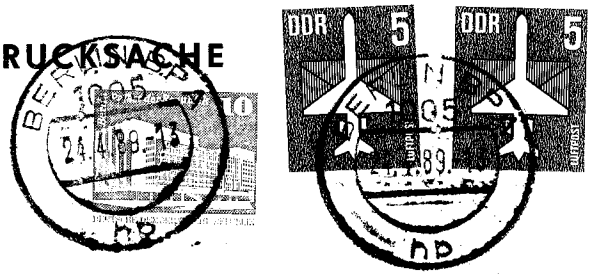
Dr. rer. nat.
R. Karl-Adolf Zech
Diplom-Mathematiker
Schliemannstraße 28
Berlin
DDR – 1058
German Democratic Republik

DRUCKSACHE



Dr. B.F. Cockburn
University of Waterloo
Dept. Computer Science
WATERLOO
Ontario N2L 3G1
Canada

Dr. rer. nat.
R. Karl-Adolf Zech
Diplom-Mathematiker

Schliemannstraße 28
Berlin
DDR – 1058 German Democratic Republic

dear colleague: free

I should be very obliged to you for sending me a copy of your paper entitled:

CS-88-84.48 sent June 21/89

Switch-level testability of the dynamic CMOS PLA.

please.

Thanking you in advance Yours sincerely,

Karl Zech

UNIVERSITY OF SOUTHERN CALIFORNIA

INFORMATION SCIENCES INSTITUTE

4676 Admiralty Way Suite 1001
Marina del Rey, California 90292-6695
213/822-1511

April 14, 1989

Department of Computer Science
University of Waterloo
Waterloo, Ontario
N2L 3G1
ATTN: Research Report Secretary

Dear Colleague,

I would like to obtain a copy of the following technical report(s):

CS-88-48
SWITCH-LEVEL TESTABILITY OF THE DYNAMICALLY CMOS PLA
by B.F. Cockburn and J.A. Brzozowski

Please send the technical report(s) to the following address:

USC Information Sciences Institute Library
4676 Admiralty Way
Marina del Rey, CA 90292-6695
ATTN: Linda Louie

sent April 21/89

Enclosed is a money order in the amount of $2.00 to cover the charges for the report.

Thank-you,

Linda Louie

Linda Louie
Library Assistant

Printing Requisition / Graphic Services

26005

1. **Please complete unshaded areas** on form as applicable.
2. **Distribute copies** as follows: White and Yellow to Graphic Services. **Retain Pink Copies for your records.**
3. On completion of order the **Yellow copy will be returned** with the printed material.
4. **Please direct enquiries, quoting requisition number and account number,** to **extension 3451.**

TITLE OR DESCRIPTION

Switch-Level Testability of the Dynamic CMOS PLA CS-88-48

DATE REQUISITIONED	DATE REQUIRED	ACCOUNT NO.
Dec. 20/88	ASAP	1 2 6 6 3 1 8 4 1

REQUISITIONER – PRINT	PHONE	SIGNING AUTHORITY
J.A. Brzozowski	4441	

MAILING INFO – NAME	DEPT.	BLDG. & ROOM NO.	
Sue DeAngelis	C.S.	DC 2314	[X] DELIVER [ ] PICK-UP

Copyright: I hereby agree to assume all responsibility and liability for any infringement of copyrights and/or patent rights which may arise from the processing of, and reproduction of, any of the materials herein requested. I further agree to indemnify and hold blameless the University of Waterloo from any liability which may arise from said processing or reproducing. I also acknowledge that materials processed as a result of this requisition are for educational use only.

NUMBER OF PAGES: 45

NUMBER OF COPIES: 50

TYPE OF PAPER STOCK: [ ] BOND [ ] NCR ____ PT. [ ] COVER [ ] BRISTOL [ ] SUPPLIED [ ] ~~140M~~ Alpac Ivory

PAPER SIZE: [ ] 8½ x 11 [ ] 8½ x 14 [ ] 11 x 17 [ ] ~~10 pt Rolland Tint~~ 10x14 Glosscoat

PAPER COLOUR: [ ] WHITE [X] ____ INK: [X] BLACK [ ] ____

PRINTING: [ ] 1 SIDE ____ PGS. [X] 2 SIDES ____ PGS. NUMBERING: FROM ____ TO ____

BINDING/FINISHING: [ ] COLLATING [ ] STAPLING [ ] ____ HOLE PUNCHED [ ] PLASTIC RING

FOLDING/PADDING: 7x10 saddle stitched CUTTING SIZE

Special Instructions

Beaver Cover

Both cover and inside in black ink please

COPY CENTRE — OPER. NO. | BLDG. | MACH. NO.

DESIGN & PASTE-UP — OPER. NO. | TIME | LABOUR CODE

D 0 1

D 0 1

D 0 1

TYPESETTING — QUANTITY | OPER. NO. | TIME | LABOUR CODE

PAP 0 0 0 0 0 0 — T 0 1

PAP 0 0 0 0 0 0 — T 0 1

PAP 0 0 0 0 0 0 — T 0 1

PROOF

PRF

PRF

PRF

NEGATIVES — QUANTITY | OPER. NO. | TIME | LABOUR CODE

FLM — C 0 1

FLM — C 0 1

FLM — C 0 1

FLM — C 0 1

FLM — C 0 1

PMT

PMT — C 0 1

PMT — C 0 1

PMT — C 0 1

PLATES

PLT — P 0 1

PLT — P 0 1

PLT — P 0 1

STOCK

0 0 1

0 0 1

0 0 1

0 0 1

BINDERY

RNG — B 0 1

RNG — B 0 1

RNG — B 0 1

MIS 0 0 0 0 0 0 — B 0 1

OUTSIDE SERVICES

$ ____ COST

TAXES – PROVINCIAL [ ] FEDERAL [ ]

GRAPHIC SERV. OCT. 85 482-2

To John

From Sue

Date Dec. 16

memo

University of Waterloo

Would you please sign the attached print req. and indicate how many copies you would like done.

Thanks.

Sue

SO please

To

From

Date

memo

University of Waterloo

Grie John B

20 more

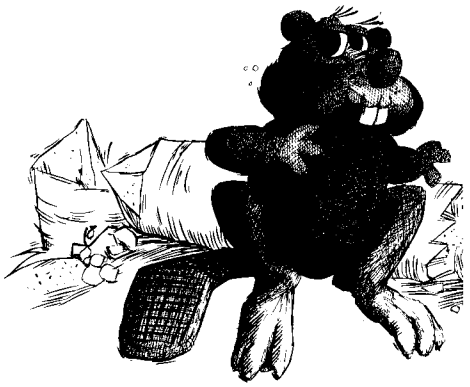
copies of

88-48

Feb. 3/89

UNIVERSITY OF WATERLOO
COMPUTER SCIENCE DEPARTMENT

UNIVERSITY OF WATERLOO
COMPUTER SCIENCE DEPARTMENT



Switch-Level Testability of the Dynamic CMOS PLA

B.F. Cockburn
J.A. Brzozowski

Research Report
CS-88-48

December, 1988

Switch-Level Testability of the Dynamic CMOS PLA†

B.F. Cockburn and J.A. Brzozowski

Department of Computer Science
University of Waterloo
Waterloo, Ontario
Canada, N2L 3G1

ABSTRACT

Functional testing, as opposed to parametric testing, plays an important role in testing VLSI integrated circuits. However, it appears that designs are not always carefully analysed a priori to determine precisely which faults are *clean*, i.e. testable by logic means alone. The programmable logic array (PLA) is a popular circuit form used to implement a system of Boolean functions over a set of input variables. This report considers the testability of the dynamic CMOS PLA with respect to an extended set of switch-level faults models, namely: node faults, transistor stuck-opens and stuck-ons, interconnect breaks, ohmic shorts, and crosspoint faults. Single occurrences of each fault model are classified as either clean, unclean, or clean subject to conditions on the logical products and output functions computed by the PLA. Finally, a modified dynamic CMOS PLA design is described and its increased switch-level testability properties are stated.

1. Introduction

The programmable logic array (PLA) is a regular circuit structure that implements a system of Boolean functions over a set of input variables. A PLA is essentially a direct hardware implementation of the two-level sum-of-products (SOP) forms of the functions. In dynamic CMOS, as in other integrated circuit technologies, better performance is achieved

† This work was supported by the Natural Sciences and Engineering Research Council of Canada, Grant A0871. This paper was presented at the Third Technical Workshop - New Directions for IC Testing, Halifax, Nova Scotia, October 26-27, 1988.

by implementing the closely related NOR-NOR form instead of the obvious AND-OR form.

A PLA has a roughly rectangular layout. Its physical size in one direction is determined by the number of parallel input and output lines, and in the orthogonal direction by the number of product lines. A PLA's logical behaviour is determined by the irregular placement of crosspoint devices in the AND and OR planes. Structural regularity tends to lead to compact layouts and fewer design errors, making the PLA attractive in VLSI applications [MeCo80]. Computer-aided design (CAD) tools are available which automatically compile a correct PLA layout given the desired output functions [LeGaEl84]. The regularity of the PLA facilitates fault modelling, testability analysis, design modification for increased testability, and test pattern generation, as witnessed by the large volume of published work (See [SoGa86] for a good survey).

In this paper, the testability of the dynamic CMOS NOR-NOR PLA described in [WeEs85] (which we will call the standard dynamic CMOS PLA) is examined with respect to an expanded set of fault models. The fault models, described later, cover the spectrum from switch-level models that closely reflect the topology of the physical layout, to higher-level models that are more closely related to the SOP structure of the output functions. The various PLA fault models have not always been clearly specified in previous work. Furthermore, such natural switch-level fault models as the transistor stuck-on and the interconnect break have not been treated adequately. Typically, testability analyses for PLAs have used a variety of fault models at different levels of abstraction, without considering a consistent and complete set of switch-level fault models.

Definitions and conventions used in this paper are given in Section 2. The internal structure and timing of the dynamic CMOS NOR-NOR PLA are reviewed briefly in Section 3. An expanded set of fault models appropriate for CMOS PLAs is presented in Section 4. Possible redundancies that could arise in a PLA are then discussed in Section 5. The testability properties of the dynamic CMOS NOR-NOR PLA with respect to the fault models are established in Section 6. A modified dynamic CMOS PLA is introduced, and its enhanced testability properties described in Section 7. Section 8 concludes the paper with a brief discussion of implementation issues and costs associated with the modified design.

2. Definitions and Conventions

The positive and negative power supply lines will be denoted by V_{DD} and *GND*, respectively. A "0"("1") represents a valid low(high) voltage signal corresponding to logic 0(1). A *path* is a chain of lines connected together via the channels of MOS transistors that are in their conductive states. In a correctly designed and fault-free CMOS circuit, the signal on a line will either be a *driven* signal produced by a path to one and only one of V_{DD} or *GND*, or will be a *stored* signal produced by an isolated charge recently (e.g. within the last 10 ms [Ha87]) established by a driven signal. A *fight* [ReMe81] occurs when a sustained path exists from V_{DD} to *GND*. Fault-free circuits will be assumed free of fights for all input combinations. A "?" represents some possibly invalid intermediate voltage resulting from the given line lying on some path from V_{DD} to *GND*. Circuits that receive a ? as input may function incorrectly, or may interpret the signal unpredictably as either a valid 0 or 1. An "×" represents a signal whose value is irrelevant.

Stored signals will be written in bold font (e.g. **0**, **1**, and **?**). A *weak* 0(1), denoted by $0^w(1^w)$, is a 0(1) signal that has been degraded slightly as a result of transmission through the channel of at least one p(n)-type MOS transistor (See [WeEs85] for technical details). Individual signals are denoted by lower-case letters; vectors of signals are denoted by upper-case letters.

The vector "0...0"("1...1") represents the vector of appropriate length that contains only 0s(1s). $X_{j0}(X_{j1})$ is a Boolean vector which has 0(1) as its j-th component. When used together, X_{j0} and X_{j1} are assumed to differ only in their j-th component.

Input vector $V_0(V_1)$ is a *0(1)-vertex* of a Boolean function $f(X)$ iff $f(V_0)=0$ ($f(V_1)=1$). To avoid degenerate cases, we assume that every Boolean function of interest is *non-trivial*, i.e. it has at least one 0-vertex and at least one 1-vertex. A function f *covers* a vector X iff $f(X)=1$. A function f *covers* a function g iff f covers all of the 1-vertices of g, i.e. iff $g(X)=1$ implies $f(X)=1$, for all Boolean vectors X.

If x represents a signal in a fault-free circuit, then x^* represents the corresponding signal in a circuit affected by the fault under consideration. A fault is *clean* if there exists a *test*, i.e. a sequence of input vectors (e.g. $X_1, X_2, ..., X_n$) such that on at least one output, say f_k, the fault-free and

faulty instances of the circuit carry complementary logic signals for the final vector (i.e. $f_k(X_n) \neq f_k^*(X_n)$) [Brzo87]. The last vector of a test must not produce a ? on the designated output f_k. A fault that is not clean is *unclean*. If the uncertainty caused by the presence of a ? within a faulty circuit can potentially affect the signal that is observed on f_k, then the fault is unclean. These concepts will be discussed further in Section 6.

3. The Dynamic CMOS NOR-NOR PLA

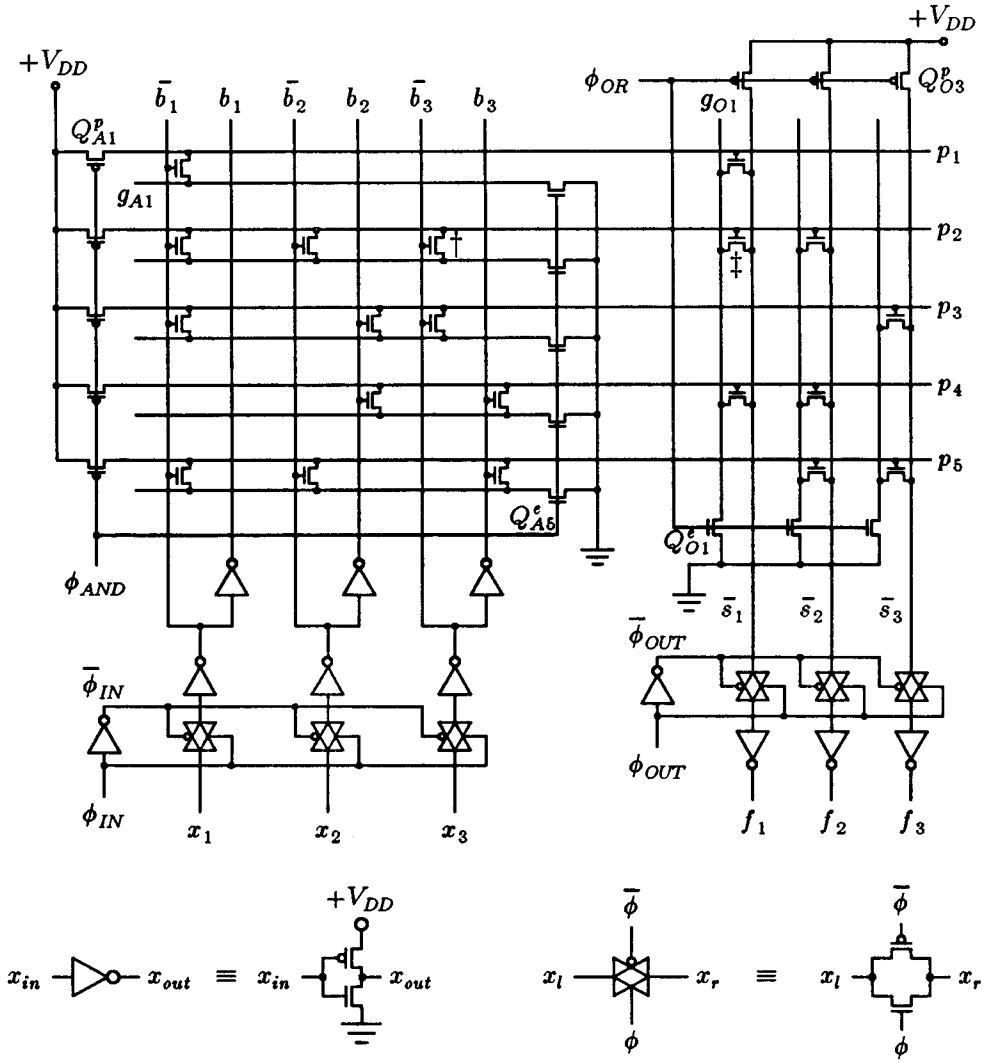
3.1. Structure and Signal Definitions

As shown by the small example in Figure 1, a PLA consists of four regular sub-regions: an array of input decoders (lower left), an AND plane (upper left), an OR plane (upper right), and an array of output buffers (lower right). Input vectors enter the input decoders on N_i primary input lines. The input decoders latch the input signals and drive up to $2N_i$ bit line signals across the AND plane. In the AND plane, N_p product line signals are computed and driven across the OR plane. In the OR plane, N_o sumbar[1] line signals are computed and driven to the output buffers. The output buffers latch the sumbar signals and drive the primary output lines that lead to other parts of the system. All events in the PLA's computations are sequenced by two or more externally generated clock signals.

The i-th input decoder latches the input signal on input line x_i and then drives non-inverted and inverted secondary signals onto bit lines b_i and $\bar{b}_i$, respectively. Multiple-input decoders [FlMa75] will not be considered. A bit line may be omitted if the corresponding signal is not used in the AND plane. As is common in dynamic CMOS designs, the input latches are assumed to be implemented using transmission gates [WeEs85]. The bit-line buffers are implemented as two levels of static CMOS inverters.

The AND(OR) plane is composed of $N_p(N_o)$ dynamic NOR gates, every gate driving a different product(sumbar) line. Each NOR gate has up to $N_i(N_p)$ inputs as determined by the presence of n-type AND(OR)-crosspoint transistors. Each AND(OR)-crosspoint transistor has its drain

[1] A sumbar line carries a signal that corresponds to the complement of a sum of products. The new term prevents confusion between the outputs of the OR plane and the primary outputs of the PLA.



(a) inverting buffer (b) transmission gate

$$f_1 = x_1 + x_1x_2x_3 + \bar{x}_2\bar{x}_3$$
$$f_2 = x_1x_2x_3 + \bar{x}_2\bar{x}_3 + x_1x_2\bar{x}_3$$
$$f_3 = x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3$$

Note: † and ‡ indicate redundant crosspoints

Figure 1 - Standard Dynamic CMOS NOR-NOR PLA Schematic

terminal connected to a product(sumbar) line, its gate terminal connected to a bit(product) line, and its source terminal connected to an AND(OR)-plane switched-ground line. Toward the beginning of the PLA computation cycle, clock $\phi_{AND}(\phi_{OR})=0$ causing each product(sumbar) line $p_j(\bar{s}_k)$ to be precharged to 1 via a p-type precharge transistor $Q^p_{Aj}(Q^p_{Ok})$. In this paper, as described in [WeEs85], it is assumed that a separate switched-ground line $g_{Aj}(g_{Ok})$ connects together the source terminals of all AND(OR)-crosspoint transistors that can drive a particular $p_j(\bar{s}_k)$. When the AND(OR) plane is evaluated, $g_{Aj}(g_{Ok})$ is connected to *GND* via an n-type evaluation transistor $Q^e_{Aj}(Q^e_{Ok})$. If the signal on a bit(product) line is 1(**1**), then the AND(OR)-crosspoint transistors to which the line is connected are turned on, driving a subset of the product(sumbar) lines to 0.

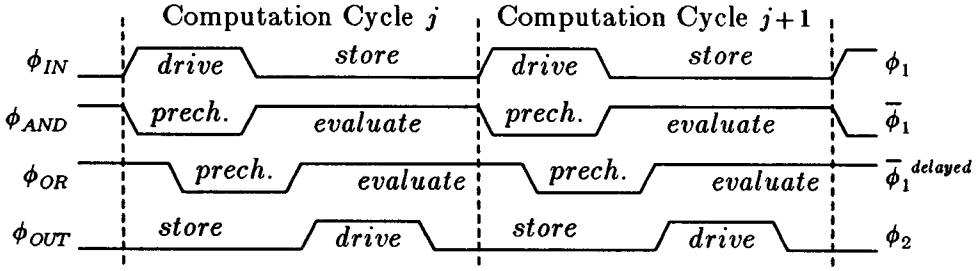
We say that a bit line $b_i(\bar{b}_i)$ *controls* a product line p_j if an AND-crosspoint transistor exists at their intersection. One verifies that $b_i(\bar{b}_i)$ controls p_j iff $\bar{x}_i(x_i)$ is a factor in the expression for p_j. Equivalently, $b_i(\bar{b}_i)$ controls p_j iff $x_i=1$ $(x_i=0)$ implies $p_j=0$. In a similar way, a product line p_j *controls* a sumbar line $\bar{s}_k$ iff an OR-crosspoint transistor exists at their intersection. We will also say that p_j *controls* output f_k iff p_j controls $\bar{s}_k$. One verifies that p_j controls f_k iff p_j is a term in the implemented SOP expression for f_k. Thus, if p_j controls f_k, then $p_j=1$ implies $\bar{s}_k=0$ and $f_k=1$. However, if $p_j=1$ implies $f_k=1$, i.e. if f_k covers p_j, it is not necessarily true that p_j controls f_k.

The signal on the k-th sumbar line $\bar{s}_k$ is latched by an output transmission gate and driven out of the PLA on primary output f_k by a static CMOS inverting buffer. If the complement of a function is easily specified using the set of intermediate products already required by the SOP forms of other functions, then a non-inverting buffer might be preferred. PLAs with non-inverting output buffers will not be considered in this paper.

3.2. Timing

Figure 2 shows a typical clocking scheme for the dynamic CMOS PLA. For other clocking schemes see [WeEs85, Ha87, HaRe85]. To prove general testability results, it is useful to abstract away details specific to particular schemes, and to retain only the essential temporal constraints in one generalized timing scheme. In fact, this has been done in Figure 1 in which appear the four general clocks ϕ_{IN}, ϕ_{AND}, ϕ_{OR}, and ϕ_{OUT}. In

actual PLA schematics, only system clocks or secondary clocks readily derived from the system clocks will be shown, i.e. ϕ_1, $\overline{\phi}_1$, $\overline{\phi}_1^{delayed}$, and ϕ_2.



Note: Generalized PLA clocks appear at the left margin; system clocks appear at the right margin.

Figure 2 - Two-Phase Non-Overlapping PLA Clocking Scheme [WeEs85]

During each *computation cycle*, i.e. the period of time between successive rising edges of the input clock ϕ_{IN}, one vector of output values is computed for one vector of input values. Clocks ϕ_{IN} and ϕ_{OUT} pass alternately through *drive* and *store* phases when they carry 1 and 0 signals, respectively. During its drive phase, $\phi_{IN}(\phi_{OUT})$ configures all of the input(output) transmission gates to allow signals to be driven through to the inputs of the input decoders (output buffers). During its store phase, $\phi_{IN}(\phi_{OUT})$ opens all input(output) transmission gates, causing the existing signals to be stored on the gate capacitances of the corresponding static inverters.

Clocks ϕ_{AND} and ϕ_{OR} pass alternately through *precharge* and *evaluate* phases when they carry 0 and 1 signals, respectively. During its precharge phase, $\phi_{AND}(\phi_{OR})$ simultaneously turns off all of the n-type evaluation transistors $Q^e_{Aj}(Q^e_{Ok})$, and turns on all of the p-type precharge transistors $Q^p_{Aj}(Q^p_{Ok})$, thus driving a 1 onto all product(sumbar) lines. During its evaluation phase, $\phi_{AND}(\phi_{OR})$ turns off the corresponding precharge transistors, and connects all the switched-ground lines $g_{Aj}(g_{Ok})$ to *GND* by turning on the corresponding evaluation transistors. Shortly after AND(OR)-plane evaluation begins, a product(sumbar) line that is connected to at

least one conductive AND(OR)-crosspoint transistor is driven, i.e. evaluated, to 0; otherwise, the line is left carrying a **1**.

4. Fault Models

In VLSI testing, it is necessary to simplify and abstract the real physical failure modes, and in their place use idealized fault models (See [AbFu86] for an overview of fault modelling). Fault models capture the principal effects of the physical defects, but are simple enough to reduce to manageable levels the complexity of the resulting problems in testability analysis and test pattern generation. At roughly the same low level of abstraction, i.e. closely related to circuit layout, lie four switch-level fault models (See [Haye84] for an overview of switch-level modelling):

interconnect break: A line of conductor (e.g. metal, polysilicon, or diffusion) is completely divided into two *segments* by an unintended gap.

ohmic short: In the course of forming the lines on a chip, a small mislaid piece of conducting, but possibly appreciably resistive, material can electrically connect two normally distinct lines. This model is more general that the ideal resistanceless bridging fault model studied by other authors (e.g. [Mei74]).

transistor stuck-open: This model is used to represent an n-type or p-type MOS transistor whose channel is permanently non-conductive. The signal on the gate terminal is assumed to be unaffected. A stuck-open fault can transform a purely combinational circuit into a sequential circuit if the fault creates an isolated line: such a line is capable of remembering the last driven logic value, thereby introducing a state variable into the circuit [Wads78].

transistor stuck-on: This model is used to represent a MOS transistor whose channel is permanently conductive. As with the stuck-open fault, the signal on the gate terminal of a stuck-on transistor is assumed to be unaffected. Stuck-on faults can produce fights and can thus be unclean [Brzo87].

In practice, very little use has been made of the interconnect break: it has been judged as providing insufficient abstraction to produce tractable problems of analysis and test pattern generation. For similar reasons, at least in gate-level analyses, the resistanceless short fault model has received far more attention than the ohmic short. The transistor stuck-

open and stuck-on fault models have recently received considerable attention because of concern that such transistor faults might not be detected by tests designed assuming the "classical" gate-level fault model, the stuck-at fault [Wads78].

The unconventional resistive ohmic short fault model is used here for several reasons. First, it is more general that the resistanceless bridging fault. Second, it has been shown that CMOS gates can be modified to make ohmic shorts to V_{DD} and *GND* clean [Brzo87]. Third, in CMOS gate networks in which 0(1)s are produced by paths through the conductive, but appreciably resistive, channels of n(p)-type MOS transistors, ohmic shorts are often indistinguishable from resistanceless shorts. Finally, an ohmic short between a transistor's source and drain is conveniently equivalent to a stuck-on fault affecting that transistor.

The node fault is a switch-level fault model that expresses a restriction on the signal carried by an electrical node. A similar fault model was also used in the switch-level fault simulator FMOSSIM [BrSh85].

node fault: The node fault corresponds to physical faults that cause the signal on an electrical node or line to remain at a valid 0 or 1, regardless of the inputs applied to the circuit. For example, a node-at-0(1) could be used to model a low-resistance short from a line to $GND(V_{DD})$.

It is useful to note some fault equivalences involving the stuck-at and the switch-level fault models. Assuming that a unique driving line stem exists for each node, a node fault is equivalent to a stuck-at affecting only the line stem. A transistor stuck-open(on) is equivalent to a stuck-at-0(1) affecting only the transistor's gate. An interconnect break that creates a segment whose signal subsequently drifts to logic 0(1) is equivalent to a stuck-at-0(1) affecting that segment. In terms of the primitive fault models defined above, a stuck-at fault affecting only one fan-out branch could be viewed naturally as a double fault: i.e. an interconnect break and a node fault affecting the resulting segment.

The five fault models just described are suitable for general CMOS circuits. In a structured circuit such as the PLA, a more abstract functional-level fault model, the crosspoint (also called contact) fault [OsHo79], has proved useful:

crosspoint fault: Either through programming errors (e.g. in field programmable PLAs), or through manufacturing defects, spurious crosspoint transistors can appear in a PLA. Alternatively, and more probably in the case of embedded custom PLAs, crosspoint transistors can fail to be created at locations where they were intended. The crosspoint fault model assumes either the appearance of an undesired and fully functioning crosspoint transistor where there shouldn't be one, or the complete absence of an intended crosspoint transistor.

Interconnect breaks and node faults can affect every line in a circuit; similarly, stuck-opens and stuck-ons can affect every transistor. Crosspoint faults can affect every crosspoint location in the AND and OR planes. However, an ohmic short affects a pair of lines. Most line pairs chosen arbitrarily from a complete line list will never be shorted together in a real circuit because the corresponding lines are far apart in the physical layout. Therefore, only certain classes of signal pairs need to be considered in ohmic short fault analyses: those pairs corresponding to lines of the same conducting layer which are close together (e.g. conductors that run parallel to one another for a relatively long distance) [GaCrVe80].

Finally, we define some terminology [Smit79] which will prove useful in the discussion of the fault properties of the PLA with respect to interconnect breaks. A *growth*(*shrinkage*) fault is a fault that increases(decreases) the number of 1-vertices covered by a product. Crosspoint faults in the AND plane are either growth or shrinkage faults. An *appearance*(*disappearance*) fault is one that causes a new product to control an output (an expected product to be omitted). Crosspoint faults in the OR plane are either appearance or disappearance faults.

5. Redundancy

Informally, a *redundant element* is a line, transistor, or sub-circuit whose absence would not affect the Boolean functions implemented at the outputs. An element that is not redundant is *irredundant*. Usually a designer would prefer to avoid including redundant elements since they add expense. In practice, however, it is often uneconomical to find completely irredundant designs. Redundancy is a consideration in testability analysis because faults in redundant elements may be undetectable and therefore unclean. The presence of a faulty redundant element can invalidate the tests for other elements in the circuit [HaRe87]. The presence of

redundant elements can affect the testability of other irredundant elements: e.g. error-correcting circuits may mask error conditions.

An input line, bit line, product line, or sumbar line is trivially redundant if there is no way in which a signal could propagate from it to any of the primary outputs. For example, a bit line is redundant if it is not connected to any AND-crosspoint transistors. Except for the illustrative example in Figure 1, we will assume that such obvious redundancies will not occur. An input line x_i can also be redundant in a functional sense if no output functionally depends[2] on the value of x_i. A product line is functionally redundant if all of the output functions could be implemented using only the other product lines. We will assume that no products will be redundant as a result of being controlled by both a bit line b_i and its complement $\bar{b}_i$. Formally, a product p_j is *irredundant with respect to output* f_k iff there exists an input X such that p_j is the only product that both controls f_k and covers X. A product is *irredundant* if it is irredundant with respect to at least one output.

A crosspoint transistor is redundant if removing it produces no logical change in the output functions. Thus, an AND-crosspoint transistor Q_A that controls a product line p_j is redundant if removing Q_A produces a new product p_j^* which is still covered by all of the original output functions controlled by p_j. An OR-crosspoint transistor between product line p_i and sumbar line $\bar{s}_k$ is redundant if, for every input vector X that is covered by p_i, there exists another product p_j controlling f_k that also covers X. If a bit(product) line is connected to no irredundant AND(OR)-crosspoint transistors, then the line itself is redundant: the line together with all of the crosspoint transistors to which it connects could be removed without changing the functions implemented by the PLA. It is easy to show that a product line p_j covers another product line p_i iff for each AND-crosspoint transistor at the intersection of p_j and a bit line, there is also an AND-crosspoint transistor at the intersection of p_i and the same bit line.

In the PLA shown in Figure 1, the bit line b_1, the AND-crosspoint transistor marked with a "†", and the OR-crosspoint transistor marked with a "‡" are redundant.

---

[3] $f(X)$ *functionally depends* on x_i iff there exists an X_{i0} such that $f(X_{i0}) \neq f(X_{i1})$.

6. Testability Properties of the Dynamic CMOS PLA

6.1. Assumptions

It is important to state clearly a few simplifying assumptions that underlie the subsequent results.

Assumption 1: We assume that a faulty circuit contains only one fault, unless otherwise stated.

Assumption 2: The voltage on a line that has been isolated from all possible driving elements (e.g. by interconnect breaks and stuck-open transistors) will have drifted to a valid logic 0 signal by the time test vectors are applied to the circuit.

Assumption 3: Faults that are detectable only if assumptions must be made as to the detailed nature (e.g. timing or electrical behaviour) of external circuits that drive or read from the PLA will be considered unclean.

Assumption 4: A weak signal, say $0^w(1^w)$, will be treated like a normal 0(1) for the purposes of fault detection.

Assumption 1 is the so-called "single fault assumption". Assumption 2 must be made to avoid the indeterminate signals that might otherwise arise on isolated lines. In many technologies, a chip's substrate is permanently connected to *GND* making drift to 0 likely as a result of leakage currents. Assumption 3 reflects a pessimism necessary to ensure that the subsequent results are free from complicated environment-specific conditions. Assumption 4 is justified by the following arguments: If a 0(1) becomes a $0^w(1^w)$ as a result of a fault, then assuming that the signal is not attenuated is pessimistic because it makes the effects of the fault harder to detect. In the dynamic CMOS PLA circuits that we will be considering, weak signals either appear immediately at the inputs of inverters, or at the primary outputs of the PLA: in both cases we assume that, with conservative timing, it is likely that a $0^w(1^w)$ would be interpreted correctly as a full-strength 0(1).

6.2. Node Faults

The testability properties, with respect to node faults, of each of the PLA's lines will be considered in turn, proceeding roughly from the inputs through to the outputs.

Proposition 1: In a standard dynamic CMOS PLA, a node fault affecting input x_i is clean iff there exists an output f_k that functionally depends on x_i.

Proof: Easily shown to be true for any combinational circuit. □

A node-at-0(1) affecting input x_i is equivalent to both: (1) a node-at-0(1) affecting the input node of the inverter that drives bit line $\bar{b}_i$, and (2) a node-at-1(0) affecting $\bar{b}_i$. In the next proposition, we consider a node fault affecting a non-inverting bit line b_i.

Proposition 2: In a standard dynamic CMOS PLA:

(a) A node-at-0 affecting bit line b_i is clean iff there exists an input vector X_{i1} and an output f_k such that: (1) f_k covers X_{i0}, and (2) f_k does not cover X_{i1}.

(b) A node-at-1 affecting bit line b_i is clean iff there exists an input vector X_{i0} and an output f_k such that: (1) f_k covers X_{i0}, and (2) all products controlling f_k that also cover X_{i0} have $\bar{x}_i$ as a factor.

Proof:

(a) *If*: That f_k covers X_{i0} but not X_{i1} implies that bit line b_i controls all of the products p_j that control f_k and also cover X_{i0}. A single-vector test for a node-at-0 affecting b_i is X_{i1}, since the fault will cause all of these same covering products p_j to be **1** instead of 0, implying that f_k will be 1 instead of 0.

Only if: If the fault is clean, there must exist a single-vector test since the fault does not introduce memory. To stimulate a node-at-0, the test vector must place a 1 on b_i: i.e. it must have the form X_{i1}. Let **P** represent the set of products controlled by b_i. A node-at-0 produces a **1** instead of a 0 on all the products in **P** that happen to cover X_{i0}; the remaining products in **P** are 0 when X_{i1} is applied, regardless of the presence of the node-at-0. The fault is observed at an output f_k as a 1 instead of a 0 if: (1) f_k is controlled by at least one of the products in **P** that covers X_{i0} (implying that f_k covers X_{i0}), and (2) f_k

does not cover X_{i1} (required to ensure that the effect of the fault is observed).

(b) *If*: Assume that an output f_k covers an input vector X_{i0}, and that all products that control f_k and also cover X_{i0} have $\bar{x}_i$ as a factor. When the fault is present, all of the products that are controlled by b_i, i.e. all products that have $\bar{x}_i$ as a factor, will be forced to 0 when X_{i0} is applied because b_i^* is 1. Thus f_k^* will be 0 instead of 1.

Only if: If the fault is clean, there must exist a single-vector test as no memory is introduced by the fault. To stimulate the fault, the test vector must have the form X_{i0}. Let **P** represent the set of products that are controlled by b_i, i.e. those products that have $\bar{x}_i$ as a factor. To observe the fault at an output f_k, f_k must cover X_{i0}, and all of the products that control f_k and also cover X_{i0} must be members of **P**, so that f_k^* is 0 instead of 1 when X_{i0} is applied. □

If node faults affecting a bit line b_i are unclean then the PLA contains redundancy. That a node-at-0 is unclean implies that each of the AND-crosspoint transistors on b_i is redundant. But all redundant AND-crosspoint transistors driven by the same bit line can be removed without changing any of the PLA's output functions [OsHo79], so b_i is itself redundant. That a node-at-1 affecting b_i is unclean implies that all of the products that b_i controls are redundant.

Next we consider the testability of node faults affecting the four general clock lines, ϕ_{IN} and ϕ_{OUT}, then ϕ_{AND} and ϕ_{OR}.

Proposition 3: In a standard dynamic CMOS PLA:

(a) A node-at-0 affecting clock $\phi_{IN}(\phi_{OUT})$ is clean.

(b) A node-at-1 affecting clock $\phi_{IN}(\phi_{OUT})$ is unclean.

(c) A node fault affecting secondary clock $\bar{\phi}_{IN}(\bar{\phi}_{OUT})$ is unclean.

Proof:

(a) A node-at-0 affecting ϕ_{IN} disconnects the input lines to all first-level input inverters from the primary input lines to the PLA. By Assumption 2, the inputs to these inverters are thus all 0. This fault is clean if at least one output function is non-trivial. Similarly, a node-at-0 affecting ϕ_{OUT} disconnects the inputs to all output inverters from the sumbar lines. By Assumption 2, these isolated lines carry valid 0

signals forcing the PLA outputs to appear affected by node-at-1 faults. Any 0-vertex of any f_k will detect the fault.

(b) A node-at-1 affecting $\phi_{IN}(\phi_{OUT})$ causes the input(output) transmission gates to be transparent: i.e. logic signals are passed through virtually unattenuated. A node-at-1 affecting ϕ_{IN} could be detected if the next input vector arrives soon enough before the OR-plane begins evaluating at the rising edge of ϕ_{OR} to alter consistently the output signals that are latched for the current input vector. In this case, a test is easily constructed. But the fault is undetectable, and thus unclean, if the input signals are stable until after the OR-plane has been evaluated, i.e. just after the falling edge of ϕ_{OUT}. Also, the fault is not reliably detectable if the timing is such that indeterminate or non-repeatable output signals are produced. Similarly, the timing of circuits that read the PLA's outputs must be considered to decide if clock ϕ_{OUT} node-at-1 is detectable. Thus by Assumption 3, the faults are unclean.

(c) A node-at-1(0) fault affecting $\overline{\phi}_{IN}$ or $\overline{\phi}_{OUT}$ is equivalent to stuck-open(on) faults affecting all p-type transistors in the input or output transmission gates, respectively. A node-at-1 does not affect the gated transmission of 0s, but attenuates 1s into 1^ws, possibly affecting the PLA's behaviour. By Assumption 4, this fault is unclean. A node-at-0 makes the corresponding transmission gates fully transparent for 1s, and allows 0s to pass through attenuated as 0^ws when the controlling clock is in its store phase. By Assumptions 3 and 4, this fault is unclean. □

Proposition 4: In a standard dynamic CMOS PLA, a node fault affecting clock $\phi_{AND}(\phi_{OR})$ is clean.

Proof: A node-at-0 affecting $\phi_{AND}(\phi_{OR})$ connects all of the products(sumbars) to V_{DD} via the AND(OR)-plane precharge transistors, and precludes paths to *GND* via the evaluation transistors. A single-vector test consists of a 0(1)-vertex for any output f_k. A node-at-1 affecting $\phi_{AND}(\phi_{OR})$ prevents all products(sumbars) from being precharged via the AND(OR) plane precharge transistors, but allows paths to *GND* via crosspoint and evaluation transistors. Under Assumption 2, any 1(0)-vertex detects the fault. □

The following two propositions consider the testability of the product and sumbar lines, and their respective switched-ground lines.

Proposition 5: In a standard dynamic CMOS PLA:

(a) A node-at-0 affecting product line p_j is clean iff p_j is irredundant;

(b) A node-at-1 affecting product line p_j is clean iff p_j controls some output f_k.

Proof: Parts (a) and (b) follow directly by noting that the faults do not introduce memory and then applying the definitions of "irredundancy" and "control", respectively. □

Proposition 6: In a standard dynamic CMOS PLA, a node-at-1 affecting switched-ground line $g_{Aj}(g_{Ok})$ in the AND(OR) plane is clean, but a node-at-0 is unclean.

Proof: A node-at-1 precludes paths between $p_j(\bar{s}_k)$ and *GND*. A test consists of any 0-vertex for some output that is controlled by p_j (consists of any 1-vertex for f_k). To detect a node-at-0, one would like to attempt to store a 1 on $g_{Aj}(g_{Ok})$ and then observe whether the signal changes to 0. But such a test is impossible because the time when the PLA outputs are observable coincides with the time when $g_{Aj}(g_{Ok})$ is connected via $Q^e_{Aj}(Q^e_{Ok})$ to *GND*. □

At this point it is noted that a node-at-0(1) affecting sumbar $\bar{s}_k$ is equivalent to both a node-at-0(1) affecting the input of the output buffer that drives f_k, and a node-at-1(0) affecting the output f_k itself. Consequently, only the last fault is considered.

Proposition 7: In a standard dynamic CMOS PLA, node faults affecting an output f_k are clean.

Proof: Any 1(0)-vertex for f_k detects a node-at-0(1) affecting f_k. □

Theorem 1: In summary, the standard dynamic CMOS PLA has the following properties with respect to single node faults:

(a) The following are unclean:

(i) node faults affecting secondary clocks $\bar{\phi}_{IN}$ and $\bar{\phi}_{OUT}$;

(ii) node-at-1 faults affecting clocks ϕ_{IN} and ϕ_{OUT};

(iii) node-at-0 faults affecting switched-ground lines in either plane.

(b) The following are clean:

(i) node-at-1 faults affecting switched-ground lines in either plane;

(ii) node-at-0 faults affecting ϕ_{IN} and ϕ_{OUT};

(iii) node faults affecting ϕ_{AND} and ϕ_{OR}.

(c) Node faults affecting the remaining lines are clean under the irredundancy conditions detailed in Propositions 1, 2, and 5.

Proof: Follows directly from Propositions 1 to 7. □

6.3. Transistor Stuck-Opens and Stuck-Ons

The PLA's transistors will be considered in the following order: the transmission gate transistors, the inverter transistors, the precharge and evaluation transistors, and finally the crosspoint transistors.

Proposition 8: In a standard dynamic CMOS PLA, a stuck-open or stuck-on fault affecting a transmission gate transistor is unclean.

Proof: Three cases need to be considered:

(a) If an n(p)-type transmission gate transistor is stuck-open, 1(0) signals continue to be transmitted well, but 0(1) signals become weak. The fault is thus unclean by Assumption 4.

(b) Say an input transmission gate transistor is stuck-on. The fault is undetectable if the PLA input signals remain stable from the time evaluation of the AND plane begins (i.e. the rising edge of ϕ_{AND}) until the time the PLA outputs are latched (i.e. the falling edge of ϕ_{OUT}). Thus under Assumption 3, the fault is unclean.

(c) Say an n(p)-type transistor in the output transmission gate for f_k is stuck-on. The fault is undetectable if the sumbar signals remain stable long enough for the circuits reading the PLA's outputs to latch them correctly. Thus, under Assumption 3 the fault is unclean. □

For a gate G embedded within a combinational network N, a *0(1)-vertex* for G is an input vector to N that produces the value 0(1) on G's output node. Within either a static or a dynamic CMOS gate, we will say that an n(p)-type transistor Q is a *bottleneck transistor* iff, for all of the gate's 0(1)-vertices, all paths from the gate's output node to $GND(V_{DD})$

pass through Q. Both transistors in a static inverter as well as the precharge and evaluation transistors in a dynamic gate are bottleneck transistors. The following lemma relates the testability of node faults affecting the output node of an embedded CMOS gate to the testability of stuck-open faults affecting the gate's bottleneck transistors.

Lemma 1: Let G be either a static or dynamic CMOS gate, embedded in a combinational network N, in which the output function from N's primary inputs to G's output node is non-trivial. A node-at-0(1) affecting G's output node is clean iff a stuck-open affecting a p(n)-type bottleneck transistor is clean.

Proof:

If: At least two test vectors are necessary to detect a stuck-open affecting a p(n)-type bottleneck transistor in a static gate, or a stuck-open affecting the precharge transistor in a dynamic gate: a 0(1)-vertex by itself would not excite the fault, and a 1(0)-vertex might lead to an indeterminate output signal if not preceded recently by a 0(1)-vertex. Two vectors are sufficient because the surrounding network is combinational. Thus the first vector must be a 0(1)-vertex, and the second vector a 1(0)-vertex. In the case of stuck-opens in a dynamic gate affecting bottleneck transistors, other than the precharge transistor, the precharge phase performs the same function as would a separate first vector, so the second vector alone suffices as a test. In all cases the effect of a stuck-open affecting a p(n)-type transistor is that, for the last test vector, a **0(1)** appears on G's output node instead of a 1(0), i.e. the output node appears to be affected by a node-at-0(1). Thus if a stuck-open fault affecting a p(n)-type bottleneck transistor is clean, the final vector of a test could also be used to detect a node-at-0(1) affecting the output node.

Only if: If a node-at-0(1) is clean, then a single-vector test must exist since the overall circuit is combinational and a node fault cannot introduce memory. To excite the fault, any test vector X must be a 1(0)-vertex for G. To detect the fault, on at least one output of N, fault-free and faulty circuits must produce complementary signals when X is applied. If G is a dynamic gate which precharges to 0(1), then X alone detects stuck-opens affecting the p(n)-type evaluation transistor and any other p(n)-type bottleneck transistors. If G is a static gate, or is a dynamic gate which precharges to 1(0), then any 0(1)-vertex for G

followed by X will detect stuck-opens affecting p(n)-type bottleneck transistors. □

In a dynamic CMOS PLA, all transistors except the transmission gate and crosspoint transistors are bottleneck transistors. This observation simplifies the testability analysis for the remaining stuck-open faults.

Proposition 9: In a standard dynamic CMOS PLA:

(a) A stuck-open fault affecting an inverter transistor in the first-level of the decoder for input x_i is clean iff there exists an output f_k that depends on x_i.

(b) A stuck-open fault affecting a p-type inverter transistor in the second-level of the decoder for input x_i is clean iff there exists an input vector X_{i1} and an output f_k such that: (1) f_k covers X_{i0}, and (2) f_k does not cover X_{i1}.

A stuck-open fault affecting an n-type inverter transistor in the second-level of the decoder for input x_i is clean iff there exists an input vector X_{i0} and an output f_k such that: (1) f_k covers X_{i0}, and (2) all products controlling f_k that cover X_{i0} have $\bar{x}_i$ as a factor.

(c) A stuck-open fault affecting an inverter transistor in the output buffer is clean.

(d) A stuck-open fault affecting a transistor in either one of the two inverting buffers that drive the secondary clocks $\bar{\phi}_{IN}$ and $\bar{\phi}_{OUT}$ is unclean.

(e) A stuck-on fault affecting an inverter transistor is unclean.

Proof:

(a) Note that a node-at-0(1) affecting input x_i is equivalent to a node-at-1(0) affecting bit line $\bar{b}_i$, then apply Lemma 1 and Proposition 1.

(b) Apply Lemma 1 and Proposition 2.

(c) Apply Lemma 1 and Proposition 7.

(d) Since clocks ϕ_{IN} and ϕ_{OUT} are free-running, a stuck-open fault affecting an n(p)-type transistor will appear like a node-at-1(0) affecting $\bar{\phi}_{IN}$ and $\bar{\phi}_{OUT}$, respectively. By part (c) of Proposition 3, the fault is unclean.

(e) A stuck-on fault affecting an n(p)-type inverter transistor is masked with an inverter input of 1(0), and produces a fight with an input of 0(1). Thus no test is possible. □

Proposition 10: In a standard dynamic CMOS PLA:

(a) A stuck-open fault affecting AND-plane precharge transistor Q^p_{Aj} is clean if product p_j is irredundant with respect to some output f_k.

(b) A stuck-open fault affecting AND-plane evaluation transistor Q^e_{Aj} is clean if product p_j controls some output f_k.

(c) A stuck-open fault affecting an OR-plane precharge(evaluation) transistor $Q^p_{Ok}(Q^e_{Ok})$ is clean.

(d) A stuck-on fault affecting a precharge or evaluation transistor in either plane is unclean.

Proof:

(a) Note that all product lines implement non-trivial functions, then apply Lemma 1 and part (a) of Proposition 5.

(b) Apply Lemma 1 and part (b) of Proposition 5.

(c) Note that a node-at-0(1) affecting sumbar $\bar{s}_k$ is equivalent to a node-at-1(0) affecting output f_k, then apply Lemma 1 and Proposition 7.

(d) A stuck-on affecting a precharge transistor will change the evaluated signal from a 0(**1**) to a ?(1). Similarly, a stuck-on affecting an evaluation transistor will change the evaluated signal from a 0(**1**) to a 0(**1**, or **?** if full precharge is inhibited). Thus, either the fault is undetectable, or it produces indeterminate signals. Therefore the faults are unclean. □

Proposition 11: In a standard dynamic CMOS PLA:

(a) A stuck-open fault affecting a crosspoint transistor Q in either plane is clean iff Q is irredundant.

(b) A stuck-on fault affecting an AND-crosspoint transistor at the intersection of bit line $b_i(\bar{b}_i)$ and product line p_j is clean iff product p_j is irredundant with respect to some output f_k.

(c) A stuck-on fault affecting an OR-plane crosspoint transistor on sumbar $\bar{s}_k$ is clean.

Proof:

(a) Note that a stuck-open affecting a crosspoint transistor has the same effect as removing the transistor, then apply the definitions for a redundant crosspoint transistor and a clean fault.

(b) A stuck-on fault affecting an AND-crosspoint transistor on product line p_j is equivalent to a node-at-0 affecting p_j. Part (a) of Proposition 5 can then be applied.

(c) A stuck-on affecting the OR-crosspoint transistor at the intersection of product p_j and sumbar $\bar{s}_k$ causes $\bar{s}_k$ to be driven to 0 at every computation cycle, causing output f_k to appear affected by a node-at-1. By Proposition 7 any 0-vertex for f_k detects the fault. □

Theorem 2: In summary, the standard dynamic CMOS PLA has the following properties with respect to single transistor stuck-open and stuck-on faults:

(a) The following are unclean:

  (i) stuck-open and stuck-on faults affecting transmission gate transistors;

  (ii) stuck-open faults affecting transistors in the inverting buffers that drive the secondary clocks $\overline{\phi}_{IN}$ and $\overline{\phi}_{OUT}$;

  (iii) stuck-on faults affecting inverter transistors;

  (iv) stuck-on faults affecting precharge and evaluation transistors.

(b) The following are clean:

  (i) stuck-open faults affecting output buffer transistors;

  (ii) stuck-on faults affecting OR-plane crosspoint transistors.

(c) All remaining stuck-open and stuck-on faults are clean under the irredundancy conditions detailed in Propositions 9, 10, and 11.

Proof: Follows directly from Propositions 8 to 11. □

6.4. Interconnect Breaks

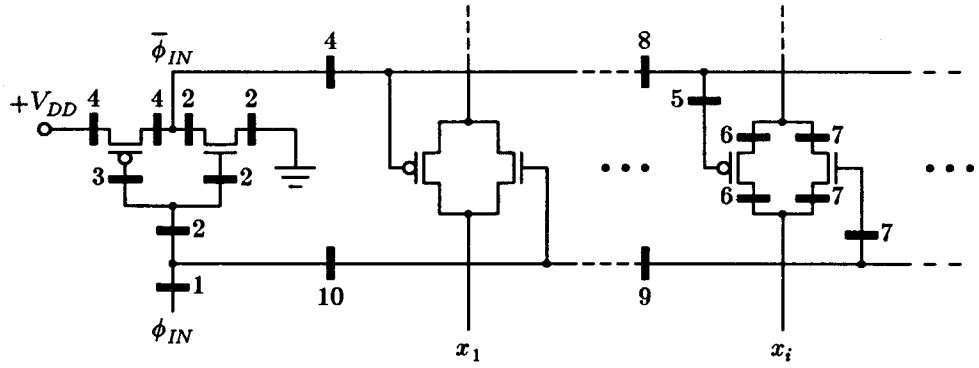
For most circuits, it is difficult to analyse the effects of interconnect breaks without detailed knowledge of the physical layout. In the case of the PLA, however, layouts tend to reflect the topology of the schematic, so it is practical to consider interconnect breaks directly.

The proofs of the following six propositions assume a layout topologically similar to the one given in [WeEs85]: i.e. a layout that has the same interconnect break properties as one corresponding to the schematic details shown in Figures 3 to 6. It has been assumed that the power supply lines have been laid out orthogonally across the first-level decoder inverters, the second-level decoder inverters, and the output buffers. It has further been assumed that the power supply lines to all inverters in the input decoders connect together first before connecting to higher levels in the power distribution tree. Thus, an interconnect break can cut off the path to $GND(V_{DD})$ for one or more inverters in the same level of the input decoder, or can affect all of the input decoders. No further assumptions will be made about the structure at higher levels in the power distribution tree.

An interconnect break can cut off all possible paths to a node from *GND*. Within the context of a long series of functional tests, it is not realistic to assume that the signal on such a node would drift to 0 according to Assumption 2, since it is difficult to ensure that a driving path to 1 was never enabled by a previous input vector. Thus, it will be assumed that a node with unidirectional drive to 0(1) will be affected by a node-at-0(1). This assumption is valid if the PLA is *initialized* [Ha87] prior to testing: i.e. a series of inputs has been applied recently that ensured that every node was driven to both 0 and 1 at least once. Applying a test set for single node faults, ignoring the resulting outputs, would initialize all irredundant lines.

Proposition 12: In a standard dynamic CMOS PLA, all interconnect breaks affecting lines in the input transmission gates and the associated clocking circuitry are unclean, with the one sole exception being a break affecting the line stem of clock ϕ_{IN} which is clean.

Proof: Figure 3 gives a detailed schematic for the input transmission gates and the associated clocking circuitry. The labelled dark rectangles represent all of the different classes of breaks. Under Assumption 2, a Class 1 break is equivalent to a node-at-0 affecting clock ϕ_{IN}, and is therefore clean according to part (a) of Proposition 3. Under Assumption 2, breaks of Classes 2 to 7 are equivalent to single transistor faults and node faults as listed in the lower half of the figure. They are thus unclean by Propositions 3, 8, and 9. Breaks of Classes 8, 9, and 10 are equivalent to



1 -	all inputs with a node-at-0	6 -	one p-type transmission gate transistor stuck-open
2 -	$\overline{\phi}_{IN}$ with a node-at-1	7 -	one n-type transmission gate transistor stuck-open
3 -	p-type inverter transistor stuck-on	8 -	many p-type transmission gate transistors stuck-on
4 -	$\overline{\phi}_{IN}$ with a node-at-0	9 -	many n-type transmission gate transistors stuck-open
5 -	one p-type transmission gate transistor stuck-on	10 -	all n-type transmission gate transistors stuck-open

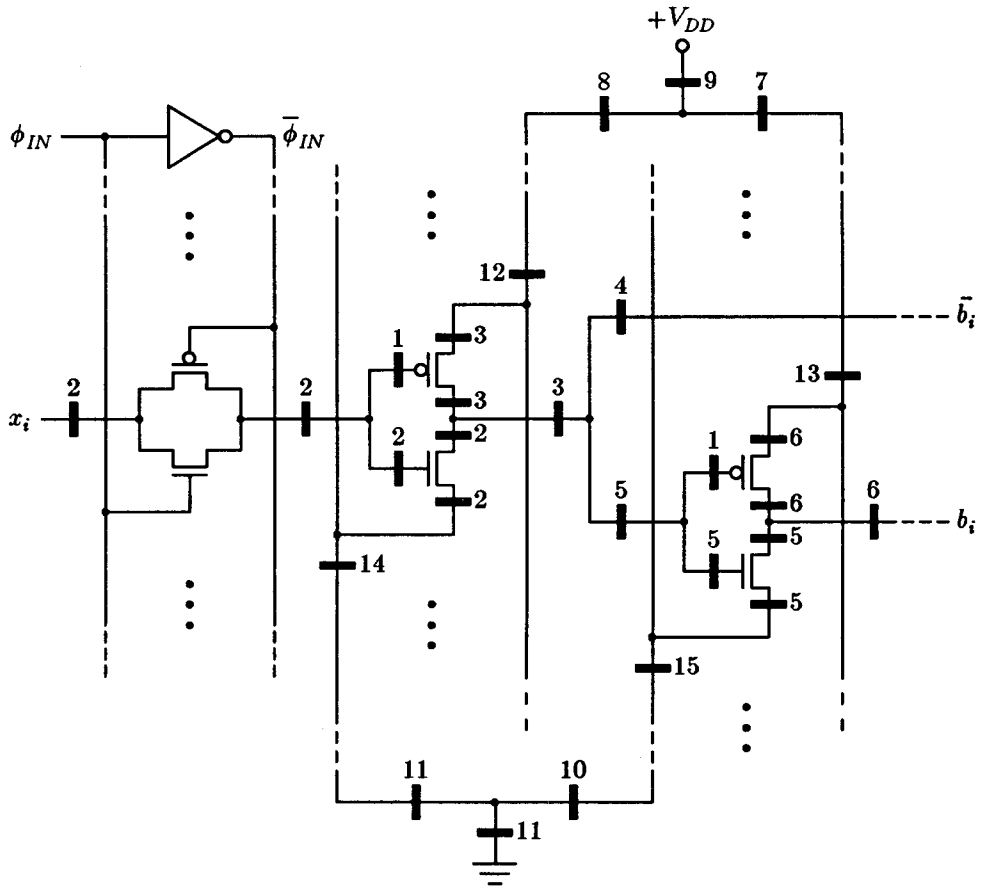
Note: Identical circuit to that used for the output transmission gates.

Figure 3 - Interconnect Breaks in the Input Transmission Gates

multiple stuck-opens or stuck-ons affecting transmission gate transistors. The argument used in the proof of Proposition 8 can be extended easily to show that these multiple faults are unclean. □

Proposition 13: In the input decoders of a standard dynamic CMOS PLA, interconnect breaks that isolate the gate of a single p-type inverter transistor are unclean. The remaining input decoder breaks are equivalent to one of the following four faulty behaviours:

(a) One or more input lines with a node-at-0(1);



1 - inverter transistor stuck-on
2 - x_i with a node-at-0
3 - x_i with a node-at-1
4 - all crosspoint transistors on $\bar{b}_i$ are stuck-open
5 - b_i with a node-at-1
6 - b_i with a node-at-0
7 - all b_is with a node-at-0
8 - all x_is with a node-at-1
9 - all b_is and $\bar{b}_i$s w. a node-at-0
10 - all b_is with a node-at-1
11 - all x_is with a node-at-0
12 - some x_is with a node-at-1
13 - some b_is with a node-at-0
14 - some x_is with a node-at-0
15 - some b_is with a node-at-1

Figure 4 - Interconnect Breaks in the Input Decoder Inverters

(b) One or more bit lines b_i with a node-at-0(1);

(c) All AND-crosspoint transistors on a bit line $\bar{b_i}$ stuck-open;

(d) All b_is and $\bar{b_i}$s with a node-at-0.

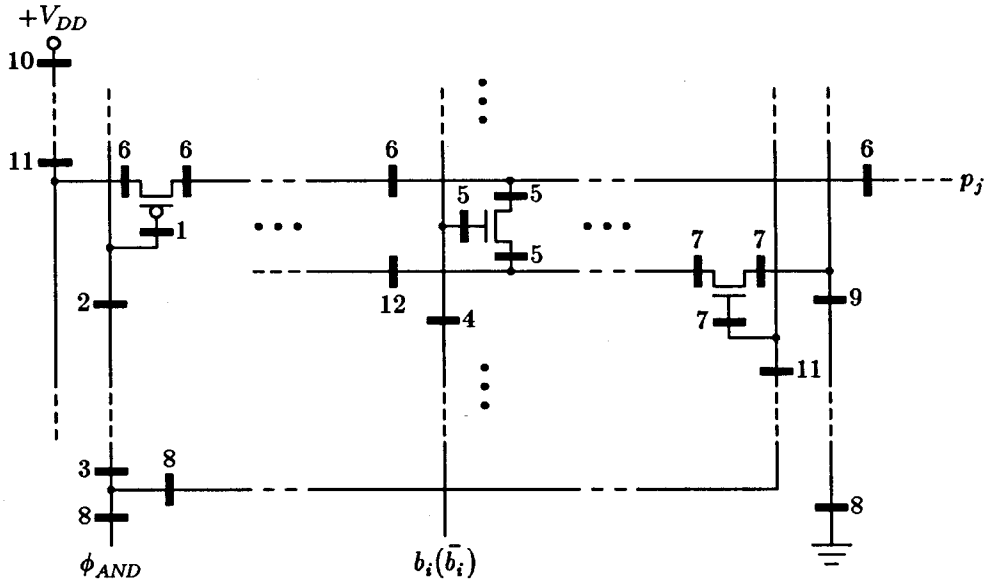
Proof: Figure 4 shows the detailed schematic for the input decoder and lists the different classes of breaks together with equivalent faults under Assumption 2. It is easily shown that breaks in Classes 2, 3, 8, 11, 12, and 14 are equivalent to one or more node faults affecting input lines. The effects of each component input node fault are orthogonal in the sense that each one creates a new set of output functions that are independent of the variable corresponding to the affected input line: a combination of input node faults can never be equivalent to a single input node fault. If any component node fault is clean then the multiple fault must be clean. Breaks in Classes 5, 6, 7, 10, 13, and 15 are equivalent to one or more node faults affecting bit lines b_i. One node-at-0(1) affecting a b_i causes growth(shrinkage) faults affecting the product lines controlled by b_i. A multiple bit line node-at-0(1) fault is clean if any component fault is clean because the effects of the individual faults cannot cancel: they are either all growth or all shrinkage faults. Breaks in Class 4 are equivalent to all transistors on a bit line $\bar{b_i}$ stuck-open. Each such stuck-open constitutes a growth fault affecting a different product line, so a Class 4 break is clean if any of the component stuck-opens is clean. Finally, breaks in Class 9 are equivalent to all b_is and $\bar{b_i}$s with a node-at-0. This multiple fault is equivalent to all PLA outputs being affected by a node-at-1, and clearly is clean because all output functions are non-trivial. Part (e) of Proposition 9 can applied to show that breaks in Class 1 are unclean. □

Proposition 14: In a standard dynamic CMOS PLA, interconnect breaks in the AND plane that isolate only the gates of one or more p-type precharge transistors are unclean. The remaining AND plane breaks are equivalent to one of the following two faulty behaviours:

(a) One or more AND-crosspoint transistors on either a single bit line, or on a single product line, are stuck-open;

(b) One or more product lines with a node-at-0(1).

Proof: Figure 5 shows the detailed schematic for the AND plane and lists the different classes of breaks together with equivalent faults under Assumption 2. It is easily shown that breaks in Classes 4, 5, and 12 are



1 - one precharge transistor stuck-on

2 - some precharge transistors stuck-on

3 - all precharge transistors stuck-on

4 - some AND-crosspoint transistors on $b_i(\bar{b}_i)$ are stuck-open

5 - a single AND-crosspoint transistor is stuck-open

6 - p_j with a node-at-0

7 - p_j with a node-at-1

8 - all product lines with a node-at-1

9 - some product lines with a node-at-1

10 - all product lines with a node-at-0

11 - some product lines with a node-at-0

12 - some AND-crosspoint transistors on p_j are stuck-open

Figure 5 - Interconnect Breaks in the AND Plane

equivalent to multiple AND-crosspoint transistor stuck-opens. Each component stuck-open is a growth fault, so any multiple stuck-open is clean if any individual stuck-open is clean. Breaks in Classes 6, 7, 8, 9, 10, and 11 are equivalent to node faults affecting product lines, either all node-at-0s or all node-at-1s. Each node-at-0(1) is a disappearance(appearance) fault, so any multiple node-at-0(1) is clean if any individual node-at is clean.

Under Assumption 2, the breaks in Classes 1, 2, and 3 are equivalent to one or more p-type precharge transistors stuck-on. The arguments in the proof of part (d) of Proposition 10 be extended to show that these last three classes of breaks are unclean. □

Proposition 15: In a standard dynamic CMOS PLA, most interconnect breaks in the OR plane are equivalent to one of the following faults:

(a) One or more OR-crosspoint transistors on either a single product line, or on a single sumbar line, are stuck-open;

(b) One or more output lines with a node-at-0(1).

Interconnect breaks that are equivalent to stuck-open redundant OR-crosspoint transistors are undetectable, and therefore unclean. A break that only isolates the gates of one or more p-type precharge transistors is unclean.

Proof: Analogous to the proof of Proposition 14. □

Proposition 16: In a standard dynamic CMOS PLA, interconnect breaks affecting lines in the output transmission gates and the associated clocking circuitry are all unclean, with one sole exception being a break affecting the line stem of clock ϕ_{OUT} which is clean.

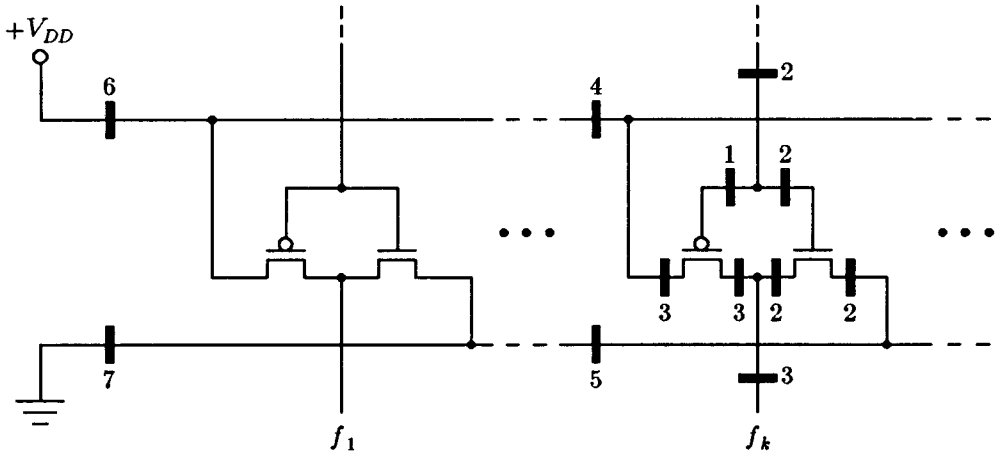
Proof: Analogous to the proof of Proposition 12. □

Proposition 17: In a dynamic CMOS PLA, interconnect breaks in the output buffers are equivalent to one or more PLA outputs having a node-at-0(1). One exception is a break that isolates the gate of a single p-type inverter transistor, which is unclean.

Proof: Easily shown by applying Assumption 2 for each of the breaks marked in Figure 6. A break equivalent to a multiple node fault affecting output lines is clean if any of the component faults is clean. □

Theorem 3: In summary, the standard dynamic CMOS PLA has the following properties with respect to single interconnect breaks:

(a) Breaks that isolate only the gates of p-type precharge or p-type inverter transistors are unclean;



1 - inverter transistor stuck-on
2 - f_k with a node-at-1
3 - f_k with a node-at-0
4 - some f_ks with a node-at-0
5 - some f_ks with a node-at-1
6 - all f_ks with a node-at-0
7 - all f_ks with a node-at-1

Figure 6 - Interconnect Breaks in the Output Buffers

(b) Breaks in the input or output transmission gate circuitry, with the exception of breaks affecting the line stems of ϕ_{IN} or ϕ_{OUT}, are unclean;

(c) All remaining breaks are equivalent to single or multiple node faults and stuck opens affecting input lines, bit lines, product lines, output lines, and crosspoint transistors. In no case can the effects of a multiple fault cancel. Thus, for all of these breaks, the fault is clean if any of the components of the equivalent multiple fault is clean, which will be the case subject to reasonable conditions of irredundancy.

Proof: Follows from Propositions 12 to 17. □

6.5. Ohmic Shorts

To a greater extent than for the other fault models, simplifying assumptions must be made as to which among the possible ohmic shorts should be considered. It is not even practical to consider shorts between all lines that may pass close to each other in the final layout, if only because such details of the layout are probably not fully known, or are difficult to extract. Instead, we shall only consider line pairs that we expect to have a significantly high probability of having shorts, such as shorts between adjacent parallel lines. Also, there is experimental support for neglecting shorts between lines of different conductive layers [GaCrVe80]. In PLAs, it is reasonable to consider shorts between adjacent input lines, product lines, and output lines. Also, shorts between adjacent bit lines (either $\bar{b}_i$ with b_i, or b_i with $\bar{b}_{i+1}$) should be considered. For dynamic PLAs, shorts between the product and sumbar lines and their respective switched-ground lines should be quite probable according to published layouts (e.g. [WeEs85]).

Proposition 18: The standard dynamic CMOS PLA has the following testability properties with respect to the ohmic short:

(a) An ohmic short from x_i to x_{i+1} is unclean;

(b) An ohmic short from $\bar{b}_i$ to b_i is unclean;

(c) An ohmic short from b_i to $\bar{b}_{i+1}$ is unclean;

(d) An ohmic short from p_j to p_{j+1} is clean iff there exists an input vector X such that either (1) p_j is the only product controlling an output f_k that covers X, and (2) X is not also covered by p_{j+1}, or the same two conditions hold with the roles of p_j and p_{j+1} reversed;

(e) An ohmic short from p_j to g_{Aj} is clean iff p_j is irredundant with respect to some output f_k;

(f) An ohmic short from $\bar{s}_k$ to $\bar{s}_{k+1}$ is clean iff f_k is not identical to f_{k+1};

(g) An ohmic short from $\bar{s}_k$ to g_{Ok} is clean;

(h) An ohmic short from f_k to f_{k+1} is unclean.

Proof:

(a) Depending on the external drivers, fights may occur when complementary signals are sent to adjacent inputs. Thus under Assumption 3, the fault is unclean. The fault would be clean if the inputs are driven

by precharged logic since wired-ANDs or -ORs would be created in the presence of shorts.

(b) Fights will occur at every computation cycle because the signals are always complementary, so the fault is unclean.

(c) Fights will occur when inputs x_i and x_{i+1} carry identical signals; with complementary signals the fault is not stimulated. Therefore, the fault is unclean.

(d) *If*: The ohmic short causes a wired-AND since the product lines are all precharged to **1** and then conditionally evaluated by the same clock. If there exists an input X such that $p_j(p_{j+1})$ is the only product controlling f_k that covers X, and X is not covered by $p_{j+1}(p_j)$, then X places a **1** on $p_j(p_{j+1})$, and 0s on all other products that control f_k as well as on $p_{j+1}(p_j)$. Such a vector X is a test because a 1 appears on f_k if the PLA is fault-free, and a 0 if the short is present. *Only if*: A single test vector X is required since the fault does not introduce memory. The first condition on X allows the signal on $p_j(p_{j+1})$ to be observed on f_k, and the second condition ensures that the fault will be stimulated.

(e) The fault is equivalent to a stuck-on fault affecting an AND-crosspoint transistor connected to p_j. Part (b) of Proposition 11 can then be applied.

(f) If f_k is not identical to f_{k+1}, then there exists an input vector X such that f_k carries a 0(1) and f_{k+1} carries a 1(0): i.e. $\bar{s}_k$ carries a 1(0) and $\bar{s}_{k+1}$ carries a 0(1). An ohmic short between two sumbar lines causes a wired-AND since they are both precharged to **1** and then conditionally evaluated by the same clock. Thus, in the presence of a short, both $\bar{s}_k$ and $\bar{s}_{k+1}$ carry a 0, and so both f_k and f_{k+1} carry a 1. The fault can be detected, so it is clean. The converse is easily shown.

(g) This fault is equivalent to a stuck-on fault affecting an OR-crosspoint transistor connected to $\bar{s}_k$. Part (c) of Proposition 11 can then be applied.

(h) Fights occur when the two adjacent outputs are driven to complementary signals. Otherwise, the fault is not detected. No test is possible, so the fault is unclean. □

6.6. Crosspoint Faults

The crosspoint fault has received much attention because of several important properties. First, it is independent of the particular technology used to implement the PLA: the fault model describes changes in the pattern of crosspoint devices. Second, the effects of crosspoint faults on the logical behaviour of the SOP forms of the output functions are well known [SoGa86]. Finally, a test set designed to detect all detectable single crosspoint faults also detects most multiple node faults, shorts, and transistor stuck-opens as well as some transistor stuck-ons [OsHo79].

Proposition 19: In a standard dynamic CMOS PLA:

(a) The absence of an AND(OR)-crosspoint transistor Q is clean iff Q is irredundant.

(b) The spurious presence of an AND-crosspoint transistor at the intersection of $b_i(\bar{b}_i)$ and p_j is clean iff there exists an input vector of the form $X_{i1}(X_{i0})$ such that p_j is the only product controlling output f_k that covers $X_{i1}(X_{i0})$.

(c) The spurious presence of an OR-crosspoint transistor at the intersection of p_j and $\bar{s}_k$ is clean iff f_k does not cover p_j.

Proof:

(a) By definition, a crosspoint transistor Q is irredundant iff removing Q causes a change in at least one output function: i.e. there exists an input vector X and an output f_k such that $f_k(X) \neq f_k^*(X)$. This last condition is equivalent to the fault being clean.

(b) *If*: Apply $X_{i1}(X_{i0})$ as a test vector. If the spurious AND-crosspoint transistor is present, then p_j carries a 0 instead of a **1**. But p_j is the only product controlling f_k that covers $X_{i1}(X_{i0})$. Therefore the fault is clean, because it causes f_k to carry a 0 instead of a 1.

Only if: A single-vector test suffices to detect the spurious presence of an AND-crosspoint transistor at the intersection of $b_i(\bar{b}_i)$ and p_j. To stimulate the fault, the vector must have the form $X_{i1}(X_{i0})$. To guarantee observability of the fault at an output f_k, p_j must be the only product controlling f_k that covers $X_{i1}(X_{i0})$. The fault will cause p_j to be evaluated to 0 instead of **1**, causing f_k to carry a 0 instead of a 1.

(c) *If*: If f_k does not cover p_j, then there must exist a vector X which, in a fault-free PLA, is covered by p_j but not by f_k. But the spurious OR-crosspoint would cause $\bar{s}_k$ to be evaluated to 0 when X is input, causing f_k^* to be 1 instead of 0.

Only if: If the fault is clean, then there must exist a single-vector test X because the fault does not introduce memory beyond one computation cycle. If the fault is to be stimulated, then p_j must cover X. The signal on a product p_j that spuriously controls an output f_k, will be observed only if f_k does not cover X in the fault-free PLA. Thus, if the fault is clean, f_k cannot cover p_j. □

An empty AND(OR)-crosspoint location is *testable* iff the spurious presence of a AND(OR)-crosspoint transistor is clean. Note that an empty location is not testable iff a crosspoint transistor placed there would be redundant.

6.7. Summary

Despite all of the numerous conditions that appear in the testability properties proved above, many faults in any given dynamic CMOS PLA will be clean. This result is exploited in the work on dynamic PLA testing reported in [HaRe85,Ha87]. Still, as we have seen, some switch-level faults are conditionally or unconditionally unclean. The main problem areas include the following:

transmission gate circuitry: Most node faults and interconnect breaks, and all stuck-opens and stuck-ons are unclean in the transmission gates and associated clocking circuitry (Props. 3, 8, 9, and 12). The problems stem from the redundancy of having two parallel pass transistors: failures affecting only one path cause degradations in timing and signal strength that are difficult to detect.

stuck-on transistors: All stuck-ons affecting static inverter, precharge, and evaluation transistors are unclean (Props. 9 and 10). The problems stem from fact that it is impossible to place the circuit in a state that allows the effects of the fault to be observed without simultaneously creating a fight. However, stuck-ons affecting crosspoint transistors are usually clean (Prop. 11).

ohmic shorts: Many shorts that are likely to occur are unclean: e.g. shorts between adjacent input lines, bit lines, and output lines (Prop. 18). By placing these lines on two different layers in alternating fashion, ohmic shorts can be made much less probable. Ohmic shorts between precharged lines, such as the product and sumbar lines, are easier to detect because the precharge-evaluate timing sequence precludes sustained fights.

redundancy: Faults affecting redundant elements are often undetectable, and therefore unclean. By choosing the products carefully, many of these redundancies can be eliminated (Props. 5, 10, 11, 18, and 19, Th. 3). Removing a redundant crosspoint transistor creates an untestable empty crosspoint location, so the designer is presented with an unfortunate choice: the possibility of an unclean disappearance/growth fault versus the possibility of an unclean appearance/shrinkage fault. In custom integrated circuits it is unlikely that fully functioning crosspoint transistors will appear due to physical failures or manufacturing flaws, so the latter choice would be preferable.

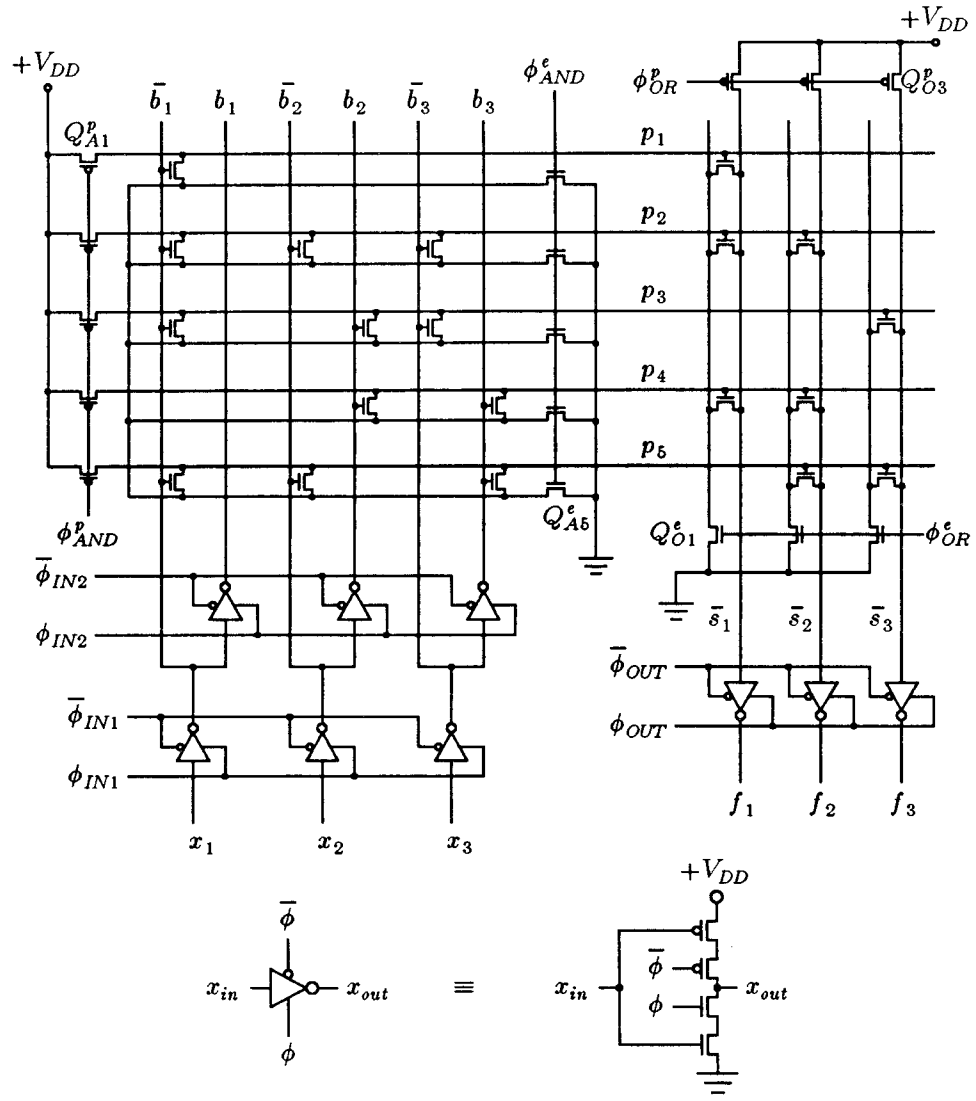
7. A Testable Dynamic CMOS PLA

7.1. Circuit Description

Figure 7 shows a circuit diagram of a modified dynamic CMOS NOR-NOR PLA that has better switch-level testability properties than the standard PLA. The modifications include the following:

(1) The input decoders and output buffers are implemented using clocked, rather than static, CMOS inverters [WeEs85]. The transmission gates are removed and their storage function is taken over by the clocked inverters. The clocked inverter circuit itself is given in the detail at the bottom of Figure 7. A one-bit wide slice of the modified input decoder is shown in Figure 8. Note that, for each input, four new *internal nodes* have been introduced, namely: m_1, n_1, m_2, and n_2.

(2) The number of general clocks is increased from 4 to 10. The two clocks ϕ_{AND} and ϕ_{OR} are split into two variants: those that are connected only to the gates of p-type precharge transistors (i.e. ϕ^p_{AND} and ϕ^p_{OR}), and those that are connected to n-type evaluation transistors (i.e. ϕ^e_{AND} and ϕ^e_{OR}). The input and output clocks must be modified to operate the new clocked inverters. Clock ϕ_{IN} is replaced by four



$$f_1 = x_1 + x_1x_2x_3 + \bar{x}_2\bar{x}_3$$
$$f_2 = x_1x_2x_3 + \bar{x}_2\bar{x}_3 + x_1x_2\bar{x}_3$$
$$f_3 = x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3$$

Figure 7 - Modified Dynamic CMOS NOR-NOR PLA Schematic

separately controlled variants: two non-inverted variants ϕ_{IN1} and ϕ_{IN2}, and two inverted variants $\overline{\phi}_{IN1}$ and $\overline{\phi}_{IN2}$. Clock ϕ_{OUT} is replaced by the independently controlled variants ϕ_{OUT} and $\overline{\phi}_{OUT}$. All 10 new clocks are controllable on a per input vector basis to be in either normal mode (i.e. *norm*) or in a test mode (i.e. held *low*(*high*) for a clock controlling n(p)-type transistors) as illustrated in Figure 9.

(3) The switched-ground lines in the AND plane are connected together forming one node g_A: product lines no longer have separate paths to *GND* via different evaluation transistors. This modification makes the multiple AND-plane evaluation transistors redundant, but they have been left in the circuit for performance considerations.

7.2. Testability Properties

The testability properties of the modified PLA are stated below in six theorems. The proofs have been omitted in most cases: they are straightforward and similar to those for the standard PLA.

Theorem 4: All node faults in the modified dynamic CMOS PLA are unconditionally or conditionally clean as follows:

(a) The following are unconditionally clean:

  (i) node faults affecting clocks ϕ^p_{AND}, ϕ^e_{AND}, ϕ^p_{OR}, ϕ^e_{OR}, ϕ_{OUT}, $\overline{\phi}_{OUT}$, all sumbars $\overline{s}_k$, all outputs f_k, all switched-ground lines g_A and g_{Ok}, and all internal nodes in the output inverters;

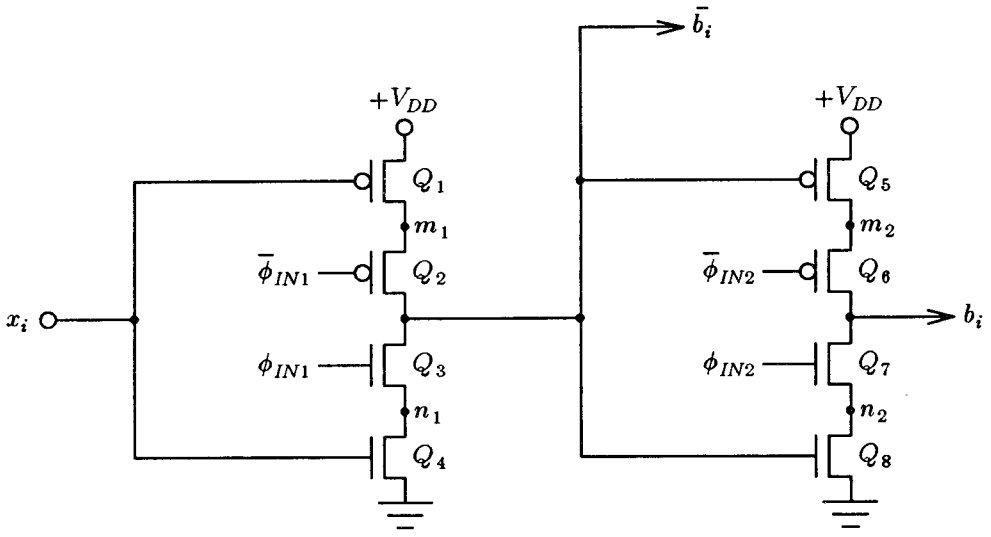
  (ii) a node-at-1 affecting a product p_j;

(b) The following are conditionally clean:

  (i) node-at-0 faults affecting clocks ϕ_{IN1} and $\overline{\phi}_{IN1}$ (ϕ_{IN2} and $\overline{\phi}_{IN2}$), provided that there exists an output f_k such that all products that control f_k contain at least one non-inverted (inverted) literal;

  (ii) node-at-1 faults affecting clocks ϕ_{IN1} and $\overline{\phi}_{IN1}$ (ϕ_{IN2} and $\overline{\phi}_{IN2}$), provided that there exists a controlling product composed only of non-inverted (inverted) literals;

  (iii) node faults affecting all inputs x_i, all bit lines b_i and $\overline{b}_i$, and all internal nodes in the input decoders, subject to conditions of irredundancy on the input lines, bit lines b_i, and product lines;



Note: ϕ_{IN1}, $\bar{\phi}_{IN1}$, ϕ_{IN2}, and $\bar{\phi}_{IN2}$ are independently controllable derivatives of a single base clock ϕ_{IN}.

Figure 8 - One Bit Slice of the Modified PLA Input Decoder

(iv) a node-at-0 affecting a product p_j iff p_j is irredundant.

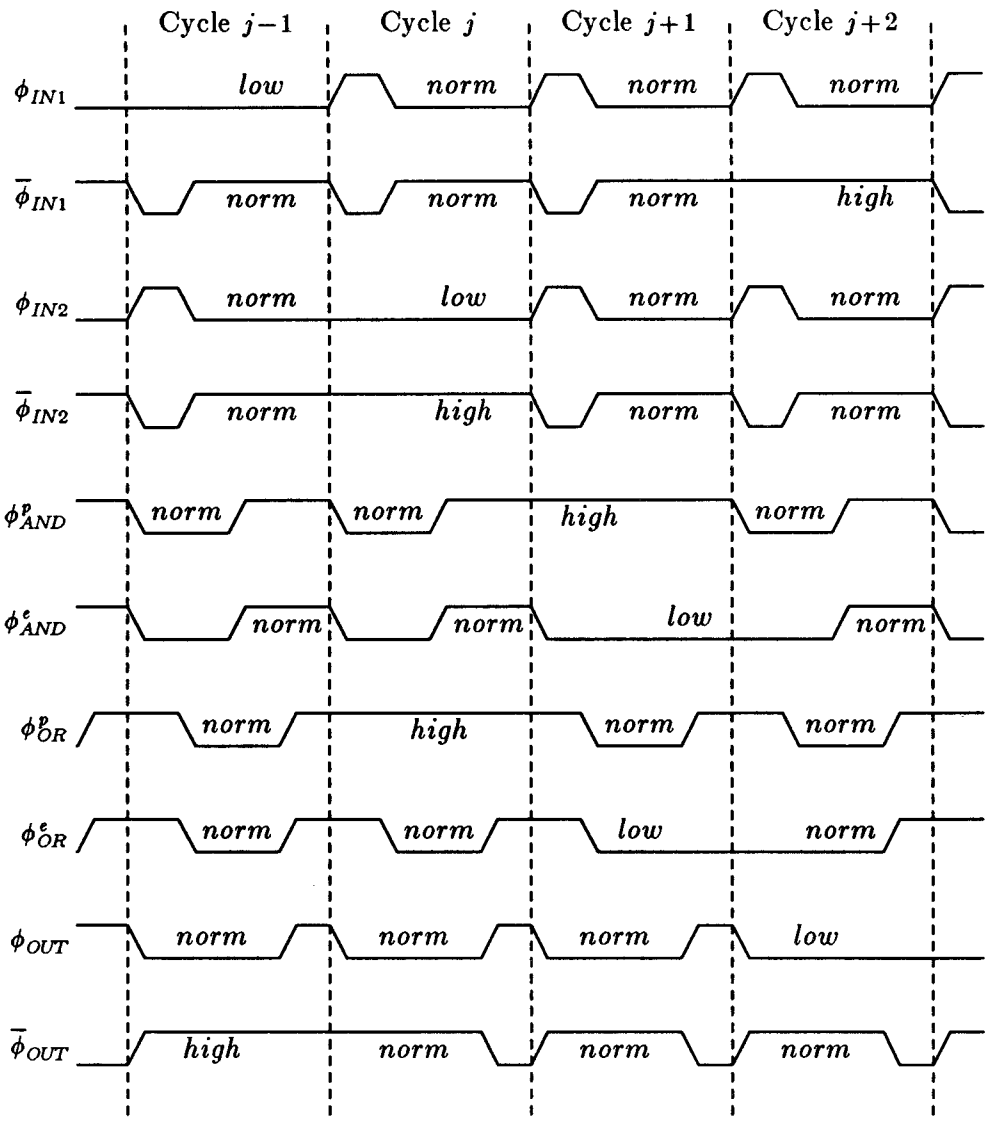
The next theorem gives the testability properties of the modified PLA with respect to single transistor stuck-opens and stuck-ons.

Theorem 5: Aside from stuck-opens affecting the deliberately redundant evaluation transistors in the AND plane, all transistor stuck-opens and stuck-ons are unconditionally or conditionally clean as follows:

(a) The following are unconditionally clean:

(i) stuck-opens affecting AND- and OR-plane precharge transistors, OR-plane evaluation transistors, and output inverter transistors;

(ii) stuck-ons affecting AND- and OR-plane precharge transistors, AND- and OR-plane evaluation transistors, OR-plane crosspoint transistors, and output inverter transistors;



Note: The waveforms are for the purposes of illustration and do not necessarily correspond to useful tests.

Figure 9 - Clock Waveforms for the Modified PLA

(b) The following are conditionally clean:

(i) stuck-opens affecting input decoder transistors and crosspoi transistors, subject to reasonable conditions of irredundancy the input lines, bit lines b_i, product lines, and crosspoint trans tors;

(ii) stuck-ons affecting input decoder transistors and AND-pla crosspoint transistors, subject to reasonable conditions of irredu dancy on the input lines, bit lines b_i, and product lines.

Stuck-opens affecting AND-plane evaluation transistors are uncle because of the deliberate redundancy in having multiple evaluatio transistors drive a single combined AND-plane switched-ground line. only one transistor were to be used then the corresponding stuck-op would be clean. The decision as to whether only one evaluation transist would be sufficient requires a detailed knowledge of the electrical prop ties of the particular combination of process and layout. The number product lines, in particular, would be an important parameter. In tl report we prefer to leave such a decision to the circuit designer.

The next theorem states the testability properties of the modifi PLA with respect to interconnect breaks.

Theorem 6: Most interconnect breaks in the modified PLA are uncon tionally clean, or clean subject to reasonable conditions of irredundanc and subject to the conditions in part (b-i) of Theorem 4.

Proof Outline: The topology of the modified PLA is very similar to th of the standard PLA, so most of the arguments underlying the proof Theorem 3 apply. Unlike in the case of the standard PLA, breaks th isolate the gates of p-type precharge and p-type inverter transistors not cause unclean faults: they are equivalent to stuck-ons affecting tho same transistors, faults that are clean by Theorem 5. Also, interconne breaks caused by transmission gates have been removed. Breaks affecti the four input clocks are equivalent to node-at-0 faults, and so will clean subject to the same conditions as those in part (b-i) of Theorem 4.

The next theorem shows the greatly increased testability of the mo fied PLA with respect to ohmic shorts.

Theorem 7: All of the following ohmic shorts in the modified dynam

CMOS PLA are clean or clean subject to reasonable conditions of irredundancy: (a) $\bar{b}_i$ to b_i, (b) p_j to p_{j+1}, (c) p_j to g_{Aj}, (d) $\bar{s}_k$ to $\bar{s}_{k+1}$, (e) $\bar{s}_k$ to g_{Ok} and (f) f_k to f_{k+1}.

Proof:

(a) If there exists an output f_k and an input vector X_{i0} such that f_k covers X_{i0} but not X_{i1}, then the following test detects the ohmic short (In the following tables a/b denotes the fault-free/faulty values):

	$\bar{\phi}_{IN2}$	ϕ_{IN2}	X	$\bar{b}_i$	b_i	f_k
X_1	*norm*	*norm*	X_{i1}	0/?	1/?	0/?
X_2	*high*	*norm*	X_{i1}	0/0	**1**/0	0/1

It is easy to show that if b_i is irredundant then there must exist a suitable vector X_{i1}. Also, if b_i irredundant but controls at least one irredundant product p_j, then a test will exist of the following second form: Let X_{i0} be an input vector which, of all the products that control an output f_k, is covered only by p_j.

	$\bar{\phi}_{IN2}$	ϕ_{IN2}	X	$\bar{b}_i$	b_i	f_k
X_1	*norm*	*norm*	X_{i0}	1/?	0/?	1/?
X_2	*norm*	*low*	X_{i0}	1/1	**0**/1	1/0

(b) When all the clocks are used in their normal mode, part (d) of Proposition 18 applies.

(c) Product line p_j will appear to be affected by a node-at-0, creating a disappearance fault. The fault is clean iff p_j is irredundant.

(d) The fault causes sumbars $\bar{s}_k$ and $\bar{s}_{k+1}$ to each implement a new sumbar formed by the union of the OR-crosspoint transistors controlling each fault-free sumbar alone. The truth table of f_k^* (and f_{k+1}^*) will be the logical OR of the truth tables of f_k and f_{k+1}. As long as f_k and f_{k+1} are not identical the fault is clean.

(e) Sumbar $\bar{s}_k$ will appear to be affected by a node-at-0, implying that output f_k will appear to be affected by a node-at-1. The fault is clear because all output functions are assumed to be non-trivial.

(f) As long as f_k and f_{k+1} are not identical, an input vector Y that produces complementary output values must exist. The following two-vector test detects the short:

	$\bar{\phi}_{OUT}$	ϕ_{OUT}	X	$f_k(f_{k+1})$	$f_{k+1}(f_k)$
X_1	*norm*	*norm*	Y	0/?	1/?
X_2	*norm*	*low*	Y	**0**/1	1/1

□

From typical PLA layouts, one would expect ohmic shorts between a non-inverting bit line b_i and the adjacent inverting bit line $\bar{b}_{i+1}$ to be probable. Unfortunately, the testability properties of this fault are complex. In some instances it will be possible to use the same clock strategy as that used in the two tests described for part (a) of Theorem 6 to observe the effects of the ohmic short as a node fault affecting b_i. A test corresponding to the first form would work if the input vector X_{i1} was such that x_{i+1} is 1; a test corresponding to the second form would work if the input vector X_{i0} was such that x_{i+1} is 0.

The next theorem gives the testability properties of the modified PLA with respect to crosspoint faults.

Theorem 8: Proposition 19 applies to the modified PLA.

Proof: The modifications to the standard PLA design do not worsen its testability properties with respect to crosspoint faults: the modified PLA can always be operated with all of its clocks in their normal mode. Thus, the results given in Proposition 19 continue to apply for the modified PLA. □

The next theorem summarizes the conditions that ensure that most of the common switch-level faults are clean in a modified dynamic CMOS PLA.

Theorem 9: All node faults, transistor stuck-opens (except those affecting the Q^s_{Aj}), transistor stuck-ons, most interconnect breaks, and most likely ohmic shorts are clean in a modified dynamic CMOS PLA that satisfies the following conditions:

(1) all input lines, bit lines, product lines, output lines, and crosspoint transistors are irredundant;

(2) no two output functions implemented on physically adjacent output lines have identical truth tables;

(3) at least one product, that controls an output, consists entirely of non-inverted literals;

(4) at least one product, that controls an output, consists entirely of inverted literals;

(5) there exists one output such that all of its controlling products contain at least one non-inverted literal;

(6) there exists one output such that all of its controlling products contain at least one inverted literal.

Proof: Follows from Theorems 4 to 8. □

8. Discussion

In this paper, the standard dynamic CMOS PLA was analyzed to determine its testability properties with respect to a comprehensive set of switch-level faults. That most switch-level faults were found to be clean should not be too surprising since the circuit has only two levels of logic, neglecting the input decoders and output buffers. The testability problems associated with transmission gate circuitry, stuck-on faults, and ohmic shorts will plague most CMOS designs. It was shown that the PLA's testability improves when the many different kinds of redundancy are eliminated, but this is usually a computationally expensive operation.

The modified PLA does have improved testability over the standard design, largely at the cost of increased clock complexity. If the overhead of generating the 10 clock variants can be shared among many large PLAs on a single IC then the modified design becomes more attractive. It should also be noted that the binary vectors controlling the clock modes can be encoded using only 5 bits since fewer than 32 different clock combinations are required to represent normal operation and all required test configurations. Most of the required clock mode combinations are given in Table 1. Thus it is possible for an external tester to control the clock modes on the fly using only 5 input pins. These pins could be used for other functions in the IC's normal operating mode.

The costs associated with meeting the conditions given in Theorem 9 are difficult to predict as they will vary with the particular PLA. Identifying redundant input lines, product lines, output lines, and crosspoint transistors in an existing PLA will undoubtedly require additional

	Clock Mode Combinations	Faults that Require Them
1.	all *norm*	crosspoint faults
2.	ϕ^p_{AND} *high*, ϕ^e_{AND} *low*	Q^p_{Aj} stuck-on
3.	ϕ^e_{AND} *low*	g_A with node-at-0
4.	ϕ^p_{OR} *high*, ϕ^e_{OR} *low*	Q^p_{Ok} stuck-on
5.	ϕ^e_{OR} *low*	Q^e_{Ok} stuck-on
6.	ϕ_{OUT} *low*	ϕ_{OUT} with node-at-1
7.	$\overline{\phi}_{OUT}$ *high*	$\overline{\phi}_{OUT}$ with node-at-0
8.	$\overline{\phi}_{IN2}$ *high*, ϕ_{IN2} *low*	ϕ_{IN1} with node-at-0
9.	$\overline{\phi}_{IN1}$ *high*, $\overline{\phi}_{IN2}$ *high*, ϕ_{IN2} *low*	$\overline{\phi}_{IN1}$ with node-at-0
10.	ϕ_{IN1} *low*, $\overline{\phi}_{IN2}$ *high*, ϕ_{IN2} *low*	ϕ_{IN2} with node-at-1
11.	ϕ_{IN1} *low*	node fault affecting an internal node in the first level of an input decoder
12.	ϕ_{IN2} *low*	node fault affecting an internal node in the second level of an input decoder
13.	$\overline{\phi}_{IN2}$ *high*, ϕ_{IN2} *low*, ϕ^e_{AND} *low*	Q^e_{Aj} stuck-on
14.	$\overline{\phi}_{IN2}$ *high*	ohmic short from b_i to $\overline{b}_i$
15.	ϕ_{IN2} *low*	ohmic short from b_i to $\overline{b}_i$
16.	$\overline{\phi}_{IN1}$ *high*	stuck-on affecting a clocked transistor in the first level of the input decoder

Table 1 - Clock Mode Combinations Used for the Modified PLA

computation. Some redundancies should only occur rarely: e.g. unconnected bit lines and duplicated output functions should not arise in most designs. Detecting and removing redundant crosspoint devices can be expected to be a difficult problem as it involves logic minimization. Meeting conditions (3), (4), (5), and (6) would, in the worst case, require the addition of two extra products and two extra outputs; however, these conditions are weak enough that few PLAs should incur the maximum cost.

9. References

[AbFu86] Abraham, J.A. and Fuchs, W.K., "Fault and Error Models for VLSI", *Proceedings of the IEEE*, v.74, no.5, May 1986, pp.639-654.

[BrSh85] Bryant, R.E. and Shuster, M.D., "Performance evaluation of FMOSSIM, a concurrent switch-level fault simulator", *Proceedings of the 1985 Design Automation Conference*, pp.715-719.

[Brzo87] Brzozowski, J.A., "Testability of Combinational Networks of CMOS Cells", pp.315-357 in *Developments in Integrated Circuit Testing*, D.M. Miller, ed., (Academic Press, New York, 1987).

[FlMa75] Fleisher, H. and Maissel, L.I., "An Introduction to Array Logic", *IBM Journal of Research and Development*, v.19, no.2, Mar. 1975, pp.98-109.

[GaCrVe80] Galiay, J., Crouzet, Y., Vergniault, M., "Physical Versus Logical Fault Model in MOS LSI Circuits: Impact on Their Testability", *IEEE Transactions on Computers*, v.C-20, no.6, June 1980, pp.531-537.

[Ha87] Ha, D.S., "On the Testable Design and Self-Test of PLAs", Ph.D. thesis, (Dept. of Electrical and Computer Engineering, Graduate College, The University of Iowa, Iowa City, Iowa, U.S.A., 1987).

[HaRe85] Ha, D.S. and Reddy, S.M., "On the Design of Testable Domino PLAs", *Proceedings of the 1985 International Test Conference*, pp.567-573.

[HaRe87] Ha, D.S. and Reddy, S.M., "On BIST PLAs", *Proceedings of the 1987 International Test Conference*, pp.342-351.

[Haye84] Hayes, J.P., "Fault Modeling for Digital MOS Integrated Circuits", *IEEE Transactions on Computer-Aided Design*, v.CAD-3, no.3, July 1984, pp.200-208.

[LeGaEl84] Leask, J., Gaboury, P., Elmasry, M.I., "PLAmate - A CMOS PLA/FSM Compiler - User's Guide", Department of Electrical Engineering, (University of Waterloo, Waterloo, Ont., 1984).

[MeCo80] Mead, C. and Conway, L., "Introduction to VLSI Systems", (Addison-Wesley, Reading MA, U.S.A., 1980).

[Mei74] Mei, K.C.Y., "Bridging and Stuck-At Faults", *IEEE Transactions on Computers*, v.C-23, no.7, July 1974, pp.720-727.

[OsHo79] Ostapko, D.L. and Hong, S.J., "Fault Analysis and Test Generation for Programmable Logic Arrays", *IEEE Transactions on Computers*, v.C-28, no.9, Sept. 1979, pp.617-627.

[ReMe81] Rem, M. and Mead, C., "A Notation for Designing Restoring Logic Circuitry in CMOS", *Proceedings of the 2nd Caltech Conference on VLSI*, (California Institute of Technology, Pasadena, CA, U.S.A., 1981).

[Smit79] Smith, J.E., "Detection of Faults in Programmable Logic Arrays", *IEEE Transactions on Computers*, v.C-28, no.11, Nov. 1979, pp 845-853.

[SoGa86] Somenzi, F. and Gai, S., "Fault Detection in Programmable Logic Arrays", *Proceedings of the IEEE*, v.74, no.5, May 1986, pp.655-668.

[Wads78] Wadsack, R.L., "Fault Modelling and Logic Simulation of CMOS and MOS Integrated Circuits", *Bell Sys. Tech. J.*, v.57, no.4, May-June 1978, pp.1449-1474.

[WeEs85] Weste, N. and Eshraghian, K., "Principles of CMOS VLSI Design", (Addison-Wesley, Reading MA, U.S.A., 1985).