A Tool to Compare Pass Transistor Logic and Static CMOS Complex Gates Logic with Minimal Implementations

Vinícius Pazutti Correia, André Inácio Reis
{vincor, andreis}@inf.ufrgs.br

Universidade Federal do Rio Grande do Sul - UFRN
Instituto de Informática
Cx. Postal 15064 - Av. Bento Gonçalves, 9500 - Campus do Vale - Bloco IV
Bairro Agronomia - Porto Alegre - RS - Brasil - 91501-970

Abstract

This paper presents a tool designed to compare different implementations of integrated circuits with switch logic. By specifying an up-to-4-input Boolean function it is possible to obtain the minimal implementation by using either pass transistor logic or static CMOS complex gate implementation. To obtain the minimal implementations, the tool works with the concept of NPN and P equivalencies between functions. These concepts are fundamental to find out which cells should be used to compose a cell library. This equivalence between functions can be verified with this tool.

1. Introduction

The final implementation of Boolean functions is usually done through the use of CMOS transistors as switches. Two main logic families can be used to perform this implementation: static CMOS logic [IBA 71] or pass transistor logic [BUC 97] [BER 97]. An algorithm for implementation based on static CMOS logic with a minimum number of gates is presented in [IBA 71]. The minimal implementation using pass transistor logic [BUC 97] [BER 97] can be directly derived from a ROBDD [BRY 86] data structure. For each function, the minimal implementation considering the number of transistors can be obtained by using either logic families.

2. Objectives of the Tool

The main goal of the tool is to provide a support tool to derive minimum implementations for Boolean functions using CMOS transistor switches. This enables the possibility of learning how a Boolean function is implemented by using CMOS transistors switches. The tool is also able to derive and show the other functions or truth tables which are equivalent to the former. The equivalence concept is determinant while choosing logic functions to compose an efficient cell library.

3. Tool Features

The following features are supported by this switch logic tool.

Logic Function Specification – The logic function is specified by a truth table that can be entered by minterms or by reading a blif file describing the implicant cubes of the function.

Minimal static CMOS logic implementation – Once a logic function has been specified the user is able to obtain a minimal implementation for the function by using static CMOS complex gates. The method used in this implementation is presented in [IBA 71].

Minimal pass transistor logic implementation – The user may also choose generating minimal implementations using pass transistor logic. The method used for generating this implementation based on the construction of a minimum sized ROBDD. The derivation of a pass transistor network from a ROBDD is straightforward. The size of a ROBDD is highly dependent on variable ordering [STE 90]. The use of worst case orderings for ROBDD is also important for didactic reasons, therefore the user is also allowed to derive a worst case implementation for pass transistor logic. This feature gives the user the idea of how the implementation size depends on the order of the selection variables.

NPN and P functional equivalencies – When a function is specified, the tool is able to detect other functions that are equivalent. Two kinds of equivalencies can be detected: NPN and P equivalencies. Table 1 shows the number of different 4-input functions, P and NPN classes, according to the size of its ROBDD representation. First column shows the BDD sizes considering the number of non-terminal nodes. BDD sizes vary from 0 to 9, and this is compatible with the upper bound for BDD size derived in [HEH 92]. All the BDD sizes in Table 1 were obtained with the same variable ordering.

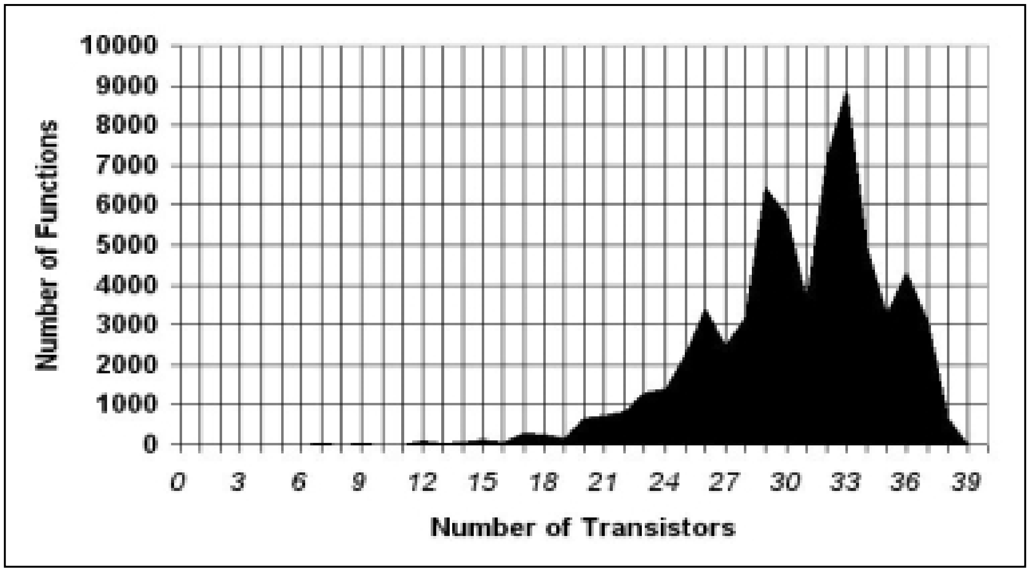

Figure 1 - Distribution of number of transistors used versus number of functions implemented.

The notion of equivalent functions and equivalency classes is very important when performing technology mapping [DET 87], because this concepts are used to match part of a logic function to be implemented with cells from a library. This matching operation is based on the concept of functional equivalence.

Tool output – The tool is able to write the implemented circuits in SPICE format.

Tool limitations – The tool is limited to 4-input single output logic functions. These limitations guarantee the optimality of the result as well as the possibility to derive equivalent functions in a fast way.

4. Results

By using the tool, the Table 1 shows a summary of all equivalencies within the complete set of 4-input functions was built. The most relevant results are that there exists only 3984 P-Classes where all 65536 4-input functions fit. This shows that a library that covers a large number of functions can be built using only a small set of cells that really have to be designed.

Table 1 - Number of different 4-input functions, P and NPN classes according to the size of its BDD representation.

BDD Size	**Functions**	**P Classes**	**NPN Classes**
0	2	2	1
1	8	2	1
2	48	6	1
3	236	26	4
4	960	204	14
5	3248	710	38
6	8928	1342	70
7	17666	1272	68
8	23280	420	25
9	11160	0	0
Total	**65536**	**3984**	**222**

The tool has also been used to verify the number of transistors used to implement all 4-input functions with CMOS pass transistor. The Figure 1 shows the distribution of number of transistors used versus number of functions implemented.

.

4. Conclusions

This paper presented a tool to compare switch logic implementations. This tool allows the comparison between any up-to-4-input Boolean function and is able to implement it with CMOS switch logic. Future work can include other implementation forms like the ones used in [CHA 72] [CUL 79] and a complete comparison of all 65536 existent functions.

References

[IBA 71] T. Ibaraki, S. Muroga, "Synthesis of Networks with a Minimun Number of Negative Gates", IEEE Transactions on Computers, Vol. C-20, No. 1,pp. 49-58, January 1971.

[BUC 97] P. Buch, A. Narayan, A. R. Newton, A. Sangiovanni-Vincentelli, "On Synthesizing Pass Transistor Networks", IWLS '97, pp. 101-108.

[BER 97] V. Bertacco, S. Minato, P. Verplaetse, L. Benini, G. de Micheli, "Decision Diagrams and Pass Transistor Logic Synthesis", IWLS '97, pp. 109-113, May 1997.

[BRY 86] R.E.Bryant. "Graph-based algorithms for Boolean function manipulation", IEEE Transactions on Computers, vol. C-35, n° 8, pp. 677-691, August 1986.

[STE 90] Steven J. Friedman and Kenneth J. Supowit, "Finding the Optimal Variable Ordering for Binary Decision Diagrams", IEEE Transactions on Computers, Vol. 39, No. 5,pp. 710-713, May 1990.

[HEH 92] Heh-Tyan Liaw and Chen-Shang Lin, "On the OBDD-Representation of General Boolean Functions", IEEE Transactions on Computers, Vol. 41,No. 6, pp. 661-664, June 1992.

[DET 87] E. Detjens, G.Gannot, R.Rudell, A.L.Sangiovanni-Vinccentelli, A.Wang. "Technology mapping in MIS" ICCAD, 1987, pp. 116-119.

[CHA 72] Charles R. Baugh, C. S. Chandersekaran, Richard S. Swee and Saburo Muroga, "Optimal Networks of NOR-OR Gates for Functions of Three Variables", IEEE Transactions on Computers, Vol. C-21, No. 2,pp. 153-160, February 1972.

[CUL 79] J. N. Culliney, M. H. Young, T. Nakagawa and S. Muroga, "Results of the Synthesis of Optimal Networks of AND and OR Gates for Four-Variable Switching Functions", IEEE Transactions on Computers, Vol. C-27, No. 1,pp. 76-85, January 1979.