May 1990

UILU-ENG-90-2216
CSG-124

LANGLEY GRANT
IN-07-CR
281600
P.43

COORDINATED SCIENCE LABORATORY
College of Engineering

AN EXPERIMENTAL STUDY OF FAULT PROPAGATION IN A JET-ENGINE CONTROLLER

Gwan Seung Choi

(NASA-CR-181335) AN EXPERIMENTAL STUDY OF
FAULT PROPAGATION IN A JET-ENGINE CONTROLLER
M.S. Thesis (Illinois Univ.) 43 p CSCL 21E

N90-23401

Unclas
G3/07 0281600

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Approved for Public Release. Distribution Unlimited.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
None

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
UILU-ENG-90-2216 CSG 124

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Coordinated Science Lab
University of Illinois

6b. OFFICE SYMBOL *(If applicable)*
N/A

7a. NAME OF MONITORING ORGANIZATION
NASA

6c. ADDRESS *(City, State, and ZIP Code)*
1101 W. Springfield Ave.
Urbana, IL 61801

7b. ADDRESS *(City, State, and ZIP Code)*
NASA Langley Research Center
Hampton, VA 23665

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
NASA

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
NASA NAG 1-602

8c. ADDRESS *(City, State, and ZIP Code)*
NASA Langley Research Center
Hampton, VA 23665

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
An Experimental Study of Fault Propagation in a Jet-Engine Controller, MS Thesis (EE)

12. PERSONAL AUTHOR(S)
Gwan Seung Choi

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
1990 May

15. PAGE COUNT
36

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
simulation, transient faults, fault injection, error propagation, empirical models

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This thesis describes an experimental analysis of the impact of transient faults on a microprocessor-based jet-engine controller, used in the Boeing 747 and 757 aircrafts. A hierarchical simulation environment which allows the injection of transients during run-time and the tracing of their impact is described. Verification of the accuracy of this approach is also provided. A determination of the probability that a transient results in latch, pin or functional errors is made. Given a transient fault, there is approximately an 80% chance that there is no impact on the chip. An empirical model to depict the process of error exploration and degeneration in the target system is derived. The model shows that, if no latch errors occur within eight clock cycles, no significant damage is likely to happen. Thus, the overall impact of a transient is well contained. A state transition model is also derived from the measured data, to describe the error propagation characteristics within the chip, and to quantify the impact of transients on the external environment. The model is used to identify and isolate the critical fault propagation paths, the module most sensitive to fault propagation and the module with the highest potential of causing external pin errors.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE *(Include Area Code)*

22c. OFFICE SYMBOL

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

AN EXPERIMENTAL STUDY OF FAULT PROPAGATION IN A JET-ENGINE CONTROLLER

BY

GWAN SEUNG CHOI

B.S., University of Illinois, 1989

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1990

Urbana, Illinois

ABSTRACT

This thesis describes an experimental analysis of the impact of transient faults on a microprocessor-based jet-engine controller, used in the Boeing 747 and 757 aircrafts. A hierarchical simulation environment which allows the injection of transients during run-time and the tracing of their impact is described. Verification of the accuracy of this approach is also provided. A determination of the probability that a transient results in latch, pin or functional errors is made. Given a transient fault, there is approximately an 80% chance that there is no impact on the chip. An empirical model to depict the process of error explosion and degeneration in the target system is derived. The model shows that, if no latch-errors occur within 8 clock cycles, no significant damage is likely to happen. Thus, the overall impact of a transient is well contained. A state transition model is also derived from the measured data, to describe the error propagation characteristics within the chip, and to quantify the impact of transients on the external environment. The model is used to identify and isolate the critical fault propagation paths, the module most sensitive to fault propagation and the module with the highest potential of causing external pin-errors.

PAGE ii INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

ACKNOWLEDGMENTS

First, I thank my advisor, Professor Ravi Iyer, for his invaluable help and guidance in regard to this thesis. I also thank the researchers at NASA AIRLAB for many useful discussions and their continous support during my work. In particular I thank Victor Carreno for his help and providing insight into the EEC131 controller. I thank Professor Resve Saleh for providing insight into SPLICE, and Pat Duba for many useful discussions regarding the fault simulator. Thanks are also due to Kumar Goswami, Rene Llames, Robert Dimpsey and Jaidip Singh for their comments on an early draft of this thesis. Special thanks and gratitude go to my parents and my brother, SeungJin, for their support and encouragement.

TABLE OF CONTENTS

CHAPTER Page

1. INTRODUCTION 1
    1.1. Related Research 2

2. TARGET SYSTEM 6

3. THE EXPERIMENTAL ENVIRONMENT 9
    3.1. Simulation Environment 9
    3.2. Graphics Analysis Facility 10
    3.3. The Experiment 12
    3.3. Validation of The Mixed-Mode Transient Simulator 14

4. IMPACT OF TRANSIENTS 17
    4.1. Charge-Level Analysis 19

5. EMPIRICAL MODELS 23
    5.1. Error Latency 23
    5.2. Error Propagation Model 24
    5.3. State Transition Model 28

6. CONCLUSIONS 32

REFERENCES 35

CHAPTER 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 valid concerns regarding the dependability of these systems. A particular source of concern is the impact of transients which are rather common in avionic environments. This is especially so since past measurements [1],[2] show that over 80 percent of all computer system failures can be attributed to transients.

This thesis describes an experimental analysis of the impact of transient faults in a microprocessor-based jet-engine controller used in the Boeing 747 and 757 aircrafts. A hierarchical simulation environment for the run-time injection of transients and for the tracing of their impact is described. A determination of the probability that a transient results in latch, pin or functional errors is made. Given a transient fault, there is approximately an 80 percent chance that there is no impact on the chip. The probability of a latch error is over 20 percent, while that of a pin error is approximately 12 percent.

An empirical model to depict the process of error explosion and degeneration in the target system is derived. The model shows that, if no latch errors occur within 8 clock cycles, no significant damage is likely to happen. Thus, the overall impact of a transient is well contained. A state transition model is derived from the measured data to describe the error propagation characteristics within the chip and to quantify the impact of transients on the chip's external environment. The model is used to identify and isolate the critical fault propagation paths, the module most sensitive to fault propagation and the module with the highest potential of causing external pin-errors.

1.1. Related Research

Several researchers have investigated the impact of transients in computer systems. An early study of the effects and detection of failures in digital systems reported in [1] showed that nearly 90 percent of failures were transient in nature. Recent studies using failure data from IBM mainframes reported in [2] also showed that nearly 85 percent of major system errors were transient in nature. Furthermore, a strong relationship was found between the occurrence of transients and the level of system activity.

Device-level analysis of the mechanisms of transient upset has been in progress for quite some time. The hazards of transient upset in dynamic RAMs was first reported in [3] where the behavior of alpha-particle induced soft errors was explored. Simulation techniques for modeling the device-level effects of cosmic particle induced transients were developed in [4] and [5]. Reference [4] used a SPICE circuit with a current source to represent collected charges generated by alpha particles. In [5], a simulation technique for modeling the ion shunt effect was developed. An approximate analytic solution which models a current transient was developed in [6]. The model includes parameters which represent the maximum current, the collection time constant of the junction, and the time constant for initially establishing the ion track. The analytic solution was validated by comparison with other computer models and is used in this study.

A series of experiments, aimed at error analysis through the physical insertion of faults, was conducted by several investigators at the NASA AIRLAB test bed facility. An experiment to study fault latency distributions through hardware fault injections is described in [7] and [8]. Information gathered from these studies shows that the data generated can provide considerable insight into error manifestation. Another useful study is [9] which describes a simulation experiment to determine the efficiency of a number of error-detection mechanisms. In [10], an approach to expose upsets in a computer system by abstraction verification is described. More recently in [11], physical fault injection was used to validate a computerized interlocking system for the French railways.

At the microprocessor level, studies have primarily focused on vulnerability assessment and software detection methods. An assessment of different transient error test methods is presented in [12]. In [13], a detailed analysis of the vulnerability of the Z80 microprocessor based on ion bombardment testing is described.

The data obtained are based on a technique which associates upsets with the machine cycle during which the errors first appear on the pins. Studies have also focused on the efficiency of the methods for software detection of transient faults. An approach which involves the development of a state transition matrix to describe the response to transient faults is described in [14]. In [15], transient faults which result in steady-state failures are analyzed and detection methods are presented.

An investigation of fault propagation in microprocessors was conducted in [16]. An experimental analysis to study error propagation from the gate to the pin-level, for stuck-at faults, was described. The target system was a Bendix BDX-930 digital avionic miniprocessor. The analysis quantified the dependency of the measured error propagation on the location of the fault and the type of instruction/micro-instruction activity. The investigation presented a methodology for quantifying error propagation from the gate to pins for stuck-at faults. In [17], new experiments to study fault and error latencies under varying workload conditions are discussed.

An important question not addressed in the above studies is the propagation of transients from the device-level through the microprocessor functional units and pins. Apart from furthering the knowledge of transient fault propagation in microprocessors, this information is crucial for further defining the vulnerability of microprocessors to transients. In [18], a preliminary experiment to quantify the impact of transients from the device to the pin-level was described. Transients with charge-levels of 0.5, 1, 2, 3 and 4 picocoulombs were injected. Logic upsets and first-order latch and pin errors were measured and analyzed via analysis of variance methods.

The above results point toward the need for more complete analysis of fault propagation characteristics. The type of functional-errors which can result from the injected transients needs to be determined. Such errors can result in serious system malfunction, especially in avionic systems. In order to isolate the critical paths in the circuit, the fault propagation between the functional units and to the external pins must be quantified. In particular, the mechanisms involved in internal propagation of latch errors (i.e., transient fault latency) and their effect at the pin-level need to be investigated and modeled.

CHAPTER 2.

TARGET SYSTEM

The target system for this study is a microprocessor used for real-time control of jet-engine functions. The system is currently used in commercial aircraft, including the BOEING-747 and the 757. The controller (EEC131, manufactured by Hamilton Standard) has two channels; the processing elements of both channels are identical. The system has a real-time reconfiguration mechanism wherein the lead channel stops its usual operation on detecting a fault and transfers control to the dual. The system incorporates a variety of fault-tolerant design features at different levels including software checks, parity checks, memory test and error counting.

The control system samples engine parameters such as the fuel flow, the temperature, the engine speed and other external inputs such as air speed and positional parameters. The sampled parameters are digitized and updated into the RAM approximately every millisecond for further processing. The controller also reads pilot inputs (from the throttle and various switches) into a RAM work area and calculates the desired control functions. The calculated functions are used to drive display indicators and to control the engine. The equations describing the control functions are programmed in the application code which resides in EPROMs.

The control system architecture thus contains microprocessors, memory units, I/O gate array chips, communication channels, frequency samplers, A/D converters and D/A converters. In this experiment, the microprocessor and its associated memory were simulated with a focus on the impact of transient errors.

The 16-bit HS1602 microprocessor (Figure 1), which is the heart of the controller, consists of six major functional units. The arithmetic and logic unit (ALU), which contains six registers, can perform double precision arithmetic operations. The control unit, which is responsible for issuing signals to control the operations of the ALU, is made up of combinational logic and several regis-

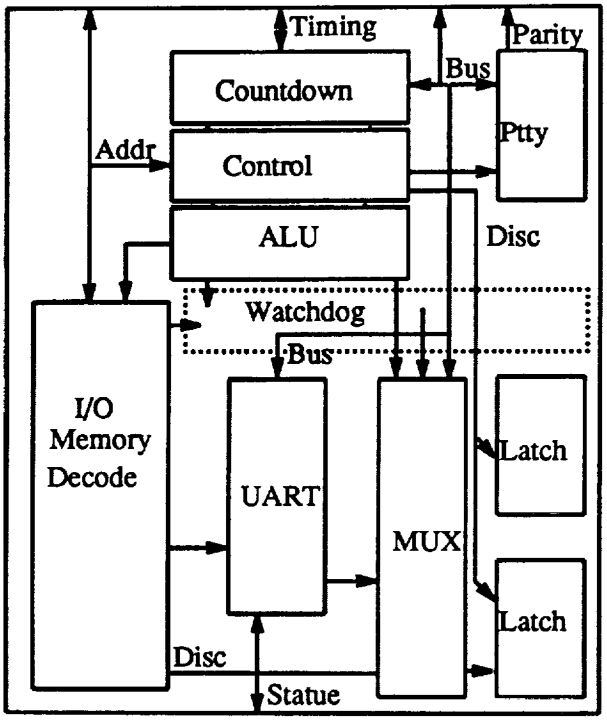

Timing
Parity
Bus
Countdown
Ptty
Addr
Control
ALU
Disc
Watchdog
Bus
I/O
Memory
Decode
UART
MUX
Latch
Latch
Disc
Statue

Figure 1: Data Flow Diagram of HS1602.

ters. The decoder unit decodes I/O signals, the multiplexer unit provides the discrete lines and buses, and the countdown unit is used to drive chip-wide clock signals. The watchdog unit provides protection against fault by resetting the processor in the event of parity error or when the application software is timed out by the software sanity timer. The signal to synchronize the dual system is also provided by this unit. The chip runs at 6 MHz and is implemented in a 3-micron technology CMOS gate-array made of 2688 blocks of 4 N-channel and 4 P-channel transistors.

CHAPTER 3.

THE EXPERIMENTAL ENVIRONMENT

3.1. Simulation Environment

In order to perform fast and accurate analysis, a mixed-mode transient fault simulator [18] based on SPLICE1 [19][1] was used. The simulator provides a fault injection and analysis environment which uses the SPLICE1 relaxation algorithm for circuit analysis. Transients can be injected without explicit modification of the circuit description. A transient injection is equivalent to a run-time modification of the circuit whereby a current source is added to the target-node,[2] thus altering the voltage level of the node over the time interval of the injected current waveform. The method allows both single and multiple transient injections. Since the injected current source is specified in a mathematical functional form, the transients can be of varying shapes and duration. Details of the implementation are given in [18].

For a comprehensive study of fault propagation in the microprocessor, a tracing facility was also developed to monitor all of the internal nodes (over

---

[1] The electrical analysis in SPLICE1 is based on the method of Iterated Timing Analysis (ITA), which been shown to be as accurate as SPICE2 in [20] and can provide a speed-up to two orders of magnitude. The logic analysis in SPLICE1 is performed using a relaxation-based method that uses MOS oriented models. Virtually unlimited levels of signal strength can be associated with each of the logic values in order to further enhance accuracy. This approach allows a correspondence between the electrical output conductance and the logic output strength. By using a fanout-dependent delay-model, which is capable of handling first-order effects, accurate delay-handling is achieved.

[2] A node is defined as a point in a conductive interconnection between electrical and/or logical elements.

4000) in the HS1602. The tracing facility is capable of monitoring each node for all processed events. The trace data for each event consist of the time of the event, the hierarchical node name and the new and previous voltage levels (for electrical nodes) or the new and previous logic levels and their strengths (for logic nodes).

3.2. Graphics Analysis Facility

A graphics analysis facility was developed (on a color SUN Workstation) to visualize the error activity in different functional units of the processor and the fault propagation on the major interconnects and at the external pins. The key features include:

1). A visual display of the impact of an injected transient.

2). The generation of selected multivariable statistical distributions to quantify the internal and external fault propagations due to transients.

3). The generation of a multistep 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. A sample of the graphical display is shown in Figure 2: (a) functional units, (b) the major interconnections, (c) the external pins, (d) error

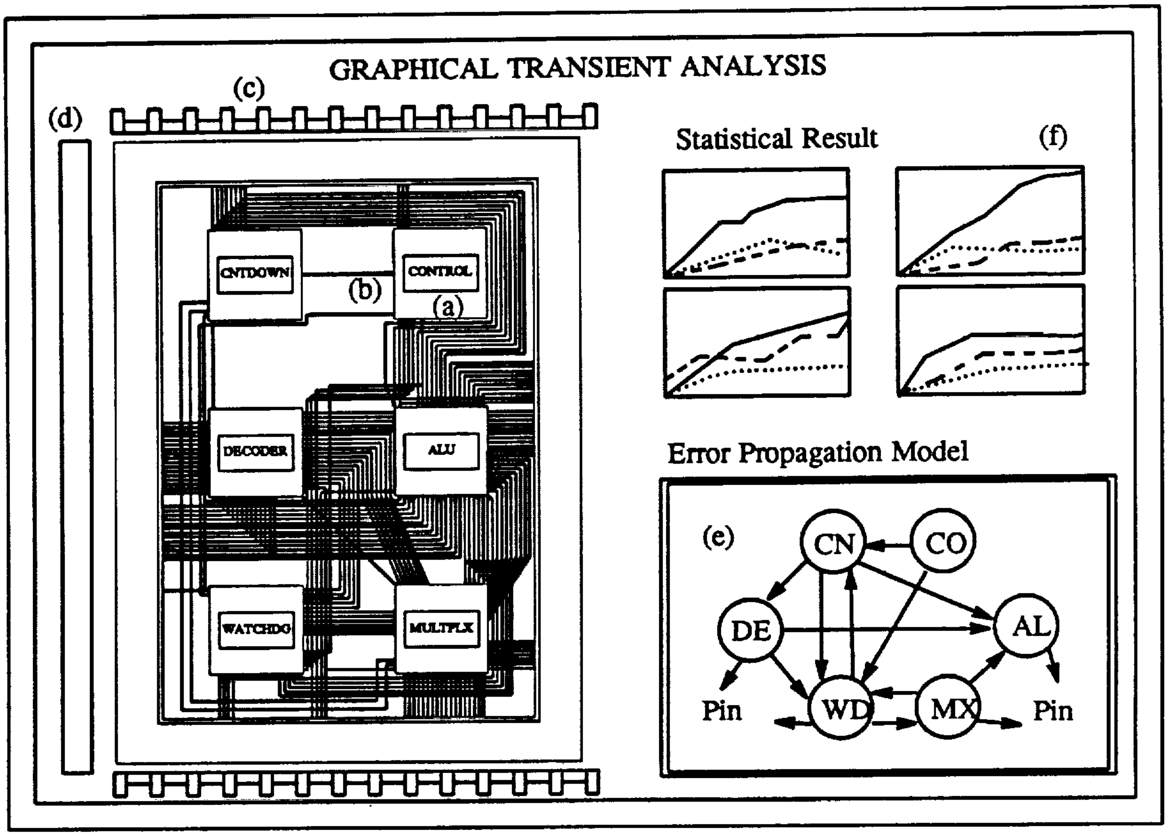

GRAPHICAL TRANSIENT ANALYSIS
(c)
(d)
CNTDOWN
CONTROL
(b)
(a)
DECODER
ALU
WATCHDG
MULTPLX
Statistical Result
(f)
Error Propagation Model
(e)
CN
CO
DE
AL
WD
MX
Pin
Pin

Figure 2: Display of Graphical Program.

sensitivity color code, (e) the derived fault propagation model and (f) statistical distributions. In the usual case, the preprocessed error data from the fault simulations form the input to the graphical program. This allows accelerated viewing of the impact of the injected transient. The propagation path of an injected transient is traced on screen by a red "blip" through various internal modules and external pins. The fault sensitive functional unit (originally represented in blue) gradually becomes yellow, then red. After each injection/simulation run, the statistical distributions of the latch and pin error characteristics and the fault propagation can be calculated and displayed. This process can also be done in real-time when graphical analysis is directly interfaced with the fault simulator.

3.3. The Experiment

In this experiment, the entire HS1602 was simulated along with its associated memory modules. In the simulations, the gates around the region of fault injection were simulated at the electrical level and the rest of the processor was simulated at the logical level. The memory modules, which were not subject to fault injection, were simulated at the functional level. The actual design parameters of HS1602 and the capacitances extracted from the circuit layout were used in the simulations. The initialization phase of the microprocessor, which consists of a watchdog test, a parity test, an instruction set test, a RAM test and a ROM sum test and ensures that all of the functional units are exercised, was simulated. The simulation includes the processor accessing one external ROM for instructions and another external ROM for the initialization parameters. Arithmetic processing and address generation are also performed.

In the experiment, transients with charge levels in the range 0.5 to 9.0 picocoulombs[3] were injected at seven randomly chosen nodes in each one of the six major functional units. Each charge level was injected at five different time points during the execution of the application code sequence. The specific waveforms used in the fault simulations follow the double-exponential function proposed in [6]

---

[3] The charge-levels chosen represent transient response of various heavy ions including 100 MeV ^{56}Fe ions, which are commonly found in the cosmic environment [21]. These 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) in reference [22].

$$I(t) = \zeta\,[e^{-t/\alpha} - e^{-t/\beta}]$$

where ζ is the approximate maximum current, α is the collection time-constant for junction and β is the ion track establishment time-constant.

The error data for the analysis were generated by comparing each faulted simulation with a fault-free simulation. An error was assumed to occur if the injected transient caused the node voltage to vary beyond a defined logic threshold. For each simulation, the recorded data included the time of fault occurrence, the location of fault, the faulted value, and the fault-free value. Each fault event was also classified as either a timing error (premature or late firing) or a value error.

The error data were then processed by a series of programs that collected statistics on the fault injections which resulted in a voltage transient large enough to result in latch and pin errors and errors at the interconnections of the functional units.

Statistics on errors resulting in a functional alteration of the processor functions were also collected. The collected statistics were classified by the charge level and by the location. In total, over 2100 fault injections/simulations have been performed.

3.4. Validation of The Mixed-Mode Transient Simulator

Recall that in this experiment a region around the injected node was simulated at the electrical level and the rest of the circuit at the logic level. In using such mixed-mode simulations, the question of the accuracy of the signal transfer between the electrical level and the logic level analysis needs to be addressed. For a transient-induced voltage in the digital circuit to stabilize, a signal must travel through a sufficient distance in the circuit. An experiment was conducted in order to determine the minimum size of this gate distance.[4]

In order to determine the correct gate distance that must be simulated at the electrical level, a 9 picocoulomb transient was injected into a randomly selected node. Initially, all gates within five gate distances from the target node were simulated at the electrical level. Next, the same injection was made with all gates within four gate distances from the target node, simulated at the electrical level. A logic comparison was made to verify the consistency between these two simulation results. Similar injections were performed with 3, 2 and 1 gate distance(s) simulated at the electrical level and again the logic comparisons were

---

[4] The gate distance is defined as the number of levels of gates between two nodes in the circuit.

PRECEDING PAGE BLANK NOT FILMED

injection locations. Again, analysis of the simulation results with different gate distances, simulated at the electrical level, was performed. Table 1 summarizes these results. For example, for combinational circuits, with a charge level of 7 picocoulombs, at least three gate distances from the point of injection need to be simulated at the electrical level for accurate results. In general, for combinational circuits, up to three gate distances from the injection point are need to be simulated at the electrical level. Transients occurring near latches require up to four gate distances to be simulated at the electrical-level analysis.

Table 1: Accuracy Experiment Results.

Minimum Gate Distance Needed in Electrical Level Analysis						
Charge Level	**Combinational Circuit**	**Latch Distance (Gate Distance)**				
		1	2	3	4	5
1 picocoulombs	1	2	1	1	1	1
3 picocoulombs	2	2	3	2	2	2
5 picocoulombs	2	3	3	2	2	2
7 picocoulombs	3	3	4	4	3	3
9 picocoulombs	3	3	4	4	3	3

CHAPTER 4.

IMPACT OF TRANSIENTS

Table 2 summarizes the overall impact of transients in the range 0.5 to 9.0 picocoulombs. In the table, a first-order error is defined as one which occurs during the first clock cycle following a transient fault injection; second- and higher-order errors are those occurring during the second and subsequent clock cycles.[6] The second column shows the number of fault injections which result in errors. The third column shows the total number of the resultant errors. For example, out of 2100 fault injections, one or more first-order latch errors occurred in 470 cases (22.4 percent), and a total of 2149 latch errors were observed.

Table 2: Impact of Transients

Transient Fault Severity			
Type	Occurrences	Total Error Count	Percentage
Injected Transients	2100		100%
First-Order Latch Errors	470	2149	22.4%
Second-and Higher-Order Latch Errors	120	1829	5.7%
First-Order Pin Errors	255	1168	12.1%
Second-and Higher-Order Pin Errors	90	839	4.3%
Functional Errors	193	747	9.2%

---

[6] Transients modeled in the experiment last no longer than one clock cycle. Thus, no latch error can occur from direct propagation after the first clock cycle. This is typical of effects of cosmic rays and the like.

A number of issues relating to the fault sensitivity of the chip are highlighted by these data. First, they show that over 20 percent of the injections result in latch errors. Given that a transient results in a latch error, the chance of multiple errors is high (an average of 4 latch errors per transient). The existence of such multiple latch errors is potentially a serious problem since these errors can subsequently propagate to the pins and lead to multiple failures.

In addition, even though only 25 percent (120 out of 470) of the latch errors propagated past the first clock clycle (i.e., the first-order), each such propagation can result, on the average, in about 15 latch errors, thus further intensifying the propagation problem. An effect of second- and higher-order latch errors is an increase in the probability of functional errors (erroneous control signals or data, which result in an alteration of the microprocessor functions).

There is almost a 10 percent chance of having functional errors. Over one-third of the total number of functional errors were due to transients in the ALU unit. Further analysis of the error data showed that a significant number of functional errors due to transients in the ALU unit were due to first-order effects.

This is because transients that latch directly on the ALU registers result in an immediate alteration of address or data information. Functional errors caused by second- and higher-order effects of transients were more dispersed among different functional units. A relationship between the second- and higher-order latch errors and functional errors is discussed further in Section 5.1.

From Table 2, the percentage of first-order pin error occurrences is significant (over 10 percent). Given a pin error, the chance of recurrence during the subsequent clock cycles is relatively high (90/255) and each propagation can result, on the average, in approximately 9 pin-errors (in comparison, there are approximately 4 pin errors resulting from the first-order propagation).

4.1. Charge Level Analysis

This section quantifies the impact on the chip of the charge level in a transient. Statistical analysis of the the error data was performed to determine the effect of different charge levels in the injected transients on the severity of latch, pin and functional errors. Figure 4 shows the frequency of latch, pin and

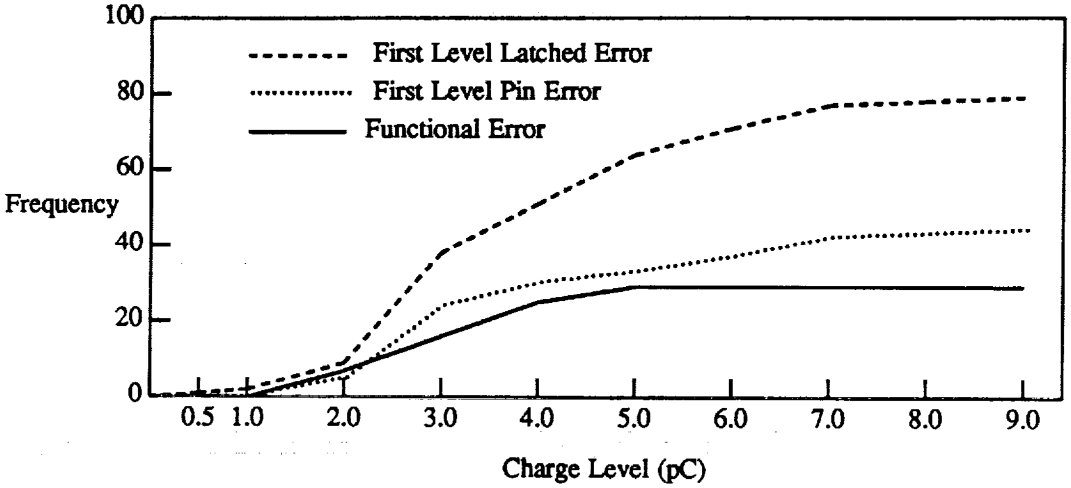


Figure 4: Error Frequency by Charge Level.

functional errors as a function of the charge level in the injected transient. First, note that beyond 7 picocoulombs, the number of error occurrences remains relatively constant, i.e., additional charge does not result in an increase in the error probability. This is because, at this charge level, essentially all the latches in the propagation path have been affected (i.e., hold erroneous values).

For latch and pin errors there is a charge threshold of 2 picocoulombs, at which a sharp increase in error activity occurs. Over 95 percent of the latch errors occurred at charge levels greater than 2 picocoulombs and 100 percent of the pin errors were observed for charges at or above 2 picocoulombs. For functional errors, however, the threshold is not so well defined.

This is most likely due to the fact that functional errors can also result from second- and higher-order latch errors (in addition to being caused by the first-order effect of a transient). The higher-order effects, of course, are not charge dependent, hence a charge threshold does not occur. Figure 5 shows the frequency of second- and higher-order latch errors and the functional upsets. Note that the frequency of the second and higher order latch errors also lacks the distinctive charge threshold.

Figure 6 shows, for each functional unit, the first-order latch and pin error distributions by the charge level in injected transients. For charge levels above the threshold, the ALU and the watchdog units have the highest probability.

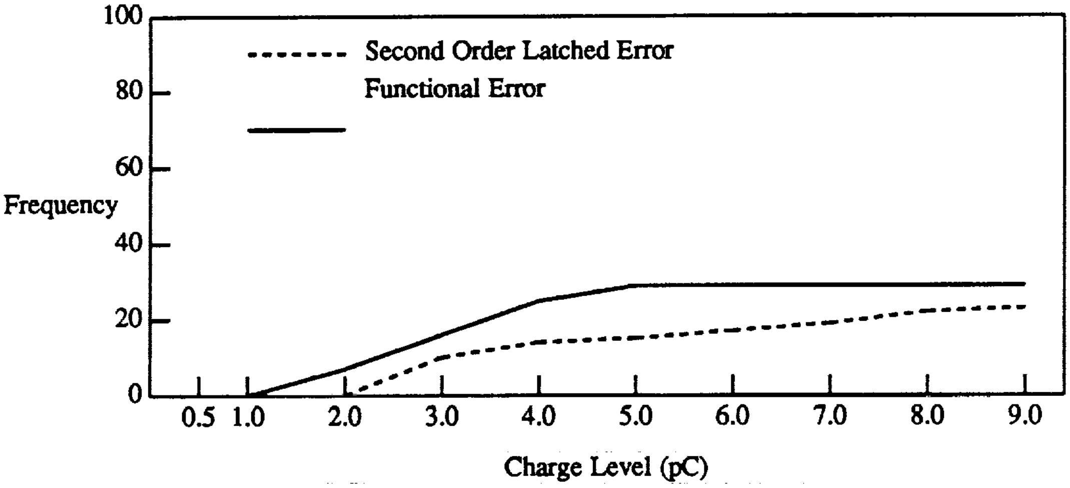
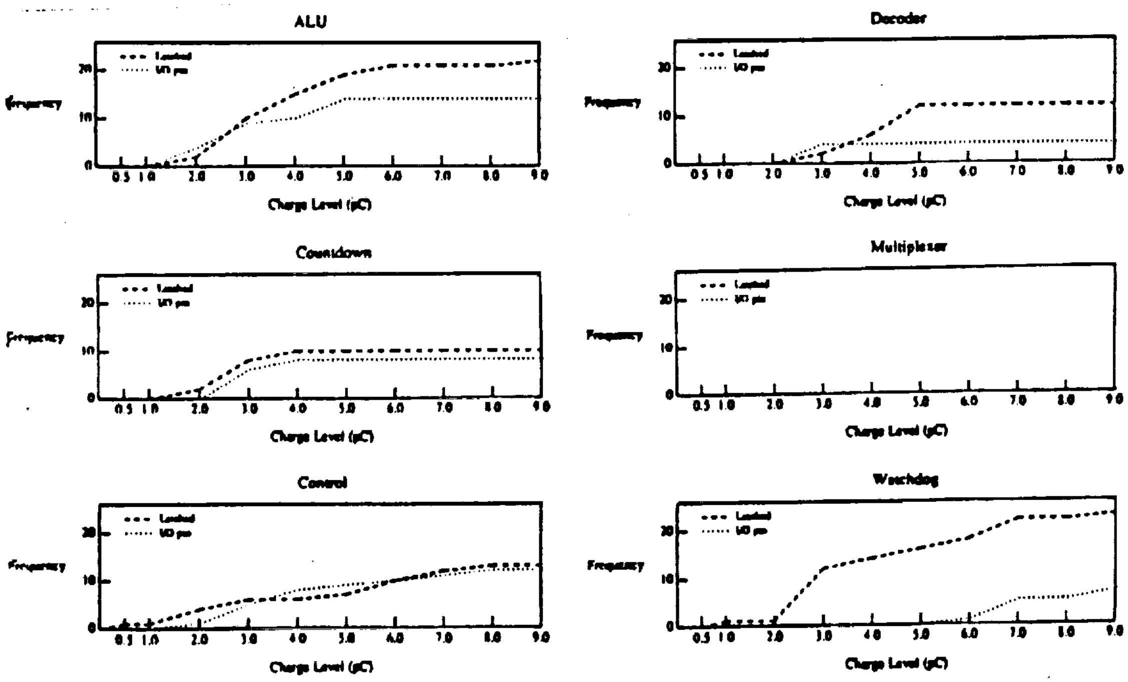


Figure 5: First-Level and Functional Error Frequency by Charge-Level.

The watchdog unit had high latch error occurrences, but pin errors occurred only for charges above 6 picocoulombs. The reason is that although an error can quite easily be latched in the numerous feed-back paths in the watchdog, it does not propagate to the external pins.

The decoder unit showed a relatively low pin error propagation probability. The chance of transients below the threshold being latched is generally small, except for the control unit. In the control unit, the possibility of having latch errors is high, even at 2 and 3 picocoulombs. The relatively small capacitive loading of the feed-back paths to the latches in the control circuit explains this low charge sensitivity.

ORIGINAL PAGE IS
OF POOR QUALITY



ALU

Decoder

Countdown

Multiplexer

Control

Watchdog

Figure 6: Error Distribution by Charge Level for Each Functional Unit.

As shown in Figure 6, the multiplexer does not have any latch or pin errors. This is so because the electrical nodes in the multiplexer unit have high capacitances due to their large number of fanouts.

CHAPTER 5.

EMPIRICAL MODELS

Fault propagation usually occurs because errors can be latched and then migrate to different sections of the chip. A latch error can stay latent and undetected until it migrates to the pins at a later time. The additional internal propagation between latches can increase the probability of generating functional upsets. Thus, a characterization of the latch-to-latch fault propagation patterns is important. A latch error can either relatch, propagate out to the I/O pins and/or disappear in each clock cycle.

5.1. Error Latency

To characterize the latency of transient faults in the circuit, the expected time (in clock cycles) for an injected transient to migrate to the pins was calculated. The expected error latency was defined as the mean value of the interval between the time of fault injection and the time at which a resultant pin error occurred. Table 6 shows the expected error latency for transients in different functional units. The expected error latency for transients in the control unit is the highest. This is because the majority of the pin errors, due to transients in the control unit, resulted from latch errors. Note that the mean latency for pin errors in the countdown and the decoder units is less than a clock cycle. All pin

Table 6: Error Latency

Mean Error Latency	
Functional Unit	**Clock Cycle**
ALU	1.43
Countdown	0.23
Control	4.67
Decoder	0.14
Multiplexer	-
Watchdog	1.94

errors in these units resulted directly from the injected transients (i.e., no pin errors in these units were due to latch errors). Thus, the mean latency for the pin errors in the countdown and the decoder units is simply the signal propagation delay from the injected location to the external pins. No pin errors were observed for transients in the multiplexer unit.

5.2. Error Propagation Model

The propagation of the latch errors in time (in clock cycles) for the control unit is illustrated in Figure 7. In this figure, the x-axis represents the clock cycles from the fault injection time and the y-axis represents the total latch error count for each clock cycle. It can be seen that, given a certain number of latch errors in the first clock cycle, the number of latch errors degenerates significantly until the fourth clock cycle. At approximately the fifth clock cycle, the number of errors rapidly multiplies. Thus, despite the fact that only on a few occasions

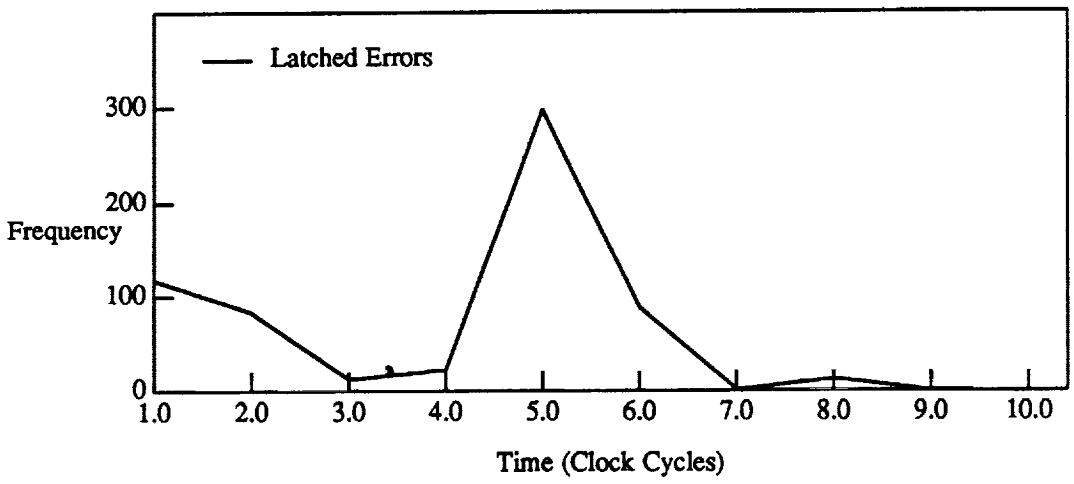


Figure 7: Latched Errors in Time.

do the latch errors last until the fifth clock cycle, when they do, the number of errors is large. This is because at this time period, the error signal enters a unit with a large number of latches and high fan-out, e.g., the ALU registers. After the sixth cycle, the number of errors degenerates significantly until finally disappearing after the eighth cycle. Thus, the impact of latch errors lasts at the most up to 8 clock cycles from the time of fault injection.

For analysis purposes, the clock cycles in which the number of latch errors increases in comparison with previous cycle are defined as "error explosion" cycles. Clock cycles in which the number of errors decreases in comparison with the previous cycle are defined as "error degeneration" cycles. In Figure 7, an error explosion occurs in the fifth clock cycle. The chance of functional errors and pin errors is likely to be maximal during this period of error explosion. In the sixth clock cycle (a degeneration cycle), the number of latch errors decreases to about one-third of the number in the fifth clock cycle.

A model to depict the process of error explosion and degeneration for the overall system is shown in Figure 8. The model is derived from the measured data to quantify the dynamics of the error propagation in the system. As seen from the model, an injected fault either becomes latched (represented by "latch-error" state) or has no impact on the circuit (represented by "fault-free state").

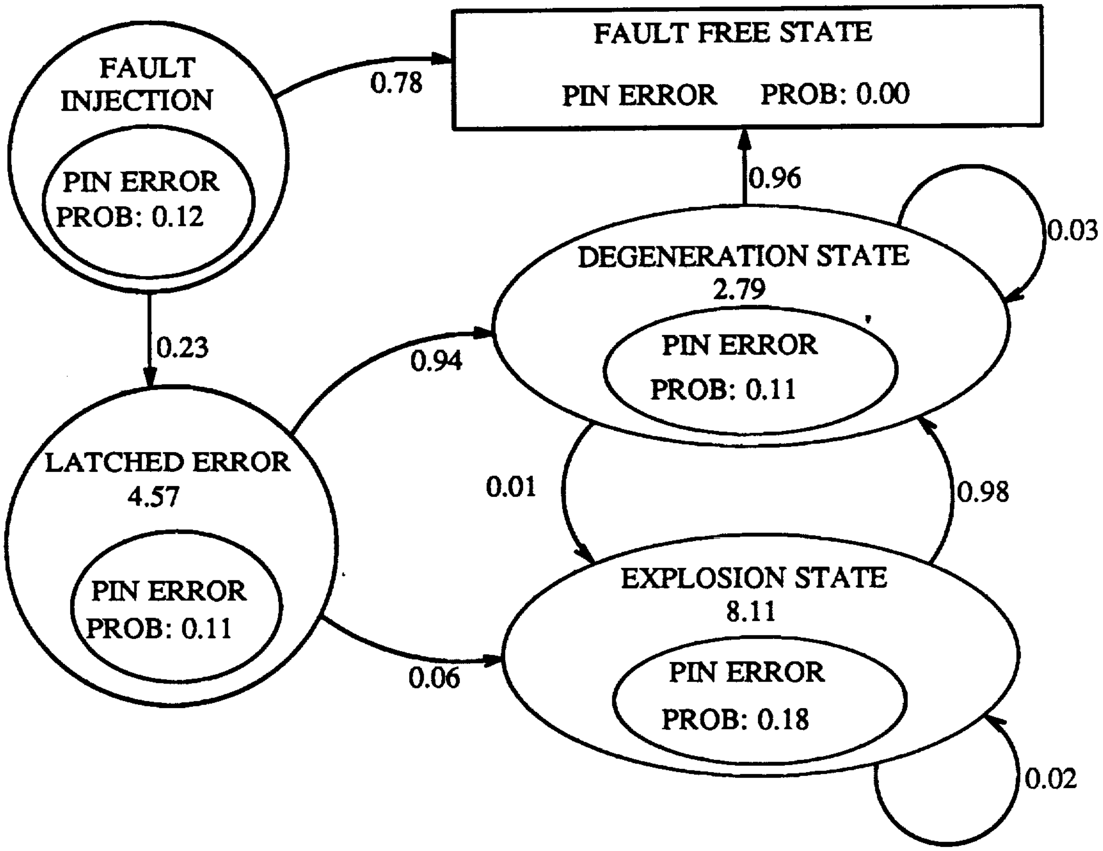

FAULT INJECTION
PIN ERROR
PROB: 0.12
0.78
FAULT FREE STATE
PIN ERROR PROB: 0.00
0.96
0.03
0.23
DEGENERATION STATE
2.79
PIN ERROR
PROB: 0.11
0.94
LATCHED ERROR
4.57
PIN ERROR
PROB: 0.11
0.01
0.98
EXPLOSION STATE
8.11
PIN ERROR
PROB: 0.18
0.06
0.02

Figure 8: Error Explosion/Degeneration Model.

The "explosion" state represents the situation where the number of latch errors in the current clock cycle is greater than that in the previous cycle. The "degeneration" state represents the opposite scenario, i.e., the number of latch errors decreases. The value assigned to a state is the average number of latch errors in that state.

Given a transient fault, there is approximately an 80 percent chance of having no impact on the chip. Although the probability of a latch error resulting in an error explosion is small (0.06), when it does occur, the average number of latches holding an erroneous value is large (8.11), i.e., although the explosion event rarely occurs, it is potentially disastrous. The probability of latch errors, in the explosion state, causing pin errors is higher than that for the degeneration state (0.18 compared to 0.11). This is clearly so because, with the larger number of latch errors, the probability of error propagation to the pins is increased. After an explosion, there is a 98 percent chance of latch errors degenerating and then becoming fault free.

In summary, the probability of sustained explosion is very low with 0.02 probability, i.e., the chance of uncontrolled propagation is small, thus the overall impact of a transient is well contained. Further, if no latch error occurs within 8 clock cycles, no significant damage is likely to happen. Thus, limited roll-back recovery techniques may be very successful.

5.3. State Transition Model

The foregoing section presented an analysis of the forward propagation in time of an injected transient. This section examines the question: given a latch error in a unit, where did it come from? The question of the internal module-to-module latch error propagation is addressed. It will be seen that the results of this analysis are useful in identifying several critical aspects of the system. Some examples include the identification of the critical error propagation paths, the determination of the module most sensitive to fault propagation and the module with the highest potential for causing external pin errors.

Figure 9 shows a state transition diagram, based on the measured data, to quantify the inter-module latch-error propagations. In the figure, the states represent error conditions in the specified functional units. Note that the model is of the inverse Markov[7] type. Thus, given a latch error in a specified unit, the model shows the probabilities that each of the other functional units are the likely error sources. For example, in the figure, given an external pin error, the probability of the ALU being the error source is 0.16; the probability of the control unit being the error source is 0.27.

---

[7] A normal forward transition Markov model can not be used to describe latch error propagation since a latch error can propagate out to multiple locations at once, i.e., a latch error can propagate to both the external pins and other latches in the circuit. Thus, an inverse Markov model is used to describe the transition from the error source to error for different states.

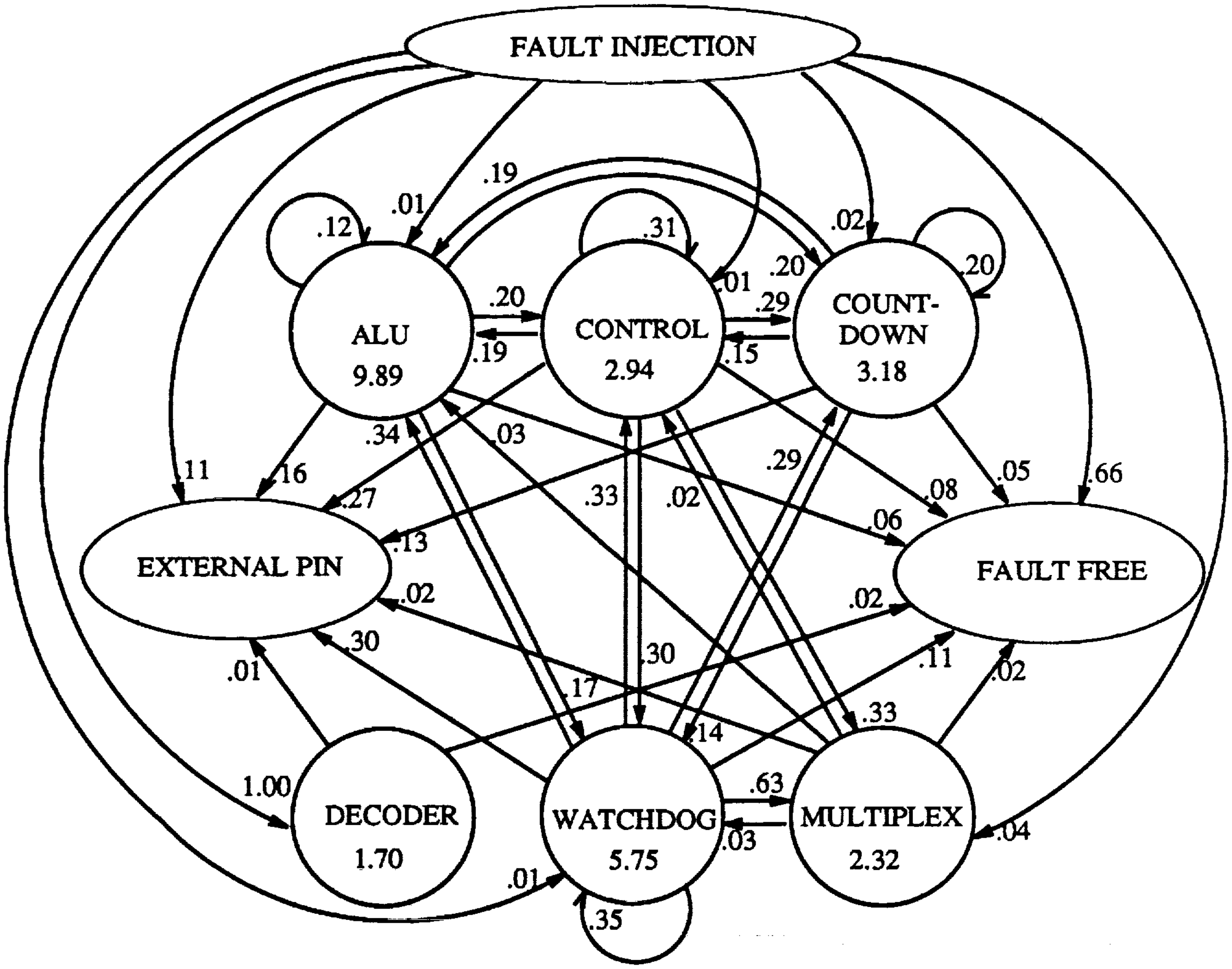

FAULT INJECTION
ALU
9.89
CONTROL
2.94
COUNT-
DOWN
3.18
EXTERNAL PIN
FAULT FREE
DECODER
1.70
WATCHDOG
5.75
MULTIPLEX
2.32
.19
.01
.12
.31
.02
.20
.20
.29
.01
.20
.19
.15
.34
.03
.33
.02
.29
.08
.05
.66
.11
.16
.27
.06
.13
.02
.02
.30
.30
.11
.02
.01
.17
.33
.14
.63
.03
1.00
.01
.35
.04

Figure 9: State Transition Model.

It can be seen from the model that latch errors in the decoder unit do not propagate to other functional units. All latch errors that occurred in this unit are due to the direct effect of the injected transient. This is because the latches in the decoder unit are well isolated from the inputs of the other units. The probability of a latch error in the decoder unit propagating to the pins is small (0.01). Thus, the decoder is not a critical unit from a fault propagation point of view.

The model addresses several issues raised at the beginning of this section. For example, the model shows that the critical fault propagation path in the system is between the control and the watchdog units. Given a latch error in the control unit, the probability that it propagated via the watchdog unit is 0.33. Conversely, the probability of the control unit being the source for a latch error in the watchdog unit is also high (0.30). In examining the other units, it is seen that although the one-way propagation probability is high in some cases (e.g., 0.63 from the watchdog unit to the multiplexer), none has a higher two-way propagation probability. Therefore, all other factors being equal, the best way of reducing inter-module error propagations is to protect the interconnections between the watchdog and the control units. Since a significant number of functional errors result from the second- and higher-order latch errors, the system level impact of providing this protection is expected to be a decrease in the probability of functional errors.

The model also shows that the module with the highest potential to cause external pin errors is the watchdog unit. Thirty percent of all pin errors were due to the latch errors in the watchdog unit. Hence, to reduce the number of pin error occurrences, the outputs of the watchdog unit should be protected. The module most sensitive to fault propagation is seen to be the ALU unit. Of all the functional units, an error occurrence in the ALU is likely to lead to the largest number of latch errors (9.89). Applying internal retry to ALU operations may be a successful way of reducing the number of latch errors.

Finally, it is seen that the probability of an injected transient directly causes pin errors is low. More than 90 percent of the pin errors are due to second- and higher-order propagation from latch errors. Similarly the probability that an injected transient directly causing a latch error is also low. Notice that less than 5 percent of the latch errors are due to the direct propagation from the injected transients. The fact that over 95 percent of latch error occurrences are due to propagations from other latch errors makes fault propagation a critical issue from a reliability perspective.

CHAPTER 6.

CONCLUSIONS

In this thesis, the effect of transients in a microprocessor-based jet-engine controller was investigated. A design automation environment to allow the run-time injection of transients and a graphical analysis facility to visualize the fault impact on the target system were described. Transients in the range 0.5 to 9.0 picocoulombs were injected and the error data were analyzed to determine the impact of the resulting latch and pin errors. A number of methods to quantify the error propagation and the error latency within the chip were developed.

The results show that, given a transient fault, there is approximately an 80 percent chance that it has no impact (latch or pin errors) on the chip. The chance of a latch error is over 20 percent while that of a pin error is approximately 12 percent. Only 25 percent of the latch errors propagated past one clock cycle, although each such propagation resulted, on the average, in about 15 latch and 9 pin errors. More than 40 percent of the pin errors were due to multistep latch-error propagations. Approximately 10 percent of the transients caused the erroneous behavior of the microprocessor's control functions (i.e., functional errors).

In order to quantify the dynamics of the error propagation in the system, a model to depict the process of error explosion and degeneration of latch errors in the overall system was derived from the measured data. The model showed that although the probability of a latch error resulting in an error explosion was very small (0.06), when it did occur, the average number of latches holding erroneous values was large (8.11), i.e., although the explosion event rarely occurs, it is potentially disastrous. The probability of sustained error explosion was also very low (0.02), i.e., the chance of uncontrolled error propagation was small. Thus, the overall impact of a transient was well contained. Further, if no latch error occurred within 8 clock cycles, there was no significant damage to the microprocessor functions. Thus, limited roll-back recovery techniques which can keep track of the machine state for up to 8 clock cycles may be very successful.

A state transition model was also derived from the measured data to describe the error propagation characteristics within the chip and to quantify the impact of transients on the external environment. The model was used to identify and isolate the critical fault propagation paths, the module most sensitive to fault propagation and the module with the highest potential of causing external pin errors. The fault propagation path between the control unit and the watchdog unit was seen to be the most critical, indicating thereby that an increase in the fault tolerance of this link may significantly improve the system dependability. The watchdog unit was seen to have the highest potential for causing external pin errors. Of all the functional units, an error occurrence in the ALU is likely to lead to the largest number of latch errors (9.89). Protection of the ALU through retry was therefore suggested to reduce the impact of the latch errors.

REFERENCES

[1] H. Ball and F. Hardy, "Effects and detection of intermittent failures in digital systems," 1969 FJCC, AFIPS Conference Proceedings, vol. 35, pp. 329-335.

[2] R. K. Iyer and D. J. Rossetti, "A measurement-based model for workload dependence of CPU errors," IEEE Transactions on Computers, vol. C-35, pp. 511-519, June 1986.

[3] T. C. May and M. H. Woods, "Alpha-particle-induced soft errors in dynamic memories," IEEE Transactions on Electron Devices, vol. ED-26, pp. 2-9, January 1979.

[4] R. J. McPartland, "Circuit simulations of alpha-particle-induced soft errors in dynamic RAM's," vol. SC-16, pp. 31-34, February 1981.

[5] R. Johnson, S. Diehl-Nagle and J. Hauser, "Simulation approach for modeling single event upsets on advanced CMOS SRAMS," IEEE Transactions on Nuclear Science, vol. NS-32, pp. 4122-4127, December 1985.

[6] G. C. Messenger, "Collection of charge on junction nodes from ion tracks," IEEE Transactions on Nuclear Science, vol. NS-29, pp. 2024-2031, December 1982.

[7] K. G. Shin and Y.H. Lee, "Error detection process - model, design, and its impact on computer performance," IEEE Transactions on Computers, vol. C-33, pp. 529-540, June 1984.

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

[9] B. Courtois, "Some results about the efficiency of iimple mechanisms for the detection of microcomputer malfunctions," Digest, FTCS-9, The Ninth International Symposium on Fault Tolerant Computing, pp. 71-74, June 1979.

[10] M. E. Schmid, R. L. Trapp, A. E. Davidoff and G. M. Masson, "Upset exposure by means of abstraction verification," Digest, FTCS-12, The Eleventh International Symposium on Fault Tolerant Computing, pp. 237-244, 1982.

[11] J. Arlat, Y. Crouzet and J. Laprie, "Fault-injection for dependability validation," LAAS Research Report no. 88-363, December 1988.

[12] R. Koga, W. A. Kolasinski, and M. T. Marra, "Techniques of microprocessor testing and SEU-rate prediction," IEEE Transactions on Nuclear Science, vol. NS-32, pp. 4219-4224, December 1985.

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

[14] R. E. Glaser and G. M. Masson, "Transient upsets in microprocessor controllers," Digest, FTCS-11, The Eleventh International Symposium on Fault Tolerant Computing, pp. 165-167, 1981.

[15] J. Sosnowski, "Evaluation of transient hazards in microprocessor controllers," Digest, FTCS-16, The Sixteenth International Symposium on Fault Tolerant Computing, pp. 364-369, 1986.

[16] D. Lomelino and R. K. Iyer, "Error propagation in a digital avionic processor: a simulation-based study," Proc. Real Time Systems Symposium, pp. 218-225, Dec. 1986.

[17] R. Chillarege and R. K. Iyer, "Measurement-based analysis of error latency," IEEE Transactions on Computers, vol. C-36, pp. 529-537, May 1987.

[18] P. Duba and R. K. Iyer, "Transient fault behavior in a microprocessor: A case study," 1988 ICCD Proceedings, October 1988.

[19] R. A. Saleh, "Iterated timing analysis and SPLICE1," Memorandom No. UCB/ERL M84/2, Electronics Research Laboratory, University of California, Berkeley, 1984.

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

[21] J. Stephen, T. Sanderson, D. Mapper, J. Farren, R. Harboe-Sorensen and L. Adams, "A comparison of heavy ion sources used in cosmic ray simulation studies of VLSI circuits," IEEE Transactions on Nuclear Science, vol. NS-31, no. 6, December 1984.

[22] D. Nichols, W. Price, W. Kolasinski, R. Koga, J. Pickel, J. Blandford, Jr., and A. Waskiewicz, "Trends in part susceptibility to single event upset," IEEE Transactions on Nuclear Science, vol. NS-32, no. 6, December 1985.

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

THE GRADUATE COLLEGE

February 1990

WE HEREBY RECOMMEND THAT THE THESIS BY

GWAN SEUNG CHOI

ENTITLED AN EXPERIMENTAL STUDY OF FAULT PROPAGATION IN A JET-ENGINE CONTROLLER

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF SCIENCE

R. K. Iyer
Director of Thesis Research

N. Narayana Rao
Head of Department

Committee on Final Examination†

Chairperson

† Required for doctor's degree but not for master's.

O-517

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

GRADUATE COLLEGE DEPARTMENTAL FORMAT APPROVAL

THIS IS TO CERTIFY THAT THE FORMAT AND QUALITY OF PRESENTATION OF THE THESIS
SUBMITTED BY GWAN SEUNG CHOI AS ONE OF THE
REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE
ARE ACCEPTABLE TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING.
Full Name of Department, Division or Unit

7 February 1990
Date of Approval

Departmental Representative