Efficient Methods for a Simple Disjoint Decomposition and a Non-Disjoint Bi-Decomposition

Hiroshi Sawada, Shigeru Yamashita and Akira Nagoya

{sawada,ger,nagoya}@cslab.kecl.ntt.co.jp

NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, JAPAN

1

Outline

- Background
- Our methods
  - Simple Disjoint Decomposition (SDD)
  - non-disjoint Bi-Decomposition (Bi-Decomp)
- Experimental results
- Conclusions

2

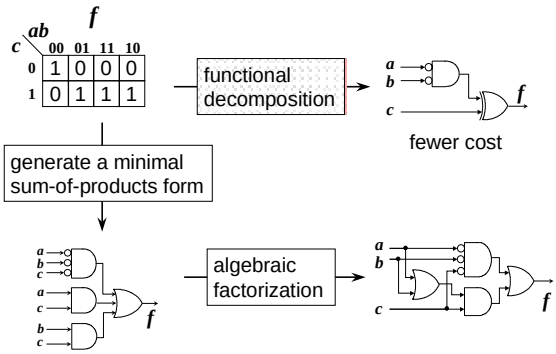
Background

Decomposition of a Boolean function

- Algebraic factorization / decomposition [1]
  - on a sum-of-products form
  - very efficient in terms of the computation time
- Functional decomposition [2-7]
  - a more powerful technique
  - manipulates a Boolean function directly
  - the BDD technique [8-10]

3

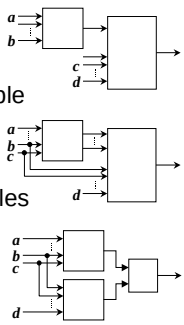
Example



4

Functional Decompositions

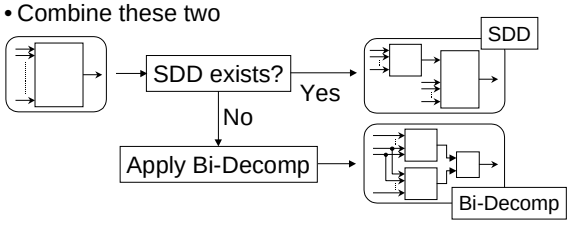
- Simple disjoint decomposition [2,3]
  - no common variable
  - only one newly introduced variable
- Roth-Karp decomposition [4]
  - common variables
  - multiple newly introduced variables
- Non-disjoint bi-decomposition [5-7]
  - common variables
  - decomposes in parallel



5

Our Accomplishment

- Efficient methods for
  - Simple Disjoint Decomposition (SDD) [11]
  - non-disjoint Bi-Decomposition (Bi-Decomp) [12]
- Combine these two



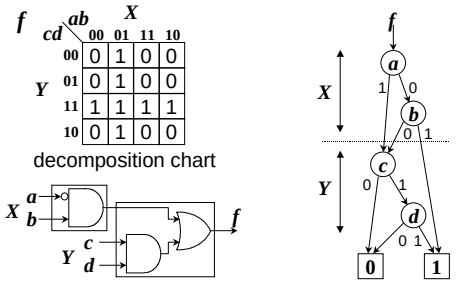
6

Outline

- Background
- Our methods
  - Simple Disjoint Decomposition (SDD)
  - non-disjoint Bi-Decomposition (Bi-Decomp)
- Experimental results
- Conclusions

7

Simple Disjoint Decomposition

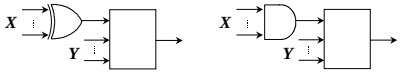


How to find a variable partition **{ *X, Y* }** that gives a decomposition efficiently?

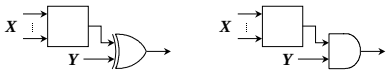
8

Our Approach

- Limit the types of decompositions to be found
  - ***X*** is a set of symmetric variables



  - the size of ***Y*** is 1



- Many of the other types can be found
  - by applying the above two recursively

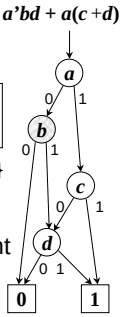
9

Detecting Symmetries of Two Variables

- Detect asymmetry [13]
  - from the structure of a BDD
    - dependent on ***b*** and ***d***
    - independent of ***c***

  ⇒ asymmetric in **{*b, c*}** or **{*d, c*}**

- Detect symmetry
  - expensive if two variables are distant
  - for pairs of variables not filtered out



10

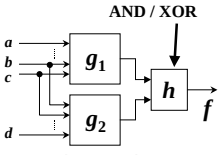
Outline

- Background
- Our methods
  - Simple Disjoint Decomposition (SDD)
  - non-disjoint Bi-Decomposition (Bi-Decomp)
- Experimental results
- Conclusions

11

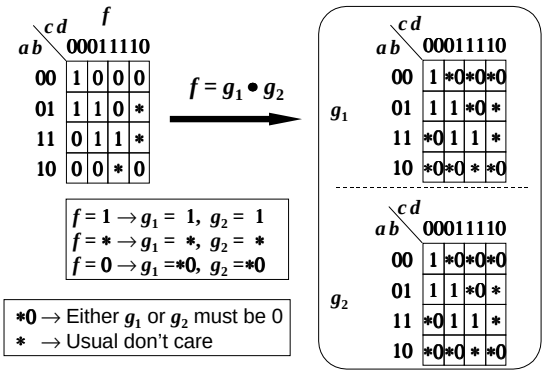
Non-Disjoint Bi-Decomposition

- Consider two cases for ***h***
  - 2-input AND
  - 2-input XOR
- Produce an initial solution
- Improve it by the Branch-and-Bound search
  - Cost: the total number of variables in g_1 and g_2
  - Each step: a variable elimination procedure



12

Initial Solution (AND-decomposition)

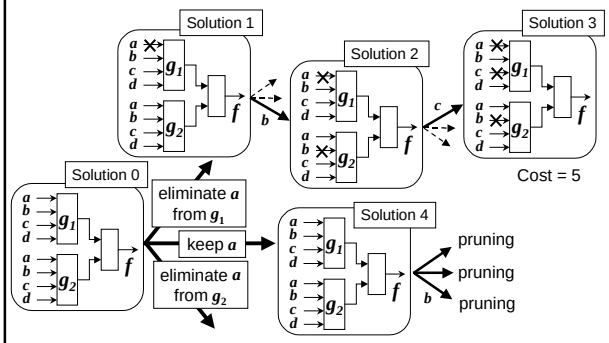


$f = 1 \rightarrow g_1 = 1,\ g_2 = 1$
$f = * \rightarrow g_1 = *,\ g_2 = *$
$f = 0 \rightarrow g_1 = *0,\ g_2 = *0$

$*0 \rightarrow$ Either g_1 or g_2 must be 0
$* \rightarrow$ Usual don't care

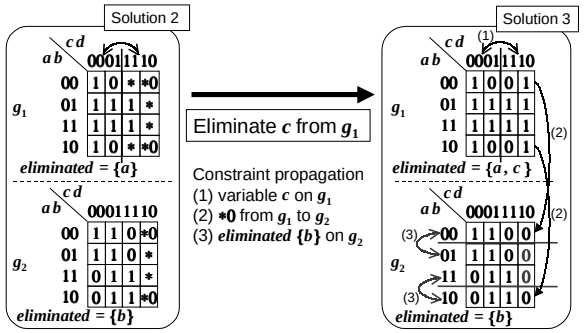
13

Overview of the Branch-and-Bound search



14

Variable Elimination



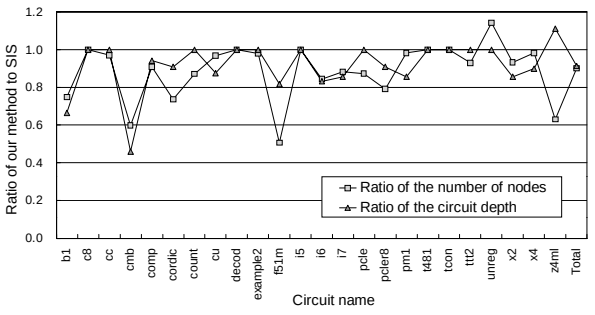
15

Experiment

- For multi-level logic circuits
- Collapsed each maximum fanout free cone into a large node
- Decomposed each node recursively until 2-input
  - using our method
    - SDD + Bi-Decomp
  - using SIS [14] for comparison
    - "decomp" (algebraic decomposition) + "map"

16

Experimental Results



- Execution time: around one second

17

Conclusions

- Two functional decomposition methods
  - Simple disjoint decomposition
    - optimal circuit structure
    - applicable only if it exists in a function
  - Non-disjoint bi-decomposition
    - produce circuits with fewer depths
- Future works
  - Other criteria for the best bi-decomposition

18

Algebraic Decomposition [1], Functional Decomposition [2, 3, 4], Bi-Decomposition [5, 6, 7], Functional Decomposition using BDD [8, 9, 10], Related Publication by the Authors [11, 12], Detecting Symmetric Variables [13], SIS [14]

References

[1] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. R. Wang, "MIS: A Multiple-Level Logic Optimization System," *IEEE Trans. CAD*, vol. CAD-6, pp. 1062–1081, Nov. 1987.

[2] R. L. Ashenhurst, "The Decomposition of Switching Functions," in *Proc. an Int'l Symposium on the Theory of Switching*, pp. 74–116, Apr. 1957.

[3] H. A. Curtis, *A New Approach to the Design of Switching Circuits*. Van Nostrand, 1962.

[4] J. P. Roth and R. M. Karp, "Minimization Over Boolean Graphs," *IBM Journal*, pp. 227–238, Apr. 1962.

[5] D. Bochmann, F. Dresig, and B. Steinbach, "A new decomposition method for multilevel circuit design," in *Proc. EDAC*, pp. 374–377, Feb. 1991.

[6] B. Steinbach and A. Wereszczynski, "Synthesis of Multi-Level Circuits Using EXOR-Gates," in *Proc. of Reed-Muller'95*, pp. 161–168, Aug. 1995.

[7] T. Sasao and J. T. Butler, "On Bi-Decompositions of Logic Functions," in *Notes of International Workshop on Logic Synthesis (IWLS'97)*, May 1997.

[8] S. Chang and M. Marek-Sadowska, "Technology Mapping via Transformations of Function Graphs," in *Proc. Int'l Conf. Computer Design*, pp. 159–162, Oct. 1992.

[9] Y.-T. Lai, M. Pedram, and S. Vrudhula, "BDD Based Decomposition of Logic Functions with Application to FPGA Synthesis," in *Proc. Design Automation Conf.*, pp. 642–647, June 1993.

[10] T. Sasao, "FPGA Design by Generalized Functional Decomposition," in *Logic Synthesis and Optimization* (T. Sasao, ed.), pp. 233–258, Kluwer Academic Publishers, 1993.

[11] H. Sawada, S. Yamashita, and A. Nagoya, "Restructuring Logic Representations with Easily Detectable Simple Disjunctive Decompositions," in *Proc. Design, Automation and Test in Europe Conf.* , pp. 755–759, Feb. 1998.

[12] S. Yamashita, H. Sawada, and A. Nagoya, "New Methods to Find Optimal Non-Disjoint Bi-Decompositions," in *Proc. ASP-DAC*, pp. 59–68, Feb. 1998.

[13] D. Möller, J. Mohnke, and M. Weber, "Detection of Symmetry of Boolean Functions Represented by ROBDDs," in *Proc. Int'l Conf. Computer-Aided Design*, pp. 680–684, Nov. 1993.

[14] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli, "SIS: A System for Sequential Circuit Synthesis," Memorandum UCB/ERL M92/41, Univ. of California, Berkeley, May 1992.