Fault-Tolerant Computer Architecture Based on INMOS Transputer Processor

Final Report

NASA/ASEE Summer Faculty Fellowship Program --1987

Johnson Space Center

Prepared by:	Jorge L. Ortiz, Ph.D.
Academic Rank:	Associate Professor
University & Department:	Electrical and Computer Engineering Department University of Puerto Rico Mayaguez,P. R. 00709-5000
NASA/JSC	
Directorate:	Engineering and Development
Division:	Avionic Systems
Branch:	Flight Data Systems
JSC Colleague:	Michael M. Thomas
Date:	August 7, 1987
Contract Number:	NGT 44-001-800

ABSTRACT

Redundant processing has been used for several years in mission flight systems. In these systems, more than one processor performs the same task at the same time but only one processor is actually in real use. A fault-tolerant computer architecture based on the unique features provided by INMOS Transputers is presented in this report. The Transputer architecture provides several communication links that allow data and command communication with other Transputers without the use of a bus. Additionally the Transputer allows the use of parallel processing to increase the system speed considerably.

The processor architecture consists of three processors working in parallel keeping all the processors at the same operational level but only one processor is in real control of the process. The design allows each Transputer to perform a test to the other two Transputers and report the operating condition of the neighbor processors. A graphic display has been developed to facilitate the identification of any problem by the user.

I. Introduction

The concept of redundant processing has been used in the space for long time specially for critical maneuvers like landing or launching. In these cases, several processors had been working in parallel performing the same task but only one of then is in real control of the process. If something goes wrong with this computer the system operator or astronaut can switch the operation to another processor.

Recently, the C. S. Draper Laboratory designed a fault-tolerant processor called Advance Information Processing System (AIPS) with the concept of maintaining three processors (or more) working redundantly and testing each other to "vote" on the status of the other processors. In this fashion, the user has the information about the system performance on real time. This system is linked by a data communication bus called the Inter-Computer Bus (IC) for communication between processors and other I/O devices.

A fault-tolerant computer architecture based on the unique features provided by INMOS Transputer has shown to be an adequate alternative to this kind of processor. Among the characteristics that can improve the design of the processor are the serial communication links that allow data and command communications with other Transputers without the use of a bus, and the capability of parallel processing to increase the system speed. Therefore, a Transputer Fault-Tolerant Processor (TFTP) designed based on the Transputers could mean a faster more reliable processor.

Discussion and Results

The first objective of this research was to design a fault-tolerant processor with a parallel architecture based on the INMOS Transputer. Two solutions to this problem were presented at the moment. The first one, presented by Mr. Dennis Taylor, uses four T414 transputers with three of then working in parallel an the remaining one will be the coordinator as shown in Figure 1. Three parallel processors will perform the same task while the coordinator will compare the results of the operations and report to the user if it finds a fault in one of the processors. This architecture is very efficient and is easy to keep control on the parallel processors but the possibility exists that the fault could come from the coordinator itself. The second architecture, shown in Figure 2, consist of keeping three processors (or more) working in parallel. All of then will be kept in the same level, but only one of then will perform the real operation. Each of then will keep performing tests to the other two processors and reporting to the operator the results of these tests. The three Transputers are interfaced to the Transputer Development System (TDS) in an IBM AT compatible. The design allows each Transputer to perform an evaluation to the other two Transputers and report the operating condition (based on that test) of the neighbor Transputers. The test consists of sending an integer constant to a processor and the processor under test will return its square value. This result is analyzed and compared with the previously known solution to làter send a report to the host computer. At this moment, three Transputers are running in parallel, performing the indicated test, and finally showing its report on the screen.

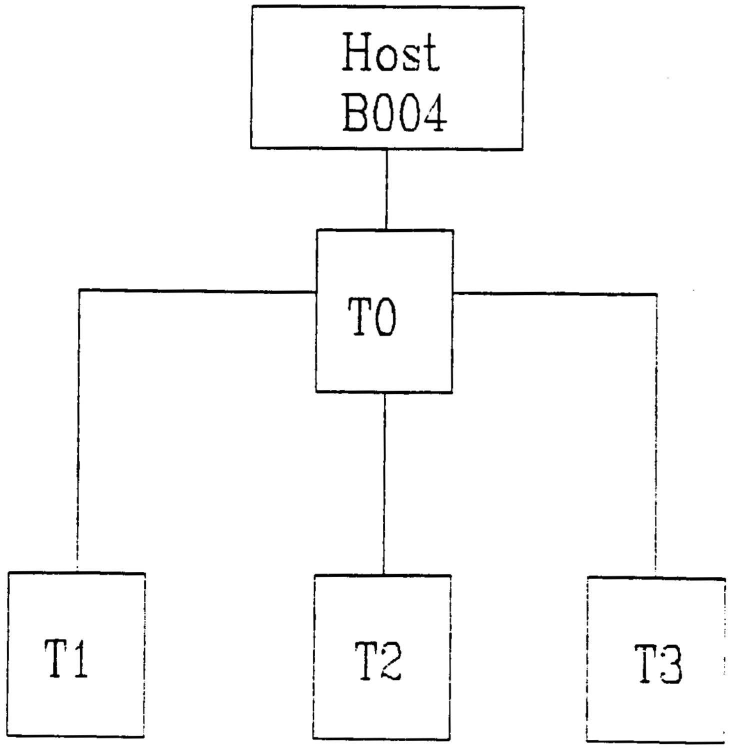


Figure 1. Fault-Tolerant Processor With a Coordinator Processor

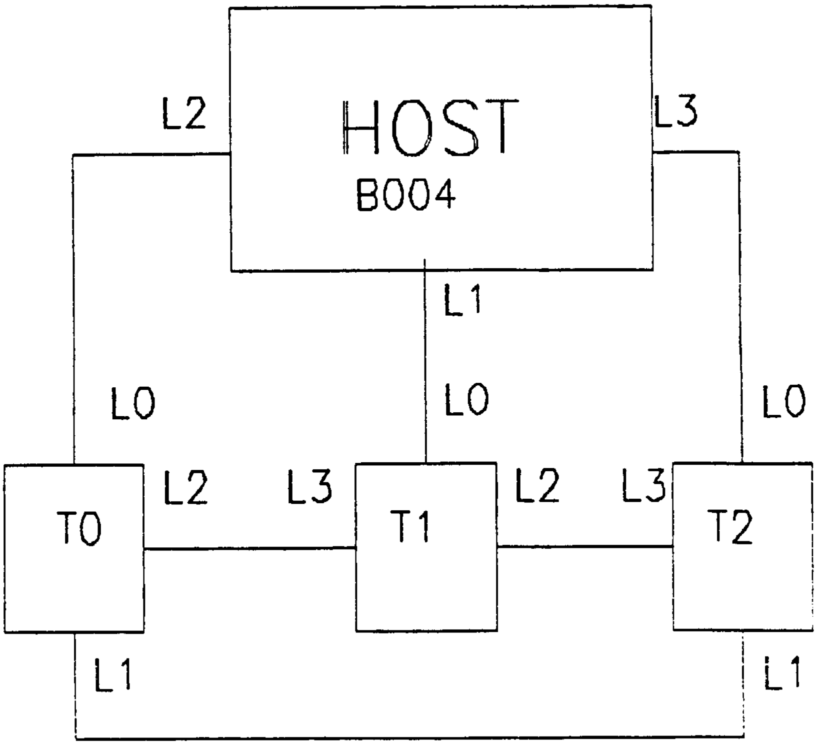


Figure 2. Transputer Fault-Tolerant Processor Architecture

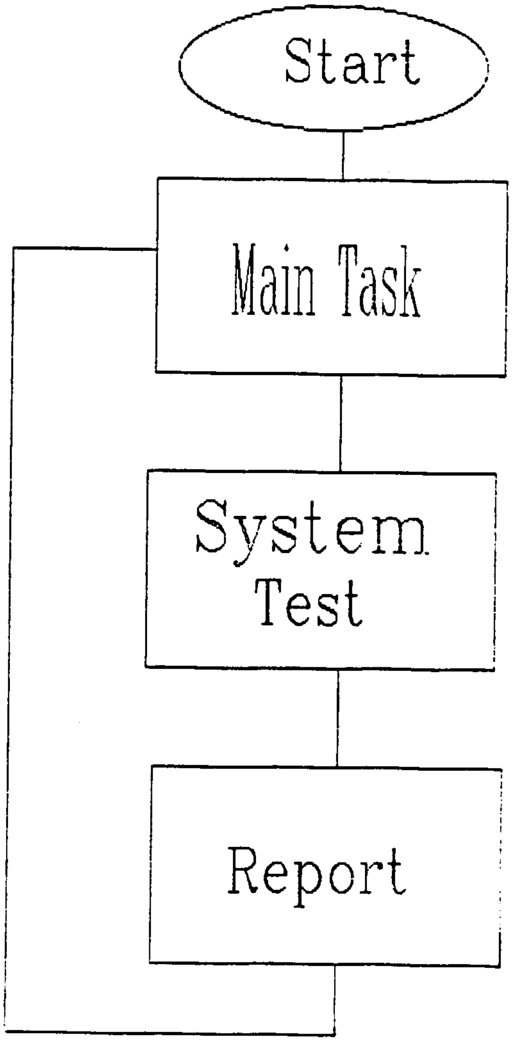


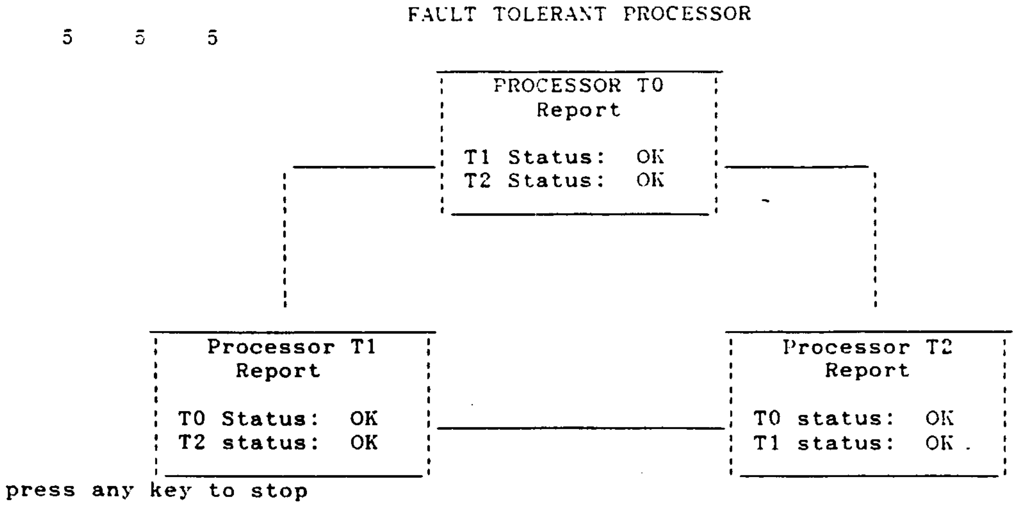
Figure 3. Program Flow Chart

The three processors are kept executing the same software sequence at the same time. As shown in Figure 3, the processors start executing the main task followed by the system testing algorithm and finally ending the sequence with the system status report that is sent to the host processor. The software to implement the sequence was written in Occam.

A graphic display was developed to facilitate the identification of any problem by the user. The system shows a constantly updated screen detailing the status of each processor and the result of the tests performed between the processors. Figure 4 illustrates the screen graphic display when all processors are on line in normal operation, as it shows processor T0 is reporting on the screen the status of T1 and T2 (the other two processors) and since everything is normal at the moment these processors are "ok".

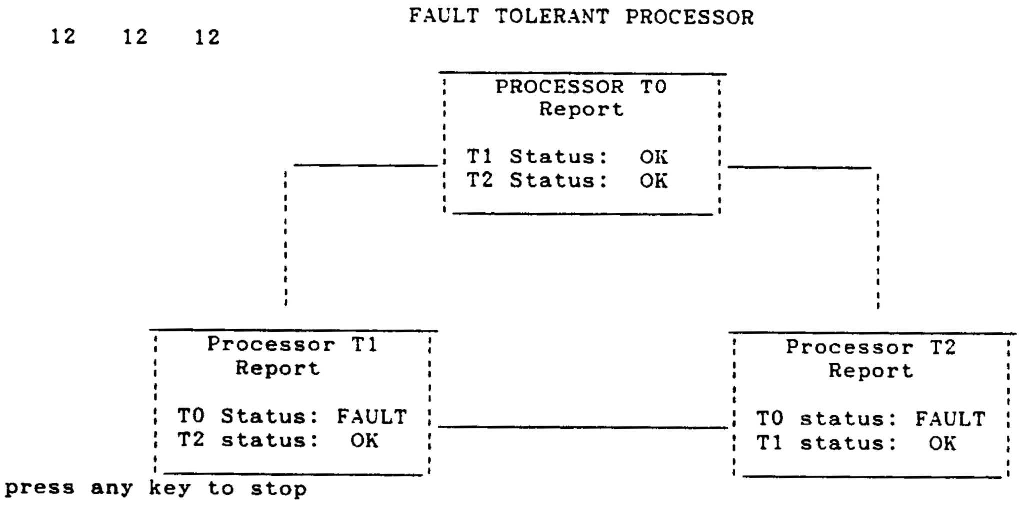
One of the faults that the processor can detect is a software fault, where the processor on test for some reason does not get the correct answer for a numeric operation. As it can be seem on Figure 5 processors T1 and T2 found that processor T0 has a software fault and they display the occurrence of that fault on the status of processor T0. A similar software fault is simulated in processor T1 and the results are shown in Figure 6.

Another fault that can be simulated in this system, is a communication fault or hardware fault. As shown in Figure 7 a fault has occurred in a link at point a, and Figure 8 shows that due to this fault the host processor could not receive the status report from processor T2. Also, processors T0 and T2 acknowledge that a fault has been detected on the mentioned processor.



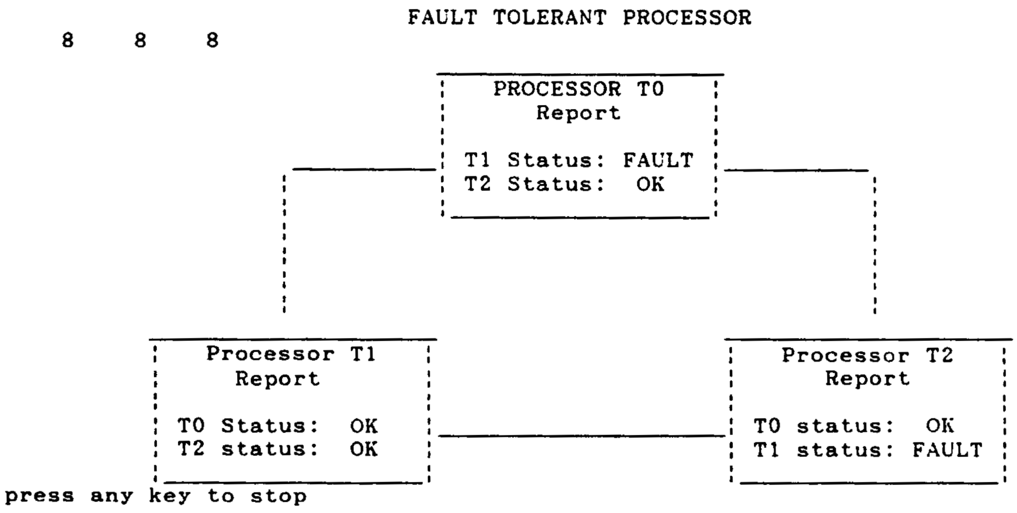
```
                              FAULT TOLERANT PROCESSOR
        5     5     5
                             ______________________
                            |   PROCESSOR T0       |
                            |      Report          |
                            |                      |
               _____________| T1 Status:   OK      |_____________
              |             | T2 Status:   OK      |             |
              |             |______________________|             |
              |                                                  |
              |                                                  |
              |                                                  |
     ______________________                        ______________________
    |   Processor T1       |                      |   Processor T2       |
    |     Report           |                      |     Report           |
    |                      |                      |                      |
    | T0 Status:   OK      |______________________| T0 status:   OK      |
    | T2 status:   OK      |                      | T1 status:   OK .    |
    |______________________|                      |______________________|
press any key to stop
```

Figure 4. Transputer Fault-Tolerant Processor in Normal Operation



```
                                 FAULT TOLERANT PROCESSOR
  12    12    12
                                 PROCESSOR T0
                                    Report

                                T1 Status:  OK
                                T2 Status:  OK


          Processor T1                                Processor T2
             Report                                      Report

       T0 Status: FAULT                            T0 status: FAULT
       T2 status:  OK                              T1 status:  OK

press any key to stop
```

Figure 5. Software Fault in Processor T0



```
                              FAULT TOLERANT PROCESSOR
     8    8    8
                                 ____________________
                                | PROCESSOR T0       |
                                |    Report          |
                                |                    |
                   _____________| T1 Status: FAULT   |_____________
                  |             | T2 Status:   OK    |             |
                  |             |____________________|             |
                  |                                                |
                  |                                                |
                  |                                                |
          ____________________                        ____________________
         | Processor T1       |                      | Processor T2       |
         |    Report          |                      |    Report          |
         |                    |                      |                    |
         | T0 Status:   OK    |______________________| T0 status:   OK    |
         | T2 status:   OK    |                      | T1 status: FAULT   |
         |____________________|                      |____________________|
press any key to stop
```

Figure 6. Software Fault in Processor T1

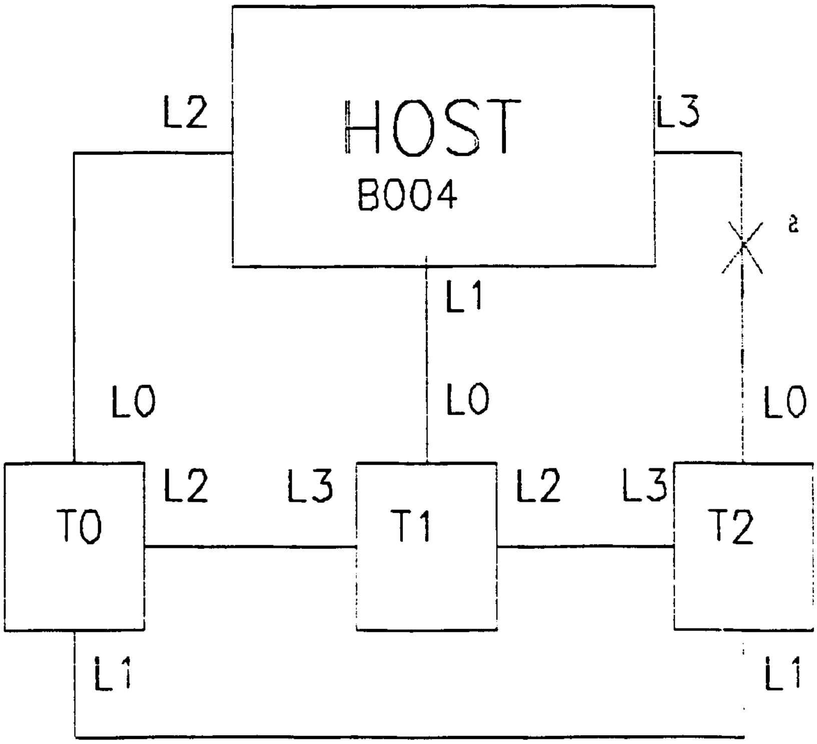
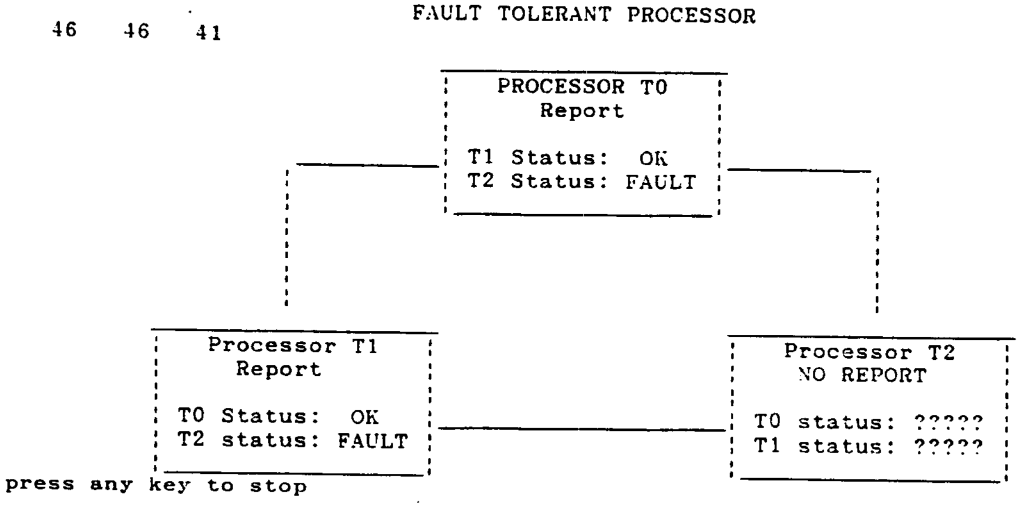


Figure 7. Fault in a Link at Point a.



```
46    46    41              FAULT TOLERANT PROCESSOR

                          PROCESSOR T0
                             Report

                         T1 Status:  OK
                         T2 Status: FAULT

     Processor T1                              Processor T2
        Report                                  NO REPORT

  T0 Status:  OK                          T0 status: ?????
  T2 status: FAULT                        T1 status: ?????
press any key to stop
```

Figure 8. Hardware or Communication Fault in Processor T2.

Conclusions

The Transputer Fault-Tolerant Processor has shown to be an excellent alternative when a reliable processor is needed. More research has to be done to improve link communications, its synchronization, and link resetting after a hardware fault occurs. However the TFTP is potentially faster than other fault-tolerant processors due to the Transputer parallel processing capacity and its specially designed Occam language to facilitate concurrent processing.

References

[1]. Daniel P. Siewiorek, Robert S. Swary, "The Theory and Practice of Reliable System Design," Digital Press, c 1982.

[2]. Jacob A. Abraham, et al,"Fault Tolerance Techniques for Systolic Arrays", IEEE Trans. Computers, July 1987, pp. 65-74.

[3]. Roger Allan,"Technology Report for Fault-Tolerant Computing, Software is Finding a Powerfull Ally in Hardware",Electronic Design,October 1985,pp.111-116.

[4]. L. Beaudet, F. Eshragh, "System-level diagnostics Troubleshoot Multiprocessors", Computer Design, July 1987, pp. 77-82.

[5]. J. Bond,"Parallel-Processing Concepts Finally Come Together in Real Systems",Computer Design,June 1987, pp.51-74.

[6]. Advanced Information Processing System (AIPS), Demostration Notes, The Charles Stark Draper Laboratory,Inc,Cambridge,Massachusetts.