

User-Friendly Tracing of BDDs

Sérgio Queiroz de Medeiros, David Déharbe
{sergio,david}@consiste.dimap.ufrn.br

Universidade Federal do Rio Grande do Norte - UFRN
Departamento de Informática e Matemática Aplicada – DIMAp
Laboratório de Concepção de Sistemas - Consiste

Abstract

*BDDs are a data structure widely used in the formal verification and synthesis of circuits nowadays. **BDDmeter** is a graphical extension to BDD libraries and allow visualizing the dynamical behavior of the BDDs. **BDDmeter** makes it possible to see, as graphic or as text, important informations for the user and designer of BDD-based tools, making it easier to understand unexpected bottlenecks during execution.*

1 Introduction

Binary Decision Diagrams, or BDDs [BRY87], are a data structure used nowadays in several activities of integrated circuit design. BDDs are a canonical representation of Boolean functions provided that the order of input variables is fixed. In addition to this uniqueness property, a BDD can be exponentially more compact than its corresponding truth table representation. But the BDD for a given function can have many different forms, depending on the permutation of the input variables - and sometimes the size of BDDs greatly varies with the order.

If the function is very sensitive to the variable ordering, its BDD may vary extremely (from linear to exponential size). Such functions often appear in real world application. Several authors have proposed heuristics to help solve this problem [RUD93, ATB94, KF98].

BDDmeter make it possible to see in real time the evolution of the characteristics of the BDDs, including during variables reordering and to understand which variables contribute to the bottleneck.

Our tool, **BDDmeter**, is a visual plug-in for BDD libraries. **BDDmeter** shows graphically in real time several data related to BDDs, such as number of nodes for each variable, the total size of the BDD, beyond other informations that aim to help the user of these libraries.

2 Tool description

2.1 Using BDDmeter

To use **BDDmeter**, the client application uses the usual methods of the BDD library. When some graphical interface function is required, this called is transfered to the graphical library. Three functions are added to BDDs library related to the graphical interface:

- Ⓣ**bdd_gui_init** dispatches the *thread* of the graphical interface. This thread accesses the informations of the BDDs library in mode read only.
- Ⓣ**bdd_gui_message** allows the client application to send messages for the user through a dedicated text window in the graphical interface.
- Ⓣ**bdd_gui_quit** stops the execution of the graphical interface *thread*.

The *threads* [SUN96] were implemented with the POSIX library (pthreads), and the graphical interface with the library Qt [TRO00]. Both libraries are free in a large number of plataforms.

It is important to emphasize that including **BDDmeter** in client applications of the BDD libraries is therefore very easy and may be done in few minutes by the developer of the application.

2.2 Description of the Interface

The interface of **BDDmeter** has a main window (fig. 1), where three other windows may be activated: (1) the interface shows a histogram that represent the number of nodes associated the each variable (fig. 2); (2) the graphics of the evolution of the total BDD size; (3) several statistics about the BDDs (fig. 3)

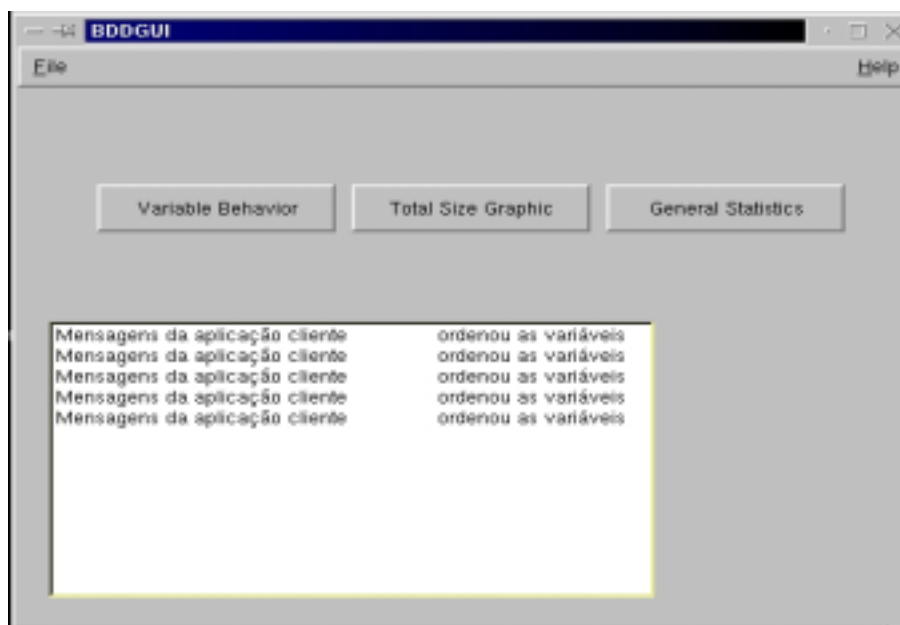


Figure 1 - Main window.

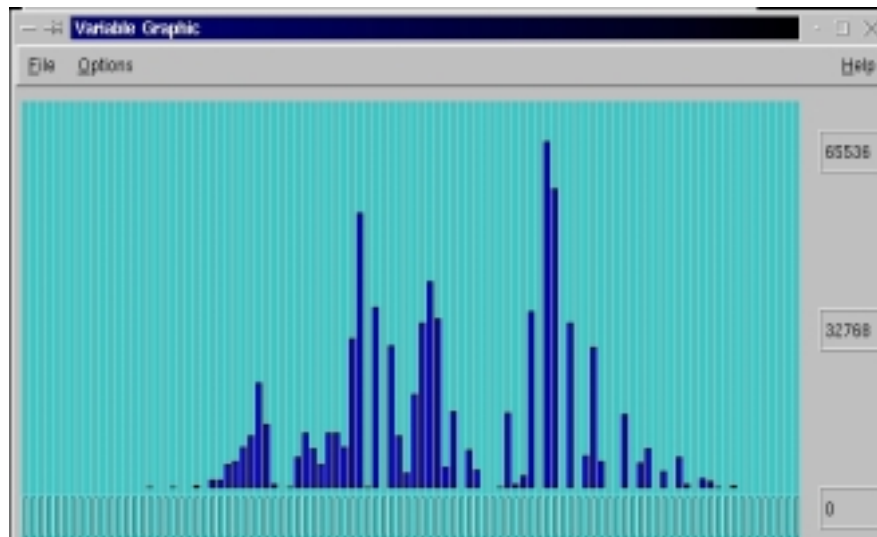


Figure 2 - Graphic of the variables.

The informations of each window of **BDDmeter** are constantly refreshed, and the user can choose the time interval between the refreshes.

In the window of fig. 2, the user can choose how the size of each variable should be displayed: individual size of each variable (histogram) or the total size associated until that variable (accumulating histogram).

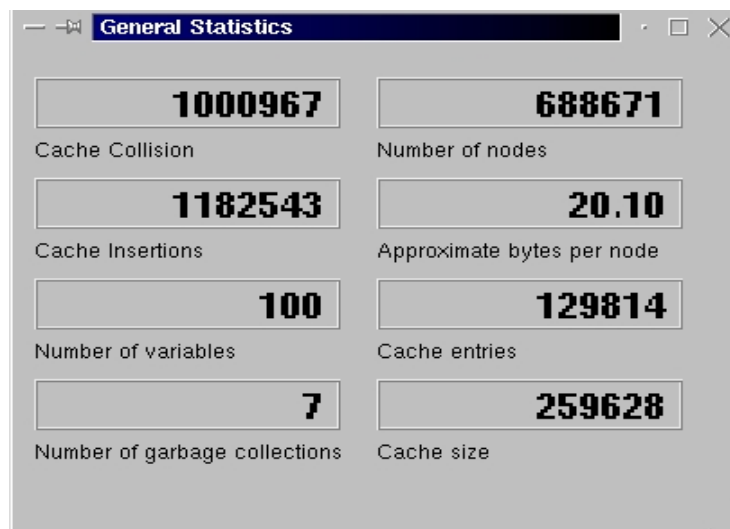


Figure 3 - General statistics.

3 Conclusions

The tool is still being improved, as prototypes were presented to researchers working on the development of BDD-based verification tools. There are good indications that the idea of allowing a dynamic graphical visualization of BDDs is useful for the BDD community. **BDDmeter** was quickly incorporated with a formal verification tool, fulfilling one initial requisite.

New functionalities are constantly added to **BDDmeter**, trying to supply necessities of possible users of the tool.

Finally, we are initiating a new project, **BDDtuner**, that it will make it possible to dynamically modify the configuration of the BDD. For instance, the user will be able to reorder the variables at any moment of the execution.

4 References

- [ATB94] AZIZ, A., TASIRAN, S., and BRAYTON, R. K. BDD variable ordering for interacting finite state machines. In **31st annual conference on Design Automation conference: DAC'94**, pages 283-288, 1994.
- [BRY86] BRYANT, R. E., Graph-based algorithm for boolean function manipulation. **IEEE Transactions Computers**, C(35):1035-1044, 1986.
- [KF 98] KAMHI, G. and FIX, L. Adaptative variable ordering for symbolic model checking. In **1998 IEEE/ACM international conference on Computer-aided design: ICCAD'98**, pages 359-365, 1998.
- [RUD 93] RUDELL, R. Dynamic variable ordering for ordered binary decision diagrams. In **1993 IEEE/ACM international conference on Computer-aided design: ICCAD'93**, pages 42-47, 1993.
- [SUN96] SUN Microsystems. **Programming with threads**, 1996.
- [TRO 00] TROLLTECH. **QT Reference Documentation** (version 2.2.1), 2000.