Development of practical ATPG tool with flexible interface

Masayoshi YOSHIMURA  
FLEETS  
Sawara-ku, Fukuoka, 814–0001, Japan  
yosimura@fleets.jp

Yusuke MATSUNAGA  
Kyushu University  
Sawara-ku, Fukuoka, 814–0001, Japan  
matsunaga@slrc.kyushu-u.ac.jp

An ATPG tool is constructed by many functions which are pattern generation functions, fault simulation functions, static compaction functions, etc. For evaluating new idea of only one subfunction accurately, it is necessary to develop all functions of the ATPG tool. It is a serious problem for evaluating new ideas.

Many of efficient techniques like static learning don’t satisfy all evaluation standards of ATPG tools (high fault efficiency, short length of test patterns, and short CPU time, etc.). Effectiveness of these efficient techniques is according to the structure of circuits. If ATPG tools aren’t practical, these techniques are not accurately evaluated.

In FLEETS, practical and open EDA tools include ATPG tool have been developed for evaluating new ideas. The purpose of developing these EDA tools is to provide environment that facilitates evaluating new ideas. An ATPG tool which has flexible interfaces is developed as a part of environment for developing EDA tools.

This ATPG tool has been developed for evaluating new ideas. To achieve this propose, new algorithms as well as the existing algorithms have been developed. Flexible command interfaces are designed to facilitate adding new ideas in the future. The command interface has been written in Tcl/Tk[2]. Table 1 shows the command interfaces.

A flexible interface was added to this ATPG tool to conduct various experiments. One of examples is that relations between order of selecting faults, number of patterns and CPU Time are examined. Another is that correlation of the number of test patterns and processing time to combination of each algorithm of pattern generation, static compaction, and fault simulation is examined. And, relations between results of static learning and pattern generation are examined.

This ATPG tool has been written in C language and can be executed on Linux and Solaris. This ATPG tool was evaluated to combinational circuits of ITC99 benchmark circuits[1]. A fault model was the single stuck-at fault model. The ATPG tool was executed on Pentium4 3.05GHz with 1GB Memory running Linux OS. In the result to 21 all circuits, the number of patterns was 776, processing time were 459.3 seconds and fault efficiency were 99.74% on average.

Practical and open ATPG tool for evaluating new ideas has been described. New ideas of ATPG tools can be easily evaluated in a short term.

One of the problems in the future is to enable an addition of the algorithm by the user. For solving this problem, it will be designed APIs of ATPG tool which is used by users when users will develop the new algorithms.

Acknowledgment

This research was supported by a grant of Fukuoka project in the Cooperative Link of Unique Science and Technology for Economy Revitalization (CLUSTER) of Ministry of Education, Culture, Sports, Science and Technology (MEXT).

The authors thank the members of System JD Co., Ltd for supporting to evaluation of ATPG tools.

References