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. Particulary 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 controler 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 controled 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 publication 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 selftest. 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.M.Ligthart [11,12]. Specifically, it has the following advantageous properties:

Supported in part by a grant from the European Community within ESPRIT project 824

- coverage of all multiple stuck-at, crosspoints and bridging faults
- very low hardware overhead, consisting of few additional input lines and some additional product terms
- no time delay occurs during normal operation
- folded PLAs are permitted
- the basic PLA structure is preserved

This scheme was considered superior to others with regard to the criteria mentioned above, therefore selected as the method of choice.

The basic idea in this scheme is the ability to activate an individual product term by adding some additional inputs to the AND-plane of the PLA instead of using e.g. shiftregisters. This method is very similar to the decoder concept by Sato [13], but the number of extra inputs is significantly lower than in the decoder approach.

A PLA is testable according to E.J.McCluskey if for each product p_i there exists an input vector $t_{i,0}$ such that:

$$d_H(t_{i,0},p_j) \geq 2\,;\ i,j \in [1,n_p] \wedge i \neq j \qquad x_{1,i} \oplus x_{2,i} = \begin{cases} x_{1,i}\overline{x_{2,i}} + \overline{x_{1,i}}x_{2,i} \\ \quad \text{for } x_{1,i}, x_{2,i} \in \{0,1\} \\ 0 \quad \text{for } x_{1,i} = X \vee x_{2,i} = X \end{cases}$$

$$d_H(x_1,x_2) = \sum_{i=1}^{n_I} x_{1,i} \oplus x_{2,i}$$

Cubical notation is assumed; n_I and n_p denote the number of inputs and product lines respectively. In [11] it is shown, that $d_H(t_{i,0},p_j)=1$ 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 $\lceil log_2 n_p \rceil$. 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 testvectorset 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 testvectorset by simulated annealing for this number of inputs*
4. *is there a dh<2 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 equilibrium condition.

The test procedure consits of applying all input vectors $t_{i,0}$ (main test patterns) as well as all $t_{i,j}$ (auxiliary test patterns), obtained by flipping bit j of main test pattern $t_{i,0}$, 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 wether 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 relation between plane faults and redundancy that can be used to repair the defect:

redundancy	faults (plane)
redundant productterms	growth, shrink, disappearance, appearance bridge(AND), stuck-at(AND)
redundant outputs	bridge(OR), stuck-at(OR)
redundant inputs	stuck-at(inputs)

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

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 existance 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 reparable 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 wether any errors occured during the reconfiguration phase.

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

a) *missing device fault at position $p_{i,k}$ of the AND-plane:* The auxiliary test pattern $t_{i,k}$ is incapable of deactivating product p_i. A 1 instead of the expected 0 is observed on all outputs connected to this product.
b) *missing device at position $p_{i,f}$ of the OR-plane:* Output *f* remains inactive during application of the test patterns $t_{i,0}..t_{i,ni}$. At least $t_{i,0}$, however ought to select *f*.
c) *extra device at a don't care position $p_{i,k}$, where $t_{i,0,k}=p_{i,k}$:* the product and all corresponding output lines are erroneously deactivated by auxiliary test pattern $t_{i,k}$. Extra device defects at care positions are treated as product stuck-at-0 faults.
d) *extra device at a don't care position $p_{i,k}$, where $t_{i,0,k}<>p_{i,k}$:* product p_i is activated solely by $t_{i,k}$. At least the main test pattern $t_{i,0}$ therefore causes a faulty output vector in failing to select the output lines connected to p_i.
e) *extra device fault at position $p_{i,f}$ of the OR-plane:* output line *f* is activated in addition to all outputs normally selected by $t_{i,0}$.
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 p_i connected to f, however, an auxiliary test pattern $t_{i,k}$ deactivating it. The stuck-at 1 error results in a 1 instead of the expected 0 on application of $t_{i,k}$.
h) *product line p_i stuck-at 0:* main test pattern $t_{i,0}$ is incapable of activating the product line. None of the corresponding outputs is activated.
i) *product stuck-at 1:* in a fault-free and testable PLA each product p_i is deactivated by at least one auxiliary test pattern $t_{i,k}$. No such $t_{i,k}$ exists for a product stuck-at 1.
j) *input line k stuck-at 0:* a product p_i connected to input k is not deactivated by $t_{i,k}$, resulting in an erroneous output vector.
k) *input line k stuck-at 1:* any product connected to input k is 'in effect' stuck-at 0.

Defects f,g,j,k involving input and output lines are irreparable, since only redundant products are available. A circuit must be rejected, if any such error occurs. Unfortunately, test results are sometimes ambiguous in this respect. Four definitions are introduced to discuss this fact in a more general way.

Definition 1:A defect is called *terminal* if it cannot be repaired i.e. f,g,j,k
Definition 2: A defect is called *nonterminal* if it can be repaired i.e. a,b,c,d,e,h,i
Definition 3: A defect is called *equivalent* if it is not possible to decide if a defect PLA can be repaired, but its possible to specify the product term that must be reconfigured in the reconfigurable case.
Definition4: A defect is called *masked* if it is not possible to decide if the masked product term is defect free.

Single Fault Assumption

Assume that the fault set of a defective PLA consists of only single errors. In this case some *terminal* and *nonterminal* defects may be *equivalent* regarding their input/output behaviour, provided additional conditions are satisfied, as listed below:

-{a,j}: in the case of error type j), only a single product line may be connected to the faulty input, if equivalence to error type a) is to be preserved.
-{g,i}: due to the single error restriction the faulty product in case g) may be connected to only a single output line.
-{b,f}: the missing device $p_{i,f}$ in case b) must be the only device present on output *f* . Otherwise *f* will be activated by some other product p_j, violating the stipulated equivalency.

-{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 equivalencies 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,i} becomes feasible. Apart from *equivalency* 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:
$$P = \begin{pmatrix} 0 & 1 & X & 1 & 1 \\ X & 0 & 0 & 1 & 1 \end{pmatrix} \quad T = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

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

In additon 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 dublicate some other product p_j. An error on line p_j (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 wether 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.

```
repair procedure for reconfigurable PLAs

while pattern_set <> {} do
 begin
  select test pattern and apply it to the PLA;
  if not(output correct) then
    case TYPE(defect) of
      nonterminal: perform_reconfiguration;
      terminal   : FATAL DEFECT;
      equivalent : perform_reconfiguration;
                   run_verify(repairable);
                   if not repairable then FATAL DEFECT;
      masked     : for all masked product terms do
                    begin
                     perform_reconfiguration;
                     run_verify(repairable);
                     if not repairable then FATAL DEFECT;
                    end;
    end;
 end;
```

Acceleration of the Testing Sequence

The presented scheme involves some drawbacks: Storage of a huge amount of test patterns. In our case we need $(n_i+1)*p_p$ 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 shiftregisters 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 productterms are provided within the PLAs. This method of choice implies two problems: first how to deactivate a faulty producterm, 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, precharge and discharge transistors, schmitt-triggers and drivers are considered to be fatal. If we denote A_{con} the area in which a defect is fatal and A_{rek} 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:

$$Y = (1 + \mu * A_{con} * D/\beta)^{-\beta} * \sum_{i=0}^{n} \frac{\Gamma(\beta + i) \qquad (\mu * A_{rek} * D/\beta)^{i}}{i! \; \Gamma(\beta) \quad (1 + \mu * A_{rek} * D/\beta)^{\beta + i}}$$

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/cm². The defect cluster was assumed to be 1. The following table summarizes the obtained results:

<u>overhead for defect location:</u>

3 additional inputs
7 additional product terms

#redundant terms	area µm²	yield
0	2756108	0.81
1	2770435	0.93
2	2784761	0.94
3	2799107	0.95

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

[1] R.K.Brayton,G.D.Hachtel,G.D.McMullen,
A.L.Sangiovanni-Vincentelli
"Logic Minimization Algorithms for
VLSI Synthesis"
Kluwer Academic Publishers,
Hingham, MA, 1984

[2] G.D.Hachtel
"Techniques for Programmable Logic
Array Folding"
Proc. of the 19th Design Automation
Conference,Las Vegas 1982,pp.147-155

[3] H.F.S.Law, S.Gai
"PLA Design for Bellmac-32a
Microprocessor"
Proc. of Int.Conf. on Circuits
and Computers, 1982, pp.161-164

[4] F.Somenzi, S.Gai
"Fault Detection in Programmable
Arrays"
Proc. of the IEEE, Vol.74,No.5,
May 1986,pp.655-668

[5] R.Treuer, H.Fujiwara, V.K.Agrawal
"Implementing a Built-In Self-Test
PLA Design"
IEEE Design&Test, April.1985,
pp.37-48

[6] J.Rajski, J.Tyszer
"Easily testable PLA Design"
Proc. of the Euromicro 1984,
pp.139-146

[7] C.L.Wey, M.K.Vai, F.Lombardi
"On the design of a Redundant
Programmable Logic Arry"
IEEE Journal of Solid-State Circuits,
Vol.SC-22,No.1,Feb.1987,pp.114-117

[8] C.L.Wey, F.Lombardi
"Analysis and Design of Repairable
PLAs"
Proc. of the Eurocomp, May 1987,
pp.363-366

[9] C.L.Wey
"On Yield Consideration for the Design
of RPLAs"
Proc. of the 24th Design Automation
Conference, Miami 1987,pp.622-628

[10] S.Bozorgui-Nesbat, E.J.McCluskey
"Lower Overhead Design for testability
of PLAs"
Proc. of the 1984 IEEE Int. Test
Conference, Oct.1984,pp.856-865

[11] S.Bozorgui-Nesbat, J.Khakbaz
"Minimizing extra Hardware for Fully
Testable PLA Design"
Digest of the ICCAD-85, Nov.1985,
pp.102-104

[12] E.H.L.Aarts, F.P.M.Beenker,
M.M.Ligthart
"Design-for-Testability of PLAs Using
Statistical Cooling"
Proc. of the 23th Design Automation
Conf., Las Vegas 1986,pp.339-345

[13] T.Sato, Y.Tohma
"A New Configuration of PLA with
Function Independent Test"
Technical Report, Tokyo Institute of
Technology, Tokyo, October 1982

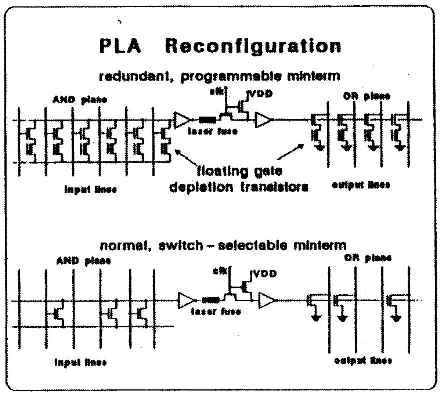
[14] "Defect Location and Reconfiguration
in Programmable Logic Arrays"
Technical Report HST/D42A
Technical University of Darmstadt, June 1987

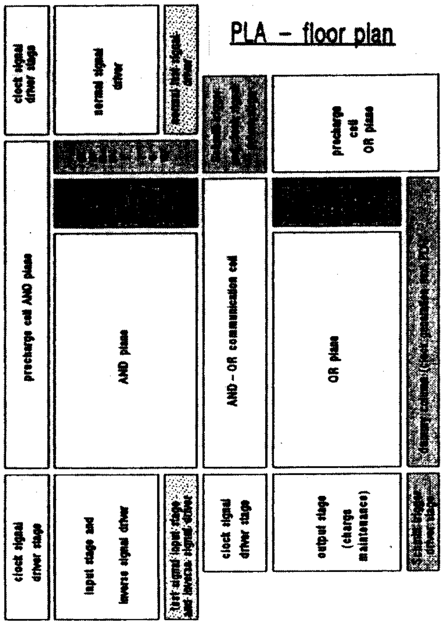
[15] H.Fujiwara, K.Kinoshita
"A Design of PLAs with Universal
Tests"
IEEE Trans.Computers,Vol.C-30,No.11,
Nov.1981,pp.823-828

[16] K.K.Saluja, K.Kinoshita, H.Fujiwara
"An Easily testable Design of PLAs for
Multiple Faults"
IEEE Trans.Computers,Vol.C-32,No.11,
Nov.1983,pp.1038-1046

[17] C.H.Stapper, F.Armstrong, K.Saji
" Integrated Circuit Yield Statistics"
Proc. IEEE 71,April 1983,pp.453-470

[18] "A New Memory Technology Is About
To Hit The Market"
Electronics/February 18, pp.91-95




PLA - floor plan
clock signal driver stage
normal signal driver
precharge cell OR plane
precharge cell AND plane
AND plane
AND - OR communication cell
OR plane
clock signal driver stage
input stage and inverse signal driver
test signal input stage and inverse signal driver
clock signal driver stage
output stage (charge maintenance)
Schmitt trigger driver stage

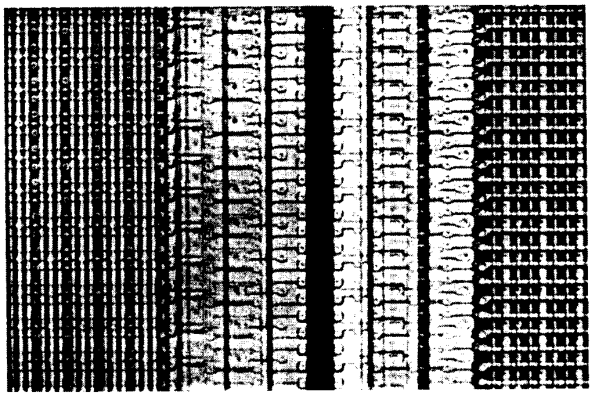


figure 1: Communication circuit and photomicrograph

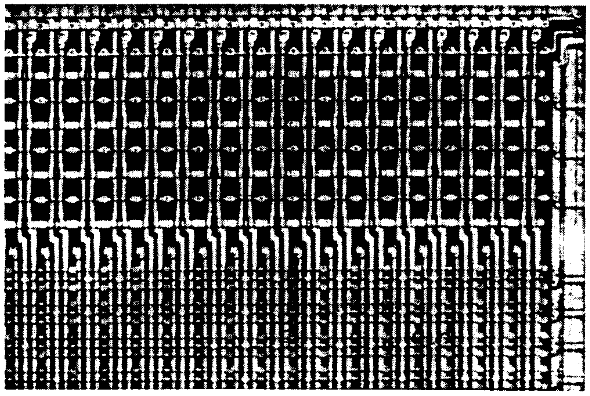


figure 2: Redundant product term and photomicrograph

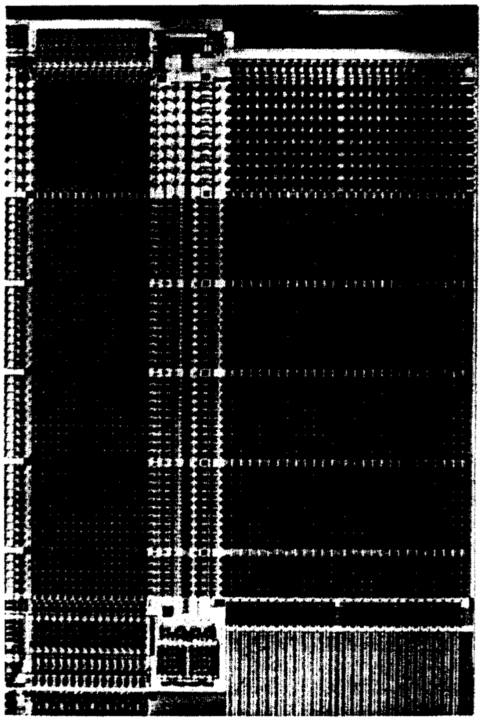


figure 3: Photomicrograph of a reconfigurable PLA