ADOLT -- An ADaptable On - Line Testing Scheme for VLSI Circuits

A. Maamar & G. Russell
Department of Electrical & Electronic Engineering
The University
Newcastle upon Tyne NE1 7RU, England UK

Abstract

ADOLT permits the error detection capabilities of a CED scheme to be adapted to the error detection requirements of an application. This reduces the impact of the scheme on the design in terms of area overheads and the effect on performance. The scheme uses a slightly modified version of Dong's Code [1] and gives a more efficient implementation than previous methods[2,3].

1. Introduction:

Research has shown that as the scale of integration increases circuit operation is becoming more susceptible to intermittent and transient faults. Although, when these types of faults are 'active' they manifest themselves as stuck-at faults, current test strategies cannot detect them due to their short duration and their random occurrence. In order to detect an intermittent or transient fault it is necessary to incorporate some sort of Concurrent Error Detection (CED) mechanism which will continuously monitor the operation of a circuit by comparing the actual response of the circuit to some predicted value enabling deviations from the norm to be identified. The ability to detect intermittent or transient faults in circuits is becoming increasingly important with the wide spread use of VLSI circuits in 'safety critical' applications in, for example, aerospace and automotive industries, petro-chemical and nuclear power plants etc., where an intermittent or transient fault could have dire consequences.

The ADOLT Scheme, outlined in this paper, permits the error detection capabilities of a CED scheme to be adapted to the error detection requirements of a given application so reducing the impact of the scheme on the design in terms of area overheads and the effect on circuit performance. The CED scheme uses a slightly modified version of Dong's Code [1] and gives a more efficient implementation of a CED scheme than previous work in this area [2,3].

2. Analysis of the Implementation of the ADOLT Scheme:

To assess the feasibility of implementing a CED scheme using the modified Dong's Code a 32-bit RISC Processor [4] was designed. The processor has 42 instructions in its set. The extra hardware required for the implementation of the scheme amounted to 42%. This overhead was not entirely due to the checkers, check symbol generators and check symbol prediction circuitry required by the CED scheme alone but also from the use of an entirely separate check symbol register file from the data register file and check symbol carry generator separate from the carry generator in the ALU, in order to expose faults which would otherwise go undetected. For example if a combined data and check symbol register file is used, an address decoder fault would select the wrong location; however the data selected would be extracted with the corresponding check symbol which would be 'correct' for the data actually selected - a error which cannot be detected unless separate register files are used.

In implementing any Design for Testability scheme several factors must be considered, namely, area overheads, error coverage, effect on performance and ease of use.

2.1 Area Overhead:

In order to compare this scheme with previous work, the area overheads associated with implementing this scheme on an ALU, alone, was estimated.

Design A -- Berger Checked ALU [2]
Number of extra gates = 635.
Design B -- Bose-Lin Checked ALU [3]
Number of extra gates = 521.
Current Design.
Number of extra gates = 466.

The use of the modified Dong's Code gave, approximately, a 27% and 11% reduction in gate count over Designs A and B respectively.

2.2 Error Coverage:

The code detects all unidirectional errors except those, which affect only the information bits and have weight equal to (m+1) or its multiples [1]. In other words if m is set to 7, the number of information bits is 32, and the errors affect only the information bits, then the code can detect any unidirectional error of weight not equal to 8, 16, 24 and 32, but all other weights can be detected. To achieve this error coverage 5 checkbits are required To compare the effectiveness of the error detection capability of this code, with respect to the number of checkbits, consider the error coverage of the Bose-Lin Code [3], given by 2^{r-2}+ r-2; if r = 5, -- the maximum number of errors which could be detected would be 11 and no more; which is a poorer error coverage than that obtained from Dong's Code. The code can also detect some other types of errors. For example, if the checkbits are affected by any number of unidirectional errors then the code can detect all types of errors (unidirectional or bi-directional errors which affect the information bits) this comes from the fact that the check symbols of the code form a set of unordered words in which no check symbol can be changed into another by any unidirectional errors; this is an advantage over the Berger Code itself. The error detection capability of the code is summarised below.

Type of error affecting the Information Bits	Type of error affecting the Check Bits	Number of errors detected by the code
(Unidirectional) $1 \rightarrow 0$ **OR** $0 \rightarrow 1$	Error Free	Errors of weight ≠ (m+1) or its multiples
(Unidirectional) $1 \rightarrow 0$ **OR** $0 \rightarrow 1$	$1 \rightarrow 0$ **OR** $0 \rightarrow 1$	All Errors
(Bidirectional) $1 \rightarrow 0$ **AND** $0 \rightarrow 1$	$1 \rightarrow 0$ **OR** $0 \rightarrow 1$	All Errors
Error Free	$1 \rightarrow 0$ **OR** $0 \rightarrow 1$	All Errors

Table 1 -- Error detection capability

2.3 Effect on Performance:

Within the RISC processor the effect on performance amounts to the time taken to generate the check symbol of the result of an operation and comparing it with the predicted value. Checking the integrity of the input data to the ALU together with the prediction of the check symbol of the result of an operation is done whilst the ALU is processing the input data, consequently incurs no time penalty, this has been verified via simulation even for the shortest instruction execution.

2.4 Ease of Use:

An important aspect of implementing any information redundancy scheme is its universal applicability to all generic function types used in a design, that is arithmetic, logic and storage. Dong's code can be applied to all types of functions hence a unified coding scheme can be used throughout a design. Furthermore the same method of check symbol generation is used regardless of the number checkbits required, this is not the case, for example, in the Bose-Lin Code.

3. Conclusions:

The characteristics of the CED technique have been analysed and compared with previous work. It is concluded that:

a) The overheads in terms of gate count is an improvement when compared with other designs. Furthermore, since fewer gates are required for its implementation its effect on performance will be less.

b)The technique has a better error detection capability than previous methods discussed.

c) The technique is easy to use permitting a unified design methodology throughout a circuit since:

i) It can be applied to all generic function types.

ii) The method of check symbol prediction is independent of the number checkbits used.

Furthermore in this method of CED the number of checkbits used is independent of the number data bits being processed giving a degree of flexibility on the error coverage implemented depending upon the application.

The viability of this approach as a mechanism for introducing a CED capability into a VLSI circuit has been demonstrated through the design of a 32-Bit RISC Processor, the increased gate count amounted to 42%, which is much less than that resulting from duplication and is secure from common mode faults.

4. References:

[1] H . Dong , " Modified Berger Codes for the Detection of Unidirectional Errors " 12th Int. Symp. Fault-Tolerant Comp., June 1982, pp 317-320

[2] J. Lo, S. Thanawastien, T. R. Rao, and, M. Nicolaidis ", An SFS Berger Check Prediction ALU and its Application to Self-Checking Processor Designs", IEEE Trans on CAD, vol 11, no 4, April 1992, pp 525-540.

[3] S.S. Gorshe, B.Bose, "A Self Checking ALU Design with Efficient Codes", Proceedings 14th VLSI Test Symposium, 1996, pp 157-161.

[4] A. Maamar, G.Russell, "A 32-Bit RISC Processor with Concurrent Error Detection", Proceedings 24th Euromicro Conference, August 1998, pp461-477.