Exploiting Test Resource Optimization in Data Path Synthesis for BIST *

Xiaowei Li
Dept. of Computer Science and Technology
Peking University
Beijing 100871, P.R.China

Paul Y.S.Cheung
Dept. of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong

Abstract

Area and test time are two major overheads encountered during data path synthesis for BIST. This paper presents an attempt towards testability enhancement in data path BIST synthesis by considering two factors simultaneously. It is achieved by incorporating two testability constraints in data path synthesis. Experimental results are presented to demonstrate the effectiveness of the proposed (data path) BIST synthesis approach.

1. Introduction

Investigation of synthesis methods that take into account BIST techniques has recently received attention from the research community. These BIST-oriented approaches usually assume the presence of a PRPG for test pattern generation and a MISR for responses compression. The modules which are required to perform a test (i.e., PRPG and MISR) are known as *test resources*. Since the BIST logic is combined with the system logic, opportunities exist for the synthesis techniques to generate hardware that can be shared by both the system and test operation, resulting in improved performance and reduced cost.

Area and test time are major overheads encountered when using BIST techniques. Approaches have been proposed either towards the minimal (extra) area overhead solution (MAOS), or the minimal test (application) time solution (MTTS). Both approaches have merits and limitations. In general, the area overhead of MTTS may be too high and the test time of MAOS may be too long. Neither is acceptable when both attributes are important.

The MAOS-oriented approaches focus on minimizing area overhead. The basic approach is to maximize the sharing of test registers resulting in a fewer number of registers being modified for BIST [1-3]. The goal is to reduce BIST area overhead without sacrificing the quality of the test. The main limitation is the test time may be too long. The MTTS-oriented approaches focus on exploiting test concurrency. The basic approach is to maximize the test concurrency resulting a fewer test session [4]. The goal is to reduce test time. The main limitation is the area overhead may be too high.

MAOS-oriented approaches dominate current research. A prime concern in BIST implementation is the area overhead due to the modification of normal registers to be test registers. One of the difficulties is the *register self-adjacency* problem. To cope with it, methods have been proposed either to avoid producing such self-adjacent register [2] or to minimize the number of it [1] during synthesis process.

The key aspect of this study is to propose an approach to BIST resource optimization. This is formulated as an 0/1 integer linear programming. Experimental results on academic benchmarks demonstrate the effectiveness of the proposed approach.

2. Test Resource Optimization

The synthesis model used in this work assumes single-cycle operations, individual register storage and point-to-point multiplexer (MUX) connectivity.

Give a scheduled data flow graph (DFG) and a module assignment, we exploit register assignment process with the objective of minimizing the cost:

$$C = w_1 T + w_2 A \qquad (1)$$

where T and A represent test application time and test resources cost, respectively. w_1 and w_2 are non-negative weighting factors.

2.1 Data Path Behavioral Synthesis

The register assignment problem can be modeled as coloring register conflict graph (RCG). A coloring of this graph corresponds to a valid register assignment with each color corresponding to a register. All nodes with the same color can be mapped to the same register in the final implementation.

We define the test resource sharing effectiveness $E(v)$ of a variable v as the sum of the number of modules for which v is a PRPG-variable and the number of modules

* This work was supported in part by the *Croucher* Foundation Grant #360/062/0994.

0-7695-0104-4/99 $10.00 © 1999 IEEE

for which v is a MISR-variable. We associate a E(v) with each variable v (node in the RCG). Using this measure, the assignment process can be guided by choosing merges that result in large increases in the E(v) of registers. Details of the metrics, and the algorithm which use it (to color the RCG), can be found in [8].

2.2 Structural Synthesis for BIST

The BIST resources minimization problem is to decide which register to modify and the modes (PRPG, MISR or BILBO) to add, in such a way that every module in the given data path can be tested with minimal number of registers being modified as test registers.

Consider a data path comprising m modules M^k (k=1,2,...,m) and r register R^i (i=1,2,...,r). We associate with each register R^i, three integer variables, namely, R^{ip}, R^{im} and R^{ib}. Each of them has a value of 0 or 1. In a BIST solution, each register R^i is either a normal register (unmodified for BIST) or one type of a BIST register, i.e. PRPG, MISR, or BILBO. This gives us the following constraint for every R^i,

$$R^{ip} + R^{im} + R^{ib} \leq 1 \qquad (2)$$

As a BIST solution, any module M^k should have at least one input register act at pattern generator. It also has at least one output register act as response compressor. A PRPG or BILBO can be used to generate necessary test patterns. An MISR or BILBO can be used to compress test responses. This gives us the constraints for covering every *input* port of M^k with a BIST register:

$$\sum (R^{ip} + R^{ib}) \geq 1 \qquad (3)$$

and covering every *output* port of M^k with a BIST register:

$$\sum (R^{im} + R^{ib}) \geq 1 \qquad (4)$$

We associate a modification cost with each register: C^{ip}, C^{im} and C^{ib} are costs of a PRPG, MISR and BILBO, respectively. The total area of the test registers is:

$$A = \sum_{i=1}^{r} (C^{ip} * R^{ip}) + \sum_{i=1}^{r} (C^{im} * R^{im}) + \sum_{i=1}^{r} (C^{ib} * R^{ib}) \qquad (5)$$

The size of the ILP is linear with respect to the number of registers and modules in the synthesized data path.

2.3 Optimized BIST Solution

If two modules share PRPG-resources, they can receive test pattern simultaneously. If they share a MISR-resource, they can not compress test responses concurrently. Hence, sharing of MISR-resources increases test sessions. The trade-off between the BIST resources sharing and the test application time has to be made when using objective (1).

To make this problem more tractable, solutions for objective (5) has to be partitioned into groups, only the one with minimal test sessions is considered. The detailed procedure can be found in [8].

4. Experimental Study

We present some results on three high-level synthesis benchmarks (the 2nd order differential equation *DiffEq* [5], the Tseng's DFG *Tseng* [6] and the auto regression filter element *AR_Filter* [7]) in table 1.

Table 1. Experimental Results on Benchmarks.

DFG	*DiffEq*	*Tseng*	*AR_filter*
Module Assignment	1+, 2*, 1-	2+, 1*, 1-	4+, 8*
# Register	7	8	16
# MUX	26	31	48
# CBILBO	1	2	0
# BILBO	2	1	7
# PRPG	2	2	8
# MISR	2	2	1

5. Conclusions and Ongoing Work

In this paper, we exploited test resources optimization during data path synthesis for BIST. A trade-off between extra area and test application time was given. Approach and experimental results were outlined. There are several ways the presented approaches can be improved upon [8].

As part of our future work, the proposed test resource optimization approach will be tested on more academic (and industrial) benchmark circuits.

References

[1] L.Avra,"Allocation and Assignment in High-Level Synthesis for Self-Testable Data Paths", *Proc. of ITC'91*, pp.463-472

[2] C.Papachristou, S. Chiu and H.Harmanani, "SYNTEST: a method for high-level SYNthesis with self-TESTability", *Proc. of IEEE ICCDS'91*, pp.458-462

[3] I.Parulkar, S.Gupta and M.Breuer, "Data Path Allocation for Synthesizing RTL Designs with Low BIST Area Overhead", *Proc. of ACM/IEEE DAC.* 1995, pp.395-401

[4] A.Orailoglu and I.G.Haris, "Microarchitectural Synthesis for Rapid BIST Testing", *IEEE Trans. on CAD,* Vol.16, No.6, 1997, pp.573-586

[5] P.G.Paulin and J.P.Knight, "Force-Directed Scheduling for the Behavioral Synthesis of ASICs", *IEEE Trans. on CAD.,* Vol.8, No.6, 1989, pp.661-679

[6] C.-J.Tseng and D.P. Wiewiorek, "Automated Synthesis of Data Paths in Digital Systems", *IEEE Trans. on CAD*, Vol.5, No.3, 1986, pp.379-395

[7] R.Jain, A.C. Parker and N.Park, "Predicting System-Level Area and Delay for Pipelined and Non-pipelined Designs", *IEEE Trans. on CAD,* Vol.11, No.8, 1992, pp.955-965

[8] X.Li, et.al, "Exploiting Test Resource Optimization in Data Path Synthesis for BIST", *Technical Report*, Sept., 1998