55 4839
P44

November 1989
NAG1-602

UILU-ENG-89-2240
CSG-116

COORDINATED SCIENCE LABORATORY
College of Engineering

LANGLEY
GRANT
IN-07-CR
256753
38P.

TRANSIENT FAULT BEHAVIOR IN A MICROPROCESSOR: A CASE STUDY

Patrick Duba

(NASA-CR-186240) TRANSIENT FAULT BEHAVIOR IN A MICROPROCESSOR: A CASE STUDY (Illinois Univ.) 38 p CSCL 21E
N90-17633
Unclas
G3/07 0256753

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Approved for Public Release. Distribution Unlimited.

TRANSIENT FAULT BEHAVIOR IN A MICROPROCESSOR: A CASE STUDY

BY

PATRICK DUBA

B.S., University of Maryland, 1985

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineerir.g
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1989

Urbana, Illinois

"This work was supported in part by NASA grant NAG 1-602 and in part by the Semiconductor Research Corporation"

ABSTRACT

This thesis describes an experimental analysis to study the susceptibility of a microprocessor-based jet engine controller to upsets caused by current and voltage transients. A design automation environment which allows the run-time injection of transients and the tracing from their impact device to the pin level is described. The resulting error data are categorized by the charge levels of the injected transients by location and by their potential to cause logic upsets, latched errors, and pin errors. The results show a 3 picoCouloumb threshold, below which the transients have little impact. An ALU transient is most likely to result in logic upsets and pin errors (i.e., impact the external environment). The transients in the countdown unit are potentially serious since they can result in latched errors, thus causing latent faults. Suggestions to protect the processor against these errors, by incorporating internal error detection and transient suppression techniques, are also made.

ACKNOWLEDGMENTS

I would like to thank Mr. Victor Carreno at NASA Langley Research Center for many useful discussions. Professor Resve Saleh for his help with SPLICE, the design staff at Hamilton Standard for providing circuit details for the HS1602 processor, and Professor Ravi Iyer, my advisor, who spent many hours going over this research, providing valuable insight and technical advice. Thanks are also due to Gwan Choi, Sharon Peterson and Rochelle Hochman for their assistance in preparing this manuscript.

TABLE OF CONTENTS

Page

1. INTRODUCTION

Transients in computer system hardware can occur from sources such as power transients, capacitive and inductive crosstalk and cosmic particle interventions. Previous studies [Iyer86] have shown that approximately 80% of computer system failures can be attributed to transients. Thus, a study of transient fault propagation is essential for the design of reliable computer systems. The impact of transients is difficult to model since the mechanisms involved in their generation and propagation are highly complex. An experimental study can provide valuable insight and help develop a structured basis for analytical modeling.

This thesis describes an experimental analysis to study the susceptibility of microprocessor functional units to upsets caused by current and voltage transients. The resulting transient fault propagation throughout the processor is also analyzed. The target system for the study was a microprocessor-based digital jet-engine controller. The processor (an HS1602) was designed by Hamilton Standard for commercial aircraft and made available to NASA Langley AIRLAB. The general objective of this study was to develop a design automation environment which would allow the run-time injection of transients and their tracing from the device to the pin-level.

Transients of varying peak current levels were injected into all the functional units. The resulting error data were categorized by charge levels of the injected transients and by their potential to cause logic upsets, latched errors, and pin errors. Probabilities of different types of error activity were computed, and an analysis of variance was performed. Results show that charge levels below 3 picoCoulombs have no measurable impact on the processor. The functional units most susceptible to the transients are identified. The ALU and the countdown unit were found to be the critical units. Suggestions for the improvement of the processor's transient fault sensitivity are made.

The next chapter discusses related research. Chapter 3 contains a detailed description of the experimental procedure and measurements. Chapter 4 presents an assessment of the severity of transient faults by charge-level and location. An analysis of fault propagation is presented in Chapter 5. The final section highlights the important results and draws the key conclusions.

2. 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 [Ball69] showed that nearly 90% of failures were transient in nature. In [McConnel79] statistical distributions of transients were derived from automatic error logs on DEC systems. Recent studies using failure data from IBM mainframes reported in [Iyer86] also showed that over 80% of major system errors were transient in nature. Furthermore, a strong relationship was found between 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 were first reported in [May79] where the behavior of alpha-particle-induced soft errors is explored. Simulation techniques for modeling the device level effects of cosmic particle-induced transients have been developed in [McPartland81] and [Johnson85]. In [McPartland81] a SPICE circuit with a current source is used to represent the collected alpha-generated charge. In [Johnson85] a simulation technique for modeling the ion shunt effect was developed. An approximate analytic solution which models a resultant current transient is developed in [Messenger82]. 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 our study.

A series of experiments aimed at error analysis through the physical insertion of faults have been 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 [Shin84a,b]. Experience gathered from these studies shows that the data generated can provide considerable insight into error manifestation. Another useful study is [Courtois79] which describes a simulation experiment to determine the efficiency of a number of error-detection mechanisms.

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 [Koga85]. In [Cusick85] a detailed analysis is presented of the vulnerability of the Z80 microprocessor to ion-bombardment. Studies have also focused on the efficiency of methods for software detection of transient faults. An approach that involves the development of a state-transition matrix to describe the response to transient faults is described in [Glaser81]. In [Sosnowski86] 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 [Lomelino86]. 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/microinstruction activity. The investigation presented a methodology for quantifying error propagation from the gate to pins for stuck-at faults. In [Chillarege87] new experiments to study fault and error latency 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 to the output pins. We have attempted to develop a comprehensive design automation environment for quantifying the impact of transients from the device to the pin level. Apart from further knowledge of transient fault propagation in microprocessors, this information is crucial for quantifying the vulnerability of microprocessors to transients.

3. THE EXPERIMENT

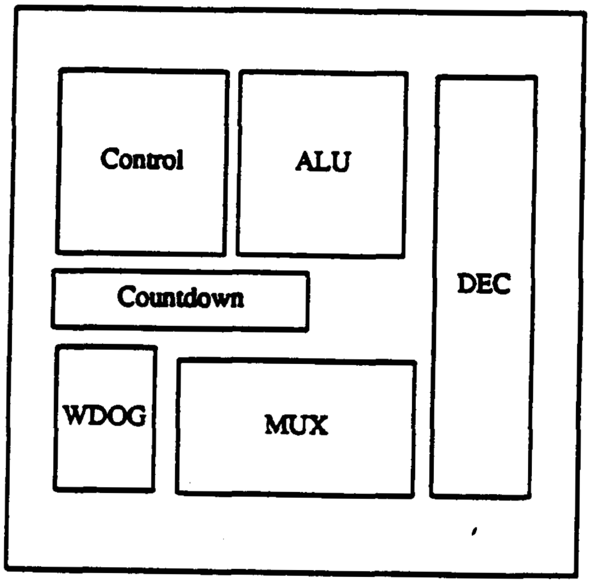
3.1. Synopsis

The microprocessor used in our study is part of a real-time control system which monitors and controls the functioning of a jet engine currently used in commercial aircraft. The system has a dual channel control which incorporates a variety of fault-tolerant design features integrated at different levels. Some of these features include a comparison of digital and analog I/O parameters, software checks, memory checks and self-test.

The control system, under study, periodically samples engine parameters such as temperature, fuel consumption, pilot inputs (from throttle and switches) and other external inputs (such as air speed). The sampled data are discretized and processed by the microprocessor, and the results are used to control the engine functions and to drive display indicators. The equations describing the control functions are incorporated into the application code which resides in EPROMs. The control system architecture thus includes microprocessors, memory units, communication channels and A/D and D/A converters. In this study we simulate the microprocessor and its associated memory with a view toward quantifying the impact of transient errors.

The microprocessor consists of seven functional units. The arithmetic and logic unit (ALU) contains six registers and can perform double precision arithmetic operations. The control unit is made up of combinational logic and is responsible for issuing signals which control the operation of the ALU. The decoder unit decodes memory enables and I/O signals. The multiplexer provides the discrete lines and buses. The countdown unit provides the chip-wide, clock signals. The watchdog unit provides synchronizing signals for the dual processor and a reset in case of parity error or a failure of the software sanity timer to be reset. The application code periodically resets the software sanity timer (a failure to reset the software timer indicates an erroneous code execution sequence). A functional diagram of the HS1602 is shown in Figure 1. The chip was implemented in a three-micron CMOS gate array and has a six-MHz clock cycle.

Figure 1. Functional Diagram of the HS1602



3.2. The Simulator

SPLICE1, a mixed-mode simulator, was chosen for use in this study. The choice was motivated by the need to perform an accurate and fast analysis at both the electrical level (where transients were injected) and at the logic level (to permit simulation of the entire processor). The implementation and algorithmic details of SPLICE1 are given in [Saleh84] ; only a brief description is given here. The electrical analysis in SPLICE1 is based on the method of Iterated Timing Analysis (ITA). This technique incorporates a nonlinear relaxation method together with event-driven selective tracing. The ITA method has been shown to be as accurate as SPICE2 [Saleh87] (assuming identical device models) and can provide a speedup of up to two orders of magnitude. The logic analysis in SPLICE1 is performed using a relaxation based method that uses MOS oriented models. Accuracy is further enhanced by providing virtually unlimited levels of signal strength that can be associated with each of the logic states. This approach allows a correspondence between the electrical output conductance and the logic output strength. Accurate delay handling is achieved through a fanout-dependent delay model which is capable of handling first-order effects. A number of enhancements were made to the basic SPLICE1 simulator in order to efficiently handle transient fault injection and perform fault propagation analysis. These are outlined in the next section.

3.3. Simulator Enhancements

Enhancements to SPLICE1 were required to facilitate our investigation and provide a design automation environment suitable for large-scale error analysis. Since we were conducting a statistical study, and over a thousand simulations were needed to be performed, the usual method of injecting transients [Diehl82] by modifying the circuit description was unsatisfactory. A new technique which modified the SPLICE1 relaxation algorithm and permitted injection of transients without explicit modification of the circuit description was developed. The method is equivalent to a run-time modification of the circuit whereby a current source is added to the circuit. Since the injected current source is specified as a mathematical function, the transients can be of varying shapes and durations. The method is not limited to single fault injection, and it also allows the injection of transients at multiple nodes and at different times. Transient fault injection at a node[1] is equivalent to altering the voltage level of the node over the time interval of the injected current waveform. A logic fault can occur because the injected current transient may cause the node voltage to vary beyond the logic threshold and thus change the logic state of the node.

SPLICE1 originally permitted traces of a maximum of 16 nodes. The HS1602, however, has over 4000 nodes when described completely at the logic level. Thus, for a comprehensive study of fault propagation in the microprocessor, it was necessary to monitor all the circuit nodes. Accordingly, a new facility for tracing all internal nodes was also designed and implemented. Additionally, new element models including

[1] A node is defined as a point of interconnection between electrical elements or logic elements or a combination of electrical and logical elements.

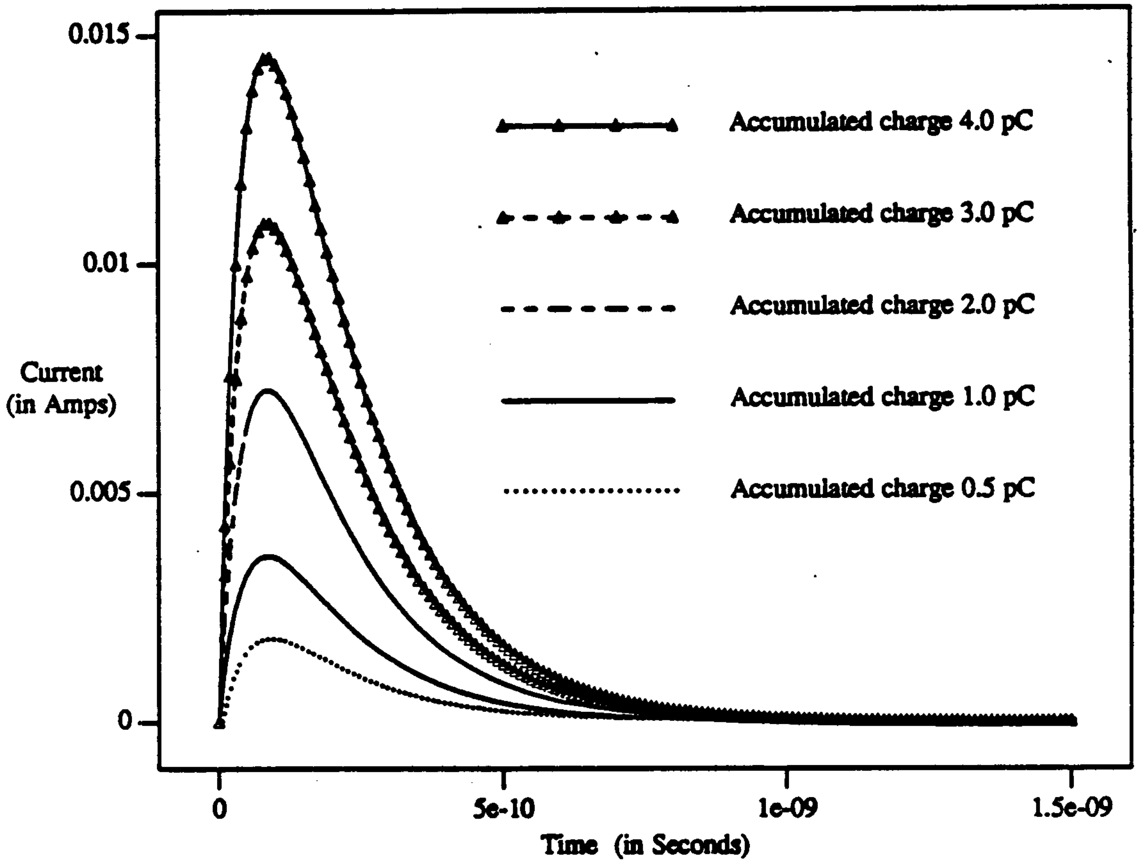
functional ROM and RAM models were also added.

3.4. Transient Fault Injection

Transients in integrated circuits can originate from a number of sources. The tools described in the foregoing sections allow the injection of transients as user specifiable current waveforms injected into the electrical analysis of SPLICE1.

The specific waveforms injected in our study follow the functions proposed in [Messenger82], which depict the current transient response for an ion particle penetration of a diffusion area. Figure 2 shows the five current waveforms used in the experiment. The injected current transients correspond to charge accumulations of 0.5, 1.0, 2.0, 3.0 and 4.0 picoCoulombs.

Figure 2. Five Current Waveforms



Several modifications were made to SPLICE1 to permit fault injection in run-time. In order to describe these modifications, the circuit simulation procedure in SPLICE1 is outlined below:

```
Process BLT circuit description file to produce input file.
Simulate input file with SPLICE1.
Process the SPLICE1 output file.
```

A simplified algorithm which includes the electrical analysis routine (Iterated Timing Analysis) is given below:

```
Begin Simulation
While (scheduled node change events in the time queue)
    For (each fanout list in the queue for the current time)
        For (each element in the fanout list)
            For (each fanout node i of the element)
                For (each fan-in element j of node i)
                    If (element j is an electrical element) then
                        Begin electrical analysis
                        Set sum of current Isum = 0
                        Set sum of conductance Gsum = 0
                        For (each fan-in node k of element j)
                            Gsum = Gsum + equivalent conductance at node k
                            Isum = Isum + total current flowing into node k
                            Calculate new change in voltage for this iteration
                            Calculate new node voltage
                            Determine if node has converged
```

Modifications were made to the event queue processing, the electrical analysis and other sections of the code. These modifications allowed the simulator to determine the time at which a transient was to be injected and to alter the electrical evaluation for transient injection. In addition to the usual SPLICE inputs, the modified version requires the name of an external file which contains the node(s) at which the transient(s) are to be injected, the time of injection and the current waveform parameters. A simplified algorithm for the modified simulator with added sections highlighted is given below:

```
Read fault injection file
Begin Simulation
While (scheduled node change events in the time queue)
    At each time step schedule any fault injection nodes
    For (each fanout list in the queue for the current time)
        For (each element in the fanout list)
            For (each fanout node i of the element)
                For (each fan-in element j of node i)
                    If (element j is an electrical element) then
                        Begin electrical analysis

                        If fault injection node
                            Call injection current function
                            Set Isum to the returned value
                        Else
                            Set sum of current Isum = 0

                        Set sum of conductance Gsum = 0
                        For (each fan-in node k of element j)
                            Gsum = Gsum + equivalent conductance at node k
                            Isum = Isum + total current flowing into node k
                            Calculate new change in voltage for this iteration
                            Calculate new node voltage
                            Determine if node has converged
```

In summary, the modified simulator allows transient injections without any explicit modification of the circuit description and a resulting additional recompilation for the simulator. The algorithm can also be used for multiple transient fault injections at multiple nodes.

3.5. Experimental Measurements

Our simulations were based on the model parameters used in the HS1602 design and the capacitances extracted from the circuit layout. The simulations involved the microprocessor executing the initialization phase of its application code. During the initialization phase, the processor accesses the external ROM for instructions and another external ROM for initialization parameters. Arithmetic processing and address generation are also performed. Care was taken to ensure that all the functional units at which transient fault injections were made were exercised. Fault injections were made at seven randomly chosen nodes in six of the functional units. For each node, all five charge levels were injected. Each charge level was injected at five different time-points during the execution of the application code sequence. This amounted to over 1000 fault injections/simulations.

Developing the postprocessor for the trace data was a novel project in itself. A trace facility was added to SPLICE1 which was capable of monitoring all the internal and external nodes. The error-data for analysis was generated by comparing each faulted simulation with a fault-free simulation. An error event was defined as either a logic state change or a voltage level change large enough to cause a node to be rescheduled for evaluation at a future time. The trace data for each event consisted of the time of the event, the new and previous voltage levels (electrical nodes) or the new and previous logic levels and logic strengths (logic nodes). Several programs were written to process the trace data. The faulted simulation trace files were processed to extract all of the fault-event information. The fault information consisted of the fault-free values, the faulted values, the times, and the node name. Each fault-event was also classified as either a timing error (premature or late firing) or a value error. The fault information files were then processed by a series of programs that collected statistics on groups of nodes, e.g., the input nodes to the ALU. This postprocessing also classified the collected statistics by charge level and time of fault injection and determined the following:

1.) Upsets: Fault injections which resulted in a voltage transient large enough to constitute a logic level error.

2.) Latched Errors: Fault injections which resulted in voltage transients which caused errors in the first level latches.

3.) Module Errors: Fault injections which resulted in voltage transients that caused errors at the I/O nodes of the functional units.

4.) Pin Errors: Fault injections which resulted in voltage transients that propagated to errors at the chips I/O pins.

A versatile statistical package (SAS) was used to perform a range of analysis on the error data. The results of the analysis are presented in the following sections.

4. ANALYSIS OF TRANSIENTS BY CHARGE LEVEL AND LOCATION

The overall results from the experiment with transients in the 0.5 to 4.0 picoCoulomb range are shown in Table 1. The table shows that the injected transients have a 41.6% chance of causing a logic upset, a 5.7% chance of resulting in a latched error (i.e., of causing a latent yet permanent error in the circuit) and a 5.6% chance of the error propagating to the pins (i.e., immediately affecting the external environment).

Table 1. Transient Fault Severity

Type of Error	Occurrences	Percentage
Injected Transients	1050	100%
Logic Upsets	437	41.61%
Latched Errors	60	5.71%
Pin Errors	59	5.61%

Thus, nearly 50% of the injected transients have no impact on the microprocessor behavior. In about 50% of the cases, a measurable change of circuit behavior occurs. However, only 11% of all injected transients cause either a permanent change in circuit behavior or affect the external environment. These results further show that transients below 3.0 picoCoulombs have little or no impact on the circuit. In the next section the impact of different charge levels in the transients is investigated. The impact on the different functional units is also evaluated.

4.1. Charge Level Analysis

This section analyzes the simulation results to quantify the sensitivity of the processor (logic upsets, latched and pin errors) to the magnitude of the injected transients. Depending on the operating environment, semiconductor devices are vulnerable to diffusion area penetration by ion particles of different energy levels. In our study, we injected charges varying from 0.5 picoCoulomb to 4.0 picoCoulombs. This represents the charge deposited by particles ranging from low energy alpha particles to the higher energy ion particles [Johnson85].

Statistical analysis of the error data was performed to determine the effect of different charge levels on upsets, latched errors, and pin errors. Figure 3 shows the distribution of logic upsets by charge level in the injected transient.

Figure 3. Distribution of Logic Upsets by Charge Level

```
                  UPSET BY CHARGE AND MODULE
                      FREQUENCY BAR CHART
    MIDPOINT
     CHARGE                                          FREQ.  CUM.
                                                           PERCENT

5.00000E-13    |AB                                     22    5.03
               |
1.00000E-12    |ABC                                    30   11.89
               |
2.00000E-12    |AABBCCDEF                              64   26.53
               |
3.00000E-12    |AAAAABBBBCCCDDEEFFF                   150   60.85
               |
4.00000E-12    |AAAAABBBBCCCCDDEEEFFF                 171  100.00
               |
               --------+-------+-------+--
                      50      100     200

                          FREQUENCY

SYMBOL MODULE       SYMBOL MODULE       SYMBOL MODULE

  A     AL            B     CD            C     CO
  D     DC            E     MX            F     WD
```

The distribution shows a charge threshold (3 picoCoulombs) at which a sharp increase in error activity occurs. Note that over 73% of the upsets were observed for charges of 3 picoColoumbs and 4 picoColoumbs. The ALU, control, and countdown are shown to be the major contributors for charge levels below 3 picoColoumbs. All the functional units make a significant contribution to the upsets at 4 picoColoumbs. Figure 4 shows a similar histogram for latched errors. Here the potential of a transient to cause latched errors is analyzed.

Figure 4. Histogram of Latched Errors

```
                LATCH BY CHARGE AND MODULE
                   FREQUENCY BAR CHART
    MIDPOINT
     CHARGE                                     FREQ.  CUM.
                                                      PERCENT
                |
5.00000E-13     |                                   1    0.06
                |
1.00000E-12     |                                   2    0.17
                |
2.00000E-12     |                                   7    0.58
                |
3.00000E-12     |AAABBBBBCCCCF                    663   39.01
                |
4.00000E-12     |AAAABBBBBBBBCCCCCCFF            1052  100.00
                |
               --------+-------+-----
                      400     800

                      FREQUENCY

SYMBOL MODULE       SYMBOL MODULE       SYMBOL MODULE

  A     AL            B     CD            C     CO
  D     DC            E     MX            F     WD
```

For each node where a transient fault injection was made, the fanout was traced to the latches. Here the critical charge level is even more distinct than for upsets. Over 99% of the latched errors occurred for charge levels of 3 and 4 picoColoumbs. The count-down, control, and ALU are seen to be the major contributors to latched errors. Even though a transient below the threshold can cause a logic upset, the probability of it being latched is small. Above the threshold, the probability of a transient being latched is dominant for the countdown unit. This dominance is probably due to its function of generating the system clocks used by most of the chip-latches.

Figure 5 shows the error distributions at the I/O pins. Once again, the critical charge level is seen with over 98% of the pin errors observed for charges of 3 and 4 pC. The ALU, control, and countdown continue to be the major contributors, with some contribution made by the decoder.

Figure 5. Error Distribution at the I/O Pins

```
                          PINS BY CHARGE AND MODULE
                             FREQUENCY BAR CHART
   MIDPOINT
    CHARGE                                                     FREQ.  CUM.
                                                                      PERCENT
                  |
5.00000E-13       |                                               0     0.00
                  |
1.00000E-12       |                                               0     0.00
                  |
2.00000E-12       |                                              40     1.11
                  |
3.00000E-12       |AAAAABBBBBBBCCCCDD                          1430    40.68
                  |
4.00000E-12       |AAAAAAAABBBBBBBBBBCCCCCCCCDDD               2144   100.00
                  |
                  --------+-------+-------+-----
                         600     1200    1800

                                FREQUENCY

SYMBOL MODULE          SYMBOL MODULE          SYMBOL MODULE

  A     AL               B     CD               C     CO
  D     DC               E     MX               F     WD
```

4.2. Analysis by Functional Unit

The impact of the injected transients on the different functional units, discussed in Section 4.1, is quantified in this section. The probability of a fault, injected into a given functional unit, causing an upset, a latched error, or a pin error is shown in Table 2.

Table 2. Probability of Upset, Latched and Pin Errors

Functional Unit	Upset	Latched Error	Pin Error
Arithmetic	0.571429	0.085714	0.131429
Countdown	0.468571	0.114286	0.080000
Control	0.474286	0.102857	0.080000
Decoder	0.308571	0.000000	0.045714
Multiplexer	0.314286	0.000000	0.000000
Watchdog	0.360000	0.040000	0.000000

These probabilities were calculated from a base of 1050 transient fault injection simulations. In calculating the probability, the error activity resulting from a transient fault is counted either as existing or not existing, i.e., the probabilities only refer to whether or not an injected fault causes an error (upset, latched or pin).

The table shows that the ALU has the highest logic upset probability, with a 57% chance of a transient resulting in an upset. This high upset probability can be explained by the small capacitive loading of many of the nodes in the bit-sliced design. Furthermore, a transient in the ALU also has the greatest probability of causing a pin error. This is probably because the ALU drives the 16 address pins. The ALU also has a significant probability of latched errors, although this is somewhat lower than those for the control unit. Thus, even though it is not the most complex unit (the control is the most complex), the ALU is seen to be very critical from a fault sensitivity viewpoint. The high latch error probability for the control unit is most likely due to the large number of ALU latches that are driven by the control unit. Transients in the countdown are most likely to result in latched errors. This is due to the fact that the countdown unit produces the clocking signals for a large number of latches in the processor.

All of the functional units have at least a 30% chance of a transient resulting in a logic upset. None of the functional units has more than a 14% chance of having an error latched or observed at the pins. However, for most functional units the probability of a latched error is greater than that of a pin error. This higher latch error probability reveals a potential weakness in the circuit from a dependability viewpoint. The fact that an error propagates immediately to the pins makes it easily detectable by external error detection circuits. A latched error, however, can stay latent and undetected until it migrates to the pins at a later time. Thus the possibility exists of computing with bad data before an error is detected at the pins. An additional concern lies in the fact that if multiple latent latched errors are discovered simultaneously, or nearly so, the result could be a catastrophic system failure.

Although the impact of a latched error is of concern, the fact that the countdown unit is responsible for a significant proportion of the latched errors offers a relatively simple solution for increasing the fault tolerance of the system. Since the countdown unit has relatively few output lines, it may be possible to monitor these lines for errors, rather than the latches themselves. An alternative approach is to harden the devices in the latches and the countdown unit to withstand higher levels of transients.

For two of the functional units that drive the I/O pins (the ALU and Decoder), the probability of a pin error is greater than that of a latched error. The multiplexer (which drives the I/O pins) showed no significant probability of latched or pin error activity. This insensitivity is due to the lack of latches in the fanout path of the multiplexer. Additionally, no pin errors occurred due to a multiplexer transient. This was due to the fact that no upsets occurred on the multiplexer nodes that directly fanned into the I/O drivers, probably due to the high capacitive load of these nodes.

5. FAULT PROPAGATION BETWEEN FUNCTIONAL UNITS

The results were also analyzed for fault propagation from one unit to another. We define external fault propagation as the probability of a transient fault in one functional unit causing error activity on the I/O nodes of the other functional units. The I/O nodes of all functional units were monitored for this purpose. Tables 3 and 4 show the results of this analysis. Table 3 compares the probability of a transient affecting the injected unit (source error prob.) with the probability that external propagation occurs (ext. error prob.).

Table 3. Probability of Source and External Functional Units Propagation

Functional Unit	Source Functional Unit	External Functional Units
Control	0.08	0.08
Arithmetic	0.137143	0.0857143
Decoder	0.0228571	0.0114286
Multiplexer	0.0857143	0
Watchdog	0.00571429	0.00571429
Countdown	0.102857	0.102857

Note that the probabilities are once again calculated based on the existence or nonexistence of any amount of error activity. The countdown unit, followed by the ALU, has the greatest probability of causing error activity at the I/O nodes of the external functional units. There is a greater than 10% chance that a transient in the countdown unit will affect at least one other functional unit. It can be seen from Table 2 that a significant amount of this fault propagation results in lasting error activity as the latched and pin error probabilities. Table 3 also shows that the countdown, watchdog, and control units have the same probability of a transient affecting I/O nodes of the other functional units as affecting their own I/O nodes. The ALU and decoder have almost half as much probability of affecting I/O nodes of other functional units as their own. The multiplexer showed no significant probability of affecting the I/O nodes of the other functional units. The countdown, watchdog, and control provide signals which are used almost exclusively within the chip. The multiplexer provides no signals that are used by the other functional units. In Table 4 results are quantified by specific external functional units.

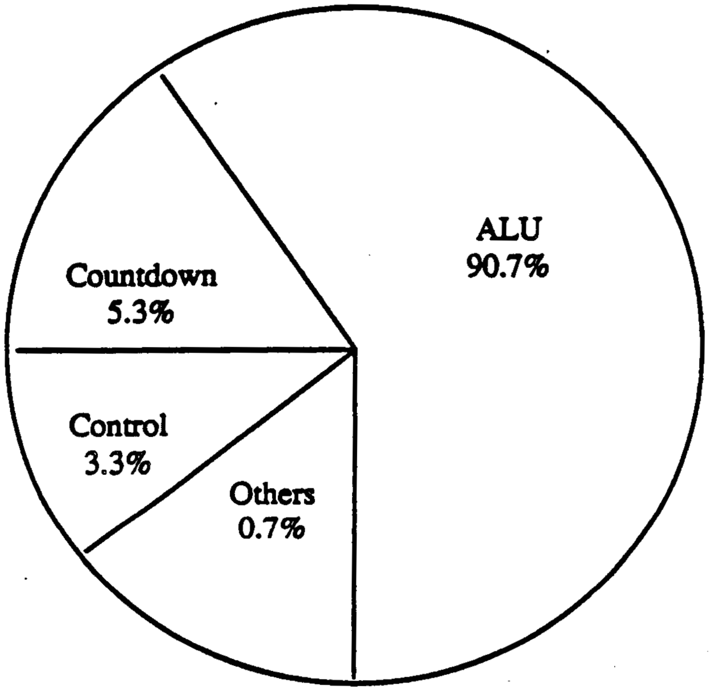
Table 4. Probability of Functional Unit I/O Node Error

Functional Unit	Control	Arithmetic	Decoder	Multiplexer	Watchdog	Countdown
Control	0.08	0.0514286	0.08	0.08	0.08	0
Arithmetic	0.08	0.137143	0.0114286	0.0857143	0.00571429	0
Decoder	0.0114286	0.0114286	0.0228571	0.0114286	0	0
Multiplexer	0	0	0	0.0857143	0	0
Watchdog	0	0	0	0.00571429	0.00571429	0
Countdown	0.102857	0.08	0	0	0.102857	0.102857

The results shown in this table indicate that the control, the watchdog, and the ALU are the functional units most affected by transients in the countdown unit, i.e., by the specific propagation path. A few observations can also be made from Table 4. First, a fault in the multiplexer and watchdog unit is least likely to propagate externally. The reason for the relative isolation of the multiplexer should be clear from the preceding discussion. With the watchdog, a possible reason for the low propagation may be due to the fact that the watchdog has a single output line which is triggered only under very specific conditions (parity error in the ROM; failure of the software sanity timer to reset). The countdown and the watchdog units are least impacted by faults in other units. This is most likely because the countdown unit has little fan-in from other units; a similar situation exists with the watchdog as well. Generally, no strong correlations are seen between the two units. Exceptions are the correlation between the countdown and the control unit, although this is only a one-way correlation. This correlation is probably due to the fact that the countdown unit produces the timing signals which determine the sequencing of the control unit. A similar situation exists between the countdown and the watchdog units. Other noteworthy correlations include the ALU and the multiplexer units. This correlation is caused by the many ALU latches that feed the multiplexer. Again, only one-way correlation is observed.

Finally, we used the analysis of variance (ANOVA) methods to quantify the sensitivity of pin-level errors to error activity in the different functional units. The objective was to identify the most critical unit from an external error propagation viewpoint. Briefly, an analysis of variance allows the determination of the impact of variations in one or more independent variables on a dependent variable. Thus, our analysis quantified the contribution of the different functional units to variations in pin-level error activity. The results of the ANOVA are shown in Figure 6. The ANOVA indicates the output pin error activity is most sensitive to error activity in the ALU.

Figure 6. Results of the ANOVA



In summary, the analysis of this section has quantified the impact of current and voltage transients occurring in the different functional units of the processor. An ALU transient is most likely to result in logic upsets and pin errors. While the logic upsets by themselves are not of much concern, the pin errors can impact the external environment and may require external error detection circuitry. The transients in the countdown unit are more serious since they can result in latched errors, thus causing latent faults. The protection of the processor against these errors requires a careful redesign of the circuit to incorporate internal error detection and transient suppression techniques.

6. CONCLUSIONS

This thesis has described an experimental analysis to study the susceptibility of a microprocessor-based jet engine controller to upsets caused by current and voltage transients. The development of a design automation environment which allows not only the automatic injection of transients in run-time, but also the evaluation of the impact of the fault as well, is described. Major results specific to the HS1602 are outlined below:

- Charge levels below 3 picoCoulombs have no significant impact on the processor or the external environment. We speculate that other processors have similar threshold levels for transient faults.
- Transients in the ALU were most likely to cause logic upsets and also had the greatest impact on output pin errors. The pin-level error activity is most sensitive to the ALU error activity.
- Transients in the countdown unit are most likely to propagate to the other units and to cause latched errors (a potentially serious problem).
- The control unit, even though the most complex, is not the most fault sensitive.
- The results also suggest the locations where it may be most cost beneficial to introduce additional error detection and tolerance capabilities. By protecting the countdown unit, a significant reduction may be possible in the occurrence of latched errors, thus reducing the possibility of catastrophic failure due to current transients.

REFERENCES

[Ball69]
H. Ball and F. Hardy, "Effects and Detection of Intermittent Failures in Digital Systems," 1969 FJCC, AFIPS Conference Proceedings, vol. 35, pp. 329-335.

[Chillarege87]
R. Chillarege and R.K. Iyer, "Measurement-Based Analysis of Error Latency," IEEE Transactions on Computers, vol. C-36, pp. 529-537, May 1987.

[Courtois79]
B. Courtois, "Some Results About the Efficiency of Simple Mechanisms for the Detection of Microcomputer Malfunctions," Digest, FTCS-9, The Ninth International Symposium on Fault Tolerant Computing, pp. 71-74, June 1979.

[Cusick85]
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.

[Diehl82]
S.E. Diehl, A. Ochoa, P.V. Dressendorfer, R. Koga, and W.A. Kolasinski, "Error Analysis and Prevention of Cosmic Ion-Induced Soft Errors in Static CMOS RAMS," IEEE Transactions on Nuclear Science, vol. NS-29, pp. 2032-2039, December 1982.

[Glaser81]
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.

[Iyer86]
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.

[Johnson85]
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.

[Koga85]
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.

[Lomelino86]
D. Lomelino and R.K. Iyer, "Error Propagation in a Digital Avionic Processor: A Simulation-Based Study," Proceedings Real Time Systems Symposium, pp. 218-225, December 1986.

[May79]
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.

[McConnel79]
S.R. McConnel, D.P. Siewiorek, and M.M. Tsao, "The Measurement and Analysis of Transient Errors in Digital Computing Systems," Digest of Papers, Ninth Annual International Symposium on Fault-Tolerant Computing, pp. 67-70, IEEE Computer Society, 1979.

[McPartland81]
R.J. McPartland, "Circuit Simulations of Alpha-Particle-Induced Soft Errors in Dynamic RAM's," vol. SC-16, pp. 31-34, February 1981.

[Messenger82]
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.

[Saleh84]
R.A. Saleh, "Iterated Timing Analysis and SPLICE1," Memorandum No. UCB/ERL M84/2, Electronics Research Laboratory, University of California, Berkeley, 1984.

[Saleh87]
R.A. Saleh, "Nonlinear Relaxation Algorithms for Circuit Simulation," Memorandum No. UCB/ERL M87/21, Electronics Research Laboratory, University of California, Berkeley, 1987.

[Shin84a]
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.

[Shin84b]
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.

[Sosnowski86]
J. Sosnowski, "Evaluation of Transient Hazards in Microprocessor Controllers," Digest, FTCS-16, The Sixteenth International Symposium on Fault Tolerant Computing, pp. 364-369, 1986.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

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-89-2240 (CSG-116)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
NASA

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 Avenue
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)*
See 7b.

10. SOURCE OF FUNDING NUMBERS

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

11. TITLE *(Include Security Classification)*
Transient Fault Behavior in a Microprocessor: A Case Study

12. PERSONAL AUTHOR(S)
Duba, Patrick

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*
November, 1989

15. PAGE COUNT
37

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
transient faults, simulation, fault injection, case study, avionic microprocessor, data analysis, latched errors

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

This thesis describes an experimental analysis to study the susceptibility of a microprocessor-based jet engine controller to upsets caused by current and voltage transients. A design automation environment which allows the run-time injection of transients and the tracing from their impact device to the pin level is described. The resulting error data are categorized by the charge levels of the injected transients by location and by their potential to cause logic upsets, latched errors, and pin errors. The results show a 3 picoCouloumb threshold, below which the transients have little impact. An ALU transient is most likely to result in logic upsets and pin errors (i.e., impact the external environment). The transients in the countdown unit are potentially serious since they can result in latched errors, thus causing latent faults. Suggestions to protect the processor against these errors, by incorporating internal error detection and transient suppression techniques, are also made.

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

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED