A Fail-Safe CMOS Logic Gate

V. Bobin and S. Whitaker
NASA Space Engineering Research Center
for VLSI System Design
University of Idaho
Moscow, Idaho 83843

Abstract- **This paper reports a design technique to make Complex CMOS Gates fail-safe for a class of faults. Two classes of faults are defined. The fail-safe design presented has limited fault-tolerance capability. Multiple faults are also covered.**

1 Introduction

All fault-tolerance techniques are based on fault models. Most of the work in fault-tolerance has been directed towards a fault-set which consists of the so-called *stuck-at* faults; *stuck-at 0* and *stuck-at 1* [1]. This fault model assumes that a line or net is stuck at the logical value 0 or 1 as a result of the failure. Traditionally, in the treatment of fault-detection and fault-tolerance, additional assumptions have been made, such as mutual independence of faults, identical probability for all possible faults, and that of a single fault in a given circuit. At higher levels of integration, all these assumptions become unacceptable. Many of the failures are the results of process characteristics and are technology- and layout-dependent. Thus all faults are not equally probable. In a given circuit, there may be some faults that are more likely than others. It is well known that all physical defects cannot be represented by the simple stuck-at model. A single defect at the physical level could result in multiple faults throughout the circuit especially when it affects a signal line feeding multiple points. Faults of structural origin may be better modeled as *shorts* or *opens* [1].

Redundancy at the gate level as well as careful coding of states have been used by many schemes to achieve logical fail-safeness [2,3]. The dominant integrated technology today is CMOS. In CMOS, two of the most likely faults are due to the *stuck-on* and *stuck-open* failures of individual transistors [4]. In this paper, the effects of these types of MOS transistor failures on the functioning of complex CMOS logic gates are considered. The respective faults are termed transistor stuck-on faults and transistor stuck-open faults. The effect of stuck-at faults on input signal lines is to make transistors either stuck-on or stuck-open. Thus stuck-on and stuck-open faults of the MOS transistors and stuck-at faults on signal lines make up the fault-set addressed by this paper. A design method which makes the logic gate fail-safe is suggested. The effects of specific classes of multiple faults are also taken into account. There are two specific faults that cannot be handled by the fail-safe design presented here. However, in general, this design is fail-safe for multiple faults also. It is seen from simulations that in several cases, the gate design presented here is fault-tolerant, but fault-tolerance capability depends on the sizes of the individual transistors and also on their configuration.

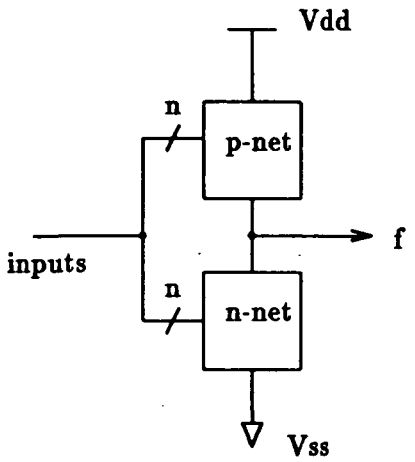

Figure 1: Complex CMOS Logic Gate

2 Complex CMOS Logic Gates

The general structure of complex CMOS logic gates is as shown in Figure 1. It consists of a network of pMOS transistors called the p-net or the pull-up network connected between the Vdd bus and the output node and a network of nMOS transistors called the n-net or the pull-down network connected between the Vss bus and the output node. The input signals are applied to both the pull-up and pull-down network. During normal operation only one path, either from Vdd to the output or from Vss to the output is enabled at any given time by the input combination applied [5]. Thus the output is pulled high to Vdd or pulled low to Vss leading to the logical operation. Since there is no completed path from Vdd to Vss during normal operation, there is no static power dissipation in the gate which is one of its desirable properties. There is however, dynamic power dissipation during the switching of the gate from one logic state to another because the capacitance at the output node must be charged to the proper logic level.

There are parasitic capacitances associated with each node in the network as well as the MOS transistor gate terminals where the inputs are applied. These capacitances cause delays during switching of the logic gates, which are proportional to the magnitude of the capacitances. The speed of operation of the gate is determined by the sizes or W/L ratios of the transistors, the configuration of the transistors, and the magnitude of the capacitances in the circuit. The low to high switching speed is governed by the size and configuration of the pMOS transistors in the pull-up network whereas the high to low switching speed depends on the size and configuration of the nMOS transistors in the pull-down network [5].

3 Fail-Safe Logic Design

Fail-safe logic gates are defined as follows [2,6].

Definition 1 *A logic gate is said to be 0-fail-safe (1-fail-safe) if any failure causes only incorrect 0 (1) output. An output 1 (0) is always correct.*

This means that upon failure, the 0-fail-safe (1-fail-safe) circuit assumes a 0 (1) output state. Thus an output of 0 (1) could be faulty or correct whereas an output of 1 (0) is always correct.

3.1 The Static Latch

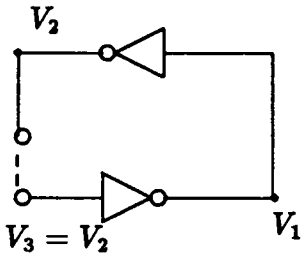

Figure 2: The Static Latch

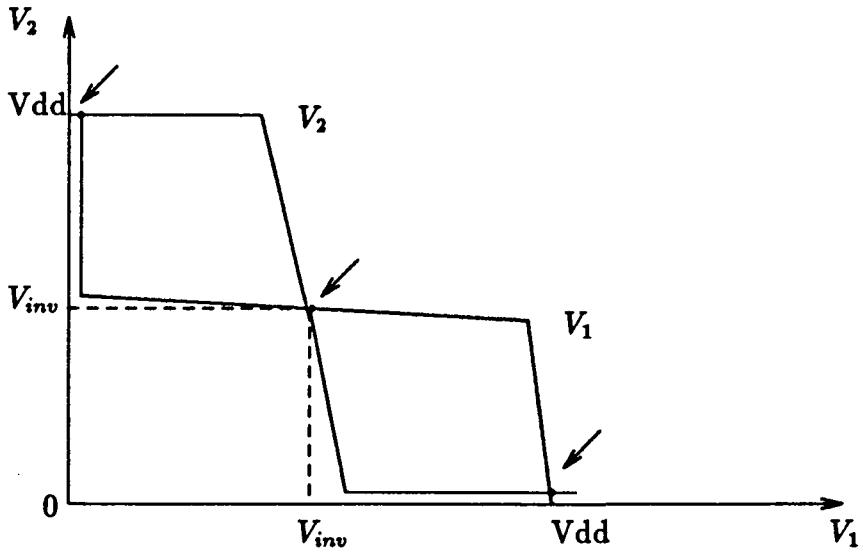

Figure 3: Three Static Solutions of Static Latch

A latch is used in a digital circuit to hold a logic state. The static latch remembers its last state until a new input is applied and as long as the power supply is on. The static latch contains elements with power gain that continually restore the integrity of the signals representing the state. The static latch is analogous to restoring logic [7]. The simplest form of static latch consists of a pair of cross-coupled inverters as shown in Figure 2.

The bistable latch has three quasi-static solutions as shown in Figure 3. In Figure 3, the solutions are found graphically from the individual transfer curves of the two inverters. As can be seen from the figure, the static solution in the middle is an unstable equilibrium

point, since a slight perturbation about this point pushes the latch to one of its stable points. If the two inverters of the latch are identical, then by symmetry, the unstable equilibrium point is where both the nodes are at the same voltage *Vinv*. From the transfer curves of the inverters it can be seen that unless the input voltage is very close to *Vinv*, the inverter output is a valid logic level, either high or low. This property is used in the fail-safe gate described next.

3.2 Fail-Safe Complex CMOS Gate

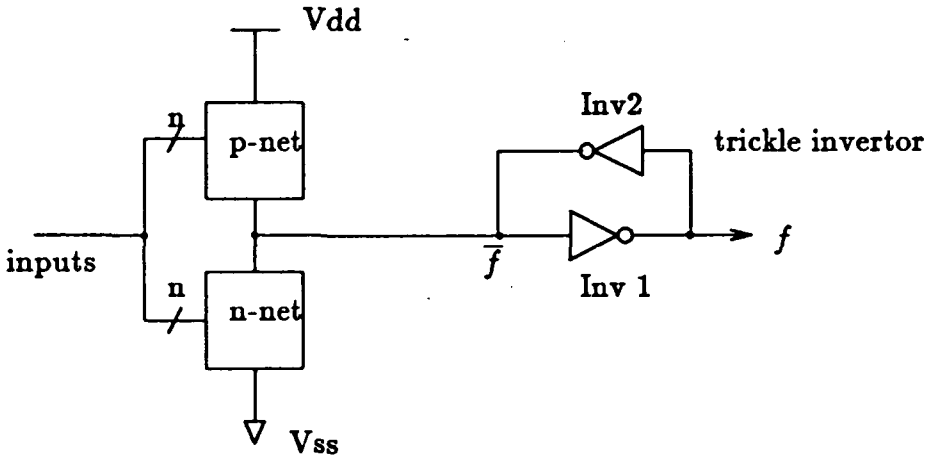

Figure 4: Fail-Safe Complex CMOS Gate

The fail-safe complex CMOS gate has the same general structure as the regular complex CMOS gate except for the addition of two inverters Inv1 and Inv2 connected back to back at the output node. These two inverters connected back to back constitute a static latch structure at the gate output.

Definition 2 *A Trickle Inverter is a CMOS inverter whose transistors have W/L ratios less than 1.*

The inverter Inv2 is a trickle inverter so that its output can be overdriven by the output of the p-net or the n-net. The gate output is determined solely by the input combination applied. This structure is shown in Figure 4. The purpose of the static latch is to provide the necessary positive feedback to make the gate output a definite logic level in the event of MOS transistor failures.

Definition 3 *A Safe Gate is a complex CMOS logic gate with two inverters connected back to back at its output node. The inverter whose output is connected to the output node of the regular complex CMOS structure is a trickle inverter. The input node of the trickle inverter is driven by the primary output node of the safe gate.*

A fail-safe complex CMOS gate is a *safe gate.*

Definition 4 *A Basic Gate is a safe gate with the static latch at the output removed; ie., it is the complex CMOS gate from which the safe gate is derived.*

Thus the basic gate is designed using the complement of the function $\bar{f}$ rather than the function f itself. If f is 0-fail-safe, $\bar{f}$ is 1-fail-safe and vice versa.

The circuit in Figure 5 is a fail-safe complex CMOS gate which implements the function shown. SPICE simulations were carried out on this circuit to study the effects of various faults on the logical operation of the gate. The SPICE simulations used parameters from a typical $2\mu m$ process to model the nMOS and pMOS transistors. As an example for the effect of a fault on the operation of the *safe gate*, assume that the nMOS transistor m10 has failed by being stuck-on solidly. This failure is modeled by turning the transistor permanently on. Its gate is kept at Vdd (5V) permanently. Consider the input transition from $[A,B,C] = [0,1,1]$ to $[A,B,C] = [0,1,0]$. The gate output f should change from a logic level of 1 to a logic level of 0. SPICE simulation of this transition shows that $\bar{f}$ changes from 0V to 2.93V, whereas f does indeed change from 5V to 0.1V. Thus in this case, the fail-safe gate has tolerated this stuck-on fault.

Fault-tolerance capability depends on the actual circuit configuration and the relative sizes of the transistors. It also depends on the extent of the failure; in other words, on the fault model. However, fail-safeness is independent of these. It is seen that f which is the output of Inv1 gives a better logic level compared to the output of the *basic gate*. This is true in general, because any failure in the *basic gate* can lead to a degradation of the logic level at the node $\bar{f}$. This is the reason why the output of Inv1 is designated the primary output of the fail-safe CMOS gate. It is clear that if the transistors of Inv1 fail, the output of the *safe gate* can be at an undefined logic level. Therefore in this discussion it is assumed that Inv1 is free of faults.

If one of the nMOS transistors in the fail-safe complex CMOS gate described here (excluding the one in Inv1) is stuck-on, the input to Inv1 can approach 0 erroneously, but never 1. This holds if several of the nMOS transistors are stuck-on also. The same is the case when several of the pMOS transistors (except the one in Inv1) are stuck-open or when both occur simultaneously, ie., pMOS transistors stuck-open and nMOS transistors stuck-on. These faults are also analogous to the input signal lines being stuck-at-1. In all these cases, the primary output of the fail-safe complex CMOS gate can be made 1 erroneously, but never 0.

In the fault-set addressed in this paper, there are six possible kinds of faults; nMOS transistor(s) stuck-on, nMOS transistor(s) stuck-open, pMOS transistor(s) stuck-on, pMOS transistor(s) stuck-open, gate input(s) stuck-at 1, and gate input(s) stuck-at 0. These elements of the set are grouped into two classes of faults as defined below.

Definition 5 *If nMOS transistors are stuck-on or pMOS transistors are stuck-open or both happen simultaneously, then* Class A *faults occur. This includes gate inputs* stuck-at 1.

Definition 6 *If pMOS transistors are stuck-on or nMOS transistors are stuck-open or both happen simultaneously, then* Class B *faults occur. This class also includes gate inputs* stuck-at 0.

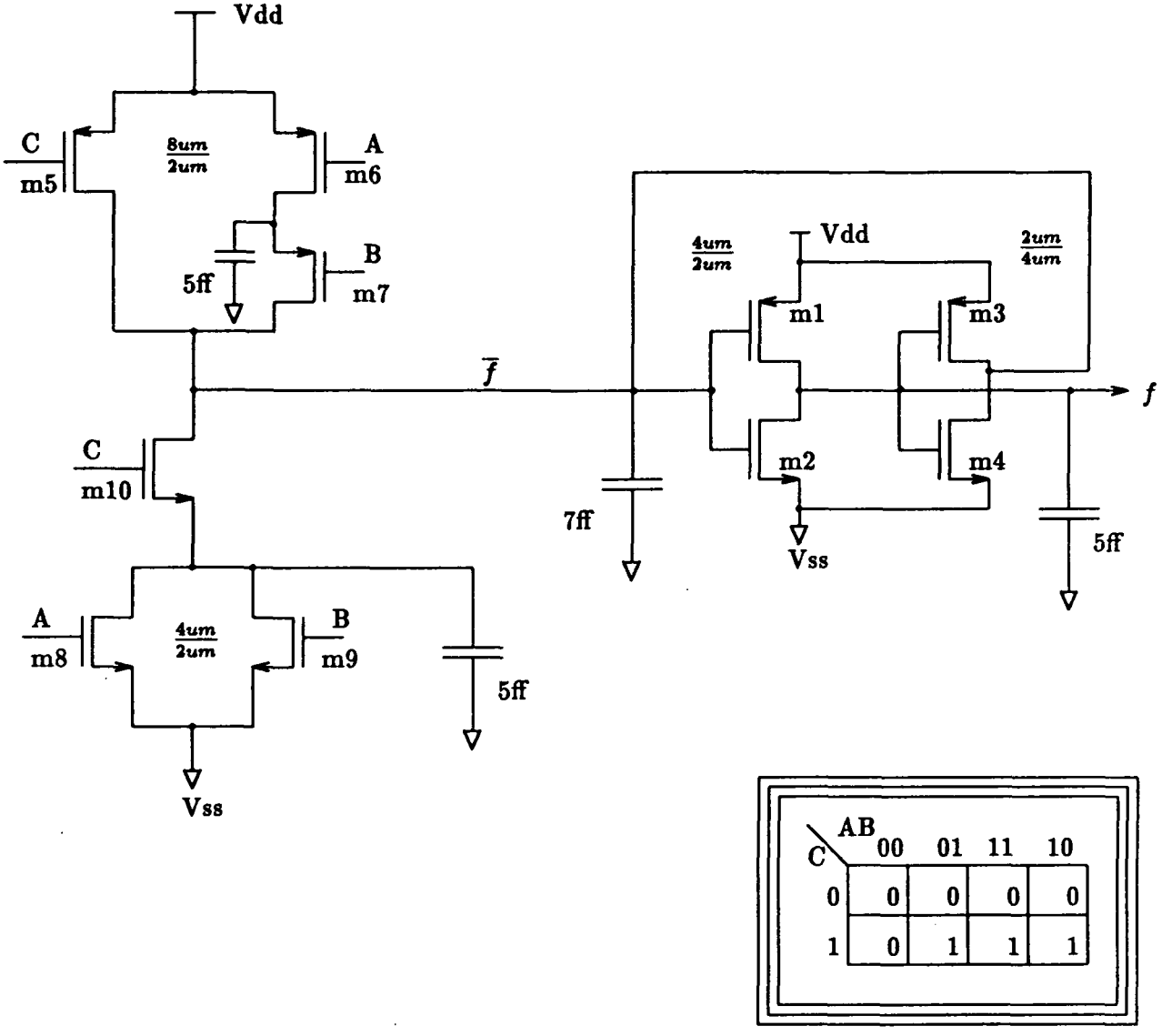

Figure 5: Fail-Safe Complex CMOS Gate Example

In the discussion on static CMOS latches, it was mentioned that the output of the static latch is a valid logic level except for a very small range of input voltage around its unstable equilibrium point. In the case of the fail-safe complex CMOS gate, the static latch structure consists of Inv1 and Inv2. Since Inv2 is a trickle inverter, the voltage level at the node $\bar{f}$ is determined mainly by the transistors in the *basic gate*. Therefore the voltage at the primary output (node f) of the fail-safe complex CMOS gate depends on the voltage at node $\bar{f}$ and the voltage transfer characteristic of Inv1. For the Inv1 of Figure 5 with Vdd=5V, SPICE simulation shows that the output is a valid 1 (greater than 4.1V) for an input that ranges from 0V to 1.85V and the output is a valid 0 (less than 0.9V) for an input range from 2.03V to 5V. Thus the output of the *safe gate* gives a valid logic level as long as the input node of Inv1 is not forced to a voltage between 1.85V and 2.03V due to failure. Since this is a small range of voltage, the probability of this occurring is small. The probability of an invalid logic level occurring at the output of the *safe gate* due to transistor failures depends on the sharpness of the voltage transfer characteristics of Inv1 and Inv2; the probability becomes smaller as the inverter characteristics become sharper.

The primary output of the fail-safe complex CMOS gate is a logic 1 or a logic 0 even in the presence of faults due to the positive feedback arrangement at the output. When *Class A* faults occur, the primary output of the gate can become 1 erroneously, but not 0. Thus for Class A faults, the *safe gate* is 1-fail-safe. It follows that for *Class B* faults, the *safe gate* is 0-fail-safe. Within a given class of faults, multiple faults are also covered; ie., the *safe gate* remains fail-safe even for multiple transistor faults as long as *Class A* and *Class B* faults do not occur simultaneously. This is because the gate is fail-safe for a given class of faults and this property is maintained irrespective of the number of faulty transistors.

When *Class A* and *Class B* faults occur simultaneously, the *safe gate* loses its fail-safe property. In the case where pMOS transistors and nMOS transistors are stuck-open at the same time, the primary gate output can behave in several different ways.

1. If both the n-net and the p-net are stuck-open in the basic gate, the primary output of the *safe gate* retains its previous value because of the feedback arrangement. The output need not be safe.

2. If both transistors in the inverter Inv1 are stuck-open, the primary output floats, but the output node of the basic gate is true. Thus the primary output is not safe since a floating output is not safe.

3. The primary output is not affected when both transistors of the trickle inverter Inv2 are stuck-open. As long as the p-net and the n-net in the basic gate are not stuck-open at the same time, the primary output is safe.

Besides the three possibilities discussed, there are other ways in which the primary gate output can behave depending on the positions of the transistors stuck-open. The other possible way in which *Class A* and *Class B* faults occur at the same time is when nMOS transistors and pMOS transistors are stuck-on at the same time. For this case, in general, the primary gate output depends on the relative resistances of the p-net and the n-net. The primary output is not safe in this case.

The fail-safe complex CMOS gate design is useful when one class of faults is more probable than the other. The distinction between *Class A* faults and *Class B* faults underlines the fact that all faults are not equally probable. The probability of occurrence of different faults depends on the environment and also on the manufacturing process and layout of the integrated circuit. For example, in an environment where the MOS devices are exposed to ionizing radiation, their threshold voltage varies in the negative direction (the nMOS threshold voltage "rebounds" or starts varying in the positive direction at very high total dose levels, about 10^6 rads(Si)) [8]. Thus nMOS devices are likely to be stuck-on whereas pMOS devices are more likely to be stuck-open. Similarly, in certain manufacturing processes, more than 60 percent of MOS chip failures were attributed to the positive charges trapped in the gate oxide [9]. Thus it can be seen that for both the examples given, *Class A* faults are more probable than *Class B* faults and fail-safe design can be done accordingly.

4 Modeling Transistor Faults

The transistor fault-model should reflect the actual mechanism of failure of the device. From the discussion on the threshold voltage variation of MOS transistors when exposed to ionizing radiation, it is seen that nMOS devices tend to turn on more easily, whereas pMOS devices become harder to turn on. This is true until the nMOS transistors begin to "rebound", and this does not happen until the total dose level becomes extremely high. Another effect is that nMOS transistors start to exhibit a leakage current from drain to source, which is relatively small in magnitude. Thus in the worst case, nMOS devices could be stuck-on whereas pMOS devices could be stuck-open in a radiation environment. In reality however, they are likely to be made *partially stuck-on* and *partially stuck-open.* This means that they are not solidly on in the digital sense, or totally off, but are weakly conducting in the stuck-on condition and can conduct very weakly in the stuck-open condition.

For SPICE simulation purposes, *Class A* transistor faults occurring in a radiation environment were modeled by changing the threshold voltage of the transistors. In the threshold voltage model, nMOS transistors stuck-on were simulated by changing their threshold voltage to about -0.2 Volts, so that they were slightly on even with a gate voltage of about 0 Volts. Using the same principle, pMOS transistors stuck-open were simulated with a threshold voltage of about -5.5 Volts, so that they were off even with a gate voltage of about 0 Volts applied to their gates, with a Vdd of 5 Volts. Thus in this model, nMOS transistors stuck-on had a high resistance. The actual values of the modified threshold voltages used, have no effect on the fail-safeness of the logic gate, but have a bearing on the fault-tolerance capability of the gate. This threshold voltage model has nMOS transistors stuck-on with a high equivalent resistance and pMOS transistors totally non-conducting in their stuck-open condition. One consequence of the nMOS transistors stuck-on having a high resistance was that the primary output of the *safe gate* tolerated the nMOS transistor stuck-on faults. These faults would not have been tolerated if the transistors were stuck-on with very low resistance. The fault-tolerance capability also

depends on the sizes of individual transistors in the circuit. The pMOS transistors stuck-open were totally non-conducting in the model used. Thus the faults were not tolerated, although the primary output still remained safe. If the pMOS transistors conduct to a certain extent in their stuck-open state, then some of the faults would be tolerated by the *safe gate.*

As an example for the fault simulation using the threshold model, consider the circuit in Figure 5. Consider the input transition from $[A, B, C] = [0, 1, 1]$ to $[A, B, C] = [0, 1, 0]$. SPICE simulation of this transition with the threshold voltages of all nMOS transistors (including Inv1) set at -0.2V shows that the output f changes from 4.99V to 0V, whereas $\bar{f}$ changes from 0V to 4.99V. It should be noted here that the nMOS transistors are only partially stuck-on in this case, so that the fault was easily tolerated.

Simulation of the faults where the transistors are stuck-on solidly, or totally stuck-open, was done by keeping the gates of the corresponding transistors at the respective logic level. The stuck-at faults on the input signal lines were also simulated in a similar manner. Thus transistors stuck-on solidly or completely stuck-open were simulated by keeping their gates stuck-at-1 or stuck-at-0. In Figure 5 assume m5 stuck-open completely. For the input transition of the previous example, SPICE simulation shows that the primary output remains steady at 5V. Thus the fault is not tolerated, but the 1-fail-safe property for the *Class A* fault is maintained. The output is safe.

A simple model for transistors stuck-open and stuck-on would be to replace them with resistances. This is based on the principle that in the steady state (dc), the transistor can be replaced by a resistance of appropriate value if its drain current and drain to source voltage are known. SPICE simulation of a typical nMOS transistor with a W/L ratio of $\frac{4\mu m}{2\mu m}$ shows that it has a dc resistance varying from 1K to 3K in the "on" state. Thus the conducting transistor has a resistance of the order of a Kilohm. If it is assumed that a stuck-open transistor has a resistance that is 2 orders of magnitude higher, and a stuck-on transistor has a resistance that is an order of magnitude lower in the worst case, then the stuck-open transistor can be replaced by a 100K resistor, and the stuck-on transistor can be replaced by a 100 Ohms resistor. In the circuit of Figure 5 assume m10 stuck-on with a resistance of 100 Ohms. For the input transition described in the two previous examples, SPICE simulation of the fail-safe complex CMOS gate with the stuck-on transistor replaced by the 100 Ohms resistor shows that the primary output f remains at 5V, while the node $\bar{f}$ goes from 0V to 1.02V. Thus the primary output remains safe, even though the fault is not tolerated. It must be noted that for the device size considered, 100 Ohms is a rather small resistance even for a transistor conducting heavily. It was interesting to note that the pMOS stuck-open faults in the two inverters of the *safe gate* were tolerated when the high resistance model was used. The stuck-open fault in the *basic gate* was not tolerated, but the 1-fail-safe property was maintained by the *safe gate.*

5 Summary

A CMOS gate which is fail-safe for a given class of faults was presented. SPICE simulations indicate that the fail-safe gate can tolerate several faults. Fail-safeness is preserved for multiple faults also. Fail-safe systems can be duplicated for the correction of failures in one system [2]. Basic fail-safe circuits should be studied further for the realization of fail-safe logical hardware design.

Acknowledgement: This research was supported in part by NASA under grant NAGW-1406, and by the Idaho State Board of Education under research grant 89-041. The authors would like to express sincere gratitude to Barbara Martin for her help in the preparation of the manuscript.

References

[1] R. Negrini, M. Sami, and R. Stefanelli, *Fault-Tolerance through Reconfiguration of VLSI and WSI Arrays*, Cambridge, MA, The MIT Press, 1989, Chap. 2

[2] H. Mine and Y. Koga, "Basic Properties and a Construction Method for Fail-Safe Logical Systems", IEEE Trans. on Electronic Computers, Vol. EC-16, No. 3, pp. 282-289, June 1967

[3] Y. Tohma, Y. Ohyama, and R. Sakai, "Realization of Fail-Safe Sequential Machines using k-out-of-n Code", IEEE Trans. on Computers, Vol. C-20, pp. 1270-1275, Nov. 1971

[4] R. Chandramouli, "On Testing Stuck-Open Faults", Proceedings of the 13th Fault Tolerant Computing Symposium, pp. 258-265, 1983

[5] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley, 1985, Chap. 4

[6] D. Sawin, III and G. Maki, "Fail-Safe Asynchronous Sequential Machines", IEEE Trans. on Computers, Vol. C-2 , pp. 675-677, June 1975

[7] L. Glasser and D. Dobberpuhl, *The Design and Analysis of VLSI Circuits*, Addison-Wesley, 1985, Chap. 5

[8] T. Ma and P. Dressendorfer, *Ionizing Radiation Effects in MOS Devices and Circuits*, New York, NY, John Wiley and Sons, 1989, Chap. 5

[9] C. Timoc et al., "Logical Models of Physical Failures", Proceedings of the International Test Conf., pp. 546-553, 1983