(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 January 2006 (05.01.2006)

PCT

(10) International Publication Number
WO 2006/002069 A2

(51) International Patent Classification:
G06F 11/18 (2006.01)

(21) International Application Number: PCT/US2005/021063

(22) International Filing Date: 15 June 2005 (15.06.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/867,894 15 June 2004 (15.06.2004) US

(71) Applicant *(for all designated States except US)*: **HONEYWELL INTERNATIONAL INC.** [US/US]; 101 Columbia Road, P.O. Box 2245, Morristown, NJ 07960 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **WOLFE, Jeffrey, M.** [US/US]; 702 51st Street East Apt. 1011 A, Bradenton, FL 34208 (US). **RAMOS, Jeremy** [US/US]; 13350 US Hwy 19 North, Clearwater, FL 33764 (US). **COPENHAVER, Jason** [US/US]; 380 Bearded Oaks Circle, Sarasota, FL 34232 (US).

(74) Agents: HOIRIIS, David et al.; Honeywell International Inc., 101 Columbia Avenue, P.O. Box 2245, Morristown, NJ 07960 (US).

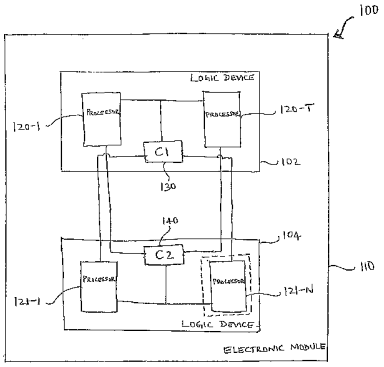
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: REDUNDANT PROCESSING ARCHITECTURE FOR SINGLE FAULT TOLERANCE



(57) Abstract: An electronic module is provided. The module includes a first logic device having at least two processors and a first comparator and a second logic device having at least one prcessor and a second comparator. Each of the at least two processors are coupled to each of first and second comparators. The first and second comparators operate as a distributed comparator system. Each comparator independently identifies faults in the processors.

REDUNDANT PROCESSING ARCHITECTURE FOR SINGLE FAULT TOLERANCE

TECHNICAL FIELD

[0001] The present invention relates generally to the field of redundant architectures, and in particular, to developing a single fault tolerant architecture.

BACKGROUND INFORMATION

[0002] Space missions require the highest upset rate immunity available to overcome Single Event Upsets (SEUs). At times, electronic systems can operate outside normal parameters thereby producing faulty data. In some circumstances, failure of these systems can be catastrophic. Although radiation hardened processors are available, they offer lower performance and higher cost than commercial off the shelf processors. In order to use non-hardened devices, a fault tolerant scheme is used in architectures that include one or more redundant systems. To identify the faulty system, voting mechanisms are used. The voting mechanisms process the simultaneous outputs of the redundant systems to determine the system producing the correct data. The voting mechanisms used in these architectures are implemented through software. However, software implementations of the voting mechanisms are very slow.

[0003] One assumption using voting mechanisms is that only one fault occurs at a time. This single fault assumption allows identification of the faulty output. In some applications three systems operate simultaneously, so that when one system fails, it can be easily identified by the other two. Essentially, the third system casts the tie-breaking vote. When only two systems are used, it is possible to identify an error, but, not which output is incorrect because a majority vote is required to determine the failed system.

[0004] Therefore, there is a need in the art for an improved architecture using off the shelf components that provide a faster solution to overcome the SEU problems in architectures.

SUMMARY OF INVENTION

[0005] Embodiments of the present invention address enhancing the speed of voting mechanism processes in systems with redundant processors.

[0006] In one embodiment a system is provided. The system includes a first logic device having at least two processors and a first comparator and a second logic device having at least one processor and a second comparator. Each of the at least two processors are coupled to each of the first and second comparators. The first and second comparators operate as a distributed comparator system. Each comparator independently identifies faults in the processors.

[0007] In another embodiment, a method for identifying a fault in a programmable device having at least three redundant processors each having a software implemented fault tolerance (SIFT) function is provided. The method includes generating an output at each of the at least three redundant processors, receiving the generated outputs at each of the SIFTs, performing a first portion of a comparison function using each of the SIFTs, performing a second portion of the comparison function using a hardware comparator, combining results of each of the first portion of the comparison function and the second portion of the comparison function and analyzing the combined results. The analysis includes performing a voting function of the combined results, comprising checking for a minority vote and determining faulty processors among the at least three redundant processors based on the minority vote.

BRIEF DESCRIPTION OF DRAWINGS

[0008] Figure 1 is a block diagram of the architecture of one embodiment of an electronic module having redundant processors and comparators.

[0009] Figure 2 is a block diagram of the architecture of one embodiment of a programmable logic device having redundant processors, software implemented fault tolerance functions and a hardware comparator.

[0010] Figure 3 is a block diagram of one embodiment of a hardware comparator.

[0011] Figure 4 is a flowchart of one embodiment of a method for accelerating a voting mechanism in a programmable device.

DETAILED DESCRIPTION

[0012] In the following detailed description of the preferred embodiments, reference is made to the accompanying drawings that from a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

[0013] Figure 1 is a block diagram of the architecture of one embodiment of an electronic module 110, indicated generally at 100 according to the teachings of the present invention. Electronic module 110 includes a single fault tolerant architecture having first and second, redundant logic devices 102 and 104. In one embodiment, the electronic module 110 is an Application Specific Integrated Chip (ASIC). In another embodiment, the electronic module 110 is a Printed Wired Assembly (PWA).

[0014] In one embodiment, logic devices 102 and 104 are field programmable gate arrays (FPGAs) that are programmed to include two processors in each. Logic device 102 includes a comparator 130 and logic device 104 includes comparator 140. Comparators 130 and 140 combine to perform the function of comparing in a distributed architecture. In one embodiment, a distributed architecture is directed to dividing or separating the comparison function into two or more units. In one embodiment, both comparators 130 and 140 perform substantially the same function and in another embodiment they complement each other and together complete the function of comparison. The distributed architecture used for comparators 130 and 140 provides an additional layer of redundancy.

[0015] In another embodiment, in order to achieve greater redundancy, the first logic device 102 has processors 120-1 through 120-T and the second logic device has processors 121-1 through 121-N. In first and second logic devices 102 and 104, processors 120-1 through 120-T and 121-1 through 121-N are programmed to perform substantially the same function. In addition to the normal system functions performed by each processor 120-1 through 120-T and 121-1 through 121-N, embodiments of the present invention have processors 120-1 through 120-T and 121-1 through 121-N generating output that is utilized to compare against each other for possible faults in any of the processors 120-1 through 120-T and 121-1 through 121-N. Both comparators 130 and 140 are coupled with each of processors 120-1 through 120-T and 121-1 through 121-N.

[0016] In operation, each output generated by processors 120-1 through 120-T and 121-1 through 121-N are compared against each other using a distributed comparator mechanism comprising comparators 130 and 140. Individual bit-level comparisons are performed and the results are then redistributed to the processors 120-1 through 120-T and 121-1 through 121-N. Since the comparison is performed in hardware it is faster than software implementations that maybe run on processors 120-1 through 120-T and 121-1 through 121-N. In one embodiment, comparators 130 and 140 are identical and hence provide additional redundancy in performing a fault identification function. In another embodiment, comparators 130 and 140 share the workload of comparing the processor outputs and thereby reduce the total processing time. Comparators 130 and 140 depend on each others results to complete the fault identification function.

[0017] In one embodiment, the electronic module 110 includes at least three redundant processors 120-1 to 120-T and 121-1 to 121-N. It is required that a minimum of three redundant processors will be necessary to determine a single faulty processor among processors 120-1, 120-T, 121-1 and 121-N. When only two processors are present, the fact that a faulty processor exists can be identified, but which processor is faulty cannot be determined. The addition of a third processor provides a tie-breaking vote in determining the faulty processor.

[0018] Figure 2 is a block diagram of the architecture of one embodiment of an electronic module 210, indicated generally at 200 according to the teachings of the present invention. In one embodiment, electronic module 210 includes one or more optional processors 220-1 to 220-M. In one embodiment, electronic module 210 operates essentially as a programmable logic device. In another embodiment, processors 220-1 through 220-M are separate processors separated from electronic module 210. In each of these embodiments processors 220-1 to 220-M are independent and are de-coupled from each other for the architecture to be radiation hardened.

[0019] In one embodiment, the electronic module 210 includes at least three redundant processors 220-1 to 220-M. It is required that a minimum of three redundant processors will be necessary to determine a single faulty processor among processors 220-1, 220-2 and 220-3. In the event there are only two processors 220-1 and 220-2 present, the fact that a faulty processor exist can be identified, however, which one of the two processors 220-1 and 220-2 is faulty cannot be determined. The addition of a third processor will provide the tie-breaker vote in determining the faulty processor.

[0020] In Figure 2, processors 220-1 to 220-M perform substantially the same function. Each of processors 220-1 to 220-M includes a software implemented fault tolerance (SIFT) function represented by 222-1 to 222-M. SIFT 222-1 through 222-M is a programming interface that has a library set to perform the function of comparison, management, voting and reset functions with regard to the processors 220-1 through 220-M. SIFT 222-1 through 222-M supports the detection of a single event error of one of the processors 220-1 through 220-M. In one embodiment, the operation of SIFT functions 222-1 through 222-M are de-coupled to perform independently from the operations of the processors 220-1 through 220-M. The comparison is performed at some application (e.g., SIFT) designated checkpoint. Each SIFT 222-1 through 222-M performs its tasks at some pre-determined application period that is specified by the designer. In one embodiment, also included in the electronic module 210 is a hardware comparator 230 that is implemented in the remaining space of the electronic module 210. Hardware comparator 230 reduces the time and effort to perform the comparison function executed by each of the SIFTs 222-1 through 222-M and relieves some of the burden of executing the comparison function for SIFT 222-1 through 222-M. In one embodiment, the hardware comparator 230 is radiation hardened by using a triple modular redundancy (TMR) implementation.

[0021] In one operation, the SIFT software 222-1 through 222-M is executed on each processor. As the software is being executed, a portion of the comparison function is forwarded to the hardware comparator 230 to perform the bit-level comparison in hardware. The hardware comparator 230 enhances SIFT 222-1 through 222-M in performing the comparison faster through hardware. After the completion of the comparison task, the hardware comparator 230 forwards the results to SIFT 222-1 through 222-M. Upon completion of the execution of the SIFT software, the results are forwarded to each of the processors 220-1 through 220-M. In one embodiment, each SIFT 222-1 through 222-M sends a message to the faulty processor among processors 220-1 through 220-M to initialize or reset the processor. In one embodiment, the faulty processor may also receive a copy of the current image of one of the other processors 220-1 through 220-M that would enable it to continue its duties as required.

[0022] Figure 3 is a block diagram of one embodiment of a hardware comparator, indicated generally at 300. In one embodiment, hardware comparator 300 is as found in Figure 2 above labeled as 230. Hardware comparator 300 comprises a binary comparator 310, control logic 320, selector 335 and a broadcaster 340. In one embodiment, binary comparator 310 is coupled to a control logic device 320 which in turn is coupled to a selector 330.

[0023] In operation, each of SIFT 222-1 through 222-M generates outputs C_1-C_M. Outputs C_1 through C_M are received by a binary comparator 310. The binary comparator 310 performs a bit-level comparison to detect any change in bit positions between processor outputs thereby determining if there is a faulty or failed processor. The result of the bit comparison in binary comparator 310 is forwarded to control logic device 320. Control logic device 320 generates a control signal based on the comparison results. The control signal from control logic device 320 triggers selector 330. In the event of a failed processor, a control signal from control logic 330 triggers selector 330 to choose an output other than the failed output to be sent to broadcaster 340. Broadcaster 340 broadcasts the chosen signal back to all the processors 120-1 through 120-T, 121-1 through 121-N and 220-1 through 220-M and the failed processor is reset consequently.

[0024] Figure 4 is a flow chart of one embodiment of a method for accelerating a voting mechanism in an electronic module as shown generally at 400, according to the teachings of the present invention. Method 400 is performed by an electronic module such as electronic module 210 shown in Figure 2 above. The electronic module includes a hardware comparator and at least three or more redundant systems each having a software implemented fault tolerance function. The method begins at block 410 where each of the redundant processors generates an output for analysis. At block 420, each of the outputs from the redundant processors are received by the SIFT. A portion of the output is sent to a hardware comparator in block 430. The method then proceeds to block 440 and performs a comparison process in the hardware comparator. The method proceeds to block 450 and the results of the comparison process are retrieved by the SIFT. In block 460 the results are checked to determine if there is a minority vote. When there is no minority vote the method proceeds to block 490 where it is terminated. The presence of no minority vote would indicate to all of the processors that there are no faulty processors present.

[0025] Alternatively, when there is a minority vote, the method proceeds to block 470. At block 470, the failed processors are identified, assuming that the number of failed processors is less than the number of properly working processors. The method then proceeds to block 480 where the identified failed processors are initialized or corrective action is performed on it. Following block 480, the method terminates at block 490.

CONCLUSION

[0026] Embodiments of the present invention have been described. The embodiments provide a redundant architecture that can increase the speed of the voting mechanism. Ordinarily, the voting mechanism is performed using a software process. This invention reduces the time taken to perform the voting process and also reduces the effort of the software by transferring critical bit comparison operation and performing it in a hardware device on the same programmable device. The electronic module of the described invention can be assembled using many off the shelf components in providing a faster solution to overcome SEU problems. This is particularly advantageous for overcoming SEU problems in architectures used in space applications.

[0027] Although specific embodiments have been illustrated and described in this specification, it will be appreciated by those of ordinary skill in the art that any arrangement that is calculated to achieve the same purpose may be substituted for the specific embodiment shown. This application is intended to cover any adaptations or variations of the present invention.

CLAIMS

What is claimed is:

1. A system comprising:

a first logic device having at least two processors and a first comparator; and

a second logic device having at least one processor and a second comparator;

wherein each of the at least two processors are coupled to each of the first and second comparators;

wherein the first and second comparators operate as a distributed comparator system;

wherein each comparator independently identifies faults in the processors.

2. The system of claim 1, wherein the first and second logic devices are programmable.

3. The system of claim 1, wherein each of the first and second comparators is a hardware device.

4. A logic device comprising:

at least three processors that are adapted to perform substantially the same function;

at least one hardware comparator;

wherein each of the at least three processors includes a software implemented fault tolerance function (SIFT);

wherein each of the at least three SIFTs is communicatively coupled to each of the hardware comparators; and

wherein each of the SIFTs operates independently of each other to identify faults in one or more of the at least three processors.

5. A system comprising:

a logic device having at least three processors; and

at least one comparator coupled to each of the at least three processors;

wherein each processor includes a software implemented fault tolerance (SIFT) function;

wherein the SIFT and the at least one comparator in combination processes outputs of the at least three processors;

wherein the SIFT determines if one or more of the at least three processors has failed.

6. The system of claim 5, wherein the logic device is programmable.

7. The system of claim 5, wherein the at least one comparator is a hardware device.

8. A method for identifying a fault in a programmable device having at least three redundant processors each having a software implemented fault tolerance (SIFT) function, the method comprising:

generating an output at each of the at least three redundant processors;

receiving the generated outputs at each of the SIFTs;

performing a first portion of a comparison function using each of the SIFTs;

performing a second portion of the comparison function using a hardware comparator;

combining results of each of the first portion of the comparison function and the second portion of the comparison function; and

analyzing the combined results, wherein the analysis comprises:

performing a voting function of the combined results, comprising checking for a minority vote; and

determining faulty processors among the at least three redundant processors based on the minority vote;

9. The method of claim 8, further comprising initializing the faulty processor when a minority vote is achieved.

10. A method for accelerating a voting mechanism in a programmable device having at least three redundant processors each having a software implemented fault tolerance (SIFT) function, the method comprising:

generating an output in each of the at least three redundant processors;

receiving the generated outputs at each of the SIFTs;

performing a first portion of a comparison function using each of the SIFTs;

performing a second portion of the comparison function using a hardware comparator; and

retrieving results of each of the first portion of the comparison function and the second portion of the comparison function.

11. An electronic module, comprising:

a first logic device having at least two processors and a first comparator; and

a second logic device having at least one processor and a second comparator;

wherein each of the at least two processors are coupled to each of the first and second comparators;

wherein the first and second comparators operate as a distributed comparator system;

wherein each comparator independently identifies faults in the processors;

wherein the logic devices are combined on a single chip.

12. The electronic module of claim 11, wherein the first and second logic devices are programmable.

13. A machine readable medium having instructions stored thereon for identifying a fault in a programmable device having at least three redundant processors each having a software implemented fault tolerance (SIFT) function, the method comprising:

generating an output at each of the at least three redundant processors;

receiving the generated outputs at each of the SIFTs;

performing a first portion of a comparison function using each of the SIFTs;

performing a second portion of the comparison function using a hardware comparator;

combining results of each of the first portions of the comparison function and the second portion of the comparison function; and

analyzing the combined results, wherein the analysis comprises:

performing a voting of the combined results comprising checking for a minority vote; and

determining faulty processors among the at least three redundant processors based on the minority vote;

14. The method of claim 13, further comprising initializing the faulty processor when a minority vote is achieved.

15. A machine readable medium having instructions stored thereon for accelerating a voting mechanism in a programmable device having at least three redundant processors each having a software implemented fault tolerance (SIFT) function, the method comprising:

generating an output in each of the at least three redundant processors;

receiving the generated outputs at each of the SIFTs;

performing a first portion of a comparison function using each of the SIFTs

performing a second portion of the comparison function using a hardware comparator; and

retrieving results of each of the first portion of the comparison function and the second portion of the comparison function.

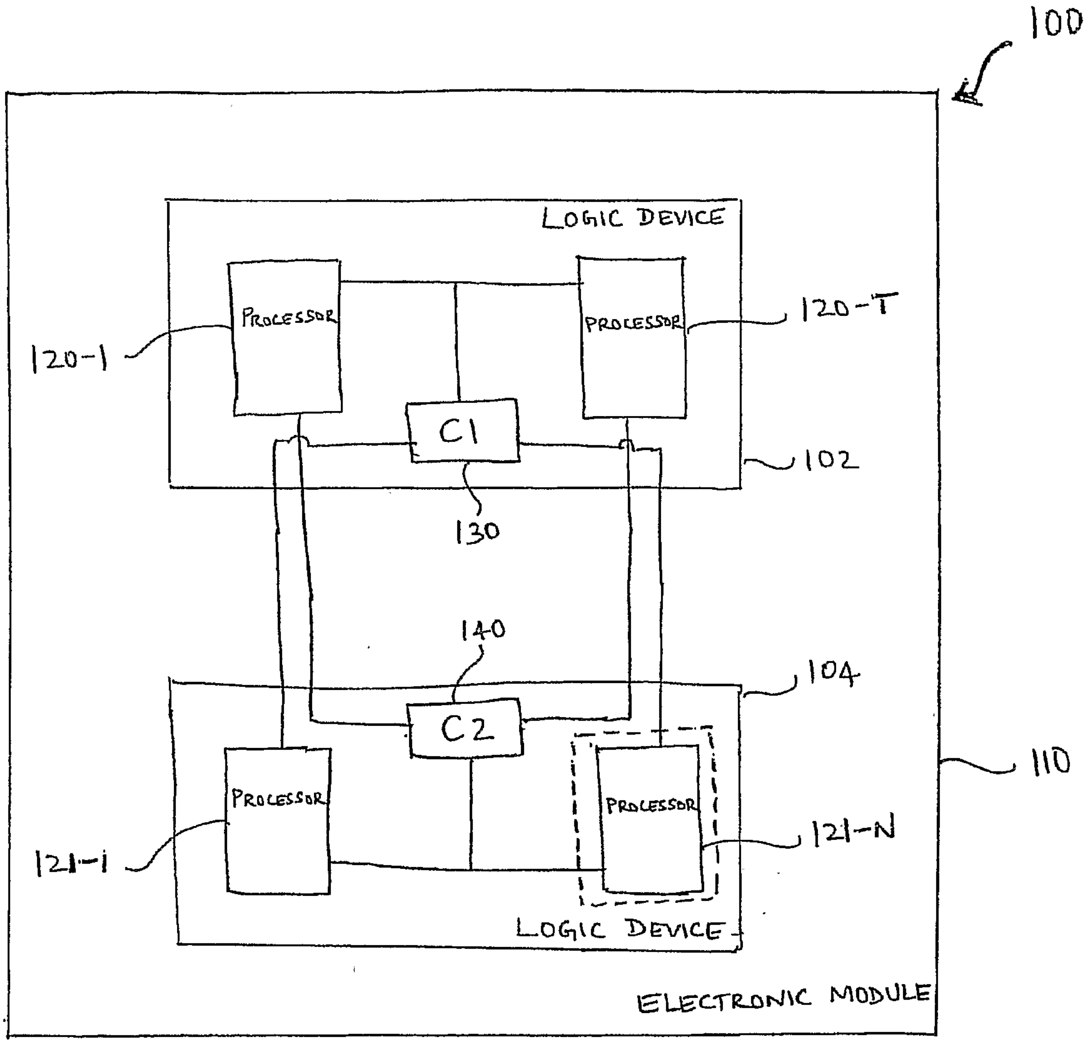

100
LOGIC DEVICE
PROCESSOR
PROCESSOR
120-1
120-T
C1
102
130
140
104
C2
110
PROCESSOR
PROCESSOR
121-N
121-1
LOGIC DEVICE
ELECTRONIC MODULE

Fig. 1

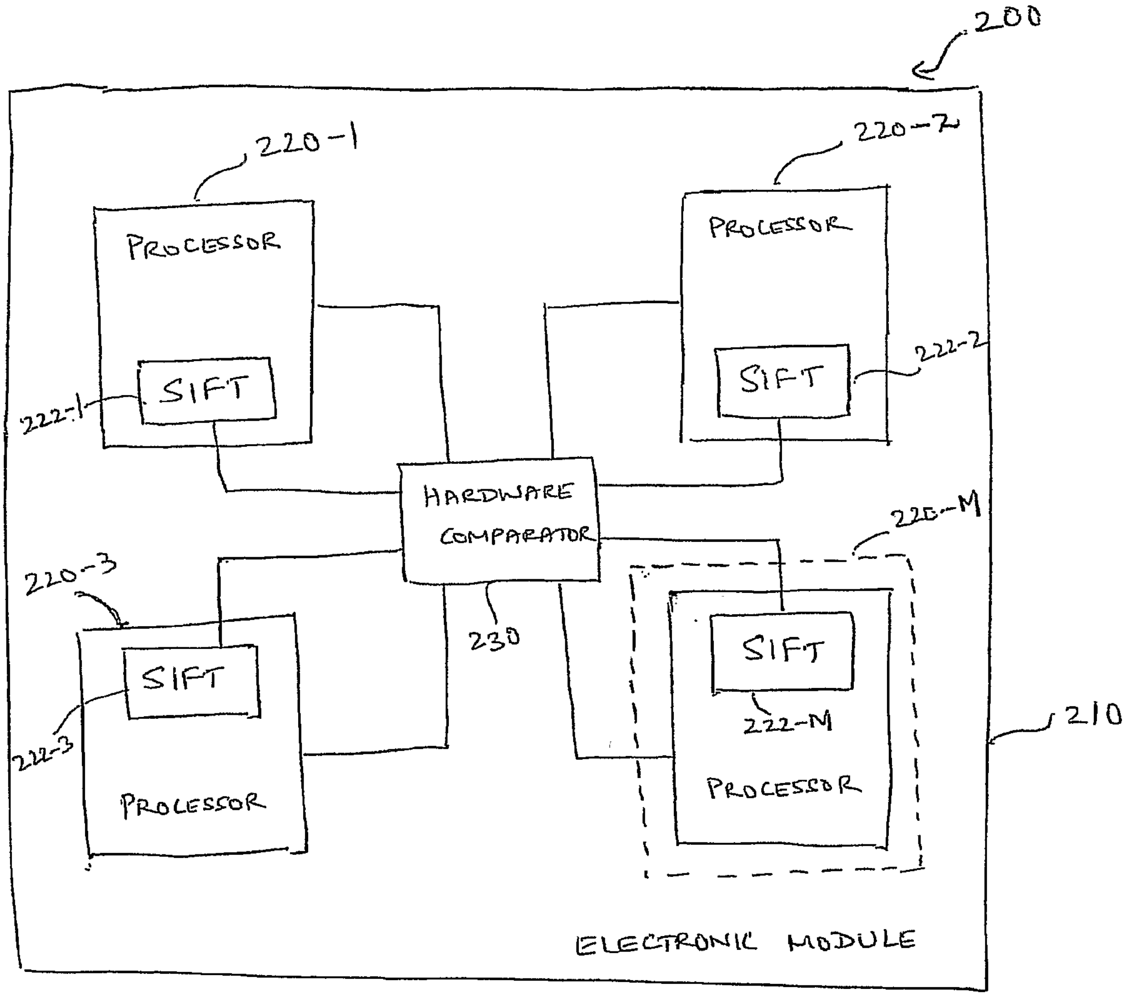

200
220-1
PROCESSOR
SIFT
222-1
220-2
PROCESSOR
SIFT
222-2
HARDWARE COMPARATOR
230
220-M
220-3
SIFT
222-3
PROCESSOR
SIFT
222-M
PROCESSOR
210
ELECTRONIC MODULE

Fig. 2

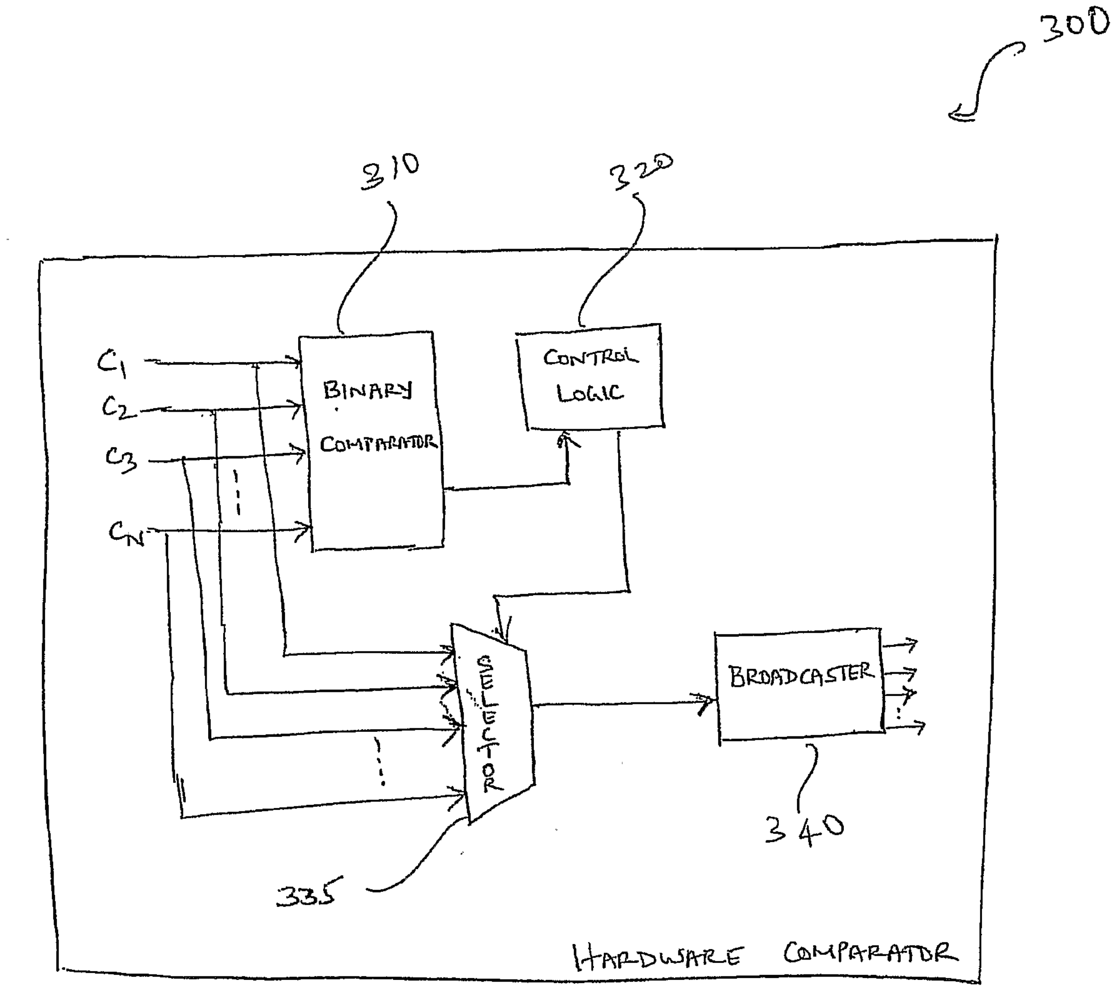

300
310
320
C1
C2
C3
CN
BINARY COMPARATOR
CONTROL LOGIC
SELECTOR
BROADCASTER
335
340
HARDWARE COMPARATOR

Fig. 3

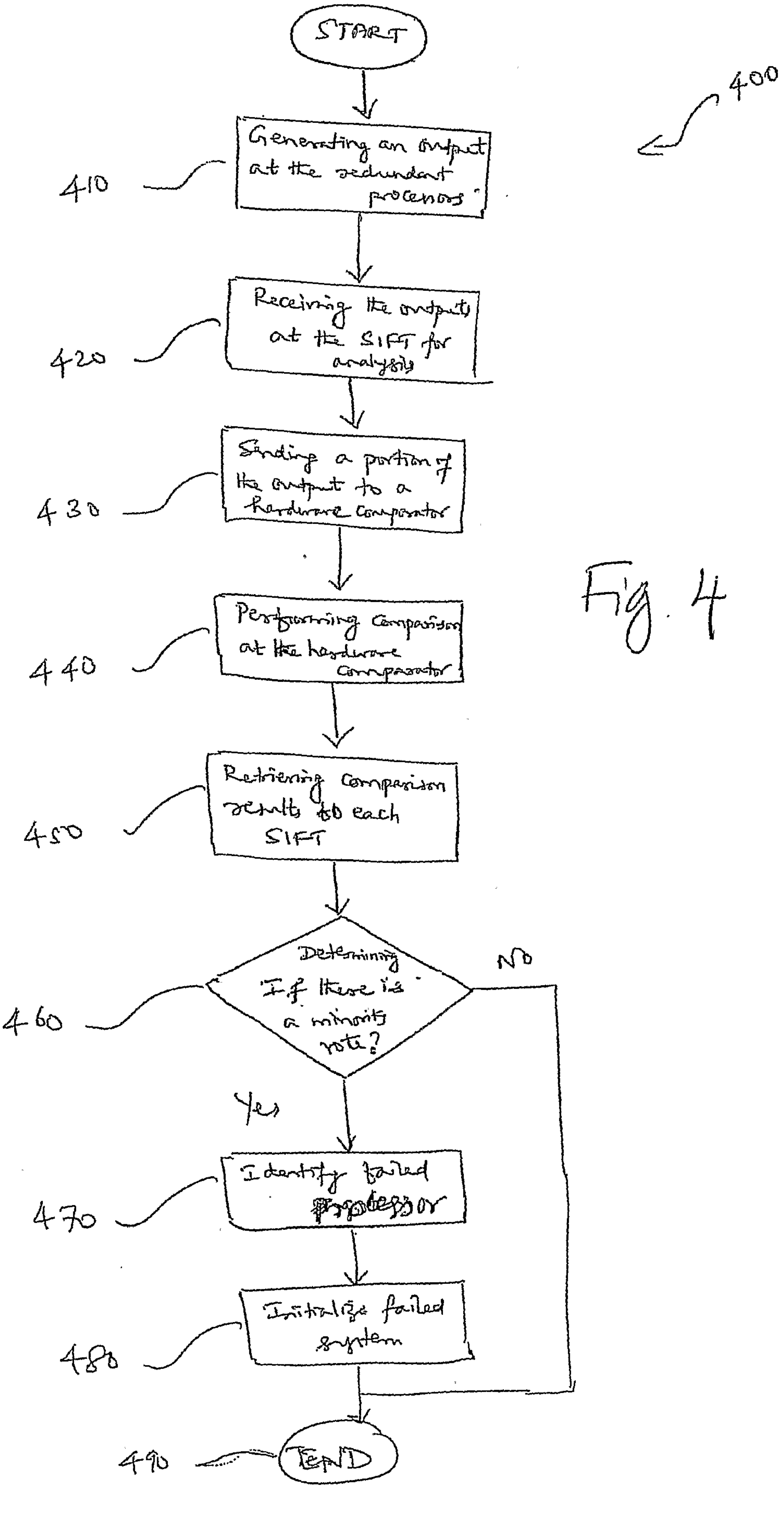

START
400
Generating an output at the redundant processors
410
Receiving the outputs at the SIFT for analysis
420
Sending a portion of the output to a hardware comparator
430
Performing comparison at the hardware comparator
440
Returning comparison results to each SIFT
450
Determining If there is a minority vote?
460
NO
Yes
Identify failed processor
470
Initialize failed system
480
END
490

Fig. 4