An approach to the design of highly reliable and fail-safe digital systems*

by HENRY Y. H. CHUANG
University of Pittsburgh
Pittsburgh, Pennsylvania

and

SANTANU DAS
North Electric Company
Delaware, Ohio

INTRODUCTION

Recent progress in electronic technology has resulted in extensive use of electronic digital or logical systems in many complex control processes where weight or size as well as reliability and safety is important. Typical examples are, to name a few, nuclear reactor control, missile guidance systems, "fly-by-wire" systems in aircraft or spaceship control, and electronic telephone switching systems (ESS). In such applications, if the system is not well protected, some failure could result in a catastrophic accident or unacceptable loss in terms of lives and resources. For the system to be safe and reliable, not only the computers or digital systems in the control loop have to be reliable, but also the sub-systems which take part in the data acquisition or control should be fail-safe and reliable. In order to lessen the chance of system breakdown and unsafe failures in applications of this nature, many protective methods have been proposed and studied. Generally, the methods can be classified as: (1) fault-masking approach,1-9 where the emphasis is on achieving higher reliability by using redundancy to mask failures, and (2) fail-safe approach10-14 where attention is directed to the design of systems in such a way that they take a predetermined safe state whenever failures occur.

Although reliability and fail-safety are both vital in many applications, a design technique which can have the merits of both the fault-masking approach and the fail-safe approach has been strikingly lacking. By properly combining the two redundancy methods (in particular, the NMR and the N-fail-safe logic), we have obtained a design approach for realizing reliable and fail-safe systems which can also have several significant advantages. These include excellent degradability, good testability, design flexibility and simplicity, as well as real-time error indication. In this article the authors expand and generalize the concepts presented in their previous short paper,15 and try to throw light on its many advantages.

THE NEW FAULT-RESTORATION METHOD

The new fault-restoration scheme is shown in Figure 1. Here the individual logic blocks have two levels of failure.15,16 In the first level, which we call the "safe output" level, the output is not correct but its value is something different from normal operating values of the output of the logic blocks. In the second level of failure, the output is neither "correct" nor "safe", and the value it takes up is one of the normal operating values (but which is not what it is supposed to have). The restorer's function is to give correct output as long as there are more copies having correct outputs than those having incorrect output, no matter how many copies have safe output. In case there are as many copies with correct output as with incorrect output (again no matter how many other copies have safe output), then the restorer output would be a safe value. The restorer function for a three-copy scheme is shown in Table I for elucidating the principle of this scheme. Here each of the three-copies, named A, B, and C, can have three distinct output states, R (right or correct), W (wrong or incorrect) and S (safe).

It is assumed that in each logic block, the probability of getting the safe output S is higher than that of having an incorrect output. In other words, more internal failures are required to result in an incorrect output than a safe output. This safe output S can be used to indicate the necessity of outside intervention and, depending on the application, provisions could be made for either manual or automatic corrective measures (say replacement of the faulty unit). Thus unlike the majority voting1,4 or other fault-restoration schemes, this scheme will have two levels of failure. The result is that the system would have a

*This work was supported by National Science Foundation under Grant GJ-35569.
high correct output reliability and still higher safe output reliability, and so reliability and fail-safety can be achieved simultaneously.

To realize such a system, ideally each individual copy in Figure 1 would be made of such logic primitives as will produce safe output S in case of failure, and the failures in the primitives would tend to manifest themselves at the outputs of the copies by driving them to S. Most of the common logical components do not exhibit such failure characteristics. In addition, since most available logical components are two-valued, we have to resort to some sort of coding to build the system we have envisioned, using available electronic technology. One practical solution is to use the N-fail-safe logic\textsuperscript{12,14} to realize the individual copies of logic. In the next section a short overview of N-fail-safe logic would be given.

N-FAIL-SAFE LOGIC

The truth table for minimal information loss\textsuperscript{12,14} N-fail-safe NOT, OR, and AND are given in Table II; and Figure 2 shows their double-rail realizations. By interchanging the two output terminals of OR and AND, we have NOR or NAND respectively. These are called N-fail-safe primitives. In this figure, the variable $x$ is coded by $(x_1, x_2)$, $y$ by $(y_1, y_2)$, and $f$ by $(f_0, f_1)$. Logical 0 and 1 are represented by (0, 1), (1, 0) respectively, and N is represented by either (0, 0) or (1, 1).\textsuperscript{*} One way to synthesize an N-fail-safe logic circuit of a given Boolean function is to first synthesize a non-fail-safe circuit and then replace each gate by its corresponding N-fail-safe primitive. Obviously the circuits synthesized by this replacement method contain at most twice as many components as needed by their non-fail-safe versions.

It is to be noted that, in such OR and AND primitives, one or both of the gates failing in the same direction (i.e., both fail to 1 or both to 0) will result in an N output even if there is an N input, provided it is also due to a fault in the same direction. Therefore, in an N-fail-safe logic system, single fault or multiple faults all in the same direction will always result in N output, while multiple faults not all in the same direction may result in an erroneous 0 or 1 output. Thus, if the N-fail-safe logic is built of ideal asymmetrical components then it will function perfectly. In our method, the N-fail-safe logic is applied in such a way that even if non-asymmetrical components are used the system would still have very high reliability and fail-safety.

### Table I—Restorer Function of a Three-Copy Scheme

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>S</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>S</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>W</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>W</td>
<td>S</td>
<td>R</td>
<td>S</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>S</td>
<td>W</td>
</tr>
<tr>
<td>W</td>
<td>S</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>R</td>
<td>W</td>
</tr>
</tbody>
</table>

### Table II—Truth Table for Minimum Information Loss N-Fail-Safe NOT, OR, and AND

<table>
<thead>
<tr>
<th>Variables</th>
<th>$x$</th>
<th>$y$</th>
<th>NOT $\overline{x}$</th>
<th>OR $x+y$</th>
<th>AND $x \cdot y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>N</td>
<td>0</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>1</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

* Takaoka assumed only $0 \rightarrow 1$ failure and therefore N is (1, 1) in his case.
An Approach to the Design of Highly Reliable and Fail-Safe Digital Systems

REALIZATION OF FAULT-RESTORATION

The implementation of our fault-restoration method using the double-rail N-fail-safe logic is best elucidated by using a simple example. Figure 3a shows a possible scheme for restoration using two copies of N-fail-safe logic. The network $fa$ and $fb$ are identical copies of N-fail-safe logic realizing a function $f$. The restorer produces output according to the combinations shown in Table III. In the table, columns A and B list outputs of the two copies and the “output” column gives the restorer output. N (which corresponds to safe state “S”) denotes either (0, 0) or (1, 1), R the correct output, and W the incorrect output.

Table III—Restorer Output For Two-Copy System

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td>N</td>
<td>R</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>W</td>
<td>N</td>
<td>W</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>R</td>
<td>W</td>
<td>N</td>
</tr>
<tr>
<td>W</td>
<td>R</td>
<td>N</td>
</tr>
</tbody>
</table>

Figure 3b shows the map for the restorer function. For realization, the N entries in the map are replaced arbitrarily by either (0, 0) or (1, 1).

Figure 3c gives the specific assignment used for the restorer shown in Figure 3a. It is to be noted that this restorer is also fail-safe.

From the collection of the Computer History Museum (www.computerhistory.org)
Unlike the other redundancy schemes, the scheme described here has two levels of failure: (1) restoration failure (when the output is not correct), and (2) catastrophic failure (when the output is neither correct nor N). Since each N-fail-safe primitive has at most two gates, if a Boolean function needs \(n\) gates in its realization, its N-fail-safe version will need at most \(2n\) gates. In the following we first derive the failure probability formulas for our two-copy system of Figure 3 and then generalize them.

(1) **Probability of N Output**

Let \(p = s + t\) be the failure probability of a gate, where \(s\) and \(t\) denote the probabilities of failing to 0 and to 1 respectively. The system can fail to correct faults and produce N output in two different ways: (1) one (or more) gate in each copy fails in the same direction (all fail to 0 or all fail to 1), and as a result both the copies give output N, which is passed on to the final output by the restorer, and (2) two (or more) gates fail in opposite directions in one of the copies, thus giving rise to a W value at the output of that copy. In the latter case, even if the other copy is working correctly, the restorer output will be N.

The probability of case 1 can be approximated by that of two gates failing in two separate copies, and is given by

\[
Q_1 = \left[2np(1-p)^{2n-1}\right] \approx 4n^2p^2
\]

The probability of case 2 can be approximated by that of two gates failing in opposite directions in a copy, which is

\[
Q_2 = \frac{\binom{2n}{2}}{2} \times 2st \times (1-p)^{2n-2} \approx 8n^3p^2t
\]

The maximum value of \(st\) is \(p^2/4\). Thus

\[
Q_w = 8n^3p^2t/4 = 2n^2p^2
\]

* This product np is small in practice (usually \(\ll 1\)). So, higher order terms are negligible.

The total probability of N output in a two-copy system is thus given by

\[
Q_N(2) = Q_1 + Q_2 = 4n^2p^2 + 2n^2p^2 = 6n^2p^2
\]

(2) **Probability of W Output**

An output W will occur if, because of some faults, one of the copies gives W output while the other copy has an output N. (We can ignore the case when both copies have W output as that probability is very small.) If it is assumed that any two gates failing in different directions in one copy give rise to a W output, while a single gate failure in another copy gives rise to N at its output, then the probability of W output for a two-copy system is

\[
Q_w(2) = 2 \times 2np \times \frac{2n}{2} \times 2st(1-p)^{2n-2} \times (1-p)^{2n-1} \approx 4n^3p^3
\]

(3) **Restoration and Catastrophic Failure Probabilities**

The restoration failure occurs when the output is either N or W, while the catastrophic failure occurs when the output is W. Therefore the restoration failure probability \(Q_r(2)\) is

\[
Q_r(2) = Q_w(2) + Q_w(2) = 6n^3p^3 + 4n^3p^3 \approx 6n^3p^3
\]

and the catastrophic failure probability \(Q_c(2)\) is simply

\[
Q_c(2) = 4n^3p^3.
\]

These are the worst case failure probabilities, as the aforementioned assumption may not result in these faults and as it has been assumed that each N-fail-safe primitive has two gates which actually is not the case for the NOT primitive. Moreover, the actual value of \(st\) will be far less than \(p^2/4\) because most electronic components have asymmetric failure characteristics (i.e., \(s \neq t\)). When ideal asymmetrical elements are used,

\[
Q_r(2) = Q_t = 4n^3p^3, \quad \text{and} \quad Q_c(2) = 0.
\]

Detailed comparison between our scheme and the other popular redundancy schemes are given in Reference 16.

(4) **Generalization**

As mentioned earlier, the restoration strategy can be extended to systems using any number of copies, odd or even. Let \(P_n, P_o, \text{ and } P_r\) be the probabilities of a copy to have "safe", "wrong", and "right" output respectively and if \(x\) is the number of copies, then it can be shown\(^a\) that the probability of N output for an \(x\)-copy system is:

\[
Q_N(x) = \sum_{x=0}^{[x/2]} \binom{x}{k} \frac{x!}{(x-2k)!k!(x-2k)!} (P_r)^{x-2k} (P_w)^k (P_o)^k,
\]

and that of W output for an \(x\)-copy system is:

\[
Q_w(x) = \sum_{x=0}^{x-1} \left[ \sum_{y=0}^{[x/2]} \binom{x}{y} \frac{x!}{(x-y-1)!y!(x-y-1)!} (P_r)^{x-y-1} (P_w)^y (P_o)^y \right] (P_r)^{x-y-1} (P_w)^y (P_o)^y.
\]

Where \([U]\) denotes the largest integer \(\leq U\). As before, the
probability of \( w \) output is really the catastrophic failure probability, while the sum of the probabilities of \( w \) output and \( N \) output is the restoration failure probability. Hence,

\[
Q_r(x) = Q_N(x) + Q_w(x)
\]

and

\[
Q_r(x) = Q_w(x)
\]

When double-rail X-fail-safe logic is used for realization, good estimations of \( P_r, P_s, \) and \( P_w \), as explained earlier, are given by:

\[
P_r = (1-p)^{2n-1} = 1
\]

\[
P_s = (1-p)^{2n} - 2np = 2np
\]

\[
P_w = (1-p)^{2n-2} \frac{2n^2}{4} = n^2p^2
\]

The failure probabilities for two-copy and three-copy systems plotted along with that of the TMR (three-copy majority voting) are shown in Figure 4.

**ADVANTAGES**

The new scheme of fault-restoration, we are advocating, is superior in performance to most of the existing redundancy schemes. Some of the important advantages are as follows.

1. **High Reliability and Fail-Safety**

   As shown in Figure 4, the (restoration) reliability of our scheme is comparable to that of TMR. But, our catastrophic failure probability is very low. Thus, our scheme can also provide good fail-safe protection which is not available in other fault-restoration schemes. A more comprehensive reliability comparison with other fault-tolerant design methods is given in Reference 16.

2. **Excellent Degradability**

   Our scheme has also degradability unmatched by any other existing redundancy method. As for instance, switching from triplex system (using three copies) to a duplex system is very easy as one can always build the logic in such a way that shutting off the power of one of the copies would force its output to \( N \). Similar switching from duplex to simplex or for that matter from any number of copies to any smaller number of copies is possible without changing the restorer. This is so because our restorer for higher number of copies always logically covers that for lower number of copies. Among all other redundancy schemes, TMR (or NMR) is the only scheme that is degradable. But even for TMR (or NMR) this sort of degradability is absent as it cannot switch from triplex to duplex, although it can switch to simplex from triplex. To do even that one has to bypass the voter which could be unwieldy.

3. **Trade-off Between Restoration and Catastrophic Failures**

   One can make trade-offs between restoration failure and catastrophic failure probabilities by altering the restorer logic or the restoration strategy, keeping the number of logic copies unchanged. As for instance, Table IV shows the different behaviors of two restorers along with the probabilities for various possible output combinations of a three-copy system. From this table, we get for restorer 1:

   restoration failure probability: \( \approx 16n^p\)
   catastrophic failure probability: \( \approx 4n^p\),

   and for restorer 2:

   restoration failure probability: \( \approx 20n^p\)
   catastrophic failure probability: \( \approx 15n^p\).

Thus, we see that by using restorer 1 instead of restorer 2 we can decrease the restoration failure at the cost of increasing the catastrophic failure. Restorer 2 corresponds to the restoration strategy based on which the generalization of failure probability analysis is done.

One of the realizations of restorer 1 is shown in Figure 5. Of course, one can always realize the restorer circuits in the same way as we did for the 2-copy restorer of Figure 3.

4. **Amenability to Hybrid Redundancy**

   In situations where a very high mission life is to be achieved spares can be used, as in the case of TMR, to obtain dynamic or hybrid redundancy. As the safe output 8 indicates failures in the network, it can be conveniently used to initiate the necessary switching of spares. Because in our system any number of copies, odd or even, can be used, the spare switching circuits are simpler.12

5. **Amenability to Simple MOS IC Realization**

   It is well-known that it is comparatively easy to realize monotonic functions using current MOS IC technology. Since functions \( f_0 \) and \( f_1 \) in an \( N \)-fail-safe logical system are both monotonic, our fault-restoration method has an edge

---

**Table IV—A Three-copy System with Two different Restorers**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>N</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>N</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>( (1-n^p)^2 \leq 1 )</td>
</tr>
<tr>
<td>W</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>3np^2</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>W</td>
<td>N</td>
<td>R</td>
<td>W</td>
<td>N</td>
</tr>
<tr>
<td>W</td>
<td>N</td>
<td>R</td>
<td>W</td>
<td>N</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>N</td>
<td>W</td>
<td>3 ( (2np)^3 = 6np^3 )</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>( (2np)^3 = 6np^3 )</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>( (2np)^3 = 6np^3 )</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>R</td>
<td>W</td>
<td>N</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>R</td>
<td>W</td>
<td>N</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>R</td>
<td>W</td>
<td>N</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>R</td>
<td>W</td>
<td>N</td>
</tr>
</tbody>
</table>

---

From the collection of the Computer History Museum (www.computerhistory.org)
over other methods from realization point of view. Further, it should be pointed out that even if one uses N-fail-safe logic, it is not necessary that one has to build it by using N-fail-safe primitives. As has been shown in Reference 13, it is also possible to obtain $f_a$ and $f_b$ of the N-fail-safe logic from direct realization. In such a case, two separate LSI (or MSI) chips can be used to realize $f_a$ and $f_b$. It is very likely that one of these LSI chips would malfunction at a time and not both at the same time. Thus, when a copy fails, its output will most likely be N and not W.

(6) Good Testability

Proective redundancies might mask detectability. This is the case for most fault-masking methods including the quadded logic and interwoven logic. Some non-protective redundancies such as those for obviating hazards could also hinder fault detectability. However, the N-fail-safe logic, with its monotonic component functions, has inherently good detectability. It has been reported that in a monotonic logic network not only all faults are detectable but also the minimal detection cover can be easily obtained. Further, it can be shown that in an N-fail-safe logic network even the redundancies for avoiding hazards cannot mask its fault detectability.

CONCLUSIONS

A new design method for the realization of highly reliable and fail-safe digital systems has been described here. The method properly combines the fault-masking and the fail-safe approaches, and thus possesses the merits of both. It also has significant advantages, such as the excellent degradability and two-level protection, which are not available in existing redundancy methods.

A major disadvantage of this method could be the high cost of realization because a copy of N-fail-safe logic might require twice as many gates and connections to realize as would that of conventional logic. However, it should be noted that the N-fail-safe component functions are always monotonic and monotonic functions generally have simpler IC realizations. Consequently, this demerit might not be so bad at all.

REFERENCES