Design of Compact and High Speed Totally Self- Checking CMOS Checkers for m-out-of-n Codes

X. Kavousianos, D. Nikolos & G. Sidiropoulos
Department of Computer Engineering and Informatics
University of Patras, 26500, Patras, Greece
&
Computer Technology Institute
Kolokotroni 3, Patra, Greece
e-mail : kabousia@ceid.upatras.gr, nikolosd@cti.gr

Abstract

This paper presents a novel method for designing TSC m-out-of-n code checkers taking into account a realistic fault model including stuck-at, transistor stuck-on, transistor stuck-open, resistive bridging faults and breaks. The proposed design method is the first method in the open literature that takes into account a realistic fault model and can be applied for all practical values of m and n. Apart from the above the proposed checkers are very compact and very fast. Another benefit of the proposed TSC checkers is that all faults are tested by single pattern tests thus the probability of achieving the TSC goal is greater than in checkers requiring two-pattern tests.

I. Introduction

A variety of error control codes has been proposed and many of them have been used to enhance the reliability of computer systems [1, 2, 3]. In order to exploit the capability of these codes, the concept of totally self checking checkers (TSC) [1, 4, 5] has been proved to be promising. These circuits can provide concurrent error detection and thus can detect transient, intermittent as well as permanent faults. Since transient faults have become increasingly dominant in VLSI circuits, providing protection against them has become very important. A circuit is a TSC checker if it is self-testing, fault-secure and code disjoint [1, 5].

***Definition* 1**. A circuit is *self-testing* for a set of faults F, if for every fault in F, the circuit produces a non-code output for at least one code input.

***Definition* 2**. A circuit is *fault-secure* for a set of faults F, if for every fault in F, the circuit never produces an incorrect code output for any code input.

***Definition* 3**. A circuit is *code-disjoint* if during fault free operation, code inputs map into code outputs and non-code inputs map into non-code outputs.

It has been shown that a large number of errors in VLSI circuits and compact laser disks are of unidirectional type [6, 7, 8]. This means that in any given data word the errors can be either $0\rightarrow1$ type or $1\rightarrow0$ type, but not both. Many codes have been developed to detect unidirectional errors, among the most known are the m-out-of-n codes [9].

Apart from the low redundancy (small number of check bits) of a code, its suitability for use in a computer system heavily depends also on the existence of a simple and fast TSC checker for this code. Unless the hardware needed to implement the checker is relatively simple compared with the hardware monitored, a fault-prone checker could increase rather than decrease the

1063-6722/97 $10.00 © 1997 IEEE

likelihood of erroneous information propagation.

The problem of designing TSC checkers for m-out-of-n codes or special classes of m-out-of-n codes as 1-out-of-n and m-out-of-2m codes under the assumption of the single stuck-at fault model has been extensively studied in the literature [10-25]. However, the conventional stuck-at fault model has been found to be inadequate for CMOS circuits [26]. CMOS is the current dominant technology for manufacturing VLSI circuits, thus new TSC checker designs are required that will take into account a more realistic fault model including apart from stuck-at, transistor stuck-open, transistor stuck-on and resistive bridging faults [26].

TSC CMOS checkers under single stuck-at and transistor stuck-open faults have been proposed in [27] for m-out-of-2m, m-out-of-2m+1, (m-1)-out-of-(2m-1) and (m+1)-out-of-(2m+1) codes and in [28] and [29] for m-out-of-2m codes. Also TSC CMOS checkers for a subset of m-out-of-2m codes under stuck-at, stuck-open, stuck-on, breaks and some bridging faults have been given in [30]. TSC checkers are used to achieve the Totally Self Checking goal (i.e. the first erroneous output of a functional block is signaled by the checker). The achievement of the TSC goal is based on two assumptions: a) faults occur one at a time, and b) there is a sufficient time interval between the occurrence of any two faults so that all required code inputs can be applied to the circuit. The stuck-open faults in the checkers proposed in [27-30] require two-pattern tests to be detected. The probability the checker to receive, during the normal operation, all the required two-pattern tests in a short period of time is much smaller than the probability to receive a test set of equal length consisting of single pattern tests. Therefore the checkers proposed in [27-30] have very small probability to achieve the TSC goal, which is the target of their use. Apart from the above drawback the checkers proposed in [27, 28] are fully unstructured and thus they are not suitable for VLSI implementation.

Recently PLA Self-Testing checkers for incomplete m-out-of-n codes and 1-out-of-n codes were proposed in [31]. Metra proposed in [32] a novel method for designing TSC 1-out-of-n code checkers under a realistic fault model including stuck-at, resistive bridging faults, breaks, transistor stuck-on and the majority of transistor stuck-open faults. It was shown that the TSC 1-out-of-n checkers proposed in [32] require impressively less area than the corresponding already known TSC checkers.

From the above it is evident that no method for designing TSC m-out-of-n code checkers, under realistic faults, has been proposed yet in the open literature. In this paper we give such a method.

The rest of the paper is organized as follows. In section II we give the design method while in section III we present the testability analysis. Discussion and comparisons are given in section IV.

II. Design method

The design of the proposed m-out-of-n code checker is based on the circuit of Figure 1, which in the sequel we will call "m-weight threshold circuit". When m or more inputs X_i of the m-weight threshold circuit are high then the output OUT is low else OUT is high. The circuit of Figure 1 is similar to the threshold function generator used in [35]. However a systematic method for designing such a circuit has not been given in [35]. A systematic method to design an m-weight threshold circuit has been given in [33]. We have proven in [33] that the transistor sizes of an m-weight threshold circuit must satisfy the following relation :

$$(m \quad 1)\frac{KP_n}{KP_p}\frac{2\ (V_{dd} \quad V_{tn})\ V_{IHMIN} \quad V_{IHMIN}^2}{(V_{dd} \quad V_{tp})^2}\frac{W}{L}$$
$$m\ \frac{KP_n}{KP_p}\frac{2\ (V_{dd} \quad V_{tn})\ V_{ILMAX} \quad V_{ILMAX}^2}{(V_{dd} \quad V_{tp})^2} \tag{1}$$

where $W = W_p/W_n$ and $L = L_p/L_n$, V_{tn} (V_{tp}) is the threshold voltages of the pmos (nmos) transistors, V_{IHMIN} (V_{ILMAX}) is the minimum HIGH (maximum LOW) input voltage which is recognized as logic 1 (0) from a driven gate and KP is the Spice parameter for $\mu \cdot Cox$.

The following relation gives all the possible values of m for which an m-threshold circuit (hence an (m-1)-out-of n checker) can be designed.

$$m\ \frac{1}{1\ (2(V_{dd} \quad V_{tn})V_{ILMAX} \quad V_{ILMAX}^2)/(2(V_{dd} \quad V_{tn})V_{IHMIN} \quad V_{IHMIN}^2)}$$

This relation is equivalent to the relation we have given in [33]. From the above relation we can see that the values of m depend not only on the values of the noise margins but also on the transition voltage of the driven circuit. Therefore, in order to achieve large values of m with acceptable noise margins we can append a transistor on each output of the m-out-of-n checker with suitably selected transition voltage, that is β_p/β_n.

Figure 2 presents an m/(m+1) programmable weight threshold circuit. When the input I is equal to one the circuit of Figure 2 behaves identically to the circuit of Figure 1, that is, it is an m-weight threshold circuit, while when the input I is equal to zero then the circuit of Figure 2 behaves as an (m+1)-weight threshold circuit. Relation (1) implies that the aspect ratios of the transistors pm_m, pm_{+1} and nm of the m/(m+1) programmable weight threshold circuit (Figure 2), which are $\frac{(W_{pm})_m}{(L_{pm})_m}$, $\frac{(W_{pm})_1}{(L_{pm})_1}$ and $\frac{W_{nm}}{L_{nm}}$ respectively, must satisfy the following relations:

$$(m \quad 1)\frac{KP_n}{KP_p}\frac{2\ (V_{dd} \quad V_{tn})\ V_{IHMIN} \quad V_{IHMIN}^2}{(V_{dd} \quad V_{tp})^2}\left(\frac{W}{L}\right)_m \quad m\frac{KP_n}{KP_p}\frac{2\ (V_{dd} \quad V_{tn})\ V_{ILMAX} \quad V_{ILMAX}^2}{(V_{dd} \quad V_{tp})^2} \tag{2}$$

and

$$m\frac{KP_n}{KP_p}\frac{2\ (V_{dd} \quad V_{tn})\ V_{IHMIN} \quad V_{IHMIN}^2}{(V_{dd} \quad V_{tp})^2}\left(\frac{W}{L}\right)_m \quad \left(\frac{W}{L}\right)_1 \quad (m+1)\frac{KP_n}{KP_p}\frac{2\ (V_{dd} \quad V_{tn})\ V_{ILMAX} \quad V_{ILMAX}^2}{(V_{dd} \quad V_{tp})^2} \tag{3}$$

where $\frac{W}{L}_m \quad \frac{(W_{pm})_m}{(L_{pm})_m}\frac{L_{nm}}{W_{nm}}$ and $\frac{W}{L}_1 \quad \frac{(W_{pm})_1}{(L_{pm})_1}\frac{L_{nm}}{W_{nm}}$

The proposed m-out-of-n checker is given in Figure 3. Module L_0 as well as module L_1 is a m/(m+1)-weight threshold circuit. We can easily see that for I=0 the module L_0 behaves like a m-threshold circuit and the module L_1 as a (m+1)-threshold circuit, while for I=1 we have the opposite. The operation of the circuit is described in Table I. W denotes the Hamming weight (number of ones) of the vector (X_1, X_2, ..., X_n). Input I is driven by clock signal with frequency equal to the half of the frequency of the system clock. The circuit of figure 3 has obviously the code disjoint property, because for each m-out-of-n encoded input, this produces the 2-rail encoded output 01 or 10 and for each non code word input this produces the non-2-rail encoded output 00 or 11 (see Table 1).

In this point we have to note that for m-out-of-n codes with $m \geq \lceil n/2 \rceil$ we can use a checker for the (n-m)-out-of-n code simply by inverting the outputs of the functional circuit.

III. Testability Analysis

In the following we prove that the proposed checkers are self-testing and fault secure for single stuck-at, transistor stuck-on and transistor stuck-open faults.

1. Faults affecting both modules L_0, L_1.
   Such faults are only stuck-at faults at the primary inputs $X_1, X_2, \ldots, X_n$ and line I.
   a. X_i stuck-at 0. When the checker receives a code word with $X_i=1$ then $Q_0=Q_1=1$.
   b. X_i stuck-at 1. When the checker receives a code word with $X_i=0$ then $Q_0=Q_1=0$.
   c. Line I stuck-at 0 or 1. These faults are detected with a checker for periodic signals [34].
2. Faults affecting only module L_0.
   a. Line Z_i stuck-at 0 or transistor nm_i stuck-open.
      When the checker receives a code word with $X_i=1$ and I=0 then $Q_0=Q_1=1$.
   b. Line Z_i stuck-at 1 or transistor nm_i stuck-on. When the checker receives a code word with $X_i=0$ and I=1 then $Q_0=Q_1=0$.
   c. Line I_2 stuck-at 0. When the checkers receives a code word and I=1 then $Q_0=Q_1=0$.
   d. Line I_2 stuck-at 1. When the checker receives a code word and I=0 then $Q_0=Q_1=1$.
   e. Line I_3 stuck-at 0 or transistor pm_{+1} stuck-on. When the checker receives a code word and I=0 then $Q_0=Q_1=1$.
   f. Line I_3 stuck-at 1 or transistor pm_{+1} stuck-open. When the checker receives a code word and I=1 then $Q_0=Q_1=0$.
   g. Line Q_0 stuck-at 0. When the checker receives a code word and I=1 then $Q_0=Q_1=0$.
   h. Line Q_0 stuck-at 1. When the checker receives a code word and I=0 then $Q_0=Q_1=1$.
   i. Transistor pm_m stuck-open. When the checker receives a code word and I=1 then $Q_0=Q_1=0$.
   j. Transistor nmos of the inverter INV stuck on. We construct the inverter with n-dominate logic so it is the same with 2d.
   k. Transistor pmos of the inverter INV stuck-on. This fault is undetectable, but does not affect the operation of the circuit, the circuit remains code disjoint. Furthermore, if this fault is followed by a detectable fault, the resulting fault is detectable.
   l. Transistor nmos of the inverter INV stuck-open. When the checker receives two successive code words with I=0 and 1 respectively, then the second code word will give $Q_0=Q_1=0$.
   m. Transistor pmos of the inverter INV stuck-open. When the checker receives two successive code words with I=1 and 0 respectively then at the second code word we have $Q_0=Q_1=1$.
3. Faults affecting only module L_1.
   a. Line Y_i stuck-at 0 or transistor nm_i stuck-open. When the checker receives a code word with $X_i=1$ and I=1 then $Q_0=Q_1$=high.
   b. Line Y_i stuck-at 1 or transistor nm_i stuck-on. When the checker receives a code word with $X_i=0$ and I=0 then $Q_0=Q_1$=low.
   c. Line I_1 stuck-at 0 or transistor pm_{+1} stuck-on. When the checkers receives a code word and I=1 then $Q_0=Q_1=1$.
   d. Line I_1 stuck-at 1 or transistor pm_{+1} stuck-open. When the checker receives a code word and I=0 then $Q_0=Q_1=0$.
   e. Line Q_1 stuck-at 0. When the checker receives a code word and I=0 then $Q_0=Q_1=1$.
   f. Line Q_1 stuck-at 1. When the checker receives a code word and I=1 then $Q_0=Q_1=0$.

g. Transistor pm_m stuck-open. When the checker receives a code word and I=0 then $Q_0=Q_1=0$.

The self-checking capability of the proposed designs with respect to resistive bridging faults and break faults on device terminals has been evaluated with extensive circuit-level simulations. Resistive bridging faults (RBFs) between two transistor terminals or between two inputs have been considered. All RBFs with connecting resistance $R \in [0,R_{max}]$ are detected, where R_{max} depends on the sizing of the transistors. For an implementation in 1μm technology with transistor aspect ratios $(W/L)pm_m=2/1$, $(W/L)pm_{m+1}=2/1$and $(W/L)nm_i=1/1$, for i=1 to n, the value of R_{max} for the various RBFs are given in Table 2. During the simulation the inputs of the checker are driven by standard cell inverters with aspect ratios (W/L)p=12 and (W/L)n=6.

The checker of figure 3 is Self Testing for all break faults on device terminals except a break on the gate terminal of the transistor pm_m in module L_0 and L_1. This break does not affect the operation of the circuit and the circuit remains code disjoint. Furthermore, if this fault is followed by a detectable fault, the resulting fault is detectable.

It is very easy to verify that for any of the above considered faults the output of the checker is the correct code output or a non-code word, therefore the checker is fault secure.

IV. Comparisons

This is the first method for designing TSC m-out-of-n code checkers for all practical values of m and n, that takes into account a realistic fault model.

The checkers proposed in [10-25] take into account only stack-at faults thus they are unsuitable for CMOS VLSI implementations. The PLA design given in [31] is valid for incomplete m-out-of-n codes and 1-out-of-n codes. Checkers only for m-out-of-2m codes taking into account apart from stuck-at faults, stuck-open faults too, were proposed in [27-29]. Their test set includes a large number of code input pairs, that increases with the value of m. For example for the 3-out-of-6 (6-out-of-12) code, the checkers given in [27], [28] and [29] require 103(7.364), 34(500) and 16(49) code input pairs respectively. The large number of code input pairs, as we have explained in the introduction, reduces significantly the probability the TSC goal to be achieved by these checkers.

Checkers for some m-out-of-2m codes under realistic faults were recently given in [30]. The checkers given in [30] have significantly greater area overhead and delay than the checkers proposed here. For example the 6-out-of-12 code checker given in [30] require 786 transistors and its delay is equal to 23.67 ns in 0.8μm implementation.

For the special case of 1-out-of-n codes, TSC checkers for realistic faults were proposed in [32]. As it is shown in Table 3 the checkers designed following the proposed method compare favorably to the checkers given in [32], achieving significant reductions with respect to area, delay and average power consumption.

V. Conclusion

In this paper we presented a new systematic method for designing TSC checkers for m-out-of-n codes including the 1-out-of-3 case. The checkers designed according to the proposed method have many benefits. They are TSC with respect to realistic faults: stuck at, transistor stuck-on, transistor stuck-open, resistive bridgings and break faults, the probability to achieve the TSC goal is greater than in other checkers, the design can be applied for any practical value of m and n and they are very compact and fast. The only undesirable characteristic of the

proposed checkers is that they exhibit static power consumption. We are currently working to the direction of reducing the static power consumption of these checkers.

References

[1] T. R. N. Rao and E. Fujiwara "Error control coding for computer systems", Englewood Cliffs, NJ: Prentice Hall, 1989.

[2] M. J. Iacopini and D. K. Vail "The fault tolerance approach of the advanced architecture on-board processor", in Dig. Papers 19th Int. FTCS, Chicago, IL, July 21-23, 1989, pp 6-12.

[3] T. J. Slegel and R. J. Veracca, "Design and Performance of the IBM enterprise system/9000 type 9121 vector facility", IBM J. Res. Develop., vol. 35, pp. 367-381, May 1991.

[4] W. C. Carter and P.R. Schneider, "Design of dynamically checked computers", IFIP Congress, vol. 2, pp. 878-883

[5] D. A. Anderson "Design of Self-Checking digital network, using coding techniques" Coordinated Sci. Lab. Univ. Illinois, Urbana-Champaign, Rep R-527, June 1984

[6] D. K. Pradhan and J. I. Stiffler, "Error correcting codes and self-checking circuits in fault tolerant computers" Computer, pp. 27-37, Mar. 1980.

[7] E. L. Leiss, "Data Integrity in Digital Optical Disks", IEEE Trans. Computers, vol 33, no. 9, pp. 818-827, Sept. 1984

[8] E. Fujiwara and D. K. Pradhan "Error -Control Coding in Computers", Computer, vol. 23, pp. 63-72, July 1990.

[9] C. V. Freiman, "Optimal Error Detection Codes for Completely Asymetric Binarry Channels", Information Control, vol. 5, pp. 64-71, March 1962.

[10] D. A. Anderson and G. Metze, "Design of totally self-checking check circuits for m-out-of-n codes" IEEE Trans. Computers, vol. 22, pp. 263-269, Mar. 1973.

[11] J. E. Smith, "The design of totally self-checking check circuits for a class of unordered codes", J. Design Automation Fault-Tolerant Computing, vol 2, pp. 321-342, Oct. 1977.

[12] M. A. Marouf and A. D. Friedman, "Efficient design of self-checking checkers for any m-out-of-n code" IEEE Trans. Computers, vol. 27, pp. 482-490, June 1978.

[13] N. Gaitanis and C. Halatsis, "A new design method for m-out-of-n TSC checkers," IEEE Trans. Computers, vol. 32, pp. 273-283, Mar. 1983.

[14] C. Efstathiou and C. Halatsis, "Modular realization of totally self-checking checkers for m-out-of-n codes", Proc. 13th FTCS, Milan, pp. 154-161, June 1983.

[15] S. Piestrak, "Design method of totally self-checking checkers for m-out-of-n codes", Proc. 13th FTCS, Milan, pp 162-168, June 1983.

[16] T. Nanya and Y. Tohma, "A 3-level realization of totally self-checking checkers for m-out-of-n codes", Proc. 13th FTCS, Milan, pp. 173-176, June 1983.

[17] C. Efstathiou and C. Halatsis, "Efficient modular design of m-out-of-2m TSC checkers, for $m=2^k-1$, $k>2$", Electronics Letters, vol. 21, pp. 1083-1084, Nov. 1985.

[18] A. Paschalis, D. Nikolos, and C. Halatsis, "Efficient modular design of TSC checkers for m-out-of-2m codes" IEEE Trans. Computers, vol. 37, pp. 301-309, Mar. 1988.

[19] S. Piestrak, The minimal test-set for sorting networks, and the use of sorting networks in self-testing checkers for unordered codes", Proc. 20th FTSC, pp. 457-464 June 1990.

[20] A. M. Pashalis, C. Efstathiou and C. Halatsis "An efficient TSC 1-out-of-3 code checker", IEEE Trans. 1990, C-39, pp. 407-411.

[21] J. Lo., and S. Thanawastien "On the design of combinational totally self-checking 1-out-of-3 code checkers" IEEE Trans. 1990, C-39, pp. 387-393

[22] T. N. Haniotakis, A. M. Pashalis and D. Nikolos "Fast and low-cost TSC checkers for 1-out-of-n and (n-1)-out-of-n codes in MOS transistor implementations" Int. J. Electronics, vol. 71, pp. 781-791, Nov. 1991.

[23] D. L. Tao, C. R. P. Hartmann and P. K. Lala " A general technique for designing totally self-checking checker for 1-out-of-n code with minimum gate delay" IEEE Trans. 1992, C-41, pp. 881-886

[24] V. V. Dimakopoulos, G. Sourtziotis, A. Paschalis and D. Nikolos, "On TSC Checkers for m-out-of-n Codes", IEEE Trans. Computers, vol. 44, No. 8, August 1995.

[25] S. J. Piestrak, "Modular Design of Self-Testing checkers for m-out-of-n Codes, 2nd IEEE Int. On-Line Workshop, Biarritz France, July 8-10, 1996.

[26] J. Shen, W. Maly and F. Ferguson "Iductive fault analysis of MOS integrated circuits, IEEE Design & Test, December 1985, pp. 26-33.

[27] S. Kundu and S. M. Reddy, "Design of TSC checkers for implementation in CMOS technology", Proc. Of Inter. Conference on Computer Design, pp. 116-119, 1989.

[28] S. Manjunath and D. Radhakrishnan, "Efficient design of CMOS TSC checkers", International Journal of Electronics, vol. 71, no. 1, pp. 67-79, 1991.

[29] Th. Haniotakis and A. Paschalis, "Efficient Structured Design of Robustly Testable CMOS TSC m-out-of-2m code checkers", 1st IEEE On-Line Testing Workshop, Nice, France, July 4-6, 1995, pp. 233-237.

[30] P. Lala, F. Busaba, M. Zhao, "Transistor-level Implementation of Totally Self-Checking Checkers for a Subset m out of 2m Codes", 2nd IEEE Int. On-Line Testing Workshop, July 8-10, 1996.

[31] S. J. Piestrak "Design of Minimal-Level PLA Self-Testing Checkers for m-out-of-n Codes" IEEE Trans. On VLSI Systems, vol. 4, No. 2, 1996, pp. 264-272

[32] C. Metra, M. Favalli and B. Ricco "Novel 1-out-of-n CMOS checker", Electronic Letters, 18 Aug. 1994, vol. 30, No 17 pp. 1398-1400.

[33] X. Kavousianos, D. Nikolos "Self-Exercising, Self-Testing k-order Comparators" 15th IEEE VLSI Test Symposium, Monterey, California, USA, April 27-May 1, 1997, pp. 216-221.

[34] A. M. Usas " A Totally Self-Checking Checker Design for the Detection of Errors in Periodic Signals", IEEE Trans. Comput. Vol. C-24 No. 5, May 1975, pp. 483-488

[35] C. Metra, J. C. Lo, "Compact and High Speed Berger Code Checker", 2nd IEEE Int. On-Line Testing Workshop, Biarritz, France July 8-10, 1996, pp. 144-149.

Table 1

I	Weight of Input Vector (W)	Output Q_0	Output Q_1
0	W<m	1	1
	W=m	0	1
	W>m	0	0
1	W<m	1	1
	W=m	1	0
	W>m	0	0

Table 2

Transistor	Drain-Source	Gate-Drain	Gate-source
nm_i	16k	16k	3.4k
pm_m	18k	16k	-
pm_{m+1}	18k	16k	3.8k
Bridging resistance between two inputs : 3k			

Table 3

Checker	Area Reduction	Delay Reduction	Average power consumption reduction
1-out-of-8	19%	58%	46%
1-out-of-16	25%	61%	41%

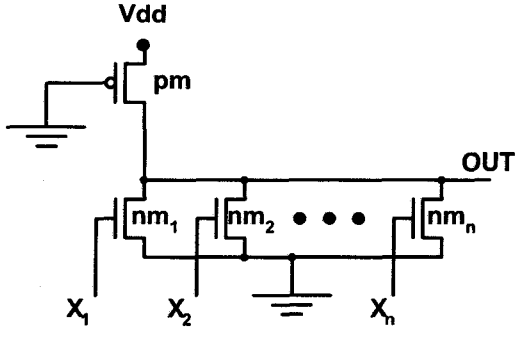

Figure 1. m-weight threshold circuit

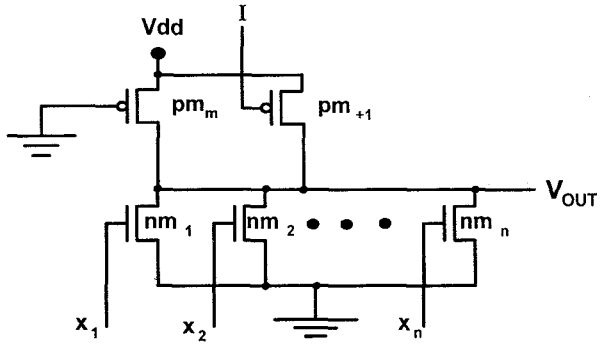

Figure 2 m/m+1 programmable weight threshold circuit

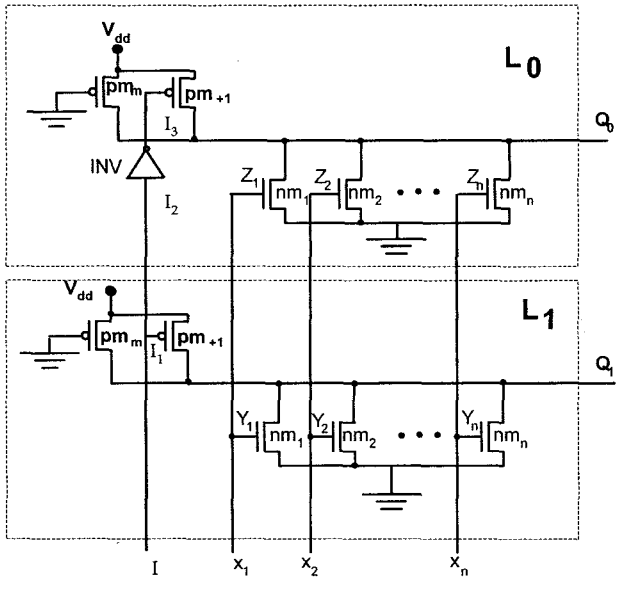

Figure 3. m-out-of-n checker