ON TESTABILITY OF CMOS-DOMINO LOGIC

Vojin G. Oklobdzija

IBM T.J.Watson Research Center
P.O.Box 218
Yorktown Heights, NY 10598

Predrag G. Kovijanic

Productivity Through Automation Inc.,
Concord, Ma. 01742

ABSTRACT

Testability of CMOS-Domino logic is considered in this analysis. It is found that even though CMOS-Domino can exhibit stuck-at open type of fault, its testability is better than ordinary CMOS.

Moreover we prove that testing of CMOS-Domino logic is compatible with the LSSD design for testability methodology, which is not the case with the ordinary CMOS logic.

INTRODUCTION

CMOS-Domino logic and its successor Cascode Voltage Switch (CVS) Logic, has created a substantial interest due to its n-MOS performance and CMOS power consumption, yet with a density not exceeding that of static n-MOS (1),(11). However, reliability and testability of Domino circuits are not well known. In this analysis, testability of single ended CVS (SCVS) and CMOS-Domino circuits is investigated assuming some common type of defects that can occur in the manufacturing process. Special concern is given to the stuck-open type of faults, which are known to represent a problem in CMOS technology (2),(3),(4). It is in our interest to find out how serious this problem is in the designs using CMOS-Domino and what procedures should be taken to test CMOS-Domino logic.

CMOS-DOMINO CIRCUIT OPERATION

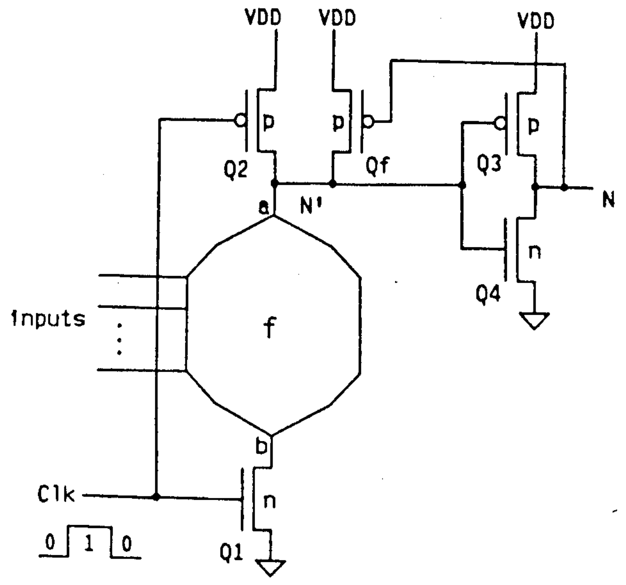
CMOS-Domino circuit is in function very similar to the dynamic n-MOS with the precharge. All the nodes N' are precharged to the high level during the precharge phase (Clk=0) through the transistor Q_2, and during the evaluation phase (Clk=1) the node N' is either discharged through the transistor Q_1 and network f, or it remains high (Fig.1.).

Node N' is followed by the CMOS inverter with the output N, which fans-out to the inputs of the functional part f of other circuits. Therefore, during the precharge phase all the outputs N (and corresponding inputs to f) are on the low logic level.

Feedback transistor Q_f, which is used in CVS Logic, helps to keep the node N' at the high level after the clock changes from the "precharge" to the "evaluation" stage (Clk=1). This assures stability of the operation and improves the circuit sensitivity to "glitches".

During the evaluation phase, if the function f is "true" the path between a and b is created and the node N' is discharged through f and Q_2. In this case Q_3 turns "on" and Q_4 "off" so that the node N makes a transition from low to high logic level. This transition will take place only if f becomes "true" in the evaluation phase. Otherwise node N will remain at the "low" level.

It should be noted that in the Domino logic this transition is always from low to high, and it is rippled through the logic from the inputs toward the output.



Single SCVS Tree

Fig.1.

If a 1 is propagated from the input toward the output, the effect of 1 (high level) will be propagated all the way to the output. If, however, a 0 is to be propagated, nothing changes after the precharge phase and everything remains on the low level (Domino is noninverting logic). Therefore the speed of Domino logic is determined by the time the effect of the 1 takes to travel through the critical path.

The understanding of how Domino logic functions should be facilitated by the Fig.2. The circuit is symbolically represented by its functional part **f** only, while the circuit associated with the clock is omitted.

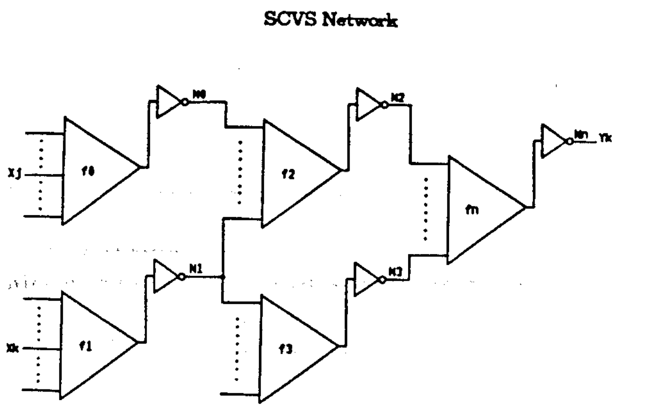


Fig.2.

FAULT CLASSES

A large number of physical defects occurring during the process of manufacturing can be covered by two classes of faults :

(a) faults that cause the transistor to be open permanently - "transistor permanently open"

(b) faults that cause the transistor to conduct permanently - "transistor permanently closed"

Class (a) covers faults such as:

missing or open contacts on the source, gate, and drain of the transistor

break in the metal polysilicon or diffusion line that connects the terminal node of the transistor

threshold voltage shift V_T which makes the device permanently open

Class (b) covers :

channel punch-through which connects source and drain of the transistor

oxide breakdown resulting in the transistor gate being shorted to drain (n-type) or source (p-type)

shifts in the threshold voltage V_T causing the transistor to conduct permanently

Defects that change the circuit configuration (bridging) or do not result in the node being permanently stuck at some logic value are not covered. Also, we assume that only a single fault can exist at the time t.

Past experience shows that a large number of multiple faults are also covered by the tests generated on these assumptions. The assumption of multiple stuck-at faults greatly complicates test generation and is generally avoided.

TRANSISTOR FAULTS

Let us consider the effect of class (b) faults on the transistors Q_1, Q_2, Q_3 , Q_4.

The effect of a transistor permanently conducting depends on the resistance ratio between p-transistor resistance R_p (Q_2,Q_3,Q_f) and n-type resistance R_n (Q_4 and pull-down network) :

1. p-transistor (Q_2,Q_3,Q_f) is "dominant" $R_p << R_n$
2. n-transistor (Q_4, pull-down network) is "dominant" $R_n << R_p$

Being "dominant" means that if both p-transistors and n-transistor (network) are conducting, the voltage on the node (N',N) will be in the range of the logic values: 1 for p-dominant and 0 for n-dominant.

In the Case 1.,the effect of class (b) fault on either Q_2 , Q_3 results in the node N being permanently stuck-at (0 for Q_2 'faulty, 1 for Q_3 faulty). This fault is detectable.

In Case 2.,the p-transistor acts as a load similar in function to the depletion transistor in n-MOS technology. The circuit will operate in a quasi n-MOS mode. If the time of operation is not considerably degraded by this type of fault, and the fault can not be sensed by the current monitors, this type of fault will never be detected. A circuit will operate "normally" in spite of the presence of the (b) type fault.

Fault (b) on Q_4 will result in N stuck-at-0 in Case 2., and will be detected. In Case 1. fault (b) might go undetected for the reasons described.

The effect of transistor Q_1 having a type (b) fault might also be undetected because all nodes N_i are 0 during the "precharge" phase which makes f=0 and disconnects the n-path. This is a particularly dangerous fault because, due to the different times needed for signals to reach the f-tree, the circuit might behave erratically having node N' not charged to the full value of 1 during the precharge phase.

We suggest designing the circuit such that it will be p-dominant; this will be assumed in the model used for test generation.

STUCK-OPEN FAULTS IN CMOS-Domino LOGIC

CMOS-Domino logic (Fig.1. and Fig.2.) is naturally partitioned into nMOS functional switching blocks, f, and the corresponding CMOS partitions consisting of the p-type pull-up transistors Q_2 and Q_3 and the n-type pull-down transistors Q_1 and Q_4.

CMOS logic gates are intrinsically tri-state devices due to their characteristic of complementary pull-up and pull-down networks.

Let us consider a single stage of a CMOS-Domino logic given in Fig.3.

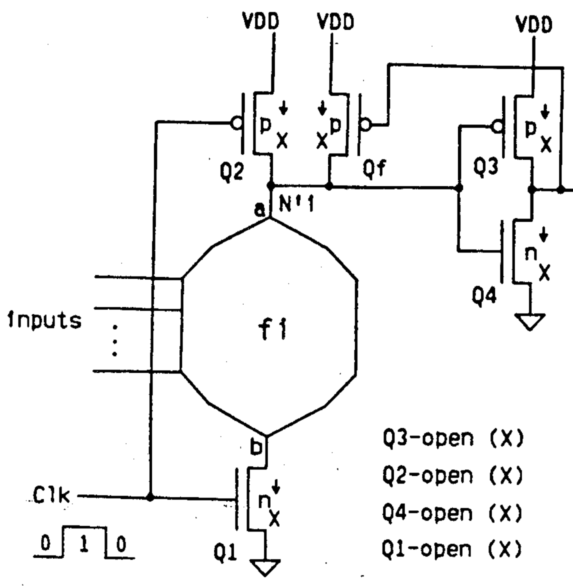


Fig.3.

On the output of the circuit, without faults, the only two steady-state values are 0 (the pull-down network is conducting and the pull-up network is nonconducting) and 1 (the pull-up network is conducting and the pull-down network is nonconducting).

There are four stuck-open faults that may occur in a single stage of CMOS-Domino logic.

SOP1 fault

The presence of a Q_3-open fault will prevent the pull-up network from conducting when the pull-down network is in a nonconducting state. Due to capacitance load present at the output, the output remains at the logic value of the previous output state, However, the length of time the state is retained depends on the leakage current at the node. This type of fault may transform a combinational network into a sequential network; it has been defined as a stuck-open (SOP) fault (4). The Q_3-open fault in Fig.3. has the same effect as disconnecting V_{dd} power supply; it will be identified as SOP1.

SOP2 fault

The presence of SOP2 fault, Q_2-open, will disconnect the path to V_{dd} and disable precharging activity of the transistor Q_2. Thus, if one of the switches Q_1 or **f** are open, N'_i will switch into the high-impedance state during the precharge phase. N'_i will retain the previous N'_i state value.

SOP3 fault

Another SOP fault caused by Q_4-open will prevent the pull-down network from conducting when a pull-up network is in a nonconducting state, creating a high-impedance state. This has the same effect of disconnecting input to the power supply V_{ss} ; it has been defined as SOP3.

CMOS-Domino logic can produce a high-impedance state on the output of the function **f** of each stage of a CMOS-Domino logic in the presence of SOP3 fault (see Fig.3.).

The switching tree **f** acts as either a closed switch (f=1) or an open switch (f=0). During the precharge phase (Clk=0) the node N'_i is pulled-up to the logic 1 level through the transistor Q_2. During the evaluation phase (Clk=1) the pull-down transistor switch Q_1 is closed and the node N'_i is pulled down to a logic level 0 if the nMOS switching tree **f** is closed (f=1) or the node N'_i remains at the logic level 1 if the nMOS switching tree **f** is open (f=0).

SOP4 fault

Similarly, SOP4 fault, Q_1-open, will disconnect the path to V_{ss} causing N'_i to come in the high-impedance state during the evaluation phase (Clk=1). N'_i will retain the previous N'_i state value (N'_i=1), but the value will decay with time due to discharging of the capacitance associated with the node by the leakage current.

TESTABILITY OF CMOS-Domino LOGIC

It has been shown (4) that stuck-open (SOP) faults cause a combinational logic to behave like a sequential logic in the presence of a SOP fault. This is directly contrary to the LSSD (12) and similar Design for Testability Methodologies which rely on the automatic test pattern generation algorithms (7,8) for the combinational logic networks.

We have identified four classes of SOP faults (SOP1, SOP2, SOP3, SOP4) at each stage of a CMOS-Domino logic.

We observe that CMOS partitions of CMOS-Domino logic are very regular, associated with clocks and the nMOS logical switching function blocks. Thus, we would like to explore the potential use of such a regular structure for testing of CMOS stuck-open faults in CMOS-Domino logic.

Our thesis is that even though CMOS-Domino logic can exhibit stuck-at open faults, its testability is better than ordinary CMOS. Moreover, we prove that testing of CMOS-Domino logic is compatible with the LSSD Design for Testability Methodology.

Let us develop a method for fault detection of SOP faults in each stage of a CMOS-Domino logic. A typical CMOS-Domino logic consisting of several stages performing functions $f_0 f_1 ... f_n$ is given in Fig.2. Note that the only observable output is the output of the last stage, the one connected to the primary output pin (or LSSD register). Each SOP fault prevents a pull-up or pull-down activity on the corresponding node (N' or N), leaving the node in its previous state. Although such a state might decay with time, a testing method independent of the decay time is desirable. Thus, in order to detect an SOP fault, the corresponding node (N' or N) should be in the opposite state of the supposed state to which the node should be pulled. We should be helped by the fact that the "precharge" phase (if functioning) always sets $N'=1$ and N=0.

Let us consider a model for the two consecutive stages of a CMOS-domino logic given in Fig.4. Application of the D-Algorithm will be assumed.

Singular cover and D-cubes for the MOS-output gate N included in Fig.5. are reproduced from (10).

Singular cover is used in the D-algorithm to determine the outputs during forward implication and the inputs during the consistency state (0 or 1) of the output line.
A D-cube is produced by combining two rows of the truth table. D stands for D=1 for a fault free line and D=0 for a faulty line. $\bar{D}$ is D's complement. The values in parenthesis give the required previous state at the output, for the particular D-cube to be valid. D-cubes are used to represent

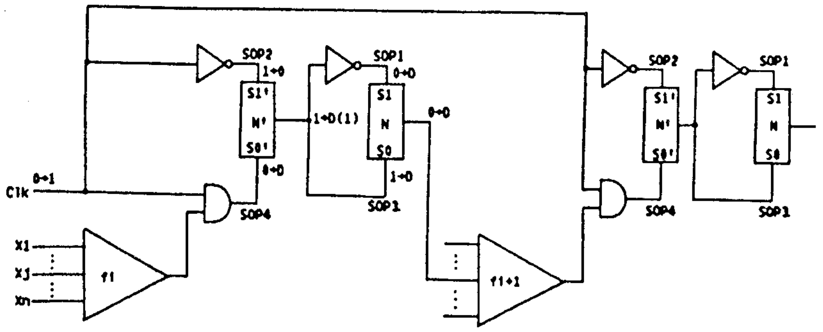


Fig.4.

the fault at the faulty line and to propagate the D, during the D-drive.

Stuck open-faults SOP1, SOP2, SOP3, and SOP4 will effectively force a 0 on the corresponding input to the integrator block N as indicated in Fig.4.

A MODEL FOR TEST GENERATION

Let us adopt a logic model for MOS transistor structures in terms of conventional logic gates and one additional modeling block, N, referred to as block "B" or an integrator in references (9) and (10).

MOS to logic transformation will be done corresponding to Table I in (10).

Since a switching tree f_i acts as either a closed switch ($f_i = 1$) or an open switch ($f_i = 0$), with a single output and many inputs, we will represent a block of the switching tree logic gates as a triangle with a single output and many inputs.

D-CUBES

S_1	S_0	OUTPUT
1	D	$\bar{D}$
1	$\bar{D}$	D
D	$\bar{D}$	D
$\bar{D}$	D	$\bar{D}$
0	D	$\bar{D}$(1)
0	$\bar{D}$	D(1)
D	0	D(0)
$\bar{D}$	0	$\bar{D}$(0)
D	D	$\bar{D}$(1)
$\bar{D}$	$\bar{D}$	D(1)

SINGULAR COVER

S_1	S_0	OUTPUT
X	1	0
1	0	1
0	0	M

Fig.5.

DETECTION OF SOP FAULTS

A single sensitizing path between a primary input X_j and a primary output Y_k can be established using a conventional automatic test pattern generation (ATPG), for instance the D-Algorithm (7). Such an existing sensitizing path will be used for detection of SOP faults.

Manual application of the D-Algorithm for the detection of SOP faults SOP1, SOP2, SOP3 and SOP4 in the first stage of the CMOS-domino logic is given in Fig.6.
Providing the existence of a sensitizing path between the single stage output and an observable output pin during the evaluation phase, we can conclude the following:

1. Due to the presence of the precharge phase which precedes the evaluation phase, SOP1 and SOP4 faults will be detected whenever a test for f_i stuck-at-0 fault for the corresponding function f_i is applied. This may occur many times for a multi-input function. Thus, detection of SOP1 and SOP4 faults does not require

any overhead over conventional stuck-at fault detection.

2. SOP2 and SOP3 faults will be detected whenever a test for f_i stuck-at-0 fault for the corresponding function f_i is applied, followed by the test for the stuck-at-1 fault at f_i during the next evaluation phase. This implies that detection of SOP2 and SOP3 faults in CMOS-Domino logic requires an ordered sequence of the test vectors needed for detection of stuck-at faults. In general this implies some overhead over conventional stuck-at fault detection.

SOP1

Clk	f_i	S'_1	S'_0	N'	S_1	S_0	N
0	1	1	0	1	0	1	0
1	1	0	1	0	D	0	D(0)

SOP2

Clk	f_i	S'_1	S'_0	N'	S_1	S_0	N
0	1	D	0	D(0)			
1	1	0	1	0	1	0	1
0	0	D	0	D(0)	$\overline{D}$	D	D(1)
1	0	0	0	D(0)	$\overline{D}$	D	D(1)

SOP3

Clk	f_i	S'_1	S'_0	N'	S_1	S_0	N
0	1	1	0	1	0	D	$D\overline{(1)}$
1	1	0	1	0	1	0	1
0	0	1	0	1	0	D	$D\overline{(1)}$
1	0	0	0	1	0	D	$D\overline{(1)}$

SOP4

Clk	f_i	S'_1	S'_0	N'	S_1	S_0	N
0	1	1	0	1	0	1	0
1	1	0	D	$D\overline{(1)}$	D	$\overline{D}$	D

Fig.6.

Since D followed by $\overline{D}$ on a primary input of the single sensitizing path during two consecutive evaluation phases, detects all SOP faults on that path, overhead for detection of SOP faults can be minimized or completely eliminated by favoring D followed by $\overline{D}$ transition on primary input pins in the automatic test pattern generator.

COMPATIBILITY WITH LSSD DESIGN FOR TESTABILITY METHODOLOGY

We have shown that detection of SOP faults in CMOS domino is compatible with LSSD Design for Testability (DFT) methodology i.e. automatic test pattern generation for combinational logic.

Any test which detects a fault in n-MOS switching function f_i must propagate D or $\overline{D}$ at the single output of f_i. D on the output of f_i will propagate as D at N and, similarly, using propagation D cubes in Fig.4., it can be shown that a $\overline{D}$ at f_i propagates as $\overline{D}$ at N. Thus, a test for a fault in any f_i is compatible with LSSD DFT methodology.

The CMOS-Domino clocking scheme is compatible with the LSSD clocks (Fig.7.). There are four independent clocks: System Clock, Shift Clock, +B Clock and -B Clock. Domino logic is clocked with the -B clock, which is a complement of +B clock. However, -B clock is made independent of +B clock; during the scan-in/out process it remains in the evaluation phase (-B=1).

Detection of SOP2 and SOP3 requires a sequence of 1-0 at the sensitized input. Since LSSD requires scan-in/out of the test vectors, it might appear that in the process of scanning-in the second test vector, set-up sequence will be destroyed, and therefore use of Stable Shift Register Latches (SSRL) will be required. This would impose an overhead for L3 latches.

However, we show that this is not necessary by the following analysis:

Detection of SOP2 and SOP3 faults in essence verifies that the precharge phase works and that the effect of precharge is propagated to the node N by pulling it to ground through the transistor Q4. Therefore it is necessary to first discharge the node N' by applying the first test vector which does that by setting f=1 and then to check that precharge works by applying the second test vector which sensitizes the path to the primary output and sets f=0 (path from "a" to ground is open).

After the first test vector (set-up vector) is scanned-in, clock -B remains in the evaluation phase (-B=1) until the second test vector is scanned in.

Being in the evaluation phase during the scan-in of the second test vector means that the primary inputs will change many times. The effect of this change is that node N' in the tree under test for SOP2 and SOP3 might be discharged again, which in essence means that the set-up sequence is repeated. Other trees might be discharged as well. However, this does not invalidate the set-up sequence and in fact it makes it more current.

Then, after the scanning of the second test vector is finished, -B passes through one normal cycle 0-1 (precharge-evaluation) after which the output from the combinational logic is "locked" in the SRL's on the output. During this cycle the following occurs:

1. precharge is performed (Clk=0)
2. the effect of precharge on the tree under test is propagated to the primary output(s) (second level of SRLs)

Testing with this procedure assumes that the node N' is capable of retaining the charge for the length of time which is in the worst case equal to the time needed for scanning-in of the second test vector.

This analysis concludes that the sequence used for testing CMOS-Domino logic is not disturbed by the scan-in/out procedure of LSSD.

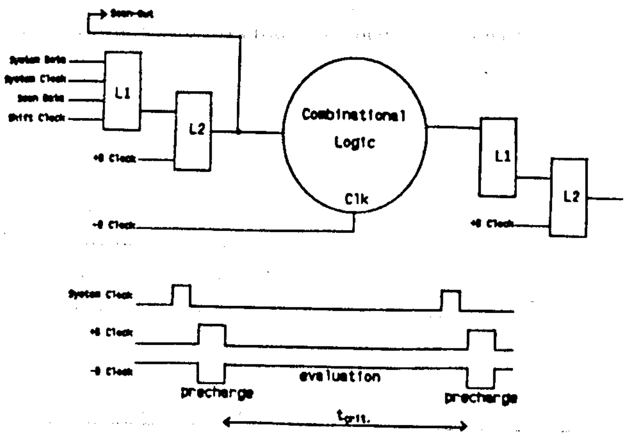


Fig.7.

We conclude that the test procedure for CMOS Domino logic is compatible with the LSSD DFT methodology. One limitation of CMOS-Domino logic is that all the gates are non-inverting. Complements of the variables can be obtained from the outputs of the SRLs only, which means that they are available only at the inputs to the next stage of the logic. However, as indicated in (1), very complex circuits can be implemented, including an ALU with two level of carry lookahead.

CONCLUSION

This study shows that CMOS-Domino technology can contain stuck-open faults which represent difficulty in test generation. However, the clock properties of this logic allow for easy verification of their existence with the relatively short test which also tests the clock circuitry and inverters.

The test for CMOS-Domino logic can be generated using ATPG from the gate level representation. The portion to be appended is short and is generated from the structural information of the logic with ATPG assistance.

It is concluded that CMOS-Domino logic is better than ordinary CMOS from the testability point of view.

REFERENCES

(1) R.H.Krambeck, et al, "High-Speed Compact Circuits with CMOS", IEEE Journal of Solid-State Circuits, Vol.SC-17,No.3, June 1982.

(2) Z.G.Vranesic, et al, "On Fault Detection in CMOS Logic Networks" 20th Design Automation Conference, Miami, Florida, June 1983.

(3) R.Chandramouli, "On Testing Stuck-Open Faults", Proceedings of 13th Fault-Tolerant Computing Symposium, Milano, June 1983.

(4) R.L.Wadsack, "Fault Modeling and Simulation of CMOS and MOS Integrated Circuits", The Bell System Technical Journal, Vol.57, No.5, pp.1449, June 1978.

(5) Y.M.El-Ziq, "Automatic Test Generation for Stuck-Open Faults in CMOS VLSI", 18th Design Automation Conference, pp.347, June 1981.

(6) S.M.Reddy, et al, "On Testable Design for CMOS Logic Circuits, Proceedings of the 1983 International Test Conference", p.435-445.

(7) J.P.Roth, "Diagnosis of Automata Failures: A Calculus and A Method", IBM Journal of Research and Development, Vol.10, p.278-291, July 1966.

(8) P.Goel and B.C.Rosales, "PODEM-X: An Automatic Test Generation System for VLSI Logic Structures", Proc. 18th Design Automation Conference, p.260-268, July 1981.

(9) Y.H.Levendel, R.P.Menon and C.E.Miller, "Accurate Logic Simulation Models for TTL Totempole and MOS Gates and Tristate Devices", Bell System Technical Journal, Vol. 60, September 1981, pp. 1271-1287.

(10) S.K.Jain, V.D.Agrawal, "Test Generation for MOS Circuits Using D-Algorithm", Proc. 20th Design Automation Conference, Miami 1983.

(11) L.G.Heller, W.R.Griffin, J.W.Davis, N.G.Thoma, "Cascode Voltage Switch Logic: A Differential CMOS Logic Family", Proc. of ISSCC, February 22-24, 1984, San Francisco.

(12) E.B.Eichelberger, T.W.Williams, "A Logic Design Structure for LSI Testability", Proc. 14th Design Automation Conference, June 1977, p.462-468.