SELF-CHECKING ADDER FOR LARGE SCALE INTEGRATION

Antonin Svoboda

Fellow, IEEE

Los Angeles, California 90024

1. Abstract

The testing of LSI chips is expensive and unsatisfactory. On the other hand there are cases (as in space ship computers) where a damaged chip must be localized and replaced. The use of self-checking chips seems to be one of several possible solutions of this problem. The theory of the structure of self-checking logical circuit is covered by literature at least at the fundamental form (see References). However, even when the design principles are supposed to be known, their application to the actual creation of a self-checking circuit of an average complexity is and will remain an art. The reason is quite simple and fundamental: optimization of design criteria (engineering qualifications, performance and physical properties of components of the circuits are entities possessing different physical dimensions - it is impossible to qualify, for instance, two circuits A,B designed for the same task by comparing their speeds and costs if A is faster than B but B is cheaper than A) will never be objective and independent of the talent or whim of the circuit designer. As an example of the design of a self-checking circuit we present here a binary adder (Full Adder) designed under the following considerations:

1: The adder is composed from gates (AND,OR,NAND,NOR,...).

2: Two level design was chosen.

3. Ripple carry addition was accepted as sufficient simplification for the design experiment.

4. Only two classes of fault were considered: Stuck at ONE, Stuck at ZERO.

5. Any single fault in the circuit must be signalized either during the activity of the circuit (clock ON) or during a test fault injection (clock OFF).

6. The number of cases where a multiple fault remains undetected must be extremely low in comparison with all possible cases.

To obtain an adder with all those requirements the following design idea is used: The adder's three bit input (X,Y,C) is transformed into an eight bit signal (S_i , i = 0,1,...,7) by using ONE FROM EIGHT CODE. This signal, produced by the first level of the circuit, is then transformed by the second level of the circuit into the desired output signal (Z,G) by using four wires and TWO FROM FOUR CODE. Ten fault signals (Fig. 1) are derived from those two codes and checked at the proper state of the clock.

This research was supported by the U. S. Office of Naval Research, Contract No. N00014-75-C-0650.

2. Single Fault Analysis

Single as well as multiple fault analysis presented here is based on a program (APL|360) which produced a detailed print-out of signal level distribution under the assumption that a fault can be present anywhere.

Symbolism Convention. The output signal of a gate is designated by the identification symbol of that gate. For instance, the output signal of the gate A_1 is designated by A_1. Because A_1 is a NOR-gate we write: $A_1 = NOR(Z_1,Z_0) = \overline{Z}_1\overline{Z}_0$, where Z_1, Z_0 are output signals of the gates Z_1, Z_0.

The equations of the self-checking adder are well defined by its schematic in Fig. 1 and they are not mentioned here explicitly because the whole matter is too elementary and seems to be a waste of space here. A short description of the APL-program is all what is needed.

The terminal prints configurations of signals (A_1, A_2,A_3,A_4) belonging to all 16 configurations of signals (Z_0,Z_1,G_0,G_1) by disregarding the fact that only four of them are faultless.

Similarly follows the print-out of configurations of signals (Z_0,Z_1,G_0,G_1) and of signals (B_1,B_2,B_3, B_4,B_5,B_6) for all 256 configurations of signals (S_0,S_1,S_2,S_3,S_4,S_5,S_6,S_7) by disregarding the fact that only eight S-configurations are faultless.

Finally follows the print-out of the configurations of signals (S_0,S_1,S_2,S_3,S_4,S_5,S_6,S_7) belonging to all 64 configurations of signals (X_0,X_1,Y_0, Y_1,C_0,C_1) by disregarding the fact that only eight of them are faultless.

The print-out is too large to be reproduced here in full. It is very useful, however, as a guide to important logical relations between groups of encoded signals and as an insurance measure against design mistakes.

The adder has three data inputs X, Y, C (Carry-in) and therefore exactly eight active states indexed by i = 0,1,...,7 (see Table 1). The state will be called active when the addition has been performed (the output signals reached stable levels after the clock impulse T has been ON long enough).

The adder has two data outputs Z (the sum), G (Carry-out). As for any adder X + Y + C = 2G + Z (where + means the ordinary addition).

The double rail signalization is used for each input and output variable so that we have Data signals: $X_1 \neq X_0$, $Y_1 \neq Y_0$, $C_1 \neq C_0$, $Z_1 \neq Z_0$,

$G_1 \neq G_0$. The relationship of complementarity is satisfied only as long as the input signals and the processing hardware are both faultless. The fault detecting adder must recognize a fault in the input.

The adder has a Fault Injection input T^*. During the active state the fault injection is switched off by making $T^* = 0$ (LOW). During the passive state ($T = 0$) the fault is injected by making $T^* = 1$ for a very short time. Then $A_3 = A_4 = B_1 = B_2 = B_3 = B_4 = B_5 = B_6 = 1$ unless some of the corresponding gates are stuck at zero or some of the gates S_k are not stuck at one.

Table 1 lists the values of all signal variables for all faultless input signal configurations and for faultless hardware elements. It is clear that $S_j = 0$ for $j = i$ and $S_j = 1$ for $j \neq i$. Note that all fault signals are at ZERO value.

The fault analysis of the adder is presented in the form of Theorems valid for the system of logical relations represented by Fig. 1 in combination with fault constraints. The present chapter is concerned with the single fault analysis so that the following postulate remains in force: The total of all faults is less than two.

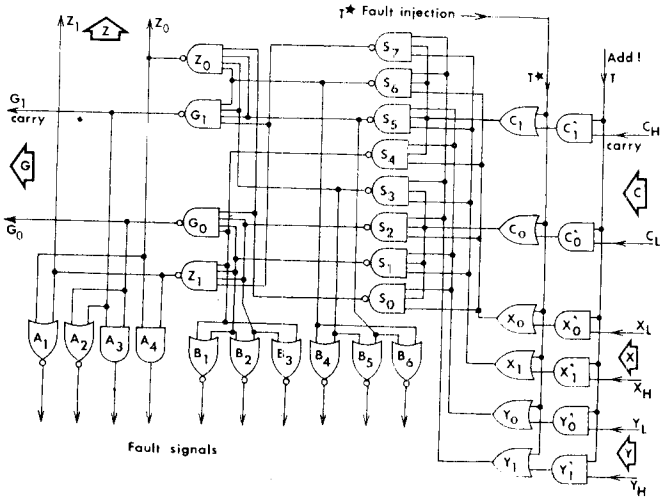

Figure 1

Look for Table 1 at the end of this article.

Faultless state for $T = T^* = 0$ (passive state and Fault Injection switched off) features the following configuration of signals (use Fig. 1):

$$X_0 = X_1 = Y_0 = Y_1 = C_0 = C_1 = Z_0 = Z_1 = G_0 = G_1 = 0$$

$$S_0 = S_1 = S_2 = S_3 = S_4 = S_5 = S_6 = S_7 = A_1 = A_2 = 1$$

Theorem 1. ($T = T^* = 0$) AND ($A_1 = A_2 = 1$) IMPLIES

[(from the set of gates $\{A_1, A_2, S_0, S_1, \ldots, S_7\}$ none is stuck at zero)

AND (from the set $\{Z_0, Z_1, G_0, G_1\}$ none is stuck at one)] .

Proof. The conclusion is obvious for the gates A_1 , A_2. If one or more of the gates S_k , $k = 0, 1, \ldots, 7$ is stuck at zero, then some of the signals Z_0, Z_1, G_0, G_1 become HIGH and A_1 or A_2 becomes LOW. The same will happen if one or more of the gates Z_0, Z_1, G_0, G_1 is stuck at one.

Result: If for ($T = T^* = 1$) we detect $A_1 = A_2 = 1$ we conclude that the gates A_1, A_2 are not stuck at zero.

Faultless state for $T^* = 1$ (a state with a Fault Injection) features the following configuration of signal values:

$$X_0 = X_1 = Y_0 = Y_1 = C_0 = C_1 = Z_0 = Z_1 = G_0 = G_1 = 1$$

$$B_1 = B_2 = B_3 = B_4 = B_5 = B_6 = A_3 = A_4 = 1$$

$$S_0 = S_1 = S_2 = S_3 = S_4 = S_5 = S_6 = S_7 = A_1 = A_2 = 0$$

Theorem 2. [($T^* = 1$) AND ($A_3 = A_4 = B_1 = B_2 = B_3 = B_4 = B_5 = B_6 = 1$)] IMPLIES

[(from the set of gates $\{A_3, A_4, B_1, B_2, B_3, B_4, B_5, B_6, Z_0, Z_1, G_0, G_1\}$ none is stuck at zero) AND (from the set of gates S_k , $k = 0, 1, \ldots, 7$} none is stuck at one)] .

Proof follows the same pattern as for Theorem 1.

Result: If the Fault Injection yields $A_3 = A_4 = 1$ and every $B_j = 1$ we conclude that the corresponding gates are non stuck at zero.

Now our attention is turned towards an active state $T = 1$ with $T^* = 0$ (without a Fault Injection).

Theorem 3. [(Every fault signal gate A_j, B_k is non-stuck at zero) AND (Every fault signal is LOW)] IMPLIES (The output signal values represent the correct sum of a faultless configuration of input signal values).

Proof. Supposing that the active state follows a passive state during which the fault signal gates were

tested and found as non-stuck at zero and the set of gates $\{Z_0, Z_1, G_0, G_1\}$ faultless (see Theorems 1, 2) we conclude:

$$(A_1 = A_4 = A_2 = A_3 = 0) \Rightarrow [(\overline{Z}_0\overline{Z}_1 + Z_0Z_1 = 0) \text{ AND } (\overline{G}_0\overline{G}_1 + G_0G_1 = 0)] \Rightarrow [(Z_0 \neq Z_1) \text{ AND } (G_0 \neq G_1)]$$

so that the absence of A-gate gault signals signalizes that the output signals satisfy the condition of complementarity. The satisfaction of this condition is not sufficient, however, to keep the S-signal configurations within the desirable limits of the one (LOW) out of eight encoding belonging to faultless input signal values. We see it in the Table 2 which is compiled from the exhaustive print-out mentioned at the beginning of this chapter.

$Z_0Z_1G_0G_1$	S								B					
	0	1	2	3	4	5	6	7	1	2	3	4	5	6
0 1 1 0	1	0	0	1	0	1	1	1	1	1	1	0	0	0
0 1 1 0	1	0	0	1	1	1	1	1	0	1	0	0	0	0
0 1 1 0	1	0	1	1	0	1	1	1	1	0	0	0	0	0
0 1 1 0	1	1	0	1	0	1	1	1	0	0	1	0	0	0
0 1 1 0	1	0	1	1	1	1	1	1	0	0	0	0	0	0
0 1 1 0	1	1	0	1	1	1	1	1	0	0	0	0	0	0
0 1 1 0	1	1	1	1	0	1	1	1	0	0	0	0	0	0
1 0 0 1	1	1	1	0	1	0	0	1	0	0	0	1	1	1
1 0 0 1	1	1	1	0	1	0	1	1	0	0	0	0	1	0
1 0 0 1	1	1	1	0	1	1	0	1	0	0	0	1	0	0
1 0 0 1	1	1	1	1	1	0	0	1	0	0	0	0	0	1
1 0 0 1	1	1	1	0	1	1	1	1	0	0	0	0	0	0
1 0 0 1	1	1	1	1	1	0	1	1	0	0	0	0	0	0
1 0 0 1	1	1	1	1	1	1	0	1	0	0	0	0	0	0
0 1 0 1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
1 0 1 0	0	1	1	1	1	1	1	1	0	0	0	0	0	0

Table 2. Configurations of S-signals and B-signals for faultless output signal configurations.

The Table 2 shows that when an S-signal values configuration produces an acceptable output (Z_0, Z_1, G_0, G_1) (e.g., an output obeying the conditions of complementarity) then the S-signal configuration contains exactly one zero (plus seven ones) only when every B-signal is low. To complete our proof we recall that the S-gates have been tested twice during the passive state and found faultless (Theorems 1, 2 !). If we accept the S-gates as faultless and consult our exhaustive printout to find all configurations of signals $(X_0, X_1, Y_0, Y_1, C_0, C_1)$ which produce S-signal configurations including exactly one zero (and seven ones) we find that only 8 such input configurations of input signals exist: those listed in Table 1. That completes the proof.

Result. Absence of fault signals during an active state means that the input and output data are correctly implemented and the hardware has produced the correct result.

3. Fault signalization

The self checking adder presented here is intended as an element of a complex logical block (chip) implemented by LSI technology. The checking principle proposed here wants the same fault checking facility for each component integrated on the chip. Nominaly:

1. The clocking provides for two checking intervals: ACTIVE, PASSIVE.

2. Two tests are triggered during the PASSIVE interval (T = 0):

   21: Test without Fault Injection ($T^* = 0$): is labeled faultless if all signals are HIGH on certain wires (A_1, A_2 type of wires).

   22: Test with Fault Injection ($T^* = 1$) is faultless if all signals are HIGH on certain wires ($A_3, A_4, B_1, B_2, B_3, B_4, B_5, B_6$ -type of wires all different from wires at 21:)

3. One test is triggered during the ACTIVE interval ($T = 1$, $T^* = 0$). When all signals have steady state the signals must be LOW on every wire mentioned above (e.g., all fault signals must be LOW).

All "A_1, A_2 -type" wires are grouped together as they come from different components on the chip. Another group is formed from all wires of the type A_3, A_4, $B_1, \ldots, B_6$. We shall refer to the first of these groups as FIRST GROUP (type A_1, A_2) and to the second of them as SECOND GROUP.

The fault signalization block can be designed in many ways. The following design is offered as an example only.

A clocking system is established (Fig. 2) which generates 4 impules T, P, T^*, Q which are periodical except for the impulse T which is permitted to enter the checking block only when the input data entering all checked elements are supposedly meaningful.

The fault detecting algorithm is described with the reference to Fig. 3.

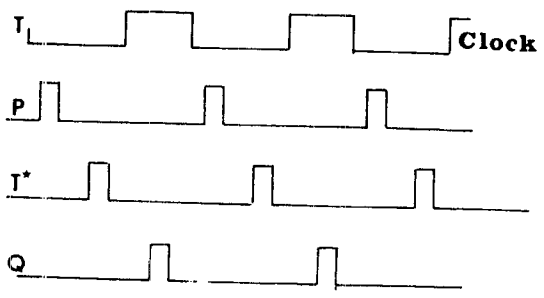

Figure 2

BEGIN with the Fault Signal reset to the value F = 0 (LOW). This reset occurs always when the power of the chip is switched on and never else. (Hardware: The R-S Flip-Flop FF is reset to 0).

1: FOR (P = 1) FORM the product of values of all FIRST GROUP signal variables. IF this product is zero THEN signalize a FAULT. (Hardware: Form a signal NF equal to the NAND of all FIRST GROUP signals. Pass the signal NF through an AND-gate to get P $\wedge$ NF. The resulting signal must set the flip-flop FF to F = 1)

2: For (T^* = 1) FORM the product of values of all SECOND GROUP signal variables. IF this product is zero THEN signalize a a FAULT. (Hardware: NAND of all SECOND GROUP variables is AND-ed with T^* to get a signal which triggers the flip-flop FF to F = 1).

3: FOR (TQ = 1) FORM the sum (Boolean) of all Fault Signal variables. IF this sum is one THEN signalize a FAULT.

(Hardware: OR all FIRST plus SECOND GROUP variables and use the resulting signal to trigger FF to F = 1).

The fault signal F can be used to stop the processing or just to energize a LED.

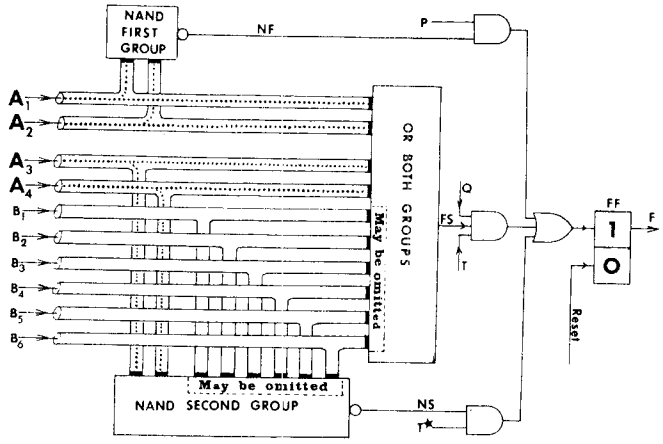

Figure 3

4. Multiple Fault Analysis

It is surprising that the adder is able to check multiple faults. We shall see that only high multiplicity of fault is necessary to prevent its detection. Theorem 1 can be extended by including the gates $X_0, X_1, Y_0, Y_1, C_0, C_1, X_0', X_1', Y_0', Y_1', C_0', C_1'$ as non-stuck at one. The Theorem 2 can be extended in a similar way by including the gates $X_0, X_1, Y_0, Y_1, C_0, C_1$ as non-stuck at zero.

We have to conclude that the favorable results of three tests (and the supposition that the fault detecting circuit was designed as fault tolerant) leave not much space for the occurrence of a multiple fault. (We see again that single fault is impossible). All what can happen is that some gates from the set $\{X_0', X_1', Y_0', Y_1', C_0', C_1'\}$ can be stuck at zero and some of the input signals $X_L, X_H, Y_L, Y_H, C_L, C_H$ faulty.

For instance if X_0' is stuck at zero and at the same time $X_L = X_H = 1$ (during the active state) there is a double fault which will pass undetected. When X_0' is stuck at zero permanently, however it will be detected later when the input signal configuration will contain $X_L = 1$, $X_H = 0$.

It is not difficult to show that there are no multiple faults which pass undetected if they are permanent.

Important criticism, If the test $T = T^* = 0$ is favorable it is proven that none from the S-gates is stuck at zero (Theorem 1). For that reason we have to conclude that any faulty configuration of S-signals listed in the Table 2 (each has more than one zero) must be caused by a faulty input signal level configuration (not by some S-gates stuck at zero). But by consulting the exhaustive print-out we discover the fact that a faulty signal level configuration does not exist which produces any faulty S-signal configuration found in the Table 2. Conclusion: the single fault detection ability of the adder will be not impaired if we leave out all B-gates. This simplification is recommended anywhere, where the reduction in multiple fault detection ability is acceptable.

Identifier of the faultless state	i =	0	1	2	3	4	5	6	7
Clock is ON, Faultinjection is OFF.	T =	1	1	1	1	1	1	1	1
Input X	X =	0	1	0	1	0	1	0	1
Input Y	Y =	0	0	1	1	0	0	1	1
Carry in C	C =	0	0	0	0	1	1	1	1
Output Z	Z =	0	1	1	0	1	0	0	1
Carry out G	G =	0	0	0	1	0	1	1	1
Signal $X_1 = X_H = X$	X_1 =	0	1	0	1	0	1	0	1
$X_0 = X_L = \bar{X}$	X_0 =	1	0	1	0	1	0	1	0
$Y_1 = Y_H = Y$	Y_1 =	0	0	1	1	0	0	1	1
$Y_0 = Y_L = \bar{Y}$	Y_0 =	1	1	0	0	1	1	0	0
$C_1 = C_H = C$	C_1 =	0	0	0	0	1	1	1	1
$C_0 = C_L = \bar{C}$	C_0 =	1	1	1	1	0	0	0	0
First level signals	S_0 =	0	1	1	1	1	1	1	1
	S_1 =	1	0	1	1	1	1	1	1
	S_2 =	1	1	0	1	1	1	1	1
	S_3 =	1	1	1	0	1	1	1	1
	S_4 =	1	1	1	1	0	1	1	1
	S_5 =	1	1	1	1	1	0	1	1
	S_6 =	1	1	1	1	1	1	0	1
	S_7 =	1	1	1	1	1	1	1	0
Second level signals	Z_0 =	1	0	0	1	0	1	1	0
$Z = Z_1$ =		0	1	1	0	1	0	0	1
	G_0 =	1	1	1	0	1	0	0	0
$G = G_1$ =		0	0	0	1	0	1	1	1
Fault signals $A_1 = A_2 = A_3 = A_4$	=	0	0	0	0	0	0	0	0
$B_1 = B_2 = B_3 = B_4 = B_5 = B_6$	=	0	0	0	0	0	0	0	0

TABLE 1. Faultless states of the adder.

REFERENCES

1. Avizienis, A., A Set of Algorithms for a Diagnosable Arithmetic Unit, Technical Report No. 32-546, Jet Propulsion Laboratory, Pasadena, California, March 1, 1964.

2. Avizienis, A., "Signed-Digit Number Representations for Fast Parallel Arithmetic," IRE Transactions, EC-10 (1961), pp. 389-400.