International Journal of Computer Applications (0975 - 8887)
Volume 162 - No.7, March 2017

Stuck-at-a-state is a Boon

Tamanna Afroze
MSc Student
Department of Computer Science and Technology
Bangladesh University of Engineering and Technology

ABSTRACT

I am trying to show stuck-at-a-state is not a hazard or damage for design architecture but can a boon for many cases. I considered different types of image analysis technique, clock rate, clock/data waveform, edge detection and RGB content's value of an image for this research to carry on. Several performability issues has been taken care for this analysis. I have shown matlab code for simulation. Distortion of an image in case of storing time and restoration time can be taken care differently. By the hardware technique, stuck the bit-value at a level we can correct the error.In order to carry on these types of analysis we are considering the fault tolerance mechanism for dependable computing system.

Keywords

RGB, Image Distortion

1. INTRODUCTION

Engineers are often called upon to design and build something that has never been tried before. Because of its novelty, the structure cannot simply be modeled after a successful example, for there is none. This was certainly the case in the mid-nineteenth century when the railroads were still relatively new and there were no bridges capable of carrying them over great waterways and gorges.A success is just thata success. It is something that works well for a variety of reasons, not the least of which may be luck. But a true success often works precisely because its designers thought first about failure. Indeed, one simple definition of success might be the obviation of failure.In fact, Nothing succeeds like success.[1]

Engineers have learned so well from failures that a major failure today is big news. We no longer expect bridges to collapse or buildings to fall in or spacecraft to explode. Such disasters are perceived as anomalous, . . . We grieve the lost lives, we search among the designers for the guilty. Yet these disasters serve the same function as the failures of an earlier era. Failures remain the engineer's best teacher, his best laboratory.[5]

Dependability concerns are integral parts of engineering design. Aspects and/or measures of dependability for computer systems include reliability, availability, performability (sustained computational capability), safety, security, testability, and maintainability. Ideally, we would like our computer systems to be perfect, always yielding timely and correct (or at least safe) results. However, just as bridges collapse and airplanes crash occasionally, so too computer hardware and software cannot be made totally immune to undesirable or unpredictable behavior. Despite great strides in component reliability and programming methodology, the exponentially increasing complexity of integrated circuits and software products makes the design of perfect computer systems nearly impossible. The field of dependable computing [096] [text] is concerned with studying the causes of imperfection in computer system (impairments to dependability), tolerance methods for ensuring correct, safe, or timely operation despite such impairments, and tools for evaluating the quality of proposed or implemented solutions. [5]

Any computer system that is designed to fail gracefully, with the minimum amount of data or program destruction, when part of the system malfunctions. Fail-soft systems close down non-essential functions and operate at a reduced capacity until the problem is resolved.

Validity is defined as the extent to which a concept is accurately measured in a quantitative study. The second measure of quality in a quantitative study is reliability, or the accuracy of an instrument.[4]
In above some paragraph I am quoting some informatiion and saying from many authors. They termed the computing, processing and error in their own way. We seldom thinks what is actually going on the bit level of our computer's processor. Many research focus on these things so thoroughly that now-a-days none care about bit-level computation except when anyone goes to design a compiler or new programming technique for underlying hardware. As computer user increases and many of them are used to tend to low-cost usage for their personal task, they seldom buy expensive computer protective care for their needs. So, I want to propose a cheap arhchitecture for cost-efficient design. Our modification methodology for Temperature Aware Microarchitecture is reconsidered for burst data. By burst data we want to show near, next and current data altogether for considering the decision making task. The reformed design diagram is as follows:
Analysis of fault detection and fault tolerance is a necessary thing for this type of architecture modification. The main things we want to take care in this work is checking fault tolerance can be done whether smartly. The paradox of our design is beyond saying. Everyone strongly says about their own work. But I can not say whether the work can fully functional as I am not testing it experimentally in the original research lab. I carried on some simulation and logical analysis which says correctness of the design.

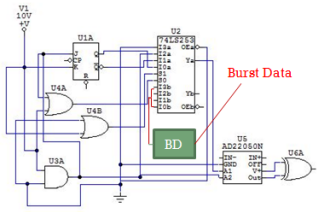

Fig. 1. Analysis of Hardware Implementation

2. RELATED WORK

[2] shows image demosaicking technique for digital camera. It is considered spatial domain image storage and restoration with electrical circuits suitability in defining the mechanism and the performance of the circuits to process the signal. I am considering similar types of research. But my one is different wholly in case of analysis but resembles in design perspective. Line segment analysis for image processing, fourier analysis and image's content viewing to restore are considered in this demosaicking mechanism.

3. MOTIVATION BEHIND WORK

The main points we want to point out in this work are limited and not sufficient:

—The research work is carried out to find the propagation error within combinatorial circuits and sequential logic gates.

—To consider stuck at 0/1 as a boon not as a danger for the circuits.

—Analysis of dependability of applications and instructions with respect to fault tolerance systems.

With this end in view we carried on several types of analysis on different types of computing needs. We described everything pictorially here one-by-one to show the things we have taken care for considering burst data.

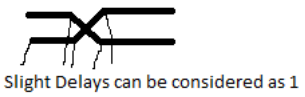

Fig. 2. Waveform shows data in operation and detection of its slight delay in order to reduce fault.

Fig. 3. Slight change to detect edge in order to detect fault.

Analysis what have been carried on figure 2-5 are:

(1) Analyse different types of faults and tolerant capability for these faults.
(2) Image analysis in case of integrated circuit design for performance analysis.
(3) Image demosaicking in case of fault tolerance analysis.
(4) How change of RGB components variation can help upgrade or increase performance of processor runtime and game engine development.
(5) Explanation of figure 5 is in the figure.
(6) Matlab code realization for performance checking.

Fig. 4. RGB values bit interpretation checking in order to fault detect.

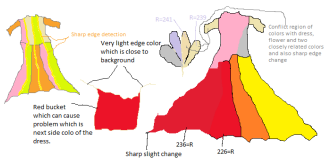

Fig. 5. Different types of fault tolerance situation.

4. PROPOSED MECHANISM

We are tolerating 3 bit error for image analysis and 1bit error for data/clock waveform. 3bit error tolerance can not hamper the image's content. In figure 4 we have shown two seperate red colors with a slight variation in the R value. The pixels vary on 7 level which is 3 bit level analysis. We can fix our burst data point to 1 level to tolerate this error. In this case we need not to fetch more data or for not switching transistor's level we can save some energy and temperature can reduce. Reduction of temperature and improving performance is our main goal behind this research. Figure 5 needs to take care differently.It is mainly for video recording. Texture change in case of capturing image can have a significant impact for this type of situation. This type of performance degradation occurs in many cases of slow computing devices. We have shown the code in the next section.

5. EXPERIMENTAL EVALUATION: MATLAB CODE REALIZATION

In matlab [3] we have checked our program for image analsis. For waveform analysis we carried on theoretic research on the original circuit.

—Gathering information from an image.

—Retrieval of information.

—Checking RGB contents value from the Image Matrix.

—Checking the difference of the nearest pixels.

—If the difference is ¡=8 we will tolerate the difference and will continue Processing Image.

—Otherwise, we will store the image and proceed to next calculation of Image Manipulation.

—Checking nearest instruction value from the Registers.

—Checking the contents of the value field for Realization.

—Manipulating the instructions based on type of Application.

Matlab code which is shown below elaborates the technique. Viewing stored image's content is shown in line 2-4. We have stored the bit value of the RGB content in a separate file and a portion of these values is shown next. The contents of the bit value shown are taken from the rightmost upper corner. We can varify these values from the original image (in figure 6) and then can check with the restored image (in figure 7).
[language=matlab]check.m

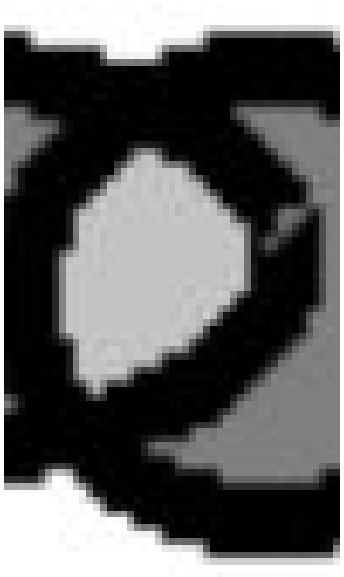

Fig. 6. Image analysed in the Matlab program.

0 0 0 18 74 125 127
2 0 0 23 76 129 127
2 9 36 66 98 131 127
0 21 80 116 121 125 127
0 21 92 131 127 125 127
1 20 81 119 125 135 127
31 63 98 123 129 124 127
64 96 118 129 130 125 127
99 123 129 127 125 125 127
121 130 127 123 124 126 127
128 129 127 128 129 129 127
125 125 128 131 129 125 127
125 124 125 128 127 124 127
132 127 123 125 128 129 127
127 127 127 127 127 127 127
127 127 127 127 127 127 127
127 127 127 127 127 127 127
127 127 127 127 127 127 127
127 127 127 127 127 127 127
127 127 127 127 127 127 127
127 127 127 127 127 127 127
127 127 127 127 127 127 127
126 125 127 130 128 124 127
131 126 123 126 128 128 127
129 126 124 126 128 127 128
127 128 130 125 109 93 95
125 128 128 110 71 36 35
98 101 102 83 40 3 2
44 46 48 40 19 0 1
1 0 0 1 1 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1
0 1 0 0 0 0 1
2 2 1 0 0 0 1
2 2 0 0 0 1 0
2 1 0 0 2 5 3
2 0 0 0 3 8 55
1 0 0 0 5 10 125
91 91 92 95 97 99 180
208 208 209 211 213 214 236
255 255 255 255 255 255 255
252 251 251 252 252 253 255

Illustration of the retrieved image from the original image's bit value is not like wholly the image we stored. Because of distortion, resolution or nearby pixels deviation image become slightly distorted from the imge stored which is shown in the following figure.

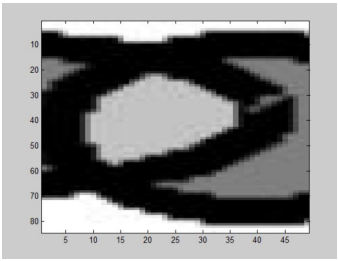

Fig. 7. Reconstructed Graph from stored Image.

6. FUTURE WORK

Many research focuses on Image analysis with fault tolerance mechanism. Image Demosaicking[2] has been studied thoroughly in the last decade. Now there is new motivation in image analysis in game programming and for GPU computation. Some new motivation are illustrated in the following paragraph.

(1) Many faults injected due to image storing, image restoration and image capture motivate to consider processor architectures.
(2) Modern multicore processors with advanced features can be a solution to consider.
(3) While considering the problem only unicore architecture is considered.
(4) Parallel programming paradigm is also a boon.
(5) New Solution: Stream Programming::: StreamIt.

7. CONCLUSION

In brief we can say that stuck-at-a-zero or stuck-at-a-one state may not always harmful for computer system design. We can tolerate errors in a marginal level say 1-bit tolerance, 2-bit tolerance, 3 bit tolerance depending on the application or computing techniques.In our case we are tolerating 1bit for clock waveform and 3 bit for

image analysis cases. I have mentioned some future work which is our next target in hardware design. It is a simple overview of an fault tolerance analysis and a simple project for the course work. We are motivated to work further for future improvement.

8. REFERENCES

[1] Henry Petroski. *Success Through Failure The Paradox of Design*. Princeton Univ Press/NJ in 2006.

[2] Bahadir K Gunturk,John Glotzbach, Yucel Altunbasak, Ronald W. Schafer, and Russel M. Mersereau. *Demosaicking:Color Filter Array Interpolation* IEEE Signal Processing Magazine,2005.

[3] www.matlab.com *Matlab*

[4] www.wikipedia.org *Wikipedia, free online encyclopedia*

[5] Behrooz Parhami *Dependable Computing: A Multilevel Approach* Appears to be in 2017.

[6] Barry W. Johnson *Design and analysis of fault-tolerant digital systems* Addison Wesley,1957