Optimization of Testability of Sequential Circuits Implemented in FPGAs with Embedded Memory*

Andrzej Krasniewski

Warsaw University of Technology, Institute of Telecommunications
Nowowiejska 15/19, 00-665 Warsaw, Poland
andrzej@tele.pw.edu.pl

Abstract. A method for testability-oriented optimization of sequential circuits implemented using FPGAs with embedded memory is presented. It specifies the content of those memory words which are not defined by the conventional FSM synthesis. The experimental results confirm its effectiveness; for the largest examined circuit, the self-test session required to achieve an acceptable level of fault escapes for the optimized design, obtained using the proposed procedure, is almost 10^6 times shorter than for the non-optimized design. The proposed method does not involve any extra circuitry or speed degradation. Also, it does not require any extra reconfiguration during self-testing.

1 Introduction

Modern FPGAs contain embedded memory. Embedded memory, in its ROM-like mode of operation (with its content determined at the time of programming) can be used to implement complex combinational or sequential circuits. In this paper, we focus on implementations of finite state machines (FSMs).

A simple implementation of an FSM, in which a memory module contains the entire combinational part of the circuit, is shown in Fig. 1(a). To reduce the size of the memory block, an alternative implementation, shown in Fig. 1(b), that includes an extra module, called *address modifier*, has been proposed [1]. Dedicated synthesis methods have been developed for the structure of Fig. 1(b) [2-4].

When the structures of Fig. 1 are implemented using FPGAs, the memory module is normally mapped to embedded memory blocks, whereas the address modifier is implemented with LUTs included in programmable logic blocks. The address register can be implemented either with flip-flops included in programmable logic blocks or as an internal register of the embedded memory array located at its input.

To thoroughly exercise the FPGA configuration corresponding to the specific user-defined application, application-dependent testing [5], also referred to as application-oriented test [6] or configuration-dependent testing [7], is performed. A number of techniques have been proposed for application-dependent testing (usually, self-testing) of sequential circuits implemented with LUT-based FPGAs [5, 7-10]. The

* This work was supported by the State Committee for Scientific Research of Poland under grant no. 4 T11D 014 24

J. Becker, M. Platzner, S. Vernalde (Eds.): FPL 2004, LNCS 3203, pp. 1067–1072, 2004.
© Springer-Verlag Berlin Heidelberg 2004

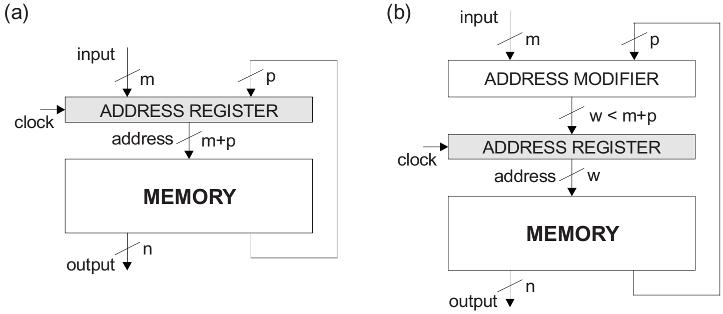

Fig. 1. Memory-based implementation of a finite state machine (FSM)

methods presented in [7-10] rely on a modification of the user-defined functions of LUTs (without changing the LUT interconnection structure) and, therefore, are not suitable for the memory-based FSM structures of Fig. 1. An analysis of the applicability of the self-test technique presented in [5] to FSMs implemented using embedded memory of FPGAs shows that a modified structure, in which an internal register of the embedded memory array located at its output serves as the FSM state/output register, is more suitable for testing than alternative FSM structures [11]. Therefore, in our study, we assume the BIST scheme shown in Fig. 2.

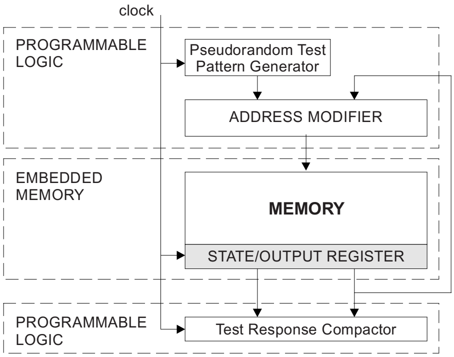

Fig. 2. Self-testing of an FSM implemented using an FPGA with embedded memory

2 Extension of Memory Specification

The method presented in this paper is intended to optimize the testability of an FSM defined by an incompletely specified next state function, that is implemented using an FPGA with embedded memory and is tested using the scheme of Fig. 2. The optimization relies on an appropriate specification of the content of those memory words which are left undefined by the conventional FSM synthesis procedure. This process is referred to as *extension of memory specification* or, simply, *memory*