A Defect-Tolerant and Fully Testable PLA

N. Wehn  M. Glesner  K. Caesar  P. Mann  A. Roth

Institut fuer Halbleitertechnik
Fachgebiet Halbleiterschaltungstechnik
Technische Hochschule Darmstadt
D-6100 Darmstadt

Abstract

This paper presents a defect-tolerant and fully testable PLA allowing for the repair of a defective chip. The repair process is described. Special emphasis is devoted to the location of defects inside a PLA. The defect location mechanism is completely topological and circuit independent and therefore easy to adapt to existing PLA generators. Yield considerations for this type of PLAs are presented.

Introduction

The programmable logic array is an important building block in VLSI circuits. Particular so, since efficient methods of logic [1] as well as topological minimization [2] were developed. Although PLAs still require a larger chip area than an equivalent circuit in random logic, their regular structure and the availability of programs to automate the synthesis process results in a significant reduction of design time. Therefore this type of circuit becomes more and more attractive e.g. in the controller design of VLSI processors, such as Intel's 80386, Motorola's 68020 and AT&T's WE32000 [3].

Increased usage of PLAs has drawn attention to the problem of testing. Despite their regularity this has proved to be a difficult task [4] and therefore this problem has been an active research area. Several PLA design-for-testability methods have been proposed e.g. [5,6]. A common feature of these designs has been the modification and augmentation of the PLA structure in such a way that each product line can be contollored individually e.g. by the use of dedicated shiftregisters.

As the complexity of circuits, as well as the density of devices increases, so does the number of defective chips. This problem may be alleviated by adding redundant elements to a circuit, that can be used to replace faulty components. Redundancy has been successfully employed by manufacturers of memory chips to increase the yield of usable circuits. Obviously it would be highly desirable to apply this technique to PLAs as well. Approaches such as Berger codes have already been implemented to design totally self-checking PLAs, but as yet, practically no research has been reported in the area of redundancy for repairing PLAs. The only publications being those of Wey and Lombardi [7-9], but only little emphasis has been devoted to fault location in it. Fault location is absolutely necessary for reconfiguration. This task is usually not a trivial one.

The PLA presented in this paper uses dynamic redundancy meaning: the spare components are provided within the system, and are restructured and switched-in in the defect case.

The PLA Testing Strategy

The large number of proposed PLA testing schemes mainly differ in their fault coverage, required hardware overhead, time delay and capability of built-in self-test. Unfortunately, test sets for PLAs with no additional hardware cannot guarantee 100% fault coverage, whereas testing methods with high fault coverage generally require registers to control the activations of product and/or input lines. This entails an increase in chip area of 15-100% and–in some cases–an additional 1-2 gate delays during normal operation of the PLA.

A very interesting testing scheme has been presented by E.J. McCluskey [10] and improved by J. Khakbaz and M. Lighthart [11,12]. Specifically, it has the following advantageous properties:
developed the following strategy to solve
non connected to
product
above, therefore selected as the
much lower than the theoretical maximum
input lines and modifying the product
PLAs can be rendered testable by adding
input lines required is
Cubical notation is assumed; \( n_i \) and \( n_p \)
denote the number of inputs and product
lines respectively. In [11] it is shown,
that \( d_k(t_i, o_j) = 1 \) is acceptable, provided
product \( p_i \) is connected to an output line
not connected to \( p_j \). Initially untestable
PLAs can be rendered testable by adding
input lines and modifying the product
terms. An upper bound on the number of
additional input lines required is
\( \log_2 n_p \). Generally the actual overhead is
much lower than the theoretical maximum
that is based on the worst-case assumption that
all product lines are identical. It can be shown that the generation of additional
input lines with the corresponding testvector set in a given PLA under the
condition of hammingdistance 2 is equivalent to the graph K-colorability problem
and therefore \( np \)-complete [14]. We have developed the following strategy to solve this problem:

1. calculate the number of additional inputs by an incremental optimization
   algorithm (=Expected_inputs)
2. Expected_inputs:=Expected_inputs-1
3. try to find a testvector set by simulated
   annealing for this number of inputs
4. is there a dhk2 then goto 5 else goto 2
5. final_result := Expected_inputs+1

Step 1 of the algorithm is solved by
setting the temperature equal to 0 in the
statistical cooling algorithm. In step 3
the cooling control parameter is decreased
by multiplying it each step by 0.9. We
leave the inner loop in the annealing
algorithm, if the costs are lower than the
best cost value found in the previous
temperature step. This criterion has been found experimentally to be a good equilibriump condition.

The test procedure consists of
applying all input vectors \( t_{1,o} \) (main test patterns) as well as all \( t_{1,j} \) (auxiliary
test patterns), obtained by flipping bit \( j \)
of main test pattern \( t_{1,o} \), and observing
the resulting outputs. If no error occurs
the PLA is fault free.

## Locating Defects in a PLA and
Provided Redundancy

Once a PLA has been identified as defective, it is necessary to determine the
exact nature of the defect and whether or not it may be repaired by utilizing the
available redundancy [7]. In order to do this, it is necessary to consider in more
detail the physical defects and their corresponding fault models. This has been
intensively studied in the literature. It has been proven that suitable fault models
for PLAs are: crosspoint faults, stuck-at faults and bridging faults. Crosspoint
faults cause growth or shrink effects in the AND-plane, appearance or disappearance
effects in the OR-plane. The following table based on the considerations in [7]
summarizes the relationship between the plane of faults and redundancy that can be used to repair the defect:

<table>
<thead>
<tr>
<th>Redundancy</th>
<th>Faults (Plane)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redundant productterms</td>
<td>growth, shrink,</td>
</tr>
<tr>
<td></td>
<td>disappearance,</td>
</tr>
<tr>
<td></td>
<td>bridge(AND),</td>
</tr>
<tr>
<td></td>
<td>stuck-at(AND),</td>
</tr>
<tr>
<td></td>
<td>bridge(OR),</td>
</tr>
<tr>
<td></td>
<td>stuck-at(OR),</td>
</tr>
<tr>
<td></td>
<td>stuck-at(inputs)</td>
</tr>
</tbody>
</table>

When taking into account the external interconnection overhead needed for redundant
inputs/outputs, it is obvious that redundant product terms are a good compromise between interconnection overhead and fault occurrence.

Defect location essentially consists of generating a set of faults capable of accounting for all observed output vectors, both erroneous and correct. If only a single such set exists the fault locations can be positively identified. If there are several such sets there is no way of deciding which corresponds to the actual defect configuration. The method of[10] does not take fault location into account and a number of problems arise in consequence, as will be shown below. Alternative testing schemes that facilitate fault location were rejected, due either to insufficient fault coverage [15]
or excessive hardware overhead [16].

In addition to the existence of main test patterns for all product lines further conditions are stipulated:

1. only redundant product lines are available
2. multiple consecutive test and reconfiguration runs are not allowed. A defect PLA is repairable only if all defect product lines can be identified and replaced at the same time.
3. only the main and auxiliary test patterns of a PLA may be used for locating faults.

The first constraint is due to layout considerations, the second and third are purely economically motivated, since test and reconfiguration of a PLA is a time-consuming process. Generating additional test vectors in order to verify specific assumptions on fault locations would greatly increase the complexity of the test procedure. These restrictions do not preclude a second run to check whether any errors occurred during the reconfiguration phase.

Crosspoints and stuck-at faults are considered in attempting to identify faults in a defect PLA. All of these faults are detectable, they cause at least one main or auxiliary test pattern to produce an erroneous output vector.

a) missing device fault at position \(pi,k\) of the AND-plane: The auxiliary test pattern \(ti,k\) is incapable of deactivating product \(pi\). A 1 instead of the expected 0 is observed on all outputs connected to this product.

b) missing device at position \(pi,f\) of the OR-plane: Output \(f\) remains inactive during application of the test pattern \(ti,i\) at all \(ti,0\), however ought to select \(f\).

c) extra device at a don't care position \(pi,s\), where \(ti,0,s=pi,s\): the product and all corresponding output lines are erroneously deactivated by auxiliary test pattern \(ti,s\). Extra device defects at care positions are treated as product stuck-at-0 faults.

d) extra device at a don't care position \(pi,s\), where \(ti,0,s<>pi,s\): product \(pi\) is activated solely by \(ti,s\). At least the main test pattern \(ti,s\) therefore causes a faulty output vector in failing to select the output lines connected to \(pi\).

e) extra device fault at position \(pi,f\) of the OR-plane: output line \(f\) is activated in addition to all outputs normally selected by \(ti,s\).

f) output line \(f\) stuck-at 0: such an output is permanently set to 0; in particular, none of the main test patterns of products connected to \(f\) are capable of activating it.

g) output line \(f\) stuck-at 1: output \(f\) is permanently set to 1. For every product \(pi\) connected to \(f\), however, an auxiliary test pattern \(ti,s\) deactivating it. The stuck-at 1 error results in a 1 instead of the expected 0 on application of \(ti,s\).

h) product line \(pi\) stuck-at 0: main test pattern \(ti,s\) is incapable of activating the product line. None of the corresponding outputs is activated.
- (b,h): only a single output line may be connected to the faulty product in case b)
- (f,h): both of the above conditions must hold: the faulty product in case h) must be connected to a single output line which, in turn, is not connected to any other product.
- (k, ...): a faulty input connected to a single product results in a product stuck-at 1 error of type h). In consequence, all equivalences stated for h) also apply to k).

Obviously terminal and nonterminal defects cannot always be distinguished. A repair may be attempted, allocating spare product lines in accordance with the optimistic assumption of a repairable defect; however, this assumption must be verified by a second run. The equivalency \([b,h]\) on the other hand may safely be ignored, since both error types may be eliminated by replacing the appropriate product line.

Multiple Fault Assumption

If multiple faults are permitted, defect equivalence is subject to less stringent restrictions. An additional equivalence of fault types \([a,1]\) becomes feasible. Apart from equivalence defect masking must be considered. Error masking occurs when several products are activated at the same time. Although this does not impair error detection, it may severely hamper defect location.

EXAMPLE:

\[
P = \begin{pmatrix} 0 & 1 & 1 & 1 \\ X & 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad T = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}
\]

The personality matrix \(P\) represents a testable PLA with main test pattern \(T\). Missing device errors at position \(p_{1,2}, p_{2,3}\) would cause product \(p_1\) to cover \(p_1\). In other words, \(p_1\) is active at any time. This results in faulty output vectors on application of test pattern \(t_{2,2}, t_{3,3}, t_{1,1}, t_{1,3}\). However, it is impossible to decide whether \(p_1, p_2\) or, both of them are defective.

In addition to products simply being covered by others, as was the case in the previous example, a combination of missing and extra device errors may actually cause a faulty product \(p_i\) to dubicate some other product \(p_j\). An error on line \(p_i\) (e.g. product stuck-at 0) may then go unnoticed, as the correct output vector is generated by \(p_i\).

Therefore products in a reparable PLA may be: fault-free, faulty or indeterminate. If products of indeterminate status are not replaced, a second test run is required in order to verify whether they are indeed fault-free. Any residual defects will then cause the circuit to be rejected. Alternatively, all faulty as well as indeterminate products might be replaced during reconfiguration, resulting in a maximum yield of fault-free circuits. The amount of redundancy required for this approach may, however, prove to be prohibitive.

The following flowchart reflects the repair procedure for reconfigurable PLAs. Note that up to now no topological and circuit dependency were required.

**Acceleration of the Testing Sequence**

The presented scheme involves some drawbacks: Storage of a huge amount of test patterns. In our case we need \((n+1)\cdot p_p\) testpatterns whereas \(n_i\) is the number of inputs and \(p_p\) the number of product terms. This involves a slow testing because of the shifting of patterns through a scan path. These drawbacks can be alleviated when taking into account that the auxiliary test patterns (ATP) can be generated from the corresponding main test pattern (MTP) by a simple hardware. This additional hardware consists mainly of two shift-registers and a multiplexer and generates for a given MTP the \(n_i\) ATPs. Since we need only the information if the actual tested product term is defect we compress the outputs of the PLA with a multiple-input signature register. This simple mechanism reduces the amount of test patterns to be stored outside and shifted-in to \(p_p\) patterns.

**Layout Considerations**

As already discussed, only redundant product terms are provided within the PLAs. This method of choice implies two problems: first how to deactivate a faulty productterm, second how to realize a programmable product term. To solve the first problem, a laser fuse is located within the communication circuit between the planes as shown in figure 1. This fuse is switched off in a faulty product term ensuring that all transistors in the OR-plane are deactivated by the clocked p-transistor. The design of the redundant
product terms is based on floating gates that permits multiple product term programming (figure 2). Input and output lines were designed on two layers that are multiple connected. This technique ensures fault-tolerance with regard to open defects, on the other hand decreased speed due to the enhanced capacitance must be accepted. Figure 3 shows the photomicrograph of such a PLA.

Yield Analysis

The productivity of chips strongly depends as well on faults caused by random defects in the materials and photolitography as on the percentage of uncorrectable areas on the chip. In our case, only defects inside the two planes can be for the most part repaired, defects in power lines, pre-charge and discharge transistors, schmitt-triggers and drivers are considered to be fatal. If we denote $A_{fatal}$ the area in which a defect is fatal and $A_{repairable}$ the area in which a defect can be repaired, we obtain the following yield model that uses negative binomial statistics [17] with $D$ as average defect density:

$$Y = \left(1 + sA_{fatal}/D\right)^{-1} \times \left(1 + sA_{repairable}/D\right)^{1/1-D}$$

This yield model was applied to a PLA, that has 30 inputs, 30 outputs and 322 product terms. This PLA is part of a reconfigurable microprocessor that was fabricated on an advanced double metal layer CMOS process with an average defect density of 2.4/cm$^2$. The defect cluster was assumed to be 1. The following table summarizes the obtained results:

<table>
<thead>
<tr>
<th>#redundant terms</th>
<th>area $\mu m^2$</th>
<th>yield</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2776108</td>
<td>0.81</td>
</tr>
<tr>
<td>1</td>
<td>2770435</td>
<td>0.93</td>
</tr>
<tr>
<td>2</td>
<td>2784761</td>
<td>0.94</td>
</tr>
<tr>
<td>3</td>
<td>2799107</td>
<td>0.95</td>
</tr>
</tbody>
</table>

This table demonstrates that an obviously yield increase takes place with one redundant minterm.

Conclusion

The design of a defect-tolerant and fully testable PLA has been presented in this paper. Special emphasis was devoted to the location and reconfiguration of defects inside a PLA. The recent presented new storage technique that is based on the ferroelectric effects [18] will offer softswitch reconfiguration and challenge new aspects in the use of PLAs with programmable product terms.

References


Paper 3.4
figure 1: Communication circuit and photomicrograph

figure 2: Redundant product term and photomicrograph

figure 3: Photomicrograph of a reconfigurable PLA