Simulated Fault Injection: A Methodology to Evaluate Fault Tolerant Microprocessor Architectures

Gwan S. Choi
University of Illinois, Urbana-Champaign
Ravishankar K. Iyer, Senior Member IEEE
University of Illinois, Urbana-Champaign
Victor A. Carreno
NASA Langley Research Center, Hampton

Key Words — **Fault-tolerance, Validation, Simulation, Fault injection, Experimental analysis**

Reader Aids —
Purpose: Widen the state of the art
Special math needed for explanations: None
Special math needed to use results: None
Results useful to: Theoreticians and analysts

Abstract — **This paper describes a simulation-based fault injection methodology to validate fault tolerant microprocessor architectures. The approach uses mixed-mode simulation (electrical/logic analysis), and injects transient errors in run-time, to assess the resulting fault-impact. To exemplify the methodology, a fault tolerant architecture which models the digital aspects of a dual channel, real-time jet engine controller is used. The level of effectiveness of the dual configuration to single and multiple transients is measured. The results indicate 100% coverage of single transients. Approximately 12 percent of the multiple transients affect both channels; none result in controller failure since two additional levels of redundancy exist.**

1. INTRODUCTION

In recent years, there has been a rapid increase in the use of digital systems to control critical avionic functions. Naturally, this has led to concerns regarding the dependability of these systems. A particular source of concern is the impact of transients which are common in avionic environments. Measurements on ground-based digital systems [2, 11, 22], show that over 85 percent of all computer failures can be attributed to transients. A study of transient fault impact is thus essential for defining the vulnerability of digital systems.

This paper discusses an experimental methodology for simulation-based validation of fault tolerant microprocessor architectures. The approach is intended to investigate critical aspects of such designs from a fault-tolerance viewpoint. The method is illustrated via an example of a fault tolerant jet-engine controller. In particular, the digital aspects of the dual-channel controller, described at the logic and functional levels, are simulated and transient fault injections are performed. The coverage of the dual technique to single and multiple transients is evaluated.

In the simulated controller, fault detection and reconfiguration are performed through transactions over communication links. Instructions specifically designed to exercise this cross-channel communication are executed. The simulated fault injection approach is illustrated by measuring the level of effectiveness of the dual configuration to transient errors. The results show that none of the single injections affect more than one channel while approximately 12 percent of the multiple injections affect both channels.

The next section discusses the related research in this area. Section 3 contains the description of the experimental environment and section 4 describes the simulated controller. Section 5 describes the experiment and quantifies the impact of single and multiple transients errors; concluding remarks appear in section 6.

2. RELATED RESEARCH

Several researchers have investigated the impact of transients in computer systems. An early study of failures in digital systems reported in [2] showed that nearly 90 percent of failures were transient in nature. More recent studies using failure data from IBM and DEC systems [11, 22] also show that over 85% of all computer failures are due to transient problems. This research also indicates a strong relationship between the occurrence of transients and the level of system activity.

Device-level analyses, of the mechanisms of logic upsets, have been in progress for quite some time. The hazards of logic upsets in dynamic RAM's were first reported in [15] wherein the behavior of alpha-particle induced soft errors was explored. In [21], a simulation technique for modeling the ion shunt effect was developed. An approximate analytical model for a current transient was proposed in [16].

At the system level, a series of experiments aimed at error analysis through the physical and simulated insertion of faults were conducted by several investigators associated with NASA AIRLAB. An experiment to study fault latency distributions through hardware fault injections is described in [23]. An investigation of fault propagation in microprocessors is discussed in [12, 14]. This analysis quantified the dependency of the measured error-propagation on the location of the fault and, on the type of instruction/micro-instruction activity. In [6, 19] techniques to determine the efficiency of error-detection mechanisms are described.

At the microprocessor level, studies have primarily focused on vulnerability assessment, and on evaluating the efficiency of error detection methods. An assessment of different transient-error test methods is discussed in [13]. In [7], a detailed analysis of the vulnerability of the Z80 microprocessor based on ion-bombardment testing is described. The development of

0018-9529/90/1000-0486$01.00©1990 IEEE

a state-transition matrix to describe the response to transient faults is described in [9]. In [24], transient faults which result in steady-state failures are analyzed and detection methods are presented.

In [3] a practical methodology for simulated fault insertions under real workloads is described. More recently [1], physical fault injection has been used to validate a computerized interlocking system for the French railways. A new approach referred to as "accelerated fault injection" has recently been proposed and illustrated on IBM mainframes [4]. The results show that the method can be used to evaluate the coverage of various hardware and software fault tolerance schemes. An automated real-time distributed fault injection environment (FIAT) is presented in [20]. The concepts and design, as well as the implementation and evaluation of the environment are discussed. In [10], a novel method of inducing transients via heavy-ion radiation from Ca^{252} source is described. The method is applied to a MC6809E microprocessor. Recordings of the error behavior are used to characterize the errors, as well as to determine coverage and latency, for several error detection schemes. The experience gathered from the above studies shows that the data generated can provide considerable insight into both error manifestation and fault impact.

An important question not addressed in the above studies is the propagation of transients from the device-level, through the microprocessor functional units, to the pins. Apart from strengthening the knowledge of transient fault propagation in microprocessors, this information is crucial for further defining the vulnerability of digital systems to transients. In [5, 8] experiments to quantify the impact of transients from the device to pin-level, in a gate-array microprocessor chip were described. Transients with charge-levels in the range of 0.5 to 9 picoCoulombs were injected. Logic upsets and first-order latch and pin errors were measured and analyzed via analysis of variance (ANOVA) methods. The mechanisms involved in internal propagation of latch-errors (ie, transient fault latency) and their effect at the pin-level were investigated and modeled.

3. THE SIMULATION ENVIRONMENT

In order to perform a fast and accurate analysis, a mixed-mode transient-fault simulator [8] based on SPLICE [18] was used. A graphical analysis facility, FOCUS, was developed (on a color SUN Workstation) to visualize the error activity in different functional units of the processor, the fault propagation on the major interconnects, and at the external pins. The key features of the FOCUS environment illustrated in [5] are:

1. A visual display of the impact of an injected transient.
2. The generation of selected statistical distributions to quanitfy the internal and external fault propagation due to transients.
3. The generation of a multi-step fault propagation model to quantify the impact of transient fault latency.

In addition, the regions of increasing latch-error occurrences are identified by their color. The interconnects through which the faults propagate are also highlighted. In the usual case, the pre-processed error data from the fault simulations form the input to the graphical program. This allows accelerated viewing of the impact of the injected transient. After each injection/simulation run, the statistical distributions of the latch and pin error characteristics, and the fault propagation are calculated and displayed.

4. THE SIMULATED CONTROLLER

The example system in our study is a microprocessor-based, dual-channel controller for real-time control of jet-engine functions. The system processes data obtained from dedicated engine sensors to provide several functions such as automatic thrust control, engine-limit protection, engine-transient control, engine-fuel and oil temperature management and thrust reverser control. The digital system architecture (figure 1) contains microprocessors, buses, memory units, I/O processors, asynchronous serial communication links, frequency samplers and A/D converters.

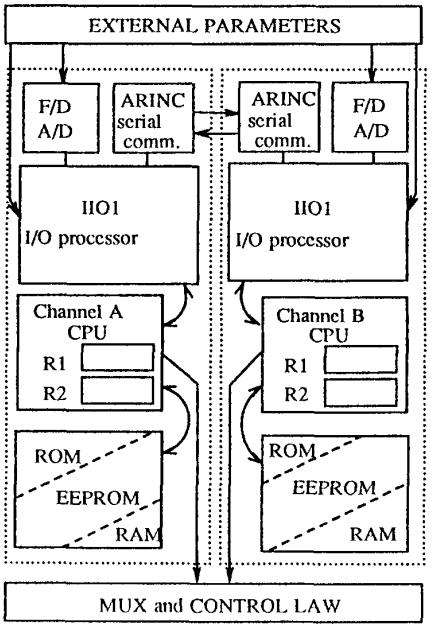

Figure 1. Simulated Engine Controller

The controller has two independent channels referred to as channel A and channel B, each consisting of a microprocessor chip. The I/O configuration of the hardware is identical for both channels with the exception of the following three control loops: the turbine cooling air loop and a thermatic rotor control loop assigned only to channel A and, another thermatic rotor control loop assigned only to channel B. All other functional loops have redundant implementations and can be controlled by either channel. In each channel, input information (eg, temperature,

test and drift signals, resolver inputs, transducer inputs, torque motor wrap-arounds, rotor speeds, pressures and test signals) is first digitized by two on-board frequency samplers and an A/D converter. This digitized input data (a single word) is then stored in the channel CPU register R1. The digitized data in channel A is sent to the channel B via a serial communication link and stored in CPU register R2 of channel B. Similarly the data in R1 in channel B is sent to register R2 in channel A. In each channel, a logic comparison of contents of registers R1 and R2 is made. If the comparison fails, a *range test*, which compares the data in R1 and R2 with the range information recorded in the ROM for the particular input variable, is performed. The content of the register *within range* is then used for continued processing. If both the data in R1 and R2 are out of range (multiple failure), the control signal (output) is synthesized from the other parameters and from previous engine states recorded in the EEROM. In cases of sustained multiple failures, a background test routine identifies the failed channel and transfers the engine control to the working channel.

The ability to detect and reconfigure through comparisons and range tests of critical input data is the main fault-tolerant aspect of the controller. In our experiment, a single channel functional error is assumed to occur if the injected transient alters the contents of either CPU registers R1 and R2 in a single channel. A dual channel functional error is defined as an event where the contents of CPU registers R1 and R2 are faulty in both channels. This event would require the invocation of the next level of protection.

5. THE EXPERIMENT

The focus of our simulation experiment was to stress the fault tolerance mechanisms, of the digital aspects, of the dual system. The following aspects of the controller were simulated: Both CPUs, at the gate and electrical levels to allow transient fault injections; the external modules, including I/O processors and memories (which were not subject to fault injection) at the functional level. Software to exercise the fault-tolerant operation of the dual system was programmed into the ROM and was executed by both processors. The executed instructions were intended to mimic the process of sampling of the engine oil temperature through the I/O processor, reading the sampled value from I/O processor, sending the value to the other channel through the crosstalk communication link, and receiving data from the other channel for comparison. The focus of the experiment was more on stressing the channel communications and less on attempting to use real data.

Transients with a charge-level between 0.5 and 8.0 picoCoulombs were first injected into the microprocessor of channel A (for single fault injections) and then into both A and B (for multiple faults injections). The charge-levels chosen represent the transient response of various heavy ions, including 100 MeV ^{56}Fe ions, which are commonly found in the cosmic environment. The levels were chosen so as to ensure that no permanent errors occur. Charge-levels approximately greater than 10 picoCoulombs are known to cause permanent latch-ups (device failure) [17]. Only the results for 8 picoCoulombs are presented here since, for charge-levels between 7 and 10 picoCoulombs the probability of error occurrences is relatively constant, [5] ie, additional charge does not result in an increase in the error probability.

The locations of the fault injections were selected in order to maximize the chance of channel failures in the system. This method is similar to the *failure acceleration* technique [4], wherein it was shown that accelerated error-to-failure scenarios can be used to estimate fault impact. For example, a transient was injected directly to register R1 at a point in time when it contained critical input data. Thus, a worst-case situation for the controller was modeled in this experiment. Locations and time-points of the injections were selected so as to alter the data value stored in the accumulators and to stress the crosstalk communication between the channels. These I/O functions were the targets for the induced failures since their short latencies allowed us to measure the error coverage without the overlapping effect of possible latent errors.

Transients were injected at eight selected nodes in the ALU and in the control units of the CPUs. Nodes were chosen to have low fan-out since the small capacitive loadings made them sensitive to logic upsets. Additionally, each node had fan-outs to critical points in the CPU, eg, the CPU registers, the CPU register control points and the external data-bus control lines. Time-points of the injections were chosen to be the clock cycles before new values were latched into the registers. In all, over 80 fault injections/simulations were performed. The choice of this number was determined by the fact that additional injection did not vary the result significantly.[1]

Recall that a single channel functional error is assumed to occur if the injected transient altered the critical input data in either R1 or R2. A dual channel functional error is defined as the event where the contents of CPU registers R1 and R2 are faulty in both channel A and channel B. The error-data for the analysis were generated by comparing each faulted simulation with a fault-free simulation. The error-data were then processed by a series of programs that collected statistics on the fault injections and the results.

5.1 *Impact of Single and Multiple Fault Injections*

The single fault injection experiments resulted in three types of errors:

1. *Error in Channel A Only*: In this case, critical data in register R1 in channel A became faulty, after the contents of R1 were sent to register R2 of channel B, ie, a correct copy of the critical input data was sent to channel B. This was not a serious problem since out of the four registers R1 and R2 in channels A and B, only one (R1 of channel A) was corrupted. The system could sustain a second fault in one of the registers and still continue to be operational.

[1]This also follows common statistical principles. By the law of large numbers, a sample of greater than 30 or so is expected to produce stable statistical distributions.

2. *Crosstalk Error*: In this case, an error was introduced during the communication between the two channels in the dual system. During the process of sending the critical data from R1 in channel A, to R2 in channel B, the I/O processor of channel A read faulty data into the I/O buffer from the bus. As a result, the I/O processor sent a faulty value to channel B. This was also not a serious problem since, as before, three of the registers contained error-free data (only R2 in channel B is faulty). Again the system could sustain a second fault with no apparent impact.

3. *Error in Both Channels*: In this case, a transient altered the content of R1 in channel A before it was sent to channel B. Thus faulty data from R1 in channel A was sent to R2 in channel B, ie, both the data in R1, channel A and R2, channel B were faulty. The system was still operational because the data in R2 in channel A and R1 in channel B were error free. However, unlike the above cases, a second fault could cause both channels to have functional errors. This is the most critical of the single fault conditions.

Table 1 summarizes the results of the experiment. In the table, the number of transient injections resulting in an error in each case is given. Note that approximately one in three transients causes an error in the critical data. Over 8 percent of the injections result in altering critical data in R1 in only one channel. There is approximately a 12 percent chance that correct data from R1 in channel A is altered during the crosstalk communication to channel B. The probability of a transient causing an error in both channels is moderately high (16%). In each of the above cases, the controller continues to operate without failure because one of the channels still provides the correct input data.

TABLE 1
Error Due to Fault Injections in The Channel A

Error Category	Error Frequencies	Fraction (%)
No Error	51	63.8
Error in Channel A Only	7	8.8
Crosstalk Error	9	12.3
Error in Channels A and B	13	16.3
Total	29	36.3

5.2 *Multiple Injections*

The dual configuration of the system is quite effective in tolerating single event faults. However, in an actual operating environment, transients do not always occur in isolation. The chances of having multiple errors as a result of external current or voltage spikes or, as a result of transients occurring on the external input lines may be significant. This is particularly true if the input lines are connected to multiple locations in the system. For example, a lightening strike on an aircraft can affect several sensor-input lines of the avionic equipment. The resulting errors may impact more than one CPU or component simultaneously and can alter critical input data in several registers. The impact of transients occurring at multiple locations, at the same time, is studied in this section. Our fault injection methodology places particular emphasis on multiple errors that can result in altering critical input data in both channels. We do not however claim to accurately model the physical transients occurring in real avionic environments.

Table 2 shows the impact of the multiple transients in the target system. In the table, over 52% of the multiple errors result in functional errors in one channel. However, only 12% of the injected transients result in causing functional errors in both channels, ie, only one in every four functional errors affects both channels. The overall coverage of multiple transients is approximately 88%. The confidence interval for this estimate is calculated in the following section. Multiple faults that alter the contents of the R1 registers in both channels are seen to be critical since they will result in the invocation of the reserve value. An increase in the fault tolerance of the input data paths to the R1 registers and their control circuits may significantly improve this aspect of system dependability.

TABLE 2
The Multiple Fault Injection Results

Fault Category	Occurrences	Fraction (%)
No Error	19	47.5
Functional Error in One Channel	21	52.5
Functional Errors in Both Channels	5	12.5

5.3 *Confidence Limits*

Assume that each multiple fault injection can cause both channels to have functional errors with a probability ρ. Assuming that the fault injections are statistically independent, the probability of a functional error on each trial is ρ. The random variable X (number of failures) has binomial probability distribution with $n=40$, $p=\rho$ and $q=1-\rho$. $\bar{X}$ in our experiment is 5. The estimated value, $\bar{\rho}$ from the experiment is 0.125 (=[number of functional errors]/[number of fault injections]). Since the number of trials is sufficiently large ($n=40$), the pdf$\{X\}$ can be approximated by a normal distribution with ($\mu=np=5.000$, $\sigma^2=np(1-p)=4.375$).

We find the 90% confidence limit [α, β] for X as follows:

$$\text{Lower 45\% limit: } \alpha=\bar{X}-Z_{0.05}(\sigma/\sqrt{n})$$

$$\text{Upper 45\% limit: } \beta=\bar{X}+Z_{0.05}(\sigma/\sqrt{n})$$

Based on the above assumptions, the 90% confidence limits for number of dual channel failures in the experiment is [4.456, 5.544], ie, the coverage of multiple transients is (86.1%, 88.9%] with 90% confidence.

6. CONCLUDING REMARKS

This paper discussed an experimental methodology for simulation-based evaluation of fault tolerant microprocessor architectures. The approach used mixed electrical and logic simulations, combined with fault injections, to evaluate the susceptibility of fault tolerant designs to transient errors. The method was illustrated on the digital aspects of a fault tolerant, dual channel jet-engine controller. The coverage of the fault tolerance technique to single and multiple transients was evaluated. The locations and the time-points of the fault injections were selected so as to maximize the chance of channel errors. Specifically, faults were injected under conditions where critical communications were taking place within the dual system. The results showed that the controller had an estimated 100% coverage against single isolated transients while, approximately 12 percent of the multiple transients affected both channels.

ACKNOWLEDGMENT

This work was supported by the National Aeronautics and Space Administration under NASA grant NAG-1-602. We thank the researchers at NASA AIRLAB, in particular Celeste Belcastro, Felix Pitts and Chuck Meissner for many useful discussions. Thanks are also due to Kumar Goswami for his comments on this manuscript.

REFERENCES

[1] J. Arlat, Y. Crouzet, J. Laprie, "Fault-injection for dependability validation of fault-tolerant computing systems", Digest, FTCS-19, *The Ninth Int'l Symp. Fault Tolerant Computing*, 1989 Jun, pp 348–355.

[2] H. Ball, F. Hardy, "Effects and detection of intermittent failures in digital systems", 1969 FJCC, *AFIPS Conf. Proc.*, vol 35, 1969, pp 329–335.

[3] R. Chillarege, R. K. Iyer, "Measurement-based analysis of error latency", *IEEE Trans. Computers*, vol C-36, 1987 May, pp 529–537.

[4] R. Chillarege, N. S. Bowen, "Understanding large system failures-A fault injection experiment", *Digest, FTCS-19, Ninth Int'l Symp. Fault Tolerant Computing*, 1989 Jun, pp 356–364.

[5] G. Choi, R. K. Iyer, Resve Saleh, Victor Carreno, "A fault behavior model for an avionic microprocessor: A case-study", *Proc., Working Conf. Dependable Computing for Critical Applications*, 1989 Aug.

[6] B. Courtois, "Some results about the efficiency of simple mechanisms for the detection of microcomputer malfunctions", *Digest, FTCS-9, Ninth Int'l Symp. Fault Tolerant Computing*, 1979 Jun, pp 71–74.

[7] J. Cusick, R. Koga, W. A. Kolasinski, C. King, "SEU vulnerability of the Zilog Z-80 and NSC-800 microprocessors", *IEEE Trans. Nuclear Science*, vol NS-32, 1985 Dec, pp 4206–4211.

[8] P. Duba, R. K. Iyer, "Transient fault behavior in a microprocessor: A case study", *Proc. 1988 IEEE Int'l Conf. Computer Design: VLSI in Computers & Processors* (ICCD-88), 1988 Oct, pp 272–276.

[9] R. E. Glaser, G. M. Masson, "Transient upsets in microprocessor controllers", Digest, FTCS-11, *Eleventh Int'l Symp. Fault Tolerant Computing*, 1981 Jun, pp 165–167.

[10] U. Gunneflo, J. Karlsson, J. Torin, "Evaluation of error detection schemes using fault injection by heavy-ion radiation", *Digest, FTCS-19, Nineteenth Int'l Symp. Fault Tolerant Computing*, 1989 Jun, pp 340–347.

[11] R. K. Iyer, D. J. Rossetti, "A measurement-based model for workload dependence of CPU errors", *IEEE Trans. Computers*, vol C-35, 1986 Jun, pp 511–519.

[12] S. Kim, R. K. Iyer, "Impact of device level faults in a digital avionic processor", AIAA/IEEE *8th Digital Avionics Systems Conf.*, 1988 Oct, pp 428–436.

[13] R. Koga, W. A. Kolasinski, M. T. Marra, "Techniques of microprocessor testing and SEU-rate prediction", *IEEE Trans. Nuclear Science*, vol NS-32, 1985 Dec, pp 4219–4224.

[14] D. Lomelino, R. K. Iyer, "Error propagation in a digital avionic processor: A simulation-based study", *Proc. Real Time Systems Symp.*, 1986 Dec, pp 218–225.

[15] T. C. May, M. H. Woods, "Alpha-particle-induced soft errors in dynamic memories", *IEEE Trans. Electron Devices*, vol ED-26, 1979 Jan, pp 2–9.

[16] G. C. Messenger, "Collection of charge on junction nodes from ion tracks", *IEEE Trans. Nuclear Science*, vol NS-29, 1982 Dec, pp 2024–2031.

[17] D. Nichols, W. Price, W. Kolasinski, R. Koga, J. Pickel, J. Blandford Jr., A. Waskiewicz, "Trends in parts susceptibility to single event upset", *IEEE Trans. Nuclear Science*, vol NS-32, 1985 Dec.

[18] R. A. Saleh, "Nonlinear relaxation algorithms for circuit simulation", Memorandum No. UCB/ERL M87/21, 1987; Electronics Research Laboratory University of California, Berkeley.

[19] M. E. Schmid, R. L. Trapp, A. E. Davidoff, G. M. Masson, "Upset exposure by means of abstraction verification", *Digest, FTCS-12, Eleventh Int'l Symp. Fault Tolerant Coomputing*, 1982 Jun, pp 237–244.

[20] Z. Segall, D. Vrsalovic, D. Siewiorek, D. Yaskin, J. Kownacki, J. Barton, "FIAT-Fault injection based automated testing environment", *Digest, FTCS-18, Eighteenth Int'l Symp. Fault Tolerant Computing*, 1988 Jun, pp 102–107.

[21] R. Johnson, S. Diehl-Nagle, J. Hauser, "Simulation approach for modeling single event upsets on advanced CMOS SRAMS", *IEEE Trans. Nuclear Science*, vol NS-32, 1985 Dec, pp 4122–4127.

[22] D. Siewiorek, R. Swarz, *The Theory and Practice of Reliable System Design*, 1982; Digital Equipment Corporation.

[23] K. G. Shin, Y. H. Lee, "Measurement of fault latency: Methodology and experimental results", Technical Report CRL-TR-45-84, 1984; Computing Research Laboratory University of Michigan, Ann Arbor.

[24] J. Sosnowski, "Evaluation of Transient Hazards in Microprocessor Controllers", *Digest, FTCS-16, Sixteenth Int'l Symp. Fault Tolerant Computing*, 1986, pp 364–369.

AUTHORS

Gwan S. Choi; Center for Reliable and High Performance Computing; Coordinated Science Laboratory; University of Illinois at Urbana-Champaign; 1101 West Springfield Avenue; Urbana, Illinois 61801 USA.

Gwan S. Choi received the BS in Computer Engineering from the University of Illinois, Urbana, Illinois in 1989, and the MS in Electrical Engineering in 1990. Currently a PhD degree candidate in Electrical Engineering at the University of Illinois, his research interests include fault-tolerant computing, reliability modeling and analysis, and simulation. Since 1988, he has been a Research Assistant in the Coordinated Science Laboratory at the University of Illinois. In addition, he has worked for CRAY Research Inc., Mendota Hights during the summer of 1987. Mr. Choi is a member of Eta Kappa Nu.

Ravi K. Iyer; Center for Reliable and High Performance Computing; Coordinated Science Laboratory; University of Illinois at Urbana-Champaign; 1101 West Springfield Avenue; Urbana, Illinois 61801 USA.

Ravi K. Iyer received his BE and PhD from the University of Queensland, Brisbane, Australia, in 1973 and 1977. From 1979 to 1983 he was at Stanford University in the Center for Reliable Computing, Departments of Electrical Engineering and Computer Science. Since the fall of 1983 he has been at the University of Illinois at Urbana-Champaign, where he holds a joint appointment as Professor of Electrical and Computer Engineering, Computer Science and the Coordinated Science Laboratory. He is also Director of the Center for Reliable and High Performance Computing and Co-Director of the Illinois

Computer Laboratory for Aerospace Systems and Software, a NASA Center of Excellence in Aerospace Computing. Prof. Iyer's research interests are in reliable and fault-tolerant computing, measurement, experimentation and statistical modeling. He has served on several program committees and is on the editorial boards of the *Journal of Electronic Testing* and the Springer Verlag *Series on Dependable Computing*. He was General Chair of 19th International Symposium on Fault-Tolerant Computing (FTCS-19), and was the Guest Editor of the *IEEE Trans. Software Engineering*, "Special Issue on Experimental Computer Science." He has been a consultant to industry and other research laboratories in reliable computer design, in particular to IBM, NCR, Bell Canada and the Jet Propulsion Laboratory. He is a senior member of IEEE, a member of ACM, Sigma Xi and the IFIP technical committee (WG 10.4) on fault tolerant computing. Prof. Iyer is an IEEE Computer Society *Distinguished Visitor*.

Victor A. Carreno; NASA; Langley Research Center; Hampton, Virginia 23665-5225 USA.

Victor A. Carreno received a BS in Electrical Engineering from the University of Puerto Rico in 1979 and a MSEE from Old Dominion University, Norfolk, Virginia in 1986. He has been a research engineer at NASA Langley Research Center since 1979 working in the Aircraft Electronic Systems, Fault Tolerant Systems, and System Validation Methods Branch. Mr. Carreno research interest is in ultra reliable systems including fault tolerance, design for validation, and mathematical formal methods for design error exclusion.

Manuscript TR90-308 received 1990 February 5; revised 1990 May 23; revised 1990 July 7.

IEEE Log Number 37706 ◄TR►

MANUSCRIPTS RECEIVED *MANUSCRIPTS RECEIVED*

"Improvement, deterioration, and optimal replacement under age-replacement with minimal repair", Kanchan Jain □ Deptt. of Statistics □ Panjab University □ Chandigarh—160 014 □ INDIA. (TR90-129)

"A note on the failure rate of a {n,1}-out-of-n:F (gracefully degrading) system", Dr. Ravi Mukkamala □ Dept. of Computer Science □ Old Dominion University □ Norfolk, Virginia 23529 □ USA. (TR90-133)

"Comment on: Reliability prediction, Fact or fancy? (an editorial)", Dr. T. G. Pham □ Faculty of Science & Engineering □ Universite de Moncton □ Moncton, New Brunswick E1A 3E9 □ CANADA. (TR90-134)

"Asymptotic analysis of a reliability estimator", S. Ejaz Ahmed □ Dept. of Mathematics & Statistics □ University of Regina □ Regina, Saskatchewan S4S 0A2 □ CANADA. (TR90-135)

"Exposure time in maintainable systems", Arne G. Sandberg □ 418—Rd. 6 FU □ Cody, Wyoming 82414 □ USA. (TR90-136)

"Some Weibull models for software failure rates", Dr. Manju Pandey, Reader □ Dept. of Zoology □ Faculty of Science □ Banaras Hindu University □ Varanasi—221 005 □ INDIA. (TR90-137)

"A 2-stage life-test based on total time on test", Y. I. Kwon □ Dept. of Industrial Engineering □ Chongju University □ Chongju □ Chungbuk 360-764 □ Republic of KOREA. (TR90-138)

"Reliability of a k-out-of-n:G system with common-cause failures", Dr. Kyung-Hee Jung □ Policy & Economic Analysis Dept. □ Korea Electrotechnology Research Institute □ POBox 20 □ Changwon □ Republic of KOREA. (TR90-139)

"Probabilistic safety analysis for systems with standby subsystems with sequentially used standbys", Dr. Zhang, Qin □ Institute of Nuclear Energy Technology □ Tsinghua University □ Beijing—100 084 □ Peop. Rep. CHINA. (TR90-141)

MANUSCRIPT RECEIVED *MANUSCRIPTS RECEIVED*

"Voting networks", Dr. Behrooz Parhami □ Dept. of Electrical & Computer Engineering □ University of California □ Santa Barbara, California 93106 □ USA. (TR90-142)

"Approx 1-sided tolerance limits for future observations for the Rayleigh distribution using regression", Dr. Mostafa S. Aminzadeh □ Dept. of Mathematics □ Towson State University □ Towson, Maryland 21204-7097 □ USA. (TR90-143)

"Perfect-debugging model for redundant software", Dr. Hsin-Hui Lin □ Institute of Information Management □ National Sun Yat-Sen University □ Hsi-Tze Wan □ Kaohsiung □ TAIWAN—R.O. CHINA. (TR90-144)

"Mil-Hdbk-217: What is wrong with it?", Puran Luthra □ MS 4210 □ Emerson Electric, Electronics & Space Div. □ 8100 West Florissant Avenue □ St. Louis, Missouri 63136 □ USA. (TR90-145)

"Software-reliability models that depend on the testing-domain", Dr. Hiroshi Ohtera □ Information Processing Center □ Okayama University of Science □ Ridai-cho 1-1 □ Okayama-shi 700 □ JAPAN. (TR90-148)

"Reliability computations of k-out-of-n:G structures", Dra. Margarita Martinez-Nebreda □ Dpto. de Matematica Aplicada □ E.T.S.I.I.T. □ c/ Alameda de Urguijo s/n □ 48013 Bilbao □ SPAIN. (TR90-149)

"Usefulness of MTTF of s-independent case in other cases", Dr. W. G. Schneeweiss, Professor □ Fernuniversitaet □ Postfach 940 □ D-5800 Hagen 1 □ Fed. Rep. GERMANY. (TR90-150)

"Comment on: Statistics & ignorance (an editorial)", John D. Healy □ Rm 2X-227 □ Bell Communications Research □ 331 Newman Springs Road □ Red Bank, New Jersey 07701-7020 □ USA. (TR90-151)

"A minimizing algorithm for sum of disjoint products with grouped-variable inversion", Dr. Mitchell O. Locks □ 137 South Palm Drive, -302 □ Beverly Hills, California 90212 □ USA. (TR90-152)