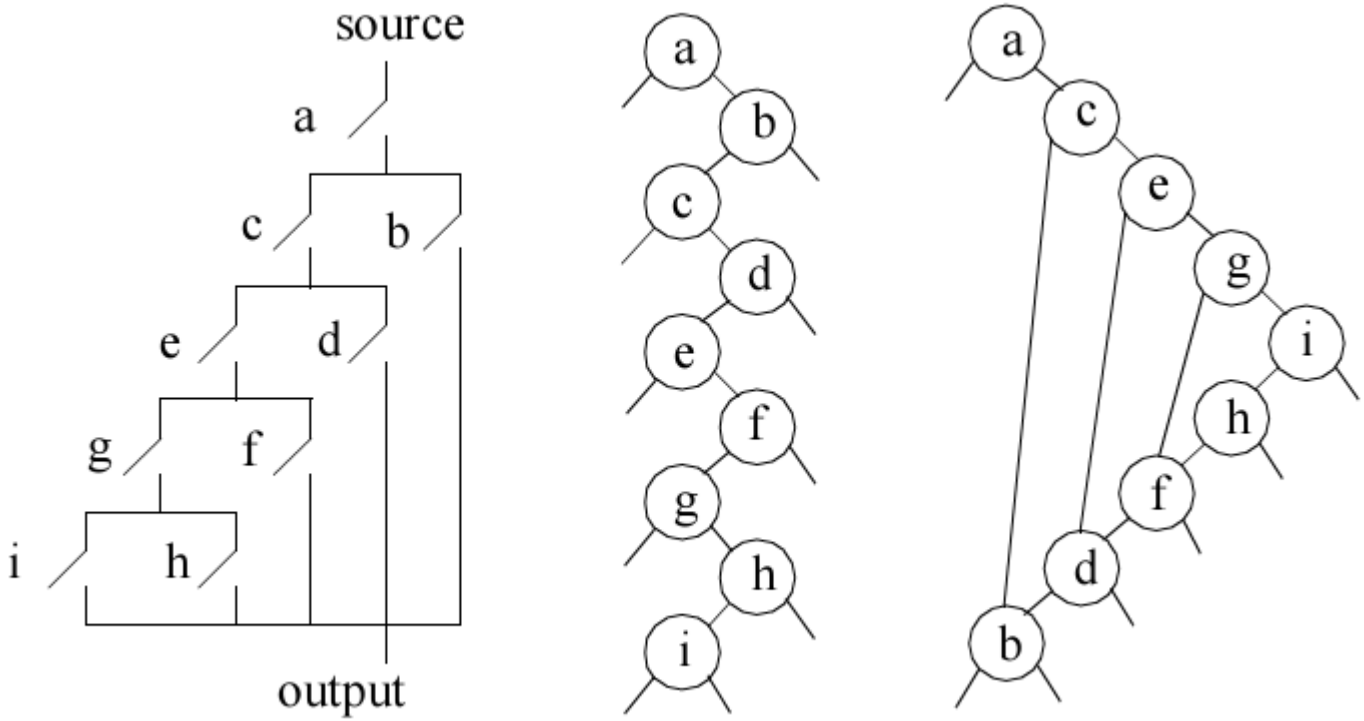
A. SP-BDD for single switch

B. Forming SP-BDD for series connection of two SP-networks

C. Forming SP-BDD for parallel connection of two SP-networks



SP-network and two variants of its SP-BDD corresponding to different associated linear orders



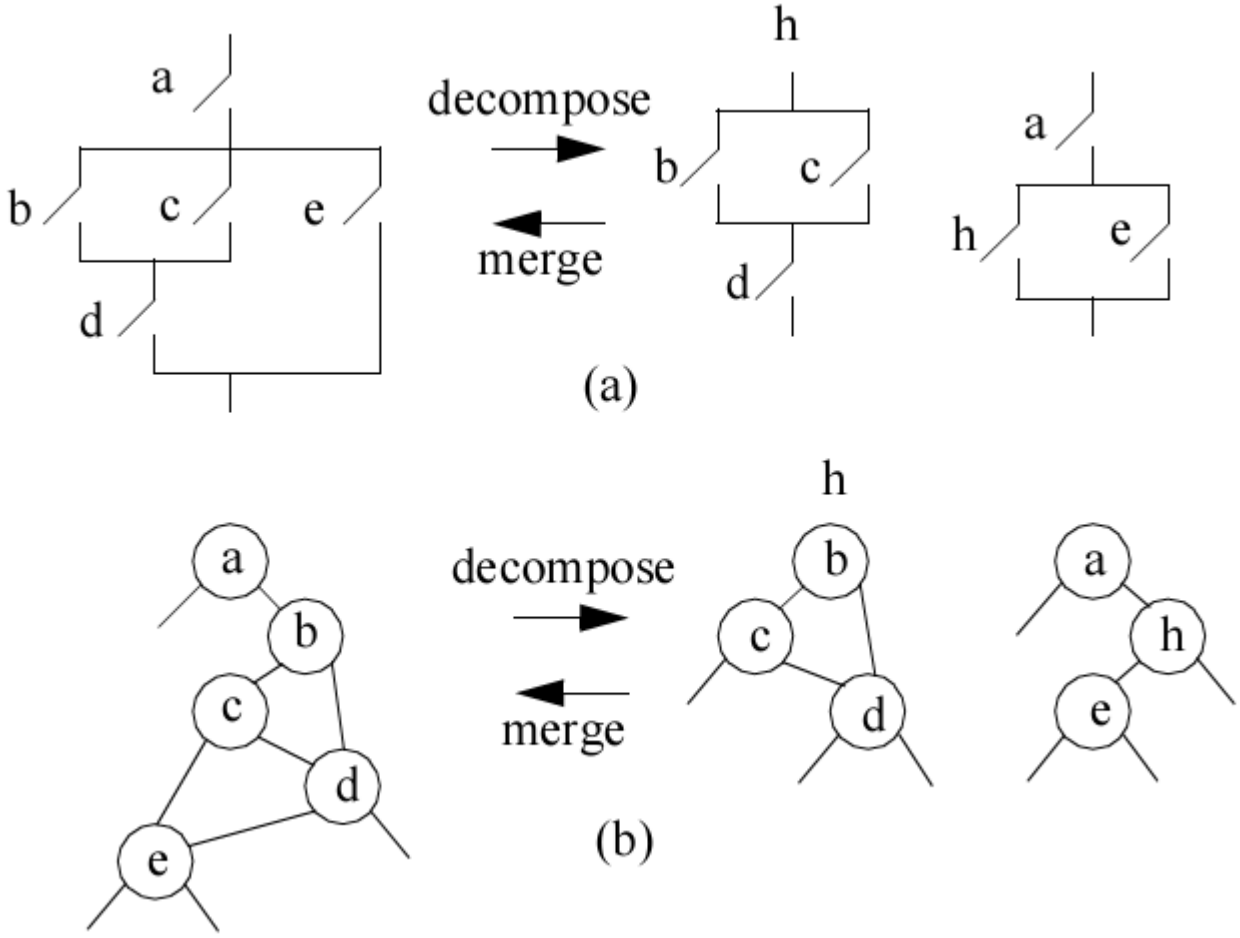
Outline

- Introduction
- Several words about general BDD
- SP-BDD: motivation
- SP-BDD: basic definitions and theorems
- SP-BDD: basic algorithms
- SP-BDD: basic applications
- SP-BDD: application in detail – fast delay calculation
- SP-BDD: application in detail – resynthesis
- Conclusion

SP-BDD: basic algorithms

- Reordering
- Merging (2 -> 1)
- Decomposition (1->2)
- Extraction (from SP-network netlist)

Decomposition and merging for SP-networks (a) and corresponding SP-BDDs (b)



Outline

- Introduction
- Several words about general BDD
- SP-BDD: motivation
- SP-BDD: basic definitions and theorems
- SP-BDD: basic algorithms
- SP-BDD: basic applications
- SP-BDD: application in detail – fast delay calculation
- SP-BDD: application in detail – resynthesis
- Conclusion

SP-BDD: basic applications

Basic applications of SP-BDD are currently connected with digital CMOS circuits (CCMOS)

Definition 9. ***Linear order associated with CMOS gate*** **is linear order associated with its pull-up SP-network, satisfying the following condition: if we consider it as linear order on corresponding pull-down transistors, then it is associated with pull-down SP-network.**

linear order associated with CMOS gate is unique

Definition 10. ***SP-BDD associated with CMOS gate*** **is SP-BDD associated with its pull-up network, constructed with use of linear order associated with this CMOS gate.**

Since this linear order is unique, associated SP-BDD (gate BDD) is canonical (unique) representation for CMOS gate. It represents both Boolean function and topology of CMOS gate. This representation is convenient for many purposes and applications.

Two useful operations on gate BDD:

- SP-restriction
- Minimization of gate BDD

Basic applications of SP-BDD:

- Fast delay calculation
- Resynthesis

Outline

- Introduction
- Several words about general BDD
- SP-BDD: motivation
- SP-BDD: basic definitions and theorems
- SP-BDD: basic algorithms
- SP-BDD: basic applications
- SP-BDD: application in detail – fast delay calculation
- SP-BDD: application in detail – resynthesis
- Conclusion

SP-BDD: application in detail – fast delay calculation

All delays of CMOS gate, both rise and fall,

are calculated using only four gate BDD linear passes

Outline

- Introduction
- Several words about general BDD
- SP-BDD: motivation
- SP-BDD: basic definitions and theorems
- SP-BDD: basic algorithms
- SP-BDD: basic applications
- SP-BDD: application in detail – fast delay calculation
- SP-BDD: application in detail – resynthesis
- Conclusion

SP-BDD: application in detail – resynthesis

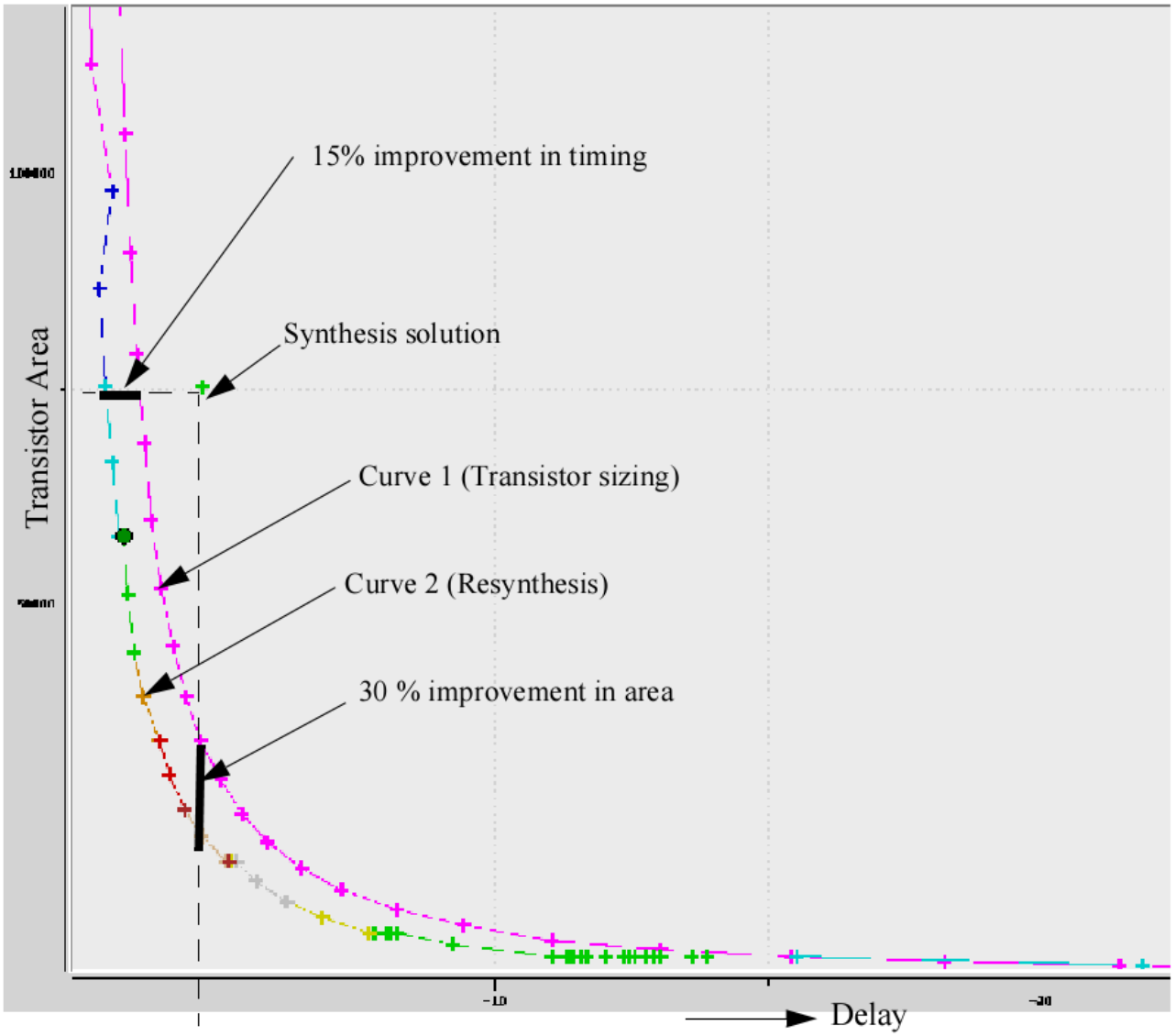
Sequential covering of large combinational circuit with windows (connected subcircuits)

Basic steps of local resynthesis:

- **DeMorgan transform**
- **reordering**
- **decomposition**
- **merging**

Use of simulate annealing with oscillating schedule

Resynthesis results



Resynthesis results