GC-10 | 999

ASAT

MILITARY TECHNICAL COLLEGE
CAIRO - EGYPT

RELIABLE FAULT-SECURE PROCESSOR FOR SPACE-FLIGHT CONTROLLERS

Dr. M. EL-Lithy

ABSTRACT

Reliability, availability are extremely important aspects of the overall performance of a digital controller. The safety, in addition, becomes a vital aspect for the space flight applications. In otherwords, a space-flight controller ought to be fault-secure.

The present paper presents an architecture, fault-secure and self-checking for a processor which is mainly intended for space-flight controllers.

The fault-secureness of a processor means that it never produces an incorrect output due to a fault unless it simultaneously announces the occurrence of that fault. Different techniques of concurrent fault-detection are applied and self-checking circuits are used, which result in minimizing the mean-time-to-detection (MTTD). The error-indication signals aid the system fault-diagnosis which reduces to minimum the repair time. Thus, the system maintainability and availability are well improved.

Lecturer, Dept. of Guidance, Military Technical College, Cairo, Egypt.

INTRODUCTION

Reliability is the probability that the system is operating correctly at any given time, while Availability is the fraction of time that the system is available for use. Maintainability relates to the ease with which the system is tested and repaired. For systems used in aerospace applications, the safety and fault-secureness imply that no fault is allowed to go undetected without announcing its occurrence to the system monitor.

The use of on-board computers for critical flight-control operations has largely expanded, especially in the manned space vehicles to ensure their mission success and safe operation. They are mainly used to perform guidance and navigation computations. The provision of digital-flight control by the on-board digital computer makes possible the use of digital filtering techniques to enhance the vehicle's stability. The digital control is simpler, lighter, more adaptable, and more flexible in satisfying the requirements of different flight regimes. It is also more relaible when a redundant arrangement is used. The on-board computer receives the data from sensors and measuring devices and processes these data to perform flight-critical functions. It also performs the function of monitoring the performance of other subsystems before and during flight. The results are presented to the flight crew on multifunction and dedicated displays. In addition to autonomous operation, the use of on-board computer for system monitoring also relieves the flight crew of exacting but routine details so that they can concentrate on the mission-oriented tasks. A functional diagram of a digital-flight controller is depicted in Fig.1. The processor, which is the heart of a digital controller consists of the main functional units depicted in Fig.2.

FAULT-SECURENESS AND SELF-CHECKING

A circuit whose output z(X,F) is encoded in an error-detecting code S is called a self-checking circuit. During fault-free operation, the output of such a circuit is always a code word; the set S is called the output code space. A checker monitors the circuit's output and detects errors by signaling the appearance of noncode words. For a given input X, a fault may or may not cause an error. If the fault does cause an error, it may change the output into either a noncode word (a detectable error) or an incorrect code word (an undetectable error) [2]

Definition (1):

A circuit is fault-secure if, for every fault from a prescribed set, the circuit never produces an incorrect code space output for code space inputs,i.e the output of a fault-secure circuit is always either correct or a noncode word.

Definition (2) :

A circuit is self-testing if, for every fault from a prescribed set, the circuit produces a noncode space output for at least one code space input

Definition (3):

A totally self-checking circuit (t.s.c) is a circuit that is self-testing for a normal input set N and a fault set F_t , and fault-secure for N and a fault set F_s i.e in a tatally self-checking circuit, all faults in F_t are

6

tested and no fault in F_s can cause an undetectable error in normal operation (Fig.3).

ERROR-DETECTING CODES AND TOTALLY SELF-CHECKING CHECKERS

The distance d of a code is the number of failures needed to change one code word into another and hence cause an undetectable error. A code of distance d is able to detect d-1 failures. Information in an error-detecting code is interpreted by a decoder that distinguishes between code words and noncode words, achieving the function of error detection and so, it may be named, a "checker". A checker must be self-checking so that a fault in the checker itself produces an error signal. For this, the output of the checker is encoded in an error-detecting code S. The simplest distance-two error-detecting codes are the duplication code (S={<00> , <11 >}) and the 1-out-of-2 c ode (S={< 01 > ,< 10> }). In the duplication code, a checker output of <01> or < 10> indicates an error, while in the 1-out-of-2 code, the outputs <00> and <11>indicate errors. The 1-out-of-2 code is generally preferred since it detects unidirectional multiple errors such as those produced by loss of power. [3-4] . Examples of t.s.c equality checkers are given in Fig. 4,5.

Choice of Error-Detecting Code

A code's cost is measured by the additional hardware required for its implementation (taking into consideration the t.s.c checkers) while the code's effectiveness is measured by its minimum distance and the size of the tested and secure fault sets. Table 1 summarizes properties of the most important codes. Codes are chosen by the following criteria:

- the hardware implementation cost.
- the effectiveness.
- its impact on the system speed.
- the fault assumption considered

The basic fault assumption is that an error-detecting code should be capable of detecting the likely single-component failures affecting only one physical intergrated circuit. Naturally, the relative checker speed and cost for given code is technology-dependent.

TOTALLY SELF-CHECKING PROCESSOR

Referring to the block diagram (Fig.2), the t.s.c design of each of its functional units is discussed separately. Generally, a bit-sliced circuit may use a single-bit code, but a byte-sliced circuit must use a byte error-detecting code while a nonsliced circuit must be duplicated.

Memory Unit

The memory is b-bit byte sliced with Kb-bit bytes data part. Referring to Table 1 and to our choice criteria, the only two possible codes are the duplication and b-adjacent codes. Other codes are excluded due to their slow or expensive checkers. Comparing both possible codes, the b-adjacent code is better as it is cheaper. Consider a memory unit of 8K-32-bit words, arranged as four 8K-8-bit PROM chips, One extra PROM chip is used for checking . It contains a check symbol for an interleaved parity i.e bit i of the check chip contains the parity over bit i of the four functional chips.
The t.s.c memory unit is given in Fig.6.

Table 1. Properties of Important Error-Detecting Codes

Code	Single-bit even-parity
Data part	n bits
Check symbol	1 bit
Fault-secure	Single bit-slice faults
Self-testing	Faults affecting fewer than all bits
Checker	n-input EXCLUSIVE OR tree and 1 inverter
Comments	Least redundancy, cheapest checker
Code	Duplication
Data part	n bits
Check symbol	n bits
Fault-secure	Faults affecting different bits in the two duplicates
Self-testing	Faults affecting different bits in the two duplicates
Checker	n-bit t.s.c. equality checker
Comments	Most redundancy, expensive checker, not self-testing for many double faults
Code	Distance-2 b-adjacent
Data part	k b-bit bytes
Check symbol	1 b-bit byte
Fault-secure	Single byte slice faults
Self-testing	Faults affecting fewer than all bytes
Checker	b k-input EXCLUSIVE OR trees and b-bit t.s.c. equality checker
Comments	Cheapest checker of all codes with b check bits, not self-testing for many k + 1-bit faults
Code	$(k,2^b)$ checksum
Data part	k b-bit bytes
Check symbol	1 b-bit byte
Fault-secure	Single byte slice faults
Self-testing	Faults affecting fewer than all bits
Checker	k-byte tree of b-bit modulo 2^b adders and t.s.c. equality checker
Comments	Cheaper than b-adjacent for parity-predicted arithmetic, slower and more expensive checker
Code	Low-cost arithmetic, A= 2^b-1
Data part	k b-bit bytes
Check symbol	1 b-bit byte
Fault-secure	Single byte slice faults except all stuck-at-0 or all stuck-at-1
Self-testing	Faults affecting fewer than all bits
Checker	k-byte tree of b-bit modulo $2^b - 1$ adders and t.s.c. equality checker
Comments	Direct implementation of arithmetic operations, not fault-secure for some single byte slice faults, slightly slower checker than checksum codes

Central-Processing Unit (CPU)

The CPU is not bit (byte) sliced and it performs both arithmatic and logical operations. All parity codes are not suitable and arithmetic codes are excluded as they are not preserved during logical operations. Thus, the only possible code is the duplication with comparison. A duplicated circuit is both self-testing and fault-secure for all faults affecting different bits in the two duplicates. Consider a 16-bit microprocessor chip (CPU), its t.s.c scheme is given in Fig. 7.

Data Paths And Input / Ouput Interfaces

By data paths we mean the communication buses, connecting processor units, with its buffer-drivers. Input/output interface circuits are of types: Buffers, MUX, Registers, Decoders, D/A and A/D converters. Except for Decoders, D/A and A/D converters, all other circuits and the buses are generally bit (byte) sliced with no interaction between bits (bytes). For such so called "nontransforming circuits" the treatment is exactly the same as the memory unit. This means that the two possible techniques for error-detection are the duplication and b-adjacent codes. Similarly, the latter one is preferred as its checker is faster and less expensive, as shown in Fig.8.

Decoders, D/A and A/D converters are treated by the duplication code as they are transforming circuits without explicit relation between their inputs and outputs .

Clock Frequency-Generator

The problem of checking periodic signals such as clocks has been recently examined by Usas [5]. He proposed a totally self-checking periodic signal checker (Fig.9) consisting of two monostable multivibrators (one-shots). M_1 is triggered on the leading edge of the input clock and produces a positive pulse of fixed width (t_{ON}). M_2 is triggered on the falling edge of the input clock and produces a positive pulse of width t_{off} . During normal operation the checker produces an output that is constantly changing between 01 and 10. It can be seen that the checker produces a noncode output (00 or 11) for any of the following faulttypes : input stuck-at faults, changes in the period and changes in the duty cycle. The error indication for period and duty cycle faults is not permanent, which must be taken into account in the design of circuitry that monitors the checker output. The given checker is totally self-checking for faults affecting a single monostable.

Error-Signal Manipulation AND Error Indication

We have now several totally self-checking checkers all over our processor. One practical problem is how to monitor the outputs of checkers. The checkers outputs are combined using totally self-checking two-rail checkers to form one pair of lines (1-out-of-2 code) to be checked. Assuming a sufficiently fast observer, one way to produce a single error indicator is given in Fig. 10 which will detect the brief appearance of a noncode output (00 or 11) and maintain an error indication until it is reset. This circuit is fault-secure for faults affecting only a single flip-flop or EX-OR gate.

COST OF SELF-CHECKING IMPLEMENTATION

Table 2 summarizes the additional hardware needed for implementation of the different units of the t.s.c processor.

Table 2 T.S.C Implementation Cost

Processor Units	Implementation Cost			
	LSI	MSI	SSI	LED
Memory	1		15	
CPU	1		11	
Data Paths & I/O - Interfaces :				
buses 16-bits (2x8)			11	
Register 16-bits (2x8)			12	
Buffers 16-bits (2x8)			12	
MUX 16-bits(4x4)			6	
Decoder 3/8		1	9	
Clock generator			3	1
Total	2	1	79	1

CONCLUSION

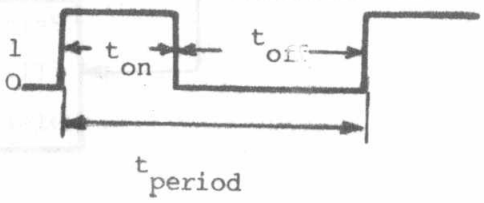
The present work provides an approach for the implementation of a hypothetical self-checking processor intended for the space-flight applications where the safety is a must. The proposed self-checking processor is fault-secure for as close as possible to 100% of all the faults affecting only a single integrated circuit and it is self - testing for the same set of faults. The self-checking processor employs off-shelf components. To reduce the count of ICs used, the implementation cost of totally self-checking (t.s.c) equality checkers ought to be decreased as those checkers require many chips when implemented with standard SSI ICs. This may be accomplished through the employment of MSI standard ICs or by developing a special IC for use in t.S.C checkers. Digital-flight controllers must be fast, highly-flexible and reliable. For that reason, the following researches must be guided in two directions:

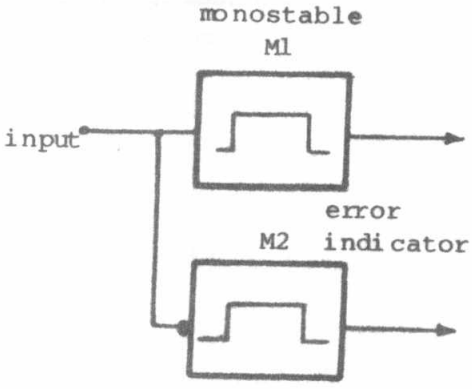
- use of microprogramming, to increase the system speed and flexibility
- application of fault-tolerance techniques to increase processor reliability, operating life and to assure proper operation during critical flight periods

REFERENCES

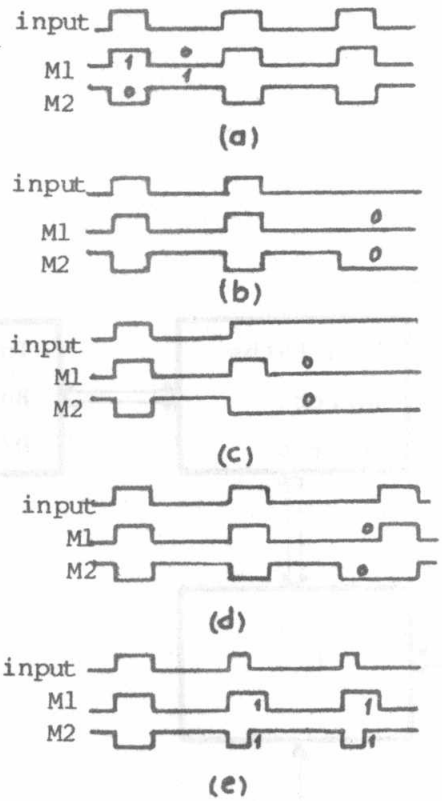
1. Cooper, A.E. and Chow. W.T., " Development of On-board Space Computer Systems", IBM Journal of Research and Development, Vol.2o,n°,1,pp 5-19, 1976.
2. Wakerly, J., " Error-Detecting Codes, Self-Checking Circuits and Applications",. North-Holland, Inc., New York (1978).
3. Anderson, D.A., and Metze, G., "Design of totally self-checking check circuits for m-out-of-n codes" IEEE Trans. Comput., Vol. C-22, No.3, pp.263-269 , March 1973.

4. Ashjaee, M.J., and Reddy, S.M. " On totally self-checking checkers for separable codes, " IEEE Trans. Comput., Vol. C-26, No. 8, pp. 737-744, August 1977.
5. Usas, A.M. " A Totally Self-Checking checker design for the detection of errors in periodic signals" IEEE Trans. Comput., Vol. C-24, No.5, pp. 483-488, May 1975.





Totally self-checking periodic-signal checker.



Checker waveforms. (a) Normal operation. (b) Input s-a-0. (c) Input s-a-1. (d) Increase In t_{off} (e). Decrease in t_{on}.

(Fig.9) Totally Self-Checking clock signal checker.

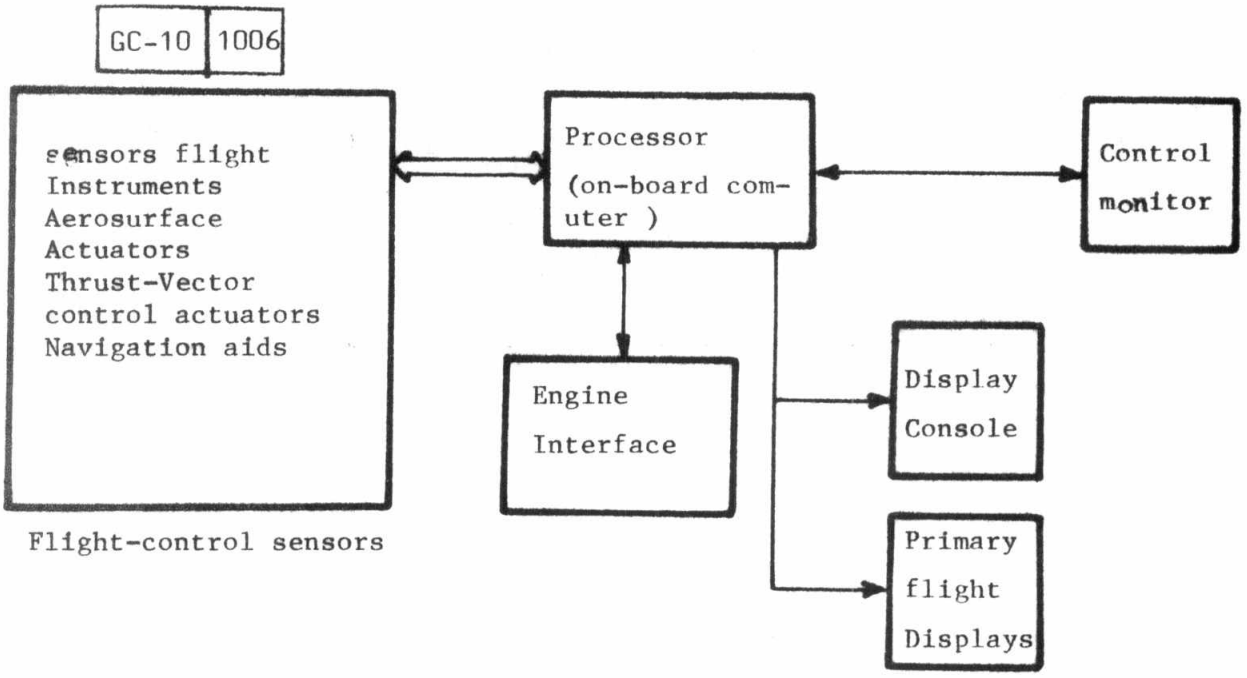


Fig. (1) Functional diagram of a Digital -Flight Controller.

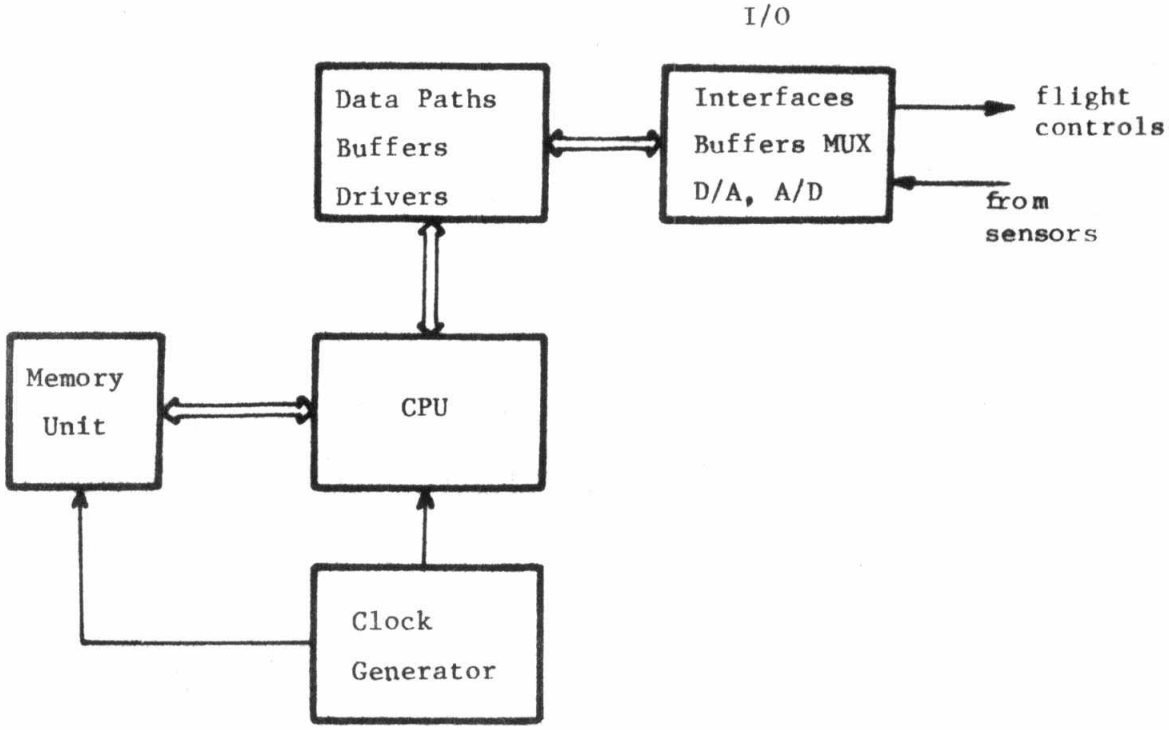


Fig.(2) Functional Units of a Processor (on-board computer).

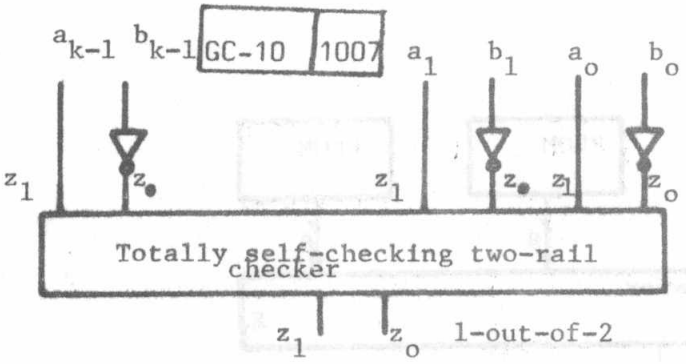


Fig.(4a) Totally self-checking equality checker.

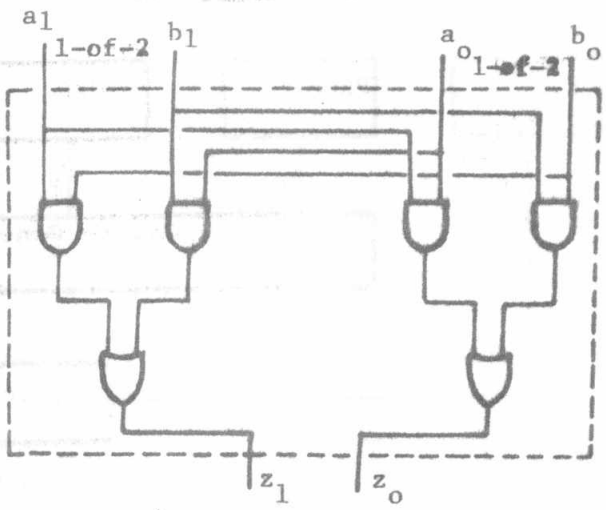


Fig.4.b Totally self-checking two-rail checker, k=2.

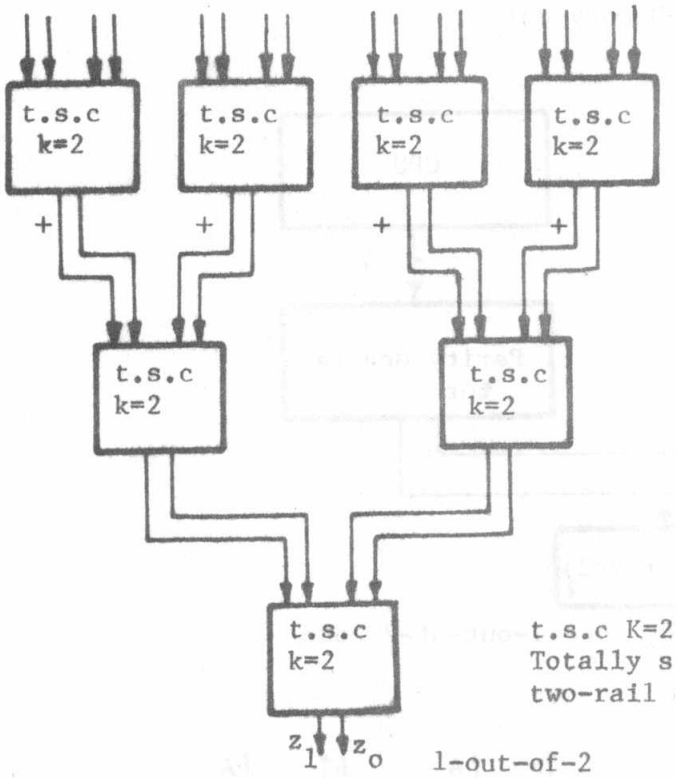


Fig.5. Totally self-checking two-rail checker , k=8.

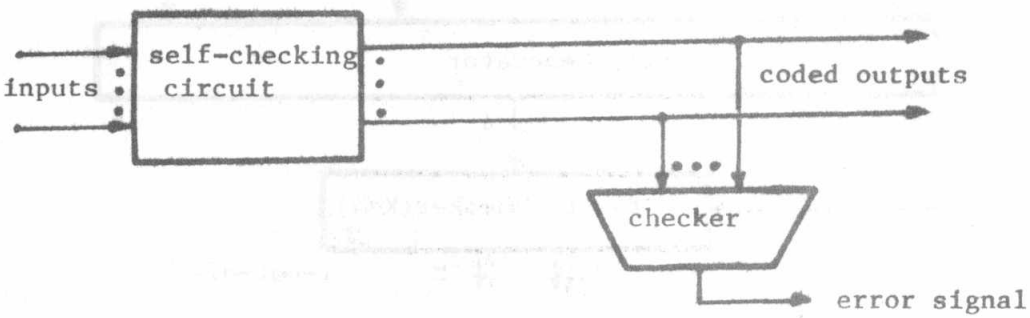


Fig. 3. Self-checking circuit and checker.

GC-10 | 1008

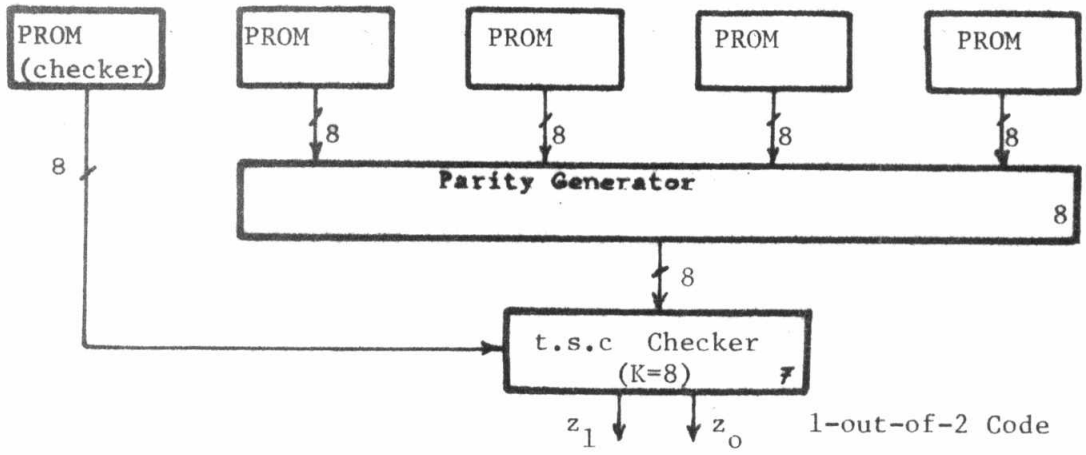


Fig.6. Self-checking Memory Unit

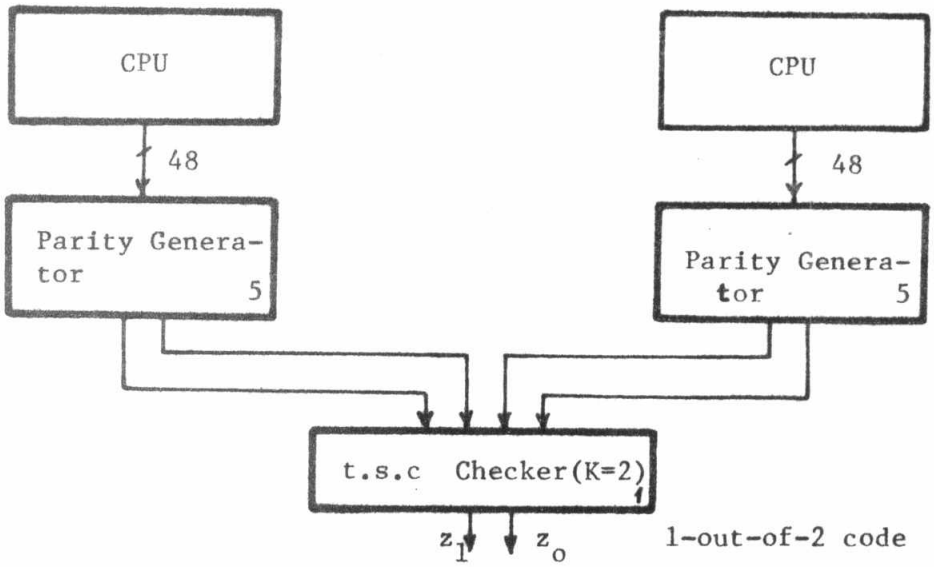


Fig.7. Self-checking CPU

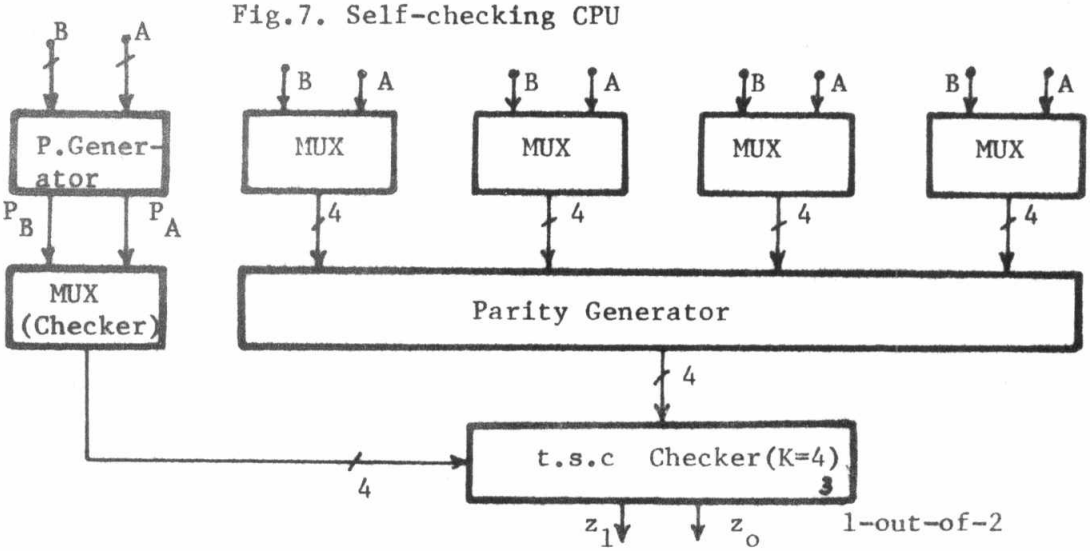


Fig.8.a Self-Checking MULTIPLEXER.

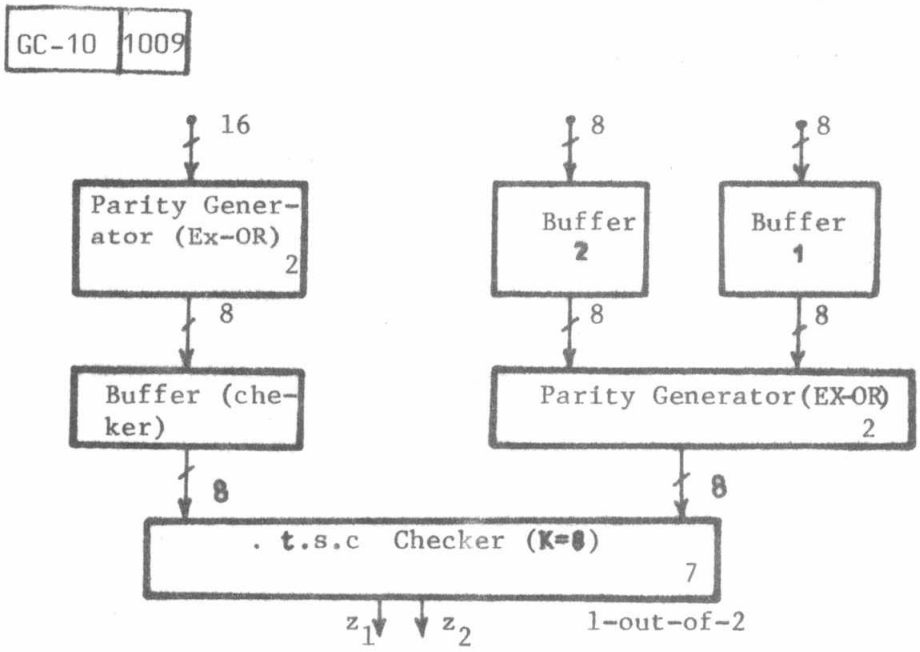


Fig.8.b. Self-checking Buffer (Register)

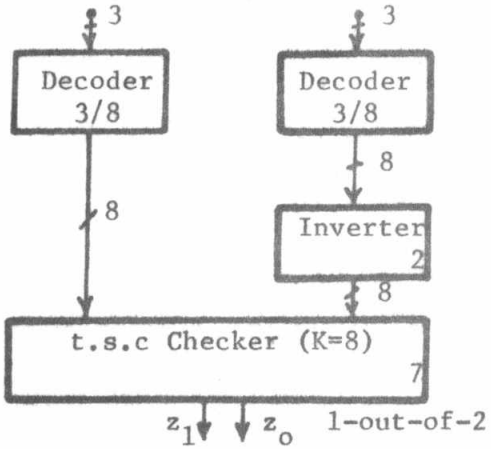


Fig.8.c Self-checking Decoder

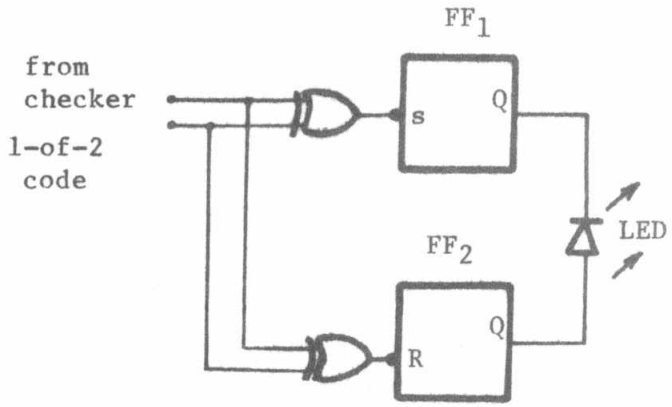


Fig.10 Fail-safe error indicator