UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ERIK SCHÜLER

THREE DIFFERENT TECHNIQUES TO COPE WITH RADIATION EFFECTS AND COMPONENT VARIABILITY IN FUTURE TECHNOLOGIES

Porto Alegre
2007

ERIK SCHÜLER

THREE DIFFERENT TECHNIQUES TO COPE WITH RADIATION EFFECTS AND COMPONENT VARIABILITY IN FUTURE TECHNOLOGIES

Tese de doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), da Universidade Federal do Rio Grande do Sul (UFRGS), como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica.

Área de concentração: Automação e Instrumentação Eletro-Eletrônica

ORIENTADOR: Prof. Dr. Luigi Carro

Porto Alegre

2007

ERIK SCHÜLER

THREE DIFFERENT TECHNIQUES TO COPE WITH RADIATION EFFECTS AND COMPONENT VARIABILITY IN FUTURE TECHNOLOGIES

Esta tese foi julgada adequada para a obtenção do título de Doutor em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: ____________________________________

Prof. Dr. Luigi Carro, UFRGS
Doutor pelo Programa de pós-graduação em Ciência da Computação. Universidade Federal do Rio Grande do Sul, Brasil

Banca Examinadora:

Prof. Dr. Fernando Abel Silveira Noguerol, Universidade de La República
Doutor pela Université Catholique de Louvain, Bélgica

Prof. Dr. Antonio Petraglia, Universidade Federal do Rio de Janeiro.
Doutor pela University of California, EUA

Prof. Dra. Fernanda Lima Kastensmidt, Universidade Federal do Rio Grande do Sul
Doutora pela Universidade Federal do Rio Grande do Sul, Brasil

Prof. Dr. Altamiro Amadeu Susin, Universidade Federal do Rio Grande do Sul
Doutor pelo Institut National Polytechnique de Grenoble, França

Prof. Dr. Gilson Ignácio Wirth, Universidade Federal do Rio Grande do Sul
Doutor pela Universität Dortmund, Alemanha

Coordenador do PPGEE: _______________________________

Prof. Dr. Marcelo Soares Lubaszewski

Porto Alegre, Outubro de 2007.

DEDICATORY

Dedicated to my parents Érico and Isolda for everything. To my wife Carolina for being her. To my sister Ingrid for the friendship and, of course, for Lucca. To my grand-mother Ezelinda for her prayers. They say we can not choose our relatives: I guess I am the luckiest person in the world for those I have. Specially dedicated to the memory of my cousin Caco, who introduced me to the electronic when we were just kids. I do miss those times and him.

ACKNOWLEDGEMENT

I would like to acknowledge the PPGEE-UFRGS – *Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul*, which supported the development of this work and provided the necessary infrastructure, including auxiliary financial support. Special thanks to Miriam Rosek, secretary of PPGEE.

I do acknowledge CAPES - *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*, which provided scholarship for the development of this thesis both in Brazil and during my excellent and very fruitful stay in France.

Thanks to the professors of the board of examiners for having accepted our invitation and to the professors of the PPGEE-UFRGS and PPGC-UFRGS – *Programa de Pós-Graduação em Computação da UFRGS*, for sharing their knowledge.

Special thanks to Maria Augusta Nunes Neto (Guta), Daniel and Renata Mesquita and Julien Dalmasso for their help in making our arrival and stay in France even better.

Three laboratories were involved during the development of this thesis: LaProT – *Laboratório de Prototipação e Testes*, LaPSI – *Laboratório de Processamento de Sinais e Imagens,* both at the Electrical Engineering Department at UFRGS, and LIRMM – *Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier*, which belongs to the University of Montpellier II, in France. I have made very good friends in all of them, who helped in this work directly or indirectly. Sorry if a miss someone: Adão, Adriane, Álisson, Borin, Balen, Carla, Hermes, Giancarlo, Josias, Letícia, Marcos, Negreiros and Rhod. I would like to stand out two guys who gave a very important contribution for this work: Daniel Scain Farenzena and Marcelo Ienczczak Erigson.

Thanks to Pascal Nouet, my friend and co-advisor in France, for his valuable suggestions. To Luigi Carro, my advisor, my very special acknowledgement. We have done a lot of work during these seven years, since I was a scholarship holder. Thank you for trusting me, for guiding me during all these years of study and for your very good friendship.

Finally, I do must acknowledge my family for all the support I had. Although spread in this big country, they have always helped me whenever necessary. To my family in Rio de Janeiro, Porto Alegre and Veranópolis, my best regards, and to my wife and her family, I just could not ask for more.

RESUMO

Existe um consenso de que os transistores CMOS irão em breve ultrapassar a barreira nanométrica, permitindo a inclusão de um enorme número desses componentes em uma simples pastilha de silício, mais ainda do que a grande densidade de integração vista atualmente. Entretanto, também tem sido afirmado que este desenvolvimento da tecnologia trará juntamente conseqüências indesejáveis em termos de confiabilidade. Neste trabalho, três aspectos da evolução tecnológica serão enfatizados: redução do tamanho dos transistores, aumento da freqüência de relógio e variabilidade de componentes analógicos. O primeiro aspecto diz respeito à ocorrência de *Single Event Upsets* (SEU), uma vez que a carga armazenada nos nós dos circuitos é cada vez menor, tornando o circuito mais suscetível a esses tipos de eventos, principalmente devido à incidência de radiação. O segundo aspecto é também relacionado ao choque de partículas radioativas no circuito. Neste caso, dado que o período de relógio tem se tornado menor, os *Single Event Transients* (SET) podem ser capturados por um *latch*, e interpretado como uma inversão de estado em um determinado bit. Finalmente, o terceiro aspecto lida com a variabilidade de componentes analógicos, a qual tende a aumentar a distância entre o projeto e o teste analógico e o digital. Pensando nesses três problemas, foram propostas três diferentes soluções para lidar com eles. Para o problema do SEU, um novo paradigma foi proposto: ao invés do uso de redundância de hardware ou software, um esquema de redundância de sinal foi proposto através de uso de sinais modulados em sigma-delta. No caso do SET, foi proposta uma solução para o esquema de *Triple Modular Redundancy* (TMR), onde o votador digital é substituído por um analógico, reduzindo assim as chances de ocorrência de SET. Para concluir, para a variabilidade de componentes analógicos, foi proposto um filtro de sinal misto no qual os componentes analógicos críticos são substituídos por partes digitais, permitindo um esquema de teste completamente digital, uma fácil substituição de partes defeituosas e um aumento de produtividade.

Palavras-chaves: SEU. SET. Variabilidade de Componentes. Confiabilidade. Tolerância a Falhas. Produtividade.

ABSTRACT

It has been a consensus that CMOS transistor gate length will soon overcome the nanometric barrier, allowing the inclusion of a huge number of these devices on a single die, even more than the enormous integration density shown these days. Nevertheless, it has also been claimed that this technology development will bring undesirable consequences as well, for what regards reliability. In this work, three aspects of technology evolution will be emphasized: transistor size shrinking, clock frequency increase and analog components variability. The first aspect concerns the occurrence of Single Event Upsets (SEU), since the charge stored in the circuit nodes becomes ever smaller, making the circuit more susceptible to this kind of events, mainly due to radiation incidence. The second aspect is also related to the hit of radiation particles in the circuit. In this case, since clock period becomes smaller, Single Event Transients (SET) may cross the entire circuit and can possibly be latched and interpreted as a state inversion of a certain bit. Finally, the third aspect deals with the analog components variability, which tends to increase the gap between the analog and digital design and test. Thinking about these three problems, we have proposed three different solutions to deal with them. To the SEU problem, a new paradigm has been proposed: instead of hardware or software redundancy, a signal redundancy approach has been proposed through the use of sigma-delta modulated signals. In the SET case, we have proposed a solution for the Triple Modular Redundancy (TMR) approach, where the digital voter is substituted by an analog one, thus reducing the chances of SET occurrence. To conclude, for the analog components variability, we have proposed a mixed-signal filter solution where critical analog components are substituted by digital parts, allowing a complete digital test approach, an easy faulty parts replacement and yield increase.

Keywords: SEU. SET. Components Variability. Reliability. Fault Tolerance. Yield.

TABLE OF CONTENTS

1 INTRODUCTION 16
2 SIGMA-DELTA MODULATION TO COPE WITH SEU IN DIGITAL CIRCUITS 19
2.1 ERROR TOLERANT SYSTEMS 23
2.2 SIGNAL REDUNDANCY PARADIGM 25
2.3 SIGMA-DELTA MODULATED-SIGNALS AND ITS USE IN REDUNDANT SIGNAL GENERATION 28
2.3.1 Sigma-Delta Modulation 29
2.3.2 Redundant Signal Generation through Sigma-Delta Modulation 34
2.3.2.1 Evaluating bit-flips in digital first-order ΣΔ modulators [SCHÜLER, 2006a, 2006e] 36
2.3.2.2 First-Order *versus* High-Order Topologies under Faults [SCHÜLER, 2006a] 42
2.3.3 Case Studies 46
2.3.3.1 Arithmetic Operations [SCHÜLER, 2005a] 46
2.3.3.2 Finite Impulse Response (FIR) Filter [SCHÜLER, 2005b, 2006b, 2006c, 2007a] 57
2.3.3.3 Infinite Impulse Response (IIR) Filter [SCHÜLER, 2007a] 64
2.3.4 Functionally Fault Tolerant DSP Microprocessor [SCHÜLER, 2006d, 2007a] 66
2.3.5 Area and Performance Comparison [SCHÜLER, 2005c] 74
2.3.5.1 Area Analysis 75
2.3.5.2 Performance Analysis 76
2.3.6 0.35µm Technology Chip Development 77
2.4 CONCLUSIONS 80
3 ANALOG VOTER TO COPE WITH SET IN TMR APPROACHES 82
3.1 DIGITAL VOTER 84
3.1.1 Simulations Results 86
3.2 ANALOG VOTER [SCHÜLER, 2005D] [LISBÔA, 2005] 88
3.2.1 Simulations Results 89
3.3 CONCLUSIONS 93
4 STATISTICAL SAMPLERS TO COPE WITH COMPONENT VARIABILITY IN ANALOG CIRCUITS 94
4.1 MIXED-SIGNAL CAPACITANCE-INSENSITIVE FILTER [SCHÜLER, 2007B, 2007C] 94
4.1.1 Input Analog Shift-Register 98
4.1.1.1 Loss in Resolution due to the Shift Step 99
4.1.1.2 Loss in Resolution due to Capacitance Variation of the Analog Shift-Register 99
4.1.1.3 Frequency Response of the Analog Shift-Register 100
4.1.2 Statistical Sampler 102
4.1.2.1 Pseudorandom Noise Generation 104
4.1.3 Adding Spare Parts: testing and replacement scheme 105
4.1.3.1 Testing the Gain of the Analog Shift-Register Buffer 107
4.1.3.2 Testing the Off-Set Voltage of the Statistical Sampler 108
4.2 CONCLUSIONS 109
5 FINAL REMARKS 111
5.1 CONTRIBUTIONS 112
5.2 FUTURE WORKS 115

LIST OF FIGURES

Figure 1: Multiple faults can occur when technology advances [ROSSI, 2005]. 20
Figure 2: Error tolerant signal generation. In (a), resolution is compatible with system requirements, while in (b) faults make resolution drop below 10 bits. 24
Figure 3: Redundant signal generation through signal/noise comparison. In (a) an n-bit word stream is generated, while in (b) an 1-bit bit stream is achieved. .. 26
Figure 4: Comparison between two different ways to generate redundant signals: through noise/signal comparison and through sigma-delta modulation. 28
Figure 5: (a) Analog to Digital conversion and (b) Digital to Analog conversion with sigma-delta modulators. 29
Figure 6: (a) First-order sigma-delta modulator implementation and (b) quantization noise-model. 29
Figure 7: (a) Signal Transfer Function (STF) and Noise Transfer Function (NTF) for a sigma-delta modulator: low-pass and high-pass to achieve high SNR. (b) Increase of the SNR according to the OSR and the modulator order. 31
Figure 8: Replicating first-order modulators to generate high-order ones. 32
Figure 9: (a) Example of hardware implementation for a digital modulator and (b) example of software implementation of the same modulator. 33
Figure 10: (a) Hardware example to implement a digital Sinc filter, whose impulse response is shown in (b). 34
Figure 11: Signal redundancy to be used in error tolerant systems: sigma-delta modulation generates the redundant signal. 35
Figure 12: 9-bit and sigma-delta representation of a sequence of 256 values. Faults are injected through single or multiple bit-flips. 36
Figure 13: Modeling a bit-flip in a certain sample x[i]. 37
Figure 14: (a) Circuit and model for an integrator with a bit-flip and (b) input and output signal for the integrator circuit and model. 38
Figure 15: Complete analytical faulty model for the first order ΣΔ modulator. 39
Figure 16: Simulation results for the model when faults are injected (a) in the input adder and (b) in the integrator. 40
Figure 17: (a) Integrator output with one bit-flip in the LSB and (b) its consequence for the circuit response. In (c), integrator output with MSB inverted and the frequency response in (d). 41
Figure 18: A second-order digital sigma-delta modulator. 42
Figure 19: (a) Second-order modulator response, (b) integrator output and (c) ramp function after fault integration for faults being injected in the second block. (d) response, (e) integrator output and (f) ramp function after fault integration for faults being injected in the first block. 44

Figure 20: Delta-sigma input/output relation for different number of faults injected and different OSR: (a) first-order and OSR=32; (b) second-order and OSR=32; (c) first-order and OSR=64; (d) second-order and OSR=64. 45
Figure 21: Continuous and discrete sigma-delta modulation. 47
Figure 22: Spectra of the continuous and discrete sigma-delta modulation modes. .. 47
Figure 23: Three methods to add bit streams: (a) direct addition, (b) carry feedback and (c) interleaving. 49
Figure 24: Comparing the three methods to add bit streams: (a) direct addition, (b) carry feedback and (c) interleaving. 50
Figure 25: Comparing the three methods to add bit streams when faults are injected in the bit streams: (a) direct addition, (b) carry feedback and (c) interleaving. 51
Figure 26: Addition through interleaving of two sigma-delta modulated bit streams, with faults injected in the addition process. As the OSR increases, also the final resolution and the fault tolerance increase as well. No fault is injected during modulation process. 52
Figure 27: Multiplication of two bit streams through the AND of the bit streams: (a) Input signal 1 in ΣΔ, (b) input signal 2 in ΣΔ and (c) multiplied bit streams. 54
Figure 28: Bit stream multiplication using XNOR gates. 55
Figure 29: Multiplication of two constants using sigma-delta modulated values with different OSR: (a) OSR=32, (b) OSR=64, (c) OSR=128 and (d) OSR=256. 56
Figure 30: Base 2 in the power of exponent varying in the range [-127 127] modulated with different OSR: (a) OSR=32, (b) OSR=64, (c) OSR=128 and (d) OSR=256. 57
Figure 31: Direct Form FIR filter implementation. 58
Figure 32: FIR filter using ΣΔ modulated signals only. Input signal is modulated through continuous modulation. 59
Figure 33: Resolution measured in SFDR versus OSR used to modulate FIR coefficients, for a constant number of faults injected in the coefficients bit streams. 59
Figure 34: Filter response, coefficients and output FFT for different number of faults injected and different OSR used to modulate each filter coefficient. 61
Figure 35: Representation of the structure used to filter signals using a discrete sigma-delta-modulation scheme. 62
Figure 36: Results obtained by filtering a noisy 20KHz signal using sigma-delta modulated signals. In (a), the input signal, in (b), the filtered signal with no faults @OSR 16, in (c) with 4 faults @OSR 16 and in (d) with 4 faults @OSR 32. 63
Figure 37: (a) Infinite Impulse Response filter classical digital implementation, and (b) structure using sigma-delta modulated input signal and coefficients for the same filter. 64
Figure 38: Finite Impulse Response filter coefficients (a) and filter response (b), showing the effects of coefficient quantization. 65
Figure 39: Infinite Impulse Response filter coefficients (a) and filter response (b), showing the effects of coefficient quantization. 66
Figure 40: Block diagram of ADSP2100 DSP microprocessor. 67
Figure 41: Sigma-Delta DSP microprocessor structure. 68

Figure 42: Practical results from the ΣΔ-DSP implementing a 16 taps FIR filter with different OSR used to modulate the coefficients, and respective filter impulse response. 70
Figure 43: n-bit PCM words DSP microprocessor structure. 71
Figure 44: Standard DSP microprocessor simulation results. In (a), fault-free and faulty filter impulse response; (b) filter coefficients, and in (c), FIR output from the fault-free and faulty DSP response. 73
Figure 45: Comparison of area versus number of bit between FIR filter using code-modulated and sigma-delta modulated signals. 76
Figure 46: Finite State Machine for (a) the code-modulated FIR, (b) continuous sigma-delta modulation FIR and (c) discrete sigma-delta modulation FIR. 77
Figure 47: Proposed circuit to be developed using a 0.35μm technology. 78
Figure 48: Top cell view generated through Encounter®. 79
Figure 49: Final mask to be used in the fabrication process. 79
Figure 50: Peak transient current and transient width as a function of strike LET and technology scaling for bulk technology. 82
Figure 51: The TMR approach: voter may become an unreliable point. 83
Figure 52: (a) NAND-NAND voter, (b) NOR-NOR voter, (c) AND-OR and (d) conventional CMOS static implementation of a voter for a TMR system. 84
Figure 53: NMOS section showing a particle hit in a reversed biased PN junction. 85
Figure 54: (a) Proposed model for the double exponential and (b) comparison between the model and the double exponential curve. 86
Figure 55: Simulated circuits for SET evaluation in a digital TMR voter. 87
Figure 56: (a) Analog voter and (b) equivalent electrical circuit. 89
Figure 57: Calculating the short circuited inverters output voltage for all possible input situations. 89
Figure 58: Simulated circuits for SET evaluation in an analog TMR voter. 90
Figure 59: The analog voter being used in a n-MR scheme. 92
Figure 60: Mixed-signal implementation of a FIR filter. 96
Figure 61: Frequency spectra to compare the responses of (a) a normal filter, (b) the proposed filter with an OSR of 256 and (c) an OSR equal to 1024. 97
Figure 62: Noise level plotted by making the difference between a FFT of a normal filter and the proposed one using (a) an OSR equal to 256, (b) 512 and (c) 1024. 98
Figure 63: Realization of an analog shift-register. 99
Figure 64: Loss due to the shift process in the analog shift-register. 99
Figure 65: 50% Monte-Carlo variation in the 2nd capacitor value of the analog delay block. 100
Figure 66: Analog register and its corresponding model. 100
Figure 67: Frequency limit of the input block: (a) due to the RC circuit and (b) due to the operational amplifier. 102
Figure 68: Statistical sampler: a comparator with a noise in the reference [SOUZA, 2004, 2005]. 102
Figure 69: Statistic acquisition with noise dithering [SOUZA, 2004, 2005]. 103
Figure 70: Distribution of a uniform noise without non-linearity generation (a) and distribution of a Gaussian noise with the respective non-linearity expressed by the harmonic (b). 103
Figure 71: Pseudo white noise generation proposed in [FLORES, 2002]. 104
Figure 72: Adding extra taps to the filter: these spare parts can easily replace non-working analog branches. 106

Figure 73: Second tap is replaced by one the existing replacement taps. 106
Figure 74: Sequence for testing the analog shift-register buffer gain. Decision is taken by comparing counted values: CASE values ARE “similar” gain is OK; CASE values ARE “different” gain is not OK. 107
Figure 75: Gain variation versus difference between counted values during the test of the analog shift-register buffer gain. 107
Figure 76: Sequence for testing the comparators off-set voltage. Decision is taken by comparing counted value: CASE value IS “small”, off-set voltage is OK; CASE value IS “large”, off-set voltage is not OK. 108
Figure 77: Off-set voltage variation versus difference counted value during the test of the analog comparator off-set voltage. 108

LIST OF TABLES

Table 1: Summarizing the bit stream addition methods. 50
Table 2: Sigma-Delta DSP instruction set. 68
Table 3: n-bit PCM DSP instruction set. 71
Table 4: Sigma-delta and standard DSP microprocessor comparison summary 74
Table 5: Number of gates *per* bit in each circuit and number of MOS transistors in each gate. 75
Table 6: SET consequences in different nodes of digital voters. 88
Table 7: SET consequences in different nodes of analog voters. 91
Table 8: Summary of simulations with the digital and the analog voters under SET occurrences. 91
Table 9: SET consequences in different nodes of analog voters for a n-MR approach. 92

LIST OF ABBREVIATIONS AND ACRONYMS

ADC: Analog to Digital Converter

ALU: Arithmetic Logic Unit

CMOS: Complementary Metal-Oxide Semiconductor

DAC: Digital to Analog Converter

DSP: Digital Signal Processing

FFT: Fast Fourier Transformer

FIR: Finite Impulse Response

FPGA: Field Programmable Gate Array

Gm-C: Transconductor-Capacitor

IIR: Infinite Impulse Response

LET: Linear Energy Transfer

LSB: Least Significative Bit

LSFR: Linear Feedback Shift Register

MAC: Multiply And Accumulate

MASH: Multi-stAge noise SHaping

MOS: Metal-Oxide Semiconductor

MOSFET: Metal-Oxide Semiconductor Field-Effect Transistor

MSB: Most Significative Bit

NTF: Noise Transfer Function

OSR: Over Sampling Ratio

PCM: Pulse Code Modulation

PDF: Probability Distribution Function

RAM: Random Access Memory

RC: Resistor-Capacitor

SC: Switched-Capacitor

SDDSP: Sigma-Delta Digital Signal Processing

SEL: Single Event Latch-Up

SET: Single Event Transient

SEU: Single Event Upset

SFDR: Spurious Free Dynamic Range

SNR: Signal-to-Noise Ratio

SoC: System-on-Chip

SOI: Silicon-on-Insulator

SRAM: Static Random Access Memory

STF: Signal Transfer Function

TMR: Triple Modular Redundancy

VHDL: Very-High-Speed Integrated Circuit Hardware Description Language

VLSI: Very-Large-Scale Integration

ΣΔ: Sigma-Delta

1 INTRODUCTION

"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year ... Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer" [MOORE, 1965].

With this empirical observation made in 1965, Doctor Gordon Earle Moore dictated one of the most famous laws of the electronic history. Although today warm debates discuss whether this law will continue to be followed or not, it is a consensus that the MOSFET gate length will soon break the nanometric barrier. Much before that, the number of transistor per chip is expected to exceed the one billion cipher [ITRS, 2006]. On the other hand, there is a branch in the semiconductor industry, which believes that the actual silicon-based CMOS technology will be able to attend the miniaturization needs by the next one or two decades. Beyond this period, different approaches in terms of material and architecture must be adopted. The single electron transistor [HADLEY, 1997], resonant tunneling transistor [CHEN, 1996], carbon nanotubes transistor [HAZEGHI, 2007] and spin-transistor [LENT, 1997] are some of the possible alternatives to silicon devices. However, there is a consensus in one point: regardless of the technology that will be used, this new technology must be able to efficiently deal with signal integrity. As the size of the transistor channel decreases, also the number of electrons (or holes) in the channel decreases. With a lower number of carriers passing through the transistor channel, although the possibility of these carriers be hit by an external particle (e.g. alpha or neutron particles) decreases, if such event occurs, the effect caused will be much higher than if there was a higher number of carriers. Another problem is related to the fact that also the power supply of the circuits has decreased. The critical charge,

that is, the amount of charge required to cause an inversion on the transistor state, depends on the circuit node capacitance in a direct proportion, and on the inverse proportion of the node voltage. Since both these variables are decreasing, also the critical charge is smaller. These two factors contribute to the sprouting of the soft faults, that is, faults induced by the incidence of an external particle or by electromagnetic noise, for example, that may have catastrophic consequences to part of a system or, in the worst case, to the whole system.

In this work we are mainly concerned about two kinds of soft fault: the Single Event Upset (SEU) and the Single Event Transient (SET). When a single particle strikes an integrated circuit element, it loses its energy via the production of electron-hole pairs, resulting in a dense ionized track in the local region. This ionization causes a transient current pulse, which, when propagated through a combinatorial logic, is named SET. The SET can possibly be stored in a flip-flop, thus generating an error in one or more bits, what is named SEU. Another SEU possibility is when a change of state is caused by a high-energy particle direct strike to a sensitive memory node. That is, an OFF transistor may become ON after a SEU occurrence. As mentioned, since the node capacitances are becoming smaller, the necessary energy to change the state is also smaller.

Another important factor to be taken into account for future technologies is the increase of the gap between the digital and the analog design. The scaling down of CMOS technology leads to great advantages in digital circuits, since low power and faster circuits can be implemented with an increased integration density. Also, design automation and test are relatively mature for medium density and state-of-the-art digital circuits. In the opposite scenario, analog circuits do not take advantage of Moore's law. On the one hand, they are necessary in most of the System-on-Chip (SoC) devices, and analog circuit size is not reduced in the same rate as digital circuits are. On the other hand, scaling down can introduce some problems. The process variability, which affects not only transistors, but also passive

components in a much higher significant amount. Also, testing analog circuits is not that easy as it is for their digital counterparts, since the signals are defined in the whole range of voltage from ground to +/-Vdd, and the output is often embedded in the SoC, thus reducing the observability. One of the main analog circuits that suffer from the scaling problem, specifically the parameter variation problem, is the filter. For analog filters, whose cutoff frequency, gain and quality factor must be, in many cases, extremely accurate, small deviations in capacitors, resistors and/or inductors values may lead to a complete mismatch between the expected and achieved cutoff frequency, for example. Also, parameters like maximum allowed ripple in the pass-band, minimum attenuation in the stop-band and others must be taken into account when designing these systems. Even if one carefully designs a capacitor to obtain a certain value, the fabrication process can not guarantee an exact replication of this capacitor all through the entire circuit, leading, very often, to the increase of expensive trimming circuits, in order to tune the filter response. Moreover, yield becomes a problem due to the same replication problem, that is, it is hard to obtain the same accuracy for all capacitors in a certain filter production lot. Back to manufacturing test, it is then necessary to identify from the specifications those circuits where parametric errors may lead to a single specification out of the 6σ range. Offsets, characteristic frequencies, quality factors and many other specifications parameters are considered.

This thesis is organized as it follows: in chapter 2 and 3, respectively, two new solutions are presented to the SEU and SET problems, where more details about SEU and SET occurrence, consequences to the circuit and related solutions will also be explained. In chapter 4 a solution for the mismatch problem is depicted, which will also contribute to a whole digital test approach, thus leading to a yield increase. Each chapter presents its own introduction and conclusion, while chapter 5 presents the final remarks and the main contributions from this work, as much as the future works.

2 SIGMA-DELTA MODULATION TO COPE WITH SEU IN DIGITAL CIRCUITS

It has been a consensus that CMOS transistor gate length will overcome the nanometric barrier, allowing the inclusion of a huge number of these devices on a single die, even more than the enormous integration density shown these days [ITRS, 2006]. Nevertheless, it has also been claimed that this integration phenomenon will bring undesirable consequences as well. One of the most critical ones is the reduction in the circuit node capacitances which, in spite of allowing faster circuits with clock speed reaching many dozens of gigahertz, will also be responsible by the increase in the soft-errors occurrence.

One of the main consequences of soft-errors is the Single Event Upset (SEU), caused by a particle hitting a CMOS junction, which can result in a bit flip that can be propagated all through the rest of the circuit operation [YANG, 1992], [MESSENGER, 1992]. Most of the reported SEU effects concern the manifestation of single faults in digital circuits, caused by the inversion of one single bit. However, multiple bits inversion has also been demonstrated in some works [SMITH, 1992], indicating that this is an expected scenario for future technologies. One can observe in figure 1 [ROSSI, 2005] that, for a given CMOS technology, a particle hit will affect one transistor. Now, if a newer technology is used, more transistors can be constructed in the same area occupied by one transistor in the older technology. As a consequence, the same particle hit can now affect more than one transistor, causing multiple SEUs.

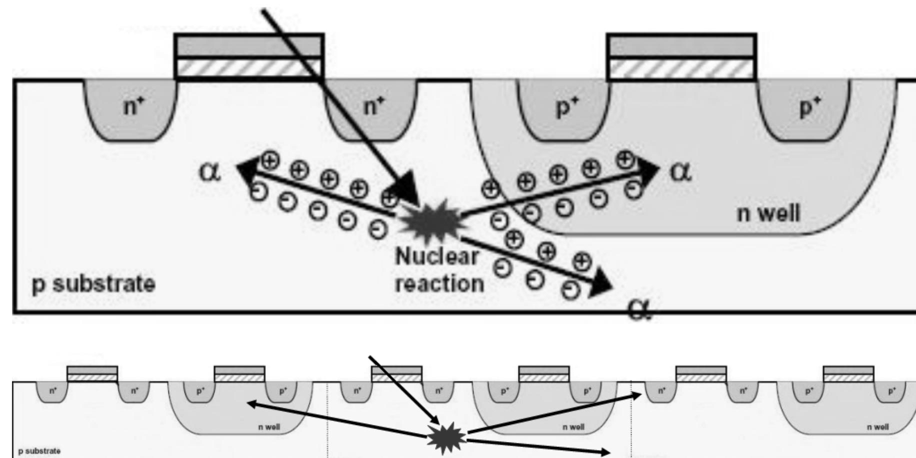


Figure 1: Multiple faults can occur when technology advances [ROSSI, 2005].

Most techniques used to develop robust circuits use different solutions, which can be divided basically in three different categories, described bellow, all of them actuating in different abstraction levels: technology, design and system level.

In the technology level, different processes are used in the transistor fabrication, such as epitaxial-bulk CMOS, which was first conceived to isolate the analog part from the digital one in mixed-signals designs. In this technique, an extra mask is used to etch a trench from the backside of the wafer all the way to the under-surface of the field oxide [BASEDAU, 1995]. Although the use of epitaxial-bulk is efficient to reduce the Single Event Latch-Up (SEL), it does not mitigate the occurrence of SEU. Another technique is the use of Silicon on Insulator (SOI), where a thin layer of silicon is placed on top of an insulator, such as silicon oxide or glass, and then the transistor is built on top of this layer [IBM, 2005]. This technique was first developed to be used in memories for space applications, since these memories built on SOI were perceived to be more resistant to SEU. Further studies showed that in order to reduce power consumption and to increase speed in digital circuits, the use of SOI could be a good alternative. This technique, however, requires the use of special fabrication process, with consequent yield limitation, and does not completely mitigate the occurrence of SEU.

Hardening at the design level includes, for example, the use of hardened gate resistor memory cells [WEAVER, 1987] and hardened CMOS memory cells with feedback structures

[RABAEY, 1996], which imply in physically larger memory blocks, since extra parts must be added to the memory cell. For example, in the gate resistor memory cells, two resistors are built using two levels of polysilicon and, although the impact in the circuit density is small, these resistors are temperature sensitive, increasing the memory vulnerability in low temperatures. Another solution in the design level is the use of codification and decodification of logic blocks, using, for example, Hamming [MACKAY, 2003] or Reed-Solomon [PLANK, 1996] techniques. Hamming Code is an error-detecting and error-correcting binary code that can detect all single-bit and double-bit errors and correct all single-bit errors. The Reed-Solomon code, however, is able to detect and to correct multiple and consecutive data errors. Although these are good solutions, when multiple faults must be corrected, the cost to do so, in terms of time and complexity, makes their use impractical. Also, multiple and simultaneous faults can not be corrected using these approaches. Moreover, the coder and decoder circuits are also sensitive to transient upsets, reducing the overall reliability, even with the added costs.

To protect digital circuits at the system level, the use of hardware or software redundancy techniques are the most known ones. For the hardware case, the use of Triple Modular Redundancy (TMR) [CHANDE, 1989] rises as the most diffused scheme. TMR in its various implementations simply implies in triplicating the sensitive block and making a vote, where the correct system response is determined by majority. Of course, some problems stand out here. For example, the area and power penalties, which are triplicated as well. The other limitation appears when one thinks in simultaneous faults, that is, if two blocks give wrong responses, by majority, the final response will also be wrong. Also, with less probability to occur, if the fault happens in the voter block, there is no way to define whether the response is correct or not. Using software redundancy, some techniques such as Algorithm Based Fault Tolerance [HUANG, 1984] and Code Flow Check are used, as well as variable

duplication [REBAUDENGO, 1998], which can automatically be implemented on the high-level code of the program. This technique, applied to memories and registers, performs two different modifications to the source code; the first one corresponds to duplicating some or all of the program variables in order to introduce data redundancy, and modifying all the operators to manage the introduced replica of the variables. The second source code modification aims at introducing consistency checks inside the control flow to periodically verify the consistency between the two copies of each variable.

Most of the previously proposed schemes are based on hardware or software redundancy, where the hardware block or the software variable is triplicated, and the correct system response is given by majority voting. Looking at the problem from a different perspective, we now propose a new kind of redundancy, based on signal redundancy.

The main idea does not imply in triplicating the signal to be processed, but in creating another way to represent the signal, in such a way that, even under the occurrence of multiple faults, the final response still sustains an acceptable resolution. The technique, which uses sigma-delta modulation (a review can be found in [NORSWORTHY, 1997]) to generate the redundant signal, is to be used in digital circuits and, as it will be shown, can imply, in some cases, in smaller and faster circuits. In the cases where area and time penalties proposed by other solutions are smaller than the solution herein proposed, the achieved fault-tolerance becomes a decisive factor to determine the use of our approach.

A new perspective is presented, where one does not need to worry whether the fault will occur or not, because even if it occurs, the circuit will be protected. That is, we do not try to mitigate or to correct the circuit response corrupted by the faults. We simply let the fault occur, because thanks to the redundancy already presented in the signal, the consequence of the fault will not be so harmful for the system response. This technique, allied to the idea of

error tolerance proposed in [GUPTA, 2004], can guarantee that the system will develop appropriate responses even under the occurrence of multiple and simultaneous faults.

2.1 Error Tolerant Systems

According to [JOHNSON, 1993], "fault tolerance is the ability of a system to continue correct operation of its tasks after hardware or software faults occur", while correct operation means that no errors occur at any system output. In [GUPTA, 2004], it is mentioned that "fault tolerance tries to provide reliable operation in the presence of lifetime faults and/or externally induced transient errors". This way, it has also been proposed in [GUPTA, 2004] the following definition of error tolerance when considering systems that can tolerate a certain amount of errors at the entire system's output: "a circuit can be error tolerant with respect to an application if it contains defects that cause internal errors and might cause external errors, and the system that incorporates this circuit produces acceptable results".

The very simple results presented in figure 2 can clarify the notion of error tolerant system, and the idea behind minimum resolution required to a certain error tolerant system generating correct output responses. In this example, a digital oscillator must generate a 10-bit minimum resolution sine wave with a certain frequency, offset and phase. This is equivalent to say that the expected Signal to Noise Ratio (SNR) of the output signal must be at least 71dB. Thus, any signal generated with a SNR below this threshold shall not be satisfactory to the system operation. However, if the faults occurring during the signal generation do not cause a perturbation high enough to make the SNR drop below this fixed value, the signal will be properly used by the system. In figure 2(a), different bits, below the sixth one, are inverted each time one point is produced by the oscillator, causing a small perturbation in the signal shape, but still maintaining a SNR higher than 73dB (10.5 bits resolution), thus above the requested 71dB. However, when the bits which are inverted change to those above the sixth

one (see figure 2(b)), the output signal presents a significant variation, causing a 11dB drop in the SNR, leading to a malfunction behavior, since the new resolution now is about 8.5 bits (62dB). So, one can say that, for this very simple circuit, it is error tolerant as long as one can be sure that faults do not occur in bits that are higher than the sixth one.

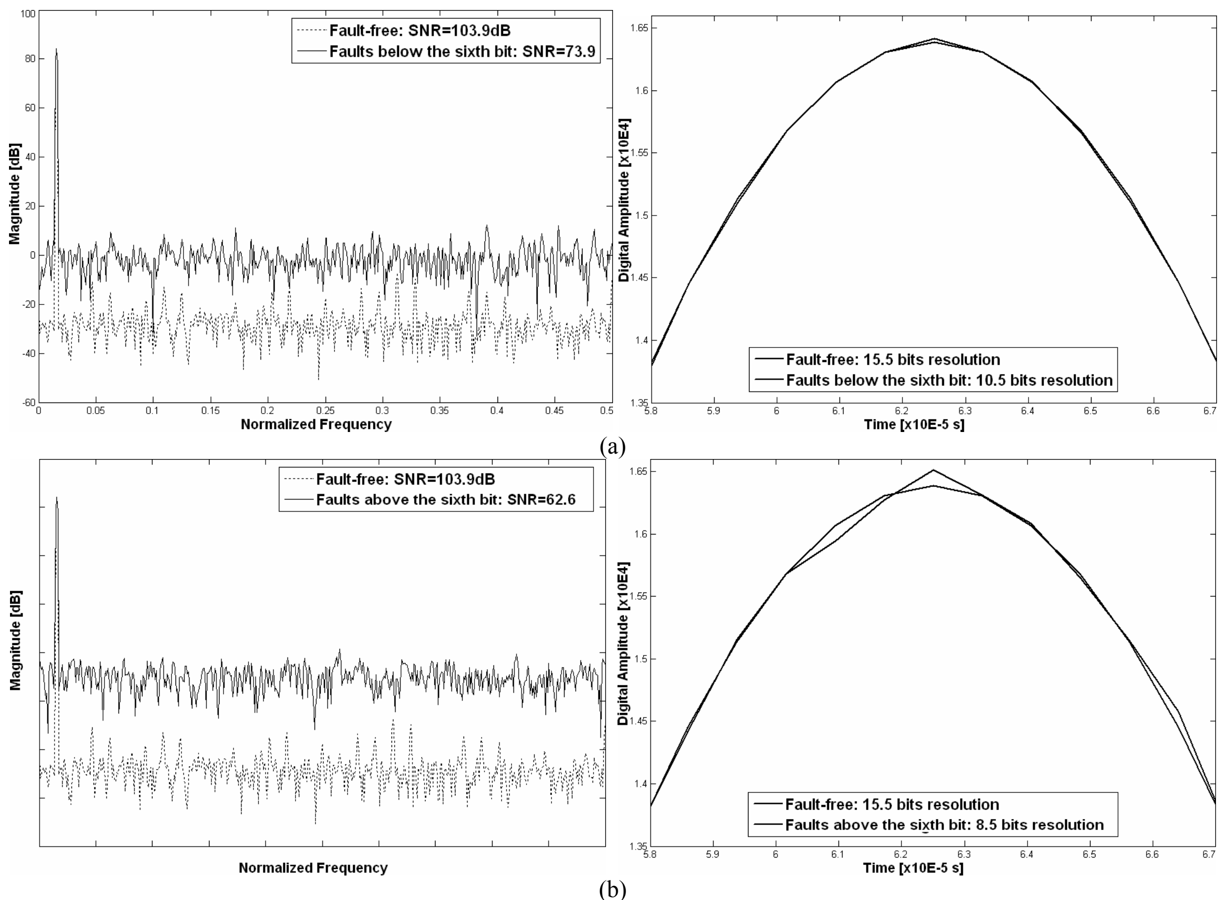


Figure 2: Error tolerant signal generation. In (a), resolution is compatible with system requirements, while in (b) faults make resolution drop below 10 bits.

Another interesting proposition, which also makes use of the error tolerance approach, is the one presented in [NEPAL, 2005], where a probabilistic-based design methodology based on Markov Random Fields is examined. According to [NEPAL, 2005], "the Markov Random Fields approach can express arbitrary logic circuits and the logic operation is achieved by maximizing the probability of correct state configurations in the logic network depending on the interaction of neighboring circuit nodes". The basic idea is that the computation is realized by propagating states through the circuit in a probabilistically fashion, assuming that a large number of nanodevices is presented, thus requiring very low power

operation with consequent probabilistic behavior, since transistor will be operating near the thermal limit. Nevertheless, since the idea presented in [NEPAL, 2005] supposes a huge number of transistors presented in the digital circuit, the solutions make use of many of these transistors to construct a simple logic gate. For example, a simple CMOS inverter, which uses only two transistors, is now developed by using 20 of these devices, while a NAND gate is conceived through the use of 60 MOS transistors, instead of four as the standard one.

With the idea of error tolerance in mind, one can find a lot of applications that contain the property of being error tolerant, that is, applications in which the final response can still be correct, even after the insertion of a large amount of faults, in our case, transient faults caused by radiation or noise interference. Also, as seen before, redundancy is one of the most used techniques to cope with fault occurrence. Based on that, we now propose a new paradigm, which is based on signal redundancy. The idea behind this solution is not based, for example, on triplicating the n-bit words of the input signal, but rather on creating a new version of the input signal, representing it in a redundant fashion through the use of another signal domain representation. The next sections describe the methodology used to create this signal, and give some results that support our decision in using such signal depiction.

2.2 SIGNAL REDUNDANCY PARADIGM

As shown before, the use of redundancy is very common when fault-tolerance must be achieved by a system. Besides the already mentioned TMR in hardware redundancy, and variable duplication in software redundancy, others techniques may be used, such as space and/or time redundancy [ANGHEL, 2000], where the concomitant use of a self-checking combinational circuit and a state-preserving element is employed. Now, based on the idea of redundancy, we propose changing the paradigm, where the redundant element is intrinsic to the data that will be processed. Although this may seem similar to a classical redundant

approach, here we do not duplicate the data, but rather we represent information in another domain, which can still be processed in a digital like manner.

The basic idea is to take a single n-bit Pulse Code Modulated (PCM) word, thus in the digital domain, and change the way this word is represented. This change is such that the n-bit word now becomes an m-bit word, where $m>n$, and it is no more a digital representation of the data, but a redundant representation, which can still be processed by digital circuits.

There are different ways to obtain redundant signals. For example, as represented in figure 3(a), one can simply compare an n-bit word with an n-bit random noise, and the output n-bit words will be a representation of the input value [JANER, 1996]. A simpler solution would be based on the same principle, but the signal comparison now generates a 1-bit bit stream, containing the input signal probabilistic representation [JANER, 1996], as shown in figure 3(b). However, in order to obtain a good resolution to represent the input signal, the output bit stream length must be in the range of thousands of bits, as demonstrate some results using this kind of signal representation in [LISBÔA, 2004].

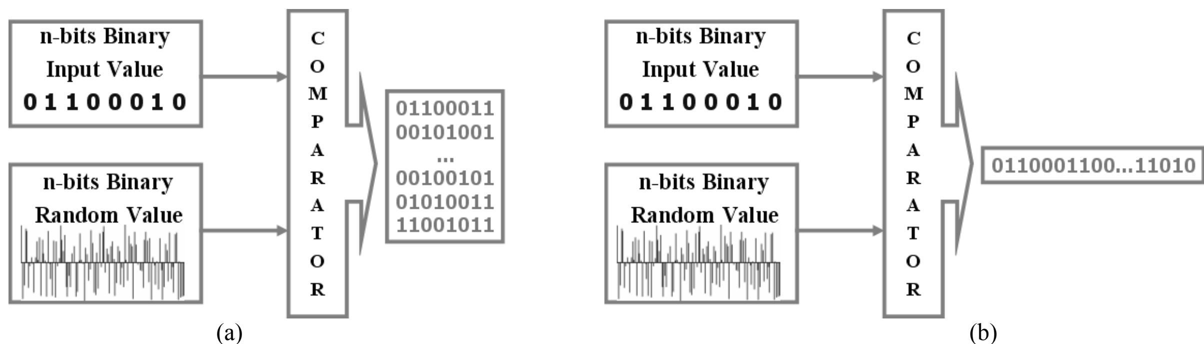


Figure 3: Redundant signal generation through signal/noise comparison. In (a) an n-bit word stream is generated, while in (b) an 1-bit bit stream is achieved.

A better way to produce signals with intrinsic redundancy, capable of representing large bit words, but with a low implementation cost and a much smaller bit stream, is by using sigma-delta modulation [NORSWORTHY, 1997], [CANDY, 1992]. With this kind of signal generation, one can represent signals with a resolution larger than 20 bits, with lots of redundancy by using a very simple scheme. It is important to emphasize that, although sigma-delta modulators are most commonly found in analog-to-digital converters (using analog

sigma-delta modulators), what we are proposing here is the use of digital sigma-delta modulators to transform an n-bit digital word into a sigma-delta representation of the same signal. The modulator output signal, represented by a sequence of zeros and ones, will then be used in the subsequent application. An explanation regarding digital sigma-delta modulators, how they work and which kind of signals they generate is presented in section 2.3.1.

To conclude the idea of signal redundancy and show the advantages of using a sigma-delta representation instead of, for example, the previously mentioned technique of signal/noise comparison, an example comparing these two kinds of signal representation is presented in figure 4. Here, a 10 KHz sine wave is sampled with an Over Sampling Ratio (OSR) equal to 64, that is, it is sampled with a frequency 64 times higher than the signal Nyquist frequency. Firstly, the sampled signal is represented by a bit stream produced through the comparison of each sampled point to a uniformly distributed random noise, as presented before in figure 3(b). On the other hand, the same sampled signal is now modulated through the use of a first-order sigma-delta modulator, generating another bit stream, also representing the input signal. As one can see, for the same OSR, thus for the same output bit stream length (128 bits per signal period in this case), the achieved SNR to the signal comparison case is approximately equal to 19dB, while for the sigma-delta modulation case, the SNR almost reaches 39dB. So, as mentioned and now demonstrated, the use of sigma-delta modulation generates more accurate results with the same number of bits in the bit stream.

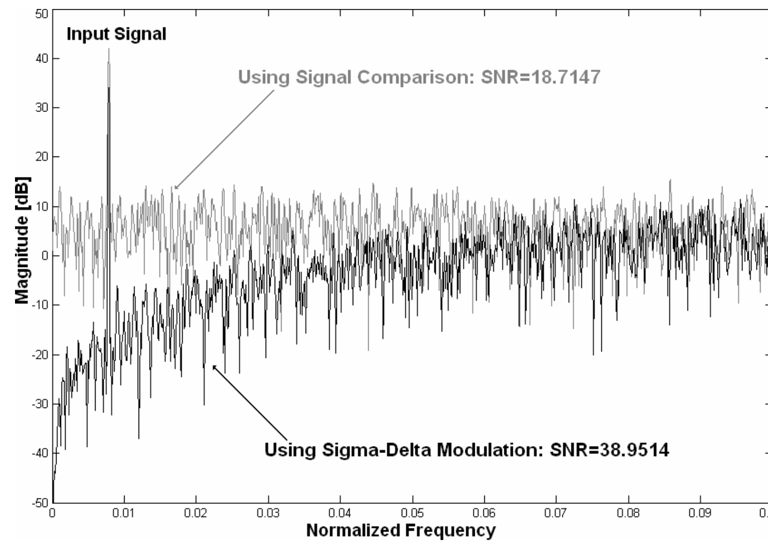


Figure 4: Comparison between two different ways to generate redundant signals: through noise/signal comparison and through sigma-delta modulation.

2.3 Sigma-Delta Modulated-Signals and Its Use in Redundant Signal Generation

The main utilization of sigma-delta modulation is in Analog-to-Digital Converters (ADC), in the manner presented in figure 5(a), where the analog input signal is over-sampled, converted to a 1-bit representation through the analog sigma-delta modulator, and then down-sampled (or decimated) in order to obtain a digital representation of the analog input signal. This kind of signal modulation can produce converted signals with high resolutions, by using small sampling ratios when compared to others over-sampled converters [CANDY, 1992]. Others applications to sigma-delta modulators are in the generation of test-signals [ROBERTS, 1995], signal-processing [MALOBERTI, 1992] [DIAS, DA FONTE, 1994] and, of course, Digital-to-Analog Converters (DAC) [NORSWORTHY, 1997], [CANDY, 1992]. In the DAC case (see figure 5(b)), the digital input signal is over-sampled, converted to the sigma-delta domain through a digital sigma-delta modulator, and finally low-pass filtered to generate the analog signal representation. A brief explanation of how a sigma-delta modulator works and which kinds of signals are generated after the modulation are presented next, in section 2.3.1.

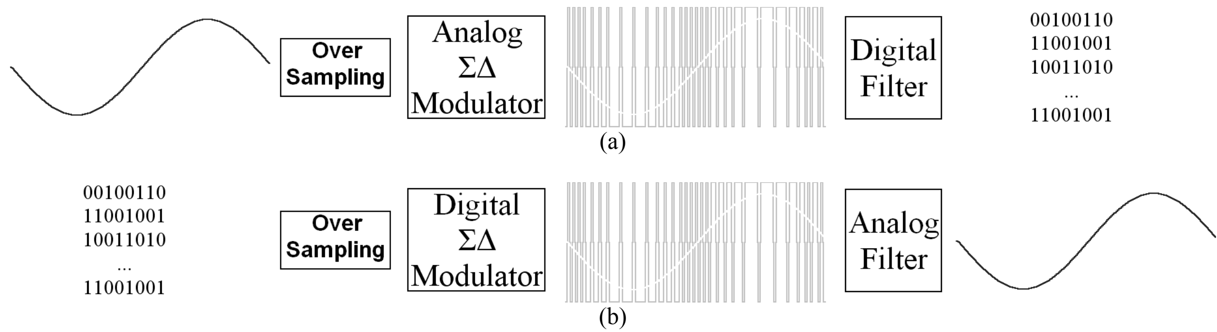


Figure 5: (a) Analog to Digital conversion and (b) Digital to Analog conversion with sigma-delta modulators.

2.3.1 Sigma-Delta Modulation

The sigma-delta modulator is the main block in a sigma-delta ADC or DAC (see figure 5), since this is the block that will pass the over-sampled input signal from the analog/digital to the sigma-delta domain for further filtering through a digital/analog low-pass filter, and consequent conversion to digital/analog representation. Consider the generic first-order sigma-delta modulator represented in figure 6(a). The structure, known as an error feedback structure, consists of four basic blocks: an input subtractor, an integrator, a quantizer and a feedback gain. The basic idea is that the quantization error produced by the two-level quantizer is fed back to the circuit input, and subtracted from the input signal. The difference between the input signal and the fed back quantization error is integrated, and then quantized to generate an output represented by a 1-bit bit stream, whose mean value is equal to the mean value of the input signal [NORSWORTHY, 1997].

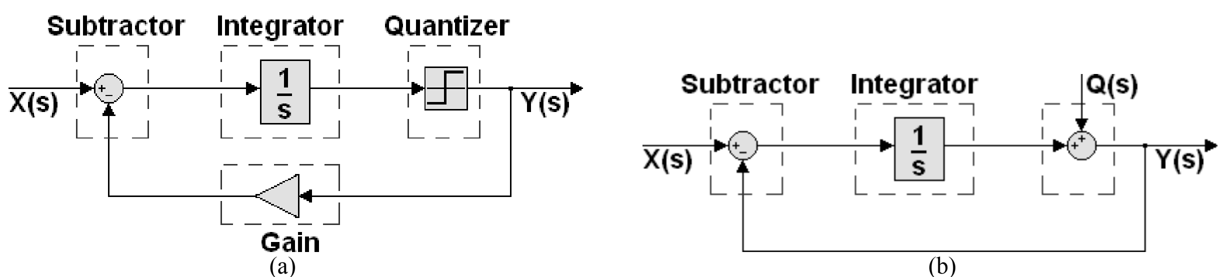


Figure 6: (a) First-order sigma-delta modulator implementation and (b) quantization noise-model.

Analyzing this kind of circuit by the frequency point of view allows one to understand why one can obtain high resolutions using such a simple circuit. To do that, some considerations must be done regarding the modulator characteristics, mainly, the quantization

noise generated by the two-level quantizer. For more details about these considerations, the reader is encouraged to consult references [NORSWORTHY, 1997] and [CANDY, 1992]. At this point, in order to present a brief analysis of how the modulator frequency response behaves, we assume that the modulator presents the following characteristics regarding the quantization error: the quantization error is largely uncorrelated from sample to sample to the input signal, and has equal probability of lying anywhere in the range ±D/2, where D is the quantization level amplitude. This way, the quantization error can be represented by a noise Q(s), as shows the model presented in figure 6(b). Analyzing the transfer function of this model from the signal input X(s) and from the quantization noise input Q(s), one has:

$$\frac{Y(s)}{X(s)} = \frac{1}{s+1} \quad \text{Signal Transfer Function (STF)} \qquad (1)$$

$$\frac{Y(s)}{Q(s)} = \frac{s}{s+1} \quad \text{Noise Transfer Function (NTF)} \qquad (2)$$

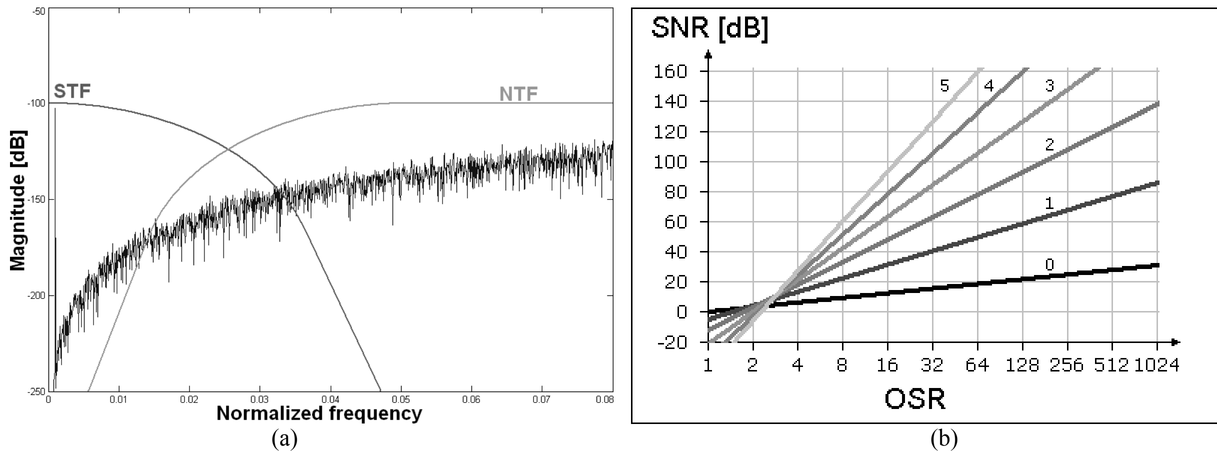
Analyzing equations (1) and (2), one can notice that the signal transfer function (STF) is a low-pass function, while the noise transfer function (NTF) is a high-pass function. As a consequence, the output signal will be represented by the input signal in low-frequencies plus the quantization noise in high-frequencies. Figure 7(a) presents a sinusoidal signal modulated by a sigma-delta modulator with a certain sample frequency. As noted, due to this particular response, one can achieve high signal to noise ratios, which can be increased by increasing the modulator over sapling ratio or the modulator order. It can be shown [NORSWORTHY, 1997] that, for the first-order modulator, the SNR *versus* OSR relation is given by:

$$SNRdb = 10\log\left(\frac{3}{2}2^{2n}OSR^{3}\frac{3}{\pi^{2}}\right) \qquad (3)$$

where *n* is the number of bits of the modulator quantizer. Equation (3) can be written as:

$$SNRdb = 1.76 + 6.02n + 30\log(OSR) - 5.17 \qquad (4)$$

that is, one has an increase of 6dB *per* bit of the quantizer and 9dB for each doubling in the sampling frequency. The increase of the SNR according to the modulator order and the OSR is shown in figure 7(b).



(a) (b)

Figure 7: (a) Signal Transfer Function (STF) and Noise Transfer Function (NTF) for a sigma-delta modulator: low-pass and high-pass to achieve high SNR. (b) Increase of the SNR according to the OSR and the modulator order.

For the higher-order sigma-delta modulators, different topologies are available. The simplest way to implement such systems is shown in figure 8. Just by substituting the modulator quantizer by a first-order modulator, higher-order modulator can be implemented. However, due to instability factor, this recursive method is limited to third-order modulators [NORSWORTHY, 1997]. The instability problem can be overcame through the use of different topologies, such as cascade-modulators (MASH) [CANDY, 1986] [UCHIMURA, 1988], utilization of multi-bit quantizer [LESLIE, 1992], use of feed-forward and feedback coefficients [CHAO, 1990] and others.

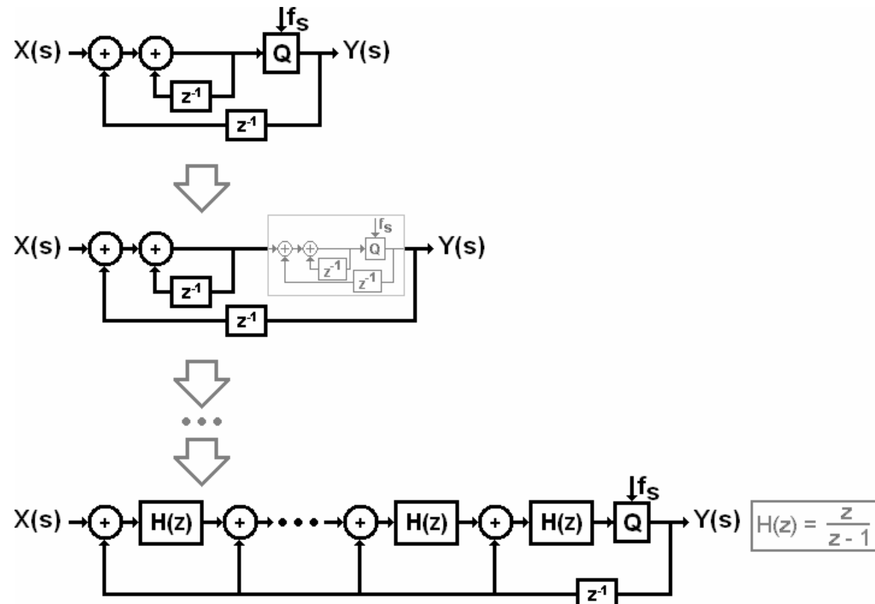


Figure 8: Replicating first-order modulators to generate high-order ones.

It is important to note that, if one desires to implement an analog modulator, thus an analog version of the blocks in figure 6(a) must be constructed, and the most common technique used to do it is a switched-capacitor implementation [BOSER, 1988]. In this case, the feedback gain is a one-level DAC, which simply feeds back a positive or a negative voltage value, depending on the value of the output bit stream. In our case, we want to make a digital modulator because the input signal is digital, and we want it to be converted to a sigma-delta representation. So, the modulator of figure 6(a) must be implemented either in hardware or software. In figure 9(a) an example of how this simple modulator could be implemented in hardware is presented. Here, the quantizer is substituted by simply taking the integrator sign, and this sign will choose whether an addition or a subtraction of the input value and a constant value is done, substituting, thus, the feedback gain, whose function is to convert the output 1-bit signal to an n-bit value compatible with the input signal. A software implementation of this same modulator is presented in figure 9 (b).

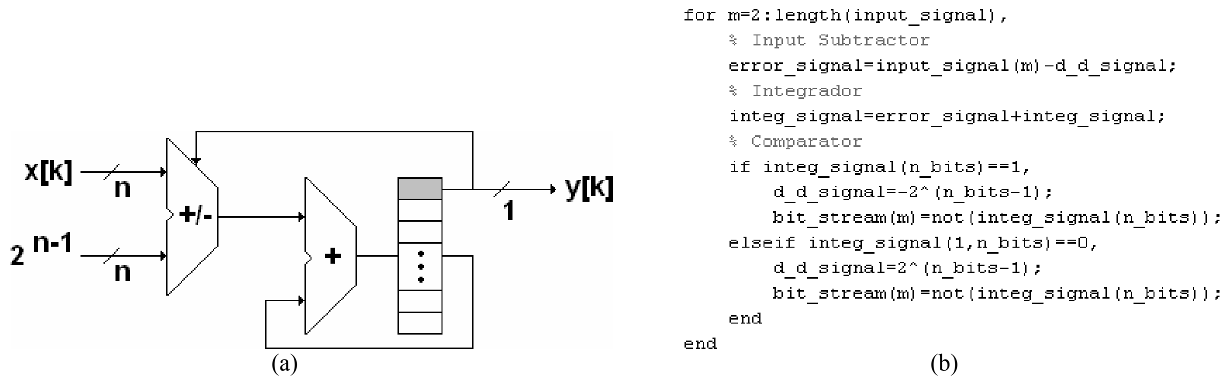


Figure 9: (a) Example of hardware implementation for a digital modulator and (b) example of software implementation of the same modulator.

To conclude the discussion about sigma-delta modulators, a brief explanation about the decimation concept must be given. The decimation process [NORSWORTHY, 1997,] [CANDY, 1992] [CROCHIERE, 1981] is the final step in an analog to digital conversion using sigma-delta modulation, where the sigma-delta bit stream is converted to an n-bit digital representation. In our case, it is used just in order to analyze the obtained values, but in a real application, the idea is that the bit stream will be used all through the process and, if necessary, be decimated only in the final stage. A common way to develop a decimation operation is through the use of a digital *Sinc* filter for example [NORSWORTHY, 1997], whose hardware implementation example is presented in figure 10(a). This filter, also known as accumulate-and-dump, simply averages the output signal, generating a digital representation of the input signal. The way the filter works is outside the scope of this work, but the basic idea is that the over-sampled bit stream is down-sampled to generate its digital representation, in such a way that the filter cut-off frequency is appropriately designed to filter the out-of-band quantization noise, as figure 10(b) demonstrates.

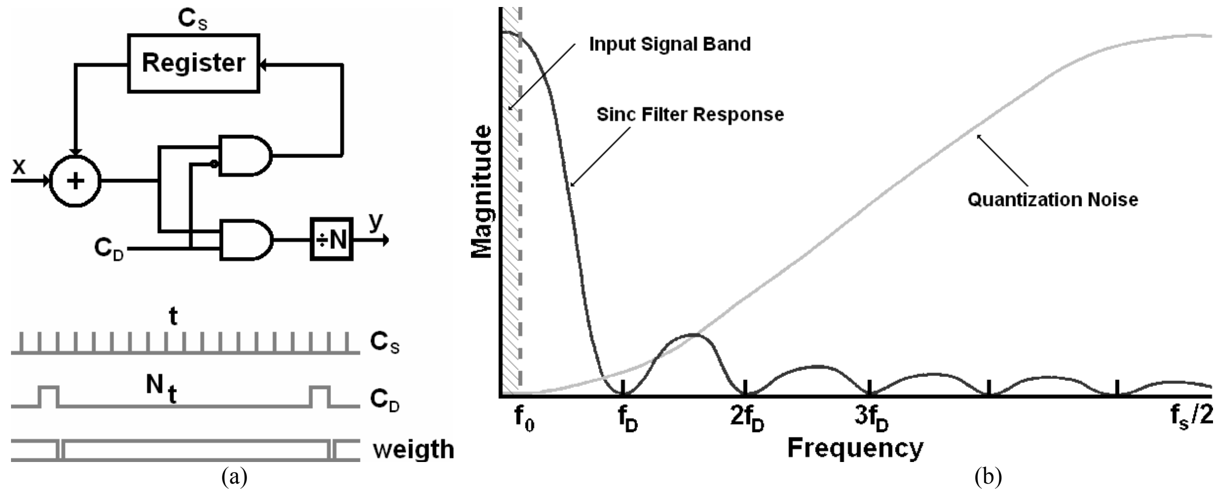


Figure 10: (a) Hardware example to implement a digital Sinc filter, whose impulse response is shown in (b).

In figure 10(a), *t* is the clock period and *N* is defined as the decimation ratio, given by the ratio between the modulation ratio and the intermediate decimation frequency [NORSWORTHY].

2.3.2 Redundant Signal Generation through Sigma-Delta Modulation

The intrinsic redundancy of sigma-delta modulated signals comes from the fact that the bit stream carries the original signal representation plus a certain quantization noise, as showed in section 2.3.1. In another point of view, one can imagine that the original n-bit words are now represented in a sequence of Least Significative Bits (LSB), since the two-level quantizer generates a 1-bit bit stream as the modulated output. So, the inversion of many of these bits shall not interfere so much in the signal resolution.

Figure 11 schematically presents our proposed approach, where sigma-delta signal modulation is used to produce data values containing an amount of redundancy able to tolerate a large number of transient faults. As noted, for an n-bit PCM data representation, if one single bit inversion occurs, the consequences for the final system response may cause a complete mismatch between the expected response and the obtained one. On the other hand, if one considers an error tolerant scheme using a data redundant signal representation, even after

the inversion of many bits (simultaneously or not) the final response will be much closer to the expected one.

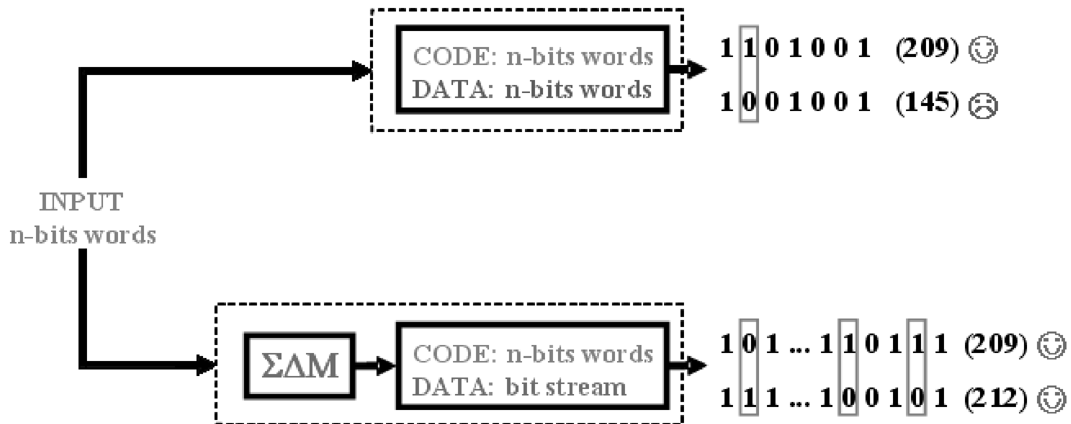


Figure 11: Signal redundancy to be used in error tolerant systems: sigma-delta modulation generates the redundant signal.

An illustrative example is presented in figure 12, where a sequence of 256 values is represented in two different ways: in a common 9-bit two-complement representation, and in a sigma-delta representation. Since for each 9-bit word the sigma-delta modulated representation will be a sequence of LSBs, then 2^8 bits in each bit stream are necessary to exactly represent the original value (the 9th bit is the sign-bit). For each 9-bit coded value, only one fault is injected, that is, only one randomly chosen bit is inverted, while for each sigma-delta representation, 10 randomly chosen bits are inverted, thus representing a much larger number of faults. As noted in figure 12, although under a much larger number of faults, the decimated sigma-delta values are always much closer to the expected value than the coded-words.

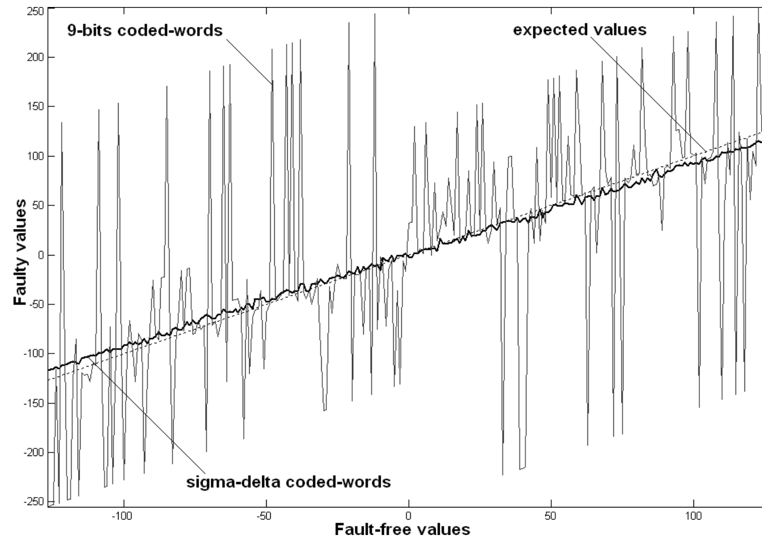


Figure 12: 9-bit and sigma-delta representation of a sequence of 256 values. Faults are injected through single or multiple bit-flips.

Something important to be noted here is the difference between faults that occur in the sigma-delta modulator and faults that occur in the system that is using these modulated signals to generate the final response. Sections 2.3.2.1 and 2.3.2.2 show a mathematical approach to investigate what happens when faults occur inside de sigma-delta modulator, either for a first-order or for higher-order modulators. On the other hand, in section 2.3.3, different case studies are presented in order to evaluate the behavior of sigma-delta modulated-signals under multiple faults. In these cases, no faults are injected in the modulator.

2.3.2.1 Evaluating bit-flips in digital first-order ΣΔ modulators [SCHÜLER, 2006a, 2006e]

Consider a generic digital circuit, which works with sampled and quantized signals represented by n-bit words. If when the quantized n-bit signal is passing through the circuit one of its bit has an inversion, this can be seen as an addition (or subtraction) of a certain value in the current sample, as shows figure 13.

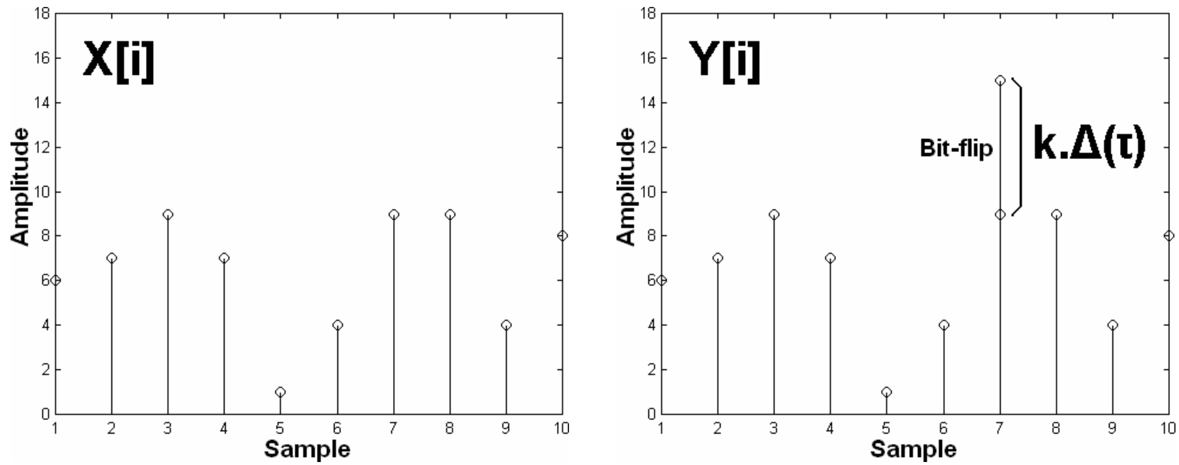


Figure 13: Modeling a bit-flip in a certain sample x[i].

Discrete-time mathematics provides us a description of such fault as a Delta Function, or more commonly, impulse function (δ). Thus, the signal at a selected part of the circuit is modeled as:

$$y[i] = x[i] + k.\delta[i - \tau],\ k \leq 2^{n} \quad (5)$$

In (5), x is the original signal (fault free) and τ is the time when the fault should occur. This model can be applied in every signal of the system, stating a generalized framework for multiple SEU. Transforming (5) into the Z domain, one obtains:

$$Y(z) = X(z) + k.z^{-\tau} \quad (6)$$

The importance of this result is that it is now possible to derive models for components with inserted faults. Such process, for example in a discrete integrator, is displayed in figure 14.

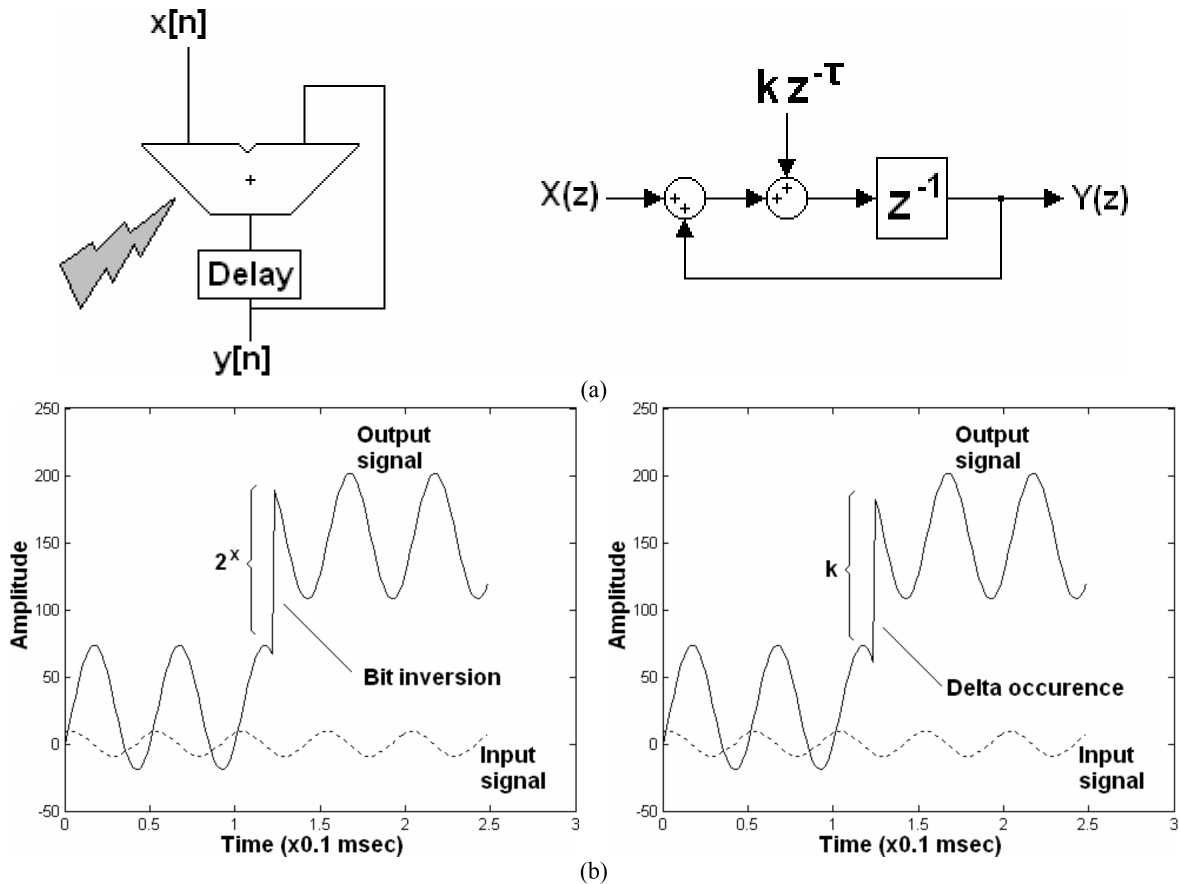


Figure 14: (a) Circuit and model for an integrator with a bit-flip and (b) input and output signal for the integrator circuit and model.

The diagrammatic version of the discrete integrator has an extra term, which represents the fault, summed just before the delay unit. Writing the respective transfer function results in:

$$Y(z) = \frac{kz^{-\tau-1}}{1-z^{-1}} + X(z)\frac{z^{-1}}{1-z^{-1}} \qquad (7)$$

A simple analysis presented in figure 14(b) exhibits the integration process where the fault appears as the addition of a certain value, proportional to the bit inverted by the fault. Applying this fault model to each block of the sigma-delta modulator model in figure 6(b), one generates the analytical model for faults in a digital first-order sigma-delta modulator, as depicted in figure 15, upon which further conclusions will be made.

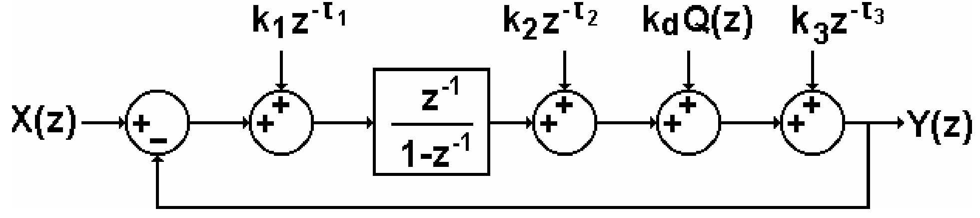


Figure 15: Complete analytical faulty model for the first order ΣΔ modulator.

For this model, k_1 represents faults being inserted in the modulator input adder, k_2 in the modulator integrator and k_3 in the modulator quantizer. The final transfer function of the model is given by:

$$Y(z) = X(z)z^{-1} + k_d Q(z)(1 - z^{-1}) + BF(z) \quad (8)$$

In which the *BF(z)* term represents the parcel corresponding to the fault, and are expressed by:

$$BF(z) = k_1 z^{-\tau_1} z^{-1} + (k_2 z^{-\tau_2} + k_3 z^{-\tau_3})(1 - z^{-1}) \quad (9)$$

Making the inverse z-transform, one obtains the difference equation:

$$y[n] = k_d(q[n] - q[n-1]) + x[n-1] + \sum_{i=1}^{3} bf_i[n] \quad (10)$$

In (10), *bf[n]* represents the faults, expressed by the addition of three terms, where each one represents the faults produced in the input adder, integrator adder and comparator, respectively. These terms are given in equations (11), (12) and (13):

$$bf_1[n] = k_1 \delta[n - \tau_1] \quad (11)$$

$$bf_2[n] = k_2(\delta[n - \tau_2] - \delta[n - \tau_2 - 1]) \quad (12)$$

$$bf_3[n] = k_3(\delta[n - \tau_3] - \delta[n - \tau_3 - 1]) \quad (13)$$

The general expected behavior for a delta-sigma modulator is represented by the two first terms in equation (8), *i. e.*, the quantization noise is translated to higher frequencies while the input signal stays in its original band. As mentioned, the parcel *BF(z)* in (8) describes the faults, represented by impulses translated in time.

In equation (9), it is possible to realize that for faults occurring before the integrator, these will appear inside the signal band. However, when faults occur after or in the integrator, they will be moved outside the signal band. This is an expected behavior for linear circuits with feedback paths and an integrator in the forwarding path [OGATA, 1994].

Simulations of the model extracted from the discrete-time equations (10)-(13), confirm this behavior. Faults in the input adder distort the output bit stream generated by the modulator in the signal band, while faults in the discrete-time integrator, or in the quantizer, have their effects shifted to higher frequencies. Figure 16 shows these simulations results, where the input signal, a sinusoidal wave, was sampled with an OSR of 32 for both runs.

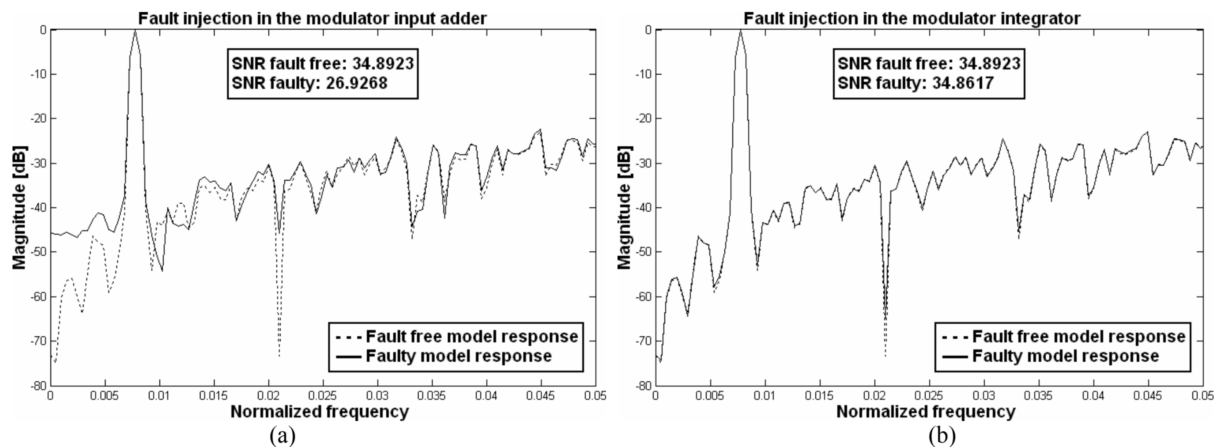


Figure 16: Simulation results for the model when faults are injected (a) in the input adder and (b) in the integrator.

Results reached so far are based on a linear model of the modulator (see figure 6(b)) and form the foundation of our understanding. However, the assumption that the quantizer is modeled by an added noise is not valid anymore when the real circuit behavior must be evaluated.

Since the circuit quantizer is a simple comparator [ROBERTS, 1995], the feedback process no longer acts as the predicted model. Taking the real circuit, thus non-linear, for simulation, the output of the discrete integrator is plotted just before the quantization step, and showed in figure 17.

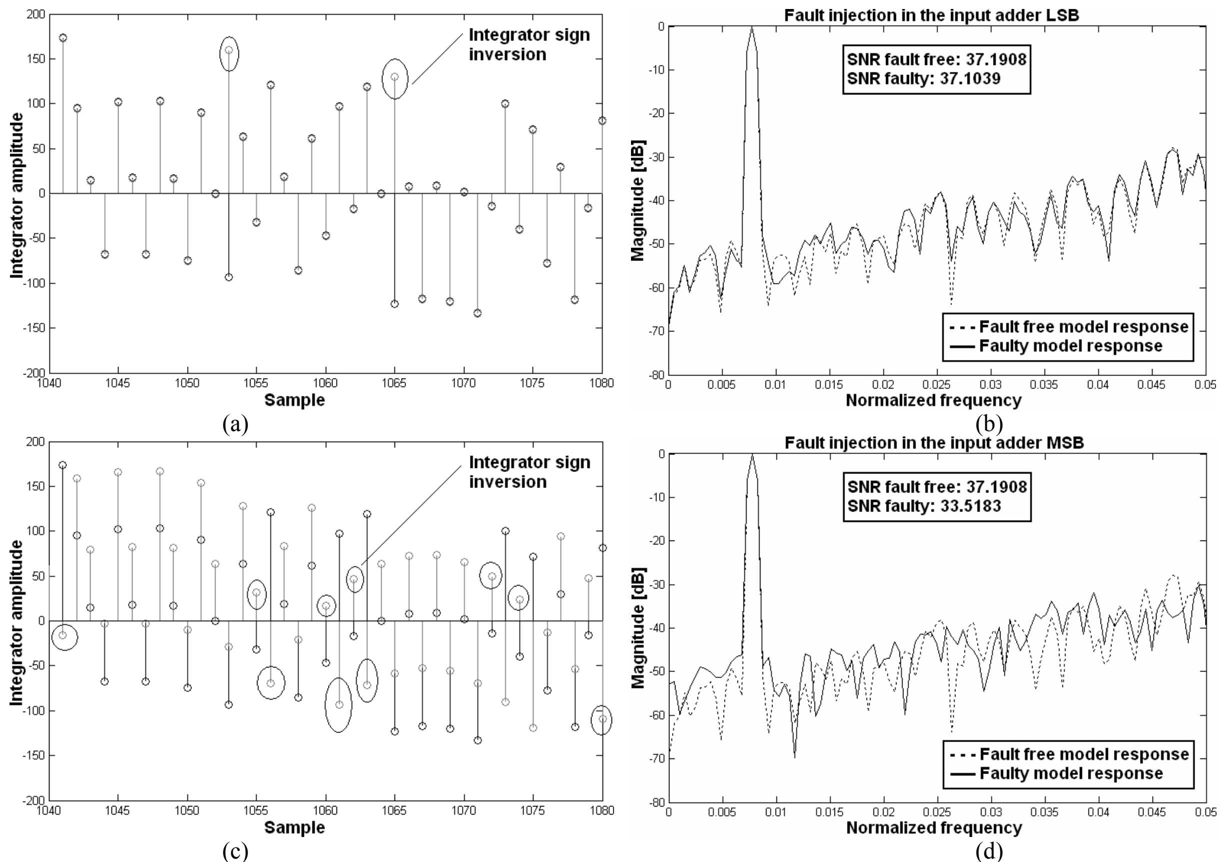


Figure 17: (a) Integrator output with one bit-flip in the LSB and (b) its consequence for the circuit response. In (c), integrator output with MSB inverted and the frequency response in (d).

It can be seen in figure 17(a) that the impulse function (added by a fault inverting the least significative bit in the output of the adder) is integrated, and keeps accumulated. For the linear model, one should expect this effect to eliminate itself, as predicted before. In the circuit, however, since the feedback is of a simple bit, the fault manifestation is obstructed by the quantizer and interferes in the bit stream only when the fault is large enough to invert the sign of the integrator output, and consequently, of the fed-back value. The consequence, showed in figure 17(b), is an interference occurring inside the signal band. Moreover, when the number of inversions increases due to the inversion of the most significative bit (see figure 17(c)), also the in-band interference increases, as plotted in figure 17(d).

One can conclude, thus, that for the ΣΔ circuit, no matter where faults occur, if their consequences are fed back, that is, if the fault causes an inversion in the integrator sign bit, the output signal can be more or less affected, depending on the number of inversions.

However, even with the occurrence of the faults, the final SNR maintains a certain resolution that can differ more or less from a non-faulty behavior.

2.3.2.2 First-Order *versus* High-Order Topologies under Faults [SCHÜLER, 2006a]

Once the consequences of faults to first-order delta-sigma modulators are presented, questions about how different topologies of delta-sigma modulators could be vulnerable to faults rise. The following analysis show the influence of bit-flips in higher-order modulators to the output bit stream, and how these faults can degrade the signal resolution. The simulations were done through a bit-flip fault model, as the one already presented in figure 13, and explained in section 2.3.2.1 for the first-order modulator.

An important fact to be taken into account to understand the consequences of faults in higher-order modulator is what happens when the fault propagates along the sigma-delta modulator circuit. Since a bit-flip can be seen as a delta function (impulse function), which will instantly add a value to the sample due to the xth bit inversion (see figure 13), when this delta passes through an integrator, it becomes a step function, that is, the effect of a single bit-flip lasts for the rest of the circuit operation.

If one now considers a second-order modulator, showed in figure 18, a similar analysis can be done. However, for this case, since there are two integrators instead of one, the fault consequence can be even more disastrous.

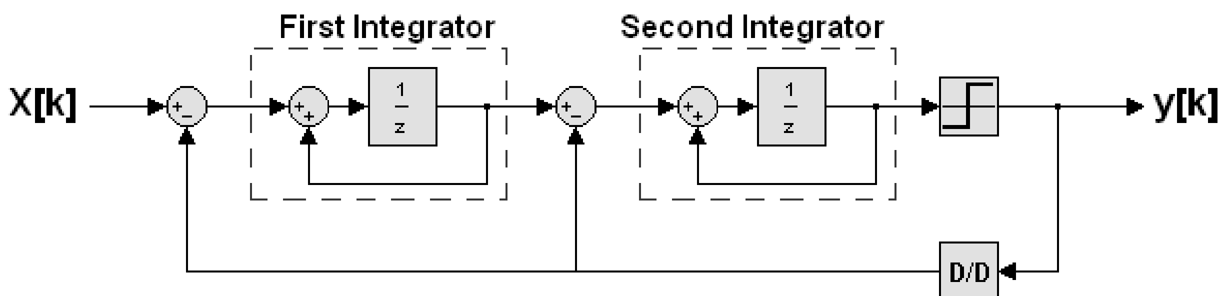


Figure 18: A second-order digital sigma-delta modulator.

Figure 19 shows the same simulations made for the first-order modulator, presented in figure 17. As seen for the second-order modulator, depending on which part of the circuit the fault occurs, the result can be more or less injurious. In figure 19(a), where one can see the fault occurring in the second subtractor of the modulator, it is possible to note that, although the signal band was affected by the fault, it was not affected as much as if the fault occurs in the first subtractor, as showed in the spectrum of figure 19(d).

This can be explained by the fact that, for faults occurring in the second subtractor, since they will be integrated only once (in the forward path), their consequences will not be as large as if they occur in the first subtractor, from where faults will be integrated twice (in the forward path). However, even for those faults hitting the second subtractor, the inversions due to these faults will be put back to the input of the circuit, thus being integrated twice, but not the faults themselves.

Comparing figure 19(b) and figure 19(e) one can see that the number of inversions in figure 19(b) is much lower than that of figure 19(e), as expected. Also, through figure 19(c) and figure 19(f), it is shown that the faults (or the inversions caused by them) are integrated twice and therefore the final result is a ramp function.

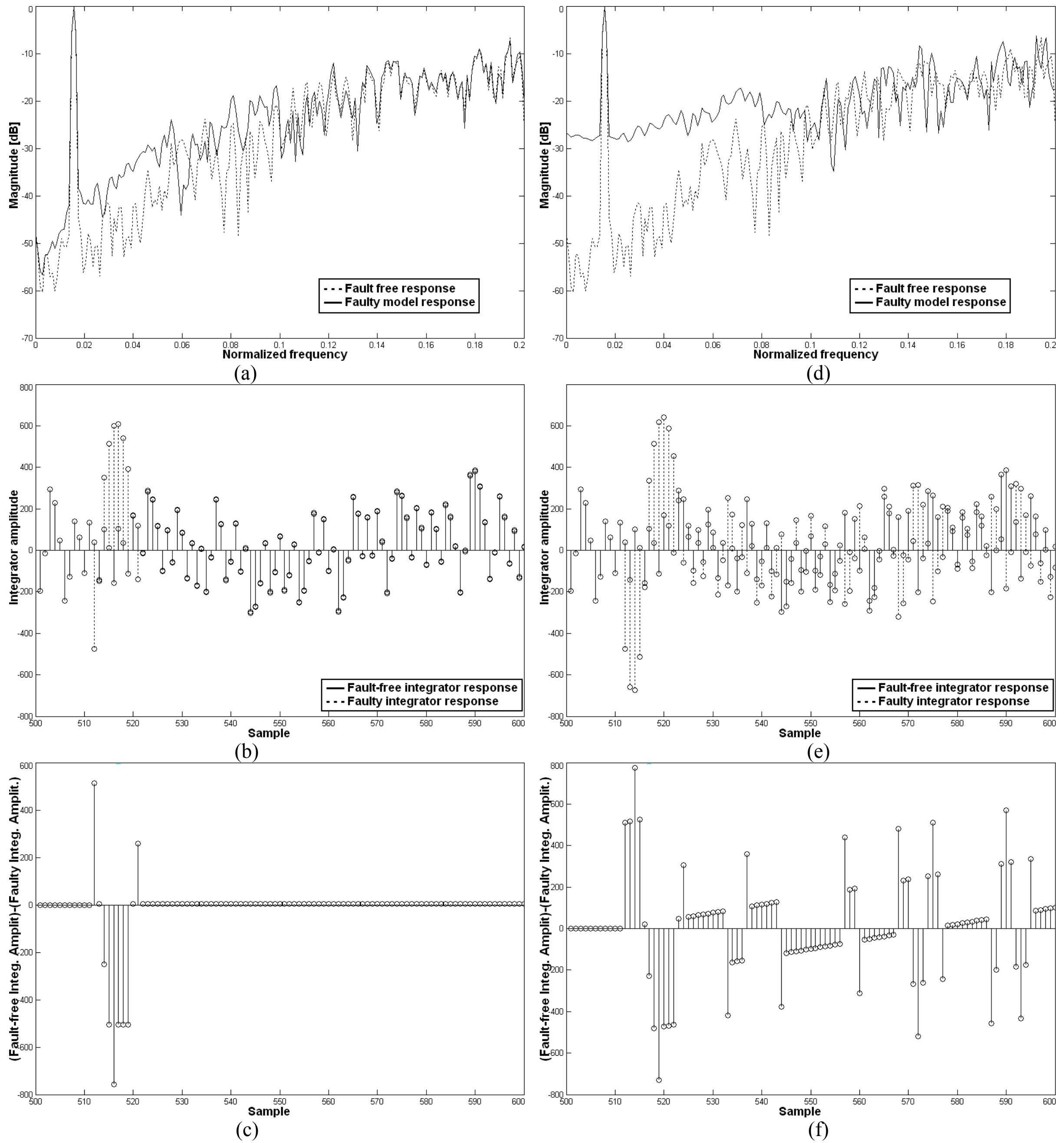


Figure 19: (a) Second-order modulator response, (b) integrator output and (c) ramp function after fault integration for faults being injected in the second block. (d) response, (e) integrator output and (f) ramp function after fault integration for faults being injected in the first block.

Taking first and second-order delta-sigma modulators, it is expected that the first-order modulator is more fault tolerant than the second-order, since if faults occur in the first block of the second-order, the results are much worse than if they occur in the second block. For the first-order modulator, however, no matter where faults occur, the consequences are not so large. This conclusion can be remarked in simulations of figure 20. The graph presents the input signal of a first and second-order modulators varied all through their input range and

two faults are inserted randomly in different parts of the circuit. The output bit streams generated with an OSR of 32 for figure 20(a) and figure 20(b) are decimated and plotted *versus* the input signal.

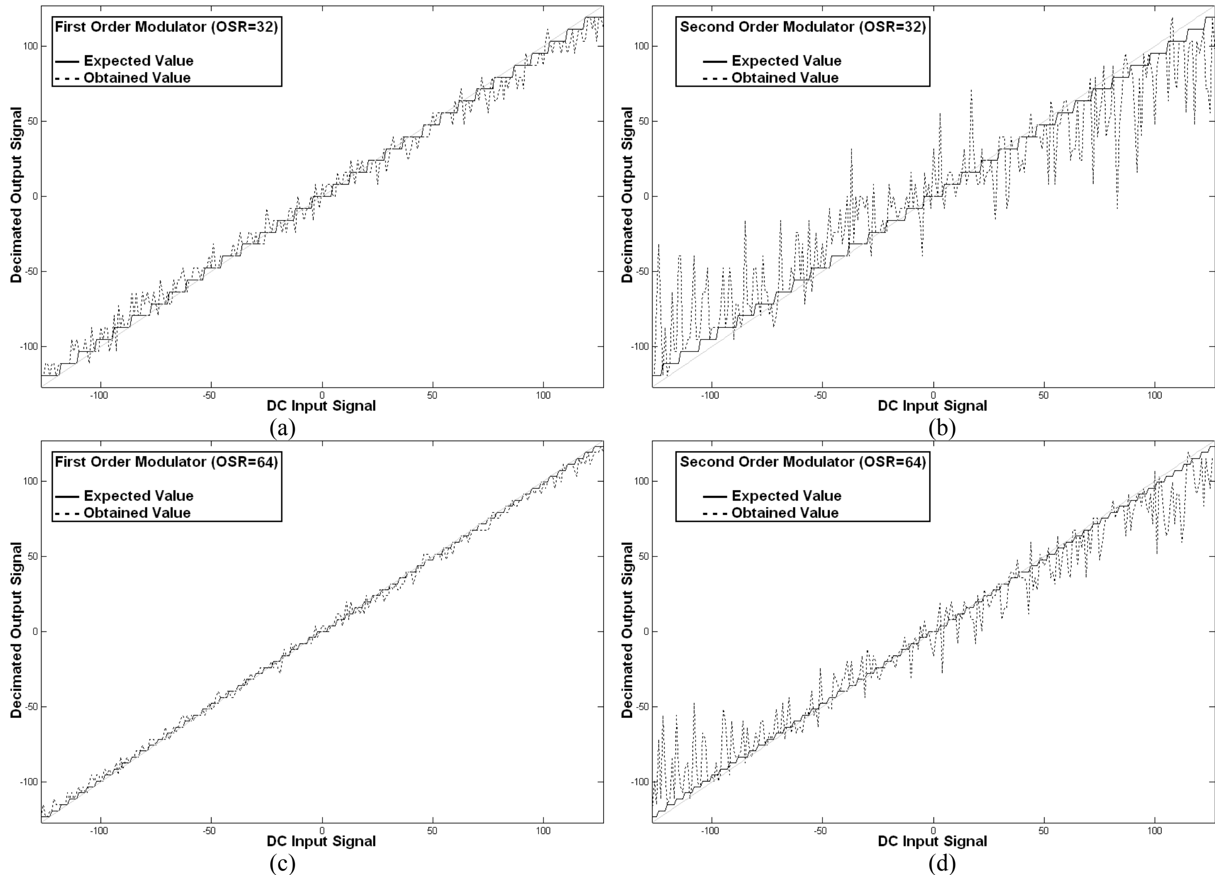


Figure 20: Delta-sigma input/output relation for different number of faults injected and different OSR: (a) first-order and OSR=32; (b) second-order and OSR=32; (c) first-order and OSR=64; (d) second-order and OSR=64.

As noted, for the second-order modulator, the results are much worse than for the first-order one, where the variation from the expected value is very small. These results can be improved by augmenting the OSR to 64, as showed in figure 20(c) and figure 20(d), but the second-order modulator still presents a higher degradation.

These results can be extended to higher-order modulators. Since more integration steps are added to the forward-path, the consequence of a bit-flip can be integrated more than once, thus degrading the output signal resolution. Also, due to the already mentioned intrinsic instability of higher-order modulators, these modulators can be even more affected by these faults.

2.3.3 Case Studies

In order to show the consequence of multiple faults in applications using sigma-delta modulated signals, some case studies were developed. First, arithmetic operations using sigma-delta signals were implemented, which will further be used in the development of more complex structures, like FIR and IIR filters. Finally, a DSP Microprocessor implemented specifically to deal with sigma-delta values is presented, and some practical results are shown. For these developed applications, faults were injected in the bit stream only. No fault was injected during the modulation process.

2.3.3.1 Arithmetic Operations [SCHÜLER, 2005a]

Four different arithmetic operations using sigma-delta modulated signals were developed: addition and subtraction, multiplication and exponentiation. Multiple faults were injected in the bit streams during the processing of these operations, through bit-flips, and the results were compared to classical implementations of the same operations using n-bit coded-modulated words.

Two different ways to make each arithmetic operation were used concerning the modulation from the digital to the sigma-delta domain. Here from now, we shall consider the “continuous” and the “discrete” operation modes. These definitions of continuous and discrete modulation do not come from the literature, but are created to this work’s pourposes. To understand the difference between these two operation modes, consider the sigma-delta modulation of a certain signal, *e.g.* a sinusoidal wave form. For the continuous modulation process, a classical sigma-delta modulation is used, that is, for each sampled point of the input signal, one bit of the bit stream is generated, and the input signal is over sampled (much more than the Nyquist frequency). On the other hand, if a discrete modulation is done, for each sampled point of the input signal (which does not need to be over sampled now), N bits will

be generated in the bit stream. In others words, the sampled value is holded during the modulation and generation of the N bits. Figure 21 better explains the difference between each kind of modulation. In figure 21(a), the continuous method is demonstrated, while the discrete mode is depicted in figure 21(b).

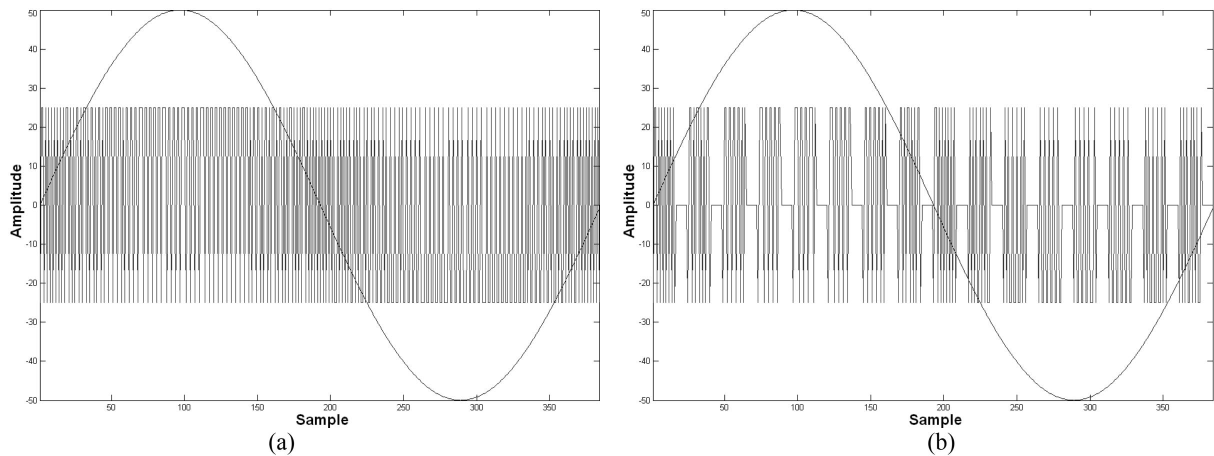


Figure 21: Continuous and discrete sigma-delta modulation.

These two modulation schemes would be exactly the same in terms of spectra, output bit stream and modulation process. However, since for the discrete mode for each new sampled point the modulation process restart, the error signal of the modulator is changed, not following the same error signal as if a continuous modulation was done. This leads to a small difference between their spectra and output bit stream as seen in figure 22.

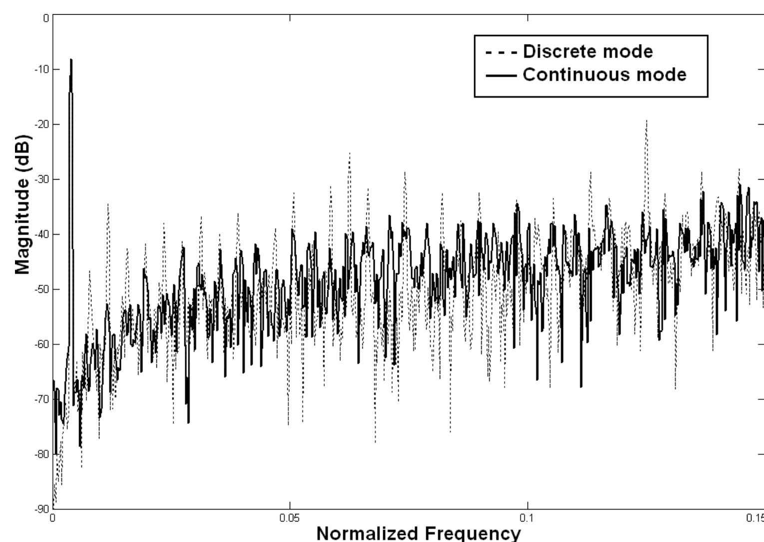


Figure 22: Spectra of the continuous and discrete sigma-delta modulation modes.

As one can see, for the discrete mode, extra harmonics appear exactly due to the discrete mode of operation, where each time a new point is sampled, the error signal returns to

another initial value. Taking this fact into account, one can conclude that the discrete modulation process is more fault-tolerant than the continuous one, because for the continuous modulation, a fault in the bit stream represents a fault in the signal itself, while in the discrete mode, a fault will represent a small variation in a sampled point, which has a constant value. For each arithmetic operation next described, these two situations will be analyzed and their fault-tolerance response analyzed, showing that, in fact, the discrete mode leads to better results in terms of resolution and tolerance, while the continuous mode contributes to better results in terms of performance and area overhead.

- *Addition/Subtraction with continuous ΣΔ modulation*

There are many different ways to add sigma-delta signals. In [DIAS, DA FONTE, 1994], the scheme shown in figure 23(a) is proposed, where two sigma-delta bit streams are digitally added, resulting in a 2-bit word stream, which is reconverted to a bit stream through another modulator. Another method is presented in figure 23(b), proposed by [O'LEARY, 1990]. In this case, the output of the circuit is the carry of a full adder, and the sum bit is stored and added to the following input bits. It is possible to show that, given two bit streams bs_a and bs_b, the output signal y is given by the following z-domain representation:

$$y(z) = bs_a(z) + bs_b(z) - (1 - z^{-1})\sum z \qquad (14)$$

For low frequencies, one has that $z^{-1} \rightarrow 1$, thus the output bit stream represents the addition of the input ones. The simplest way to add bit streams, however, is through the use of the interleaving operation, proposed in [MALOBERTI, 1991, 1992]. This technique consists, as the name accuses, in interleaving the bits of each bit stream. Since the bit streams generated by sigma-delta modulators can be seen as probabilistic signals [MALOBERTI, 1992], the use of stochastic adders, as the one in figure 23(c), is an easy solution to add these signals. Suppose two bit streams bs_a and bs_b with associated probabilities Pa and Pb that represent the values to be added, and a third bit stream bs_c, with an associated probability Pc,

used to determine which of the inputs will be transferred to the output. Figure 23(c), which implements a 2:1 multiplexer, can be used to generate a bit stream *y* representing the sum, with associated probability *PS*, so that:

$$\boldsymbol{PS} = \boldsymbol{Pc}.\boldsymbol{Pa} + (1 - \boldsymbol{Pc}).\boldsymbol{Pb} \qquad (15)$$

If *Pc* is equal to 0.5, which can be obtained simply by alternating 1s and 0s in the bit stream bs_c, expression (15) becomes:

$$\boldsymbol{PS} = 0.5.\boldsymbol{Pa} + 0.5.\boldsymbol{Pb} \qquad (16)$$

that is, the interleaving operation results in the addition of the two bit streams, divided by two.

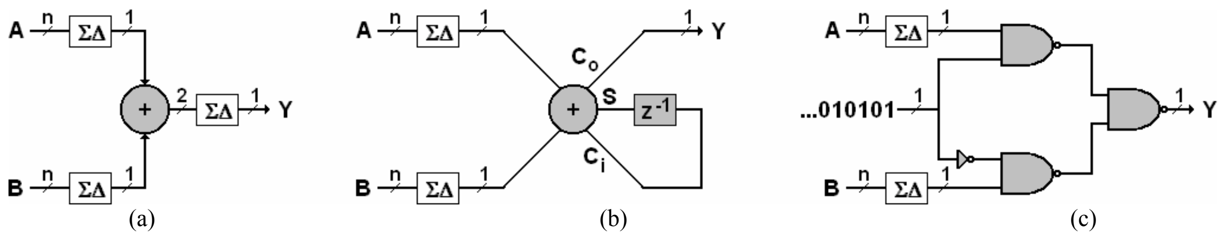


Figure 23: Three methods to add bit streams: (a) direct addition, (b) carry feedback and (c) interleaving.

In figure 24 it is presented three different simulations comparing the three addition techniques, while table 1 summarizes the advantages and disadvantages of these methods, including final bit stream length, spurious-free dynamic range (SFDR) and hardware size. For the simulations in figure 24, second-order modulators were used in all modulations. As noted, the best SNR is obtained in the third case (figure 24(c)), since no extra signal processing is done in the original bit streams. The second best SNR occurs in the direct addition (figure 24(a)) because the added bit streams are reconverted to the sigma-delta domain through another modulator, thus sending the quantization noise to high frequencies. The worst case, that using carry feedback, results in an approximation of the expected sum, as denotes equation (14).

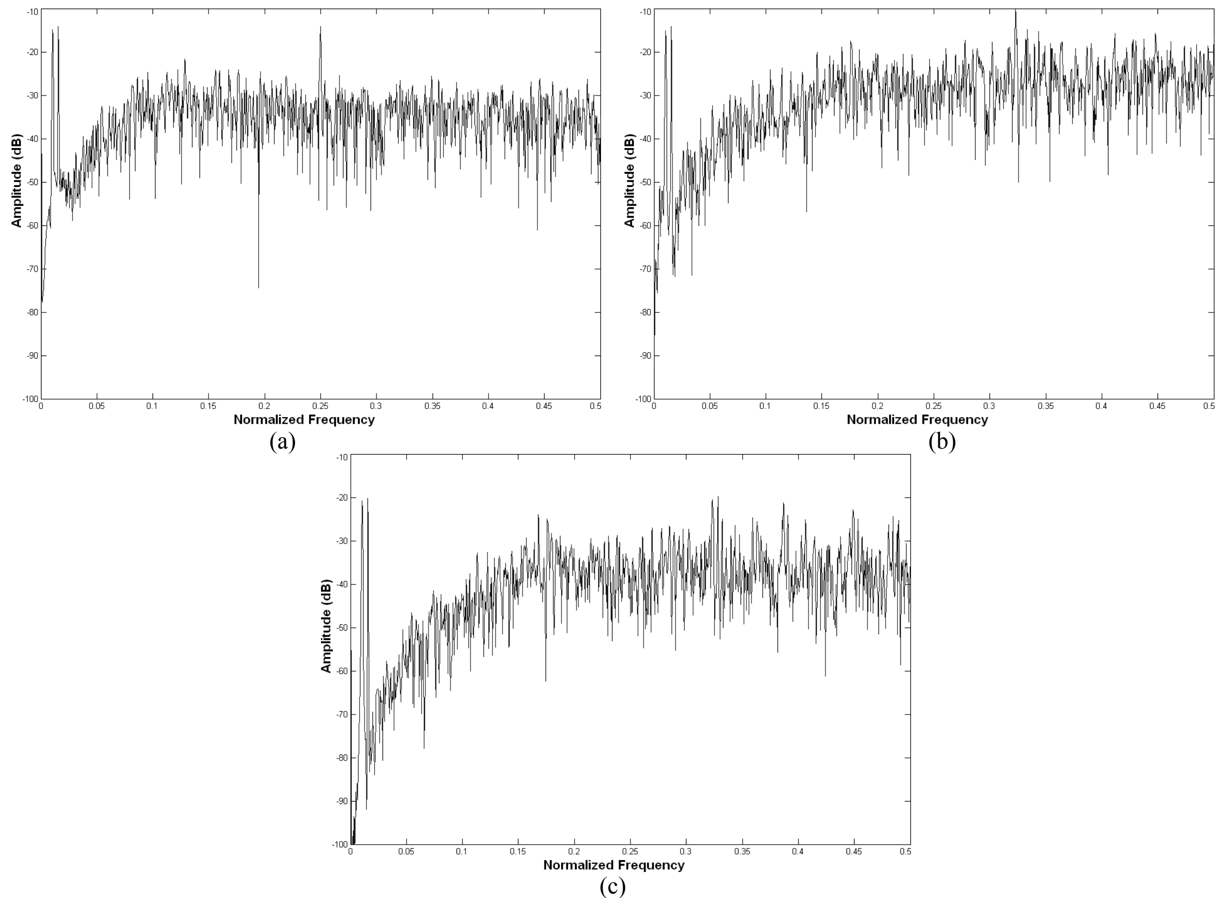


Figure 24: Comparing the three methods to add bit streams: (a) direct addition, (b) carry feedback and (c) interleaving.

Table 1: Summarizing the bit stream addition methods.

	Bit stream length	SFDR (%)	Hardware size
Direct addition	Y=length(bs_a)	77	5 gates + 3 ΣΔ
Carry feedback	Y=length(bs_a)	60	9 gates + 2 ΣΔ
Interleaving	Y=length(bs_a)+ length(bs_b)	100	3 gates + 2 ΣΔ

Something important to say is that the subtraction operation is done exactly in the same way the addition, but now the negative sign bit stream is done through the inversion of all the bits of the bit stream.

In figure 25, three comparisons are depicted. For each sigma-delta addition methodology, four faults were injected during the addition process, and the final result was compared to the addition of the same signals using a fault-free process. Faults were injected in

the input bit streams used to make the addition, through the flip of two random bits in each bit stream.

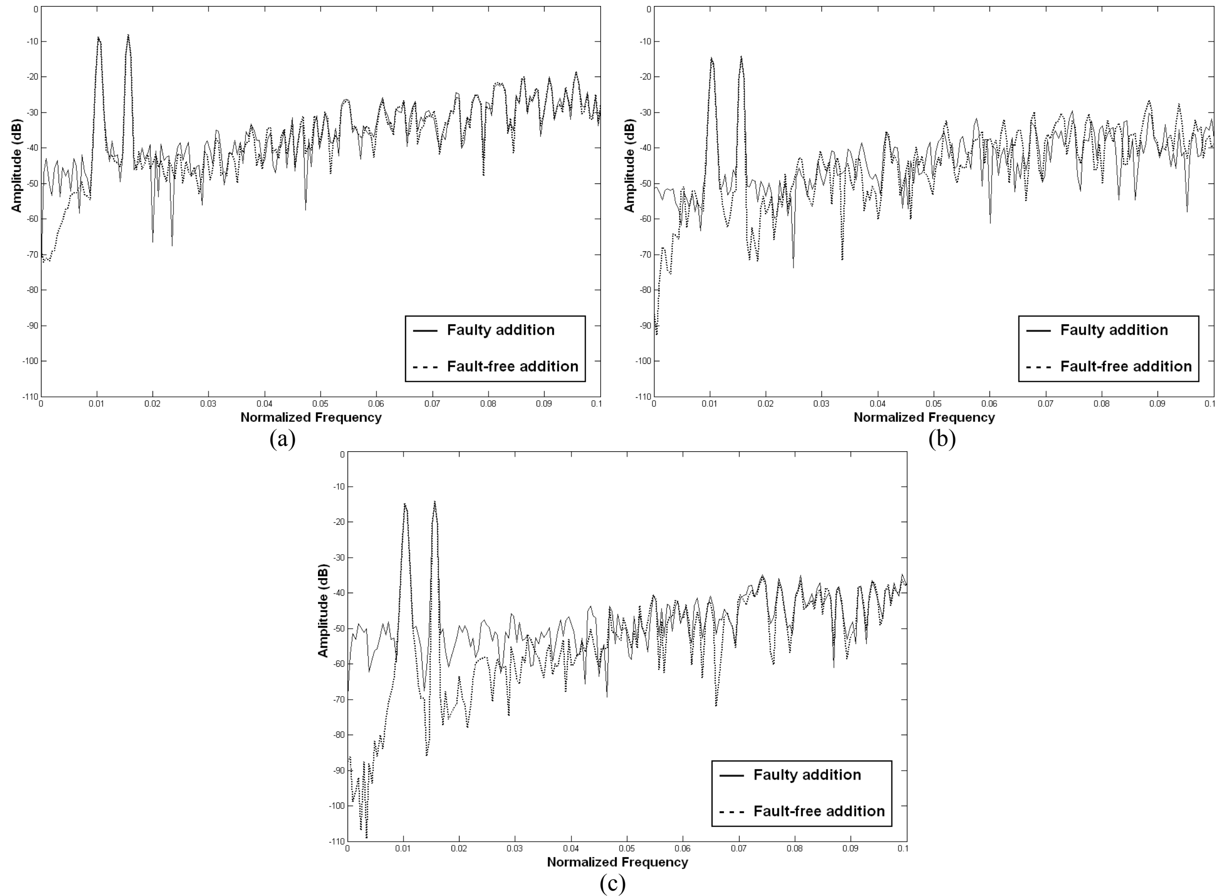


Figure 25: Comparing the three methods to add bit streams when faults are injected in the bit streams: (a) direct addition, (b) carry feedback and (c) interleaving.

As noted, for all the three cases, a significative drop in the SNR occurs. This can be seen by the noise added to the output signal band, mainly for the interleaving case. This can be explained by the fact that, when using the continuous modulation scheme, each bit stream represents an entire signal, thus, when faults occur in the bit stream, they will cause a direct influence in the signal itself, as already mentioned. As it will be shown next, for the discrete modulation, since each bit stream will represent a constant value, the consequences of the faults may be negligible to the final result.

- *Addition/Subtraction with discrete ΣΔ modulation*

The addition of bit streams generated through a discrete sigma-delta modulation can be done in the same way demonstrated for the continuous modulation. However, due to the

easiness of using the interleaving operation, this method is preferred instead of the others. An example of the use of the interleaving to add two constant values can be observed in figure 26, where two 9-bit constant values are added after being modulated to the sigma-delta domain through a first-order modulator. The final bit streams are decimated in order to analyze how far the addition results are from the expected value. No fault is injected during the modulation process, but 10 bits are inverted during the bit stream addition operation. Since these bits are randomly inverted, and thus a different sequence of ‘1’ to ‘0’ or ‘0’ to ‘1’ inversions can occur, the addition was made 200 times to evaluate the consequences of inverting bits in different positions of the bit stream.

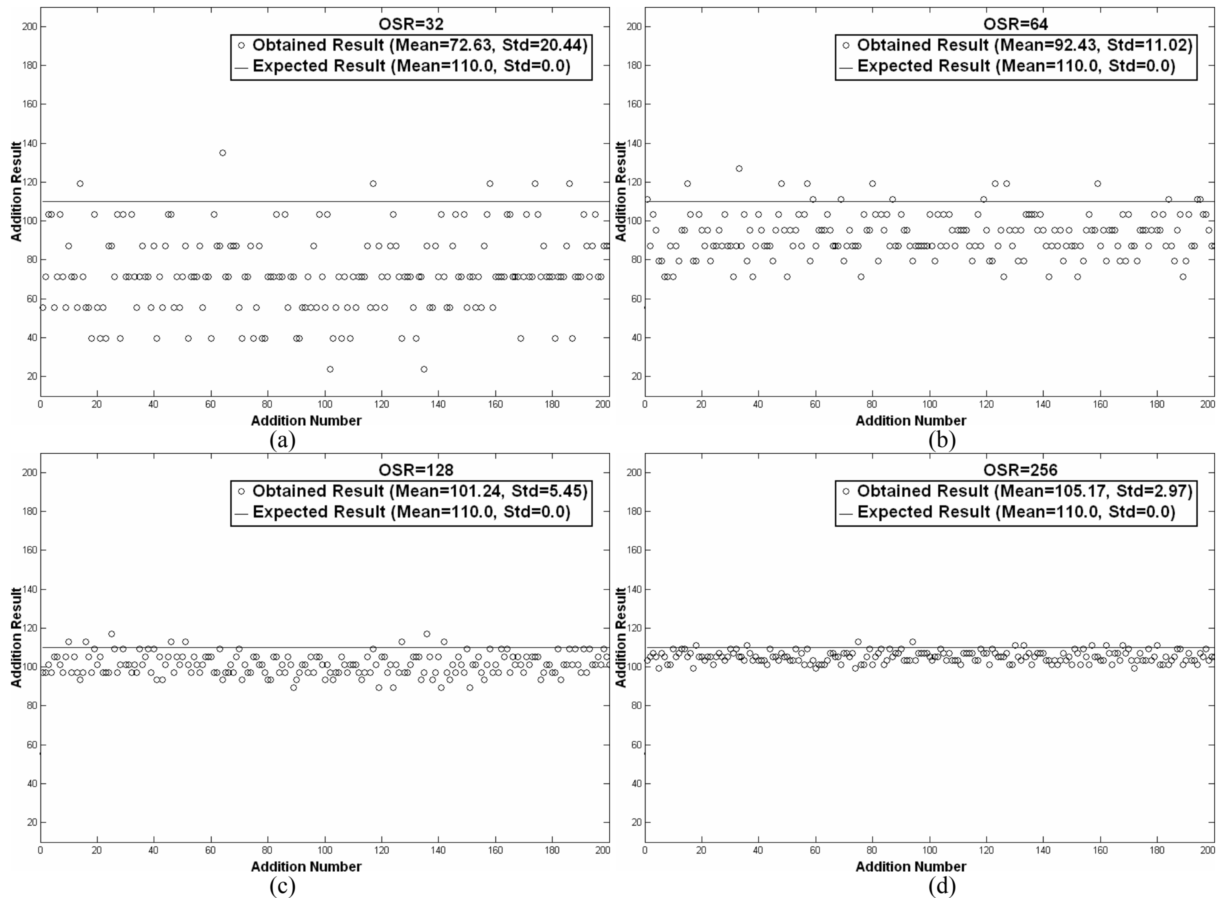


Figure 26: Addition through interleaving of two sigma-delta modulated bit streams, with faults injected in the addition process. As the OSR increases, also the final resolution and the fault tolerance increase as well. No fault is injected during modulation process.

As seen in figure 26(a), where the OSR is 32, there is a great deviation from the expected value, for almost all additions, what takes to a final mean value of 72.63 and a standard deviation of 20.44, thus far way from the expected values, which would be 110

(addition of 50 and 60) and 0, respectively. However, when the OSR is increased from 32 to 64, 128 and 256 (figure 26(b), (c) and (d), respectively), also the mean values of the additions increase from 72.63 to more than 105, and the standard deviation decreases from 20.44 to less than 3. So, since the redundancy of the operands is increased, also the obtained results resolution is increased, coming closer to the expected value.

- *Multiplication with continuous ΣΔ modulation*

The multiplication with continuous sigma-delta modulation does not generate many useful results. When dealing with bit stream representations, a multiplication operation can not be done in the same way as it is done with digital values. Instead, a simple AND operation would be necessary, since each bit stream is represented by a sequence of zeros and ones [MALOBERTI, 1992]. However, this is not an efficient way to make a bit stream multiplication. To understand that, it must be taken into account that, as mentioned in section 2.3.1, a bit stream contains the input signal plus a quantization noise. So, if a multiplication was done, it means that not only the signal is multiplied, but the noise is multiplied as well, as equation (17) shows:

$$\left(\boldsymbol{S}_A + \boldsymbol{N}_A\right).\left(\boldsymbol{S}_B + \boldsymbol{N}_B\right) = \boldsymbol{S}_A.\boldsymbol{S}_B + \boldsymbol{S}_A.\boldsymbol{N}_B + \boldsymbol{S}_B\boldsymbol{N}_A + \boldsymbol{N}_A.\boldsymbol{N}_B \qquad (17)$$

In equation (17), S_A and S_B are the input signals being multiplied, and N_A and N_B their respective quantization noise. As a consequence, the noise is spread all through the multiplication spectrum, because there will be four convolution terms in frequency: between signals, between signals and noises, and between noises. This last one leads to a white-like noise, which will contaminate the whole spectrum, as one can see in figure 27, where two sinusoidal signals are modulated with a second-order sigma-delta modulator and multiplied through the AND of their bit streams.

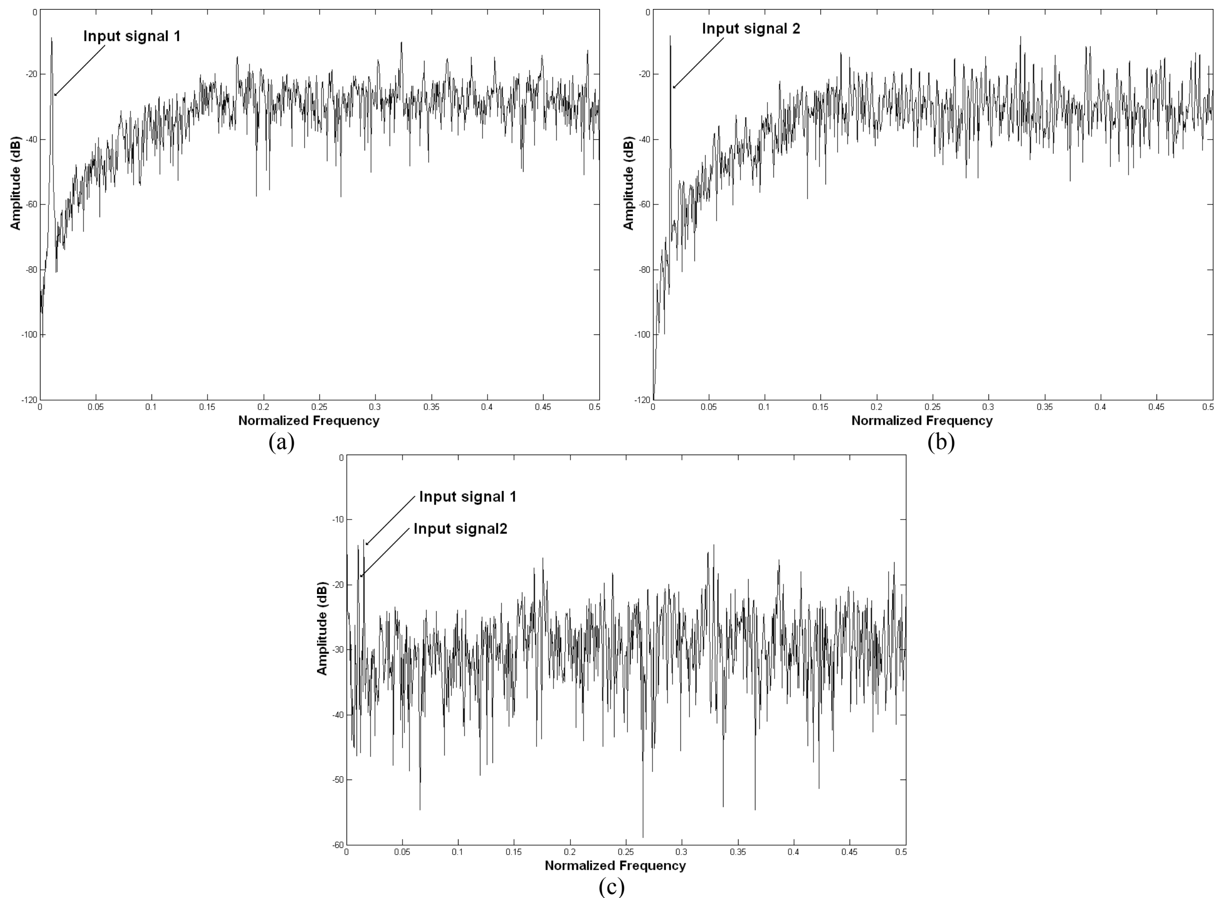


Figure 27: Multiplication of two bit streams through the AND of the bit streams: (a) Input signal 1 in ΣΔ, (b) input signal 2 in ΣΔ and (c) multiplied bit streams.

As noted, the final SNR is extremely degraded, leading this kind of operation to a useless result. This way, multiplication with discrete modulation must be used to the development of the applications, as will be explained next.

- *Multiplication with discrete ΣΔ modulation*

The idea behind the multiplication of two bit streams is quite simple: for each bit in one bit stream, one must add or subtract the other bit stream. To better understand this, suppose the multiplication of a sigma-delta bit stream, which represents a constant value A by another constant value B, represented in an n-bit coded-word. If for each bit of the bit stream representing value A the value B is added or subtracted, the final result will be the multiplication of A and B.

If now one considers the value B modulated in sigma-delta and represented by another bit stream, the idea remains the same, and the operation can be done through a set of XNOR

gates, as presents figure 28. Depending on the design requirement, more XNOR rows can be put in the multiplier in order to enhance the performance/area trade-off.

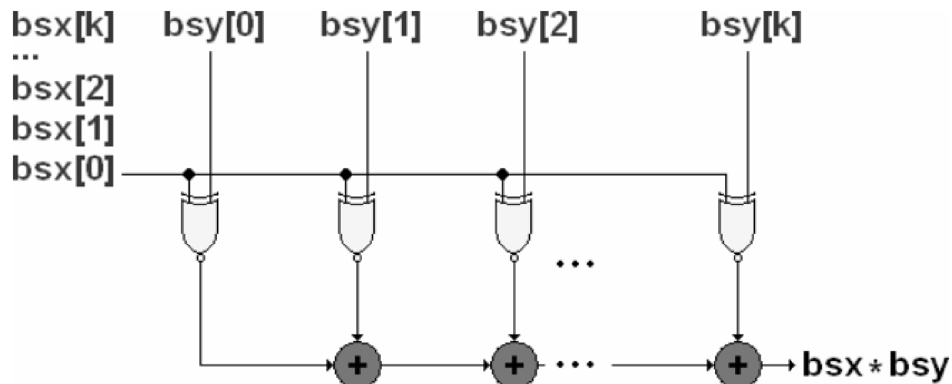


Figure 28: Bit stream multiplication using XNOR gates.

We shall now analyze the robustness of the sigma-delta multiplier. Consider, thus, the multiplication of a given 9-bit constant *A=70* by the value *B*, which varies with unitary step in the range *[-127 127]*. Both values are sigma-delta modulated with a given OSR and faults are injected in the operands through multiple bit flips. The results are compared to the multiplication of the same values using an n-bit representation with faults injected in the operands. For the sigma-delta multiplication, 5 bits are flipped in one of the operands, while for the n-bit multiplication, a single bit is flipped. Figure 29 shows the proposed comparison, where it can be seen that, even under multiple faults, the final value in the sigma-delta multiplication is much closer to the expected one, corroborating to the idea of error-tolerant systems.

In figure 29(a), an OSR equal to 32 is used to modulate each input value. As seen, the results are not as good as the ones presented in figure 29(b), where the OSR is increased to 64. In figure 29(c), the OSR is passed to 128 and, finally, to 256 in figure 29(d).

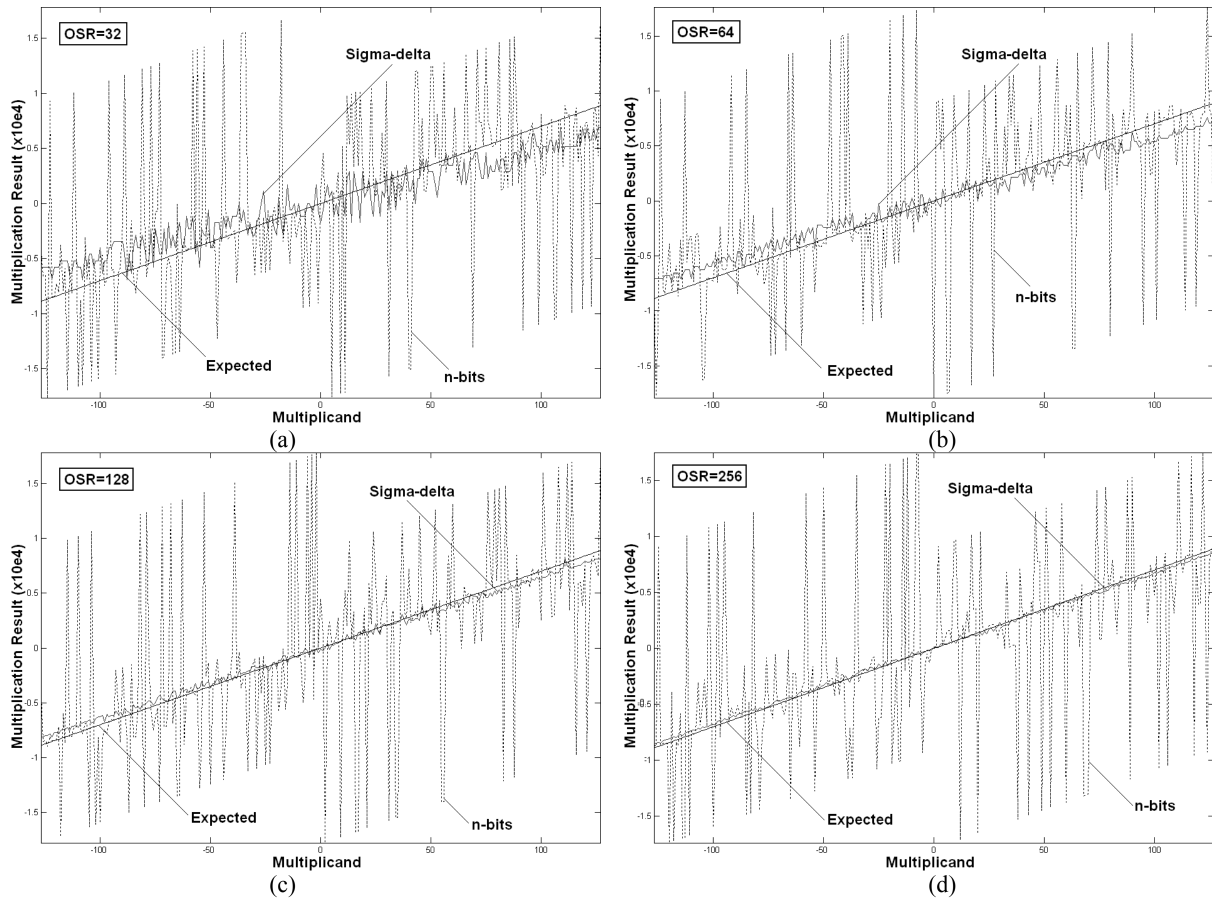


Figure 29: Multiplication of two constants using sigma-delta modulated values with different OSR: (a) OSR=32, (b) OSR=64, (c) OSR=128 and (d) OSR=256.

- *Exponentiation with discrete ΣΔ modulation*

As demonstrated before, the multiplication using continuous sigma-delta modulation does not result in good resolutions. So it is reasonable to expect that in the exponentiation process, the result of the exponential in the continuous modulation will not give good results either, since the exponentiation is based on the multiplication/division of the values.

The exponentiation with discrete modulation, however, can be done in an easy way. If one modules the exponent in sigma-delta, depending on the bit stream value, one multiplies or divides the base by its value. Figure 30 presents four simulations similar to those presented in the discrete modulation multiplication scheme, now applied to the exponentiation. The base $B=2$ is in the power of an exponent E, which varies with unitary step in the range *[-127 127]*.

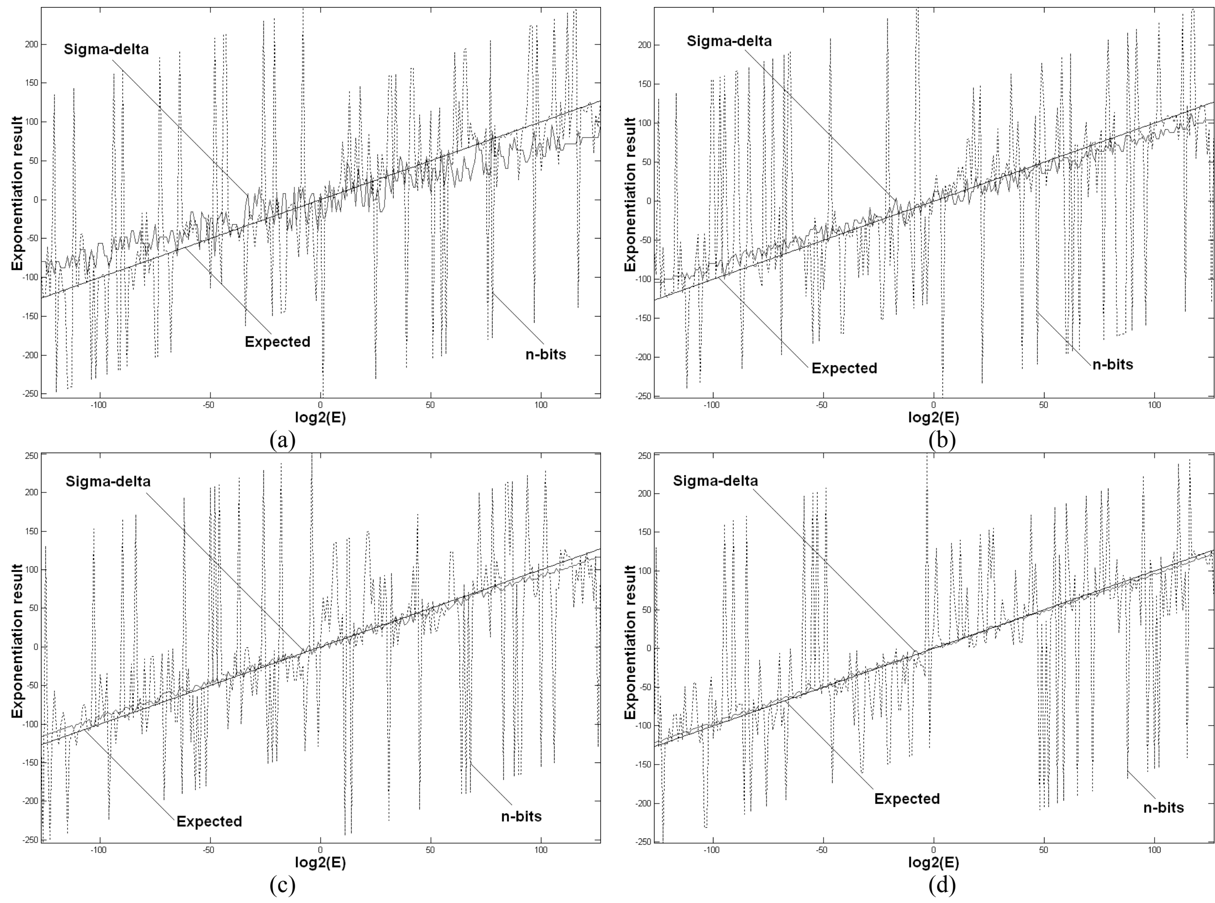


Figure 30: Base 2 in the power of exponent varying in the range [-127 127] modulated with different OSR: (a) OSR=32, (b) OSR=64, (c) OSR=128 and (d) OSR=256.

Again, in figure 30(a), an OSR equal to 32 is used to modulate each exponent value. In figure 30(b), (c) and (d) the OSR is increased to 64, 128 and 256, respectively, thus increasing the obtained resolution and approximating the curve from the expected value. The idea of increasing the OSR will be further explored in the FIR filter using sigma-delta modulation, showing that, when better resolutions or higher fault-tolerance are required, the increase in the OSR of the modulated values is a viable solution.

2.3.3.2 Finite Impulse Response (FIR) Filter [SCHÜLER, 2005b, 2006b, 2006c, 2007a]

As seen in section 2.3.3.1, both addition and multiplication using sigma-delta-modulated signals lead to a final mean value very close to the expected value, even after the insertion of multiple faults. So, one could think of extending this principle to some systems that make intensive use of arithmetic operations. An example of such application are digital

Finite Impulse Response (FIR) filters, which are digitally implemented through the use of the discrete convolution operation between the input signal and the filter coefficients, given by:

$$(y * b)[m] = \sum_{n} y[n].b[m-n] \tag{18}$$

where y is the discrete input signal and b the filter coefficients. So, to make digital FIR filters, additions and multiplications are necessary [OPPENHEIM, 1999]. In figure 31 one can a see an example of a Direct Form FIR filter structure is presented.

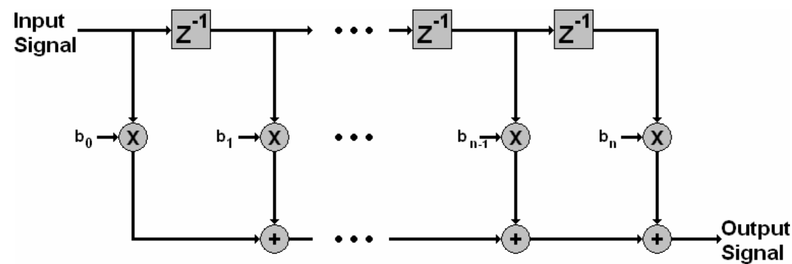


Figure 31: Direct Form FIR filter implementation.

Since two kinds of addition and multiplication are possible to be made with sigma-delta modulated signals, using discrete and continuous modulation, thus two different FIR topologies can be developed, also using the discrete and continuous modulation techniques.

- *FIR Filters with continuous ΣΔ modulation*

This section presents the results regarding the implementation of a FIR filter, which uses exclusively sigma-delta modulated signals, representing both the input signal and the filter coefficients. In this case, the input signal is modulated through a continuous modulation process, while each filter coefficient is modulated separately, each one generating a bit stream proportional to the OSR used in the modulation. In others words, to modulate each coefficient to the sigma-delta domain, each one is passed OSR times through the modulator, generating, thus, a bit stream with OSR bits *per* coefficient. The multiplication of the input signal and the coefficients is done by passing the filter coefficient value or the inverted coefficient value, depending on the sign of the input sigma-delta modulated signal. This is done via exclusive-or gates, as shown in figure 28. The addition is done by interleaving the coefficients bit streams, resulting in one filtered point. In order to analyze the final values, the resultant bit stream of

each filtered point is decimated to obtain a decimal value. The final structure of the filter is presented in figure 32.

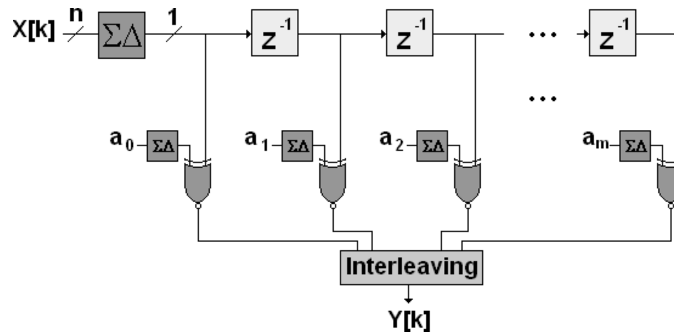


Figure 32: FIR filter using ΣΔ modulated signals only. Input signal is modulated through continuous modulation.

To have an exact match between the decimal and the sigma-delta modulated coefficients, each coefficient should be represented in a 256-bit bit stream. However, lower resolutions could be used to represent each bit stream, improving the filter performance, but reducing the final resolution. To see how the filtered signal resolution is improved by the increase of the OSR used to modulate each coefficient value, figure 33 depicts a Matlab® simulation result where the number of faults in the output bit stream is fixed, and the OSR used to modulate each coefficient is varied from 8 to 128. The SFDR of the filtered signal is measured, and then plotted versus the OSR value. As noted in figure 33, with an OSR of about 64 or more, one can obtain a constant resolution, even under the occurrence of four faults.

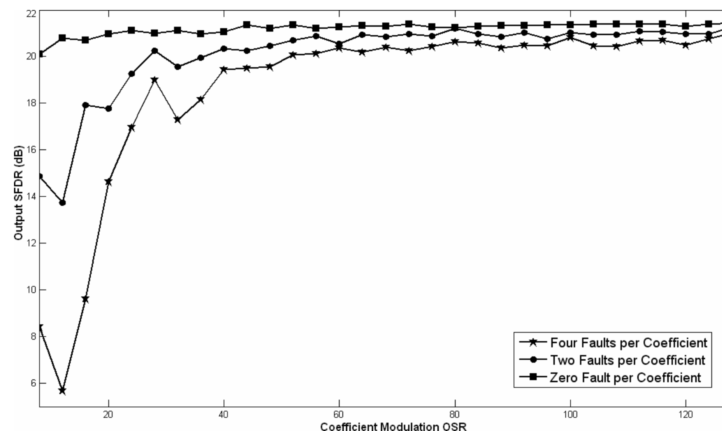


Figure 33: Resolution measured in SFDR versus OSR used to modulate FIR coefficients, for a constant number of faults injected in the coefficients bit streams.

To analyze the response of this filter, a 2KHz sinusoidal added to a white noise is filtered using the proposed circuit. The input signal is sampled with a given OSR and is sigma-delta modulated through a first-order modulator, generating 1024-bit bit stream that represents the input signal. The faults are injected in both the input adder and the integrator of the modulator through the inversion of one random bit in randomly chosen time periods. As showed in section 2.3.2.1, each fault injected inside the modulator may cause the inversion of more than one bit in the output bit stream. So, the total amount of faults that are processed by the application (in this case the filter) is much higher than the injected one.

Figure 34 shows the simulation for three different situations of faults injection in a 16-tap filter. For figure 34(a), no fault occurs, and one gets a very good approximation of the signal filtered using a common 9-bit code-modulated words and the proposed approach, where each coefficient is modulated with an OSR of 64. Although the filter transfer function do not match exactly with each other in high frequencies, the matching is almost perfect in the pass band, what guarantees a very close SNR. There is a considerable mismatch in the stop-band, which is consequence of the use of a continuous modulation scheme. This mismatch shall be eliminated with the use of the discrete modulation approach, as it will be demonstrated next.

For the second situation, showed in figure 34(b), the same OSR is used to modulate each coefficient. However, we now insert 8 faults during the modulation of each coefficient, that is, during 4 periods of 64 (the OSR for each coefficient), one random bit is inverted in the input adder and one in the integrator of the modulator. Also, 40 faults are inserted during the modulation of the input signal (20 for each part of the modulator). Figure 34(b) shows the results for the filter transfer function, which is affected only by the faults in the coefficients modulation process, the coefficients plot, also affected only by the 8 faults, and the filter output, where the faults consequences are due to the total amount of faults injected. Again,

although the filter transfer function degrades in high frequency, the degradation in low frequencies is not enough to cause a great reduction in the final SNR.

Finally, in order to show that an increase in the OSR used to modulate each coefficient can enhance the filter response, each coefficient is now modulated with an OSR equal to 128, while keeping the same number of faults. Results are showed in figure 34(c), where it is possible to see an increase in the SNR when compared to the previous experience and also an enhancement in the coefficients matching.

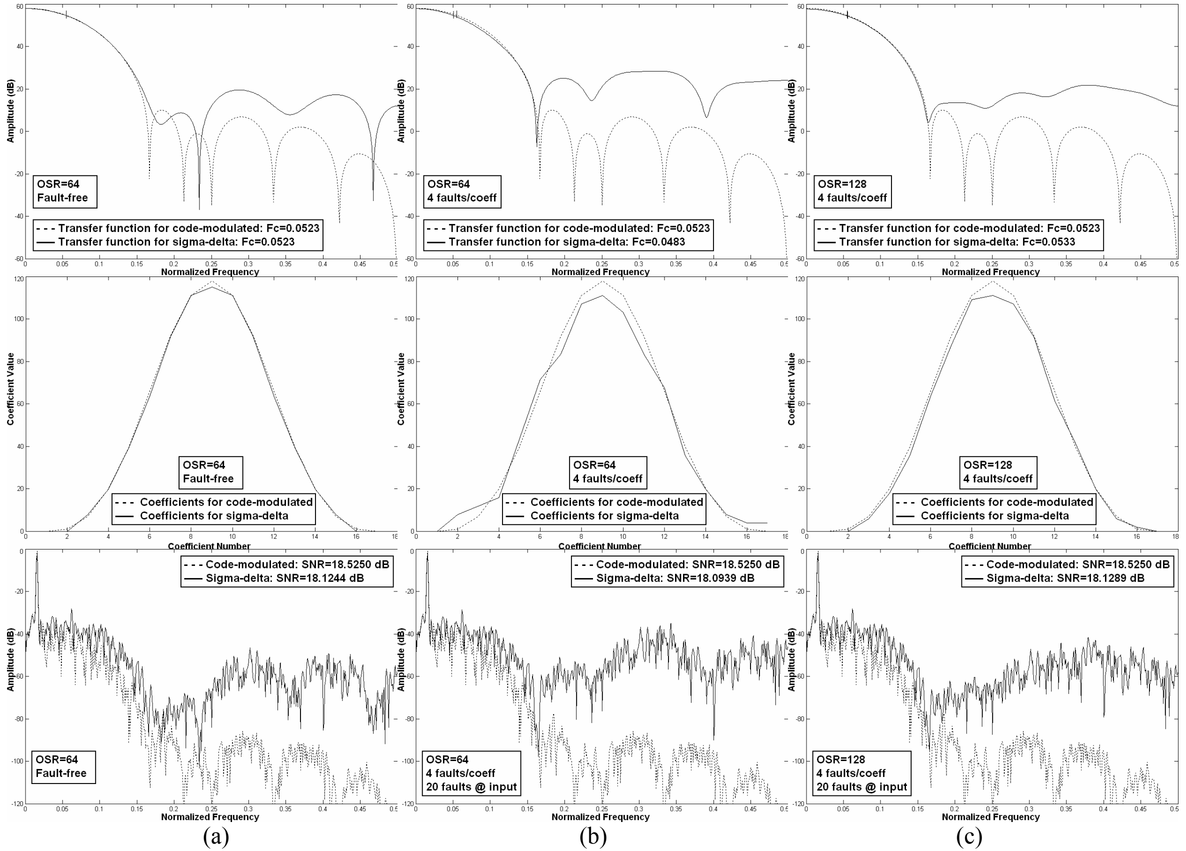


Figure 34: Filter response, coefficients and output FFT for different number of faults injected and different OSR used to modulate each filter coefficient.

- *FIR Filters with discrete ΣΔ modulation*

For the discrete-modulation scheme, one bit stream is generated for each input sampled point, thus the properties already seen for the multiplication in section 2.3.3.1 will apply. This will result in a higher resolution and fault-tolerance, as already seen but, on the other hand, a lower performance will dominate since the multiplication steps become slower. The modulation of the coefficients are done in the same way describe for the continuous-

modulation case. Figure 35 shows the structure used to implement the filter using a discrete modulation approach. Once more, the faults are simulated through the inversion of one random bit in the modulator adders, as done for the continuous-modulation filter. So, as showed in section 2.3.2.1, each fault injected inside the modulator will cause the inversion of more than one bit in the output bit stream.

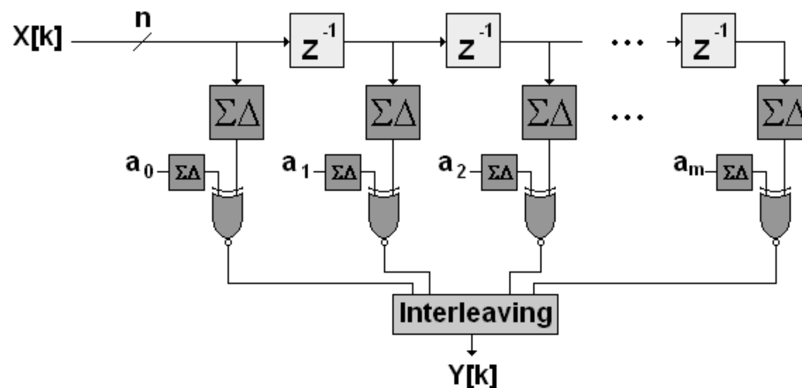


Figure 35: Representation of the structure used to filter signals using a discrete sigma-delta-modulation scheme.

Simulations results obtained for an 8-tap filter are presented in figure 36. The input signal formed by a single 20KHz tone and an amount of uniformly distributed random noise was sampled and filtered in two different ways: using the samples themselves and a common FIR filter without any fault inserted and using a sigma-delta modulation of the samples with the insertion of faults in the modulation of the input signal. All simulations results were done by taking 512 points of the input signal and modulating each of these signals. Faults were injected during the modulation of 32 of these 512 points. So, if one has ‘k’ faults injected during the modulation process, the total amount of faults is given by ‘32k’.

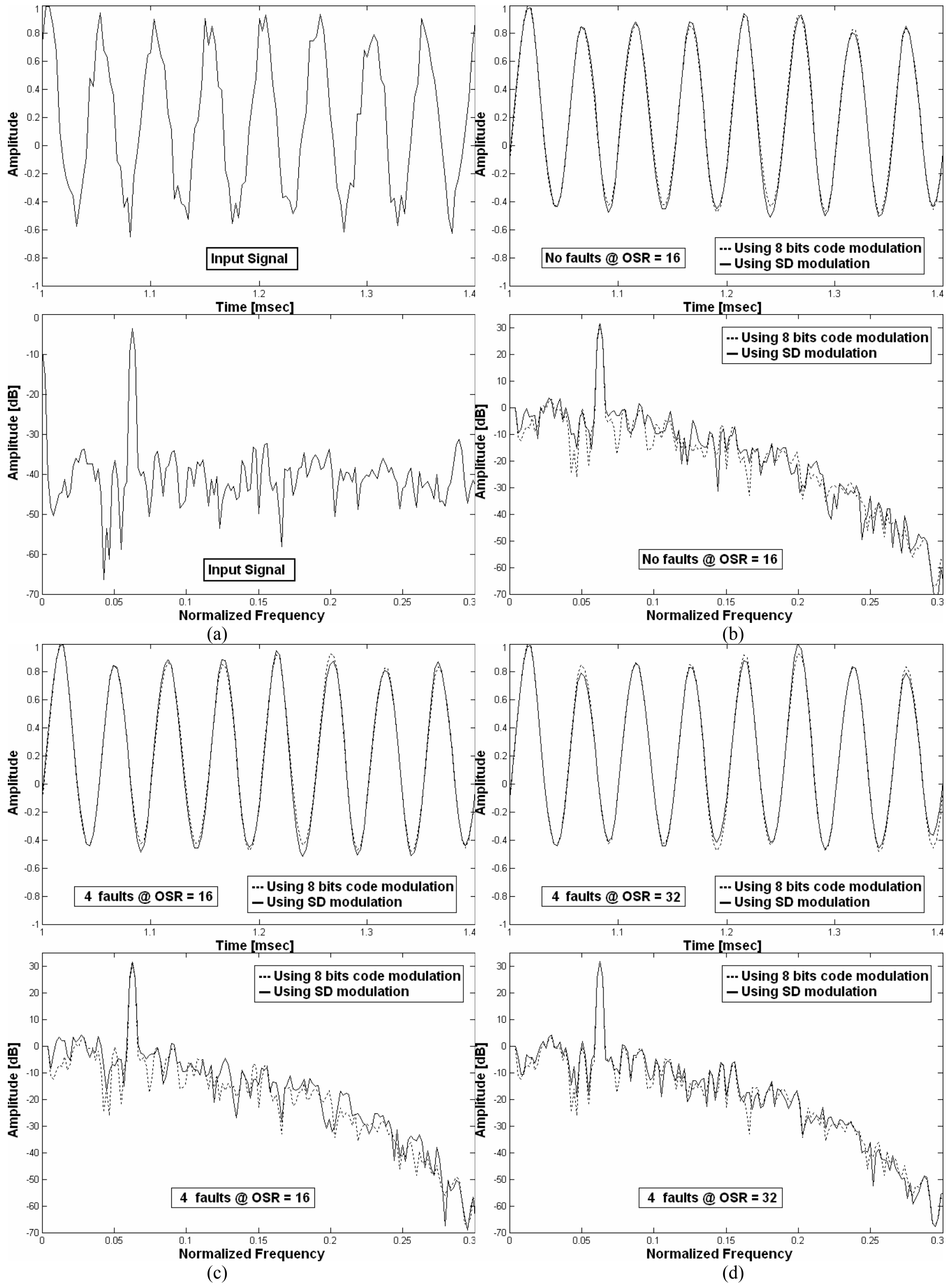


Figure 36: Results obtained by filtering a noisy 20KHz signal using sigma-delta modulated signals. In (a), the input signal, in (b), the filtered signal with no faults @OSR 16, in (c) with 4 faults @OSR 16 and in (d) with 4 faults @OSR 32.

Figure 36(a) shows the input signal in the time and frequency domain. In figure 36(b), it is shown a comparison between the output signals being filtered using a common FIR and

the proposed technique with no fault. Then, figure 36(c) presents these same results when 4 faults are injected in the modulation process of the input signal. As mentioned, these 4 faults are injected in the modulation of 32 points, resulting then in 128 bits inverted during this process. For figure 36(b) and figure 36(c), the OSR used to modulate each of the sampled points is 16. In figure 36(d), however, this OSR is increased to 32, showing that an even better resolution is achieved, even with the insertion of 4 faults.

2.3.3.3 Infinite Impulse Response (IIR) Filter [SCHÜLER, 2007a]

Another type of digital filter commonly implemented in DSP processors is the Infinite Impulse Response (IIR) filter. An example of a Direct Form II implementation is presented in figure 37(a). Note that, unlike the FIR filter (see figure 31), the output of an IIR filter depends on both the previous inputs and the previous outputs. This feedback mechanism is inherent in any IIR structure, being responsible for the infinite duration of the impulse response. An easy way to implement the filter presented in figure 37(a) by using sigma-delta modulated signals would be the one presented in figure 37(b), where the multipliers were substituted by XNOR gates, and the addition operation is implemented through interleaving.

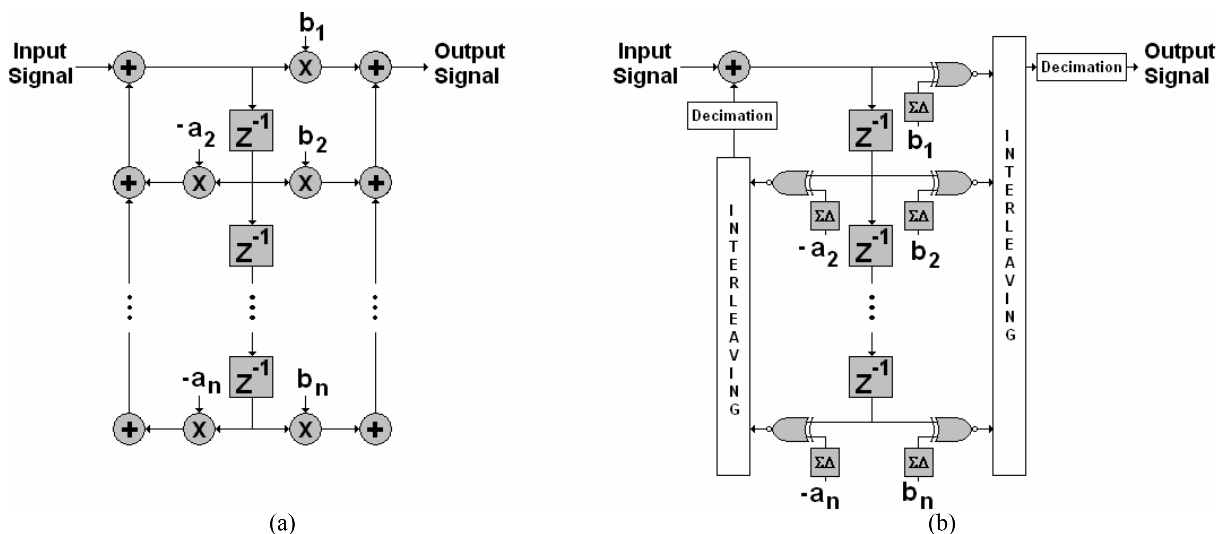


Figure 37: (a) Infinite Impulse Response filter classical digital implementation, and (b) structure using sigma-delta modulated input signal and coefficients for the same filter.

In order to implement the filter, a decimation step was included to change the 1-bit representation in the filter left-arm output, by an n-bit representation, allowing, thus, the addition with the n-bit input signal. The filter output is also decimated, but now only to have the results analyzed.

As mentioned before, the main structural difference between a FIR and an IIR filter is that the IIR presents a feedback structure. Since the IIR computes its output using the input values and the previous output values, some consequences rise, being one of the most important, the coefficient quantization sensitivity [TANSKANEN, 2000]. This can be explained, in a few words, by the fact that, when the output is not computed perfectly and is fed back, the imperfection can accumulate and completely modify the filter response. This effect can be seen in simulations results presented in figure 38 and in figure 39. In figure 38, one can see how a variation affects an 8-tap FIR filter response. As noted in figure 38(a), after modulating the coefficients in sigma-delta and decimating them in order to analyze the filter response, the coefficient quantization process causes a small variation in the filter response (figure 38(b)). On the other hand, for an IIR filter, even for a very small variation in the coefficients values (figure 39(a)) the consequences for the filter response are disastrous (figure 39(b)), causing a total mismatch between the expected and the acquired values.

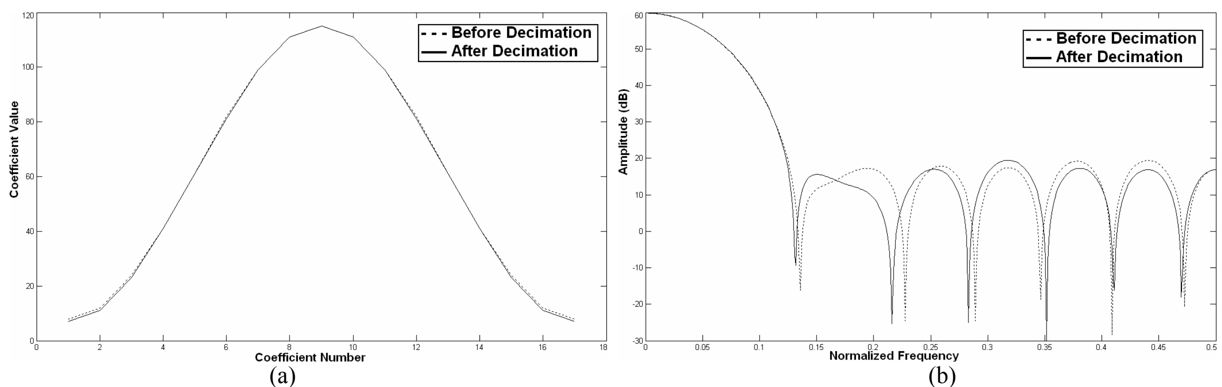


Figure 38: Finite Impulse Response filter coefficients (a) and filter response (b), showing the effects of coefficient quantization.

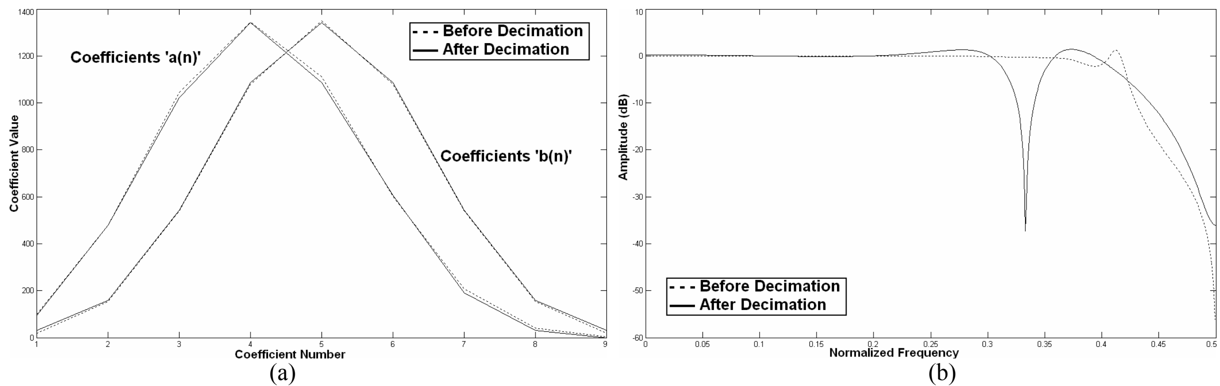


Figure 39: Infinite Impulse Response filter coefficients (a) and filter response (b), showing the effects of coefficient quantization.

Thus, due to this coefficient quantization effect, this kind of filter must be implemented by using another topology, like cascade form, parallel form, lattice-ladder, etc. For the topology presented in figure 37, one can conclude that it is not error-tolerant, and thus, should not be used with the proposed approach.

2.3.4 Functionally Fault Tolerant DSP Microprocessor [SCHÜLER, 2006d, 2007a]

Different digital signal processing functions can be easily developed through the use of DSP microprocessors, which are dedicated microprocessors able to realize many operations, like simultaneous memory access and Multiply and Accumulate (MAC) operations, in a single machine cycle. Through software programming, basic signal processing blocks like Infinite Impulse Response (IIR) filters, Finite Impulse Response (FIR) filters and Fast Fourier Transform (FFT) computation can be developed to be used in different applications involving, for example, audio and video.

In order to increase the robustness of these programmable devices, and presenting more complex applications using the proposed technique described at this work, a DSP microprocessor specifically developed to deal with sigma-delta modulated signals was described in VHDL, and the results are presented next. A comparison with a standard DSP microprocessor using n-bit code-modulated words is also presented, pointing out area and performance differences between these two implementations. As the section title proposes,

this architecture is supposed to be functionally fault tolerant, that is, the processor must generate acceptable results, even under the occurrence of multiple faults. Acceptable values, as already mentioned, are values which will generate a final response that still performs the desirable system specifications.

The DSP structure is based on the Analog Devices ADSP2100 [ANALOG, 2006], which has a relatively simple structure, composed of Multiply and Accumulate block (MAC), an Arithmetic Logic Unit (ALU) and a shifter block, besides the 16-bit bus and data addresses generators. This structure can be seen in figure 40. Also, its instruction set contains no more than 19 commands [ANALOG, 2006].

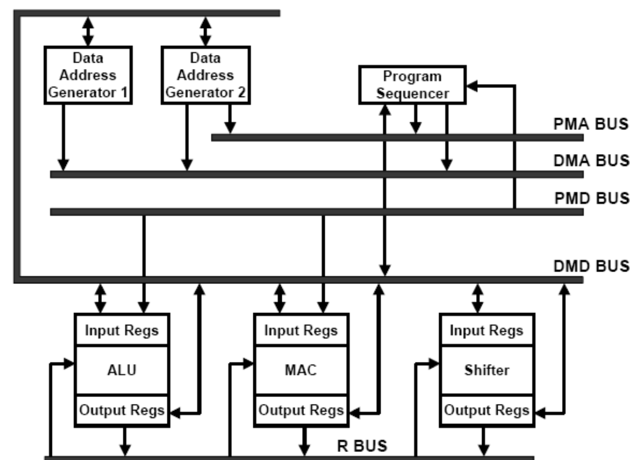


Figure 40: Block diagram of ADSP2100 DSP microprocessor.

The Sigma-Delta DSP (SDDSP), however, presents some modifications, since many Boolean operations developed in the digital domain do not work in the same way as in the sigma-delta domain [MALOBERTI, 1992] [DIAS, DA FONTE, 1994]. For example, to add two n-bit digital values, one must make an exclusive-or operation between each bit of the input values and the carry-out bit, which is also calculated through other Boolean operations. To add two sigma-delta bit streams, however, a simple interleaving operation, as already described in section 2.3.3.1, is carried out. Figure 41 shows the SDDSP internal structure, which has an ALU, a shifter block, some control signals and two internal RAM memories, one for data and one for program.

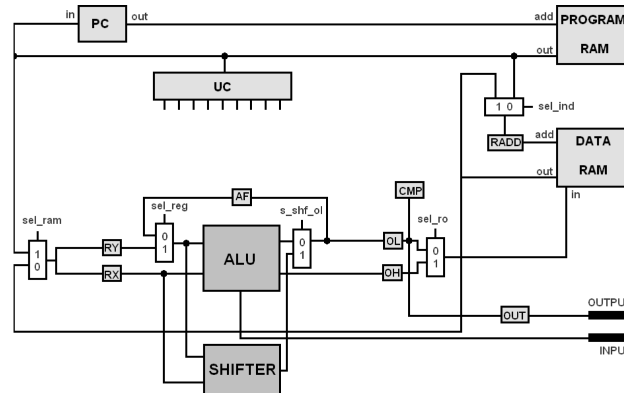


Figure 41: Sigma-Delta DSP microprocessor structure.

The instruction set comprises 27 instructions, presented in table 2, including the 19 from the ADSP2100, plus some extra instructions used to process the sigma-delta bit stream. The whole structure was described in 870 lines of VHDL, and prototyped using an Altera ACEX1K family EP1K100QC208-3 FPGA, occupying a total of 1836 logic elements, representing 36% of the total available in the FPGA.

Table 2: Sigma-Delta DSP instruction set.

Instruction	Op. Code	Instruction	Op. Code
NOP	00000	ADD	01110
LOAD	00001	SUB	01111
LOAD_IM	00010	BS_ADD	10000
LOAD_IND	00011	BS_SUB	10001
STORE	00100	JMP	10010
STORE_IND	00101	JZ	10011
PASS	00110	JNZ	10100
NOT	00111	JSHC	10101
NEGATE	01000	JNSHC	10110
AND	01001	SHIFT	10111
OR	01010	SHF_IN	11000
XOR	01011	DIV	11001
INC	01100	MOV_OUT	11010
DEC	01101		

To evaluate the DSP functionality, a 16 taps FIR filter was programmed, where both the input signal and the coefficients are modulated in sigma-delta, in a continuous way (see section 2.3.3.2). These signals were generated in Matlab® and then saved in the DSP data memory.

To simulate dynamic faults occurring in the DSP processor, faults were injected during each coefficient modulation process through the inversion of a different number of randomly chosen bits. As already seen in section 2.3.2.1, a single inversion in one bit during the modulation process can incur in a significant number of bits inverted in the generated bit stream. Since this bit stream will be used in the DSP processor, this is a good strategy to evaluate the consequence of multiple faults within the processor itself.

The filter structure is composed of simple XNOR gates, which will pass the value of the filter coefficient if the input signal is ‘1’, or the negate coefficient value otherwise. The XNOR outputs are added through an interleaving operation, and one filtered point is obtained (see section 2.3.3.2). In order to acquire the filtered results in a manner able to be evaluated, that is, not in the sigma-delta domain, but in the digital domain, a decimation block was added just after the interleaving operation. The filtered points were acquired using an Agilent Infiniium oscilloscope at sample rate of 1MSa/s. The values are then analyzed in Matlab®. The input signal is a digital 200Hz single tone with white noise added, modulated in sigma-delta with an OSR of 64, generating a total of 1024 bits in sigma-delta representation.

Figure 42 shows different responses obtained after filtering the input signal using the described filter with different OSR used to modulate each coefficient. A comparison between a fault-free and a faulty behavior is presented, both using sigma-delta modulation directly acquired from the DSP microprocessor. The respective filter impulse response is also presented. In figure 42(a), an OSR of 16 is used in each coefficient. In this case, two faults are injected during the modulation of each coefficient, both in the modulator input adder and integrator, resulting in a total of four faults per coefficient.

As noted in figure 42(a), the difference between the fault-free and faulty response is practically null, proved by the proximity between their SNR. Moreover, this difference can be reduced through the increase of the OSR used to modulate the coefficients, as presented in

figure 42(b), where an OSR of 32 is used, also with four faults injected during each coefficient modulation. Finally, increasing the OSR to 64 and the number of faults to eight per coefficient, the final response still matches the faulty-free response, as shows figure 42(c).

It may seem strange that, while improving the OSR from 32 to 64 the final SNR decays, but since the number of faults has also been increased, this is an expected behavior.

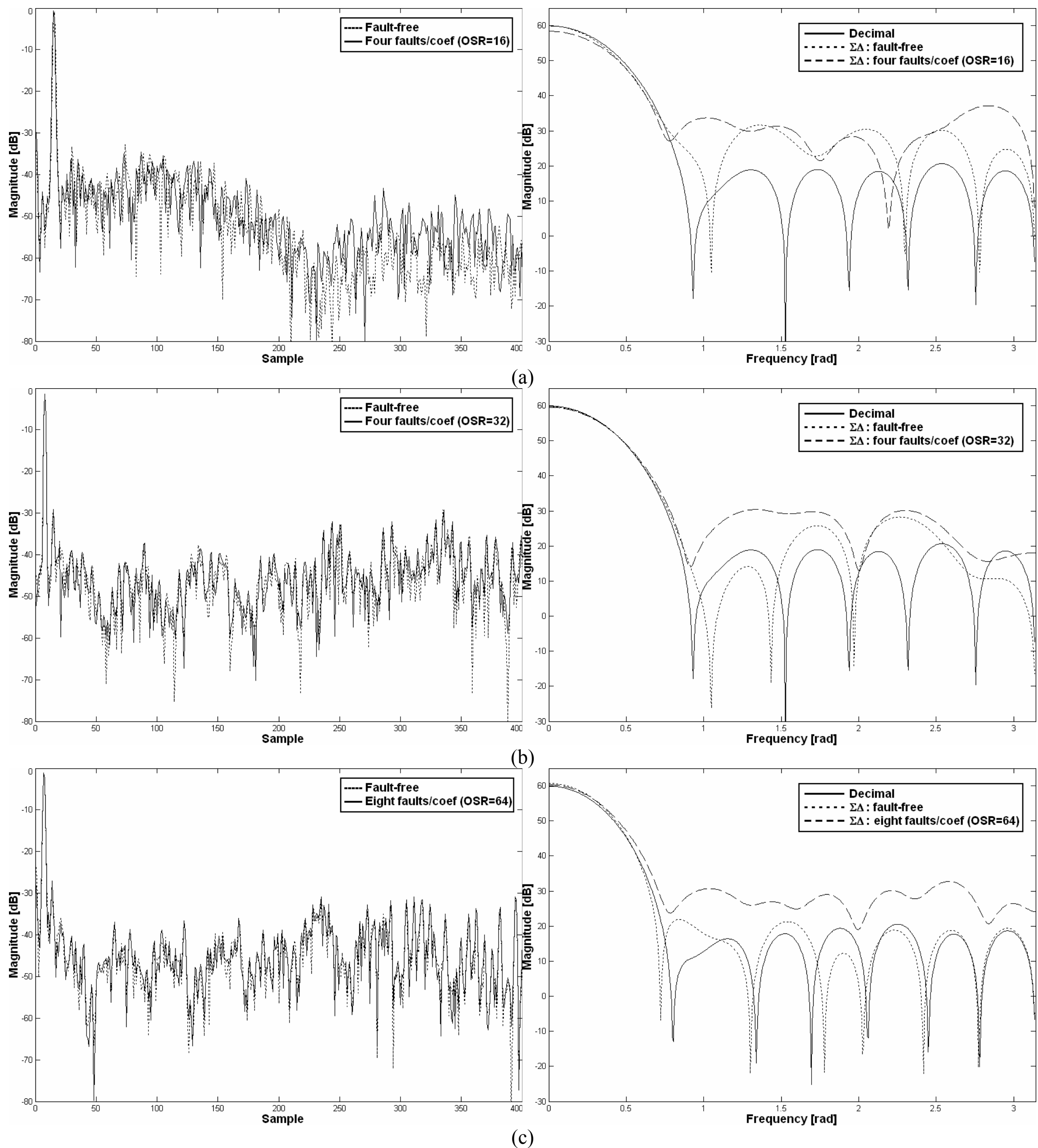


Figure 42: Practical results from the ΣΔ-DSP implementing a 16 taps FIR filter with different OSR used to modulate the coefficients, and respective filter impulse response.

Developing a new system structure requires some comparisons with standard systems, which realize the same function as the new one. This way, we must compare the sigma-delta-based DSP microprocessor to a standard implementation of the processor, that is, a DSP that uses n-bit PCM words, in order to evaluate some performance measurements like area and processing time. In fact, this DSP was the base used to create the sigma-delta version, so its structure is almost the same, with few modifications, as shows figure 43. Table 3 presents the instruction set of this microprocessor.

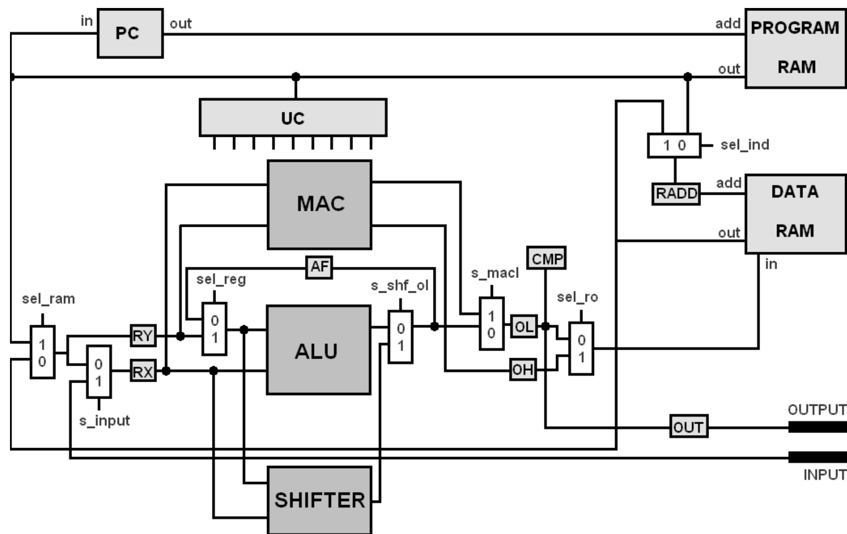


Figure 43: n-bit PCM words DSP microprocessor structure.

Table 3: n-bit PCM DSP instruction set.

Instruction	Op. Code	Instruction	Op. Code
NOP	00000	DEC	01101
LOAD	00001	ADD	01110
LOAD_IM	00010	SUB	01111
LOAD_IND	00011	JMP	10010
STORE	00100	JZ	10011
STORE_IND	00101	JNZ	10100
PASS	00110	JSHC	10101
NOT	00111	JNSHC	10110
NEGATE	01000	SHIFT	10111
AND	01001	SHF_IN	11000
OR	01010	DIV	11001
XOR	01011	MOV_OUT	11010
INC	01100		

The system is based on three blocks: an Arithmetic Logic Unit (ALU), a Multiply and Accumulate block (MAC) and a Shifter block. As noted, the most evident difference (see

figure 41 for comparison) is the presence of the MAC block, which is now necessary to make multiply and accumulate operations with the n-bit input words. All structure, such as the sigma-delta version, has 16-bit and was prototyped using the same Altera ACEX1K family EP1K100QC208-3 FPGA, occupying a total of 2330 logic elements, representing 46% of the total available in the FPGA.

To validate the DSP, the same FIR filter was implemented, now with the classical requirements for a digital filter that is, using n-bit multiplications and additions operations. This filter structure was already presented in figure 31. The filtered points were also acquired using an Agilent Infiniium oscilloscope at sample rate of 1MSa/s. The values are then analyzed in Matlab®. The input signal is the same digital 200Hz single tone with a white noise added, sampled with an OSR of 64, generating a total of 1024 16-bit PCM words.

Figure 44 presents practical results obtained from the DSP, when implementing the FIR filter. In figure 44(a), it is represented the filter impulse response when no fault is injected in the coefficients, and when one single bit is inverted in two different coefficients (b1 and b10). Note that, compared to the sigma-delta implementation, the number of injected faults is much smaller, but their consequences to the filter response are much more severe. Figure 44(b) depicts the filter coefficients, showing how faults were injected in coefficients b1 and b10. Finally, figure 44(c) shows the filter output after being converted to the decimal representation. One can clearly observe the consequences of a single fault in the filter response, which, in this case, reduced the attenuation of the filter rejection-band in almost 40dB.

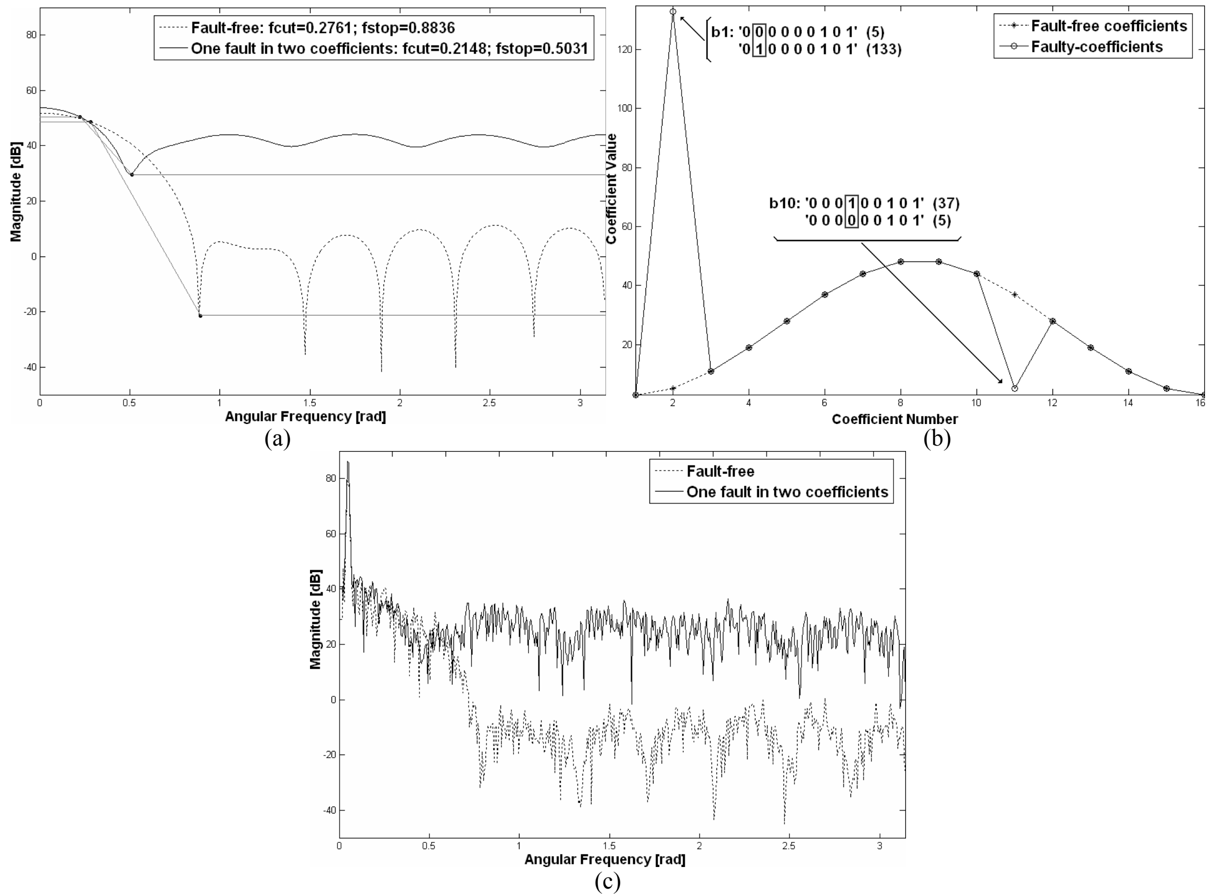


Figure 44: Standard DSP microprocessor simulation results. In (a), fault-free and faulty filter impulse response; (b) filter coefficients, and in (c), FIR output from the fault-free and faulty DSP response.

To complete the exposition about the SDDSP, table 4 summarizes some comparisons regarding both DSP processors implementations. The implementation in two different FPGA, an ACEX1K and a Cyclone one are presented. It is important to notice some interesting aspects here: although they have almost the same internal structure, the sigma-delta implementation presents a small gain in the occupied area, since it does not need to implement, for example, the standard area consuming multiplication operation, at least not in the same fashion as the normal processor when using a continuous sigma-delta modulation scheme. For the same reason, there is a gain by a factor of two in the performance aspect. Of course, this gain will decrease, certainly being also inverted, whenever more robustness and/or resolution are required. In this case, the OSR for the input signal or the coefficient realization will have to be increased, reflecting in a decrease in the sigma-delta DSP

performance. Nevertheless, this may be an acceptable compromise when a high fault tolerance must be achieved, mainly in critical parts of a system, or even for the whole system.

Table 4: Sigma-delta and standard DSP microprocessor comparison summary.

Evaluated Item	Sigma-Delta DSP	PCM DSP	ΣΔ Modulator (1st order)
Logic Elements ACEX1K EP1K100QC208-3	1882/4992 (36%)	2330/4992 (46%)	36/4992 (<1%)
Time to filter one point* ACEX1K EP1K100QC208-3	60μS	103μS	---
Logic Elements CYCLONE EP2C5F256C6	1422/4608 (30%) + 185 registers	1384/4608 (30%) + 233 registers + 2/23 embedded multiplier (9-bit elements)	30/4608 (<1%)
Time to filter one point* CYCLONE EP2C5F256C6	60μS	103μS	---

*** The same clock (30MHz) was used for both DSP. Time evaluations are made for an OSR equal to 16 for each coefficient modulation.**

Note also that, although presented in table 4, the cost to make the conversion of the digital signals to a sigma-delta representation is not computed in the area value, neither in the processing time. This can be explained by the fact that this signal conversion can be done inside the system by a sigma-delta converter, or one can supposed that the whole system already works with sigma-delta signals (coming from an ADC, for example), thus with no need for modulators inside it.

2.3.5 Area and Performance Comparison [SCHÜLER, 2005c]

Although table 4 already presents a comparison for area and performance, this is done through a FPGA construction. However, a more precise area and performance evaluation is carried out in order to analyze the penalties introduced when using the proposed techniques to improve systems reliability against SEUs. The comparisons were done with the filters presented in section 2.3.3.2 and a normal FIR filter using n-bit code-modulated words, depicted in figure 31.

2.3.5.1 Area Analysis

To calculate the area of each filter, the partial areas described in table 5 were used, all of them expressed in terms of MOS transistors, where *n* is the number of bits for each block. The sigma-delta used is that presented in figure 9(a), and the *2n* multiplier of the normal FIR filter is a serial-parallel version presented in [ERCEGOVAC, 1998].

Table 5: Number of gates *per* bit in each circuit and number of MOS transistors in each gate.

GATE	NUMBER OF MOS TRANSISTOR
AND	6
NOT	2
NAND	4
OR	6
XOR	12
BLOCK	**NUMBER OF GATES *PER* BIT**
Register	4 AND + 1 NOT
Adder	2 XOR + 2 AND + 1 OR
2:1 Multiplexer	2 NOT + 2 AND + 1 OR
Interleaving	3 NAND + 1 NOT

Consider the FIR filter using n-bit code-modulated words, presented in figure 31. The area of this filter can be given by:

$$A_{code} = (h+1).A_{multiplier_n} + h.(A_{register_n} + A_{adder_{2n}}) \tag{19}$$

where *h* is the number of taps of the filter. Given the areas presented in table 5, the final area of the filter is:

$$A_{code} = n(302.h + 200) \tag{20}$$

For the filter using continuous sigma-delta modulation of figure 32, the area is given by:

$$A_{continuous} = (h+1).A_{XOR_1} + A_{\Sigma\Delta_n} + h.A_{register_1} + A_{\text{interleaving}} \tag{21}$$

Using the values in table 5, one has:

$$A_{continuous} = 102.n + 30.h + 128 \tag{22}$$

Finally, the filter using a discrete sigma-delta modulation scheme, as the one presented in figure 35, has the following area:

$$A_{discrete} = (h+1).(A_{XOR1} + A_{\Sigma\Delta n}) + h.A_{register n} + A_{\text{interleaving}} \qquad (23)$$

Which results in:

$$A_{discrete} = n.(120.h + 102) + 114.h + 128 \qquad (24)$$

Figure 45 presents how the area of the different filters grows when the number of bits in the structure increases. This graphic was done for a number of taps equal to 32. As expected, looking at equations (20), (22) and (24), the area for the code-modulated filter grows much faster than the area for the continuous sigma-delta. However, the difference between the code-modulated and the discrete sigma-delta filter is not that large, since the last one requires one sigma-delta modulator for each branch of the filter.

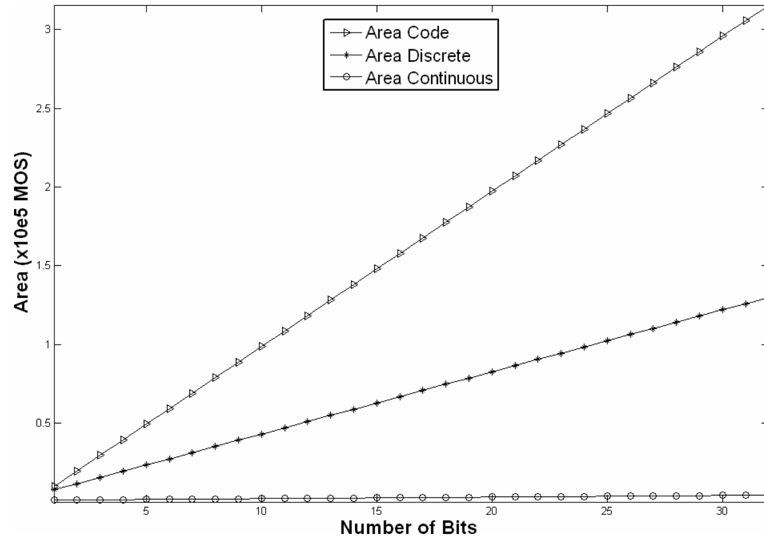


Figure 45: Comparison of area versus number of bit between FIR filter using code-modulated and sigma-delta modulated signals.

2.3.5.2 Performance Analysis

Independently of the filter implementation (code-modulated, continuous sigma-delta and discrete sigma-delta), since a convolution operation must be done (see equation (18)), two steps must be done during the calculation of each filtered point: multiplication and addition. Therefore, the difference between the performances of the filters will be present into these processes.

Figure 46 presents the finite state machine for the proposed filters. As seen, they will differ in the number of cycles needed to perform the multiplication and addition. For the multiplication in the code-modulated filter, only one cycle is necessary, the same for the addition process. On the other hand, for the multiplication in the continuous sigma-delta scheme, the number of cycles to make a multiplication is proportional to the OSR used to modulate each filter coefficient and, for the addition step, OSR times the number of taps cycles are needed. In the discrete sigma-delta approach, the scenario is harder, since OSR^2 cycles are needed to the multiplication, and $OSR^2.h$ for the addition.

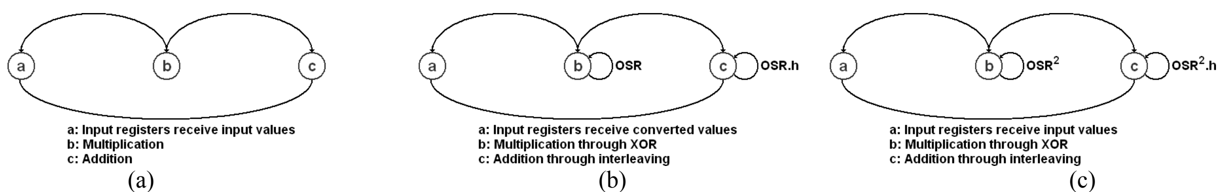


Figure 46: Finite State Machine for (a) the code-modulated FIR, (b) continuous sigma-delta modulation FIR and (c) discrete sigma-delta modulation FIR.

Of course, these situations can be improved. Like in all systems, there is a trade-off between area and performance. For the sigma-delta filters, if one wants to improve performance, a set of parallel XOR gates can be used instead of only one gate. This will reduce the number of cycles used in the multiplication step. At the limit, OSR XOR could be used in the continuous scheme, and OSR^2 in the discrete, thus realizing the multiplication in one cycle only. To improve the interleaving performance, one possible solution is to increase the frequency of the '01' sequence in the middle point of the circuit (see figure 23(c) for reference).

2.3.6 0.35µm Technology Chip Development

During the period of partial doctoral fellowship (sandwich doctorate), the development of a silicon chip was proposed in order to make comparisons between a classical and a sigma-delta implementation of a given system.

The circuit used to be implemented was the 16-tap FIR filter using discrete sigma-delta modulation, since comparisons can be easier made if one has both responses identical (see figure 34 and figure 36). Figure 47 presents the ASIC scheme. The idea is that only the operational part will be designed, since it is the protected one. The control part should be implemented externally to the chip, through a FPGA or microcontroller.

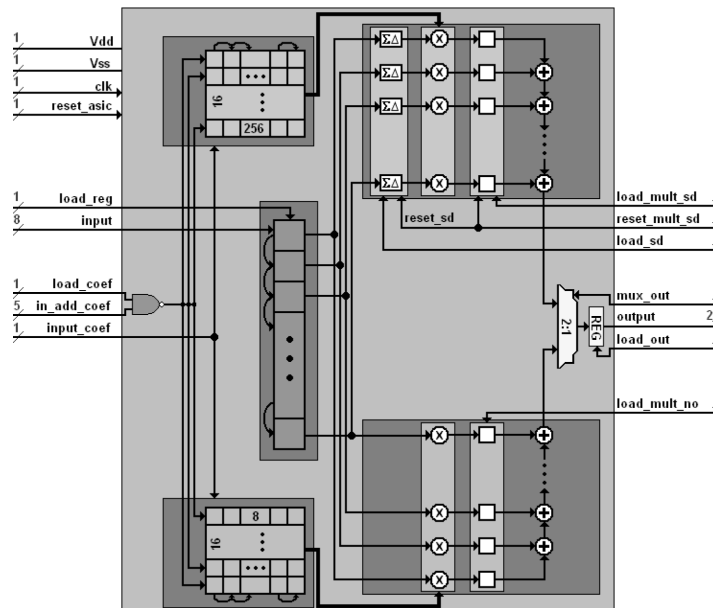


Figure 47: Proposed circuit to be developed using a 0.35μm technology.

Each filter receives its coefficient through a multiplexer, thus an 8-bit input signal is injected in the circuits to be processed. Each output filtered point is compared to a previously fault-free generated signal to realize if there is any difference. If so, coefficients are reinserted, and the whole processing restarts. By comparing the output values, one can see how far they are from a fault-free exemplar. One does not know in which part of the circuit the fault occurred, but just if it occurred in one circuit or another. A VHDL description of the circuitry was done, and through Cadence® tools a final mask was obtained.

The floor plan view, that is, the final top cell view just before the placement stage is presented in figure 48. As noted, the multiplier for the sigma-delta filter is about 9 times larger than that of the classical filter. This is due to the way the multiplier was designed. To

obtain a considerable performance, OSR XNOR gates were used to make the multiplication (see section 2.3.3.1 for details).

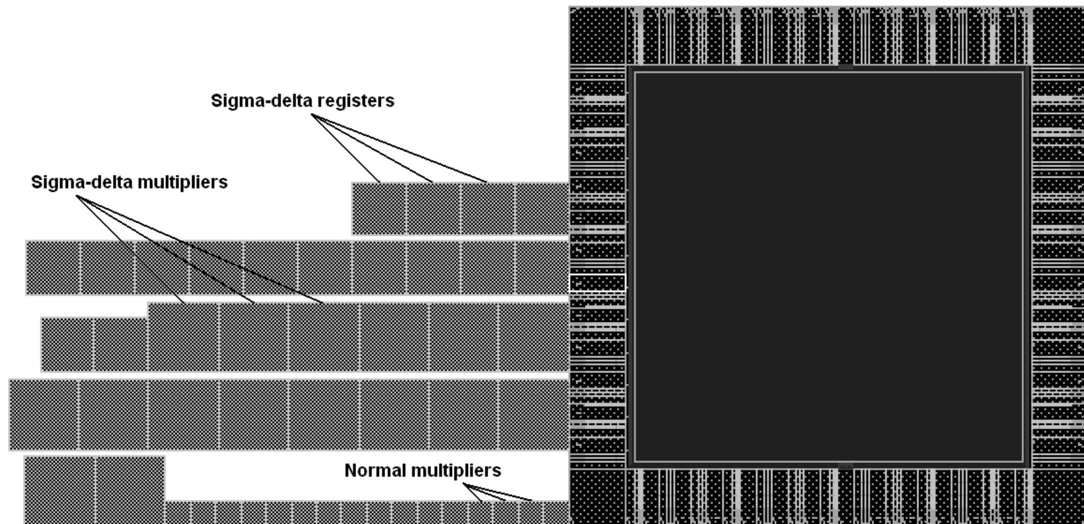


Figure 48: Top cell view generated through Encounter®.

Also, there is not the interleaving step after the multiplication stage. Instead, the XNOR outputs are already accumulated in order to generate one decimal filtered point. The mask obtained after the placement stage, adding the pads and power supply lines has about 2.4x2.4mm, and is presented in figure 49.

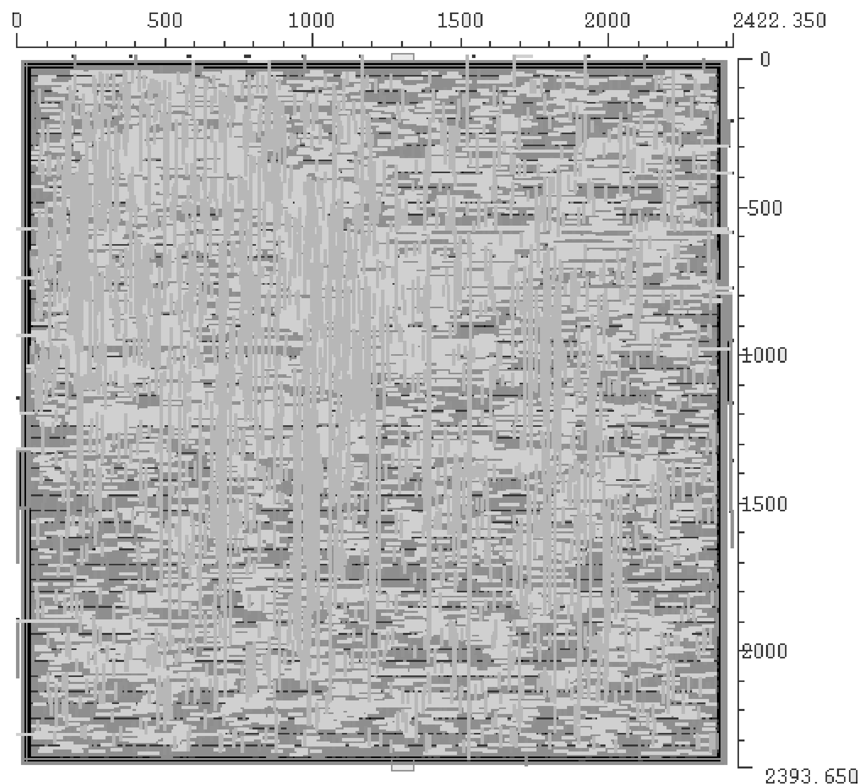


Figure 49: Final mask to be used in the fabrication process.

A total of 27876 gates were used, being 27832 (99.8%) by the core itself and 44 (0.2%) by the IO interface, corresponding to a final cell area of 6054330.200 pm^2. The circuit dissipates a static (leakage) power of about 3.5µW, an internal power (caused by the charging of internal loads) of about 63mW and a switching power (determined by the capacitive load and frequency of the logic transitions) of about 137mW. A 3.3V power supply and a 50MHz operation frequency are previewed for the prototype.

2.4 CONCLUSIONS

Based on the idea of error tolerant applications and redundancy, a new proposition in the development of digital fault-tolerant circuits was presented. Instead of trying to mitigate the SEU occurrence through technology enhancement, or to correct the circuit response through costly hardware or software redundancy, the use of signal redundancy was proposed, in such a way that, even if multiple faults occur, the system response can still sustain a good resolution, able to produce correct responses.

The use of digital sigma-delta modulators to convert an n-bit PCM word into an OSR-bit sigma-delta bit stream was demonstrated, and its fault tolerance was evaluated. As demonstrated, even under many faults occurrence, just by increasing the system over sampling ratio, it is possible to enhance the system robustness, obtaining betters resolutions and more precise responses.

To demonstrate the technique feasibility, different case studies were developed, culminating in the realization of a Digital Signal Processor (DSP) using VHDL description and in the development of a chip for further radiation tests. The DSP, specially designed to process sigma-delta modulated values, was programmed to implement a 16 taps FIR filter and, as demonstrated, even under the occurrence of multiple faults, the system response matches very closely the fault-free behavior. When compared to standard n-bit architecture,

the DSP using sigma-delta modulated signals not only presents a much higher fault tolerance, but can also present better performance. Of course, when higher robustness is required, a trade off is established, thus the performance penalty must be increased as well. In the area comparison, for a FPGA implementation, both solutions presented an equivalent area, but if memory structures must be taken into account, for larger memory space, the solution with sigma-delta processing will certainly present a larger area. Also, results for an IIR filter were presented, but due to the extremely high sensitivity presented by this kind of filter, other topologies must be studied in order to implement a more error-tolerant version for the structure.

It is important to differentiate the two different ways to make the sigma-delta signal processing: for the discrete case, an exact result can be obtained, but the area and performance parameters become an important limiting factor; for the continuous modulation scheme, faster and much smaller circuits can be made, but a difference between the sigma-delta and a classical implementation becomes evident, although much satisfactory results can still be obtained.

3 ANALOG VOTER TO COPE WITH SET IN TMR APPROACHES

The impact of a charged particle on a MOS circuit has been extensively analyzed [CHA, 1993]. An ionization in a certain part of a circuit may cause a transient current pulse, which, when propagated through a combinatorial logic, gets the denomination of Single Event Transient (SET). The current pulse width depends on several technological parameters, and may be modeled by a double-exponential current pulse [MESSENGER, 1992], given by:

$$I(t) = I_0\left(e^{\frac{-t}{\tau_a}} - e^{\frac{-t}{\tau_b}}\right) \quad (25)$$

where I_0 is approximately the maximum charge collection current, τ_a is the collection time constant of the junction and τ_b is the time constant for initially establishing the ion track. The I_0 parameter depends on the energetic particle Linear Energy Transfer (LET) value and process parameters. An example of how the current pulse amplitude and duration vary with the particle LET can be seen in figure 50, taken from [DODD, 2004], where four different technologies were used to simulate a SET.

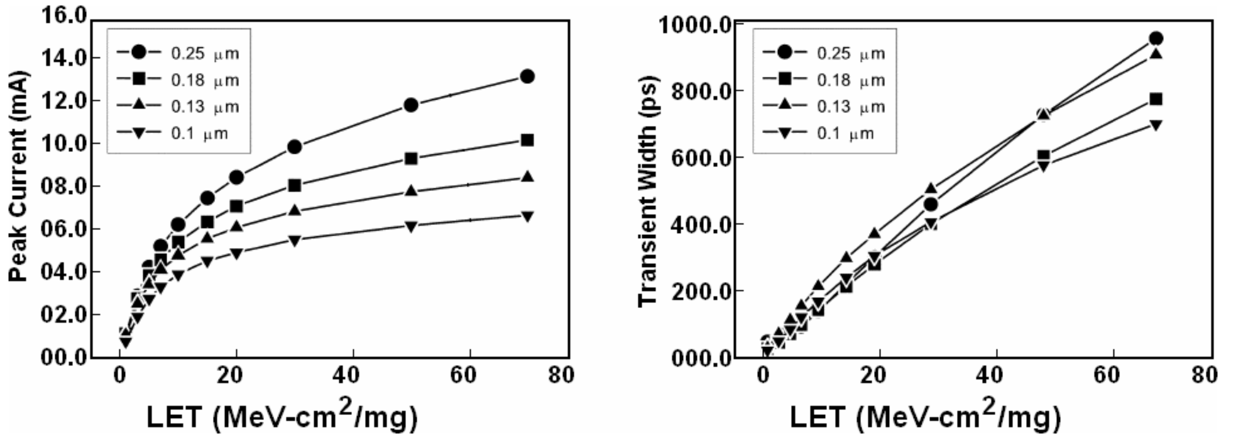


Figure 50: Peak transient current and transient width as a function of strike LET and technology scaling for bulk technology.

With the advance of CMOS technology, faster clocks become possible, with clock frequencies reaching hundreds of gigahertz, thus with clock cycles possibly smaller then the SET duration itself. This can lead to a situation where the SET can possibly be stored in a flip-flop, thus generating an error in one or more bits.

To cope with the SET problem, different solutions have been proposed. In [REBAUDENGO, 2002], two low-cost solutions to cope with SEU are compared: the error-detection capabilities of a hardware-implemented solution based on parity code and software-implemented solution based on source-level code modification. In [MONGKOLKACHIT, 2003], all latch inputs are designated as critical nodes, and for each latch input, an additional circuit is inserted between combinational circuit output and latch input. In [LIMA, 2003], a combination of Duplication With Comparison (DWC) and Concurrent Error Detection (CED) based on time redundancy is used to detect permanent faults in the programmable matrix of SRAM-based FPGAs . The implementation of a new soft error tolerance technique based on time redundancy and another based on space and time redundancy is presented in [ANGHEL, 2000]. Perhaps the most popular solution is the use of hardware redundancy, known as Triple Modular Redundancy (TMR) [CHANDE, 1989], where fault tolerance in a hardware component is improved through the triplication of this component and further vote among the outputs of these components to determine the correct result, as demonstrated in figure 51. The vote can be done by counting the number of correct outputs or by selecting the mean value [GAITANIS, 1988], [NANDURI, 1990].

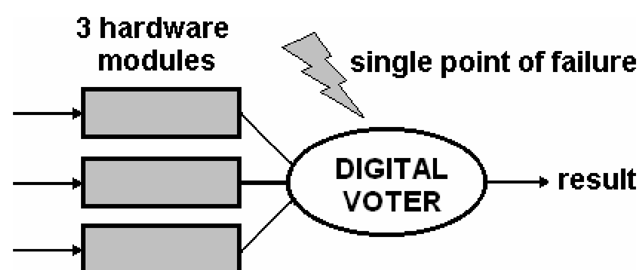


Figure 51: The TMR approach: voter may become an unreliable point.

Is interesting to note that the voter does not determine which module suffered the fault, but only if the fault occurred or not. As it will be demonstrated in section 3.1, although simple, the voter is the critical point regarding fault-tolerance in the TMR approach. If the voter presents low reliability, the whole system will be fragile, as much as the voter itself. Some solutions to increase the fault-tolerance of the voter include triplicating the voter.

However, this circuit, itself, would be a TMR and subject to the same weakness, therefore just adding area to the whole device and having the same handicap. Another solution consists in designing voters capable of testing themselves on-line with respect to their possible internal faults [METRA, 1997] [CAZEAUX, 2004]. Besides the reliability issue, other important characteristics for a voter are speed, area overhead, power consumption, circuit complexity. In section 3.2 we propose a different way to vote the TMR block output. Instead of using a traditional digital voter, analog voters are used, which, as will be demonstrated, are more reliable than their digital counterparts regarding SET occurrence.

SPICE simulations results are presented, comparing classical digital voters with the proposed analog ones. To make the simulations, the same W/L ratio were used for all the transistors (with difference between P and N), even for the digital and the analog voters. For the NMOS transistor a ratio of 6.0/0.6 is used and for the PMOS a ratio of 10.8/0.6 is used.

3.1 Digital Voter

Generally, for the case of replicated modules with *n* outputs, the voter consists of *n* voting circuits operating on a bit-by-bit basis and giving to the output the logic value present on the majority of the corresponding modules' outputs. The correct operation of the TMR system is guaranteed if at least two of the three modules produce correct outputs. Figure 52 presents four different implementation of a classical digital voter, which implement the Boolean expression *AB+AC+BC*, where *A*, *B* and *C* are the TMR outputs.

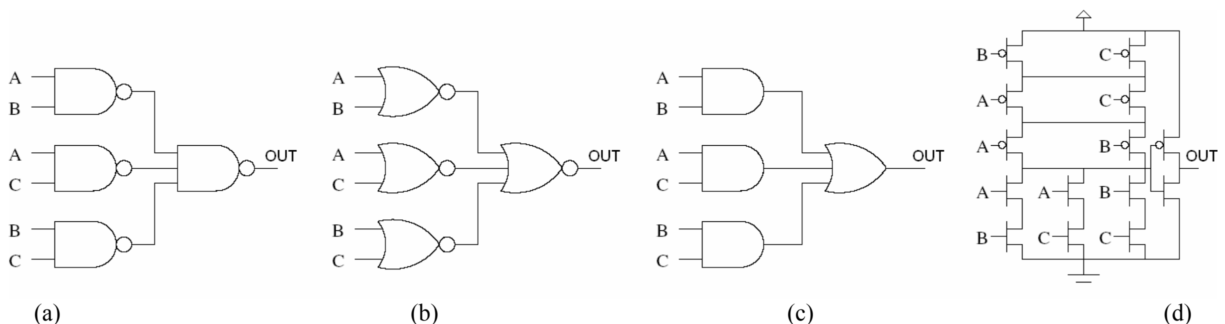


Figure 52: (a) NAND-NAND voter, (b) NOR-NOR voter, (c) AND-OR and (d) conventional CMOS static implementation of a voter for a TMR system.

These four circuits were simulated and the occurrence of SET evaluated. A further comparison to the analog version of the voter is presented in section 3.2. To model a SET occurrence, a current source is introduced at the node in the circuit where the particle hit occurs. The direction of current flow resulting from the single event strike depends upon the polarity of the transistor. If the transistor is an NMOS transistor, the current flows from the drain to the body. If it is a PMOS transistor, current flows from the body to the drain [CHA, 1993]. By studying the SET occurrence process, one can understand why this consideration. Let one observe the situation presented in figure 53, where a section of a NMOS transistor is represented. As seen, the PN junction formed by the drain and the bulk is reversed biased, thus no current flows through this junction. However, when a particle hits the transistor drain, the reversed biased junction collects the charge, and a current flows from the N to the P part, thus generating a SET, as represented by the current source in figure 53. This current source is modeled by expression (25). The same reasoning can be used to understand the process for the PMOS.

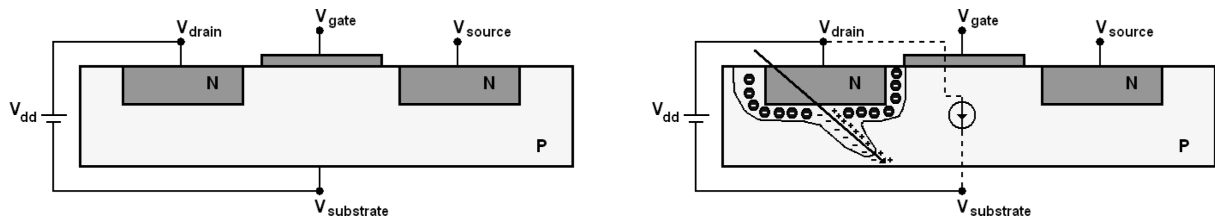


Figure 53: NMOS section showing a particle hit in a reversed biased PN junction.

As mentioned before, a SET is modeled as a double exponential [MESSENGER, 1992], but a simpler model can also be used. In [LEVEUGLE, 2004], it is proposed a model for the current spike, similar to the voltage pulse model used for SET. Figure 54(a) illustrates this model and shows the main parameters used to model de double exponential curve: pulse amplitude (*A*), rising time (*Tr*), falling time (*Tf*) and pulse width (*W*). The parameter values can be derived from the classical double exponential model, as shown in figure 54(b).

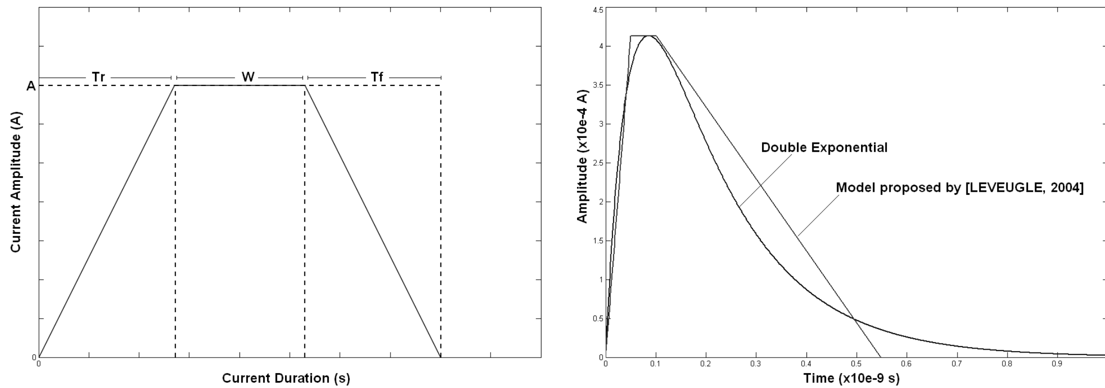


Figure 54: (a) Proposed model for the double exponential and (b) comparison between the model and the double exponential curve.

According to [LEVEUGLE, 2004], although this fault model remains at very low level, it can be used to perform injections on structural nodes in the high-level description of an analog block, by superposition of the current spike with the normal current at the target node.

3.1.1 Simulations Results

SPICE simulations were carried out to verify whether the circuit is tolerant or not to SET occurrences. During the SET incidence in the different points of the circuits, four possible events were analyzed: no alteration is detected in the voter output (N), an inversion from 'high' to 'low' state is detected (0), an inversion from 'low' to 'high' state is detected (1) or an undefined (U) state occurs (the spike amplitude is higher than the 'low' and lower than 'high' decision voltage or lower than 'high' and higher than 'low' decision voltage). The faults were injected in each node for all possible input vectors in the inverters inputs.

Figure 55 illustrates the sensitive nodes (drains) of the simulated circuits: 2-inputs NAND (figure 55(a)) and 3-inputs OR (figure 55(a)), 2-inputs NOR and 3-inputs NOR (figure 55(b)), 2-inputs NAND and 3-inputs NAND (figure 55(c)), and conventional CMOS voter (figure 55(d)). Where a NMOS and a PMOS drains form a node, two current sources are included to simulate de SET.

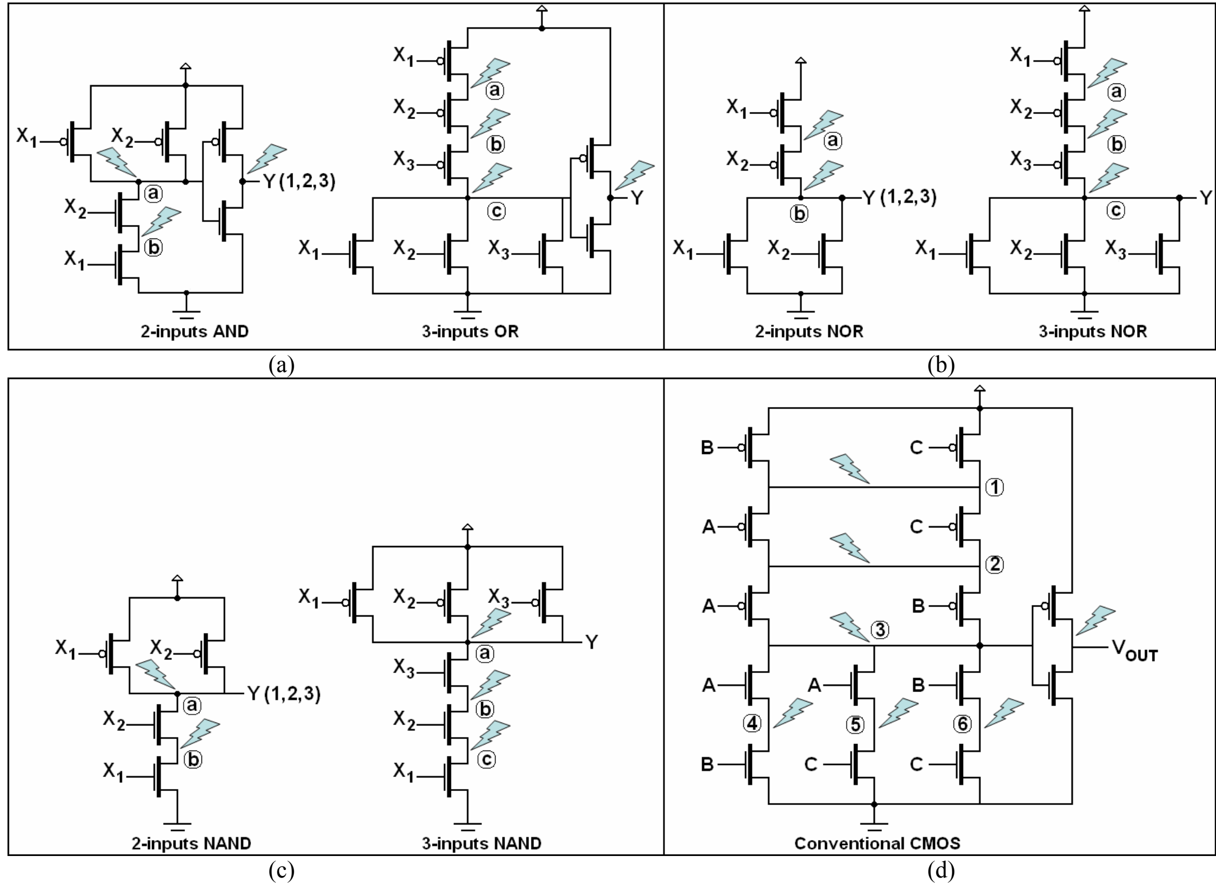


Figure 55: Simulated circuits for SET evaluation in a digital TMR voter.

The values of τ_a and τ_b in equation (25) are, respectively, 1.64e-10s and 5.0e-11s, taken from [CHA, 1993], which give approximate values to *Tr*, *Tf* and *W* of 50ps, 100ps and 0.5ns, respectively. The value of I_0 can vary due to different factors: the angle at which the injection occurs, that is, the angle of incidence of the particle [REED, 1994], the particle that hits the circuit, the technology used in the circuit, and even the node where the particle hits. Typically, this value goes from 0.1mA to 10mA. In the following simulations, four different values of I_0 were used, giving four different values for *A*: $I_0 = 1.5mA$ $(A = 0.6mA)$, $I_0 = 2.0mA$ $(A = 0.8mA)$, $I_0 = 2.5mA$ $(A = 1mA)$ and $I_0 = 3.0mA$ $(A = 1.2mA)$.

Table 6 summarizes the results from the SPICE simulations, showing the different nodes where the current was injected, and if this injection cause any of the situations described before.

Table 6: SET consequences in different nodes of digital voters.

ANDs + OR					NANDs					NORs					CMOS				
Node	0.6	0.8	1.0	1.2	Node	0.6	0.8	1.0	1.2	Node	0.6	0.8	1.0	1.2	Node	0.6	0.8	1.0	1.2
1	N	N	N	N	1	N	N	N	N	1	1	1	1	1/U	1	N	N	N	U
1a	N	N	N	N	1a	N	N	N	N	1a	1	U	1	U	2	N	U	U	1
1b	N	N	N	N	2	N	N	N	N	2	1	1	1	1/U	3	N	U	1/U	1/U
2	N	N	N	N	2a	N	N	N	N	2a	N	N	N	N	4	N	N	N	N
2a	N	N	N	N	3	N	U	1	1	3	1	1	1	1/U	5	N	N	N	N
2b	N	N	N	N	3a	N	N	N	N	3a	N	N	N	U	6	N	N	N	N
3	N	N	N	N	4a	U	U	U	U	4a	N	N	U	U	OUT	N	U	U	U
3a	N	N	1	1	4b	N	N	U	U	4b	U	U	U	U					
3b	N	N	N	N	OUT	U	U	U	U	OUT	U	1/U	0/1/U	0/1/U					
4a	N	N	N	N															
4b	N	N	N	U															
4c	N	U	1/U	1/U															
OUT	N	U	U	U															

Analyzing the results from table 6, one can see that all the voters have one or more node susceptible to problems when particles hit them. The best results are obtained for the voter using NAND gates, where there is only 1 node in 18 where any inversion is detected, from ‘low’ to ‘high’ state. The worst case is observed for the circuit with NOR gates, which presents 5 nodes in 18 sensible to SET.

3.2 Analog Voter [SCHÜLER, 2005d] [LISBÔA, 2005]

Trying to increase the robustness of the TMR voter, an analog circuit is proposed. The idea is that, since a quiescent current is constantly flowing through the circuit, higher energy level particles should hit the transistors drains in order to cause a transient event. The analog voter structure is presented in figure 56(a). It is in fact a mixed-signal voter, since the voter input is composed of three inverters with their output short circuited and fed into an analog circuit, which compares the values of the resulting voltage in the inverters outputs, and decides whether the voter output must be ‘1’ or ‘0’.

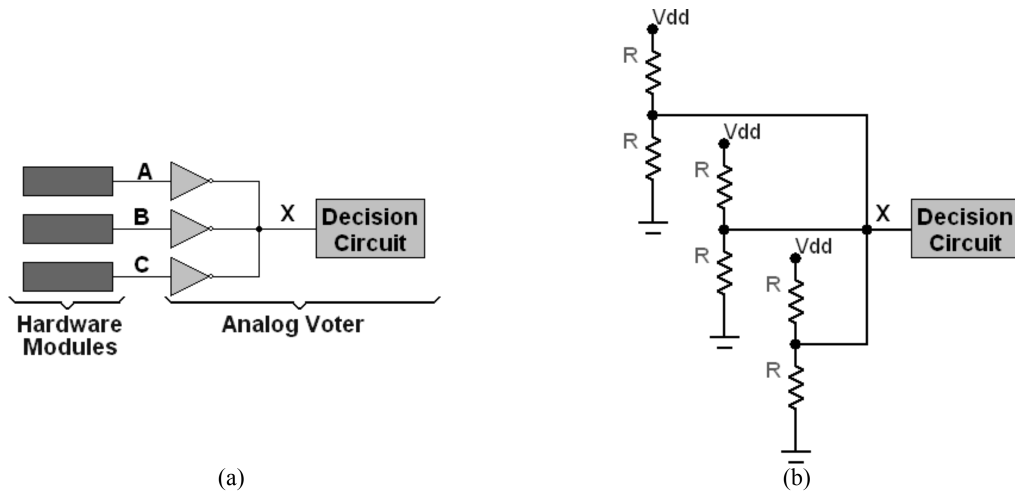


Figure 56: (a) Analog voter and (b) equivalent electrical circuit.

Figure 56(b) represents the equivalent electrical circuit of the inverters. If the transistors of the inverters are carefully designed to have the same resistance when at ON state, the voltage at the point *X* can be easily calculated, as depicted in figure 57.

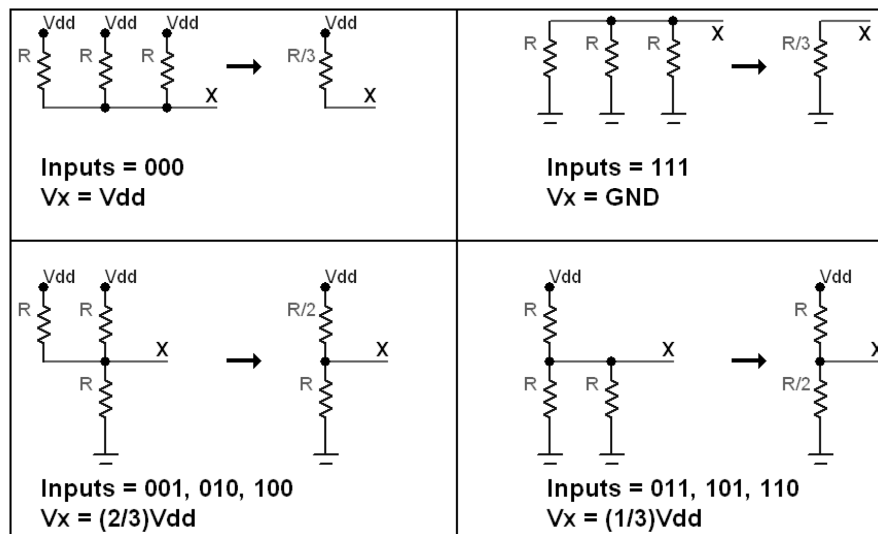


Figure 57: Calculating the short circuited inverters output voltage for all possible input situations.

Once the output voltage is defined by the input bits, it is now a matter of interpreting these voltages and generate digital values from them. To do this, different circuits were tested, and the SET occurrence evaluated to each one.

3.2.1 Simulations Results

Figure 58 represents the four different circuits tested to make the final decision in the analog voter, and their respective sensible nodes. The most evident circuit to be used is a

comparator, since the voltage at point *X* can be compared to a reference voltage and decide whether the comparator output is a '0' or a '1'. The idea to add minimum area overhead to the circuit leads us to the use of a very simple comparator, as the one presented in figure 58(a). It is basically a differential amplifier using a current-mirror load and a current-source controlled by a bias voltage. Since this current-source represents an extra sensible point into the circuit, it was also tested the same comparator, without the current source, as represented in figure 58(b). The third studied option is the one depicted in figure 58(c). It is a push-pull inverter, which simply gives a gain to the *X* point voltage, transforming the different values in two distinct digital states. Finally, the last tested circuit is a Schmitt-trigger, whose hole is similar to the inverter, that is, to restore the digital signal.

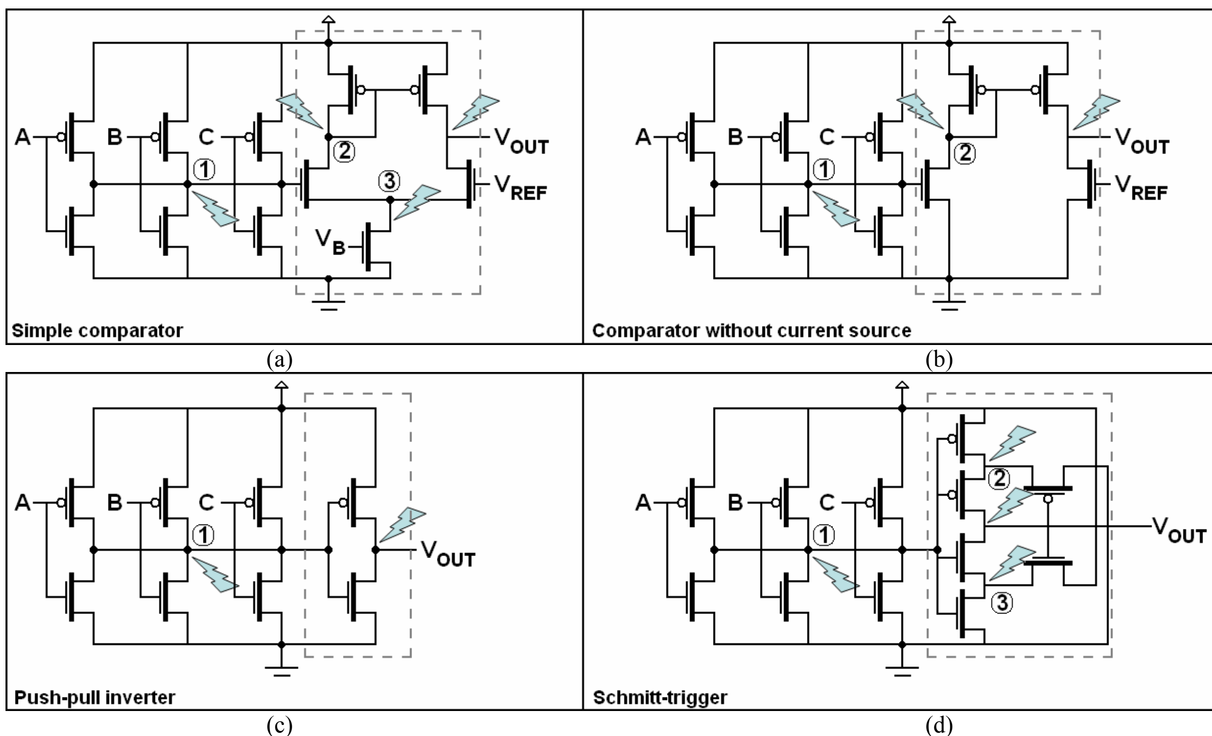


Figure 58: Simulated circuits for SET evaluation in an analog TMR voter.

These four circuits were tested exactly in the same way the digital voters were. As already mentioned, the W/L ratios are the same, just differentiating from the NMOS to the PMOS. Different amplitude SETs were injected through the use of current sources, and the output analyzed to see if there were errors. Table 7 summarizes the results from the SPICE

simulations, showing the different nodes where the current was injected, and if this injection causes any disturbance.

Table 7: SET consequences in different nodes of analog voters.

3 Inv + Comparator					3 Inv + Comparator -I					3 Inv + Schmitt trigger					3 Inv + Inv Push-Pull				
Node	0.6	0.8	1.0	1.2	Node	0.6	0.8	1.0	1.2	Node	0.6	0.8	1.0	1.2	Node	0.6	0.8	1.0	1.2
1	N	N	N	N	1	N	N	N	N	1	N	N	N	N	1	N	U	U	U
2	U	U	U	U	2	N	N	U	U	2	U	U	U	U	OUT	U	U	U	U
3	U	U	U	U	OUT	0/U	0/U	0/U	0/U	3	N	N	U	U					
OUT	0/U	0/U	0/U	0/1/U						OUT	0/U	0/1/U	0/1/U	0/1/U					

Looking at table 7 it is possible to identify one circuit that does not suffer interference from the injected currents: the push-pull inverter. For the comparators (with and without current source) and the Schmitt-trigger, only one node is sensible to SETs (with currents lower than 1.2mA, at least). To compare the results from the digital and the analog voters, it is presented in table 8 the differences between the proposed analog circuits and their digital counterparts. The number of protected drains were counted for each circuit under the different amplitudes of the SET current. Two important thinks are notable in table 8: first, the largest analog voter has less sensible drains than the smaller digital voter, thus the probability that any drain in the analog circuit be hit by a particle is smaller than in the digital one; second, to the highest energy SET simulated, the ratio between the number of protected drains and the number of sensible drains in the analog voter is higher than in the digital, except for the voter with NAND gates, but the area overhead of this solution is also larger than any other in the analog voter set.

Table 8: Summary of simulations with the digital and the analog voters under SET occurrences.

Analog Voter	# sensible drains	# protected drains @600uA	# protected drains @800uA	# protected drains @1000uA	# protected drains @1200uA
3 Inv + Comparator	5P + 6N (11)	4P + 5N (9/11)	4P + 5N (9/11)	4P + 5N (9/11)	4P + 5N (9/11)
3 Inv + Comparator -I	5P + 5N (10)	4P + 4N (8/10)	4P + 4N (8/10)	4P + 4N (8/10)	4P + 4N (8/10)
3 Inv + Schmitt Trigger	5P + 5N (10)	4P + 4N (8/10)	4P + 4N (8/10)	4P + 4N (8/10)	4P + 4N (8/10)
3 Inv + Inv Push-Pull	4P + 4N (8)	4P + 4N (8/8)	4P + 4N (8/8)	4P + 4N (8/8)	4P + 4N (8/8)
Digital Voter	**# sensible drains**	**# protected drains @600uA**	**# protected drains @800uA**	**# protected drains @1000uA**	**# protected drains @1200uA**
ANDs + OR	13P + 13N (26)	13P + 13N (26/26)	13P + 13N (26/26)	11P + 9N (20/26)	11P + 9N (20/26)
NANDs	9P + 9N (18)	9P + 9N (18/18)	9P + 9N (18/18)	7P + 8N (15/18)	7P + 8N (15/18)
NORs	9P + 9N (18)	5P + 3N (8/18)	5P (5/18)	4P (4/18)	5P (5/18)
CMOS	7P + 7N (14)	7P + 7N (14/14)	7P + 7N (14/14)	5P + 7N (12/14)	3P + 7N (10/14)

Undefined state is considered as protected, since the output state does not change.

To conclude the exposition about digital *versus* analog voters, it is important to say that the proposed analog voters can be expanded to *n*-MR approaches. As already mentioned, there is a consensus in the community that simultaneous occurrences of transient are possible. Therefore, solutions like the n-MR, are currently used to withstand with more than one fault. Once more, all known n-MR approaches suffer from the same drawback explained for TMR, since the voters are also digital circuits subject to the effects of SETs.

Figure 59 shows how an n-MR approach can be made through the use of the proposed analog voter. Just by inserting one inverter for each *n* input, and then using one of the proposed circuits to decide about the voted output.

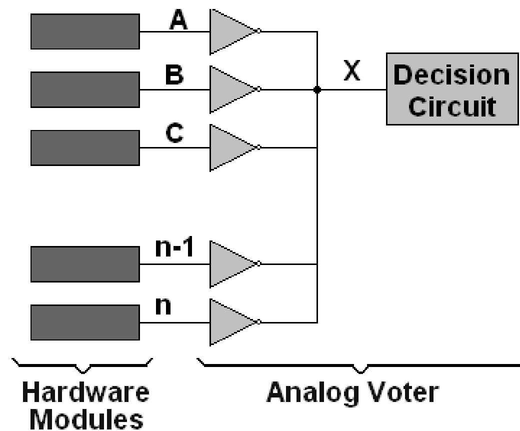


Figure 59: The analog voter being used in a n-MR scheme.

The n-MR scheme using the analog voters was tested under SET occurrences, and the results can be found in table 9. Since now there is a larger set of resulting voltages in the output of the short circuited inverters, only the comparators were used to make the decision whether the n-MR must be ‘0’ or ‘1’. Just by adjusting the reference voltage, this becomes an easy task. Looking at table 9, one can see that the results when using 5 inverters is the same that when using 3 inverters, for both comparators cases.

Table 9: SET consequences in different nodes of analog voters for a n-MR approach.

5 Inv + Comparator					5 Inv + Comparator -I				
Node	0.6	0.8	1.0	1.2	Node	0.6	0.8	1.0	1.2
1	N	N	N	N	1	N	N	N	N
2	U	U	U	U	2	U	U	U	U
3	U	U	U	U	OUT	0/U	0/U	0/U	0/1/U
OUT	0/U	0/1/U	0/1/U	0/1/U					

3.3 CONCLUSIONS

Triple Modular Redundancy schemes have been widely used to improve reliability in critical systems. The idea is to obtain a final response through the majority voting of three responses from three circuits that execute the same task. However, it has been claimed that the voter itself may become a critical point, since one single fault at this circuit will affect the final response.

Trying to increase the reliability to Single Event Transients in TMR schemes, the use of an analog voter was proposed. The idea of using three short circuited inverters and a decision circuit is described and simulations results were presented. To play the role of decision circuit, four different circuits were tested: two comparators, a push-pull inverter and a Schmitt-trigger block. As demonstrated, even with multiple faults injected in different points of these circuits, in different times, in the majority of the situations they remain stable, without propagating the faults to the output of the system, at least for those amplitudes of injected faults. Most important, when compared to digital voters, not only the number of sensible drains is reduced, but more robust results are obtained when injecting the same faults in both circuits.

The proposed idea can be easily extended to an n-MR approach, just through the insertion of more short-circuited inverters in the input of the comparator, and through the adjustment of the reference voltage of the comparator.

4 STATISTICAL SAMPLERS TO COPE WITH COMPONENT VARIABILITY IN ANALOG CIRCUITS

One of the main problems when developing analog filters in VLSI is to achieve high accuracy regarding the cutoff frequency. This is mainly due to the difficulty in obtaining accurate time constants. Small deviations in the resistor or capacitor values may lead to a very high mismatch between the expected and the achieved cutoff frequency. Although switched-capacitor or active-transistor techniques may produce good results [SCHAUMANN, 2001], the cost to use such approaches becomes another limiting factor, and only increases the tester needs. Testing of such filters is also challenging, in the sense that special equipment is required. As a final scenario, yield will also suffer, thus increasing the gap between the digital and analog design.

In this section we present the development of an analog FIR filter, which does not use passive components to tune the cutoff frequency or the quality factor. Instead, the filter coefficients are represented in a digital manner, while the input signals are represented in a bit stream fashion. This way, a digital processing may be used, and the use of expensive analog-to-digital converters is avoided. The impact of this filter architecture on test cost and possible design-for-test techniques are discussed. To increase yield, the use of spare parts is presented, showing that with this technique, a very simple scheme can be used.

4.1 MIXED-SIGNAL CAPACITANCE-INSENSITIVE FILTER [SCHÜLER, 2007B, 2007C]

There are several different ways to implement analog filters. The more classic implementation uses capacitors and resistors (RC) to define the filter parameters [SCHAUMANN, 2001]. For integrated filters, the switched capacitor (SC), i.e. a pair of switches and a capacitor replacing a resistor, is often preferred to increase the accuracy [SCHAUMANN, 2001]. Also, in the active transistor implementation, a transistor polarized

in its linear region plays the role of a resistor [TSIVIDIS, 1986]. Another possible implementation consists of using transconductors, which, in this case, will act as the active part of the filter and will avoid the use of resistors. This implementation is known as Gm-C [NAUTA, 1992].

In these four cases, one will always need to use capacitors and, as already mentioned, although capacitors can be implemented in VLSI systems, they are expensive in terms of area and, most important, the uncertainty on their absolute value is higher than ten percent [JOHNS, 1997]. Among these solutions, the SC implementation is then attractive as a better accuracy can be obtained. Nevertheless, the area overhead and the difficulty to test these circuits both affect SC circuits.

Despite the technology that will be used to design the filter, one can consider its topology. For example, a biquadratic implementation can be done through the use of a Tow-Thomas or a Sallen-Key structure, which can be done through the use of an RC network, a SC, active transistors or even Gm-C.

A simpler way to make high order filters with linear phase is by using a Finite Impulse Response (FIR) structure [OPPENHEIM, 1999]. Typically, FIR filters are implemented in digital systems, either using software or hardware implementation. Digital signal processors are suitable for this kind of filters, since operations like multiplication can be done in a single machine cycle. One of the possible structures for a FIR filter, known as direct form FIR, has already been depicted in figure 31. It is basically an input shift register which will receive each sampled value that will be multiplied by a constant. In the multiplication stage, each shift register output is multiplied by each filter coefficient (b0, b1,…bn). Finally, the multipliers outputs are added to generate one filtered point. Then, the input values are sampled and shifted, and the process restarts.

Although simple, the analog implementation of this filter raises some problems, mainly when switched capacitors are used. In this case, not only the design is difficult due to the elevate number of capacitors, but also the test becomes an important aspect. Since a set of capacitors and switches is used to define the filter coefficients, each of these components must be tested to guarantee a correct response for the filter. Examples of analog FIR filters can be found in [FISCHER, 1990], [DIAZ-SANCHES, 1996], [BURLINGAME, 2000], [CIOTA, 1996] and others.

We now introduce a different way to implement FIR filters, suitable to analog signal processing. This filter does not use capacitors to define its cutoff frequency or its quality factor, and, since the processing is digitally made, it can be easily tested without the need of complex or expensive testers. On the contrary, a single exclusive-or gate and a counter are used to test the analog part, while the digital part can be tested using classical digital testing schemes. Figure 60 presents the filter implementation.

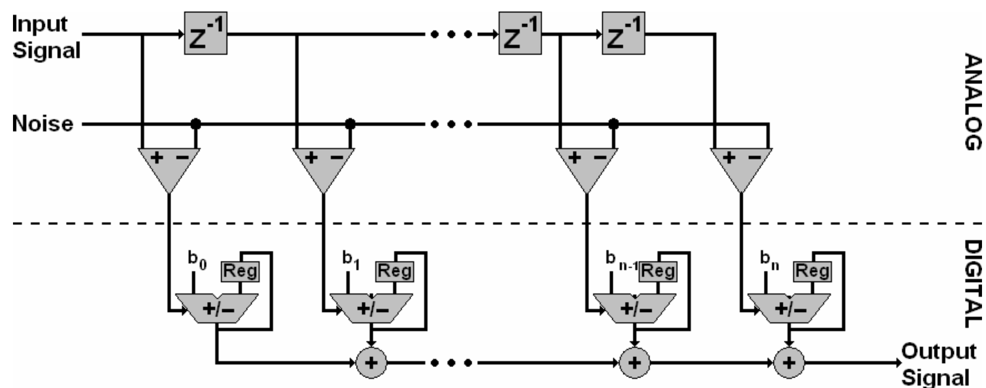


Figure 60: Mixed-signal implementation of a FIR filter.

If one compares it to the structure presented in figure 31, there is a perfect correspondence between the blocks: an analog shift register receives and shifts each sampled value; next block acts as an Analog-to-Digital Converter (ADC), which will generate a bit stream for each sampled value. Up to this point, the input analog signal is represented in a digital fashion through the bit stream, and a digital processing must follow. The next stage, thus, will perform the multiplication between each bit stream and each filter coefficient, and

make the final addition to generate a filtered point. To do this, a very simple way is to decide, for each generated bit stream, whether to add or to subtract the coefficient value, depending on the value of the bit in the stream. The analog shift-register and the statistical sampler will be better analyzed in sections 4.1.1 and 4.1.2.

Figure 61 shows a Matlab® simulation of the proposed filter, whose coefficients implement a FIR with a Kaiser windowing with beta equal to 0.5. The input signal is a 20KHz sine wave added to a white noise with ten times lower amplitude. This signal was sampled with a sampling frequency sixteen times higher than the Nyquist rate, generating 4096 sampled points. These points were filtered by the proposed FIR, and the Fast Fourier Transform of the filter output was calculated.

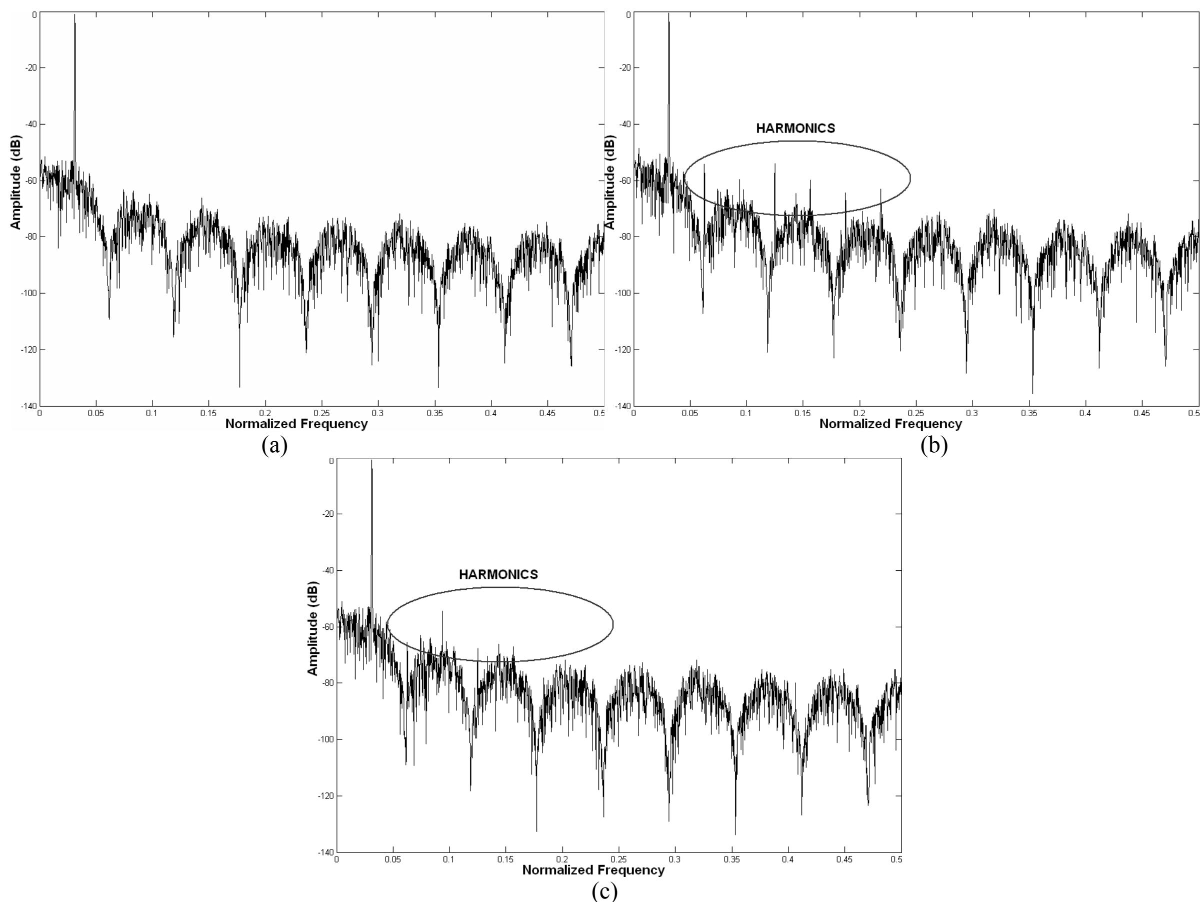


Figure 61: Frequency spectra to compare the responses of (a) a normal filter, (b) the proposed filter with an OSR of 256 and (c) an OSR equal to 1024.

In figure 61(a), the FFT of a normal implementation of the same FIR is plotted in order to compare to the responses of the proposed filter. Figure 61(b), thus, depicts the

simulation for the proposed filter, where each sampled point is compared to a noise generating 256 bits. As it can be seen, some harmonics appear since the same signal is compared many times, thus increasing the correlation and giving rise to the harmonics. However, the amplitude of such harmonics can be reduced by increasing the number of comparisons, as one can note by looking at figure 61(c), where each signal/noise comparison generates 1024 bits. This phenomenon of the reduction in the noise level of the filtered signal can be better seen in figure 62, where different numbers of bits are generated for each simulation, and the difference between a normal filter FFT and the proposed filter FFT is plotted. As noted, going from 256 to 1024 bits, the mean value of the noise level reduces to almost zero, denoting an increase in the signal to noise ratio.

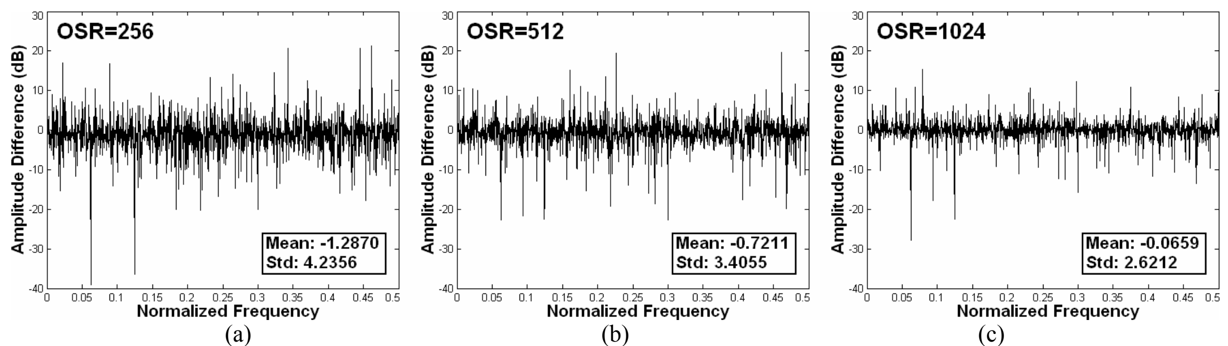


Figure 62: Noise level plotted by making the difference between a FFT of a normal filter and the proposed one using (a) an OSR equal to 256, (b) 512 and (c) 1024.

4.1.1 Input Analog Shift-Register

The analog shift register function is to receive and to shift each sampled value. This block could be implemented in many different ways [FISCHER, 1990] [DIAZ-SANCHES, 1996] [BURLINGAME, 2000], without interfering with the following stages of the filter, since its only function is to hold each sample during the comparison/multiplication process. An example of implementation of an analog shift-register is presented in figure 63. It is composed of capacitors to store the values sampled by the switches. The buffers isolate one stored value from the next one.

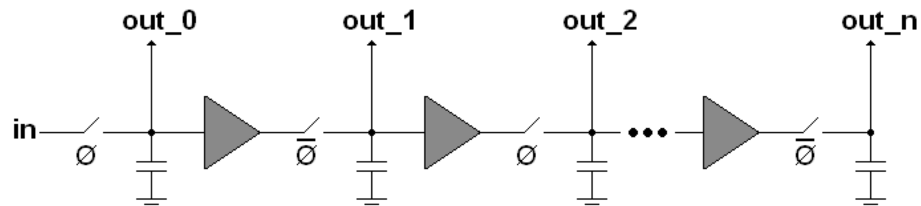


Figure 63: Realization of an analog shift-register.

There are basically three limiting factors in the realization of the analog delay block: loss in resolution due to the shift process, variations in the storage capacitor value and frequency limits. These limitations are analyzed in the next sections.

4.1.1.1 Loss in Resolution due to the Shift Step

In figure 64 it is presented a SPICE simulation showing the loss due to the shift process. A 4Vpp amplitude, 100kHz sine wave is sampled with a sampling frequency of 1MHz. For the given example of analog delay, for each shift step, there is a loss of about 950µV in the stored value. The capacitor values are 2pF.

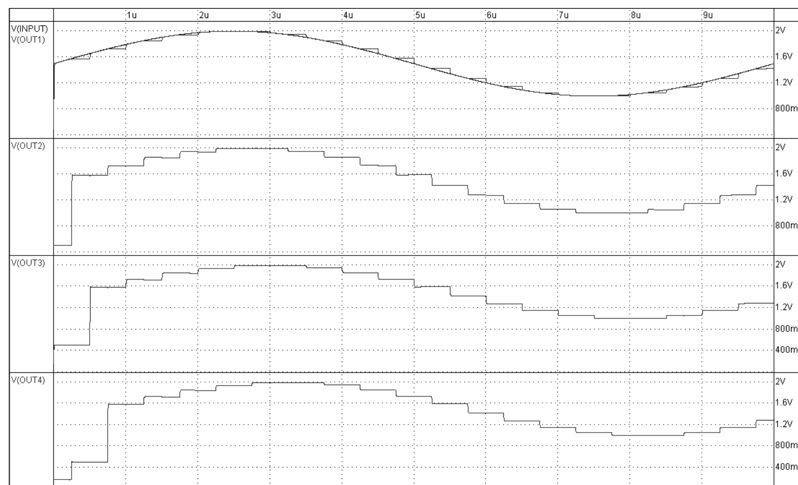


Figure 64: Loss due to the shift process in the analog shift-register.

This loss process has as direct consequence a small resolution in the output filtered signal, but does not interfere in the filter parameters.

4.1.1.2 Loss in Resolution due to Capacitance Variation of the Analog Shift-Register

Although this is supposed to be a capacitance-insensitive filter because its parameters do not depend on capacitors values, the capacitors used in the analog delay block will

certainly have influence in the final performance of the filter. In figure 65 its is shown a simulation where a 50% Monte-Carlo variation is applied to the value of the second capacitor in the shift-register.

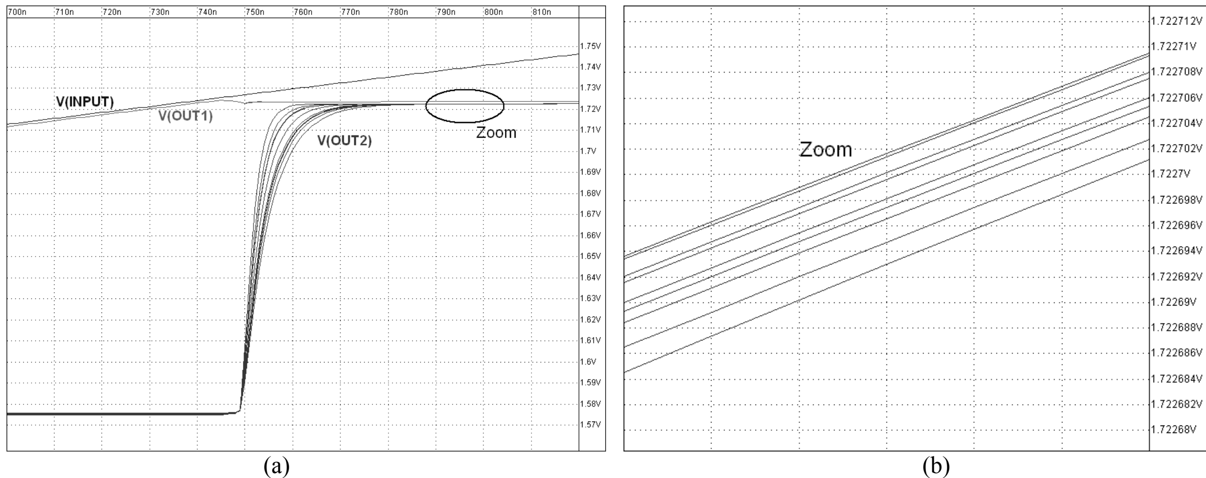


Figure 65: 50% Monte-Carlo variation in the 2nd capacitor value of the analog delay block.

In figure 65(a) one can see that the variation in the capacitor value has consequences in the shifted value mainly before the settle time. After that, the final value stabilizes, and the variation is the 5th decimal digit, as seen in figure 65(b).

4.1.1.3 Frequency Response of the Analog Shift-Register

The last limitation from the analog shift-register regards the high-frequency limit. To understand that, consider the model of one analog-register presented in figure 66. The cut-off frequency determined by the RC circuit is given by:

$$fc = \frac{1}{2.\pi.R.C} \tag{26}$$

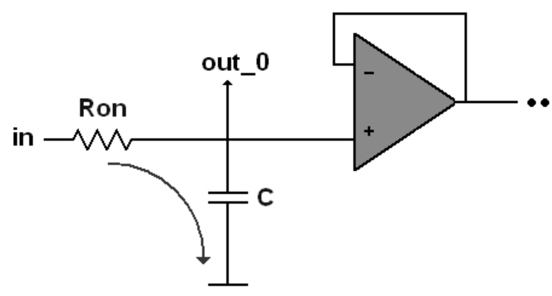


Figure 66: Analog register and its corresponding model.

For small values of the drain-source voltage [LAKER, 1994], the input switch, when in the sampling mode (closed), has a drain-source resistance given by:

$$Rds = \frac{\frac{L}{W}}{\mu.Cox.(Vgs - Vt)} \tag{27}$$

where

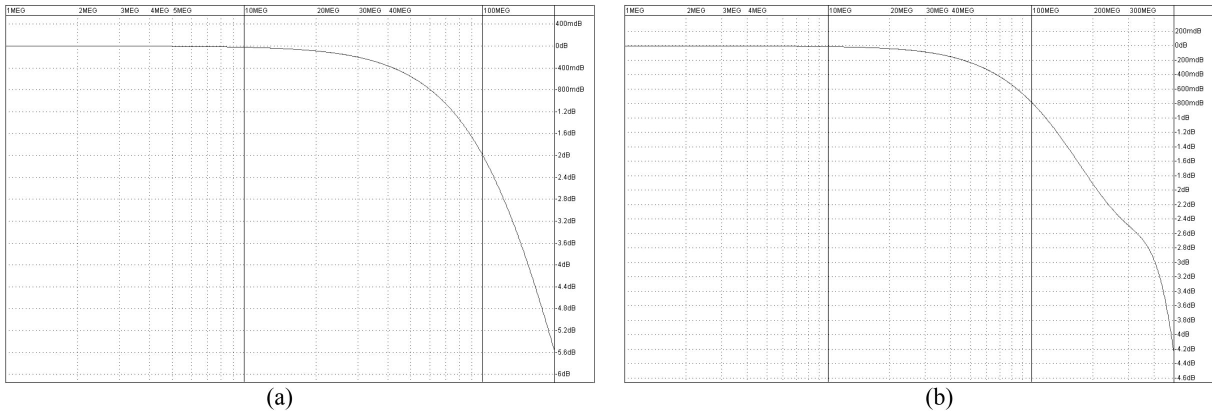
$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{28}$$

$$\theta = \frac{2,3}{t_{ox}[nm]} \tag{29}$$

$$\mu = \frac{\mu_0}{1 + \theta|Vgs - Vt|} \tag{30}$$

In equation (27), (28), (29) and (30), C_{ox} is the oxide capacitance per area, t_{ox} the oxide thickness, ε_{ox} the oxide dielectric constant and μ the carrier's mobility. The constant θ is an empiric value which depends on the oxide thickness, μ_0 the superficial mobility and V_t the transistor threshold voltage [LAKER, 1994].

Considering a 0.35µm technology, a 3.3V power supply and a transistor channel width W=2µm and length L=0.35µm, the R_{on} resistance is of about 560Ω. For a 2pF capacitance, the cut-off frequency is about 140MHz. This value can be confirmed in the SPICE simulation presented in figure 67(a), where the cut-off frequency is130MHz.



(a) (b)

Figure 67: Frequency limit of the input block: (a) due to the RC circuit and (b) due to the operational amplifier.

Although the operational amplifier can also limit the operating frequency, its cut-off frequency will be higher than that given by the RC constant. For the operational amplifier used in this example, its cut-off frequency is in 400MHz, as seen in figure 67(b).

4.1.2 Statistical Sampler

The use of this technique has been proposed in [SOUZA, 2004, 2005]. Here, a brief explanation regarding its functioning is presented. For more details, reader is encouraged to take a look at the reference [SOUZA, 2004, 2005] [NEGREIROS, 2003, 2006] [FLORES, 2002, 2004]. The statistical sampler, presented in figure 68, is based on sampling a signal statistics through stochastic quantization. That is, when the analog signal *S(t)* is compared with a multi-level random reference *N(t)*, the output Probability Distribution Function (PDF) *p(O)* will be the determinate by convolution of their individual PDF, as represented in figure 69. Since the reference bandwidth is much higher than the maximum frequency in the signal, the mean value of the output bit stream will represent the input signal.

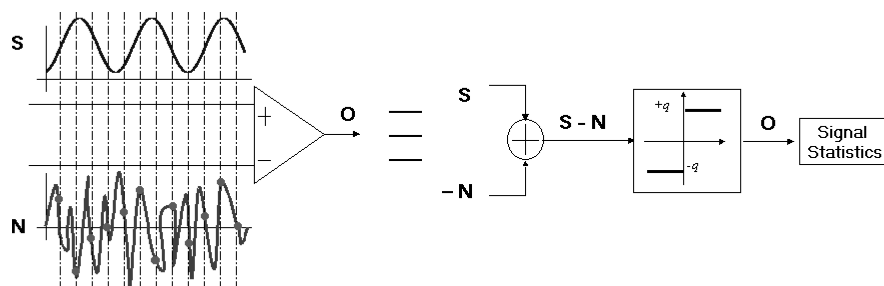


Figure 68: Statistical sampler: a comparator with a noise in the reference [SOUZA, 2004, 2005].

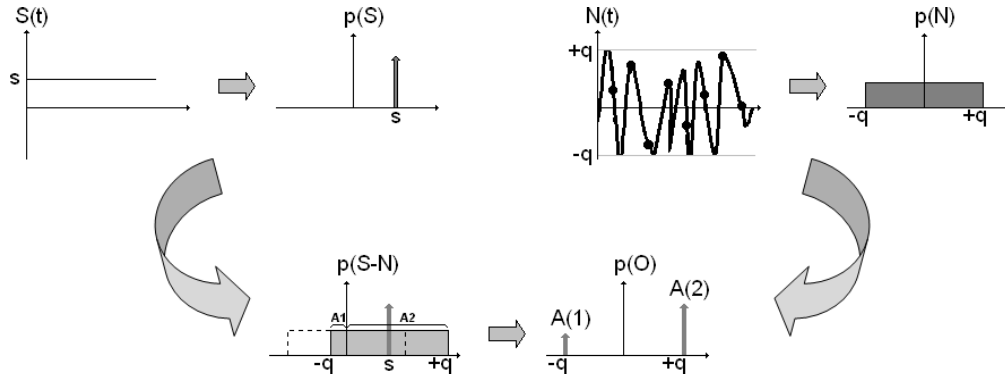


Figure 69: Statistic acquisition with noise dithering [SOUZA, 2004, 2005].

For a uniformly distributed reference a linear relation is established between the input signal and the output bit stream [SOUZA, 2004]. Using any other reference will imply in a non-linear behavior. This can be seen in simulation results presented in figure 70. If the noise used as reference is uniform, since its distribution is linear no harmonic but the signal itself appears, as seen in figure 70(a). But, if a Gaussian noise is used, its non-linear distribution will determine non-linear effects, as seen in figure 70(b).

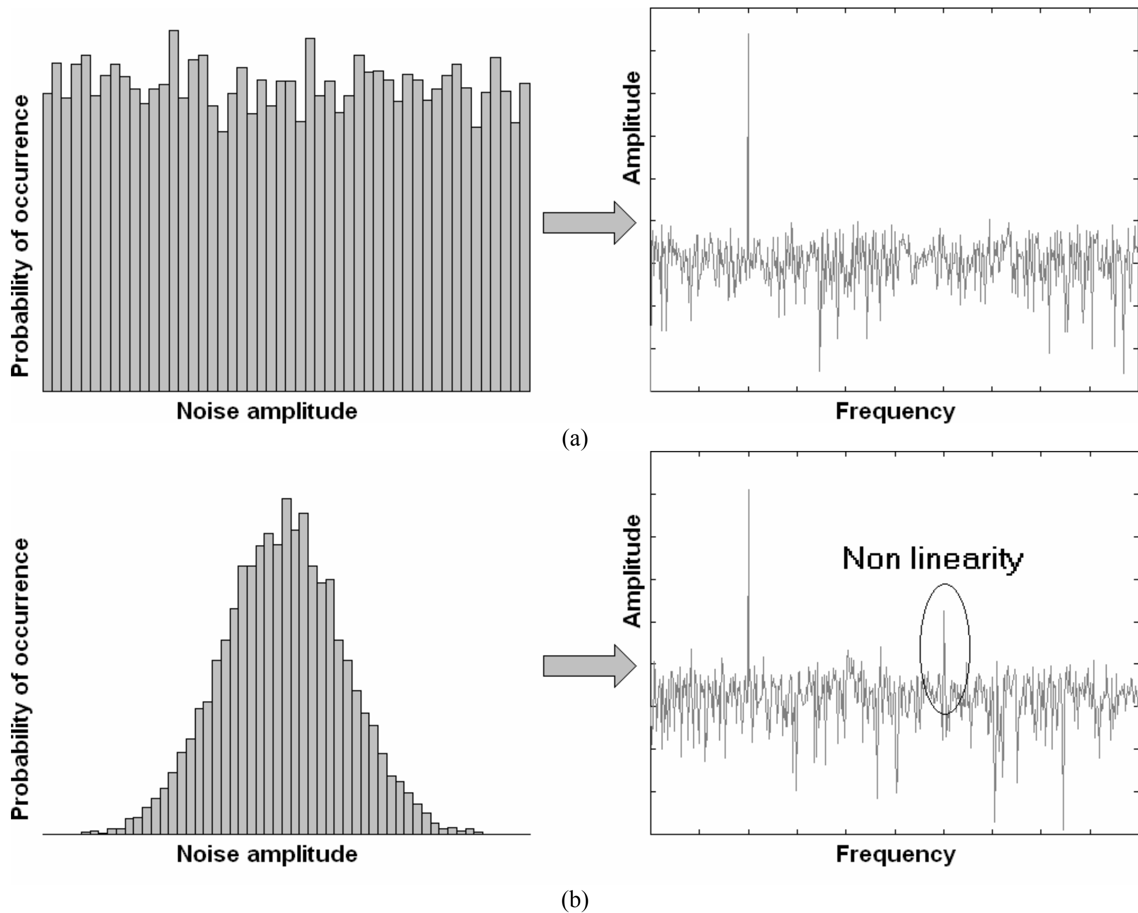


Figure 70: Distribution of a uniform noise without non-linearity generation (a) and distribution of a Gaussian noise with the respective non-linearity expressed by the harmonic (b).

4.1.2.1 Pseudorandom Noise Generation

Generating a uniformly distributed noise is not an easy task. An example of a low-cost solution to this problem has been proposed in [FLORES, 2002], where a pseudo white noise generator is developed. The circuit consists of a switch-controlled RC circuit, which will be randomly sampled through the use of a Linear Feedback Shift Register (LFSR). The idea is summarized in figure 71.

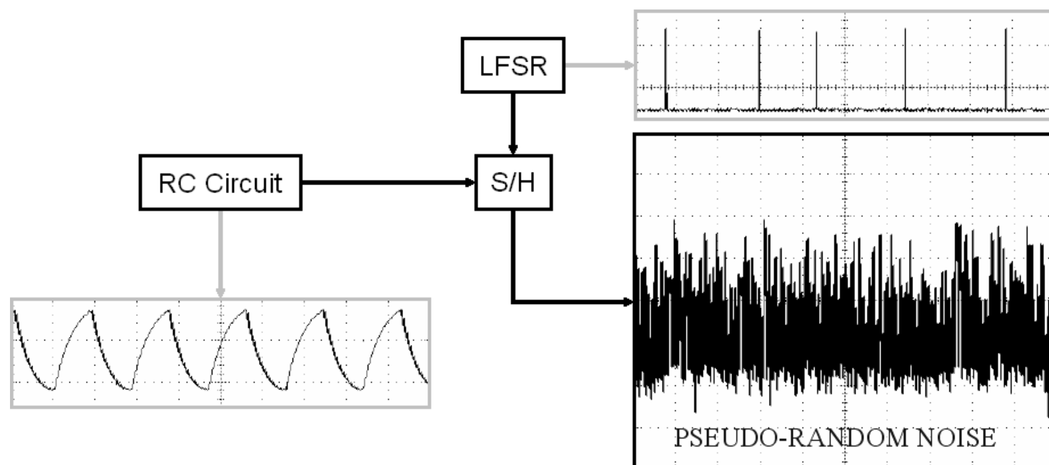


Figure 71: Pseudo white noise generation proposed in [FLORES, 2002].

Analyzing figure 71, the simplicity of the proposed generator is evident. Considering the involved components, a capacitor, a resistor, two switches, a sample and hold and a buffer are responsible for the area overhead. The number of active components is much smaller than those reported in [RENOVELL, 2000] or [BERNARD, 2001].

Once the filter functionality has been presented and its blocks studied, we shall now study how to implement spare parts into the circuit to increase yield. Due to the topology of the proposed filter, one can use extra identical taps for a replacement scheme. Next section presents the testing and replacement approach, and how this idea can be easily implemented just through the use of a set of switches.

4.1.3 Adding Spare Parts: testing and replacement scheme

This section presents how the redundant taps can be added to the proposed filter. The idea is to have many identical taps in such a way that, whenever an operating tap stops working it can be easily replaced by a good one. Also, during fabrication process, yield can be increased since the non-working taps will not determine the discard of the entire circuit.

As presented in figure 60, the proposed circuit can be divided into an analog and in a digital part. This division is also obtained when testing the circuit. It is here assumed that the digital part of the filter may be tested by any existing testing method used to test digital circuits. For the analog part, however, some test steps must be taken. Nevertheless, as it will be seen, testing the analog block can be done in a very easy and cheap way, without any need of extra circuitry.

The analog block of the filter is composed basically of three components: the analog shift cells, implemented through the use of a capacitor and a buffer (see figure 63), a set of switches and comparators, which will compare each sampled point with the noise. The test methodology proposed here consists in testing each analog branch as a whole block, that is, each tap of the FIR will be tested separately, as one single block. This allows one to use the idea of spare parts, that is, extra analog branches should be fabricated in the circuit in such a way that, in the case any analog branch fails, it can be easily replaced by a new one.

Figure 72 depicts how these extra parts can be placed in the circuit. With this configuration, just through some switches manipulations, non-working taps can be replaced by working ones, also allowing the best combination to achieve the best filter response.

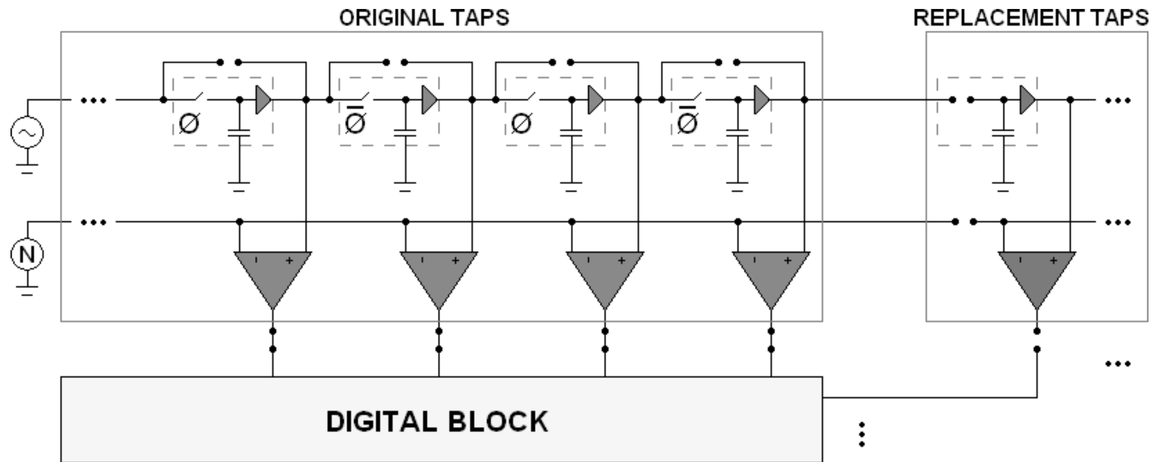


Figure 72: Adding extra taps to the filter: these spare parts can easily replace non-working analog branches.

Let us take, for example, the situation where the second tap of the filter in figure 72 is found to be a non-working one. The substitution scheme consists just in by-passing this tap through one of the switches, and add one of the extra tap, also by manipulating two switches. This situation can be seen in figure 73.

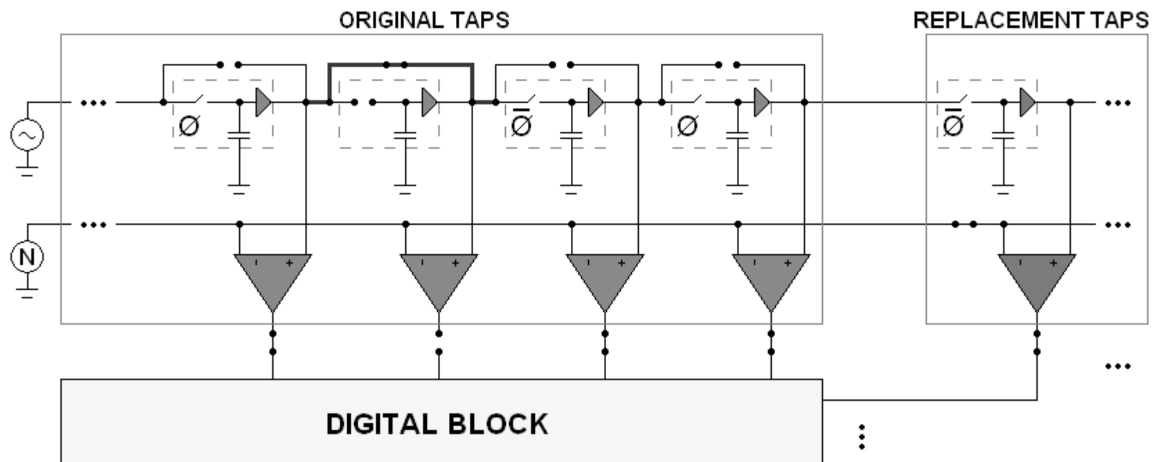


Figure 73: Second tap is replaced by one the existing replacement taps.

We shall now take a look in the way the analog taps are tested in order to find out those taps that must be substituted. Two test examples are given next: test of the analog shift-register buffer gain and test of the comparator off-set voltage. Many others parameters test can be derived from these two basic examples.

4.1.3.1 Testing the Gain of the Analog Shift-Register Buffer

To test the analog shift-register buffer gain, the simplest way is by sampling a certain constant value, generating the correspondent bit stream, shifting the sampled value and generating a new bit stream. Now, just by comparing the counted values of the generated bit streams, it is possible to determine if the buffer gain is inside the desired range. Figure 74(a) and figure 74(b) show the buffer gain test sequence, while figure 75 demonstrates how the difference between the counted values increases with the variation in the buffer gain value.

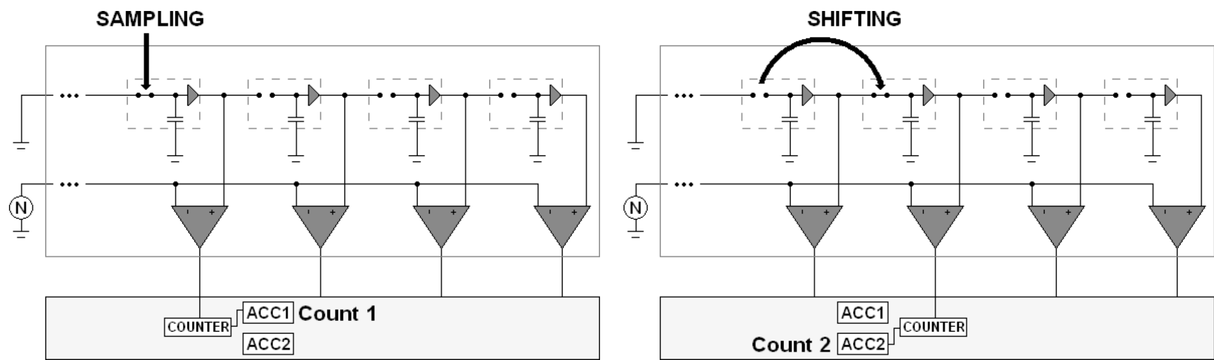


Figure 74: Sequence for testing the analog shift-register buffer gain. Decision is taken by comparing counted values: CASE values ARE "similar" gain is OK; CASE values ARE "different" gain is not OK.

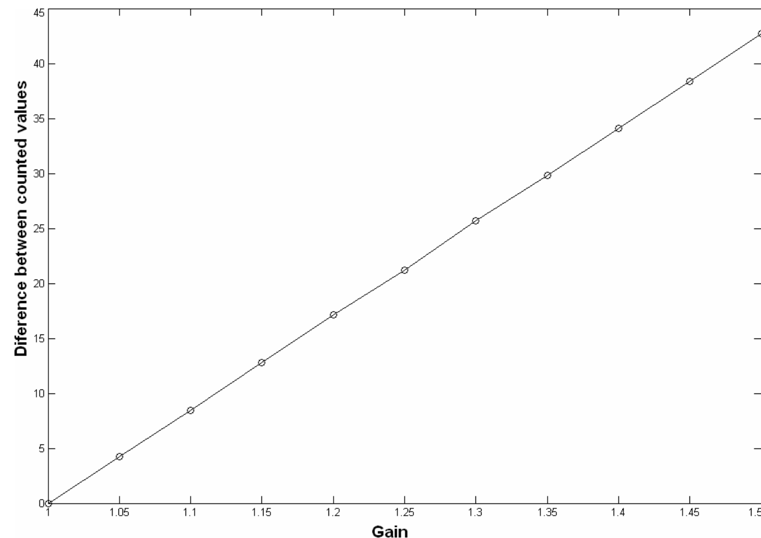


Figure 75: Gain variation versus difference between counted values during the test of the analog shift-register buffer gain.

As seen in figure 75, as the buffer gain varies from 1 to 1.5, the difference between the counted values increases linearly from 0 to 45, thus allowing detecting any considerable variation in the analog shift-register buffer gain.

4.1.3.2 Testing the Off-Set Voltage of the Statistical Sampler

The test of the off-set voltage of the comparators can be done in a similar way. First, a given value is sampled by two analog registers. Then, each comparator will generate its respective bit stream. These bit streams are XORed, and the result is counted in order to determine if the off-set voltage is inside the expected range. Figure 76 shows how this test can be done, and figure 77 how the counted value varies as the off-set voltage increases.

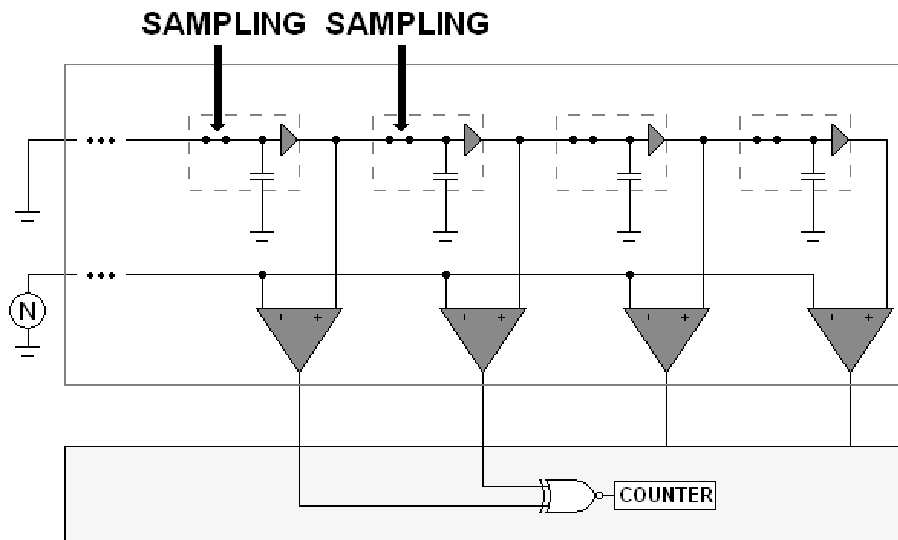


Figure 76: Sequence for testing the comparators off-set voltage. Decision is taken by comparing counted value: CASE value IS “small”, off-set voltage is OK; CASE value IS “large”, off-set voltage is not OK.

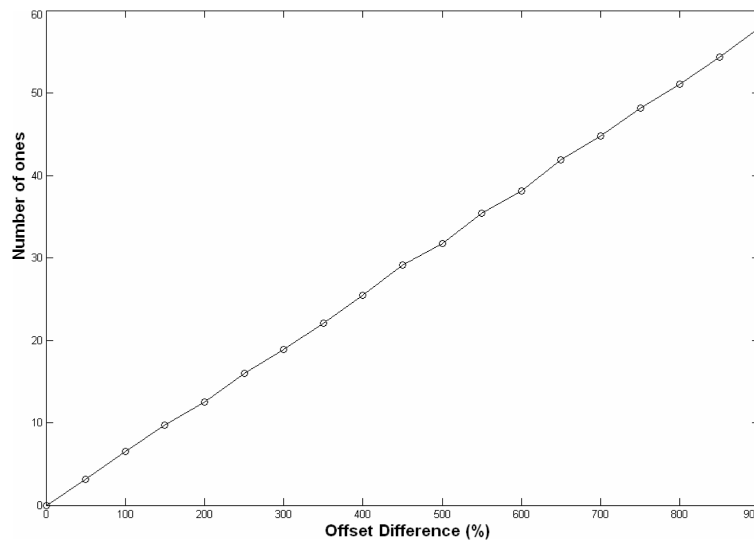


Figure 77: Off-set voltage variation versus difference counted value during the test of the analog comparator off-set voltage.

Once more one can see in figure 77 that, as the percentage of the offset difference between two comparators increases from 0% to 900%, the number of ones in the generated bit

streams also increases from 0 to 60, in a linear fashion. By using the same technique of counting and comparing different bit streams, others parameters can be measured in the filter, thus defining whether the analog branch has enough conditions to be characterized as a good or a bad branch.

4.2 CONCLUSIONS

Digital CMOS technology has been continuously shrinking, and the analog design must follow this trend in order to keep the yield and the integration density, thus not letting the gap between digital and analog design increase even more. As a consequence, analog design gets harder to design, mainly for components that have a high dependence with the technology variability.

Specifically, analog filter design presents, in most cases, problems related to capacitance or transistors mismatch, which can lead to a great discrepancy between the desired and the achieved cutoff frequency, for example. Another problem regarding this kind of filter is the cost to test it. In many cases, extra hardware must be added to the design, and different signals must be generated in order to assure good fault coverage.

It was presented here the implementation of a mixed-signal FIR filter, which does not use passive components to determine its parameters (cutoff frequency, quality factor and gain). Instead, the input signal is represented in a bit stream fashion through successive comparisons of the sampled analog input signal to a random noise. Since the signal to be filtered and the filter coefficients are in the digital domain, all the rest of the processing, including multiplication and addition, is done digitally.

A test scheme was presented, and it was shown that it is possible to test both the analog and the digital part, without the insertion of extra hardware, and without generating any extra test signal. A few switches setting put the system in the test configuration, which

uses only exclusive-or gates and counters. Also, we have presented the idea of extend the concept of spare-parts to the analog domain. Due to the circuit structure, find the non-working taps became a very easy task, such as the replacement scheme, which is no more than a simple switch manipulation task.

5 FINAL REMARKS

Three years have been dispensed in the realization of this work since the first meeting at the beginning of the doctoral period in September 2004, being one of them spent in France, between March 2006 and February 2007. Lots of works have been done in three different tasks, but all in the same area: fault tolerance.

The most developed works, the use of sigma-delta modulation to cope with SEU in digital systems presents a new paradigm in the treatment of soft faults. Instead of using hardware or software redundancy, a signal redundancy approach is proposed. The idea of using only LSB representation of a binary word leads to a more robust scenario that, together with the error tolerance idea, can generate multiple faults robust systems. Different situations were demonstrated, from arithmetic operations to DSP microprocessors, showing that the gain in robustness may overcome the penalties in area and performance.

The second developed work is related to the use of analog circuits to vote the output of digital TMR schemes. The idea behind this intent is that due to a constant current flow in the analog voter, it would be less susceptible to the occurrence of SET, what was in fact demonstrated in the simulations. Beyond this, area overhead was shown not to be a problem when using such analog circuits, proportioning, thus, more robust voters.

Finally, a third work was introduced, presenting the use of statistical sampling to be used in mixed-signals circuits in order to reduce the use of analog passive components. Since no RC circuit is used to determine, for example, the filter cut-off frequency, more precise systems can be achieved. Moreover, it was shown that, due to the circuit structure, a very easy test and replacement scheme can be used, thus increasing yield of these circuits.

Seventeen contributions grown from these three years work, and are described next. Certainly others will come.

5.1 CONTRIBUTIONS

In this section we describe the main contributions of this thesis. Three different fronts were developed during this work to deal with fault tolerance and variability in future technologies: sigma-delta signal processing to cope with SEU in digital circuits; analog voters to cope with SET in TMR approaches; statistic signal acquisition to increase analog yield and reduce analog component variability. We summarize the contributions for each of these themes in the following.

The use of sigma-delta modulation was developed in section 2, and allows the increase of fault tolerance in digital circuits due to bit-flips caused by different sources. The published contributions are:

- Schüler, E., Carro, L., *Increasing Fault Tolerance To Multiple Upsets Using Digital Sigma-Delta Modulators*. **11th International On-Line Testing Symposium (IOLTS'05).** Saint-Raphael, France, 2005. July 06-08, 2005. Pages: 255 – 259.
- Schüler, E., Carro, L., *Sigma-Delta Modulators on the Design of Reliable Digital Circuits*. **11th International Mixed-Signal Testing Workshop (IMSTW'05).** France, 2005. June 27-29, 2005. Volume: 1. Pages: 306 – 310.
- Schüler, E., Carro, L., *Reliable Digital Circuits using Sigma-Delta Modulators*. **20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05).** Monterey, USA, 2005. Oct. 03-05, 2005.
- Farenzena, D.S., Schüler, E., Carro, L., *Aumentando a Tolerância a Falhas de Circuitos Digitais através da Modulação Sigma-Delta*. **XVII Salão de Iniciação Científica**. Porto Alegre, Brasil. Out. 17 – 21, 2005.

- Schüler, E., Farenzena, D.S., Carro, L. "*Multiple simultaneous upset fault-tolerant FIR circuit design using delta-sigma modulation*". **University Booth DATE 2006**. Munich, Germany. Mar. 06 – 10, 2006.
- Schüler, E., Farenzena, D.S., Carro, L. "*Evaluating Sigma-Delta modulated signals to develop fault-tolerant circuits*". **11th IEEE European Test Symposium (ETS'06)**. Southampton, United Kingdom. May 21 – 25, 2006.
- Schüler, E., Erigson, M.I., Farenzena, D.S., Carro, L. "*Fault tolerant DSP microprocessor for ΣΔ-modulated signals*". **2nd Workshop on System Effects of Logic Soft Errors (SELSE 2)**. University of Illinois at Urbana-Champaign. Apr. 11 – 12, 2006.
- Schüler, E., Carro, L., "*Increasing reliability in future technologies systems*". **7th IEEE Latin-American Test Workshop (LATW'06).** Buenos Aires, Argentine. Mar. 26 – 29, 2006. pp. 181-185.
- Schüler, E., Farenzena, D.S., Carro, L. "*On the use of higher-order ΣΔ-modulators for reliable digital circuits design*". **12th IEEE International Mixed-Signals Testing Workshop (IMSTW'06)**. Edinburgh, United Kingdom. Jun. 21 – 23, 2006. pp. 97-101.
- Schüler, E., Erigson, M.I., Carro, L. "*Functionally Fault-Tolerant DSP Microprocessor using Sigma-Delta Modulated Signals*". **Journal of Electronic Testing: Theory and Applications (JETTA 2007)**. Springer Science & Business Media, LLC. USA, 2007.

The use of analog voters was developed in section 3, and analyses the fault tolerance in TMR approaches due to transients caused by particle hits. The published contributions are:

- Schüler, E., Carro, L., *Reliable Digital Circuits Design using Analog Components*. **11th International Mixed-Signal Testing Workshop (IMSTW'05).** Cannes, France, 2005. June 27-29, 2005. Volume:1. Pages: 166 – 170.

- Lisbôa, C. A., Schüler, E., Carro, L., *Going Beyond TMR for Protection Against Multiple Faults*. **18th Symposium on Integrated Circuits and Systems Design (SBCCI'05).** Brazil, 2005. Sept. 04-07, 2005.

Finally, the use of the statistical samplers was developed in section 4avoid the use of analog components in analog circuits, allowing also an yield increase. The published contributions are:

- Schüler, E., Negreiros, M., Nouet, P., Carro, L. "*A Digitally Testable Capacitance-Insensitive Mixed-Signal Filter*". **12th IEEE European Test Symposium (ETS'07)**. Freiburg, Germany. May 20 – 24, 2007.
- Schüler, E., de Souza Júnior, A.A., Carro, L. "*Spare Parts in Analog Circuits: a Filter Example*". **22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT 2007)**. Rome, Italy. Sept. 26 – 28, 2007.

Beyond these related contributions, some extra parallel works were also developed during the doctoral period, generating three others productions:

- Lopes, C. D., Schüler, E., Engel, P. M., Susin, A. A., *ERP signal identification of Individuals at Risk for Alcoholism using Learning Vector Quantization Network*. **2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'05)**. San Diego, USA, 2005. Nov. 14 – 15, 2005.
- Schüler, E., Carro, L., "*Increasing Analog Programmability in SoCs*". **13th Reconfigurable Architectures Workshop (RAW 2006)**. Rhodes Island, Greece. Apr. 25 – 26, 2006.

- Lubaszewski, M., Balen, T., Schüler, E., Carro, L., Huertas, J.L., "Effects of Radiation on Analog and Mixed-Signal Circuits". **Radiation Effects on Embedded Systems**. Book chapter, Springer, 2007. pp.89 – 119.

5.2 FUTURE WORKS

Although a considerable effort has been dispensed during the development of this work, some gaps must still be filled. The first proposed solution of using sigma-delta modulation to obtain reliable digital circuits regarding SEUs, although being the most developed theme, still lacks more applications with complex signal-processing approaches in order to evaluate the robustness of the system when a whole application must be developed, and compare the sigma-delta implementation of these applications with their classical digital counterparts. For example, it is proposed in [JUNI, 1995] the implementation of a sigma-delta architecture for adaptive LMS algorithms, while [FUJISAKA, 2002] presents an entire QPSK demodulator using a sigma-delta bit stream.

For the second proposed solution of using analog voters in TMR systems, more robust simulations should be used to carefully differentiate the robustness of the analog and the digital voter regarding SETs. Although the model used for the SET occurrence can give an idea about the reliability of the circuit, in order to completely understand the transient event consequences for both circuits, a complex device simulation should be carried out in every points of the circuits.

Finally, in the use of the statistical sampler instead of passive analog devices, again new applications should be implemented using the same technique. To demonstrate that the use of this approach can be extended to more analog systems, others circuits like ADC converters, PLL, VCO, etc should be adapted to the use of statistical samplers, and the test easiness and yield increase proved.

REFERENCES

ANALOG DEVICES. ADSP2100: Family Users Manual. **Datasheet**. Available at <http://www.physics.otago.ac.nz/internal/ELEC401/ADSP2100/adsp2101.html>. Last access: Sept., 2007.

ANGHEL, L.; ALEXANDRESCU, D.; NICOLAIDIS, M. Evaluation of soft error tolerance technique based on time and/or space redundancy. In: IEEE SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN, 13., Brazil, 2000. **Proceedings…** Manaus, Brazil : IEEE Press, Sept., 2000, p. 237 – 242.

BASEDAU, P.; QIUTING, H. A post processing method for reducing substrate coupling in mixed-signal integrated circuits. In: SYMPOSIUM ON VLSI CIRCUITS, Japan, 1995. **Proceedings…**Kyoto, Japan : IEEE Press, Jun., 1995, p. 41 – 42.

BERNARD, S. et al. Efficient on-chip generator for linear histogram BIST of ADCs. In: IEEE INTERNATIONAL MIXED-SIGNAL TESTING WORKSHOP, USA, 2001. **Proceedings…**Atlanta, USA : IEEE Press, Jun., 2001, v.1, p. 86-96.

BOSER, B.E.; WOOLEY, B.A. The design of sigma-delta modulation analog-to-digital converters. **IEEE Journal of Solid-State Circuits**. New York : IEEE Computer Society, v.23, n.6, Dec. 1988.

BURLINGAME, E.; SPENCER, R. An analog CMOS high-speed continuous-time FIR filter. In: EUROPEAN SOLID-STATE CIRCUITS CONFERENCE, 26., Sweden, 2000. **Proceedings…** Stockholm, Sweden : IEEE Press, Sept., 2000.

CANDY, J.C.; TEMES, G.C. **Over sampling delta-sigma data converters:** theory, design, and simulation. New York: IEEE Press, c1992. xii, 499 p. : il.

CANDY, J.C.; HUYNH, A. Double interpolation for digital-to-analog conversion. **IEEE Transactions on Communications**. New York : IEEE Computer Society, v.Com-34, n.1, Jan., 1986.

CAZEAUX, J.M.; ROSSI, D.; METRA, C. New high speed CMOS self-checking voter. In: IEEE INTERNATIONAL ON-LINE TESTING SYMPOSIUM, 10., Portugal, 2004. **Proceedings…**Madeira Island, Portugal : IEEE Press, Jul., 2004.

CHA, H.; PATEL, J. A logic-level model for alpha-particle hits in CMOS circuits. In: INTERNATIONAL CONFERENCE ON COMPUTER DESIGN, USA, 1993. **Proceedings…**Santa Clara, USA : IEEE Press, Oct., 1993, p.538 – 542.

CHANDE, P.K., RAMANI, A.K., SHARMA, P.C. Modular TMR multiprocessor system. **IEEE Transactions on Industrial Electronics**. New York : IEEE Computer Society, v.36, n.1, Feb.,1989.

CHAO, K.C.H.; NADEEM, S.; LEE, W.; SODINI, C.G. A Higher order topology for interpolative modulators for over sampling A/D converters. **IEEE Transactions on Circuits and Systems**. New York : IEEE Computer Society, v.37, n.3, Mar., 1990.

CHEN, K.J.; MAEZAWA, K.; YAMAMOTO, M. InP-based high-performance monostable-bistable transition logic elements (MOBILE's) using integrated multiple-input resonant-tunneling devices. **IEEE Electron Device Letters**. New York : IEEE Computer Society, p.127 – 129, Mar., 1996.

CIOTA, Z.; NAPIERALSKI, A.; NOULLET, J.L. Analogue realization of integrated FIR filters. **IEE Proceedings Circuits, Devices and Systems**. New York : IEEE Computer Society, v.143, i.5, p.274 – 281, Oct., 1996.

CROCHIERE, R.E.; RABINER, L.R. Interpolation and decimation of digital signals: a tutorial review. **Proceedings…** New York : IEEE Computer Society, v.69, n.3, Mar., 1981.

DIAS DA FONTE, V. Sigma-delta signal processing. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, England, 1994. **Proceedings…**London, England : IEEE Press, May, 1994, v.5, p. 421 – 424.

DIAZ-SANCHEZ, A.; RAMIRIZ-ANGULO, J. VLSI voltage mode FIR analog adaptive filters. In: IEEE MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS, 39., USA, 1996. **Proceedings…**Iowa, USA : IEEE Press, Aug., 1996, v.1, p.141 – 144.

DODD, P.E.; SHANEYFELT, M.R.; FELIX, J.A.; SCHWANK, J.R. Production and propagation of single-event transients in high-speed digital logic ICs. **Transactions on IEEE Nuclear Science**. New York : IEEE Computer Society, v.51, n.6, Dec., 2004.

ERCEGOVAC, M.D.; LANG, T.; MORENO, J. H. **Introduction to digital systems**. USA : John Wiley & Sons, 1998. ISBN:0471527998.

FISCHER, G. Analog FIR filters by switched-capacitor techniques. **IEEE Transactions on Circuits and Systems**. New York : IEEE Computer Society, v.37, n.6, p.808 – 814, Jun., 1990.

FLORES, M.G.C.; NEGREIROS, M.; CARRO, L.; SUSIN, A.A. A noise generator for analog-to-digital converter testing. In: SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN, 15., Brazil, 2002. **Proceedings…**Porto Alegre, Brazil : IEEE Press, Sept., 2002.

FLORES, M.G.C.; NEGREIROS, M.; CARRO, L. INL and DNL estimation based on noise for ADC test. **Transactions on IEEE Instrumentation and Measurement**. New York : IEEE Computer Society, v.53, n.5, Oct., 2004.

FUJISAKA, H.; KURATA, R.; SAKAMOTO, M.; MORISUE, M. Bit-stream signal processing and its application to communication systems. **IEE Proceedings Circuits, Devices and Systems**. New York : IEEE Computer Society, v.149, n.3, Jun., 2002.

GAITANIS, N. The design of totally self-checking TMR fault-tolerant systems. **IEEE Transactions on Computers**. New York : IEEE Computer Society, v.37, i.11, p.1450 – 1454, Nov., 1988.

GUPTA, S.K.; BREUER, M.A.; MAK, T.M. Defect and error tolerance in the presence of massive numbers of defects. **IEEE Design & Test of Computers**. New York : IEEE Computer Society, v.21, i.3, p.216 – 227, May, 2004.

HADLEY, P., SET Logic. In: WORKSHOP ON INNOVATIVE CIRCUITS AND SYSTEMS FOR NANO ELECTRONICS, 2., Netherlands, 1997. **Proceedings…**Delft, Netherlands : IEEE Press, Sept. 1997, p.29 – 30.

HAZEGHI, A.; KRISHNAMOHAN, T.; WONG, H.-S.P. Schottky-barrier carbon nanotube field-effect transistor modeling. **IEEE Transactions on Electron Devices**. New York : IEEE Computer Society, v.54, i.3, p.439 – 445, Mar., 2007.

HUANG, K.H.; ABRAHAM, J.A. Algorithm-based fault tolerance for matrix operations. **IEEE Transactions Computers**. New York : IEEE Computer Society, v.33, p.518 – 528, 1984.

IBM. SOI Technology: IBM's Next advance in chip design. Available at http://www-03.ibm.com/chips/about/technology/technologies/soi/.Last access: Oct., 2005.

ITRS, Semiconductor Industry Association. International Technology Roadmap for Semiconductors - ITRS 2006. Available at <http://www.itrs.net/Links/2006Update/2006UpdateFinal.htm>. Last access Sept. 2007.

JANER, C.L; QUERO, J.M.; ORTEGA, J.G.; FRANQUELO, L.G. Fully parallel stochastic computation architecture. **IEEE Transactions on Signal Processing**. New York : IEEE Computer Society, v.44, n.8, p.2110 – 2117, 1996.

JOHNS, D.; MARTIN, K. **Analog integrated circuit design**. New York : John Wiley & Sons, Inc. 1997, ISBN 0-471-14448-7.

JOHNSON, B.W., Fault tolerance. **The Electrical Engineering Handbook**. New York : CRC Press, p. 2020, 1993.

JUNI, J.; STEWART, R.W. Adaptive signal processing algorithms using sigma delta architectures. In: ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS, 28., USA, 1994. **Proceedings…**Pacific Grove, USA : IEEE Press, Oct., 1994, v.2, p.1530 – 1534.

LAKER, K.R.; SANSEN, W.M.C. **Design of analog integrated circuits and systems**. New York : McGraw Hill, p.898, 1994.

LENT, C.S.; TOUGAW, P.D. A device architecture for computing with quantum dots. **Proceedings of the IEEE.** New York : IEEE Computer Society, p.541 – 557, Apr., 1997.

LESLIE, T.C.; SINGH, B. Sigma-delta modulators with multibit quantizing elements and single-bit feedback. **Proceedings of the IEEE.** New York : IEEE Computer Society, v.139, n.3, Jun., 1992.

LEVEUGLE, R.; AMMARI, A. Early SEU fault injection in digital, analog and mixed signal circuits: a global flow. In: DESIGN, AUTOMATION AND TEST IN EUROPE CONFERENCE AND EXHIBITION, France, 2004. **Proceedings…**Paris, France : IEEE Press, Feb., 2004.

LIMA, F.; CARRO, L.; REIS, R., Designing fault tolerant systems into SRAM-based FPGAs. In: DESIGN AUTOMATION CONFERENCE, USA, 2003. **Proceedings…**California, USA, Jun., 2003, p650 – 655.

LISBÔA, C.A.L.; CARRO, L. An intrinsically robust technique for fault tolerance under multiple upsets. In: IEEE INTERNATIONAL ON-LINE TESTING SYMPOSIUM, 10., Portugal, 2004. **Proceedings…**Madeira Island, Portugal : IEEE Press, p.180, Jul., 2004.

LISBÔA, C.A.L.; SCHÜLER, E.; CARRO, L. Going beyond TMR for protection against multiple faults. In: SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN, 18., 2005, Brazil. **Proceedings…**Florianópolis, Brazil : IEEE Press, p.80 – 85, Sept., 2005.

MACKAY, D.J.C. **Information theory, inference, and learning algorithms**. Available at <http://books.google.com.br/>. Last access, Sept., 2007.

MALOBERTI, F.; O'LEARY, P. Processing of signals in their oversampled delta-sigma domain. In: INTERNATIONAL CONFERENCE ON CIRCUITS AND SYSTEMS, China, 1991. **Proceedings…** Shenzhen, China : IEEE Press, Jun., 1991.

MALOBERTI, F. Non conventional signal processing by the use of sigma delta technique: a tutorial introduction. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, USA, 1992. **Proceedings…**California, USA : IEEE Press, May, 1992, v.6, p.2645 – 2648.

MESSENGER, G.C A summary review of displacement damage from high-energy radiation in silicon semiconductors and semiconductor devices. **IEEE Transactions on Nuclear Science**. New York : IEEE Computer Society, v.39, n.3, p.468 – 473, Jun., 1992.

METRA, C.; FAVALLI, M.; RICCÒ, B. Compact and low power self-checking voting scheme. In: IEEE INTERNATIONAL SYMPOSIUM ON DEFECT AND FAULT TOLERANCE IN VLSI SYSTEMS, France, 1997. **Proceedings…**Paris, France : IEEE Press, p.137 – 145 Oct., 1997.

MONGKOLKACHIT, P.; BHUVA, B. Design technique for mitigation of alpha-particle-induced single-event transients in combinational logic. **Transactions on IEEE Device and Materials Reliability**. New York : IEEE Computer Society, v.3, n.3, Sept., 2003.

MOORE, G.E. Cramming more components onto integrated circuits. **Electronics Letters**. New York : IEEE Computer Society, v.38, n.8, Apr., 1965.

NANDURI, S.R.; EL HAKEEM, A.K.; AL-KHALILI, A.J. Fault analysis of a TMR system using multiple valued logic. In: ANNUAL INTERNATIONAL PHOENIX CONFERENCE ON COMPUTERS AND COMMUNICATIONS, 9., USA : IEEE Press, 1990. **Proceedings…**Phoenix, USA, p.23 – 29, Mar., 1990.

NAUTA, B.A. CMOS Transconductance-C filter technique for very high frequencies. **IEEE Journal of Solid State Circuits**. New York : IEEE Computer Society, v.27, Feb., 1992.

NEGREIROS, M.; SCHÜLER, E.; CARRO, L.; SUSIN, A.A. Testing RF signal paths using spectral analysis and subsampling. In: SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN, 16., Brazil, 2003. **Proceedings…**São Paulo, Brazil : IEEE Press, Sept., 2003.

NEGREIROS, M.; MICHELS, A.M.; CARRO, L.; SUSIN, A.A. Low cost RF signal generation for BIST. In: IEEE INTERNATIONAL MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS, 49., Porto Rico, 2006. **Proceedings…**San Juan, Porto Rico : IEEE Press, v.2, p.648 – 652, Aug., 2006.

NEPAL, K. et al Designing logic circuits for probabilistic computation in the presence of noise. In: IEEE DESIGN AND AUTOMATION CONFERENCE, 42., USA, 2005. **Proceedings…**California, USA : IEEE Press, Jun., 2005.

NORSWORTHY, S.R.; SCHREIER, R.; TEMES, G. C. **Delta-sigma data converters:** theory, design and simulation. New York: IEEE Press, 1997. 476p.

O'LEARY, P.; MALOBERTI, F. Bit stream adder for oversampling coded data. **Electronics Letters**. New York : IEEE Computer Society, v.26, n.20, p.1708 – 1709, Sept., 1990.

OGATA, K., **Modern control engineering**, 2.ed. Englewood Cliff : Prentice Hall, 1994.

OPPENHEIM, A.V. **Discrete-time signal processing**. 2.ed., Upper Saddle River, N.J. : Prentice Hall, c1999. xxiii, 870 p. : il.

PLANK, J.S. A tutorial on reed-solomon coding for fault-tolerance in RAID-like systems. **Technical Report UT-CS-96-332**, University of Tennessee, Jul., 1996.

RABAEY, J.M. **Digital Integrated Circuits:** a design perspective. Upper Saddle River : Prentice Hall, 1996. 702 p.

REBAUDENGO, M.; REORDA, M.S. Evaluating cost and effectiveness of software redundancy techniques for hardware errors detection. In: ANNUAL INTERNATIONAL SYMPOSIUM ON FAULT-TOLERANT COMPUTING, 28., Germany, 1998. **Digest of Fast Abstracts…** Munich, Germany : IEEE Press, Jun., 1998, p.88 – 89.

REBAUDENGO, M. et al Coping with SEUs/SETs in microprocessors by means of low-cost solutions: A Comparison Study. **IEEE Transactions on Nuclear Science.** New York: IEEE Press, v.49, n.3, Jun. 2002.

REED, R.A.; MCNULTY, P.J.; ABDEL-KADER, W.G. Implications of angle of incidence in SEU testing of modern circuits. **IEEE Transactions on Nuclear Science**. New York: IEEE Press, v.41, n.6, Dec., 1994.

RENOVELL, M. et al. Hardware minimization for an histogram-based ADC BIST. In: IEEE VLSI TEST SYMPOSIUM, 18., Canada, 2000. **Proceedings…**Montreal, Canada : IEEE Press, Apr., 2000, p.247 – 252.

ROBERTS, G.W.; LU, A.K. **Analog signal generation for built-in-self-test of mixed-signal integrated circuits**. New York, Kluwer Academic Publishers, 1995. 122p.

ROSSI, D.; OMAÑA, M.; TOMA, F.; METRA, C. Multiple transient faults in logic: an issue for next generation ICs? In: IEEE INTERNATIONAL SYMPOSIUM ON DEFECT AND FAULT TOLERANCE IN VLSI SYSTEMS, 20., USA, 2005. **Proceedings…**California, USA : IEEE Press, Oct., 2005.

SCHAUMANN, R.; VALKENBURG, M.E.V. **Design of analog filters**. New York : Oxford University Press, 2001. 737p.

SCHÜLER, E.; CARRO, L. Increasing fault tolerance to multiple upsets using digital sigma-delta modulators. In: INTERNATIONAL ON-LINE TESTING SYMPOSIUM, 11., France, 2005. **Proceedings…**Saint-Raphael, France : IEEE Press, p.255 – 259, Jul., 2005a.

SCHÜLER, E.; CARRO, L. Sigma-delta modulators on the design of reliable digital circuits. In: INTERNATIONAL MIXED-SIGNAL TESTING WORKSHOP, 11., France, 2005. **Proceedings…**Cannes, France : IEEE Press, v.1, p.306 – 310, Jun., 2005b.

SCHÜLER, E.; CARRO, L. Reliable digital circuits using sigma-delta modulators. In: IEEE INTERNATIONAL SYMPOSIUM ON DEFECT AND FAULT TOLERANCE IN VLSI SYSTEMS, 20., USA, 2005. **Proceedings…** Monterey, USA : IEEE Press, Oct., 2005c.

SCHÜLER, E.; CARRO, L. Reliable digital circuits design using analog components. In: INTERNATIONAL MIXED-SIGNAL TESTING WORKSHOP, 11., France, 2005. **Proceedings…**Cannes, France : IEEE Press, v.1, p.166 – 170, Jun., 2005d.

SCHÜLER, E.; FARENZENA, D.S.; CARRO, L. On the use of higher-order ΣΔ-modulators for reliable digital circuits design. In: IEEE INTERNATIONAL MIXED-SIGNALS TESTING WORKSHOP, 12., United Kingdom, 2006. **Proceedings…**Edinburgh, United Kingdom : IEEE Press, v.1, p. 97 – 101, Jun., 2006a.

SCHÜLER, E.; CARRO, L. Increasing reliability in future technologies systems. In: IEEE LATIN-AMERICAN TEST WORKSHOP, 7., Argentine, 2006. **Proceedings…**Buenos Aires, Argentine : IEEE Press, v.1, p.181 – 185, Mar., 2006b.

SCHÜLER, E.; FARENZENA, D.S.; CARRO, L. Multiple simultaneous upset fault-tolerant FIR circuit design using delta-sigma modulation. In: UNIVERSITY BOOTH DATE, Germany, 2006. **Proceedings…**Munich, Germany : IEEE Press, Mar., 2006c.

SCHÜLER, E.; ERIGSON, M.I.; FARENZENA, D.S.; CARRO, L. Fault tolerant DSP microprocessor for ΣΔ-modulated signals. In: WORKSHOP ON SYSTEM EFFECTS OF LOGIC SOFT ERRORS, 2., USA, 2006. **Proceedings…**Illinois, USA : University of Illinois at Urbana-Champaign, Apr. 2006d.

SCHÜLER, E.; FARENZENA, D.S.; CARRO, L. Evaluating sigma-delta modulated signals to develop fault-tolerant circuits. In: IEEE EUROPEAN TEST SYMPOSIUM, 11., **Proceedings…**Southampton, United Kingdom : IEEE Press, May, 2006e.

SCHÜLER, E.; ERIGSON, M.I.; CARRO, L. Functionally fault-tolerant DSP microprocessor using sigma-delta modulated signals. **Journal of Electronic Testing: Theory and Applications (JETTA 2007)**. USA : Springer Science & Business Media, LLC, 2007a.

SCHÜLER, E.; NEGREIROS, M.; NOUET, P.; CARRO, L. A digitally testable capacitance-insensitive mixed-signal filter. In: IEEE EUROPEAN TEST SYMPOSIUM, 12., Germany, 2007. **Proceedings…**Freiburg, Germany : IEEE Press, May, 2007b.

SCHÜLER, E.; DE SOUZA JÚNIOR, A.A.; CARRO, L. Spare parts in analog circuits: a filter example. IEEE INTERNATIONAL SYMPOSIUM ON DEFECT AND FAULT TOLERANCE IN VLSI SYSTEMS, 22., Italy, 2007. **Proceedings…**Rome, Italy : IEEE Press, Sept., 2007c.

SMITH, E.C.; SHOGA, M. Double upsets from glancing collisions: a simple model verified with flight data. **IEEE Transactions on Nuclear Science**. New York: IEEE Press, 1859 – 1864, 1992.

SOUZA JR, A.A.; CARRO, L. Robust low-cost analog signal acquisition with self-test capabilities. In: IEEE INTERNATIONAL SYMPOSIUM ON DEFECT AND FAULT TOLERANCE IN VLSI SYSTEMS, 19., France, 2004. **Proceedings…**Cannes, France : IEEE Press, Mar., 2004.

SOUZA JR, A.A.; CARRO, L.; TOUSAAD, J. Adaptive processing applied to the design of highly digital analog interfaces. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, Japan, 2005. **Proceedings…**Kobe, Japan : IEEE Press, v.6, p.5597 – 5600 May, 2005.

TANSKANEN, J.M.A. Coefficient quantization error free fixed-point IIR polynomial predictor design. In: IEEE NORDIC SIGNAL PROCESSING SYMPOSIUM, Sweden, 2000. **Proceedings…**Kolmarden, Sweden : IEEE Press, p.219 – 222, Jun., 2000.

TSIVIDIS, Y.; BANU, M.; KHOURY, J. Continuous-time MOSFET-C filters in VLSI. **IEEE Transactions on Circuits and Systems**, New York: IEEE Press, v.33, p.125 – 140, Feb., 1986.

UCHIMURA, K.; HAYASHI, T.; KIMURA, T.; IWATA, A. Oversampling A-to-D and D-to-A converters with multistage noise shaping modulators. **IEEE Transactions on Acoustics, Speech, And Signal Processing**. New York: IEEE Press, v.36, n.12, Dec., 1988.

WEAVER, H., et al. An SEU tolerant memory cell derived from fundamental studies of SEU mechanisms in SRAM. **IEEE Transactions on Nuclear Science**. New York: IEEE Press, v.34, n.6, Dec., 1987.

YANG, F.L.; SALEH, R.A. Simulation and analysis of transient faults in digital circuits. **IEEE Journal of Solid-State Circuits**. New York: IEEE Press, v.27, i.3, p.258 – 264, Mar., 1992.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ERIK SCHÜLER

SINTESE

Porto Alegre
2007

1 INTRODUÇÃO

Embora discussões diversas debatam se a já conhecida Lei de Moore será seguida ou não [MOORE, 1965], existe um consenso de que o comprimento do canal de transistores MOSFET em breve quebrará a barreira nanométrica. Muito antes disso, porém, o número de transistores por chip deverá exceder a cifra de um bilhão de transistores [ITRS, 2006]. Por outro lado, existe um ramo na indústria de semicondutores o qual acredita que a atual tecnologia baseada em silício atenderá a demanda de miniaturização nas próximas uma ou duas décadas.

Além disso, diferentes estruturas em termos de material e arquitetura deverão ser empregadas. O transistor de elétron único [HADLEY, 1997], transistor de tunelamento ressonante [CHEN, 1996], transistores de nano-tubos de carbono [HAZEGHI, 2007] e o transistor *spin* [LENT, 1997] são algumas das possíveis alternativas para os dispositivos de silício.

Independentemente da tecnologia a ser usada, existe um consenso em um sentido: essa nova tecnologia deverá ser eficiente para manter a integridade do sinal a ser processado. Conforme o tamanho do canal do transistor diminui, também o número de elétrons (ou lacunas) diminui. Com um menor número de portadores passando pelo canal, embora a possibilidade desses portadores serem atingidos por uma partícula externa diminui, se tal evento ocorrer, o efeito causado será muito maior do que se houvesse um número maior de portadores.

Outro problema relacionado é o fato da tensão de alimentação dos circuitos vir diminuindo gradativamente. A carga critica, isso é, a carga necessária para causar a inversão do estado de um transistor, depende da capacitância do nó do circuito em uma proporção direta, e numa proporção inversa da tensão desse nó. Uma vez que ambas variáveis estão diminuindo, também a carga critica se torna menor. Esses dois fatores contribuem para o

aparecimento das chamadas falhas *soft*, ou seja, falhas induzidas pela incidência de partículas externas ou por ruído eletromagnético, que podem ter conseqüências catastróficas para uma parte de um sistema, ou para o sistema inteiro.

Nesse trabalho estamos principalmente preocupados com dois tipos de falhas *soft*: o *Single Event Upset* (SEU) e o *Single Event Transient* (SET). Quando uma partícula atinge um elemento de um circuito integrado, ela perde energia através da produção de pares elétrons-lacuna, resultando em uma densa região ionizada. Essa ionização causa uma corrente transiente, a qual pode se propagar através da lógica combinacional, recebendo o nome de SET. O pulso criado pode ser armazenado em um flip-flop, gerando assim um erro em um ou mais bits, ao que se denomina SEU.

Outro fator importante a ser levado em conta nas tecnologias futuras é o aumento da distância entre o projeto digital e o analógico. O avanço da tecnologia CMOS leva a grandes vantagens em circuitos digitais, uma vez que circuitos com menor gasto de potência e mais rápidos podem ser implementados com uma densidade de integração muito maior. No cenário oposto, circuitos analógicos não obtêm as mesmas vantagens da Lei de Moore.

Um dos principais circuitos analógicos que sofrem do problema de variabilidade paramétrica é o filtro. Para filtros analógicos, cuja freqüência de corte, ganho e fator de qualidade devem ser, em muitos casos, extremamente exatos, pequenas variações em capacitores, resistores e/ou indutores podem levar a completos descasamentos entre a resposta esperada e a obtida.

Mesmo quando cuidadosamente projetado para se obter um determinado valor de capacitância, o processo de fabricação não pode garantir que todos os capacitores em um circuito terão exatamente aquele valor, tampouco que todos os capacitores de um lote de circuitos terão o mesmo valor, o que reduz também a produtividade.

2 REVISÃO BIBLIOGRÁFICA

2.1 MODULAÇÃO SIGMA-DELTA *VERSUS* SEU

As técnicas usadas no desenvolvimento de circuitos robustos a falhas do tipo *soft* podem ser divididas em três níveis: tecnologia, projeto e sistema.

Ao nível de tecnologia, diferentes processos são utilizados na fabricação do transistor, como o *epitaxial-bulk CMOS*, onde uma máscara extra é usada no processo de corrosão [BASEDAU, 1995]. Embora o uso dessa técnica seja eficiente na redução de *Single Event Latch-Up* (SEL), ela não elimina a ocorrência de SEU. Outra técnica é o uso de *Silicon on Insulator* (SOI), onde uma fina camada de silício é depositada sobre um isolante, e então o transistor é construído sobre essa camada [IBM, 2005]. Essa técnica, entretanto, requer o uso de processos especiais de fabricação, com conseqüente limitação de produtividade, além de não mitigar completamente a ocorrência de SEU.

Aumento de tolerância ao nível de projeto inclui, por exemplo, o uso de células de memória resistores [WEAVER, 1987] e células de memória CMOS com estruturas de realimentação [RABAEY, 1996], o que implica em memórias maiores fisicamente. Outra solução é o uso de blocos de codificação e decodificação, usando, por exemplo, Hamming [MACKAY, 2003] ou Reed-Solomon [PLANK, 1996]. Embora essas sejam boas soluções, quando múltiplas falhas devem ser corrigidas, o custo em termos de complexidade e performance é extremamente alto. Além disso, se falhas simultâneas ocorrerem, essas técnicas não são capazes de detectar tais eventos.

Para se proteger circuitos digitais ao nível de sistemas, o uso de redundância de hardware ou software é a técnica mais conhecida. No caso de redundância de hardware, pode-se citar o uso de Triple Modular Redundancy (TMR) [CHANDE, 1989] como uma das técnicas mais utilizadas. Ao se utilizar redundância de software, pode-se citar o uso de

Algorithm Based Fault Tolerance [HUANG, 1984] e *Code Flow Check*, assim como duplicação de variáveis [REBAUDENGO, 1998], a qual pode ser implementada em alto nível.

Levando-se em conta os esquemas de redundância apresentados, é proposto nessa tese o uso de um novo tipo de redundância, baseada na redundância de sinal. A idéia não implica em se duplicar ou triplicar o sinal a ser processado, mas em criar uma nova representação para esse sinal, de certa forma que, mesmo sob a ocorrência de múltiplas e simultâneas falhas, a resposta final ainda sustenta uma resolução suficiente para a aplicação em questão. A técnica, a qual usa modulação sigma-delta (uma revisão pode ser encontrada em [NORSWORTHY, 1997]) para gerar os sinais redundantes, é para ser usada em circuitos digitais e, como será demonstrado, pode implicar, em alguns casos, em circuitos menores e mais rápidos. Nos casos onde as penalidades de área e tempo proporcionadas por outras soluções são menores, a tolerância à falhas obtida pela técnica proposta torna-se um fator determinante para o uso desse esquema.

Uma nova perspectiva é apresentada, onde não é necessário se preocupar se a falha irá ou não ocorrer, pois mesmo que ocorra, o circuito estará protegido. A idéia é que mesmo que a falha ocorra, graças à redundância presente no sinal, mesmo que vários bits sejam invertidos, a conseqüência não será tão prejudicial à resposta do sistema. Essa técnica, associada à idéia de tolerância a erros proposta em [GUPTA, 2004], pode garantir que o sistema irá produzir respostas apropriadas mesmo sob a ocorrência de múltiplas e simultâneas falhas.

2.2 VOTADOR ANALÓGICO *VERSUS* SET EM ESTRUTURAS TMR

Para lidar com o problema do SET, diferentes soluções têm sido propostas. Em [REBAUDENGO, 2002], duas soluções de baixo custo são comparadas: as capacidades de

detecção de erros de uma solução implementada em hardware, baseada em paridade de código e uma solução implementada em software, baseada em modificações em nível de código fonte. Em [MONGKOLKACHIT, 2003] todas entradas de latch são designadas como sendo críticas, e para cada entrada de latch um circuito adicional é inserido entre a saída do circuito combinacional e a entrada do latch. Em [LIMA, 2003], uma combinação de *Duplication With Comparison* (DWC) e *Concurrent Error Detection* (CED) baseada em redundância temporal é usada para detectar falhas permanentes em matrizes programáveis de FPGAs baseados em SRAM. A implementação de uma técnica de tolerância a erros baseada em redundância temporal e outra em redundância espacial e temporal é proposta em [ANGHEL, 2000]. Talvez a mais comum das soluções seja o uso de redundância de hardware redundancy, conhecida como Triple Modular Redundancy (TMR) [CHANDE, 1989], onde a tolerância é obtida através da triplicação dos blocos de hardware e a resposta correta é obtida através de votação. O voto pode ser feito em se contando as respostas corretas ou pela obtenção do valor médio [GAITANIS, 1988], [NANDURI, 1990].

Como será mostrado na seção 3.1, embora simples, o votador é um ponto crítico em relação às falhas. Se o votador apresentar baixa confiabilidade, o sistema inteiro será frágil, tanto quanto o votador. Algumas soluções para aumentar a tolerância à falhas de do votador incluem em triplicar esse circuito. Entretanto, o circuito final seria um novo TMR e sujeito às mesmas fraquezas. Outra solução consiste em projetar votadores capazes de se autotestarem [METRA, 1997] [CAZEAUX, 2004]. Além do problema de confiabilidade, outro fator importante para o votador é a velocidade, área e consumo de potência, além da complexidade do circuito. Na seção 3.2 é proposta uma nova maneira de se votar a saída de blocos TMR. Ao invés de se usar votadores digitais tradicionais, votadores analógicos são utilizados os quais, como demonstrado, são mais confiáveis que os circuitos digitais em relação à ocorrência de SET.

2.3 AMOSTRADOR ESTATÍSTICO *VERSUS* VARIABILIDADE DE PARÂMETROS

Existem diferentes possibilidades de se desenvolverem filtros analógicos. A implementação mais comum envolve capacitores e resistores (RC) para definir seus parâmetros [SCHAUMANN, 2001]. Para filtros integrados, o capacitor chaveado (SC) em geral é preferido para aumentar a exatidão dos valores [SCHAUMANN, 2001]. Também se pode citara implementação com transistor ativo, onde um transistor polarizado em sua região linear atua como um resistor [TSIVIDIS, 1986]. Outra possibilidade é a implementação através do uso de transcondutores, os quais exercerão o papel de resistores nesse caso. Essa técnica é conhecida como Gm-C [NAUTA, 1992].

Nesses quatro casos, sempre haverá a necessidade do uso de capacitores e, como já mencionado, embora a implementação de capacitores em sistemas VLSI seja possível, eles são caros em ter,os de área e, mais importante, a incerteza de seus valores absolutos pode chegar a mais de 10% [JOHNS, 1997]. Entre essas soluções, a implantação com SC é a mais atrativa, porém a excessiva área e a dificuldade de teste tornam essa técnica bastante custosa.

Além da tecnologia a ser usada no projeto do filtro, deve-se também levar em conta a topologia do mesmo. Por exemplo, uma implementação biquadrática pode ser feita através do uso de estruturas Tow-Thomas ou Sallen-Key, as quais podem ser feitas usando-se tanto redes RC quanto SC, transistores ativos ou Gm-C.

Uma maneira mais simples de se realizar filtros com fase linear é através do uso de estruturas do tipo *Finite Impulse Response* (FIR) [OPPENHEIM, 1999]. Tipicamente, filtros FIR são implementados em sistemas digitais, usando-se tanto software quanto hardware. Embora simples, a realização analógica desses filtros apresenta alguns problemas, principalmente quando capacitores chaveados são utilizados. Nesse caso, não apenas o projeto

é difícil devido à elevada quantidade de capacitores, mas também o teste se torna um ponto importante. Exemplos de FIR analógicos podem ser encontrados em [FISCHER, 1990], [DIAZ-SANCHES, 1996], [BURLINGAME, 2000], [CIOTA, 1996] e outros.

Nesse trabalho é introduzida uma nova maneira de se realizar filtros FIR para sinais analógicos. Esse filtro não usa capacitores para definir seus parâmetros e, uma vez que o processamento de sinais se dá de maneira digital, o teste pode ser feito de uma forma bastante simples e inteiramente digital, sem a necessidade de testadores extras e caros. O contrário, uma simples porta XOR e um contador são utilizados para o teste da parte analógica, enquanto que a parte digital pode ser testada utilizando-se qualquer esquema de teste existente.

3 SÍNTESE DOS RESULTADOS

3.1 MODULAÇÃO SIGMA-DELTA *VERSUS* SEU

O uso de moduladores sigma-delta para converter palavras PCM de n-bits em palavras sigma-delta de OSR-bits foi demonstrado e sua tolerância a falhas avaliada. Como demonstrado, mesmo sob a ocorrência de várias falhas, apenas através do ajuste da taxa de sobre amostragem (OSR), é possível melhorar-se a robustez do sistema esse obter respostas mais exatas.

Para demonstrar a técnica, diferentes estudos de casos foram desenvolvidos, culminando na realização de um processador digital de sinais (DSP) usando descrição VHDL e o desenvolvimento de um chip para futuros testes de radiação.O DSP, especialmente projetado para processar sinais sigma-delta, foi programado para implementar um filtro FIR 16 taps e, como demonstrado, mesmo sob a ocorrência de múltiplas falhas, a resposta do sistema obteve uma aproximação boa da resposta sem falhas. Quando comparado com um DSP padrão usando palavras binárias de n-bits, o DSP usando sigma-delta não apenas apresentou uma tolerância à falhas maior, como também apresentou melhor performance. Obviamente, quando uma maior tolerância é requerida, um compromisso é estabelecido, reduzindo-se a performance na mesma proporção. Na comparação de área, na implementação em FPGA ambas soluções apresentaram uma área equivalente, mas se as estruturas de memória forem levadas em conta, a solução com sigma-delta certamente irá necessitar de uma área maior. Também, resultados para a implementação de um filtro IIR foram apresentadas, mas devido a grande sensibilidade dessas estruturas em relação à variação dos coeficientes, outras topologias devem ser estudadas.

É importante diferenciar-se os dois tipos de modulação apresentados: para a modulação discreta, um resultado exato é obtido, mas a área e a performance são fatores que

se tornam limitantes; para a modulação contínua, circuitos mais rápidos e menores são possíveis, mas a diferença entre a resposta obtida pelo sigma-delta em comparação com uma implementação clássica se torna evidente, embora resultados satisfatórios possam também ser obtidos.

3.2 VOTADOR ANALÓGICO *VERSUS* SET EM ESTRUTURAS TMR

Tentando-se aumentar a confiabilidade à SET em sistema de TMR, o uso de um votador analógico foi proposto. A idéia de usar três inversores em curto-circuito seguidos de um circuito de decisão foi descrito e as simulações apresentadas. No papel de circuito de decisão, quatro circuitos diferentes foram testados: dois comparadores, um inversor *push-pull* e um bloco *Schmitt-trigger*. Como demonstrado, mesmo com múltiplas falhas injetadas em diferentes pontos do circuito, em diferentes tempos, na maioria das simulações eles se mantiveram estáveis, sem propagar a falha para a saída, pelo menos para as amplitudes de corrente injetadas. Mais importante, quando comparado a votadores digitais, não apenas a quantidade de drenos sensíveis diminuiu, mas também resultados mais robustos foram obtidos quando a mesma falha foi injetada em ambos os circuitos.

A idéia proposta pode ser facilmente estendida para um sistema n-MR, simplesmente através da inserção de um inversor em curto-circuito para cada novo bloco de hardware inserido, e do ajuste da tensão de referência do comparador de decisão.

3.3 AMOSTRADOR ESTATÍSTICO *VERSUS* VARIABILIDADE DE PARÂMETROS

Nessa parte foi apresentado um filtro FIR para sinais mistos, o qual não usa componentes passivos para a definição dos parâmetros. O sinal de entrada é representado através de um trem de bits originados de sucessivas comparações entre o sinal amostrado e

um ruído randômico. Uma vez que o sinal a ser filtrado e os coeficientes do filtro estão no domínio digital, todo o resto do processamento se dá de maneira digital.

Um esquema de teste foi apresentado, e foi demonstrado que é possível testar ambas partes, analógicas e digitais, sem a inserção de hardware extra, e sem gerar sinais extras de teste. Uma pequena manipulação das chaves presentes coloca o sistema na configuração de teste, o qual utiliza apenas portas XOR e contadores. Foi também demonstrada a idéia de estender o conceito de partes sobressalentes (*spare-parts*) para o domínio analógico. Devido à estrutura do filtro, os taps que não estiverem dentro das especificações podem ser facilmente identificados, e taps de reposição inseridos no lugar desses, tudo através da manipulação de chaves.

4 CONSIDERAÇÕES FINAIS

A tese apresenta três diferentes técnicas para lidar com problemas típicos de tecnologias integradas, os quais irão de pronunciar cada vez mais conforme a evolução dessa tecnologia. O trabalho mais desenvolvido, o uso de modulação sigma-delta para lidar com SEU em sistemas digitais, apresenta um novo paradigma no tratamento de falhas *soft*. Ao invés de utilizar-se redundância em hardware ou software, a redundância de sinais é sugerida.

No segundo trabalho, apresentou-se o uso de votadores analógicos para lidar com o problema de SET em sistemas de TMR. A idéia por trás dessa técnica é que, devido à corrente DC que flui constantemente em circuitos analógicos, a influencia de SET nesses circuitos seria menor do que em circuitos digitais, onde não existe esse tipo de corrente de polarização.

Finalmente, um terceiro trabalho é introduzido, apresentando o uso de amostradores estatísticos para serem usados na realização de filtros para sinais analógicos, sem a necessidade de componentes passivos para a definição de seus parâmetros.Além disso, a tarefa de teste é facilmente executada, sem a inserção de novos circuitos testadores.

ANEXOS

A seguir os artigos publicados referentes aos três assuntos desenvolvidos no decorrer desse trabalho. São doze publicações no total, sendo oito referentes ao uso de sigma-delta para lidar com SEU, dois sobre o uso de votadores analógicos e dois sobre o uso de amostradores estatísticos para o problema da variabilidade analógica. Os artigos estão relacionados em ordem cronológica de publicação, na mesma formatação em que foi publicado.

Spare Parts in Analog Circuits: a Filter Example

Erik Schüler, Adão Antônio de Souza Júnior, Luigi Carro
Universidade Federal do Rio Grande do Sul
Departamento de Engenharia Elétrica
erik.schuler@ufrgs.br, adaojr@gmail.com, carro@inf.ufrgs.br

Abstract

Spare parts technique has been widely used in digital designs. As memory cells are more susceptible to defects and faults than logic cells, redundancy has been extensively used for enhancing defect and fault tolerance through repair by spare replacement. The technique also aims yield increase, and points to be a very good solution since density integration gets ever higher. In this work, we propose the use of spare parts to develop reliable analog circuits, thus increasing fault tolerance by choosing among many identical blocks, the best ones that will compose the circuit. An example using a mixed-signal FIR filter is presented, showing that the technique can easily be adapted to help increase yield of analog circuits, too.

1. Introduction and Literature Review

The scaling down of CMOS technology leads to great advantages in digital circuits, since low power and faster circuits can be implemented with an increased integration density. Also, design automation and test are relatively mature for medium density and state-of-the-art digital circuits.

In the opposite scenario, analog circuits do not take advantage of this trend. On the one hand, they are necessary in most of the System-on-Chip (SoC) devices, and analog circuits size is not reduced at the same rate as digital circuits are. On the other hand, scaling down can introduce some problems. Process variability affects transistors but also passive components in a significant amount. Finally, testing analog circuits is not that easy as it is for their digital counterparts, since the signals are defined in the whole range of voltage from ground to +/-Vdd, and the output is often embedded in the SoC, thus reducing the observability.

The yield and reliability problem, however, reaches both, the digital and the analog design. Nevertheless, in the digital domain, many techniques exist to solve this problem. A very well-known one uses spare parts. Among the digital circuits, memory occupies the largest portion of them. Since memory cells are more prone to defects and faults, redundancy has been extensively used for enhancing defect and fault tolerance through repair by spare (row and column) replacement [1], [2].

The idea of using redundancy in analog designs has already been presented to analog-to-digital converters. In [3], it is proposed the use of many redundant comparators put in a single die, then choose among these comparators, which ones have best characteristics to be used in the ADC.

We now extend the idea of spare parts in analog designs, and present a Finite Impulse response filter example. The idea to add extra redundant analog components is presented, and how these extra components can be used to replace failed ones is studied. As it will be shown, non-working branches can be easily tested and replaced, helping to increase yield, reliability and fault tolerance in analog designs.

This paper is organized as follows: section 2 presents the mixed-signal filter approach, and simulations results to demonstrate it functionality. In section 2 it is also presented the test

methodology, showing that it is possible to test the whole analog part through the use of a the presented digital block. Also, the use of redundant parts to the replacement of non-working ones is presented. As it will be seen, one can easily adapt this filter to use spare parts and obtain better responses, or increase reliability, for example. Finally, section 3 presents the final remarks.

2. A Mixed-Signal FIR Filter

In order to demonstrate the use of spare parts in analog circuits, a mixed-signal Finite Impulse Response (FIR) filter was implemented and simulated. Thanks to the structure of the proposed filter, the use of extra similar blocks makes possible an easy scheme to substitute non-working taps of the filter by working ones. We now present the proposed FIR filter structure, demonstrating its functionality through some simulations results.

2.1. Mixed-Signal FIR Filter Description

There are several different ways to implement analog filters. The more classic implementation uses capacitors and resistors (RC) to define the filter parameters [4]. For integrated filters, the switched capacitor (SC), i.e. a pair of switches and a capacitor replacing a resistor, is often preferred to increase the accuracy [4]. Also, in the active transistor implementation, a transistor polarized in its linear region plays the role of a resistor [5]. Another possible implementation consists of using transconductors, which, in this case, will act as the active part of the filter and will avoid the use of resistors. This implementation is known as Gm-C [6].

Despite the technology that will be used to design the filter, one can consider its topology. For example, a *biquadratic* implementation can be done through the use of a Tow-Thomas or a Sallen-Key structure, which can be done through the use of an RC network, a SC, active transistors or even Gm-C.

A simpler way to make high order filters is by using a Finite Impulse Response (FIR) structure [8]. Typically, FIR filters are implemented in digital systems, by using either software or hardware. Digital signal processors (DSP) are suitable for this kind of filters, since operations like multiplication can be done in a single machine cycle. However, analog implementation of FIR filter is also possible, but, again, they make intensive use of capacitors to define gain, cutoff frequency and quality factor. Examples of analog FIR filters can be found in [9], [10], [11] and [12], among others.

Figure 1 shows one of the possible structures for a FIR filter, known as direct form FIR. It is basically an input shift register, which will receive each sampled value that will be multiplied by a constant (the coefficients). In the multiplication stage, each shift register output is multiplied by each filter coefficient ($b_0, b_1, \ldots b_n$). Finally, the multipliers outputs are added to generate one filtered point. Then, the input values are sampled and shifted, and the process restarts.

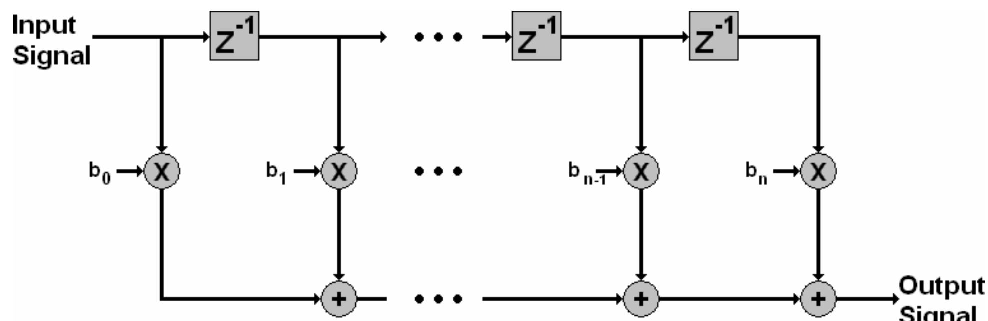


Figure 1. Direct Form FIR filters structure.

Although simple, the analog implementation of this filter raises some problems, mainly when switched capacitors are used. In this case, not only the design is difficult due to the elevate number of capacitors, but also the test becomes an important aspect. Since a set of capacitors and switches is used to define the filter coefficients, each of these components must be tested to guarantee a correct response for the filter.

We now introduce a different way to implement FIR filters, suitable to analog signal processing. This filter does not use capacitors to define its cutoff frequency or its quality factor, and, since the processing is digitally made, it can be easily tested without the need of complex or expensive testers.

Figure 2 presents the filter implementation. If one compares it to the structure presented in figure 1, there is a perfect correspondence between the blocks. First, an analog shift register is used to receive and shift each sampled value. This block could be implemented in many different ways [9] [10] [11], without interfering with the following stages of the filter, since its only function is to hold each sample during the comparison/multiplication process.

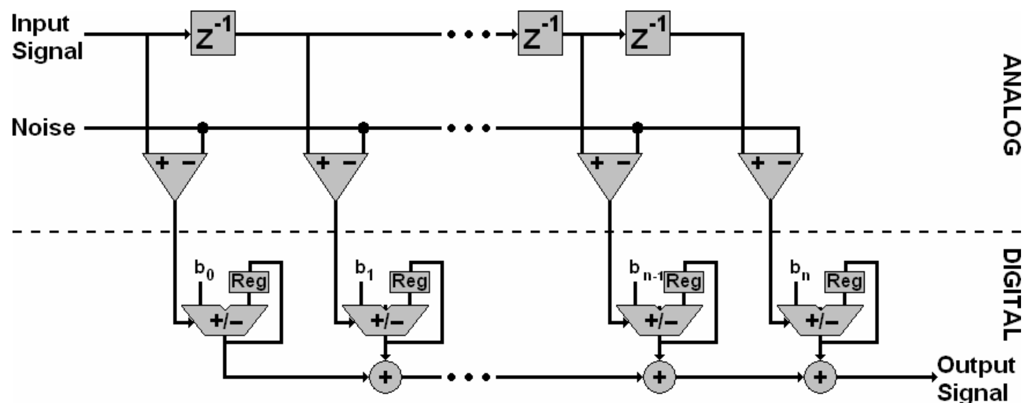


Figure 2. Mixed-signal implementation of a FIR filter.

The next step acts as an Analog-to-Digital Converter (ADC), which will generate a bit stream for each sampled value. The bit stream is generated through multiple comparisons between the sampled value and a random generated noise. This kind of signal conversion has already been presented in previous works [13] [14], whose idea is based on sampling a signal statistics through stochastic quantization. That is, when the analog signal S(t) is compared with a multi-level random reference N(t), the output Probability Distribution Function (PDF) p(O) will be the determined by convolution of their individual PDFs, as represented in figure 3. Since the reference bandwidth is much higher than the maximum frequency in the signal, the mean value of the output bit stream will represent the input signal.

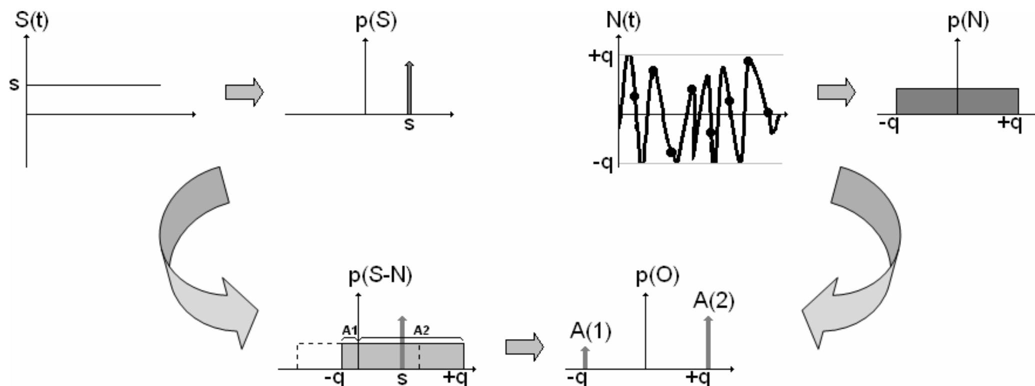


Figure 3. Statistic acquisition with noise dithering.

Up to this point, the input analog signal is represented in a digital fashion through the bit stream, and a digital processing must follow. The next stage, thus, will perform the multiplication between each bit stream and each filter coefficient, in a digital manner. To do this, a very simple way is to decide, for each generated bit stream, whether to add or to subtract the coefficient value, depending on the value of the bit in the stream.

Figure 4 shows a Matlab® simulation of the proposed filter, whose coefficients implement a FIR with a Kaiser windowing with *beta* equal to 0.5. The input signal is a 20KHz sine wave added to a white noise with ten times lower amplitude. This signal was sampled with a sampling frequency sixteen times higher than the Nyquist rate, generating 4096 sampled points. These points were filtered by the proposed FIR, and the Fast Fourier Transform of the filter output was calculated.

In figure 4(a), the FFT of a normal implementation of the same FIR (see figure1) is plotted in order to compare to the responses of the proposed filter. Figure 4(b), thus, depicts the simulation for the proposed filter, where each sampled point is compared to a noise generating 256 bits. As it can be seen, some harmonics rise due to the signal/noise comparison. However, the amplitude of such harmonics can be reduced by increasing the number of comparisons, as one can note by taking a look at figure 4(c), where each signal/noise comparison generates 1024 bits. This phenomenon of the reduction in the noise level of the filtered signal can be better seen in figure 5, where different numbers of bits are generated for each simulation, and the difference between a normal filter FFT and the proposed filter FFT is plotted. As noted, going from 256 to 1024 bits, the mean value of the noise level reduces to almost zero, denoting an increase in the signal to noise ratio.

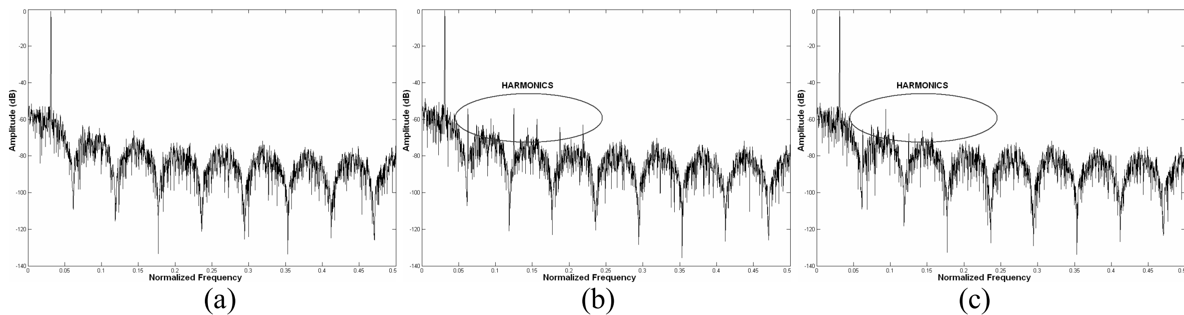


Figure 4. Frequency spectra to compare the responses of the (a) proposed filter with an OSR of 256, (b) an OSR equal to 1024 and (c) a normal filter.

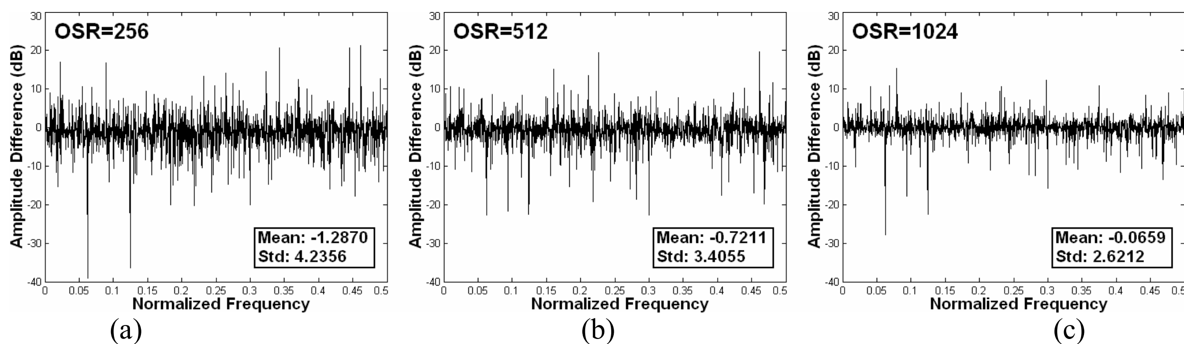


Figure 5. Noise level plotted by making the difference between a FFT of a normal filter and the proposed one using (a) an OSR equal to 256, (b) 512 and (c) 1024.

Once the filter functionality has been presented, we shall now study how to implement spare parts into the circuit. Due to the topology of the proposed filter, one can use extra identical taps for a replacement scheme. Next section presents the testing and replacement approach, and how this idea can be easily implemented just through the use of a set of switches.

2.2. Adding Spare Parts: testing and replacement scheme

This section presents how the redundant taps can be added to the proposed filter. The idea is to have many identical taps in such a way that, whenever an operating tap stops functioning it can be easily replaced by a good one. Also, during fabrication process, yield can be increased since the non-working taps will not determine the discard of the entire circuit.

As presented in figure 2, the proposed circuit can be divided into an analog and in a digital part. This division is also obtained when testing the circuit. It is here assumed that the digital part of the filter may be tested by any existing testing method used to test digital circuits. For the analog part, however, some test steps must be taken. Nevertheless, as it will be seen, testing the analog block can be done in a very easy and cheap way, without any need of extra circuitry.

The analog block of the filter is composed basically of three components: the analog shift cells, implemented through the use of a capacitor and a buffer (see figure 6), a set of switches and comparators, which will compare each sampled point with the noise.

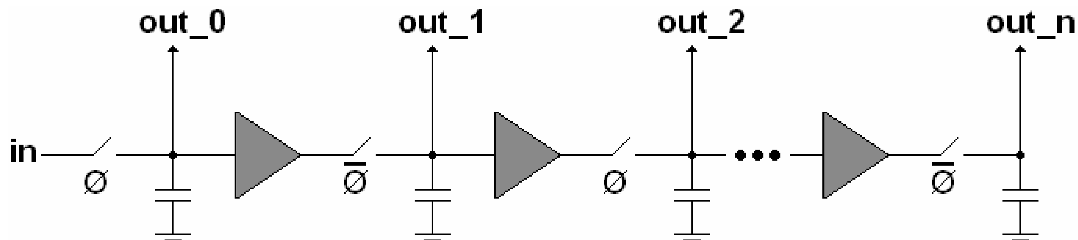


Figure 6. Realization of the analog shift-register.

The test methodology proposed here consists in testing each analog branch as a whole block, that is, each tap of the FIR will be tested separately, as one single block. This allows one to use the idea of spare parts, that is, extra analog branches should be fabricated in the circuit in such a way that, in the case any analog branch fails, it can be easily replaced by a new one.

Figure 7 depicts how these extra parts can be placed in the circuit. With this configuration, just through some switches manipulations, non-working taps can be replaced by working ones, also allowing the best combination to achieve the best filter response.

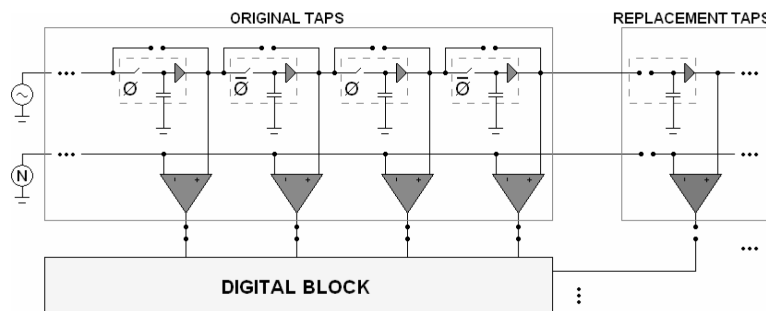


Figure 7. Adding extra taps to the filter: these spare parts can easily replace non-working analog branches.

Let us take, for example, the situation where the second tap of the filter in figure 7 is found to be a non-working one. The substitution scheme consists just in by-passing this tap through one of the switches, and add one of the extra tap, also by manipulating two switches. This situation can be seen in figure 8.

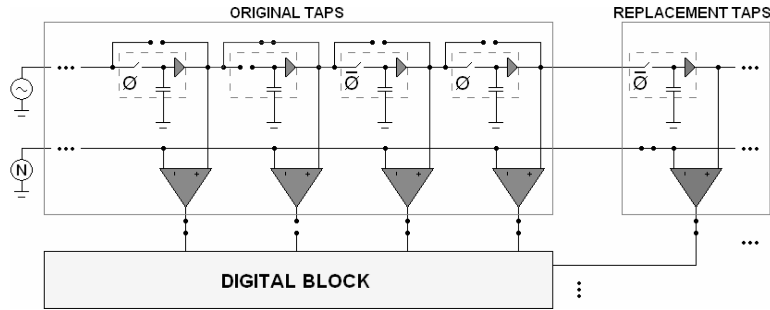


Figure 8. Second tap is replaced by one the existing replacement taps.

We shall now take a look in the way the analog taps are tested in order to find out those taps that must be substituted. Two test examples are given next: test of the analog shift-register buffer gain and test of the comparator off-set voltage. Many others parameters test can be derived from these two basic examples.

To test the analog shift-register buffer gain, the simplest way is by sampling a certain constant value, generating the correspondent bit stream, shifting the sampled value and generating a new bit stream. Now, just by comparing the counted values of the generated bit streams, it is possible to determine if the buffer gain is inside the desired range. Figure 9 shows the buffer gain test sequence, while figure 10 demonstrates how the difference between the counted values increases with the variation in the buffer gain value.

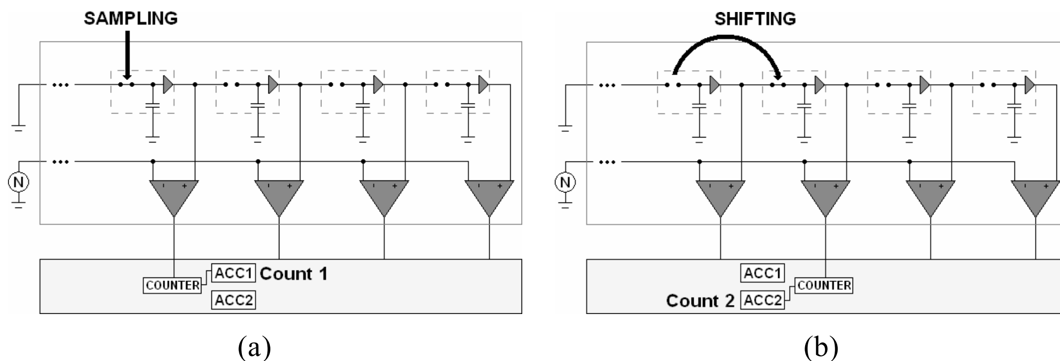


Figure 9. Sequence for testing the analog shift-register buffer gain. Decision is taken by comparing counted values: CASE values ARE “similar” gain is OK; CASE values ARE “different” gain is not OK.

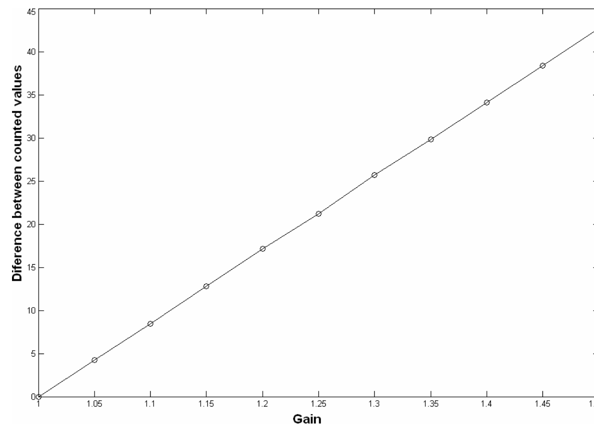


Figure 10. Gain variation versus difference between counted values during the test of the analog shift-register buffer gain.

As seen in figure 10, as the buffer gain varies from 1 to 1.5, the difference between the counted values increases linearly from 0 to 45, thus allowing to detect any considerable variation in the analog shift-register buffer gain.

The test of the off-set voltage of the comparators can be done in a similar way. First, a given value is sampled by two analog registers. Then, each comparator will generate its respective bit stream. These bit streams are XORed, and the result is counted in order to determine if the off-set voltage is inside the expected range. Figure 11 shows how this test can be done, and figure 12 how the counted value varies as the off-set voltage increases.

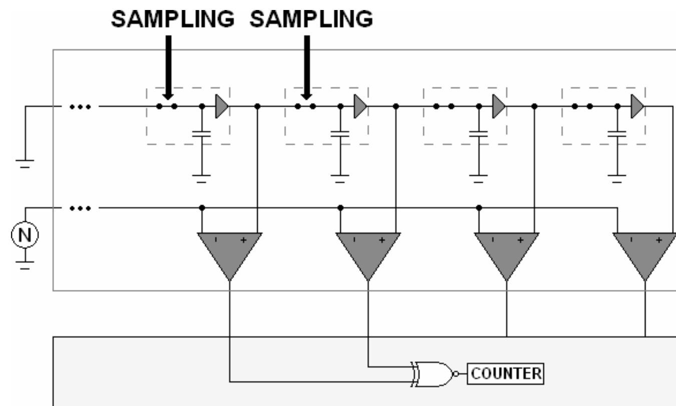


Figure 11. Sequence for testing the comparators off-set voltage. Decision is taken by comparing counted value: CASE value IS "small", off-set voltage is OK; CASE value IS "large", off-set voltage is not OK.

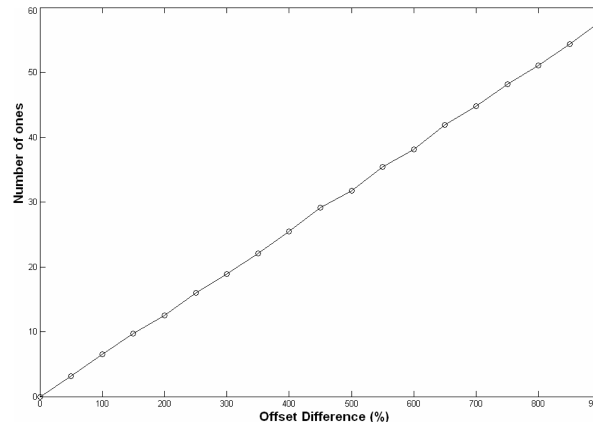


Figure 12. Off-set voltage variation versus difference counted value during the test of the analog comparator off-set voltage.

Once more one can see in figure 12 that, as the percentage of the offset difference between two comparators increases from 0% to 900%, the number of ones in the generated bit streams also increases from 0 to 60, in a linear fashion.

By using the same technique of counting and comparing different bit streams, others parameters can be measured in the filter, thus defining whether the analog branch has enough conditions to be characterized as a good or a bad branch.

3. Conclusions

Digital CMOS technology has been continuously shrinking, and the analog design must follow this trend in order to keep the yield and the integration density, thus not letting the gap between

digital and analog design increase even more. As a consequence, analog system gets harder to design, mainly for components that have a high dependence with the technology variability.

In the digital domain, many techniques have been used to increase yield, reliability and fault tolerance. A very common way to do this is by using spare parts, that is, include hardware redundancy in order to substitute failed blocks by good ones.

In this work we have presented the idea of extend the concept of spare-parts to the analog domain. To implement the idea, a mixed-signal FIR filter was presented, showing that it is possible to substitute non-working parts by working ones in a very simple way. Since the circuit test is also facilitated due to the circuit structure, finding the non-working taps also becomes a very easy task.

Future works include extending the herein proposed idea to others analog circuits, and demonstrate how extra analog redundancy can also help to increase yield in this domain.

4. References

[1] W. R. Moore, “A Review of Fault-Tolerant Techniques for the Enhancement of Integrated Circuit Yield”, Proceedings of the IEEE, Volume 74, Issue 5, May 1986 Page(s):684 – 698.

[2] M. Choi, N. Park, F. Lombardi, Y.B. Kim and V. Piuri, “Optimal Spare Utilization in Repairable and Reliable Memory Cores”. Proceedings of the 2003 International Workshop on Memory Technology, Design and Testing (MTDT’03).

[3] M. P. Flynn, C., Donovan and L., Sattler, “Digital Calibration Incorporating Redundancy of Flash ADCs”, IEEE Transactions on Circuits and Systems II: analog and digital signal processing, vol. 50, no. 5, may 2003.

[4] Schaumann, R.; Valkenburg, M.E.V., “Design of Analog Filters”, New York: Oxford University Press, 2001. 737p.

[5] Tsividis, Y.; Banu, M.; Khoury, J., “Continuous-time MOSFET-C filters in VLSI”, IEEE Transactions on Circuits and Systems, New York, v.33, Feb. 1986. p.125 – 140.

[6] Nauta, B.A., “CMOS Transconductance-C filter technique for very high frequencies”, IEEE Journal of Solid State Circuits. v.27, feb. 1992.

[7] D. Johns and K. Martin, "Analog Integrated Circuit Design", John Wiley & Sons, Inc. 1997, ISBN 0-471-14448-7.

[8] Oppenheim, Alan V., “Discrete-time signal processing”, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999. 870 p.

[9] Fischer, G., “Analog FIR filters by switched-capacitor techniques”, Circuits and Systems, IEEE Transactions on, v.37, Issue 6, June 1990. p: 808 – 814.

[10] Diaz-Sanchez, A.; Ramiriz-Angulo, J., “ VLSI voltage mode FIR analog adaptive filters”, Circuits and Systems, 1996., IEEE 39th Midwest symposium on, v.1, 18-21 Aug. 1996. p:141 – 144.

[11] Burlingame, E.; Spencer, R., “An analog CMOS high-speed continuous-time FIR filter”, Solid-State Circuits Conference, 2000. Proceedings of the 26th European, 19-21 Sept. 2000. p:288 – 291.

[12] Ciota, Z.; Napieralski, A.; Noullet, J.L., “Analogue realisation of integrated FIR filters”, Circuits, Devices and Systems, IEE Proceedings. v.143, Issue 5, Oct. 1996. p:274 – 281.

[13] de Souza, A.A., Jr.; Carro, L., “Robust low-cost analog signal acquisition with self-test capabilities”, Defect and Fault Tolerance in VLSI Systems, 2004. DFT 2004. Proceedings. 19th IEEE International Symposium on.10-13 Oct. 2004. p(s):239 – 247.

[14] de Souza, A.A., Jr.; Carro, L., “Statistical acquisition for embedded instrumentation”, Instrumentation and Measurement Technology Conference, 2004. IMTC 04. Proceedings of the 21st IEEE. v.1, 18-20 May 2004. p:314 – 317.

A Digitally Testable Capacitance-Insensitive Mixed-Signal Filter

Erik Schüler[1], Marcelo Negreiros[1], Pascal Nouet[2] and Luigi Carro[3]
[1]*Universidade Federal do Rio Grande do Sul, Departamento de Engenharia Elétrica*
[2]*Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier*
[3]*Universidade Federal do Rio Grande do Sul, Instituto de Informática*

eschuler@eletro.ufrgs.br, negreiro@eletro.ufrgs.br, nouet@lirmm.fr, carro@inf.ufrgs.br

Abstract

One of the main problems when developing analog filters in VLSI is to achieve high accuracy regarding the cutoff frequency. This is mainly due to the difficulty in obtaining accurate time constants. Testing of such filters is also challenging, in the sense that special equipment is required. Small deviations in the resistor or capacitor values may lead to a very high mismatch between the expected and the achieved cutoff frequency. Although switched-capacitor or active-transistor techniques may produce good results, the cost to use such approaches becomes another limiting factor, and only increases the tester needs. In this work, we present the development of an analog FIR filter, which does not use passive components to tune the cutoff frequency or the quality factor. Instead, the filter coefficients and the input signal are represented in a bit stream fashion, and are digitally processed, thus avoiding the use of expensive analog-to-digital converters. The impact of this filter architecture on test cost and possible design-for-test techniques are discussed in this paper.

1. Introduction

The scaling down of CMOS technology leads to great advantages in digital circuits, since low power and faster circuits can be implemented with an increased integration density. Also, design automation and test are relatively mature for medium density and state-of-the-art digital circuits.

In the opposite scenario, analog circuits do not take advantage of this trend. On the one hand, they are necessary in most of the System-on-Chip (SoC) devices, and analog circuits size is not reduced in the same rate as digital circuits are. On the other hand, scaling down can introduce some problems. Process variability affects transistors but also passive components in a significant amount. Finally, testing analog circuits is not that easy as it is for their digital counterparts, since the signals are defined in the whole range of voltage from ground to +/- Vdd, and the output is often embedded in the SoC, thus reducing the observability. It is then mandatory to consider self-testable analog blocks.

One of the main analog circuits that suffer from the scaling problem, specifically the parameter variation problem, is the filter. For analog filters, whose cutoff frequency, gain and quality factor must be, in many cases, extremely accurate, small deviations in capacitors, resistors and/or inductors values may lead to a complete mismatch between the expected and achieved cutoff frequency, for example. Even if one carefully designs a capacitor to obtain a certain value, the fabrication process can not guarantee an exact replication of this capacitor all through the entire circuit, leading, very often, to the increase of expensive trimming circuits, in order to tune the filter response. Moreover, yield becomes a problem due to the same replication problem, that is, it is hard to obtain the same accuracy for all capacitors in a certain filter production lot. Back to manufacturing test, it is then necessary to identify from the specifications those circuits where parametric errors may lead to a single specification out of the 6σ range. Offsets, characteristic frequencies, quality factors and many other specifications parameters are considered.

In order to cope both with the parameter variations and the test problem, we present in this work the implementation of an analog filter, whose cutoff frequency and quality factor are not based on the value of passive components such as capacitors, resistors or inductors. Instead, these factors are digitally determined. The filter, as will be shown, is a mixed-signal implementation, where the analog part can be easily tested simply through an exclusive-or gate, which is also already present in the circuit, with no need for signal processing computations (like a FFT, for example), neither expensive signal generators nor expensive mixed-signal testers. For the digital part test, any already existing low cost test scheme can be used.

The paper is organized as follows: section 2 presents the analog capacitance-insensitive filter approach, as well as its performance evaluation achieved through Matlab® simulations. In section 3, the test methodology is explained, showing that it is possible to test the whole analog part through the use of a single exclusive-or gate, and simulation results are provided. Finally, section 4 presents the final remarks.

2. Capacitance-Insensitive Analog Filter

There are several different ways to implement analog filters. The more classic implementation uses capacitors

This work has been partially supported by CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

and resistors (RC) to define the filter parameters [1]. For integrated filters, the switched capacitor (SC), i.e. a pair of switches and a capacitor replacing a resistor, is often preferred to increase the accuracy [1]. Also, in the active transistor implementation, a transistor polarized in its linear region plays the role of a resistor [2]. Another possible implementation consists of using transconductors, which, in this case, will act as the active part of the filter and will avoid the use of resistors. This implementation is known as Gm-C [3].

In these four cases, one will always need to use capacitors and, as already mentioned, although capacitors can be implemented in VLSI systems, they are expensive in terms of area and, most important, the uncertainty on their absolute value is higher than ten percent [4]. Among these solutions, the SC implementation is then attractive as a better accuracy can be obtained. However, the area overhead and the difficulty to test these circuits both affect SC circuits.

Despite the technology that will be used to design the filter, one can consider its topology. For example, a biquadratic implementation can be done through the use of a Tow-Thomas or a Sallen-Key structure, which can be done through the use of an RC network, a SC, active transistors or even Gm-C.

A simpler way to make high order filters is by using a Finite Impulse Response (FIR) structure [5]. Typically, FIR filters are implemented in digital systems, either using software or hardware implementation. Digital signal processors (DSP) are suitable for this kind of filters, since operations like multiplication can be done in a single machine cycle. However, analog implementation of FIR filter is also possible, but, again, they make intensive use of capacitors to define gain, cutoff frequency and quality factor. Examples of analog FIR filters can be found in [6], [7], [8] and [9], among others.

Figure 1 shows one of the possible structures for a FIR filter, known as direct form FIR. It is basically an input shift register which will receive each sampled value that will be multiplied by a constant. In the multiplication stage, each shift register output is multiplied by each filter coefficient (b0, b1,…bn).

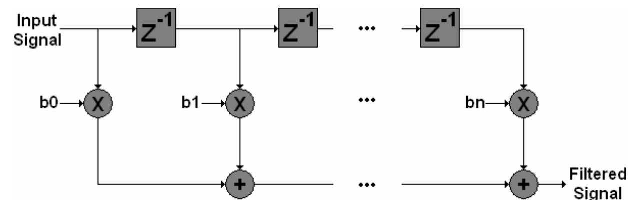


Figure 1. Direct Form FIR filters structure.

Finally, the multipliers outputs are added to generate one filtered point. Then, the input values are sampled and shifted, and the process restarts.

Although simple, the analog implementation of this filter raises some problems, mainly when switched capacitors are used. In this case, not only the design is difficult due to the elevate number of capacitors, but also the test becomes an important aspect. Since a set of capacitors and switches are used to define the filter coefficients, each of these components must be tested to guarantee a correct response for the filter.

We now introduce a different way to implement FIR filters, suitable to analog signal processing. This filter does not use capacitors to define its cutoff frequency or its quality factor, and, since the processing is digitally made, it can be easily tested without the need of complex or expensive testers. On the contrary, a single exclusive-or gate is used to test the analog part, while the digital part can be tested using classical digital testing schemes, as it will be presented in section 3.

Figure 2 presents the filter implementation. If one compares it to the structure presented in figure 1, there is a perfect correspondence between the blocks. First, an analog shift register is used to receive and shift each sampled value. This block could be implemented in many different ways [6] [7] [8], without interfering with the following stages of the filter, since its only function is to hold each sample during the comparison/multiplication process. As it will be shown in section 3, the way this block is defined does not interfere in the test scheme either.

The next step acts as an Analog-to-Digital Converter (ADC), which will generate a bit stream for each sampled value. The bit stream is generated through multiple comparisons between the sampled value and a random generated noise. This kind of signal conversion has already been demonstrated in previous works [10] [11].

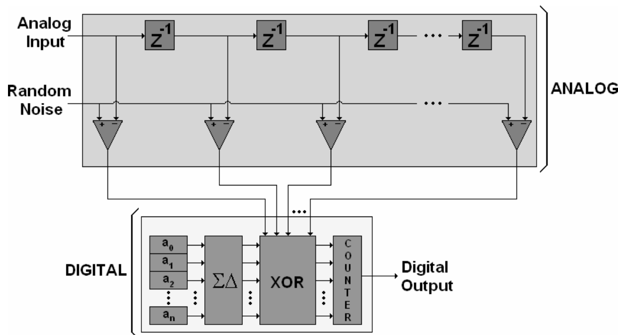


Figure 2. Capacitance-insensitive analog FIR filter implementation.

The idea is based on sampling a signal statistics through stochastic quantization. That is, when the analog signal S(t) is compared with a multi-level random reference N(t), the output Probability Distribution Function (PDF) p(O) will be the determinate by convolution of their individual PDF, as represented in figure 3. Since the reference bandwidth is much higher than the maximum frequency in the signal, the mean value of the output bit stream will represent the input signal.

Since now the input analog signal is represented in a digital fashion through the bit stream, a digital processing must follow. The next stage, thus, will perform the multiplication between each bit stream and each filter coefficient, in a digital manner. The coefficients are also represented as bit streams, but now using a different technique, which has also been presented in previous works [12] [13].

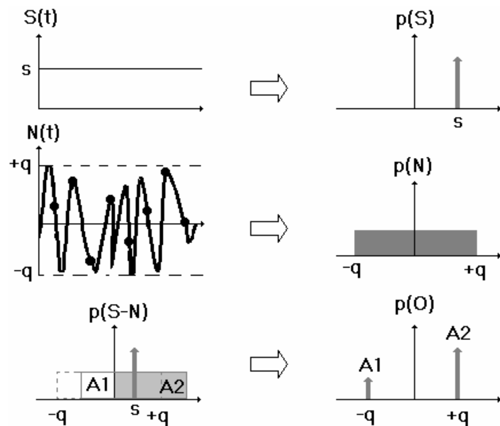


Figure 3. Statistic acquisition with noise dithering.

The idea is that each filter coefficient is passed through a digital sigma-delta modulator [14], as represented in figure 4. This way, a redundant representation of each coefficient value is generated. The intrinsic redundancy of sigma-delta modulated signals comes from the fact that the bit stream carries the original signal representation plus a certain quantization noise [14]. In another point of view, one can imagine that the original n-bits words are now represented in a sequence of Least Significative Bits (LSB), since the sigma-delta modulator generates a 1-bit bit stream as the modulated output.

Now, since both, multiplier and multiplicand, have the same representation, that is, a sequence of bits representing the original value, the product can be obtained simply by doing an exclusive-or operation between all bits of one bit stream with each bit of the other bit stream. The mean of the output bit stream represents the product. Finally, the last stage comprises the addition step. This operation can be done just through the use of an up/down counter, whose output will represent one filtered point. This value can now be used in a digital way or, if desired, can be reconverted to the analog domain through the use of a single Digital-to-Analog Converter (DAC).

Since the input signal is analog and the whole signal processing occurs by means of a digital process, one can conclude that, in fact, the circuit is insensitive to capacitance variations.

Figure 5 shows a Matlab® simulation of the proposed filter. A comparison with a normal FIR filter implementation is showed. For the frequency analysis in figure 5(a), a 16 taps FIR was simulated, generating 512 points for the Fast Fourier Transform (FFT) computation. In figure 5(b), the time response is presented. As it can be noted in figure 5(a), there is a small difference between the responses, mainly in high frequencies. The difference between the signal-to-noise ratios (SNR) is minimal.

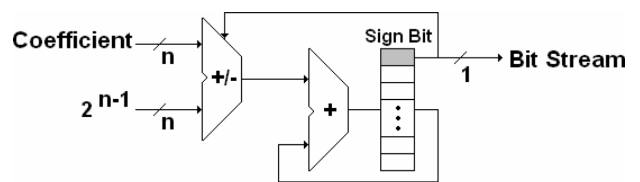
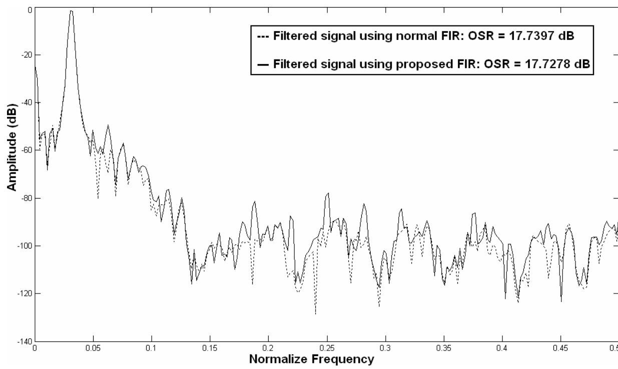


Figure 4. Coefficient modulation by a first-order digital sigma-delta modulator.

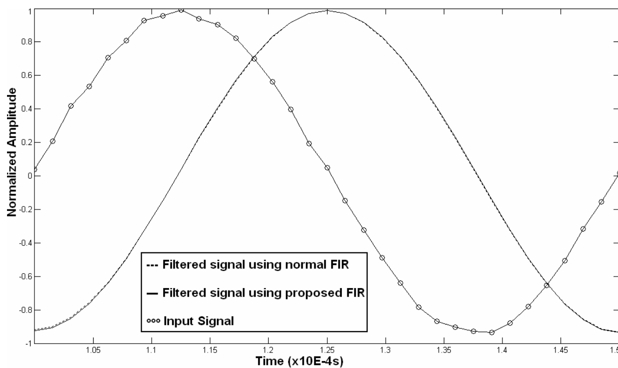
This difference occurs mainly due to the noise introduced into the signal band after the comparison process. However, the contribution of this noise can be reduced simply by increasing the over sampling ratio (OSR) of each sampled value of the input signal [10].

For this simulation an OSR of 256 for both the input signal samples and the sigma-delta modulated coefficients were used. Each sample of the input signal is generated with an OSR of 16.

The next section describes the test methodology, which has no need of extra testers, but only the components already present in the circuit.



(a)



(b)

Figure 5. In (a), frequency analysis to compare the proposed FIR to a common implementation of the same filter; in (b), the same comparison showing the time response.

3. Test Methodology

As presented in figure 2, the proposed circuit can be divided into an analog and in a digital part. This division is also obtained when testing the circuit.

It is here assumed that the digital part of the filter may be tested to concentrate on the analog part composed with an analog shift register and a bunch of comparators that provide the bit stream of each analog state of the shift register. Also, since analog shift cells are implemented through the use of a capacitor, a set of switches and a buffer, these three components should also be tested. In our case, however, the test methodology consists in testing each analog branch as a whole block, that is, each tap of the FIR will be tested separately, as one single block. Beyond the reduction in test complexity, this methodology guarantees that, no matter how the analog delays cells are implemented, the test approach will be always the same.

The idea to test each branch is quite simple. Let us suppose the analog branches represented in figure 6(a). In this example, the analog delay cell is implemented using a sample and hold structure, represented by a capacitor and a switch inside a dashed square. But, as mentioned, this test could be used with any implementation of this block.

The test methodology consists in testing each analog branch through the digital structure, already present in the filter. The idea is to reconfigure the switches at the input of the analog delays in such a way that all capacitors in these blocks will sample, in the same time, the same input value, as shown in figure 6(b). Then, each stored value is compared to the same random noise through each comparator, thus generating the same bit stream. It is easy to see that, in the case where any part of one of the analog branch does not works properly, its correspondent bit stream will not equal the other ones, thus disclosing the fault.

In order to compare the test bit streams, a simple exclusive-or operation, already present in the multiplication stage, is done for each two bit streams, and in the case one differs from another, a set of '1s' will appear in the exclusive-or output.

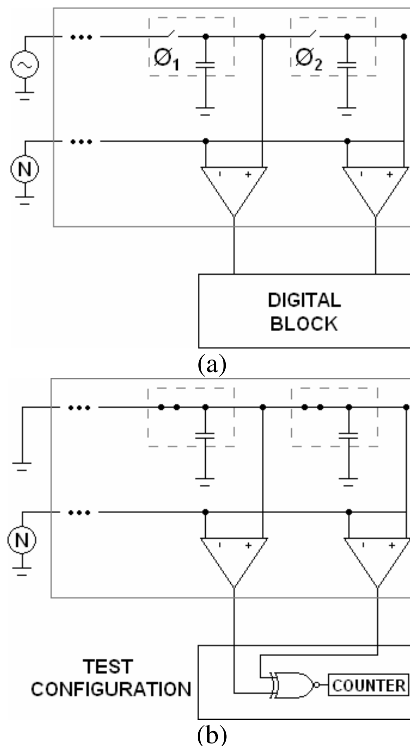


Figure 6. (a) Two taps example of the analog block using a sample and hold scheme and (b) test configuration to find out the faulty tap.

Taking as an example the structure of figure 6(a), let us suppose now that two analog taps are being tested to find out if there is a significative offset in one of the comparators.

First, the test configuration is assumed, as demonstrated in figure 6(b). To test the offset of the comparator, the sample and hold switches are closed and the input signal is set to zero Volt. This way, all comparators will receive the same input, and compare it to the same noise. However, since the comparators have different offsets, the generated bit stream will differ from each other. This difference can be easily measured by counting the numbers of ones in the output of the exclusive-or gate. The simulation result depicted in figure 7 shows how the number of ones in the exclusive-or output varies as the offset of one of the comparator being tested varies from 1uV to 10uV. Only 256 samples were used to count each exclusive-or output.

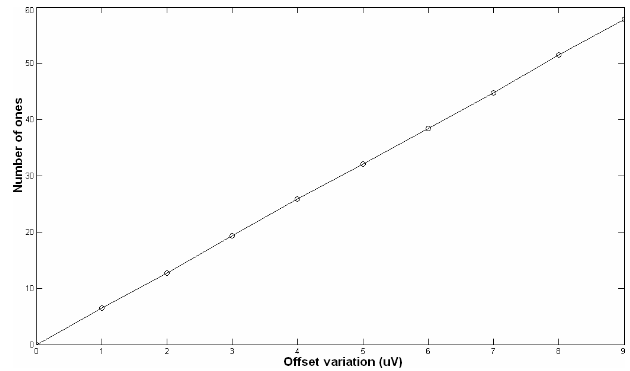


Figure 7. Variation of number of ones in the XOR output as the difference between the offsets of the comparators varies from 0 to 90uV, in steps of 1uV.

Through the use of multiple comparisons between the analog taps, one can also find out which one is out of the specifications values, or does not work properly. For example, if tap number 1 is compared to tap number 2 and there are errors, and then tap number 1 is now compared to tap number 3 and there are no errors, there is a great probability that the tap number 2 has some kind of fault, or does not achieves the desired operation.

Since now it is also possible to identify which tap is out of order, this tap could be possibly substituted by an extra tap, which could be fabricated exclusively to replace the defective ones. Figure 8 shows an example where the second original tap presents some kind of fault and is then replaced by one of the replacement taps, just through some switches settings.

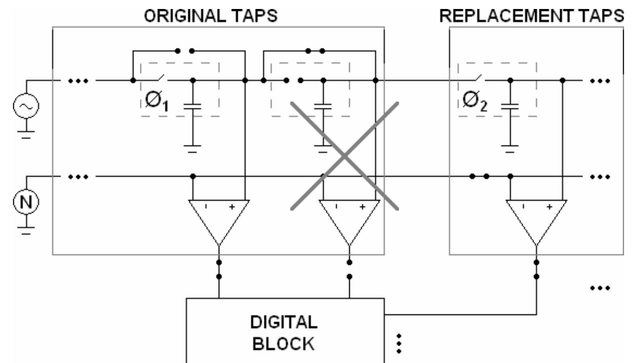


Figure 8. Changing an out-of-specification analog tap by an extra replacement tap.

Thus, as seen, the test methodology for this kind of filter is extremely easy to implement. Since all analog

signals are converted to the digital domain in a very low cost way, and then processed through the use of a structure already present in the system, there is no need of extra testing blocks or complexes operations. Also, since it is possible to easily identify which block is the defective one, this block can be replaced by an identical one, fabricated for this purpose, only through some switches settings.

4. Conclusions

Digital CMOS technology has been continuously shrinking, and the analog design must follow this trend in order to keep the yield and the integration density, thus not letting the gap between digital and analog design increase even more. As a consequence, analog design gets harder to design, mainly for components that have a high dependence with the technology variability.

Specifically, analog filter design presents, in most cases, problems related to capacitance or transistors mismatch, which can lead to a great discrepancy between the desired and the achieved cutoff frequency, for example. Another problem regarding this kind of filter is the cost to test it. In many cases, extra hardware must be added to the design, and different signals must be generated in order to assure good fault coverage.

In this work we have presented the implementation of a mixed-signal FIR filter, which does not use passive components to determine its parameters (cutoff frequency, quality factor and gain). Instead, the input signal is represented in a bit stream fashion through successive comparisons of the sampled analog input signal to a random noise. The filter coefficient is represented by a digital bit stream generated by a digital sigma-delta converter. Since both signals are in the digital domain, all the rest of the processing, including multiplication and addition, is done digitally.

It was shown that it is possible to test both the analog and the digital part, without the insertion of extra hardware, and without generating any extra test signal. A few switches setting put the system in the test configuration, which uses only one exclusive-or gate and a counter. Also, it was shown that it is possible to diagnose a faulty tap in the filter, and easily replace it by a spare one thus increasing the yield of the analog filter.

5. References

[1] Schaumann, R.; Valkenburg, M.E.V., “Design of Analog Filters”, New York: Oxford University Press, 2001. 737p.

[2] Tsividis, Y.; Banu, M.; Khoury, J., “Continuous-time MOSFET-C filters in VLSI”, IEEE Transactions on Circuits and Systems, New York, v.33, Feb. 1986. p.125 – 140.

[3] Nauta, B.A., “CMOS Transconductance-C filter technique for very high frequencies”, IEEE Journal of Solid State Circuits. v.27, feb. 1992.

[4] D. Johns and K. Martin, "Analog Integrated Circuit Design", John Wiley & Sons, Inc. 1997, ISBN 0-471-14448-7.

[5] Oppenheim, Alan V., “Discrete-time signal processing”, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999. 870 p.

[6] Fischer, G., “Analog FIR filters by switched-capacitor techniques”, Circuits and Systems, IEEE Transactions on, v.37, Issue 6, June 1990. p: 808 – 814.

[7] Diaz-Sanchez, A.; Ramiriz-Angulo, J., “ VLSI voltage mode FIR analog adaptive filters”, Circuits and Systems, 1996., IEEE 39th Midwest symposium on, v.1, 18-21 Aug. 1996. p:141 – 144.

[8] Burlingame, E.; Spencer, R., “An analog CMOS high-speed continuous-time FIR filter”, Solid-State Circuits Conference, 2000. Proceedings of the 26th European, 19-21 Sept. 2000. p:288 – 291.

[9] Ciota, Z.; Napieralski, A.; Noullet, J.L., “Analogue realisation of integrated FIR filters”, Circuits, Devices and Systems, IEE Proceedings. v.143, Issue 5, Oct. 1996. p:274 – 281.

[10] de Souza, A.A., Jr.; Carro, L., “Robust low-cost analog signal acquisition with self-test capabilities”, Defect and Fault Tolerance in VLSI Systems, 2004. DFT 2004. Proceedings. 19th IEEE International Symposium on.10-13 Oct. 2004. p(s):239 – 247.

[11] de Souza, A.A., Jr.; Carro, L., “Statistical acquisition for embedded instrumentation”, Instrumentation and Measurement Technology Conference, 2004. IMTC 04. Proceedings of the 21st IEEE. v.1, 18-20 May 2004. p:314 – 317.

[12] Schüler, E., Farenzena, D.S., Carro, L., “Evaluating Sigma-Delta modulated signals to develop fault-tolerant circuits”. Proceedings of 11th IEEE European Test Symposium, 2006. ETS’O6, Southampton, UK, 21-25 May, 2006.

[13] Schüler E., Carro, L., “Reliable Digital Circuits Design using Sigma-Delta Modulated Signals”, 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2005 (DFT’05), 03-05 Oct. 2005, USA.

[14] Norsworthy, S.R.; Schreier, R.; Temes, G. C. “Delta-sigma data converters: theory, design and simulation”. New York: IEEE Press, 1997. 476p.

Functionally Fault-Tolerant DSP Microprocessor using Sigma-Delta Modulated Signals

Erik Schüler[1], Marcelo Ienczczak Erigson[1], Luigi Carro[2]

[1]Electrical Engineering Department – Universidade Federal do Rio Grande do Sul
Av. Osvaldo Aranha, 103 - Porto Alegre - RS, Brazil
[2]Informatics Institute – Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500 - Porto Alegre - RS, Brazil
eschuler@eletro.ufrgs.br, mierigson@inf.ufrgs.br, carro@inf.ufrgs.br

Special issue on Test, Defect Tolerance and Reliability of Nanoscale Devices

Abstract. *The occurrence of soft-faults in digital circuits due to single event upsets (SEU) caused by particle hits has been reported in many works, and it has been claimed that, as the transistor dimensions shrink, multiple and simultaneous faults will be a common scenario in future technologies. Many techniques have been proposed to cope with these kinds of faults, most of them based on hardware or software redundancy. In this work, we present a new paradigm, which is based on signal redundancy, that is, the signal to be processed will contain a certain amount of redundancy, in such a way that, even under the occurrence of multiple faults, the final results will sustain a good resolution for some applications. A DSP microprocessor that uses the technique was prototyped, and some results are presented and compared to typical n-bits binary coded DSP microprocessor architecture, showing the advantages of using the proposed approach.*

Keywords: fault-tolerance, error tolerant system, sigma-delta, single event upset (SEU), Digital Signal Processing (DSP)

1. Introduction

It has been a consensus that CMOS transistor gate length will soon overcome the nanometric barrier, allowing the inclusion of a huge number of these devices on a single die, even more than the enormous integration density shown these days. Nevertheless, it has also been claimed that this integration phenomenon will bring undesirable consequences as well. One of the most critical is the reduction in the circuit node capacitances which, in spite of allowing faster circuits with clock speed reaching hundreds of gigahertz, will also be responsible by the increase in the soft-errors occurrence.

One of the main consequences of soft-error is the Single Event Upset (SEU), caused by a particle hitting a CMOS junction, which can result in a bit flip that can be propagated through all the rest of the circuit operation [1], [2]. Most of the reported SEU effects concern the manifestation of single faults in digital circuits, caused by the inversion of one single bit. However, multiple bits inversion has also been demonstrated in some works [3], indicating that this is an expected scenario for future technologies.

Different solutions have been proposed to cope with the SEU problem, in different abstraction levels: hardening by technology, hardening by design and hardening at the system level. These techniques are further discussed here and, as it

will be shown, they can not completely guarantee the right circuit functionality when multiple and simultaneous faults occur. Also, when the previous techniques come close to ensuring the circuit behavior, the cost becomes excessively high.

Most of the previously proposed schemes are based on hardware or software redundancy, where the hardware block or the software variable is triplicated, and the correct system response is given by majority voting [4] [5]. Looking at the problem from a different perspective, we now propose a new kind of redundancy, based on signal redundancy.

The main idea does not imply in triplicating the signal to be processed, but in creating another way to represent the signal, in such a way that, even under the occurrence of multiple faults, the final response still sustains an acceptable resolution. The technique, which uses sigma-delta modulation [6] to generate the redundant signal, is to be used in digital circuits and, as it will be shown, can imply, in some cases, in smaller and faster circuits. In the cases where area and time penalties proposed by other solutions are smaller than the solution herein proposed, the achieved fault-tolerance becomes a decisive factor to determine the use of our approach.

The basic idea regarding our solution has already been presented in some of our previous works [7], [8]. In this work, to better demonstrate the approach and to increase its application range, a Digital Signal Processor (DSP) that processes sigma-delta modulated signals was developed using VHDL description, and some results for a Finite Impulse Response (FIR) filter, which was programmed in the DSP processor, are presented and analyzed. Also, an identical DSP architecture using n-bits code-modulated words instead of sigma-delta-modulated signals was implemented in order to make comparisons between area, performance and robustness of these two solutions.

This work presents in section 2 a literature review, showing some techniques developed to implement fault tolerant circuits. In section 3, the error tolerant systems are analyzed, opening way to present the signal redundancy approach, discussed in section 4. In section 5, it will be shown that sigma-delta modulated signal can in fact be fault tolerant due to the signal redundancy intrinsically present, and a brief review regarding this kind of signal modulation is presented. Section 6 shows the DSP developed to process sigma-delta signals, and the results obtained after programming a FIR filter with added faults in the system behavior. In the same section, a comparison with an n-bits binary coded word structure is developed, showing the main advantages and disadvantage of using each technique. To complete the achieved results section, some preliminary simulation results regarding the implementation of an Infinite Impulse Response (IIR) filter are presented. To conclude, section 7 presents our final remarks, current and future works.

2. Robustness by Classical Redundancy and Others Techniques

Most techniques used to develop robust circuits use different solutions, which can be divided basically in three different categories, described bellow, all of them actuating in different abstraction levels: technology, design and system level.

In the technology level, different processes are used in the transistor fabrication, such as epitaxial-bulk CMOS, which was first conceived to isolate the analog part from the digital one in mixed-signals designs. In this technique, an extra mask is used to etch a trench from the backside of the wafer all the way to the under-surface of the field oxide [9]. Although the use of epitaxial-bulk is efficient to reduce the Single Event Latch-Up (SEL), it does not mitigate the occurrence of SEU. Another technique is the use of Silicon on Insulator (SOI), where a thin layer of silicon is placed on top of an insulator, such as silicon oxide or glass, and then the transistor is built on top of this layer [10]. This technique

was first developed to be used in memories for space applications, since these memories built on SOI were perceived to be more resistant to SEU. Further studies showed that in order to reduce power consumption and to increase speed in digital circuits, the use of SOI could be a good alternative. This technique, however, requires the use of special fabrication process, with consequent yield limitation, and does not completely mitigate the occurrence of SEU.

Hardening at the design level includes, for example, the use of hardened gate resistor memory cells [11] and hardened CMOS memory cells with feedback structures [12], which imply in physically larger memory blocks, since extra parts must be added to the memory cell. For example, in the gate resistor memory cells, two resistors are built using two levels of polysilicon and, although the impact in the circuit density is small, these resistors are temperature sensitive, increasing the memory vulnerability in low temperatures. Another solution in the design level is the use of codification and decodification of logic blocks, using, for example, Hamming [13] or Reed-Solomon [14] techniques. Hamming Code is an error-detecting and error-correcting binary code that can detect all single-bit and double-bit errors and correct all single-bit errors. The Reed-Solomon code, however, is able to detect and to correct multiple and consecutive data errors. Although these are good solutions, when multiple faults must be corrected, the cost to do so, in terms of time and complexity, makes their use impractical. Also, multiple and simultaneous faults can not be corrected using these approaches. Moreover, the coder and decoder circuits are also sensitive to transient upsets, reducing the overall reliability, even with the added costs.

To protect digital circuits at the system level, the use of hardware or software redundancy techniques are the most known ones. For the hardware case, the use of Triple Modular Redundancy (TMR) [4] rises as the most diffused scheme. TMR in its various implementations simply implies in triplicating the sensitive block and making a vote, where the correct system response is determined by majority. Of course, some problems stand out here. For example, the area and power penalties, which are triplicated as well. The other limitation appears when one thinks in simultaneous faults, that is, if two blocks give wrong responses, by majority, the final response will also be wrong. Also, with less probability to occur, if the fault happens in the voter block, there is no way to define whether the response is correct or not. Using software redundancy, some techniques such as Algorithm Based Fault Tolerance [15] and Code Flow Check are used, as well as variable duplication [5], which can automatically be implemented on the high-level code of the program. This technique, applied to memories and registers, performs two different modifications to the source code; the first one corresponds to duplicating some or all of the program variables in order to introduce data redundancy, and modifying all the operators to manage the introduced replica of the variables. The second source code modification aims at introducing consistency checks inside the control flow to periodically verify the consistency between the two copies of each variable.

As seen, most of the approaches used to develop robust systems regarding soft errors try to correct the wrong response, for example coding and decoding, or try to guarantee that the obtained response is already the right one, as is the case of hardware/software redundancy. Some others, e.g. SOI and special memory structures, pursuit to mitigate the SEU occurrence.

We now present a new perspective, where one does not need to worry whether the fault will occur or not, because even if it occurs, the circuit will be protected. That is, we do not try to mitigate or to correct the circuit response corrupted by the faults. We simply let the fault occur, because thanks to the redundancy already presented in the signal, the consequence of the fault will not be so harmful for the system response. This technique, allied to the idea of error tolerance proposed in [16] and explained in section 3, can guarantee that the system will develop appropriate responses even under the occurrence of multiple and simultaneous faults.

3. Error Tolerant Systems

According to [17], “fault tolerance is the ability of a system to continue correct operation of its tasks after hardware or software faults occur”, while correct operation means that no errors occur at any system output. In [16], it is mentioned that “fault tolerance tries to provide reliable operation in the presence of lifetime faults and/or externally induced transient errors”. This way, it has also been proposed in [16] the following definition of error tolerance when considering systems that can tolerate a certain amount of errors at the entire system’s output: “a circuit can be error tolerant with respect to an application if it contains defects that cause internal errors and might cause external errors, and the system that incorporates this circuit produces acceptable results”.

The very simple results presented in figure 1 can clarify the notion of error tolerant system, and the idea behind minimum resolution required to a certain error tolerant system generating correct output responses. In this example, a digital oscillator must generate a 10-bits minimum resolution sine wave with a certain frequency, offset and phase. This is equivalent to say that the expected Signal to Noise Ratio (SNR) of the output signal must be at least 71dB. Thus, any signal generated with a SNR below this threshold shall not be satisfactory to the system operation. However, if the faults occurring during the signal generation do not cause a perturbation high enough to make the SNR drop below this fixed value, the signal will be properly used by the system. In figure 1(a), different bits, below the sixth one, are inverted each time one point is produced by the oscillator, causing a small perturbation in the signal shape, but still maintaining a SNR higher than 73dB (10.5 bits resolution), thus above the requested 71dB. However, when the bits which are inverted changes to those below the eighth one (see figure 1(b)), the output signal presents a significant variation, causing a 11dB drop in the SNR, leading to a malfunction behavior, since the new resolution now is about 8.5 bits (62dB). So, one can say that, for this very simple circuit, it is error tolerant as long as one can be sure that faults do not occur in bits that are higher than the eighth one.

Another interesting proposition, which also makes use of the error tolerance approach, is the one presented in [18], where a probabilistic-based design methodologies based on Markov Random Fields is examined. According to [18], “the Markov Random Fields approach can express arbitrary logic circuits and the logic operation is achieved by maximizing the probability of correct state configurations in the logic network depending on the interaction of neighboring circuit nodes”. The basic idea is that the computation is realized by propagating states through the circuit in a probabilistically fashion, assuming that a large number of nano-devices is presented, thus requiring very low power operation with consequent probabilistic behavior, since transistor will be operating near the thermal limit. Nevertheless, since the idea presented in [18] supposes a huge number of transistors presented in the digital circuit, the solutions make use of many of these transistors to construct a simple logic gate. For example, a simple CMOS inverter, which uses only two transistors, is now developed by using 20 of these devices, while a NAND gate is conceived through the use of 60 MOS transistors, instead of four as the standard one.

With the idea of error tolerance in mind, one can find a lot of applications that contain the property of being error tolerant, that is, applications in which the final response can still be correct, even after the insertion of a large amount of faults, in our case, transient faults caused by radiation or noise interference. Also, as seen in section 2, redundancy is one of the most used techniques to cope with fault occurrence. Based on that, we now propose a new paradigm, which is based on signal redundancy. The idea behind this solution is not based, for example, on triplicating the n-bits words of the input

signal, but rather on creating a new version of the input signal, representing it in a redundant fashion through the use of another signal domain representation. The next sections describe the methodology used to create this signal, and give some results that support our decision in using such signal depiction.

4. Signal Redundancy Paradigm

As shown before, the use of redundancy is very common when fault-tolerance must be achieved by a system. Besides the already mentioned TMR in hardware redundancy, and variable duplication in software redundancy, others techniques may be used, such as space and/or time redundancy [19], where the concomitant use of a self-checking combinational circuit and a state-preserving element is employed. Now, based on the idea of redundancy, we propose changing the paradigm, where the redundant element is intrinsic to the data that will be processed. Although this may seem similar to a classical redundant approach, here we do not duplicate the data, but rather we represent information in another domain, which can still be processed in a digital like manner.

The basic idea is to take a single n-bits Pulse Code Modulated (PCM) word, thus in the digital domain, and change the way this word is represented. This change is such that the n-bits word now becomes an m-bits word, where $m>n$, and it is no more a digital representation of the data, but a redundant representation, which can still be processed by digital circuits, as it will be shown in section 5.

There are different ways to obtain redundant signals. For example, as represented in figure 2(a), one can simply compare an n-bits word with an n-bits random noise, and the output n-bits words will be a representation of the input value [20]. A simpler solution would be based on the same principle, but the signal comparison now generates a 1-bit bit stream, containing the input signal probabilistic representation [20], as shown in figure 2(b). However, in order to obtain a good resolution to represent the input signal, the output bit stream length must be in the range of thousands of bits, as demonstrate some results using this kind of signal representation in [21].

A better way to produce signals with intrinsic redundancy, capable of representing large bit words, but with a low implementation cost and a much smaller bit stream, is by using sigma-delta modulation [6], [22]. With this kind of signal generation, one can represent signals with a resolution larger than 20 bits, with lots of redundancy by using a very simple scheme. It is important to emphasize that, although sigma-delta modulators are most commonly found in analog-to-digital converters (using analog sigma-delta modulators), what we are proposing here is the use of digital sigma-delta modulators to transform an n-bits digital word into a sigma-delta representation of the same signal. The modulator output signal, represented by a sequence of zeros and ones, will then be used in the subsequent application. An explanation regarding digital sigma-delta modulators, how they work and which kind of signals they generate is presented in section 5.

To conclude the idea of signal redundancy and show the advantages of using a sigma-delta representation instead of, for example, the previously mentioned technique of signal/noise comparison, an example comparing these two kinds of signal representation is presented in figure 3. Here, a 10 KHz sine wave is sampled with an Over Sampling Ratio (OSR) equal to 64, that is, it is sampled with a frequency 64 times higher than the signal Nyquist frequency. Firstly, the sampled signal is represented by a bit stream produced through the comparison of each sampled point to a uniformly distributed random noise, as presented before in figure 2(b). On the other hand, the same sampled signal is now modulated through the use of a first-order sigma-delta modulator, generating another bit stream, also representing the input signal. As one can

see, for the same OSR, thus for the same output bit stream length (128 bits per signal period in this case), the achieved SNR to the signal comparison case is approximately equal to 19dB, while for the sigma-delta modulation case, the SNR almost reaches 39dB. So, as mentioned and now demonstrated, the use of sigma-delta modulation generates more accurate results with the same number of bits in the bit stream.

The next section presents some mathematical and simulation results, which show that the use of signal redundancy, more specifically signals generated with sigma-delta modulators, are in fact fault tolerant, allowing one to use these signals to develop error tolerant systems. Also, a brief review regarding these kinds of generators is presented to give support to our decision in using such approach.

5. Sigma-Delta Modulation and Its Use in Redundant Signal Generation

The main utilization of sigma-delta modulation is in Analog-to-Digital Converters (ADC), in the manner presented in figure 4(a), where the analog input signal is over-sampled, converted to a one-bit representation through the analog sigma-delta modulator, and then down-sampled (or decimated) in order to obtain a digital representation of the analog input signal. This kind of signal modulation can produce converted signals with high resolutions, by using small sampling ratios when compared to others over-sampled converters [22]. Others applications to sigma-delta modulators are in the generation of test-signals [23], signal-processing [24] [25] and, of course, Digital-to-Analog Converters (DAC). In the DAC case (see figure 4(b)), the digital input signal is over-sampled, converted to the sigma-delta domain through a digital sigma-delta modulator, and finally low-pass filtered to generate the analog signal representation. A brief explanation of how a sigma-delta modulator works and which kinds of signals are generated after the modulation are presented next.

5.1. Sigma-Delta Modulation

The sigma-delta modulator is the main block in a sigma-delta ADC or DAC (see figure 4), since this is the block that will pass the over-sampled input signal from the analog/digital to the sigma-delta domain for further filtering through a digital/analog low-pass filter, and consequent conversion to digital/analog representation. Consider the generic first-order sigma-delta modulator represented in figure 5(a). The structure, known as an error feedback structure, consists of four basic blocks: an input subtractor, an integrator, a quantizer and a feedback gain. The basic idea is that the quantization error produced by the two-level quantizer is fed back to the circuit input, and subtracted from the input signal. The difference between the input signal and the fed back quantization error is integrated, and then quantized to generate an output represented by a one-bit bit stream, whose mean value is equal to the mean value of the input signal [6].

Analyzing this kind of circuit by the frequency point of view allows one to understand why one can obtain high resolutions using such a simple circuit. To do that, some considerations must be done regarding the modulator characteristics, mainly, the quantization noise generated by the two-level quantizer. For more details about these considerations, the reader is encouraged to consult references [6] and [22]. At this point, in order to present a brief analysis of how the modulator frequency response behaves, we assume that the modulator presents the following characteristics regarding the quantization error: the quantization error is largely uncorrelated from sample to sample to the input signal, and has equal probability of lying anywhere in the range ±D/2, where D is the quantization level amplitude. This way, the

quantization error can be represented by a noise Q(s), as shows the model presented in figure 5(b). Analyzing the transfer function of this model from the signal input X(s) and from the quantization noise input Q(s), one has:

$$\frac{Y(s)}{X(s)} = \frac{1}{s+1} \quad \text{Signal Transfer Function (STF)} \qquad (1)$$

$$\frac{Y(s)}{Q(s)} = \frac{s}{s+1} \quad \text{Noise Transfer Function (NTF)} \qquad (2)$$

Analyzing equations (1) and (2), one can note that the signal transfer function (STF) is a low-pass function, while the noise transfer function (NTF) is a high-pass function. As a consequence, the output signal will be represented by the input signal in low-frequencies plus the quantization noise in high-frequencies, as shows figure 6, which presents a sinusoidal signal modulated by a sigma-delta modulator with a certain sample frequency. As noted, due to this particular response, one can achieve high signal to noise ratios, which can be increased by increasing the modulator over sapling ratio or the modulator order.

To conclude the discussion regarding sigma-delta modulators, it is interesting to note that, if one desires to implement an analog modulator, thus an analog version of the blocks in figure 5(a) must be constructed, and the most common technique used to do it is a switched-capacitor implementation [26]. In this case, the feedback gain is a one-level DAC, which simply feeds back a positive or a negative voltage value, depending on the value of the output bit stream. In our case, we want to make a digital modulator because the input signal is digital and we want it to be converted to a sigma-delta representation. So, the modulator of figure 5(a) must be implemented even in hardware or software. In figure 5(c) an example of how this simple modulator could be implemented in hardware is presented. Here, the quantizer is substituted by simply taking the integrator sign, and this sign will choose whether an addition or a subtraction of the input value and a constant value is done, substituting, thus, the feedback gain, whose function is to convert the output 1-bit signal to an n-bits value compatible with the input signal.

We now demonstrate that the use of sigma-delta signal modulation can be useful to develop fault tolerant systems, and some results show that even under multiple fault occurrence, the final signal resolution can still produce coherent values.

5.2. Redundant Signal Generation through Sigma-Delta Modulation

The intrinsic redundancy of sigma-delta modulated signals comes from the fact that the bit stream carries the original signal representation plus a certain quantization noise, as showed in section 5.1. In another point of view, one can imagine that the original n-bits words are now represented in a sequence of Least Significative Bits (LSB), since the two-level quantizer generates a 1-bit bit stream as the modulated output. So, the inversion of many of these bits shall not interfere so much in the signal resolution.

Figure 7 schematically presents our proposed approach, where sigma-delta signal modulation is used to produce data values containing an amount of redundancy able to tolerate a large number of transient faults. As noted, for an n-bits PCM data representation, if one single bit inversion occurs, the consequences for the final system response may cause a complete mismatch between the expected response and the obtained one. On the other hand, if one considers an error tolerant scheme using a data redundant signal representation, even after the inversion of many bits (simultaneously or not) the final response will be much closer to the expected one.

Something important to be noted here is the difference between faults that occur in the sigma-delta modulator and faults that occur in the system that is using these modulated signals to generate the final response. First, let us investigate what happens when faults occur inside the modulator, during the modulation process, and the consequences of these faults to the output bit stream, and, consequently, to the system response.

Looking at the modulator depicted in figure 5(c), one can see that there are basically two blocks where faults can occur. They are the modulator input adder and the modulator integrator adder. Since we are dealing with transient faults, which could possibly invert one or more bits, we can model this bit-flip, occurring in an n-bits word inside the modulator, by a simple addition (or subtraction) of a certain amount in this word magnitude, during an instantaneous time-period. That is, this fault injection can be modeled as a Dirac Delta Function, in such a way that:

$$y[i] = x[i] + k.\delta[i - \tau],\ k \leq 2^n \quad (3)$$

Where 'x' is the original value (fault-free), 'k' is the magnitude of the amount added due to the bit inversion and 'τ' is the time when the fault occurs. For a first-order modulator, as the one in figure 5(c), the addition or subtraction of a certain amount in a specific sample during the modulation process can result in a great variation of the output bit stream, causing the inversion of more than one bit in this bit stream. Although many works have presented very powerful mathematical approaches to model sigma-delta modulators, this is not our goal here, so some simulation results will be presented to show what happens when bits are inverted inside a digital sigma-delta modulator. A deeper explanation considering what happens when faults occur in sigma-delta modulators can be found in [27], where a mathematical approach and an analytical model for a faulty modulator are presented.

If one observes figure 5(c), it can be noted that there is only one way in which a fault can appear inside the signal band, and that is through a non-expected inversion of the integrator sign bit, which is the only signal fed back to the circuit input. So, it does not matter where the fault occurs, but only if this occurrence causes a high enough addition (or subtraction) in the integrator output value in such a manner that the integrator sign bit suffers an inversion when it should not to suffer it. Consider, thus, simulation results presented in figure 8, where the modulator of figure 5(c) was simulated in Matlab®, and results of fault injection in the different blocks of the circuit where acquired. The input signal, a digital 20 KHz input sine wave, was modulated using an OSR of 32. Figure 8 also presents the modulator integrator output plotted versus its non-faulty behavior, showing how the number of inversions in the integrator sign bit affects the modulated output signal.

In figure 8(a) one can see that, due to the small number of inversions in the integrator sign bit (figure 8(b)), there is also a small variation in the output bit stream, what can be verified by the tiny variation at the final SNR (from 30.84 to 30.09). In this case, the 2nd least significative bit of the input adder is inverted once during the modulation process. However, when the inversion occurs at the most significative bit (MSB), also in the modulator input adder, a higher number of inversions happens (figure 8(d)), and also the output bit stream will suffer a great mismatch from the faulty-free response, as seen in figure 8(c), where the SNR suffers a variation from 30.84 to 24.30. It is interesting to see that, although the probability of occurrence of faults in such a small circuit is low, if it happens, one single fault in a specific bit may cause the inversion of a huge number of bits in the output bit stream, determining a significant degradation in the final Signal to Noise Ratio.

Consider now that the fault does not occur inside the modulator, but in the application that is using the modulated signal, that is, the faults will cause an inversion in the output bit stream, already outside the modulator. It is expected that,

due to the redundancy presented into this signal, even with the inversion of many bits, the final result still sustains a large enough resolution to produce correct results. This can be observed in the results presented in figure 9, where two 9-bits constant values are added through interleaving operation, after being modulated to the sigma-delta domain using the modulator of figure 5(c). The interleaving operation is a simple way to add one or more bit streams [24], and can easily be achieved by the simple circuit depicted in figure 10, which implements a simple 2:1 multiplexer.

The final bit streams are then decimated in order to analyze how far the addition results are from the expected value. The decimation process [6] is the final step in an analog to digital conversion using sigma-delta modulation, where the sigma-delta bit stream is converted to an n-bits digital representation. In our case, it is used just in order to analyze the obtained values, but in a real application, the idea is that the bit stream will be used all through the process and, if necessary, be decimated only in the final stage. A common way to develop a decimation operation is through the use of a digital *Sinc* filter for example [6], whose hardware implementation example is presented in figure 11(a). This filter, also known as accumulate-and-dump, simply averages the output signal, generating a digital representation of the input signal. The way the filter works is outside the scope of this work, but the basic idea is that the over-sampled bit stream is down-sampled to generate its digital representation, in such a way that the filter cut-off frequency is appropriately designed to filter the out-of-band quantization noise, as figure 11(b) demonstrates.

In our addition example, each 9-bits word is modulated in sigma-delta with a certain OSR, thus generating OSR bits in each output bit stream. No fault is injected during the modulation process, but 10 bits are inverted during the bit stream addition operation. Since these bits are randomly inverted, and thus a different sequence of ‘1’ to ‘0’ or ‘0’ to ‘1’ inversions can occur, the addition was made 200 times to evaluate the consequences of inverting bits in different positions of the bit stream.

As seen in figure 9(a), where the OSR is 32, there is a great deviation from the expected value, for almost all additions, what takes to a final mean value of 72.63 and a standard deviation of 20.44, thus far way from the expected values, which would be 110 (addition of 50 and 60) and 0, respectively. However, when the OSR is increased from 32 to 64, 128 and 256 (figures 9(b), 9(c) and 9(d), respectively), also the mean values of the additions increase from 72.63 to more than 105, and the standard deviation decreases from 20.44 to less than 3. So, since the redundancy of the operands is increased, also the obtained results resolution is increased, coming closer to the expected value.

As demonstrated so far, the use of sigma-delta modulation is a good alternative to be used in digital circuits development, whose fault tolerance must be increased due to their application in critical systems, or because they are to be used in radioactive environments, for example. The next section presents the development of a DSP microprocessor used to process sigma-delta modulated signals and, as it will be shown, more complexes digital applications can be implemented using the proposed approach, guaranteeing its robustness against bit flips.

6. Functionally Fault Tolerant DSP Microprocessor

Different digital signal processing functions can be easily developed through the use of DSP microprocessors, which are dedicated microprocessors able to realize many operations, like simultaneous memory access and Multiply and Accumulate (MAC) operations, in a single machine cycle. Through software programming, basic signal processing blocks

like Infinite Impulse Response (IIR) filters, Finite Impulse Response (FIR) filters and Fast Fourier Transform (FFT) computation can be developed to be used in different applications involving, for example, audio and video.

In order to increase the robustness of these programmable devices, and presenting more complex applications using the proposed technique herein described, a DSP microprocessor specifically developed to deal with sigma-delta modulated signals was described in VHDL, and the results are presented next. A comparison with a standard DSP microprocessor using n-bits code-modulated words is also presented, pointing out area and performance differences between these two implementations. As the section title proposes, this architecture is supposed to be functionally fault tolerant, that is, the processor must generate acceptable results, even under the occurrence of multiple faults. Acceptable values, as already mentioned, are values which will generate a final response that still performs the desirable system specifications.

6.1. Sigma-Delta DSP Microprocessor Description

The DSP structure is based on the Analog Devices ADSP2100 [28], which has a relatively simple structure, composed of Multiply and Accumulate block (MAC), an Arithmetic Logic Unit (ALU) and a shifter block, besides the 16-bits bus and data addresses generators. Also, its instruction set contains no more than 19 commands.

The Sigma-Delta DSP (SDDSP), however, presents some modifications, since many Boolean operations developed in the digital domain do not work in the same way as in the sigma-delta domain [24] [25]. For example, to add two n-bits digital values, one must make an exclusive-or operation between each bit of the input values and the carry-out bit, which is also calculated through other Boolean operations. To add two sigma-delta bit streams, however, a simple interleaving operation, as already described in section 5.2, is carried out.

Figure 12 shows the SDDSP internal structure, which has an ALU, a shifter block, some control signals and two internal RAM memories, one for data and one for program. The instruction set comprises 27 instructions, including the 19 from the ADSP2100, plus some extra instructions used to process the sigma-delta bit stream. The whole structure was described in 870 lines of VHDL, and prototyped using an Altera ACEX1K family EP1K100QC208-3 FPGA, occupying a total of 1836 logic elements, representing 36% of the total available in the FPGA.

As noted in figure 12, there is not a specific MAC block in the circuit. This can be explained by the fact that, when dealing with bit stream representations, a multiplication operation can not be done in the same way as it is done with digital values. Instead, a simple AND operation would be necessary, since each bit stream is represented by a sequence of zeros and ones [24]. However, this is not an efficient way to make a bit stream multiplication. To understand that, it must be taken into account that, as mentioned before, a bit stream contains the input signal plus a quantization noise. So, if a multiplication were done, it means that not only the signal is multiplied, but the noise is multiplied as well, as equation (4) shows:

$$(S_A + N_A).(S_B + N_B) = S_A.S_B + S_A.N_B + S_B N_A + N_A.N_B\,, \qquad (4)$$

where 'S_A' and 'S_B' are the input signals being multiplied, and 'N_A' and 'N_B' their respective quantization noise. As a consequence, the noise is spread all through the multiplication spectrum, because there will be four convolution terms in frequency: between signals, between signals and noises, and between noises. This last one leads to a white-like noise, which will contaminate the whole spectrum. However, there are many DSP operations that depend on signal multiplication

so, as it will be shown in the FIR and in the IIR implementations, others techniques can be used to make the multiplication of bit streams, without incurring in noise spread.

6.2. Practical Results

To evaluate the DSP functionality, a 16 taps FIR filter was programmed, where both the input signal and the coefficients are modulated in sigma-delta. These signals were generated in Matlab® and then saved in the DSP data memory. To simulate dynamic faults occurring in the DSP processor, faults were injected during each coefficient modulation process through the inversion of a different number of randomly chosen bits. As already seen in section 5.2, a single inversion in one bit during the modulation process can incur in a significant number of bits inverted in the generated bit stream. Since this bit stream will be used in the DSP processor, this is a good strategy to evaluate the consequence of multiple faults within the processor itself.

The filter structure is composed of simple XOR gates, which will pass the value of the filter coefficient if the input signal is '1', or the negate coefficient value otherwise, as shows figure 13(a). The XOR outputs are added through an interleaving operation, and one filtered point is obtained. In order to acquire the filtered results in a manner able to be evaluated, that is, not in the sigma-delta domain, but in the digital domain, a decimation block was added just after the interleaving operation. The filtered points were acquired using an Agilent Infiniium oscilloscope at sample rate of 1MSa/s. The values are then analyzed in Matlab®. The input signal is a digital 200Hz single tone with white noise added, modulated in sigma-delta with an OSR of 64, generating a total of 1024 bits in sigma-delta representation.

Figure 14 shows different responses obtained after filtering the input signal using the described filter with different OSR used to modulate each coefficient. A comparison between a fault-free and a faulty behavior is presented, both using sigma-delta modulation directly acquired from the DSP microprocessor. In figure 14(a), an OSR of 16 is used in each coefficient. In this case, two faults are injected during the modulation of each coefficient, both in the modulator input adder and integrator, resulting in a total of four faults per coefficient. Remember that, since these faults were injected during the modulation process, a higher number of bits were inverted in the generated bit stream, which represents each filter coefficient. As noted in figure 14(a), the difference between the fault-free and faulty response is practically null, proved by the proximity between their SNR. Moreover, this difference can be reduced through the increase of the OSR used to modulate the coefficients, as presented in figure 14(b), where an OSR of 32 is used, also with four faults injected during each coefficient modulation. Finally, increasing the OSR to 64 and the number of faults to eight per coefficient, the final response still matches the faulty-free response, as shows figure 14(c). It may seem strange that, while improving the OSR from 32 to 64 the final SNR decays, but since the number of faults has also been increased, this is an expected behavior.

In order to analyze how the filtered signal resolution is improved by the increase of the OSR used to modulate each coefficient value, figure 15 depicts a Matlab® simulation result where the number of faults in the output bit stream is fixed, and the OSR used to modulate each coefficient is varied from 8 to 128. The Spurious Free Dynamic Range (SFDR) of the filtered signal is measured, and then plotted versus the OSR value. As noted in figure 15, with an OSR of about 64 or more, one can obtain a constant resolution, even under the occurrence of four faults.

The next section presents some preliminary simulation results regarding the implementation of an IIR filter and, as it will be shown, this kind of filter can also be easily developed by using sigma-delta modulated signals. However, as results

will demonstrate, other resources must be used in order to develop a more robust IIR implementation, since the sensitivity to coefficients variation presented by these filters is extremely high.

6.3. Simulation Results for an IIR Filter with Sigma-Delta Modulators

Another type of digital filter commonly implemented in DSP processors is the Infinite Impulse Response (IIR) filter. An example of a Direct Form II implementation is presented in figure 16(a). Note that, unlike the FIR filter (see figure 13(b)), the output of an IIR filter depends on both the previous inputs and the previous outputs. This feedback mechanism is inherent in any IIR structure, being responsible for the infinite duration of the impulse response.

An easy way to implement the filter presented in figure 16(a) by using sigma-delta modulated signals would be the one presented in figure 16(b), where the multipliers were substituted by XOR gates, and the addition operation is implemented through interleaving. In order to implement the filter, a decimation step was included in order to change the 1-bit representation in the filter left-arm output, by an n-bits representation, allowing, thus, the addition with the n-bits input signal. The filter output is also decimated, but now just in order to have the results analyzed. This filter was implemented and simulated in Matlab®, and some results were acquired and are analyzed next.

As mentioned before, the main structural difference between a FIR and an IIR filter is that the IIR presents a feedback structure. Since the IIR computes its output using the input values and the previous output values, some consequences rise, being one of the most important, the coefficient quantization sensitivity [29]. This can be explained, in a few words, by the fact that, when the output is not computed perfectly and is fed back, the imperfection can accumulate and completely modify the filter response. This effect can be seen in simulations results presented in figure 17 and 18. In figure 17, one can see how a variation affects a FIR filter response. As noted in figure 17(a), after modulating the coefficients in sigma-delta and decimating them in order to analyze the filter response, the coefficient quantization process causes a small variation in the filter response (figure 17(b)). On the other hand, for an IIR filter, even for a very small variation in the coefficients values (figure 18(a)) the consequences for the filter response are disastrous (figure 18(b)), causing a total mismatch between the expected and the acquired values. Thus, due to this coefficient quantization effect, this kind of filter must be implemented by using another topology. For the topology presented in figure 16, one can conclude that it is not error-tolerant, and thus, should not be used with the proposed approach. However, other topologies have been proposed in order to overcome the sensitivity problem, such as the use of cascade structures to implement higher-order IIR filter. By now, results for these topologies and others are part of our current works.

As mentioned before, a standard DSP processor using PCM words was also implemented in order to make some comparisons between both implementations. The next section shows the DSP structure, as well as some comparisons regarding fault tolerance, area and performance penalties.

6.4. ΣΔ *versus* Code-Modulated Word Comparison

Developing a new system structure requires some comparisons with standard systems, which realize the same function as the new one. This way, we must compare the sigma-delta-based DSP microprocessor to a standard implementation of the processor, that is, a DSP that uses n-bits PCM words, in order to evaluate some performance measurements like area and processing time. In fact, this DSP was the base used to create the sigma-delta version, so its structure is almost the same, with few modifications, as shows figure 19.

The system is based on three blocks: an Arithmetic Logic Unit (ALU), a Multiply and Accumulate block (MAC) and a Shifter block. As noted, the most evident difference (see figure 12 for comparison) is the presence of the MAC block, which is now necessary to make multiply and accumulate operations with the n-bits input words. All structure, such as the sigma-delta version, has 16-bits and was prototyped using the same Altera ACEX1K family EP1K100QC208-3 FPGA, occupying a total of 2330 logic elements, representing 46% of the total available in the FPGA.

To validate the DSP, the same FIR filter was implemented, now with the classical requirements for a digital filter that is, using n-bits multiplications and additions operations. This filter structure is presented in figure 13(b). The filtered points were also acquired using an Agilent Infiniium oscilloscope at sample rate of 1MSa/s. The values are then analyzed in Matlab®. The input signal is the same digital 200Hz single tone with a white noise added, sampled with an OSR of 64, generating a total of 1024 16-bits PCM words. Figure 20 presents practical results obtained from the DSP, when implementing the FIR filter. In figure 20(a), it is represented the filter impulse response when no fault is injected in the coefficients, and when one single bit is inverted in two different coefficients (*b1* and *b10*). Note that, compared to the sigma-delta implantation, the number of injected faults is much smaller, but their consequences to the filter response are much more severe. Figure 20(b) depicts the filter coefficients, showing how faults were injected in coefficients *b1* and *b10*. Finally, figure 20(c) shows the filter output after being converted to the decimal representation. One can clearly observe the consequences of a single fault in the filter response, which, in this case, reduced the attenuation of the filter rejection-band in almost 40dB.

To complete our exposition, table 1 summarizes some comparisons regarding both DSP processors implementations. The implementation in two different FPGA, an ACEX1K and a Cyclone one are presented. It is important to notice some interesting aspects here: although they have almost the same internal structure, the sigma-delta implementation presents a small gain in the occupied area, since it does not need to implement, for example, the standard area consuming multiplication operation, at least not in the same fashion as the normal processor. For the same reason, there is a gain by a factor of two in the performance aspect. Of course, this gain will decrease, certainly being also inverted, whenever more robustness and/or resolution are required. In this case, the OSR for the input signal or the coefficient realization will have to be increased, reflecting in a decrease in the sigma-delta DSP performance. But this may be an acceptable compromise when a high fault tolerance must be achieved, mainly in critical parts of a system, or even for the whole system.

Note also that, although presented in table 1, the cost to make the conversion of the digital signals to a sigma-delta representation is not computed in the area value, neither in the processing time. This can be explained by the fact that this signal conversion can be done inside the system by a sigma-delta converter, or one can supposed that the whole system already works with sigma-delta signals, thus with no need for modulators inside it.

7. Final Remarks

Based on the idea of error tolerant applications and redundancy, a new proposition in the development of digital fault-tolerant circuits was presented. Instead of trying to mitigate the SEU occurrence through technology enhancement, or to correct the circuit response through costly HW or SW redundancy, the use of signal redundancy was proposed, in such a way that, even if multiple faults occur, the system response can still sustain a good resolution, able to produce correct responses.

The use of digital sigma-delta modulators to convert an n-bits PCM word into an OSR bits sigma-delta bit stream was demonstrated, and its fault tolerance was evaluated. As demonstrated, even under many faults occurrence, just through increasing the system over sampling ratio, it is possible to enhance the system robustness, obtaining betters resolutions and more precise responses.

To demonstrate the technique feasibility, a Digital Signal Processor (DSP) was developed using VHDL description. The DSP, specially designed to process sigma-delta modulated values, was programmed to implement a 16 taps FIR filter and, as demonstrated, even under the occurrence of multiple faults, the system response matches very closely the fault-free behavior. When compared to standard n-bits architecture, the DSP using sigma-delta modulated signals not only present a much higher fault tolerance, but can also present better performance. Of course, when higher robustness is required, a trade off is established, thus the performance penalty must be increased as well. In the area comparison, for a FPGA implementation, both solutions presented an equivalent area, but if memory structures must be taken into account, for larger memory space, the solution with sigma-delta processing will certainly present a larger area. Also, some preliminary results for an IIR filter were presented, but due to the extremely high sensitivity presented by this kind of filter, other topologies must be studied in order to implement a more error-tolerant version for the structure.

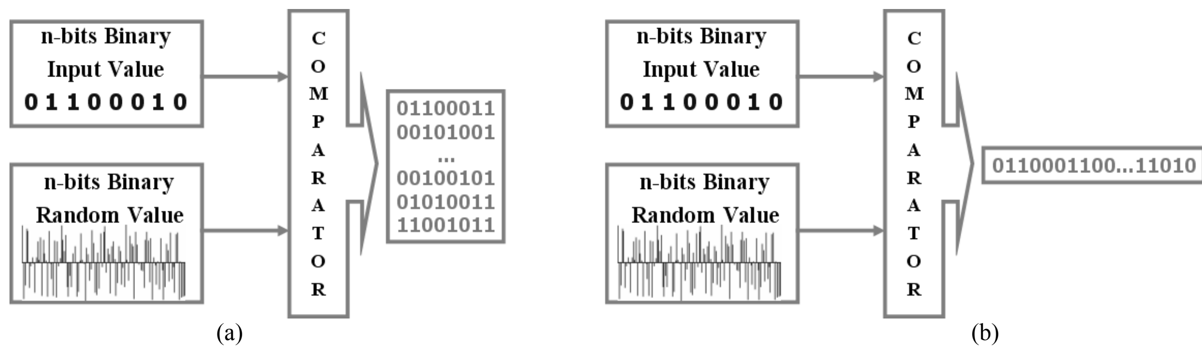
Currently, some works are being developed, such as the research for new IIR topologies, beyond efforts to provide new functions within the DSP, like a fault-tolerant FFT algorithm, which does not use digital values during its computation. Finally, future works include the development of more complex signal-processing approaches, which must be implemented in order to evaluate the robustness of the system when a whole application must be developed, and compare the sigma-delta implementation of these applications with their classical digital counterparts.

Figures, Captions and Tables:

(a)

(b)

Figure 1: Error tolerant signal generation. In (a), resolution is compatible with system requirements, while in (b) faults make resolution drop below 10 bits.



(a) (b)

Figure 2: Redundant signal generation through signal/noise comparison. In (a) an n-bits word stream is generated, while in (b) an 1-bit bit stream is achieved.

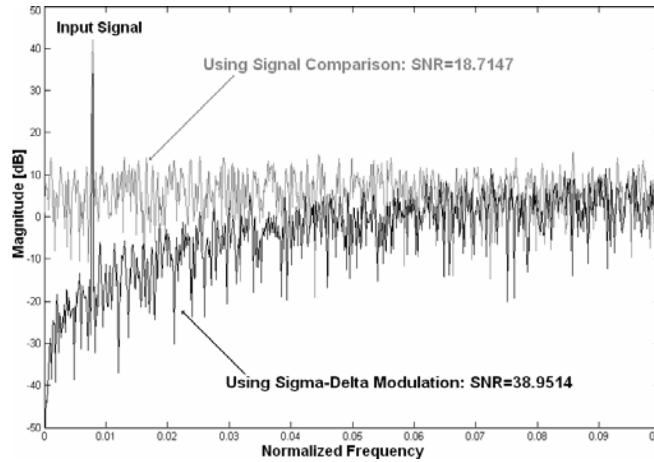


Figure 3: Comparison between two different ways to generate redundant signals: through noise/signal comparison and through sigma-delta modulation.

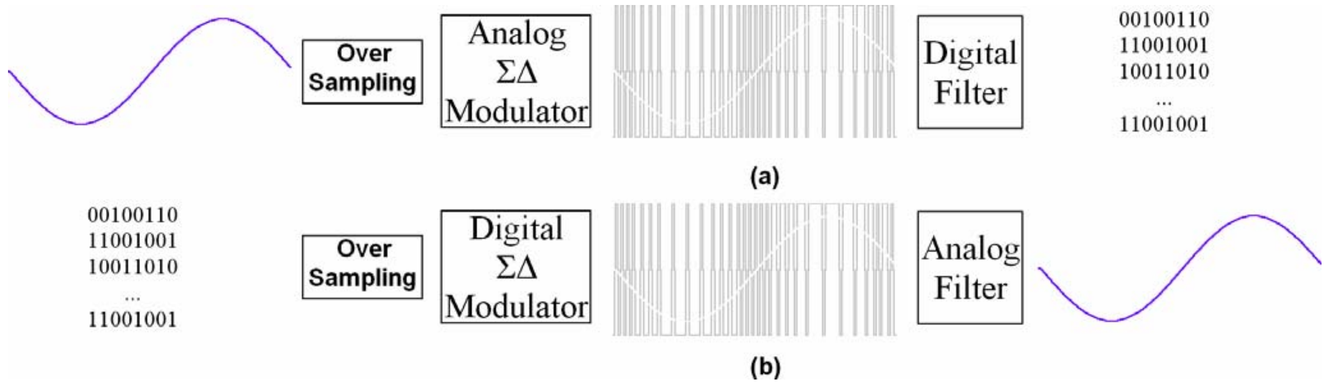


Figure 4: (a) Analog to Digital conversion and (b) Digital to Analog conversion with sigma-delta modulators.

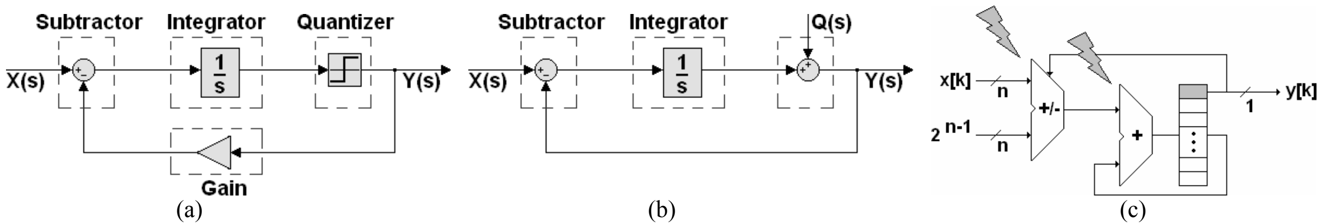


Figure 5: (a) First-order sigma-delta modulator implementation, (b) quantization noise-model and (c) example of hardware implementation for a digital modulator with faults being injected in the input and in the integrator adder.

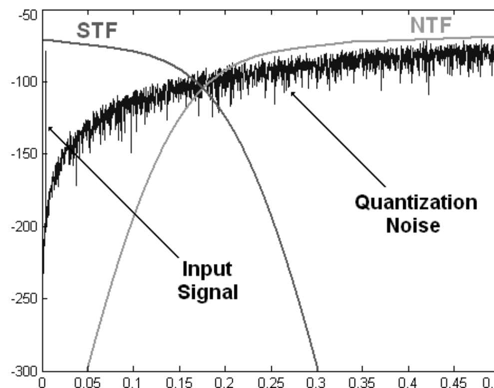


Figure 6: Signal Transfer Function (STF) and Noise Transfer Function (NTF) for a sigma-delta modulator: low-pass and high-pass to achieve high SNR.

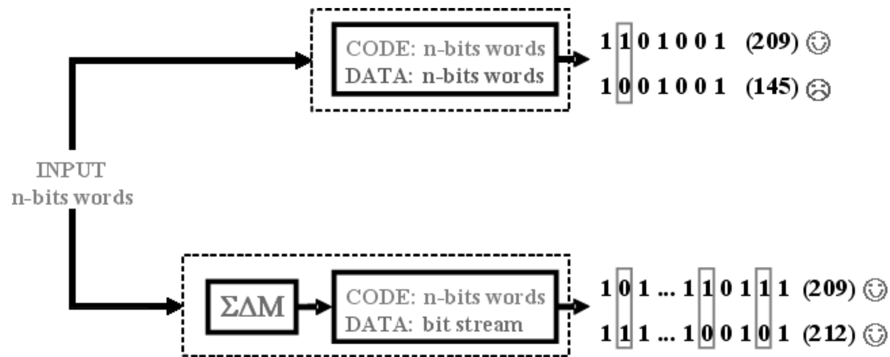


Figure 7: Signal redundancy to be used in error tolerant systems: sigma-delta modulation generates the redundant signal.

Faut-free: SNR=30.84
Faulty: SNR=30.09

(a)

Points without inversion
Points with inversion

(b)

Faut-free: SNR=30.84
Faulty: SNR=24.30

(c)

Points without inversion
Points with inversion

(d)

Figure 8: (a) Output bit stream frequency response when a small number of integrator sign bit inversion occurs (b), and (c) output bit stream frequency response when a high number of integrator sign bit inversion occurs (d).

OSR=32
○ Obtained Result (Mean=72.63, Std=20.44)
— Expected Result (Mean=110.0, Std=0.0)
Addition Result
Addition Number
(a)

OSR=64
○ Obtained Result (Mean=92.43, Std=11.02)
— Expected Result (Mean=110.0, Std=0.0)
Addition Result
Addition Number
(b)

OSR=128
○ Obtained Result (Mean=101.24, Std=5.45)
— Expected Result (Mean=110.0, Std=0.0)
Addition Result
Addition Number
(c)

OSR=256
○ Obtained Result (Mean=105.17, Std=2.97)
— Expected Result (Mean=110.0, Std=0.0)
Addition Result
Addition Number
(d)

Figure 9: Addition through interleaving of two sigma-delta modulated bit streams, with faults injected in the addition process. As the OSR increases, also the final resolution and the fault tolerance increase as well. No fault is injected during modulation process.

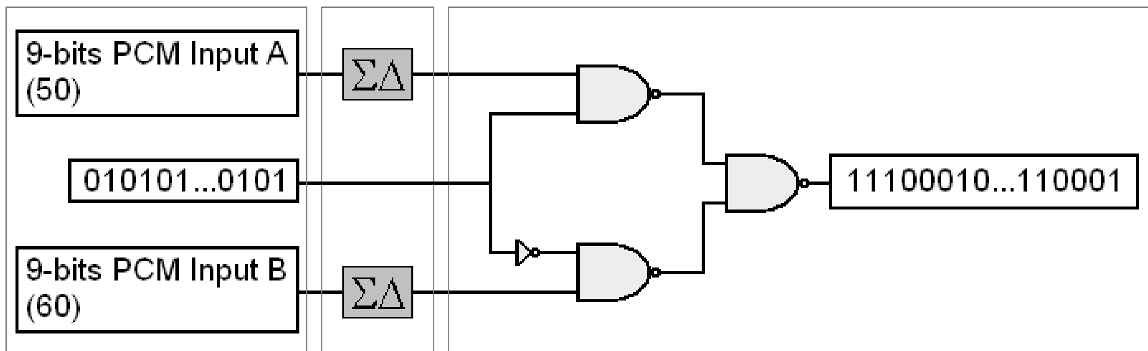
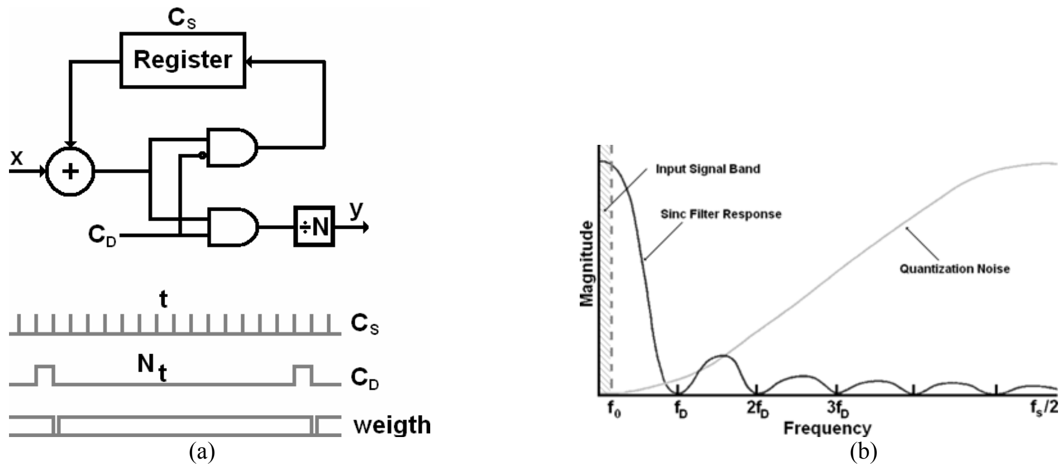


Figure 10: Interleaving of two bit streams: the addition can be done by using only four gates (3 NAND plus 1 INV).



(a) (b)

Figure 11: (a) Hardware example to implement a digital *Sinc* filter, whose impulse response is shown in (b).

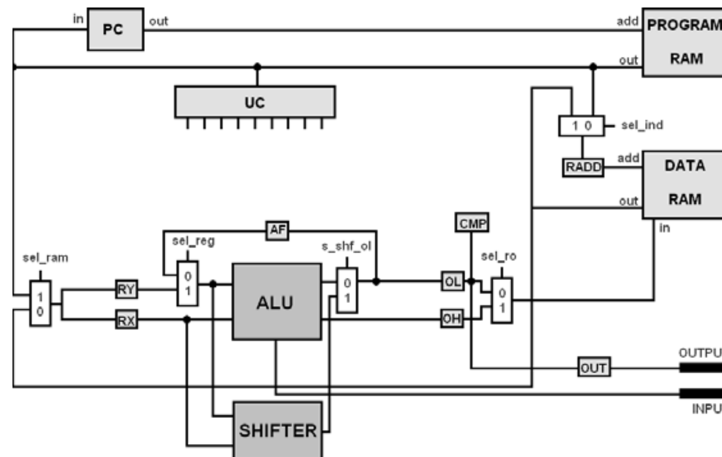
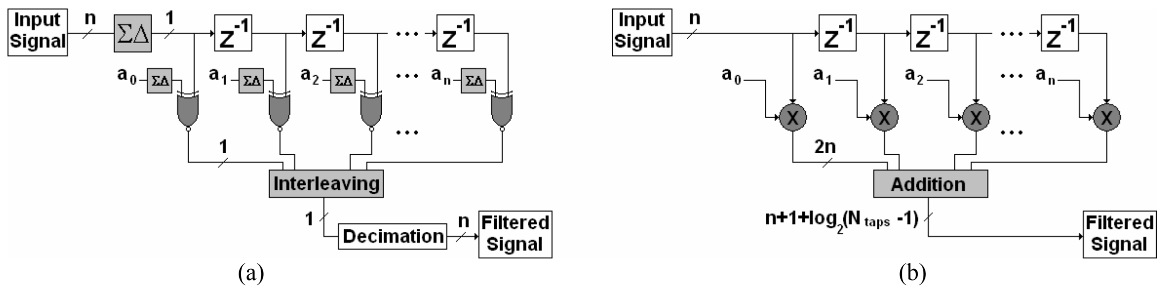


Figure 12: Sigma-Delta DSP processor block diagram.



(a) (b)

Figure 13: (a) Finite Impulse Response filter structure using sigma-delta modulated input signal and coefficients, and (b) classical digital implementation for the same filter.

(a) (b)

(c)

Figure 14: Practical results from the ΣΔ-DSP implementing a 16 taps FIR filter with different OSR used to modulate the coefficients.

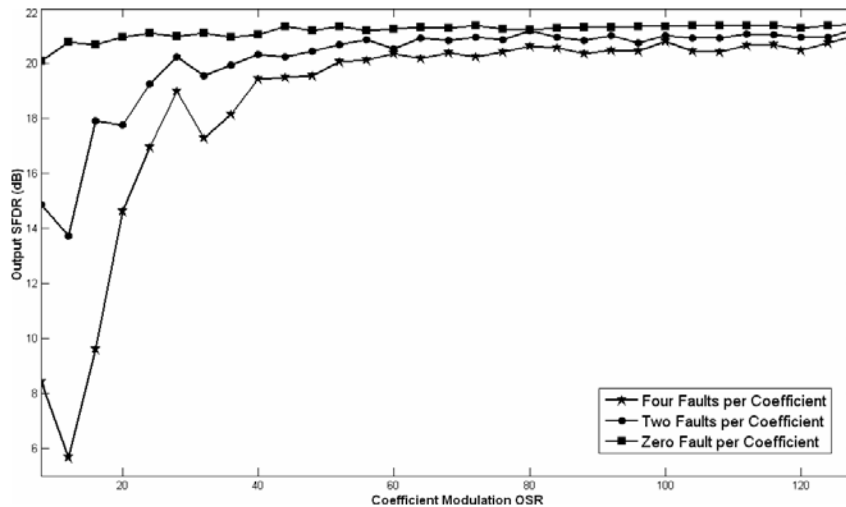


Figure 15: Resolution measured in SFDR *versus* OSR used to modulate FIR coefficients, for a constant number of faults injected in the coefficients bit streams.

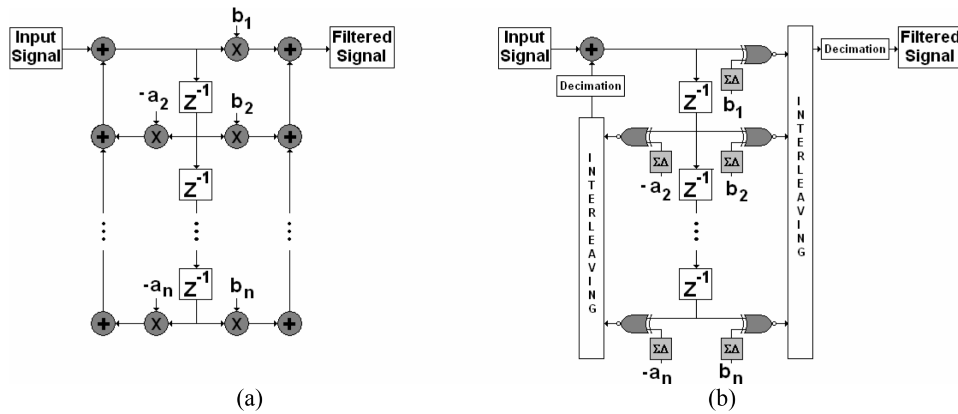


Figure 16: (a) Finite Impulse Response filter classical digital implementation, and (b) structure using sigma-delta modulated input signal and coefficients for the same filter.

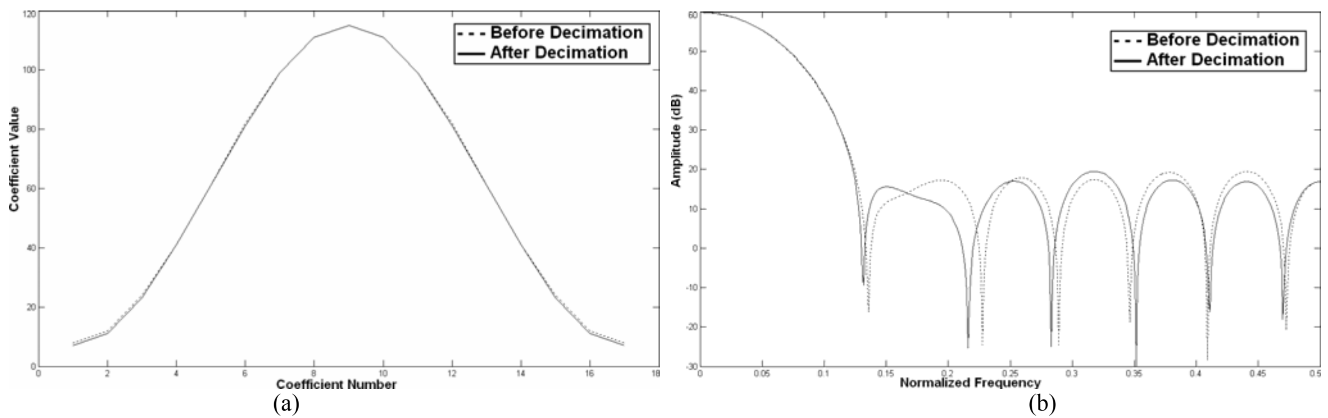


Figure 17: Finite Impulse Response filter coefficients (a) and filter response (b), showing the effects of coefficient quantization.

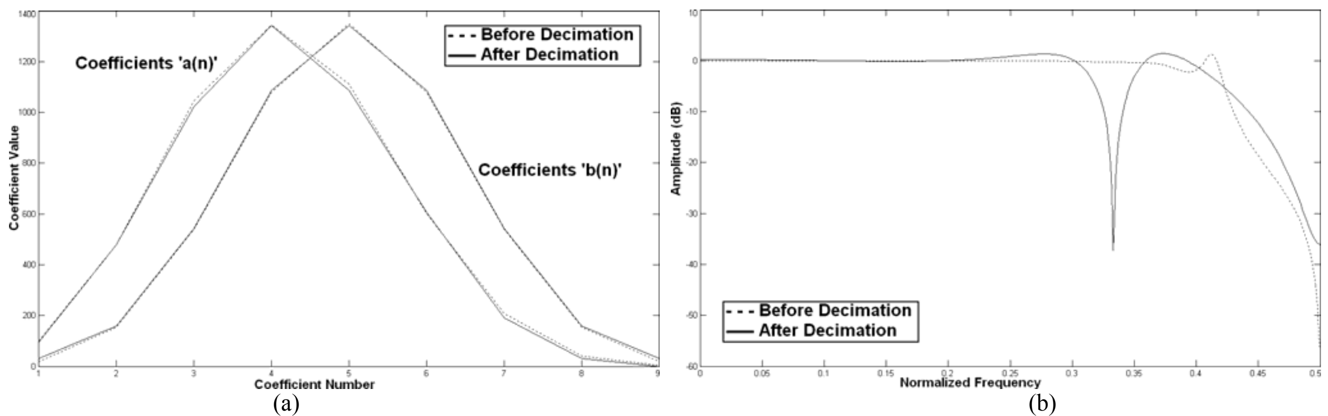


Figure 18: Infinite Impulse Response filter coefficients (a) and filter response (b), showing the effects of coefficient quantization.

Figure 19: n-bits PCM words DSP processor block diagram.

(a)

(b)

(c)

Figure 20: Standard DSP microprocessor simulation results. In (a), fault-free and faulty filter impulse response; (b) filter coefficients, and in (c), FIR output from the fault-free and faulty DSP response.

Table I: Sigma-delta and standard DSP microprocessor comparison summary

Evaluated Item	Sigma-Delta DSP	PCM DSP	ΣΔ Modulator (1st order)
Logic Elements ACEX1K EP1K100QC208-3	1882/4992 (36%)	2330/4992 (46%)	36/4992 (<1%)
Time to filter one point* ACEX1K EP1K100QC208-3	60µS	103µS	---
Logic Elements CYCLONE EP2C5F256C6	1422/4608 (30%) + 185 registers	1384/4608 (30%) + 233 registers + 2/23 embedded multiplier (9-bits elements)	30/4608 (<1%)
Time to filter one point* CYCLONE EP2C5F256C6	60µS	103µS	---

*** The same clock (30MHz) was used for both DSP. Time evaluations are made for an OSR equal to 16 for each coefficient modulation.**

7. References

[1] Yang, F.L., Saleh, R.A. "Simulation and Analysis of Transient Faults in Digital Circuits". IEEE Journal of Solid-State Circuits, Vol: 27, Issue: 3, March 1992. Pages: 258 – 264.

[2] G. C. Messenger, "A summary review of displacement damage from high-energy radiation in silicon semiconductors and semiconductor devices", IEEE Transactions on Nuclear Science, vol. 39, Issue: 3, June 1992, pp. 468 – 473.

[3] Smith, E.C., Shoga, M., "Double Upsets from Glancing Collisions: A Simple Model Verified with Flight Data". IEEE Transactions on Nuclear Science 39, 1859-1864 (1992).

[4] Chande, P.K., Ramani, A.K., Sharma, P.C., "Modular TMR Multiprocessor System". IEEE Transactions on Industrial Electronics. vol.36, n.1, Feb.1989.

[5] Rebaudengo, M., Reorda, M.S., "Evaluating cost and effectiveness of software redundancy techniques for hardware errors detection". Digest of Fast Abstracts: FTCS-28, The 28th Annual International Symposium on Fault-Tolerant Computing, June 23-25, 1998, Munich (Germany), pp. 88-89.

[6] Norsworthy, S.R.; Schreier, R.; Temes, G. C. "Delta-sigma data converters: theory, design and simulation". New York: IEEE Press, 1997. 476p.

[7] Schüler, E., Carro, L., "Increasing Fault Tolerance to Multiple Upsets Using Digital Sigma-Delta Modulators". 11th International On-Line Testing Symposium (IOLTS'05). France, 2005. June 27-29, 2005.

[8] Schüler E., Carro, L., "Reliable Digital Circuits Design using Sigma-Delta Modulated Signals", 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2005 (DFT'05), 03-05 Oct. 2005, USA.

[9] Basedau, P., Qiuting, H., "A post processing method for reducing substrate coupling in mixed-signal integrated circuits". Symposium on VLSI Circuits, Kyoto, 8-10 June, pp. 41-42, 1995.

[10] IBM. "SOI Technology: IBM's Next Advance in Chip Design". Last access: Oct. 2005. http://www-03.ibm.com/chips/about/technology/technologies/soi/

[11] Weaver, H., et al. "An SEU Tolerant Memory Cell Derived from Fundamental Studies of SEU Mechanisms in SRAM". IEEE Transactions on Nuclear Science. vol. 34, n.6, Dec. 1987.

[12] Rabaey, J.M., "Digital Integrated Circuits - A Design Perspective". Upper Saddle River : Prentice Hall, 1996. 702 p.

[13] Cota, E. F., et al, "Synthesis of an 8051-Like Microcontroller Tolerant to Transient Faults". Journal of Electronic Testing: Theory and Applications, Dordrecht, v. 17, n. 2, p. 149-162, 2001.

[14] Plank, J. S., "A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems". Technical Report UT-CS-96-332, University of Tennessee, July, 1996.

[15] Huang, K.H., Abraham, J.A., "Algorithm-Based Fault Tolerance for Matrix Operations", IEEE Transactions Computers, vol. 33, 1984, pp. 518-528.

[16] Gupta, S. K., Breuer, M. A. and Mak, T. M. "Defect and Error Tolerance in the Presence of Massive Numbers of Defects". IEEE Design & Test of Computers, Volume: 21, Issue: 3, May 2004. Pages: 216 – 227.

[17] Johnson, B.W., "Fault Tolerance," The Electrical Engineering Handbook, R.C. Dorf, ed., CRC Press, 1993, p. 2020.

[18] Nepal, K., Bahar, R. I., Mundy, J., Patterson, W. R., Zaslavsky, A., "Designing Logic Circuits for Probabilistic Computation in the Presence of Noise", Proceedings of 42nd IEEE Design and Automation Conference (DAC2005), June 13–17, 2005, Anaheim, California, USA.

[19] Anghel, L., Alexandrescu, D. and Nicolaidis, M. "Evaluation of soft error tolerance technique based on time and/or space redundancy". Proceedings of 13rd IEEE Symposium on Integrated Circuits and Systems Design (ICSD 2000),, Manaus, Brazil, September 2000. Pages: 237 – 242.

[20] Janer, C.L, Quero, J.M., Ortega, J.G., Franquelo, L.G., "Fully Parallel Stochastic Computation Architecture". IEEE Transactions on Signal Processing, vol. 44, n. 8, pp.2110 – 2117, IEEE Computer Society, New York, 1996.

[21] Lisbôa, C. and Carro, L. "Arithmetic Operators Robust to Multiple Simultaneous Upsets". Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems - DFT 2004, pp. 289-297, ISBN0-7695-2241-6. IEEE Computer Society, New York, October 2004.

[22] Candy, J.C., Temes, G.C., "Over sampling delta-sigma data converters : theory, design, and simulation". New York: IEEE Press, c1992. xii, 499 p. : il.

[23] Robert, G.W., Lu, A.K., "Analog Signal Generation for Built-In-Self-Test of Mixed-Signal Integrated Circuits", Kluwer Academic Publishers, 1995. 122p.

[24] Maloberti, F., "Non Conventional Signal Processing by the use of Sigma Delta Technique: a Tutorial Introduction". Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), Vol: 6, May 1992. Pages: 2645 – 2648.

[25] da Fonte Dias, V. "Sigma-delta signal processing". IEEE International Symposium on Circuits and Systems, 1994. ISCAS'94, Volume: 5, 30 May-2 June. Pages: 421 – 424.

[26] Boser, B.E., Wooley, B.A., "The Design of Sigma-Delta Modulation Analog-to-Digital Converters". IEEE Journal of Solid-State Circuits, vol. 23, no. 6, Dec. 1988.

[27] Schüler, E., Farenzena, D.S., Carro, L., "Evaluating Sigma-Delta modulated signals to develop fault-tolerant circuits". Proceedings of 11th IEEE European Test Symposium, 2006. ETS'O6, Southampton, UK, 21-25 May, 2006.

[28] Available at http://www.analog.com Last access: Jan. 2006.

[29] Tanskanen, J. M. A., "Coefficient Quantization Error Free Fixed-Point IIR Polynomial Predictor Design", Proc. 2000 IEEE Nordic Signal Processing Symposium, Kolmarden, Sweden, June 2000, pp. 219-222.

On the use of higher-order ΣΔ-modulators for reliable digital circuits design

Erik Schüler, Daniel Scain Farenzena, Luigi Carro

Electrical Engineering Department
Universidade Federal do Rio Grande do Sul
Porto Alegre, Brazil
eschuler@eletro.ufrgs.br, daniel.s.f@ufrgs.br, carro@eletro.ufrgs.br

Abstract-**Process technology has progressed to nanometric scale transistors gate length, thus faults regarding effects due to electromagnetic interference, free ions and particles are increasing. Efforts to reduce this susceptibility were made in many works, as low tolerance systems can have their functions severely affected. In our previous work, first-order delta-sigma digital modulation was successfully used to cope with single-event upsets in digital circuits. In the present work we analyze the use of higher-order delta-sigma modulation to design reliable digital circuits for future technologies.**

1. Introduction

Soft-errors mitigation has became a must in many critical systems, since the occurrence of these phenomena has increased with the transistor size shrinking. Bit-flips, caused by Single-Event-Upsets (SEUs) and transient pulses latching, related as Single Event Transients (SETs), have been related in the literature [1] [2], and future scenario is expected to be even worse. Since the total charge stored in the circuit nodes becomes smaller, the occurrence of multiple and simultaneous faults may become a common fact in smaller technologies.

We have proposed in past works [3][4] a technique that consists in modulating the signal to be processed in the delta-sigma domain in such a way that the redundancy presented in the modulated signal can reduce the effects of the faults occurred during the system operation, thus allowing one to build more reliable digital circuits, regarding Single Event Upsets (SEUs). In a practical application like a digital filter, even with several simultaneous faults the impact on the total SNR was close to zero.

Now, our goal is to compare first and second-order delta-sigma digital modulators regarding fault immunity. The reliability of the circuit assured in the first-order scheme was our motivation to the second-order delta-sigma modulator benchmarking, since better resolutions can be achieved through the use of higher-order modulators, with a lower over sampling ratio. However, as it will be shown, for second-order modulators the fault tolerance is not so high, since faults will cause worse consequences to the final value of the modulator due to its double integration.

This paper is divided as it follows: section II presents some previous results showing that the proposed technique is in fact a good solution for multiple faults occurrence. Section III explains the theory behind measuring fault tolerance on delta-sigma modulators and how it is possible to extract important parameters of the output signal. Section IV exhibits the performance of the first and second-order over fault insertion. Finally, section V concludes our analysis.

2. ΣΔ modulation for digital safety circuits

The use of ΣΔ modulation to achieve high fault-tolerance in digital circuits has already been demonstrated in our previous works. Some others results are showed here, in order to enforce this affirmation.

Figure 1 presents a first-order sigma-delta modulator, based on the topology presented in [5]. Fault injection is done during the modulation process, through the inversion of one randomly chose bit in the output word of the modulator blocks, in a randomly chose time. The consequence of this inversion, as will be demonstrated in section III, is the inversion of more than one bit in the modulator output bit stream.

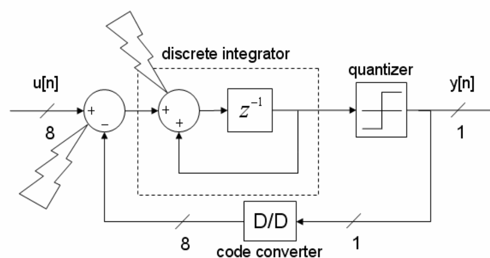


Figure 1. First-order delta-sigma digital modulator.

In figure 2, two tones are modulated to the ΣΔ domain for further addition through the use of the interleaving operation [6]. Each tone modulation is done using an Over Sampling Rate (OSR) equal to 64, that is, each tone is sampled with a frequency 64 times higher than its Nyquist frequency.

In figure 2(a), no fault is injected during the modulation process and one has a SNR of more than 24dB. Now, twenty randomly bits are inverted (from 0 to 1 or vice-versa) during the modulation process, being ten bits inverted in the modulator input adder and ten in the modulator integrator. As seen in figure 2(b), even after this amount of faults injected, the SNR remains in a good level (more than 23db).

With this brief analysis, it is possible to realize that the use of ΣΔ signal processing is in fact fault tolerant, and could be an interesting solution for some critical applications. We shall present in next section a mathematical analysis of what happens when bits are flipped during a sigma-delta modulation process, for a further comparison with higher-order modulators, presented in section IV.

3. Fault analysis in a delta-sigma modulator

Once the fault tolerance capabilities of first-order delta-sigma modulation applied to digital circuits is presented, questions about how different topologies of delta-sigma modulators could be vulnerable to faults rise. The following analysis show the influence of bit-flips in higher-order modulators to the output bit stream, and how these faults can degrade the signal resolution.

The simulations were done through a bit-flip fault model, as the one presented in figure 3, where a signal passes through an integrator, and one bit is inverted, resulting in a certain amount added to the output signal.

Discrete-time mathematics provides us a description of such fault as a *Dirac Delta Function*, or more commonly, impulse function (δ). Thus, the signal at a selected part of the circuit is modeled as (1):

$$y[i] = x[i] + k.\delta[i-\tau]\,,\ k \le 2^n \qquad (1)$$

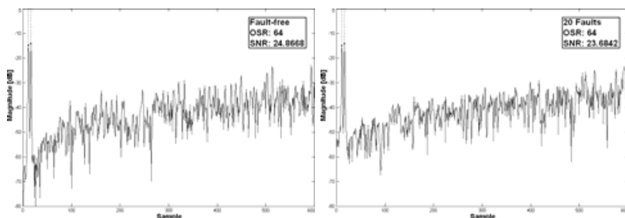


Figure 2. Fault-free (a) and faulty behavior (b) for the addition of two tones using ΣΔ-modulated signals.

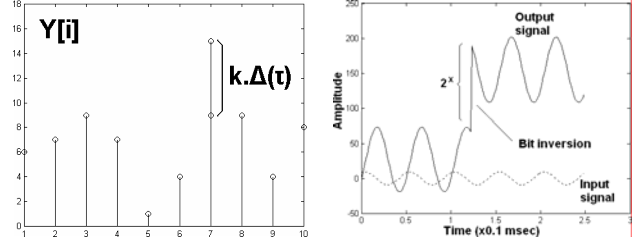


Figure 3. Model for bit flip in a given sample.

In (1), x is the original signal (fault-free) and τ is the time when the fault occurs. This model can be applied in every signal of the system, stating a generalized framework for multiple SEU.

An important fact to be taken into account is what happens when the fault propagates along the sigma-delta modulator circuit. Since a bit flip can be seen as a Dirac delta function (impulse function), which will instantly add a value to the sample due to the xth bit inversion (see figure 3), when this delta passes through an integrator, it becomes a step function, that is, the effect of a single bit flip lasts for the rest of the circuit operation.

Let us take the first-order modulator showed in figure 1. The error signal is integrated by the digital accumulator, whose amplitude is compared to a reference value to generate the 1-bit output. The feedback path is composed of a digital-to-digital converter, which will make a conversion of the 1-bit output to an n-bits input, in such a way that, if the 1-bit output is '1', the feedback value is -2^{n-1}, and if the output is '-1', the feedback generates +2^{n-1}.

For some conditions regarding the quantization error, the output bit stream average value will track the input signal [7]. Then it is expected that any fault occurring in the feedback path deteriorates the output signal or, in others words, the band signal resolution will be affected. If one analyses the circuit showed in figure 1 it is possible to see that in fact what is put back to the input is the integrator sign value and any inversion in this sign will directly affect the modulated signal.

Such behavior can be seen in figure 4, where a 20 kHz sinusoidal signal is modulated with an OSR of 32. Only one fault is injected in the circuit, through the inversion of the most significative bit in the output of the modulator input adder. As one can note, the output spectrum (figure 4(a)) is heavily affected when the number of inversions in the integrator sign is elevated (figure 4(b)). Also, figure 4(c) shows the predicted integration of the Dirac delta function, becoming a step function due to its integration.

Next section compares first and second-order modulators and shows that, due to the presence of two integrators in the second-order modulator, its tolerance to faults can be smaller than the first-order one.

4. Comparing performance of first and second-order delta-sigma modulators

Starting with the first-order modulator, we can quantify its fault robustness by means of the technique presented before. Since the signal is integrated only once, the fault causing an impulse in one sample becomes a step and, thus, a certain number of inversions in the feedback value occurs.

As seen in figure 5, the number of inversions is directly influenced by the bit inverted by the fault. For example, with the modulation a 20 kHz sinusoidal signal using the circuit in figure 1 (8 bits word plus sign bit) with an OSR of 16 and inverting the least significative bit (LSB) in different instants of the modulation process, the maximum number of inversion in the integrator sign is six (figure 5(a)). If the bit being inverted is the fourth, the number of inversion jumps to 25 (figure 5(b)). Finally, for the worst case scenario, if the most significative bit (MSB) is inverted the total amount of inversion is about 400 (figure 5(c)).

If one now considers a second-order modulator, showed in figure 6, a similar analysis can be done. However, for this case, since there are two integrators instead of one, the fault consequence can be even more disastrous.

Figure 7 shows the same simulations made for the first-order modulator, presented in figure 4. As seen for the second-order modulator, depending on which part of the circuit the fault occurs, the result can be more or less injurious. In figure 7(a), where one can see the fault occurring in the second subtractor of the modulator, it is possible to note that, although the signal band was affected by the fault, it was not affected as much as if the fault occurs in the first subtractor, as showed in the spectrum of figure 7(d).

This can be explained by the fact that, for faults occurring in the second subtractor, since they will be integrated only once (in the forward path), their consequences will not be as large as if they occur in the first subtractor, from where faults will be integrated twice (in the forward path). However, even for those faults hitting the second subtractor, the inversions due to these faults will be put back to the input of the circuit, thus being integrated twice, but not the faults themselves.

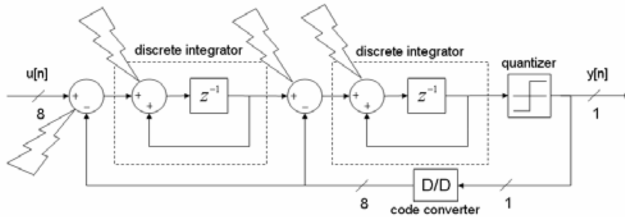


Figure 6. Second-order delta-sigma digital modulator.

Comparing figures 7(b) and 7(e) one can see that the number of inversions in figure 7(b) is much lower than that of figure 7(e), as expected. Also, through figures 7(c) and 7(f), it is shown that the faults (or the inversions caused by them) are integrated twice and therefore the final result is a ramp function.

Taking first and second-order delta-sigma modulators, it is expected that the first-order modulator is more fault tolerant than the second-order, since if faults occur in the first block of the second-order, the results are much worse than if they occur in the second block. For the first-order modulator, however, no matter where faults occur, the consequences are not so large. This conclusion can be remarked in simulations of figure 8. The graph presents the input signal of a first and second-order modulators varied all through their input range and 2 faults are inserted randomly in different parts of the circuit. The output bit streams generated with an OSR of 32 for figure 8(a) and 8(b) are decimated and plotted versus the input signal.

As noted, for the second-order modulator, the results are much worse then for the first-order one, where the variation from the expected value is very small. These results can be improved by augmenting the OSR to 64, as showed in figure 8(c) and 8(d), but the second-order modulator still presents a higher degradation.

These results can be extended to higher-order modulators. Since more integration steps are added to the forward-path, the consequence of a bit-flip can be integrated more than once, thus degrading the output signal resolution. Also, the intrinsic instability of higher-order modulators [8] can be even more affected by these faults.

5. Conclusions

Delta-sigma modulated signals have been demonstrated to be fault tolerant in our previous works. In this work, first and second-order digital modulators where compared in order to evaluate their robustness regarding transitory faults.

Simulation of faults on first and second-order delta-sigma digital modulators were performed by flipping bits in the words at the output of the building blocks of the circuit. The effects of these faults were tracked with the aid of a fault free circuit model and could be measured and quantified.

Although first-order modulator need higher oversampling ratios to get the same resolution as second-order one, the first-order presents a higher fault tolerance for the same number of faults injected in the circuit, since the integration of the error signal is done only once.

Similar results can be extended to higher order modulators, with even worse results since the number of integrators increases.

6. References

[1] G. C. Messenger, “A summary review of displacement damage from high-energy radiation in silicon semiconductors and semiconductor devices”, IEEE Transactions on Nuclear Science, vol. 39, Issue: 3, June 1992, pp. 468 – 473.

[2] Yang, F.L.; Saleh, R.A. “Simulation and Analysis of Transient Faults in Digital Circuits”. IEEE Journal of Solid-State Circuits, Vol: 27, Issue: 3, March 1992. Pages: 258 – 264.

[3] Schüler E., Carro, L., “Reliable Digital Circuits Design using Sigma-Delta Modulated Signals”, 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2005 (DFT’05), 03-05 Oct. 2005, USA.

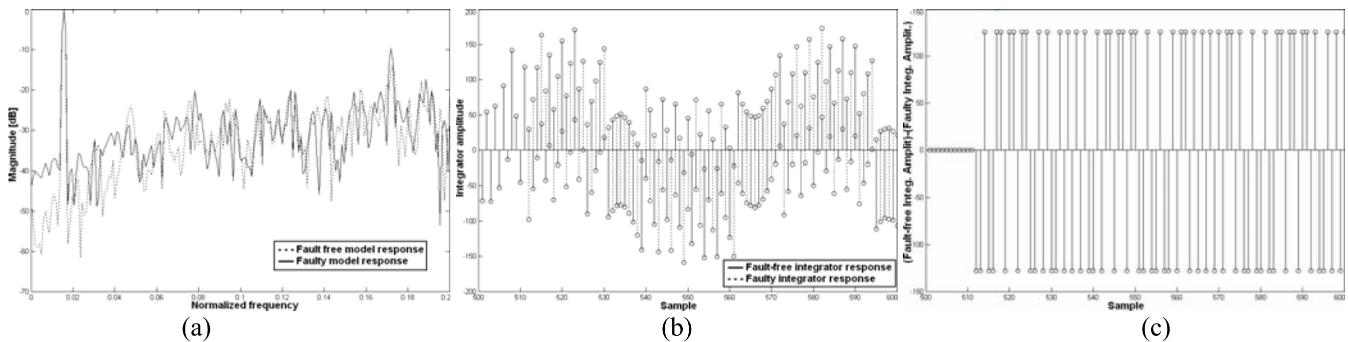
[4] Schüler, E., Carro, L., “Increasing Fault Tolerance To Multiple Upsets Using Digital Sigma-Delta Modulators”. 11th International On-Line Testing Symposium (IOLTS’05). France, 2005. June 27-29, 2005.

[5] G. W. Robert amd A. K. Lu, “Analog Signal Generation for Built-In-Self-Test of Mixed-Signal Integrated Circuits”, Kluwer Academic Publishers, 1995. 122p.

[6] Maloberti, F. “Non Conventional Signal Processing by the use of Sigma Delta Technique: a Tutorial Introduction”. Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), Vol: 6, May 1992. Pages: 2645 – 2648.

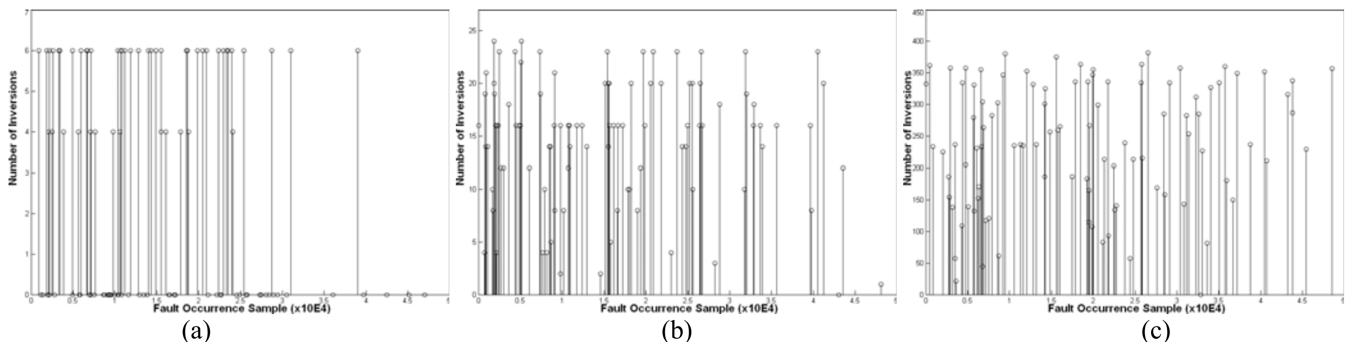
[7] S. R. Norsworthy, R. Schreier and G. C. Temes, “Delta-sigma data converters: theory, design and simulation”, New York: IEEE Press, 1997. 476p.

[8] Candy, J.C., Temes, G.C., “Oversampling delta-sigma data converters : theory, design, and simulation”. New York: IEEE Press, c1992. xii, 499 p. : il.



(a) (b) (c)

Figure 4. (a) Faulty and fault-free first-order output spectrum; (b) integrator output showing sign inversion and (c) step function after integrating the fault.



(a) (b) (c)

Figure 5. Number of inversions depending on the bit that is inverted: (a) LSB, (b) middle bit, (c) MSB.

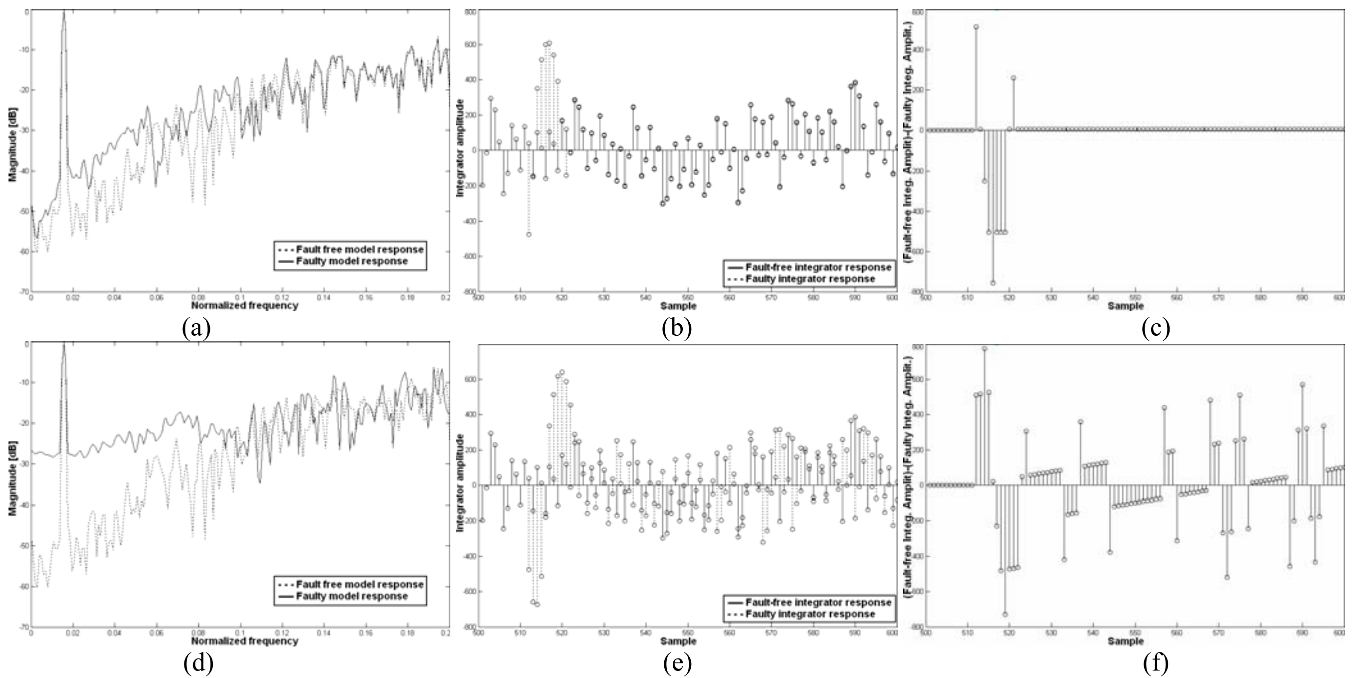


Figure 7. Second-order modulator response (a), integrator output (b) and ramp function after fault integration (c) for faults being injected in the second block and response (d), integrator output (e) and ramp function after fault integration (f) for faults being injected in the first block.

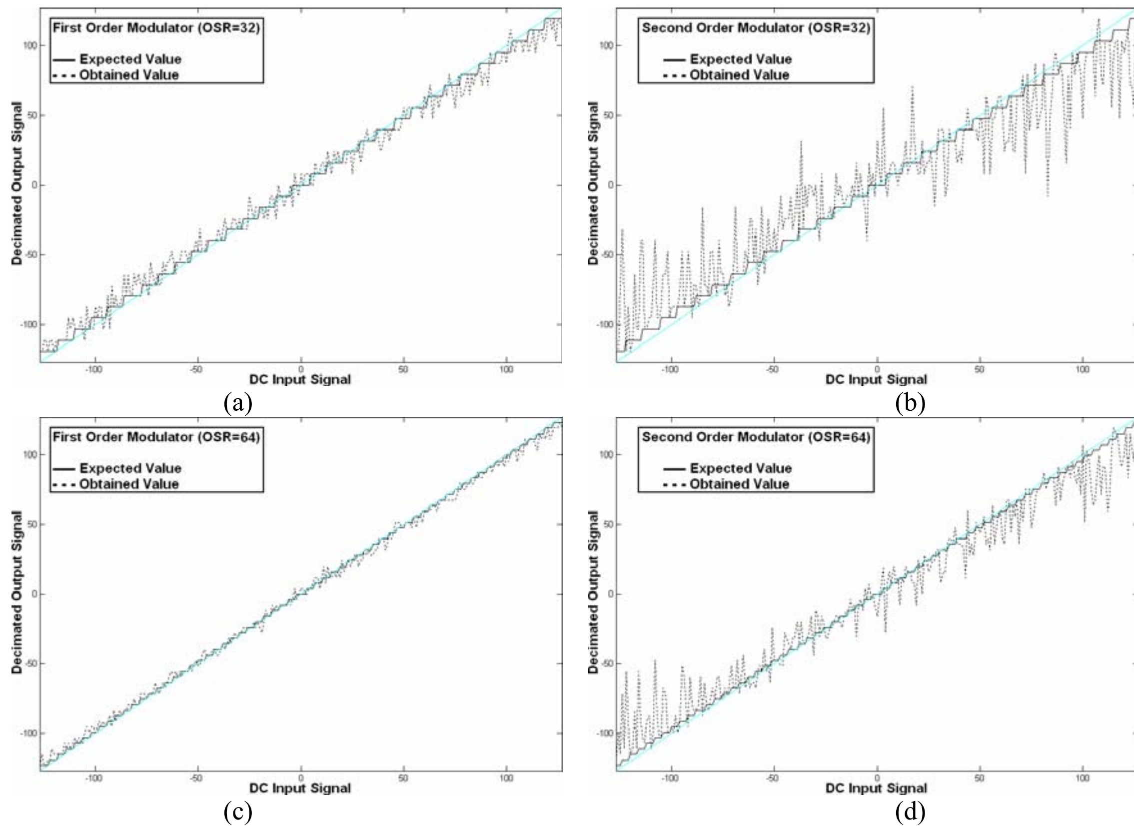


Figure 8. Delta-sigma input/output relation for different number of faults injected and different OSR: (a) first-order and OSR=32; (b) second-order and OSR=32; (c) first-order and OSR=64; (d) second-order and OSR=64.

Increasing reliability in future technologies systems

Erik Schüler, Luigi Carro
Universidade Federal do Rio Grande do Sul, Departamento de Engenharia Elétrica
eschuler@eletro.ufrgs.br, carro@eletro.ufrgs.br

Abstract

Process technology has progressed to nanometric scale transistors gate length, thus faults regarding effects due to electromagnetic interference, free ions and particles are increasing. Efforts to reduce this susceptibility were made in many works, as low tolerance systems can have their functions severely affected. This work presents the use of sigma-delta modulators that are used to develop a FIR filter where, even with the insertion of a large amount of transient faults, one can still obtain a non-faulty behavior in the final SNR

1. Introduction and literature review

The progressive reduction in the transistors dimensions will allow in the year 2008 the integration of more than one billion of these devices in a single die. The estimative is that the channel length, today in 65nm, is in the year 2016 in 22nm, becoming even smaller [1].

The size reduction associated to the circuit power supply reduction will bring as consequence the reduction in the total charge stored in the circuits nodes, what can take to the occurrence of the Single Event Effects (SEEs) [2].

Between the different types of SEE, the Single Event Upsets (SEUs) [3] are the responsible by the occurrence of the soft-errors, that is, the inversion in the memory or registers states. Although this phenomenon can occur even in analog, optical and digital circuits, its consequence in digital circuits, as mentioned, is the inversion in one or more bits, having as consequence, depending on which bit is inverted, a total mismatch between the achieved and the desired response.

Many solutions have been proposed to cope with these problems. These solutions can be divided in three groups: hardening by technology, hardening by design and hardening by system. In the solutions in the technology level, different processes are used in the transistor fabrication, for example, epitaxial-bulk (epi-bulk) CMOS and Silicon On Insulator (SOI). When hardening by design, specific solutions are used for each design. For example, the use of Triple Modular Redundancy (TMR), hardened gate resistor memory cells and hardened CMOS memory cells with feedback structures, or even the use of codification and decodification of logic block through the use, for example, of the Hamming Code or Reed-Solomon Code. Finally, the development of more robust systems can be done through the use of redundancy techniques in software (variable duplication) or hardware (TMR), and the insertion of blocks for error detection and correction. Many others techniques have been proposed, such as those described in [4], [5] and [6].

Some of these solutions, however, are not completely tolerant to SEUs occurrence as, for example, the use of SOI. Some others have the limitation of do not support multiple faults occurrence, what is an expected scenario in future technologies.

Recently, it was proposed in [7] the use of redundant signal, what would support the occurrence of multiple faults. The redundant signal to be processed is generated through the comparison of the binary signal with a uniformly distributed noise, many times. The resultant bit stream, whose mean value is equal to the input signal, has now a certain amount of redundancy, in such a way that, even if many bits are inverted, the achieved response of the system still has a good resolution.

However, in order to obtain reasonable resolutions to use this technique in different applications, the bit stream size is in the order of thousands of bits, what becomes the solution very limited in terms of speed.

So, the research for low-power, low-area high performance solutions still exists, tending to increase as the integration scaling increases.

This work presents a technique to cope with the effects of bit flips in digital circuits based on the solution proposed in [7], however with the bit stream being generated through the use of a sigma-delta ($\Sigma\Delta$) modulators [8]. By using this kind of modulation, the representation of a given value can be done through the use of a lower quantity of bits (in the order of dozens), while still achieving a high resolution.

This paper is divided as it follows: in section II, it is shown that signals modulated in sigma-delta domain are in fact more tolerant to faults, due to the amount of redundancy present in its representation. In order to show that a small deviation from the original value does not interfere so much in the final result of some applications, section III shows some results regarding the implementation of a FIR filter using the proposed technique. Finally, section IV presents the conclusions and some future work.

2. Fault tolerance in the ΣΔ domain

Sigma-delta-modulated signals, although requiring over sampling, have a great advantage if compared with code-modulated signals, regarding fault tolerance: since sigma-delta-modulated signals are over sampled, each bit stream that represents an amplitude value carries an amount of redundancy that, even with faults inserted, can still represent the mean value of the original signal.

Figure 1 shows a sequence of simulations, where the decimated output values of the sigma-delta converter are plotted when varying the input signal over the input range. Faults are injected during the simulation process in the sigma-delta modulator, both in the input adder and in the integrator outputs. Faults are simulated through the inversion of randomly chose bits in the block output word. In other words, for an 8-bits adder, one of the 8 output bits is inverted in a given moment (either from logic zero to one or vice-versa).

As one can see in figure 1(a), where no fault is injected, the modulator output values do not correspond exactly to the input signal for all input values, due to the quantization error. However, it is possible to achieve a very good approximation of such value, and this can be very useful when used in applications that tolerate small error percentages [9]. Also, this approximation can be improved through the increase of the Over Sampling Rate (OSR), but not without some time penalties.

Looking at figure 1(b), when one fault is injected in the modulator (both in the input subtractor and in the integrator, as shown in figure 2), the output values maintain a certain mean value that still approximate to the input one. Even when the number of faults is increased to four (figure 1(c)), one still has a good I/O relation, which, as mentioned, can be improved by increasing the modulator OSR, as shown in figure 1(d). Next section shows the results of the implementation of a digital FIR filter, where the signals to be processed are in the sigma-delta domain. It will be shown that, even inserting many faults, the achieved response still has a good SNR, when comparing to the faulty-free behavior.

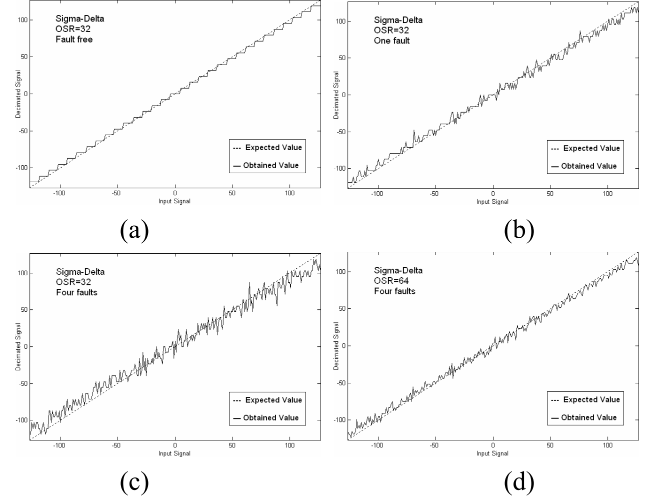


Figure 1: Sigma delta input/output relation for different number of faults injected.

3. ΣΔ signals in fault tolerant FIR

This section presents some results regarding the use of sigma-delta-modulated signals in more complexes applications when faults are injected. The proposition is to analyze the convolution case, which is the basis of filters operation. Figure 3 shows the structure used to filter noisy signals using the proposed technique. The faults are simulated through the inversion of one random bit in the modulator adders, as done in section II.

Some techniques to filter signals in the sigma-delta domain have already been proposed [11, 12]. In this case, once we are modulating each of the quantized signal to the sigma-delta domain, the filter structure should have some modifications. However, one can still benefits from the fact that the generated bit streams are a sequence of 0 and 1, then, one can just chooses whether he wants or not add the coefficients values, instead of making a multiplication of the input signals by these coefficients. An 8 taps low pass FIR was used in the simulations.

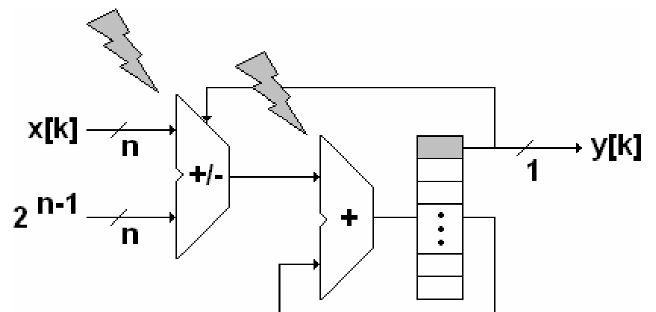


Figure 2: Sigma delta modulator based on the model proposed in [10].

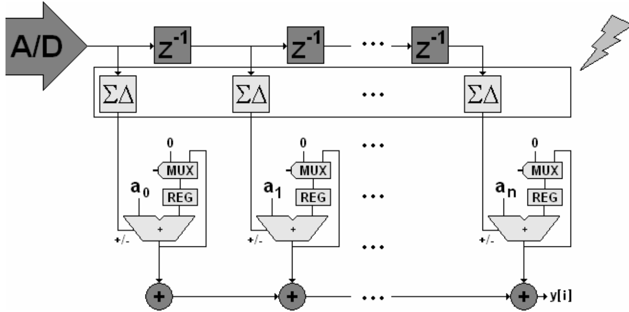


Figure 3: Representation of the structures used to filter signals using sigma-delta-modulated signals.

Simulations results obtained are presented in figure 4. The input signal formed by a single 20KHz tone and an amount of uniformly distributed random noise was sampled and filtered in two different ways: using the samples themselves and a common FIR filter without any fault inserted and using a sigma-delta modulation of the samples with the insertion of faults. All simulations results were done by taking 512 points of the input signal and modulating each of these signals. Faults were injected during the modulation of 32 of these 512 points. So, if one has 'k' faults injected during the modulation process, the total amount of faults is given by '32k'. The fault injection mechanism is the same described in section II.

Figure 4(a) shows the input signal in the time and frequency domain. In figure 4(b), it is shown a comparison between the output signals being filtered using a common FIR and the proposed technique with no faults. Then, figure 4(c) presents these same results when four faults are injected in the modulation process. As mentioned, these four faults are injected in the modulation of 32 points, resulting then in 128 bits inverted during this process. For figures 4(b) and 4(c), the OSR used to modulate each of the sampled points is 16. In figure 4(d), however, this OSR is increased to 32, showing that an even better resolution is achieved, even with the insertion of 4 faults. The code-modulated FIR results presented in these figures are just to make comparisons so, no faults are injected in these simulations.

4. Conclusions

This work presented the use of sigma-delta modulators in the generation of fault-tolerant signals, intended to be used in mean-based systems that make intensive use of arithmetic operations. It was shown that systems that use such technique can handle multiple faults, while still producing results that are very close to those produced when using fault-free code-modulated signals.

In order to show that such small deviation from the original value does not cause a great disturb in the final result, a FIR filter was developed. Signals modulated in the sigma-delta domain were filtered with the insertion of a great amount of faults, and compared with fault-free code-modulation FIR, showing that the tolerance to faults in the sigma-delta solution is good enough to obtain results with a very small difference between the SNR of these two solutions.

Although the proposed approach leads to the necessity of higher frequency systems, since the sigma-delta modulator needs high over sampling rates to achieve a good resolution, the high tolerance to multiple faults makes its use advantageous. However, as mentioned, for systems that must multiply the input bit-stream by a certain value, since the bit-stream if formed by a sequence of zeros and ones, the multiplication can be substituted by a simple addition, what can contribute for the reduction of the area taken for processing. For the FIR case, if the coefficients have also a sigma-delta representation, the area overhead can be even smaller.

Future works include the investigation on the effects of the dynamic reduction of the over sampled ratio, in order to adjust the amount of tolerance to errors with the rate of errors present in the system, since particles do not reach the system at constant intervals. A neural network approach will also be developed in order to realize radiation tests to compare the proposed technique results against results using standard n-bits words.

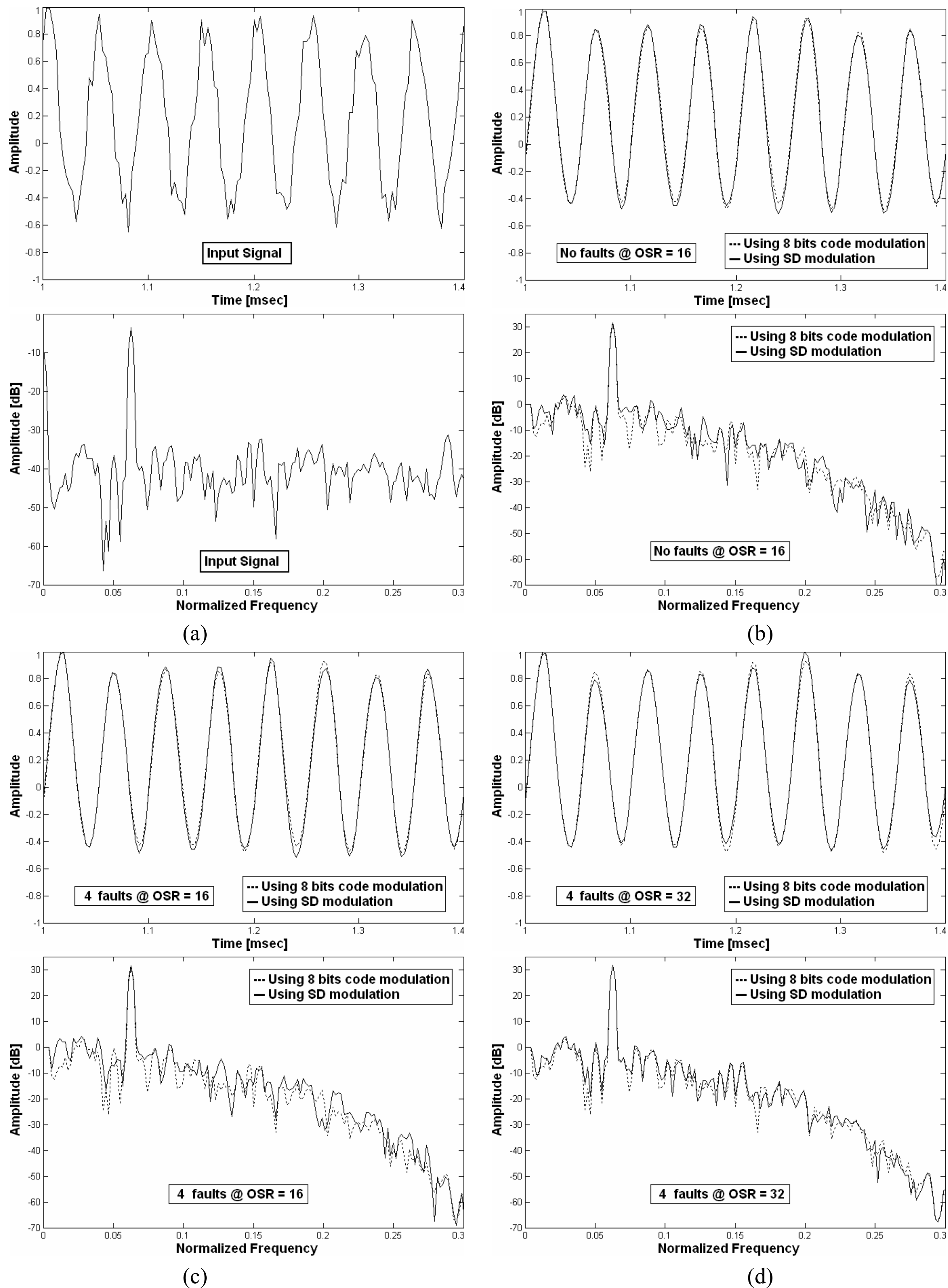


Figure 4: Results obtained by filtering a noisy 20KHz signal using sigma-delta modulated signals. In (a), the input signal, in (b), the filtered signal with no faults @OSR 16, in (c) with 4 faults @OSR 16 and in (d) with 4 faults @OSR 32.

5. References

[1] Semiconductor Industry Association. International Technology Roadmap for Semiconductors - ITRS 2004 last access Jan. 26, 2005. http://www.itrs.net/Common/2004Update/2004Update.ht

[2] Messenger, G.C. “A summary review of displacement damage from high-energy radiation in silicon semiconductors and semiconductor devices”. IEEE Transactions on Nuclear Science, Volume: 39, Issue: 3, June 1992. Pages: 468 – 473.

[3] Yang, F.L.; Saleh, R.A. “Simulation and Analysis of Transient Faults in Digital Circuits”. IEEE Journal of Solid-State Circuits, Volume: 27, Issue: 3, March 1992. Pages: 258 – 264.

[4] Anghel, L., Alexandrescu, D. and Nicolaidis, M. “Evaluation of soft error tolerance technique based on time and/or space redundancy”. Proceedings of 13rd IEEE Symposium on Integrated Circuits and Systems Design (ICSD 2000),, Manaus, Brazil, September 2000. Pages: 237 – 242.

[5] Lima, F., Carro, L. and Reis, R. “Techniques for Reconfigurable Logic Applications: Designing Fault Tolerant Systems into SRAM-based FPGAs”. Proceedings of the International Design Automation Conference, DAC 2003, ACM, New York, 2003. Pages: 650 – 655.

[6] Krithivasan, R., et.al., “An SEU Hardening Approach for High-Speed SiGe HBT Digital Logic”. IEEE Transactions on Nuclear Science, vol. 50, n. 6, dec. 2003.

[7] Lisbôa, C. and Carro, L. “Arithmetic Operators Robust to Multiple Simultaneous Upsets”. Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems - DFT 2004, pp. 289-297, ISBN0-7695-2241-6. IEEE Computer Society, New York, October 2004.

[8] Norsworthy, S.R., Schreier, R., Temes, G. C. “Delta-sigma data converters: theory, design and simulation”. New York: IEEE Press, 1997. 476p.

[9] Gupta, S. K., Breuer, M. A. and Mak, T. M. “Defect and Error Tolerance in the Presence of Massive Numbers of Defects”. IEEE Design & Test of Computers, Volume: 21, Issue: 3, May 2004. Pages: 216 – 227.

[10] Robert, G. W., Lu, A. K., “Analog Signal Generation for Built-In-Self-Test of Mixed-Signal Integrated Circuits”. Kluwer Academic Publishers, 1995. 122p.

[11] Wong, P.W.; Gray, R.M., “FIR filters with sigma-delta modulation encoding”. IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume: 38, Issue: 6, June 1990. Pages: 979 – 990.

[12] Thompson, A.C.; O'Shea, P.; Hussain, Z.M.; Steele, B.R., “Efficient single-bit ternary digital filtering using sigma-delta modulator”. IEEE Signal Processing Letters, Volume: 11, Issue: 2, Feb. 2004. Pages: 164 – 166.

Fault tolerant DSP microprocessor for ΣΔ-modulated signals

Erik Schüler, Marcelo Ienczczak Erigson, Daniel Scain Farenzena, Luigi Carro
Electrical Engineering Department
Universidade Federal do Rio Grande do Sul
Porto Alegre, Brazil
eschuler@eletro.ufrgs.br, mierigson@inf.ufrgs.br, daniel.s.f@ufrgs.br, carro@eletro.ufrgs.br

Abstract

As process technology has progressed to nanometric scale transistors gate length, faults regarding effects due to electromagnetic interference, free ions and particles have increased. Efforts to reduce this susceptibility were made in many works, as critical systems can have their functions severely affected. This work presents the use of digital sigma-delta modulation to cope with fault occurrence. The technique is used to prototype a DSP microprocessor, which processes ΣΔ-modulated signals only, with increased fault tolerance. Practical results show that the proposed approach can cope with multiple faults, while keeping a satisfactory signal-to-noise ratio when compared to a faulty-free behavior.

1. Introduction and Literature Review

Soft-errors mitigation has became a must in many critical systems, since the occurrence of these phenomena has increased with the transistor size shrinking. Bit-flips, caused by Single-Event-Upsets (SEUs) and transient pulses latching, related as Single Event Transients (SETs), have been related in the literature [1] [2], and future scenario is expected to be even worse, since the total charge stored in the circuit nodes becomes smaller.

Many solutions have been proposed to cope with these problems, and these solutions can be divided in three groups: hardening by technology, hardening by design and hardening by system. For the technology level, different processes are used in the transistor fabrication, such as epitaxial-bulk CMOS and Silicon On Insulator (SOI). When hardening by design, specific solutions are used for each design. For example, the use of hardened gate resistor memory cells and hardened CMOS memory cells with feedback structures, or even the use of codification and decodification of logic block through the use, for example, of the Hamming Code or Reed-Solomon Code. Finally, the development of more robust systems can be done through the use of redundancy techniques in software (variable duplication) or hardware (TMR), and the insertion of blocks for error detection and correction.

Some of these solutions, however, are not completely tolerant to SEU/SET occurrence as, for example, the use of SOI, which has also as a limiting factor, the need for special fabrication process. Some others have the limitation of not supporting multiple and simultaneous faults, what is an expected scenario in future technologies.

Based on the idea of hardware and software redundancy, we have proposed in [3] [4] a new paradigm: signal redundancy. With this approach, current CMOS technology can be used in the circuit development, while multiple and simultaneous faults are supported. What is proposed is the use of redundant microprocessor words, in such a way that, even if multiple bits were inverted, the final results still maintains a certain resolution, which can be even improved through a simple parameter variation. To generate the redundant signal, sigma-delta modulation [5] is used, allowing one to obtain high resolution signals, with high fault tolerance, but with minimum cost as speed penalties [4].

In this paper, we present a DSP microprocessor, which processes sigma-delta-modulated signals only, that is, instead of using an n-bits code-modulated word, the values processed by the DSP are in the sigma-delta domain.

The paper is divided as it follows: section II presents a brief mathematical analysis and some simulation results, which show that sigma-delta-modulated signals are in fact fault-tolerant. In section III, the DSP microprocessor is described, and the results of the implementation of a FIR filter using the DSP are presented. Finally, section IV presents some conclusions and future works.

2. Robustness in Sigma-Delta Domain

Let us consider a digital circuit working with sampled and quantized signals, represented by n-bits words. If when the quantized n-bits signal is passing through the circuit one of its bit suffers an inversion, caused by a particle strike, for example, this inversion can be seen as the addition (or subtraction) of a certain value in the current sample.

Discrete-time mathematics provides a description of such fault as a *Dirac Delta Function*, or more commonly, impulse function (δ). Thus, the signal at a selected part of the circuit can be modeled as (1), where x is the original value (fault-free), k is the magnitude of the amount added due to the bit inversion and τ is the time when the fault occurs:

$$y[i] = x[i] + k.\delta[i-\tau],\ k \le 2^n \qquad (1)$$

Applying this model to the sigma-delta modulator showed in figure 1(a), and representing this modulator by its linear model [5], presented in figure 1(b), one has the complete analytical faulty model of the sigma-delta modulator, presented in figure 1(c). For the model in figure 1(c), *k1* represents faults being inserted in the modulator input adder, *k2* in the modulator integrator and *k3* in the modulator quantizer (output comparator).

The final transfer function of the model is given in (2), where *BF(z)* represents the parcel corresponding to the fault, and is expressed by (3). Making the inverse z-transform, one obtains the difference equation (4), where bf_i is expressed by (5)-(7):

$$Y(z) = k_d Q(z)(1 - z^{-1}) + X(z)z^{-1} + BF(z) \quad (2)$$

$$BF(z) = k_1 z^{-\tau_1} z^{-1} + (k_2 z^{-\tau_2} + k_3 z^{-\tau_3})(1 - z^{-1}) \quad (3)$$

$$y[n] = k_d(q[n] - q[n-1]) + x[n-1] + \sum bf_i[n] \quad (4)$$

$$bf_1[n] = k_1 \delta[n - \tau_1] \quad (5)$$

$$bf_2[n] = k_2(\delta[n - \tau_2] - \delta[n - \tau_2 - 1]) \quad (6)$$

$$bf_3[n] = k_3(\delta[n - \tau_3] - \delta[n - \tau_3 - 1]) \quad (7)$$

The general expected behavior for a ΣΔ modulator is represented by the two first terms in equation (4), i.e., the quantization noise translated to higher frequencies while the input signal stays in its original band. As mentioned, the parcel *BF(z)* in (2) describes the faults, represented by impulses translated in time. In (3), it is possible to realize that for faults occurring before the integrator, these will appear inside the signal band. However, for faults occurring after or in the integrator, they will be moved outside the signal band. This is an expected behavior for linear circuits with feedback paths and an integrator in the forwarding path [6].

Results reached so far are based on a linear model of the modulator. However, the assumption that the quantizer is modeled by an added noise is not valid whenever the real circuit behavior must be evaluated.

Since the circuit quantizer is a simple comparator, which uses the integrator sign bit as its output, the feedback process no longer acts as the predicted model. Taking the real circuit, intrinsically non-linear, for simulation, the output of the discrete integrator is plotted just before the quantization step for a faulty and fault-free modulator, as showed in figure 2.

It can be seen in figure 2(a) that the impulse function (added by a fault inverting the LSB in the output of the adder) is integrated, and keeps being accumulated. For the linear model, one should expect this effect to fade, as predicted before. In the real circuit, however, since the feedback is a single bit, the fault is not obstructed, and will interfere in the bit-stream only when the fault is large enough to invert the sign of the integrator output.

The consequence, showed in figure 2(b), is an interference occurring inside the signal band, in opposition of what was predicted by the model. Moreover, when the number of inversions increases, for example due to the inversion of the most significative bit, also the in-band interference increases, as plotted in figure 2(c) and 2(d).

One can conclude, thus, that for the ΣΔ circuit, no matter where faults occur, if their consequences are fed back, that is, if the fault causes an inversion in the integrator sign bit, the output signal can be more or less affected, depending on the number of inversions.

However, even with the occurrence of the faults, the final signal-to-noise ratio (SNR) maintains a certain resolution that does not differ so much from a non-faulty behavior.

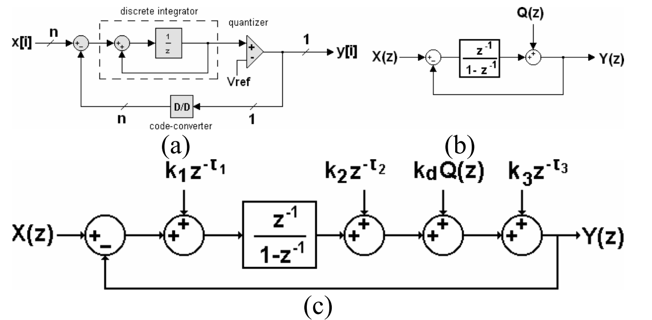


Figure 1. First order digital ΣΔ modulator (a), its general linear model (b) and complete analytical faulty model (c).

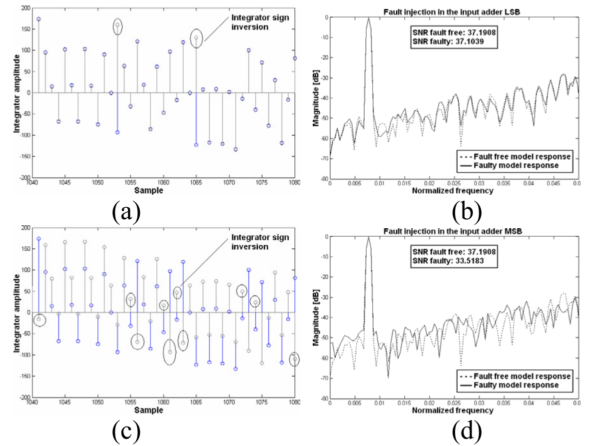


Figure 2. Integrator output signal. Difference in amplitudes reveals integrated delta function.

This can be seen in the simulation presented in figure 3, where two tones are modulated to the ΣΔ domain for further addition through the use of the interleaving operation [7]. Each tone modulation is done using an Over Sampling Rate (OSR) equal to 64, that is, each tone is sampled with a frequency 64 times higher than its Nyquist frequency.

For figure 3(a), no fault is injected during the modulation process and one has a SNR of more than 24dB. Now, twenty randomly bits are inverted (from 0 to 1 or vice-versa) during the modulation process, being ten bits inverted in the modulator input adder and ten in the modulator integrator. As seen in figure 3(b), even after this amount of faults injected, the SNR remains in a good level (more than 23db).

With this brief analysis, it is possible to realize that the use of ΣΔ signal processing is in fact fault tolerant, and could be an interesting solution for some critical applications. We shall present in next section, the implementation of a DSP microprocessor that uses this kind of signal modulation which, as will be demonstrated, can achieve very good results even in the presence of multiple faults. Some others applications using this approach can be found in [3] and [4].

3. DSP Microprocessor for ΣΔ Signals

The development of different DSP functions, like FIR or IIR filters, FFT computation and other functions can be easily made through the use of a DSP microprocessor, which is able to make multiply and accumulate operations in a single cycle, for example.

To make these functions with a high level of fault tolerance, with minimum area or speed overhead, we have developed a DSP microprocessor to build such functions when the signals to be processed are in the sigma-delta domain.

To do so, a VHDL description of a DSP based on the architecture of the Analog Devices ADSP2100 instruction set [8] has been developed.

(a) (b)

Figure 3. Fault-free (a) and faulty behavior (b) for the addition of two tones using ΣΔ-modulated signals.

The structure of a programmable processor to use ΣΔ coded signals is similar to the classical code modulated processor. However, the ΣΔ signals may have different lengths, which can vary from a few bits (16, for example), to hundreds of bits (128 or 256), depending on the desired resolution and, moreover, on the expected fault tolerance. The bits quantity is a function of the OSR used during the modulation process. For example, to obtain a value modulated in sigma-delta with an output bit-stream of 32 bits, the signal must pass 32 times through the sigma-delta modulator.

The DSP VHDL description follows the structure presented in figure 4, where one can note two main blocks, the Arithmetic Logic Unit (ALU) and Shifter, and two memories blocks, being one for the data and another for the program.

An Altera ACEX1K family EP1K100QC208-3 FPGA was used to prototype the DSP. A total of 1836 logic elements were used, representing 36% of the total available in the FPGA. Also, 8192 memory bits (16%) are used to the data and program memory.

To evaluate the DSP functionality, a 16 taps FIR filter was programmed. For the filter, both the input signal and the coefficients are modulated in sigma-delta. These signals were generated in Matlab® and then saved in the DSP data memory. Faults were injected during each coefficient modulation process through the inversion of a different number of randomly bits.

The filter structure is a simple sort of XOR gates, which will pass the value of the filter coefficient if the input signal is '1', or the negate coefficient value otherwise. The XOR outputs are added through an interleaving operation, and one filtered point is obtained.

In order to acquire the filtered results in a manner able to be evaluated, that is, not in the sigma-delta domain, but in the digital domain, a decimation block was added just after the interleaving operation. The filtered points were acquired using an *Agilent Infiniium* oscilloscope at sample rate of 1MSa/s. The values are then analyzed in Matlab®.

Figure 4. Sigma-Delta DSP processor block diagram.

The input signal is a 200Hz single tone with a white noise added, modulated in ΣΔ with an OSR of 64. A total of 1024 bits in sigma-delta representation were generated. In figure 5 it is shown different responses obtained after filtering the input signal using the described filter with different OSR used to modulated each coefficient. A comparison between a fault-free and a faulty behavior is presented, both using ΣΔ modulation. Faults are injected during the modulation of each of the coefficients, and the filter impulse response is presented.

In figure 5(a), an OSR of 16 is used in each coefficient. In this case, two faults are injected during the modulation of each coefficient, both in the modulator input adder and integrator, resulting, thus, in a total of four faults per coefficient. As seen, the difference between the fault-free and faulty response is practically null. Moreover, this difference can be reduced through the increase of the OSR used to modulate the coefficients, as presented in figure 5(b), where an OSR of 32 is used, also with four faults per coefficient. Finally, increasing the OSR to 64 and the number o faults to eight per coefficient, the final response still matches the faulty-free response, as shows figure 5(c).

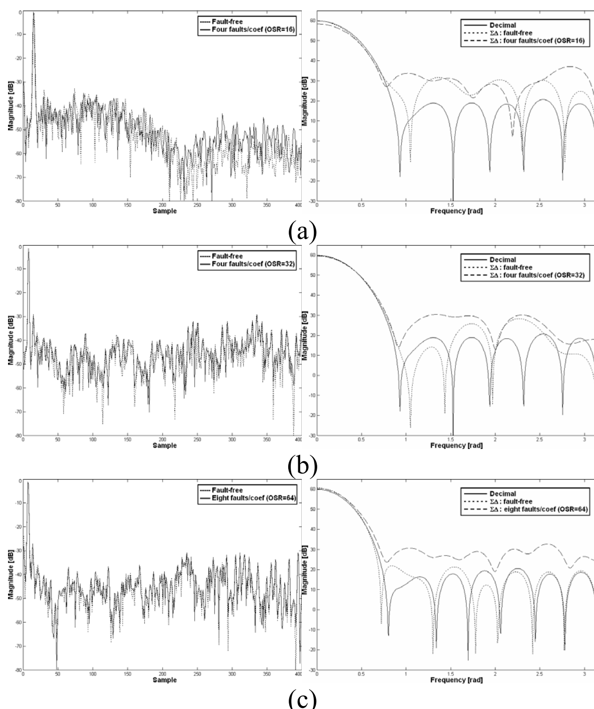


Figure 5. Practical results for a 16 taps FIR filter with different OSR used to modulate the coefficients, and filter impulse response.

4. Conclusions and Future Works

This work presented a DSP microprocessor used to develop fault tolerant applications, through the use of sigma-delta signal processing. A brief mathematical approach was presented, supporting the simulations presented, which show that the use of sigma-delta signals is in fact a good solution regarding multiple fault occurrence in digital circuits.

Practical results were obtained from programming a FIR filter using the DSP VHDL description, and it was shown that, even with the insertion of multiple faults, the final responses still matches a non-faulty behavior.

Future works include providing new functions within the DSP, like IIR filter, FFT and others. Also, a comparison between the ΣΔ DSP and a custom DSP using code-modulated words will made in order to compare area, performance and power overheads.

5. References

[1] Messenger, G.C. “A summary review of displacement damage from high-energy radiation in silicon semiconductors and semiconductor devices”. IEEE Transactions on Nuclear Science, Vol: 39, Issue: 3, June 1992. Pages: 468 – 473.

[2] Yang, F.L.; Saleh, R.A. “Simulation and Analysis of Transient Faults in Digital Circuits”. IEEE Journal of Solid-State Circuits, Vol: 27, Issue: 3, March 1992. Pages: 258 – 264.

[3] Schüler, E., Carro, L., “Increasing Fault Tolerance To Multiple Upsets Using Digital Sigma-Delta Modulators”. 11th International On-Line Testing Symposium (IOLTS’05). France, 2005. June 27-29, 2005.

[4] Schüler, E., Carro, L., “Reliable Digital Circuits using Sigma-Delta Modulators”. 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT’05). Monterey, USA, 2005. Oct. 03-05, 2005.

[5] Norsworthy, S.R.; Schreier, R.; Temes, G. C. “Delta-sigma data converters: theory, design and simulation”. New York: IEEE Press, 1997. 476p.

[6] Ogata, Katsuhiko. “Modern Control Engineering”. Englewood Cliffs, NJ: Prentice Hall, 1990.

[7] Maloberti, F. “Non Conventional Signal Processing by the use of Sigma Delta Technique: a Tutorial Introduction”. Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), Vol: 6, May 1992. Pages: 2645 – 2648.

[8] Available at http://www.analog.com Last access Jan. 2006.

Evaluating Sigma-Delta modulated signals to develop fault-tolerant circuits

Erik Schüler, Daniel Scain Farenzena and Luigi Carro
Universidade Federal do Rio Grande do Sul, Departamento de Engenharia Elétrica
eschuler@eletro.ufrgs.br, carro@eletro.ufrgs.br, daniel.s.f@ufrgs.br

Abstract

As microelectronics evolves smaller into the nanometric scale, external interferences starts to be harmful to the system expected behavior. As classical systems do not handle adequately faults caused by such sources, new topologies are proposed. Our present work proposes a solution for this problem consisting on the use of sigma-delta modulation in order to obtain a fault-tolerance even in presence of multiple faults. This paper provides the mathematical analysis and demonstration to support the proposed approach.

1. Introduction

Silicon process for the production of microcircuits brought the possibility of low-power, huge transistor density and high performance chips. These advantages came with the critical drawbacks of system becoming more susceptible to external interferences, like heavy or high-energized particles. Due to the ever-decreasing gate length of the transistor, phenomenon with less energy turns to be harmful to the circuit behavior.

Such disturbances can be, for example, a Single Event Upset (SEU) or a Single Event Transient (SET). The SEU, causing bits inversions in digital systems (bit-flips), may change the system to a malfunctioning state. On the other hand, the SET, generating a transient pulse, which can be latched, may also cause interferences in the right system operations [1], [2].

Fault tolerant systems have been studied for a long time, and many solutions have been proposed such as the insertion of codes for error correction [3] and the use of software [4] or hardware [5] redundancy. However, none of these solutions is able to cope with simultaneous faults and, when they are able to do so, the cost increase makes the solution impractical, or at least very expensive. Also, as faults are increasing greatly with the new technologies for circuit production and, for submicron technologies, probabilistic operation of gates are expected, becoming the future scenario even worst, new solutions are needed.

Delta-sigma modulation proved to be a suitable solution for multiple faults occurrence, as we presented in past work [6]. This paper shows a deeper study of the modulator behavior when operating under severe conditions of bit-flips, and how these bits inversions will affect the applications responses.

In section II, a mathematical analysis is depicted, showing the consequences of bit-flips in digital sigma-delta modulators to the generated bit stream. These bit streams are the basis for the fault-tolerance circuits, since they are used instead of common n-bits code-modulated words. In order to demonstrate the advantages of using such approach to obtain fault-tolerant applications, section III presents results regarding the implementation of a digital filter using only sigma-delta modulated signals to represent both the input signal and the filter coefficients. Finally, section IV presents conclusion and future works.

2. Evaluating bit-flips in digital modulators

Soft errors, as seen by digital systems, can be, for example, bit-flips. This simple statement permits us to model the fault as a summed up number that changes the magnitude of the amplitude of the original signal by a power of two. Multiple faults can then be modeled summing powers of two, thus, allowing us to use any number for the magnitude of the fault inserted up to 2^n, where n is the input word length.

Discrete-time mathematics provides us a description of such fault as a *Dirac Delta Function*, or more commonly, impulse function ( ). Thus, the signal at a selected part of the circuit is modeled as (1).

$$y[i] = x[i] + k.\delta[i - \tau],\ k \leq 2^n \qquad (1)$$

In (1), x is the original signal (fault free) and is the time when the fault should occur.

This model can be applied in every signal of the system, stating a generalized framework for multiple SEU. Transforming (1) into the Z domain, one obtains equation (2).

$$Y(z) = X(z) + k.z^{-\tau} \qquad (2)$$

The importance of this result is that it is possible now to derive models for components with inserted faults. Such process, for example in a discrete integrator, is displayed in figure 1(a).

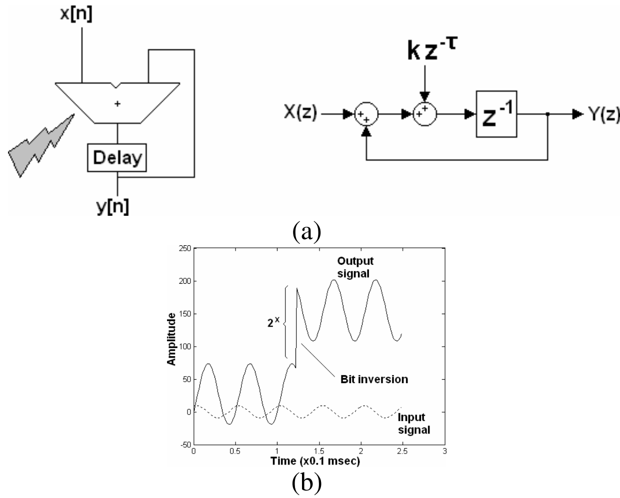


Figure 1. Discrete-time integrator with faults inserted.

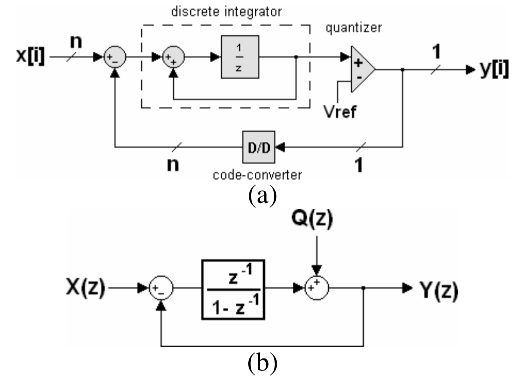


Figure 2. First order ΣΔ modulator scheme.

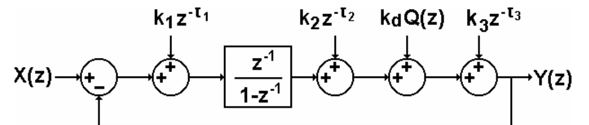


Figure 3. Complete analytical faulty model for the first order modulator.

The diagrammatic version of the discrete integrator has an extra term, which represents the fault, summed just before the delay unit. Writing the respective transfer function results in (3).

$$Y(z) = \frac{kz^{-\tau-1}}{1 - z^{-1}} + X(z)\frac{z^{-1}}{1 - z^{-1}} \quad (3)$$

A simple analysis presented in figure 1(b) exhibits the integration process where the fault appears as the addition of a certain value, proportional to the bit inverted by the fault. Applying this fault model to the sigma-delta modulators presented in figure 2(a), one has its model depicted in figure 2(b). The quantizer inhibits our ability to use circuit analysis techniques because of its non-linear behavior. The impossibility of finding a transfer function for this circuit resulted in the proposition of [7] in that, if the quantization error is largely uncorrelated from sample to sample and has the same probability of lying anywhere into the level spacing Δ/2 defined by the comparator, the output bit stream will represent the input signal with a certain resolution. This classical result has the consequence, on the sigma-delta topology, in the analytical scheme presented in figure 2(b).

Applying the fault model of figure 1(a) to each block of the modulator model in figure 2(b), one generates the final analytical model depicted in figure 3, upon which further conclusions will be made. For this model, *k1* represents faults being inserted in the modulator input adder, *k2* in the modulator integrator and *k3* in the modulator quantizer. From now on, the next steps are based on the equations extracted from the above circuit. The final transfer function of the model is given in (4).

$$Y(z) = X(z)z^{-1} + k_d Q(z)(1 - z^{-1}) + BF(z) \quad (4)$$

In which the *BF(z)* term represents the parcel corresponding to the fault, and are expressed by (5).

$$BF(z) = k_1 z^{-\tau_1} z^{-1} + (k_2 z^{-\tau_2} + k_3 z^{-\tau_3})(1 - z^{-1}) \quad (5)$$

Inversing the z-transform one obtains (6).

$$y[n] = k_d(q[n] - q[n-1]) + x[n-1] + \sum_{i=1}^{3} bfi[n] \quad (6)$$

In (6), bf[n] represents the faults, expressed by the addition of three terms, where each one represents the faults produced in the input adder, integrator adder and comparator, respectively. These terms are given in equations (7), (8) and (9):

$$bf_1[n] = k_1\delta[n - \tau_1] \qquad (7)$$

$$bf_2[n] = k_2(\delta[n - \tau_2] - \delta[n - \tau_2 - 1]) \qquad (8)$$

$$bf_3[n] = k_3(\delta[n - \tau_3] - \delta[n - \tau_3 - 1]) \qquad (9)$$

The general expected behavior for a delta-sigma modulator is represented by the two first terms in equation (4), i. e., the quantization noise translated to higher frequencies while the input signal stays in its original band. As mentioned, the parcel *BF(z)* in (4) describes the faults, represented by impulses translated in time. In (5), it is possible to realize that for faults occurring before the integrator, these will appear inside the signal band. However, when faults occur after or in the integrator, they will be moved outside the signal band. This is an expected behavior for linear circuits with feedback paths and an integrator in the forwarding path [8]. Simulations of the model extracted from the discrete-time equations (6)-(9), confirms this behavior. Faults in the input adder distorts the output bitstream generated by the modulator in the signal band, while faults in the discrete-time integrator, or in the quantizer, have their effects shifted to higher frequencies. Figure 4 shows these simulations results, where the input signal, a sinusoidal wave, was sampled with an *over sampling rate* (OSR) of 32 for both runs.

Results reached so far are based on a linear model of the modulator (see figure 2(b)) and form the foundation of our understanding. However, the assumption that the quantizer is modeled by an added noise is not valid anymore when the real circuit behavior must be evaluated. Since the circuit quantizer is a simple comparator, which uses the sign bit of the word as its output [9], the feedback process no longer acts as the predicted model. Taking the real circuit, thus non-linear, for simulation, the output of the discrete integrator is plotted just before the quantization step, and showed in figure 5.

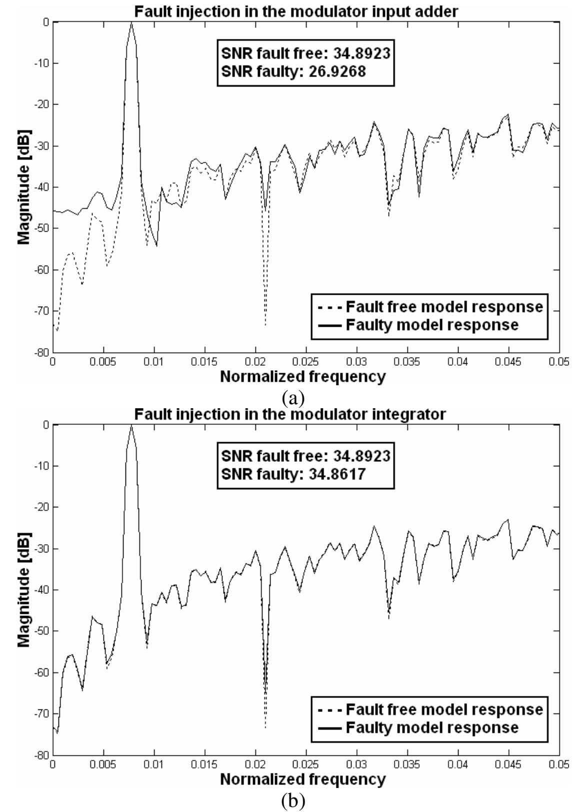


Figure 4. Simulation results for the model when faults are injected (a) in the input adder and (b) in the integrator.

It can be seen in figure 5(a) that the impulse function (added by a fault inverting the least significative bit in the output of the adder) is integrated, and keeps accumulated. For the linear model, one should expect this effect to eliminate itself, as predicted before. In the circuit, however, since the feedback is of a simple bit, the fault manifestation is obstructed by the quantizer and interfere in the bitstream only when the fault is large enough to invert the sign of the word. The consequence, showed in figure 5(b), is an interference occurring inside the signal band. Moreover, when the number of inversions increases due to the inversion of the most significative bit (see figure 5(c)), also the in-band interference increases, as plotted in figure 5(d).

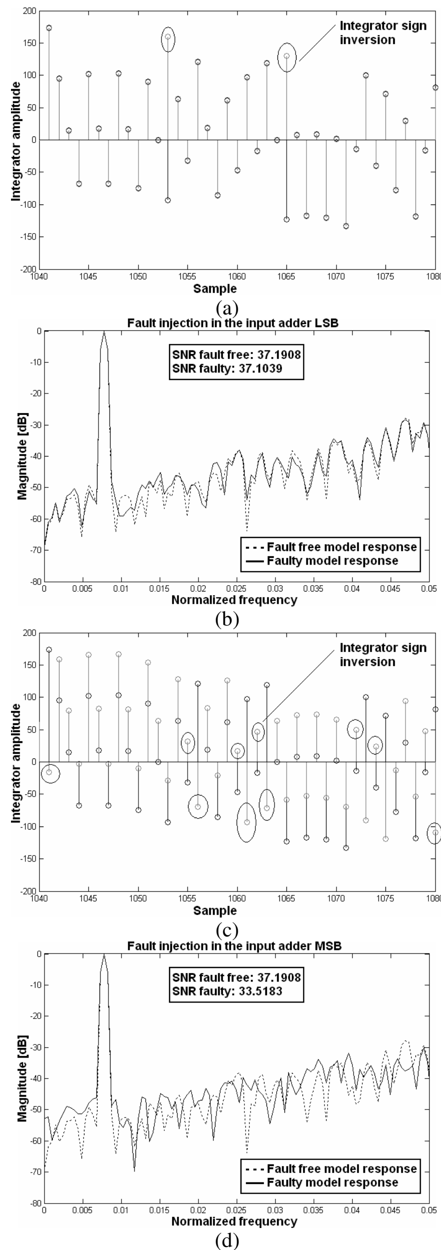


Figure 5. Integrator output signal. Difference in amplitudes reveals integrated step function.

One can conclude, thus, that for the circuit, no matter where faults occur, if their consequences are fed back, that is, if the fault causes an inversion in the integrator sign bit, the output signal can be more or less affected, depending on the number of inversions. However, even with the occurrence of the faults, the final signal-to-noise ratio (SNR) maintains a certain resolution that does not differ so much from a non-faulty behavior. This will be showed in next section, where a Finite Impulse Response filter is developed using signals only, and a huge amount of faults is injected, without compromise the final result.

3. Fault tolerant digital filter using modulated signals only

Despite the consequences of bit-flips in the sigma-delta modulator presented before, it will be shown that, even with the insertion of many faults in the modulation process, which can affect the output signal in one way or another, the final result for complexes applications may sustain a good resolution. This section presents the results regarding the implementation of a Finite Impulse Response (FIR) filter, which uses exclusively sigma-delta modulated signals, representing both the input signal and the filter coefficients. All results are from simulations, developed in Matlab®. The multiplication of the input signal and the coefficients is done by passing the filter coefficient value or the inverted coefficient value, depending on the sign of the input sigma-delta modulated signal. This is done via exclusive-or gates, as shown in figure 6. Once a number of coefficients equal to the number of taps is passed (inverted or not), the addition is done by interleaving the coefficients bit streams [10], resulting in one filtered point. In order to analyze the final values, the resultant bit stream of each filtered signal is decimated to obtain a decimal value. The input signal, formed by a 2KHz sinusoidal plus a white noise, is sampled and quantized with an OSR equal to 32, generating 1024 points with 8 bits plus a sign bit. Each of these points are passed by a digital sigma-delta modulator, as the one showed in figure 2(a). Thus, the bit stream that represents the input signal has 1024 points, too.

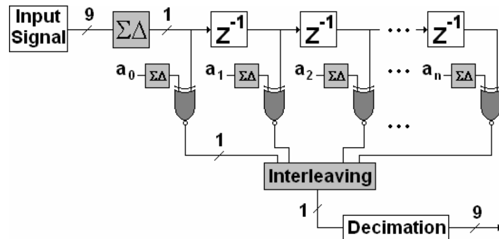


Figure 6. FIR filter using modulated signals only.

The coefficients are also represented in 8 bits plus a sign bit. To modulate each coefficient to the sigma-delta domain, each one is passed OSR times through the modulator, generating, thus, a bit stream with OSR bits for each coefficient. The faults are injected in both the input adder and the integrator of the modulator through the inversion of one random bit in randomly chose time periods. As showed in section II, each fault injected inside the modulator may cause the inversion of more than one bit in the output bit stream. So, the total amount of faults that are processed by the application (in this case the filter) is very higher than the injected one. Figure 7 shows the simulation for three different situations of faults injection. For figure 7(a), no fault occurs, and one gets a very good approximation of the signal filtered using a common 9-bits code-modulated words and the proposed approach, where each coefficient is modulated with an OSR of 64. Although the filter transfer function do not match exactly with each other in high frequencies, the matching is almost perfect in the pass band, what guarantees a very close SNR. For the second simulation, the same OSR is used to modulate each coefficient, however we now insert 8 faults during the modulation of each coefficient, that is, during 4 periods of 64 (the OSR for each coefficient), one random bit is inverted in the input adder and one in the integrator of the modulator. Also, 40 faults are inserted during the modulation of the input signal (20 for each part of the modulator). Figure 7(b) shows the results for the filter transfer function, which is affected only by the faults in the coefficients modulation process, the coefficients plot, also affected only by the 8 faults, and the filter output, where the faults consequences are due to the total amount of faults injected. Again, although the filter transfer function degrades in high frequency, the degradation in low frequencies is not enough to cause a great reduction in the final SNR. Finally, in order to show that an increase in the OSR used to modulate each coefficient can enhance the filter response, each coefficient is now modulated with an OSR equal to 128, while keeping the same number of faults. Results are showed in figure 7(c), where it is possible to see an increase in the SNR when compared to the previous experience and also an enhancement in the coefficients matching.

4. Conclusions

This paper presented an analysis regarding the occurrence of bit-flips in digital sigma-delta modulators, in order to evaluate their consequences to the bit stream. A model for this kind of fault was proposed and applied to the modulator circuit. As seen, the model presented a behavior just as the expected one, that is, for faults in blocks before the integrator they appear inside the signal band, while for faults in the integrator or after it, they are in high frequency and, thus, with very small consequences for the circuit response. When analyzing the circuit, the important think to be considered is the feedback value, which can have its value changed due to faults even in the integrator and in the input adder. Finite Impulse Response filter implemented using sigma-delta modulated signals only was simulated and, despite the huge amount of faults injected, results showed that the final signal-to-noise ratio sustains a very good approximation to a non-faulty behavior. Currently, different kinds of applications are been developed using sigma-delta modulated signals, in order to evaluate fault-tolerance for more complexes circuits like DSP processors, control systems and neural networks.

5. References

[1] Messenger, G.C. “A summary review of displacement damage from high-energy radiation in silicon semiconductors and semiconductor devices”. IEEE Transactions on Nuclear Science, Vol: 39, Issue: 3, June 1992. Pages: 468 – 473.

[2] Yang, F.L.; Saleh, R.A. “Simulation and Analysis of Transient Faults in Digital Circuits”. IEEE Journal of Solid-State Circuits, Vol: 27, Issue: 3, March 1992. Pages: 258 – 264.

[3] Plank, J. S. “A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems”. Department of Computer Science, University of Tennessee.

[4] Anghel, L.; Alexandrescu, D.; Nicolaidis, M. “Evaluation of soft error tolerance technique based on time and/or space redundancy”. 13th ICSD, Brazil, Sept 2000. p.237-242.

[5] Chande, P.K., Ramani, A.K., Sharma, P.C., “Modular TMR Multiprocessor System”. IEEE Transactions on Industrial Electronics. vol.36, n.1, Feb.1989.

[6] Schüler, E., Carro, L., “Reliable Digital Circuits using Sigma-Delta Modulators”. 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT’05). Monterey, USA, 2005. Oct. 03-05, 2005.

[7] Norsworthy, S.R.; Schreier, R.; Temes, G. C. “Delta-sigma data converters: theory, design and simulation”. New York: IEEE Press, 1997. 476p.

[8] Ogata, Katsuhiko. “Modern Control Engineering”. Englewood Cliffs, NJ: Prentice Hall, 1990.

[9] Robert, G. W.; Lu, A. K. “Analog Signal Generation for Built-In-Self-Test of Mixed-Signal Integrated Circuits”. Kluwer Academic Publishers, 1995. 122p.

[10] Maloberti, F. “Non Conventional Signal Processing by the use of Sigma Delta Technique: a Tutorial Introduction”. Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), Volume: 6, May 1992. Pages: 2645 – 2648.

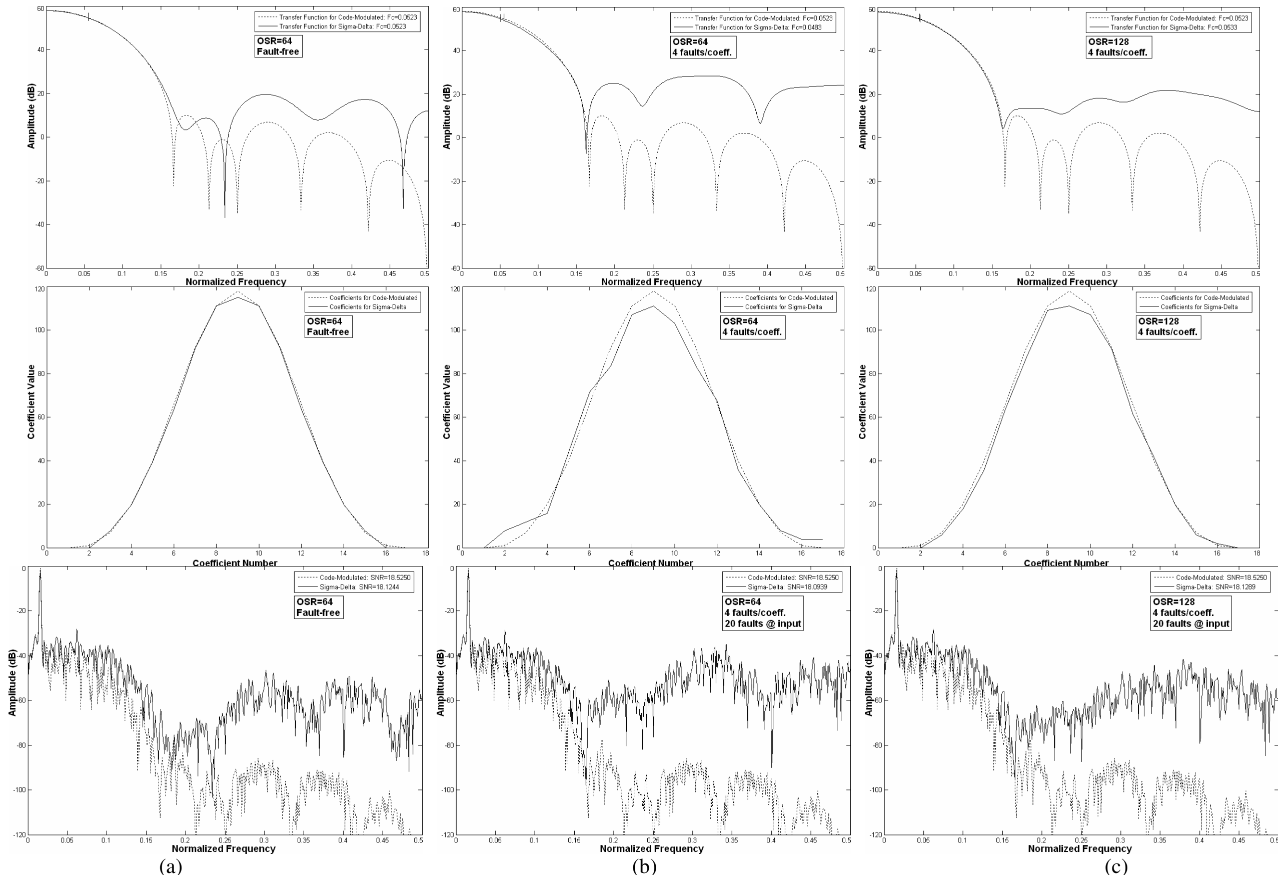


Figure 7. Filter simulation for different number of faults injected and different OSR used to modulate each filter coefficient.

Reliable Digital Circuits Design using Sigma-Delta Modulated Signals

Erik Schüler, Luigi Carro
Universidade Federal do Rio Grande do Sul
Departamento de Engenharia Elétrica
{eschuler, carro}@eletro.ufrgs.br

Abstract

As the transistor gate length goes straightforward to the sub-micron dimension, the possibilities of occurrence of external interferences in these devices also increase. Moreover, the process variability will further degrade this scenario. The direct effect of such external and/or intrinsic interferences is, in many cases, the total mismatch between the desired answer of the system and the achieved answer resulting from single bit flips. This way, new techniques must be studied in order to guarantee the correct operation of these systems. This work presents the use of a totally digital sigma-delta modulator that is used to develop arithmetic operations, which are further used to develop a FIR filter. Simulations results show that, even with the insertion of a large amount of faults, one can still obtain a non-faulty behavior in the SNR of complex application.

1. Introduction

The MOSFET gate length is expected to be 22nm in the year of 2008. This way, more than one billion of transistors will be able to be integrated in a single die [1]. With this number of devices, there is a clear possibility of creating systems with more and more resources. However, with such a small gate length, also the possibility of occurrence of some problems that were not expected to occur in systems a few years ago arises. One of the most important problems is the single event upset or simply SEU [2, 3], which have already been detected in many situations.

As the size of the transistor channel decreases, also the number of electrons (or holes) in the channel decreases. With a lower number of carriers passing through the transistor channel, although the possibility of these carriers being hit by an external particle (e.g. alpha or neutron particles) decreases, if such event occurs, the effect caused will be much higher than if there was a higher number of carriers. Such interference may cause a SEU, which can cause, for example, a single bit flip that may damage the system or subsystem behavior in such way that the achieved answer can differ a lot from the expected one.

Also, as the frequency of operation of these circuits increases, even a very short time transient pulse caused by a radiation and/or electromagnetic noise may affect the output of combinational circuits, as well as the values stored in memory elements.

These phenomena may occur in digital, analog, and even optical components, or may have effects in surrounding circuitry. Such events, causing transient random faults, forces one to search for new design approaches to reduce the effects of these faults, which can occur even in fully tested and approved circuits.

This work presents a technique that consists of modulating the signal to be processed in the sigma-delta domain, in such a way that the redundancy presented in the modulated signal can reduce the effects of the faults occurred during the system operation, thus allowing one to

build more reliable circuits, regarding SEUs. In a practical application like a digital filter, even with several simultaneous faults the impact on the total SNR was close to zero.

This paper is divided as it follows: section II makes a review of some techniques used to improve the performance of systems subject to faults. In section III, it is shown that signals modulated in sigma-delta domain are in fact more tolerant to faults. In section IV it is presented a comparison between the proposed sigma-delta approach and another solution, which uses the redundancy principle. In both cases, multiple simultaneous faults are injected, showing that the first solution presents much better results. In order to show that a small deviation from the original value does not interfere so much in the final result of some applications, section V shows some results regarding the implementation of a digital FIR filter using the proposed technique. Finally, section VI presents the conclusions and some future work that are in process of development.

2. Related works

Many techniques have been developed to cope with the mentioned SEU faults. In [4], two low-cost solutions to cope with SEU are compared: the error-detection capabilities of a hardware-implemented solution based on parity code and software-implemented solution based on source-level code modification.

The implementation of a new soft error tolerance technique based on time redundancy and/or space redundancy is presented in [5]. Also, the use of triple modular redundancy (TMR) is discussed and proposed in [6] and [7], which also proposes the use of double modular redundancy.

These techniques were developed and presented good results when single soft errors were presented. Recently, it was presented in [8] a technique that is able to cope with multiple soft errors. Such technique is based on the use of larger redundant words (bit-streams) to represent the signals to be processed, and is supposed to be used in systems that work with the concept of error tolerance proposed in [9].

The final goal of error tolerance consists basically in dealing with systems that are error tolerant regarding their final application. This means that, even under the presence of multiples faults, which could cause internal and/or external errors, the systems still produce acceptable results. Thus, using this concept, for some applications, a certain degree of error is acceptable.

This work is based on the same idea presented in [8], however, the generation of the bit-streams is developed through the use of a sigma-delta modulator. When compared with the bit-stream process generation presented in [8], which is based on the use of the probability associated with some digital quantity, the use of sigma-delta modulation can lead to a better representation [10]. As a result, the reduction in the necessary length of the bit-stream that represents the quantity can be achieved, without loosing signal resolution.

The results presented in this work make the use of a digital first order sigma-delta modulator that converts an eight bits input signal to its sigma-delta domain representation, for further computation. All results were achieved through Matllab® simulation.

3. Fault tolerance in the sigma-delta domain

Sigma-delta-modulated signals, although requiring over sampling, have a great advantage if compared with code-modulated signals, regarding fault tolerance: since sigma-delta-modulated signals are over sampled, each bit-stream that represents an amplitude value carries an amount of redundancy that, even with some faults inserted, can still represent the mean value of the original signal.

Figure 1 shows an example, where a 20KHz sinusoid signal is modulated by the digital first order sigma-delta modulator showed in figure 2 [11], with an over sampling rate (OSR) of 16. It can be seen that a good SNR is maintained even after the insertion of a different number of faults, in different bits of the subtractor and integrator of the sigma-delta modulator (see figure 2).

Faults insertion is done during the modulation process, in a random time, through the inversion of one bit in the input subtractor and in the integrator adder. In the first case, shown in figure 1(a), no fault is injected. The tones that appear beyond the input one, are known as idle tones [10], and do not have any relation with faults injections. In figure 1(b), one can see that, with the injection of one fault done through varying the least significative bit of the adders one time, the spurious free dynamic range (SFDR) maintains practically the same. Finally, when the inverted bit is the most significative one (figure 1(c)), there is a small variation in the spectral shape, but with no addition of spurious tones into the signal band.

Figure 3 shows a sequence of simulations, where the decimated output values of the sigma-delta converter are plotted when varying the input signal over the input range, which is determined by the number of bits of the modulator adders. That is, if one has an n bits input signal and it is used a two's complement representation for the negative numbers, it is possible to show that the input subtractor and the integrator adder must have n+1 bits.

As one can see in figure 3(a), the modulator output values do not correspond exactly to the input signal for all input values, due to the quantization error. However, it is possible to achieve a very good approximation of such value, which can be used in systems that do not require a very great precision, or in applications that tolerate small error percentages, and even in systems that use cascading calculations, like digital filters, where such approach can still be used with an acceptable interference in the final result, as it will be shown in section IV. Also, this approximation can be improved through the increase of the OSR, but not without some time penalties.

Looking at figure 3(b), when one fault is injected in the modulator (both in the input subtractor and in the integrator), the output values maintain a certain mean value that still approximate to the input one. Even when the number of faults is increased to four (figure 3(c)), one still has a good I/O relation, which, as mentioned, can be improved by increasing the modulator OSR, as shown in figure 3(d).

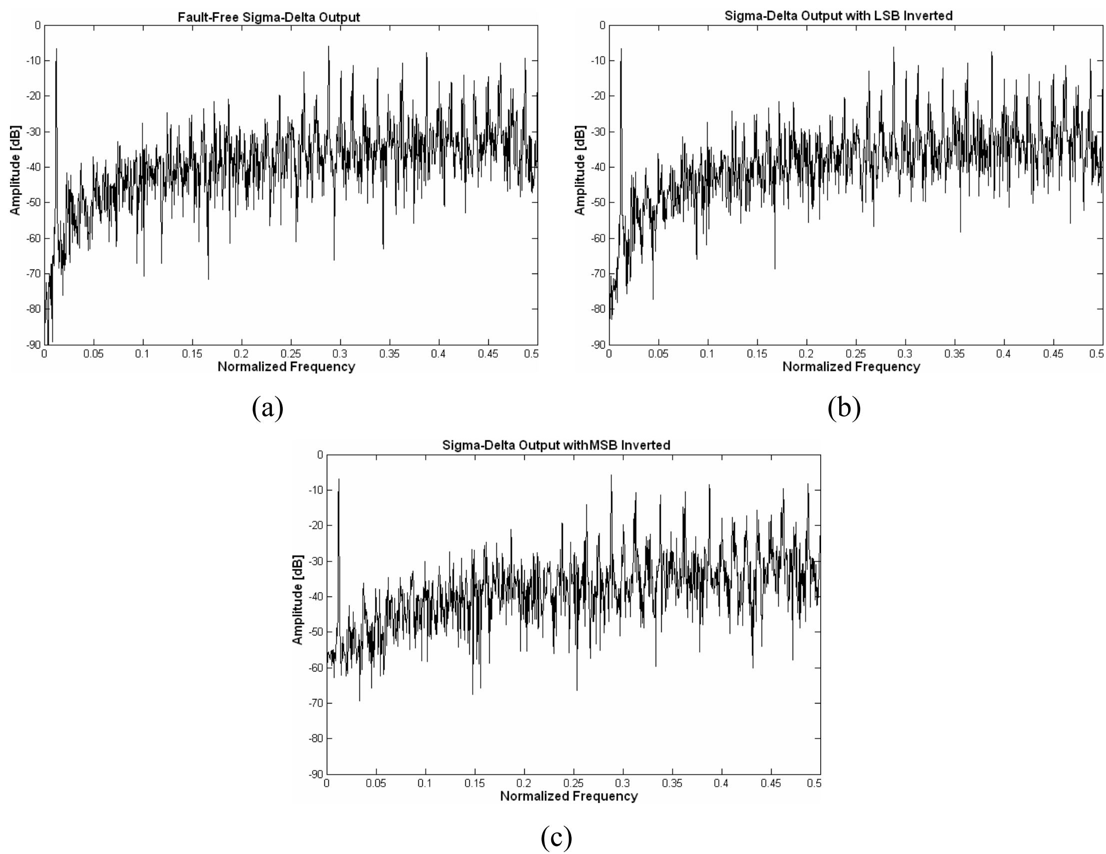


Figure 1: Frequency domain of a 20KHz single tone modulated by a digital first order sigma-delta modulator (a) with no faults, (b) with one fault injected in the LSB of the modulators adders and (c) with one fault injected in the MSB of the modulators adders.

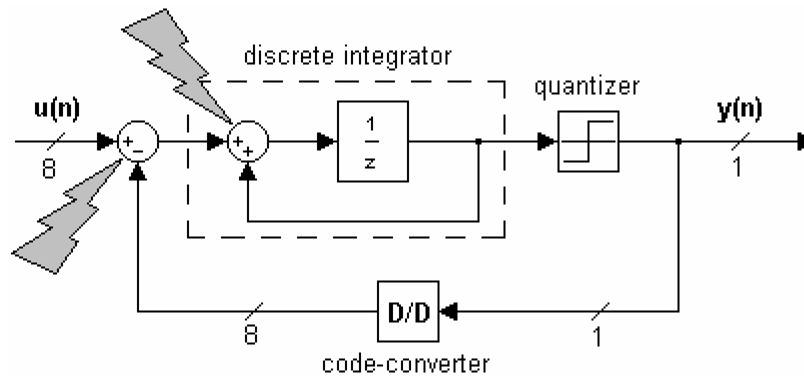


Figure 2: Digital first order sigma-delta modulator, according [11], showing where faults were injected.

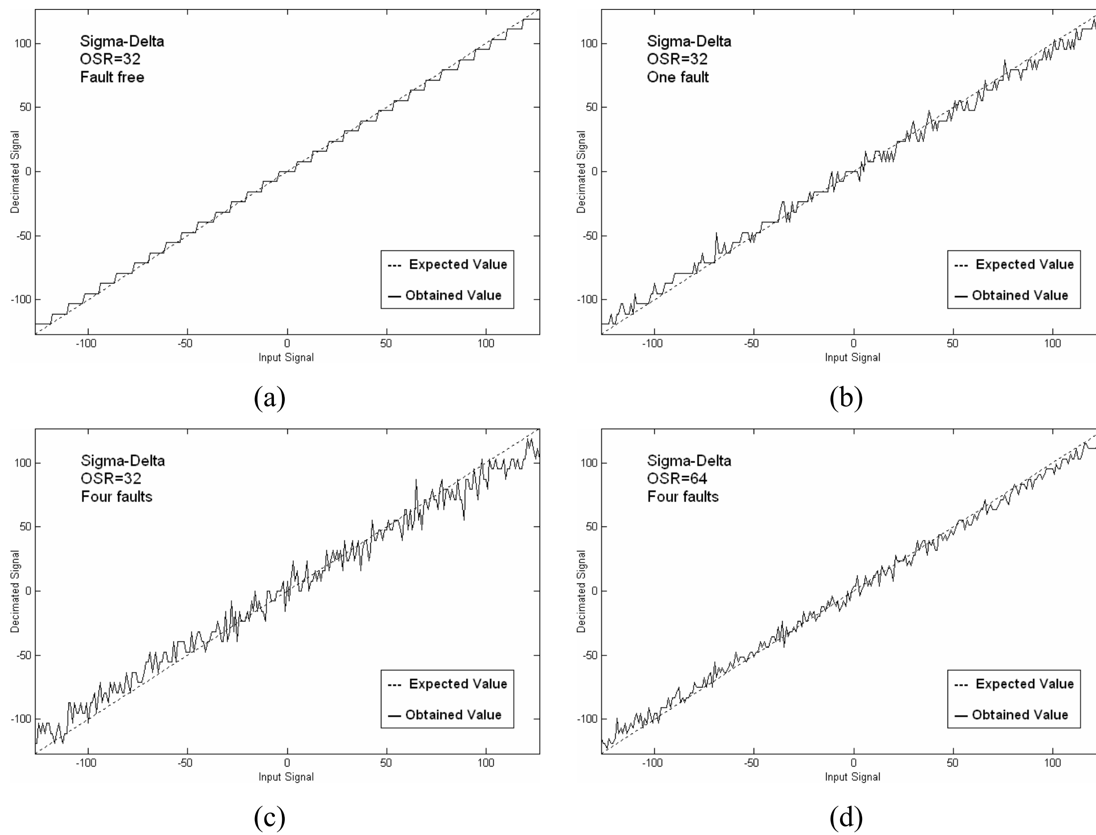


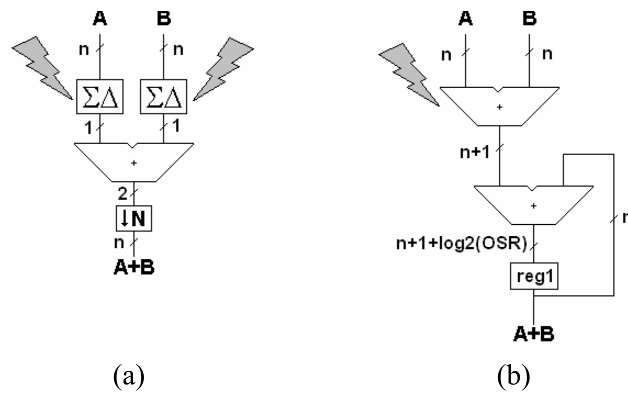
Figure 3: Sigma delta input/output relation for different number of faults injected.

4. A comparison between sigma-delta and exhaustive addition

Before continuing with the proposed approach using sigma-delta modulated signals, let us consider another way that can lead to a good final result when multiple faults are injected. This approach, such as the sigma-delta one, makes use of temporal redundancy, and could be seen as an alternative to the modulator. However, it will be shown that the modulator is still a better choice, once the fault tolerance presents a better performance.

In [12] it is proposed the signal processing using signals in the sigma-delta domain. Addition, multiplication and Boolean operators have been proposed and demonstrated. In this work, however, we are mainly interested in adding two bit-streams, since with this operation, it is possible to develop even multiplication, square roots and other arithmetic operations, which can be used to develop more complexes applications like IIR and FIR filters. So, consider the addition of two numbers. As proposed in [12], if one modulates both input signals into sigma-delta domain and makes the addition of the outputs bit streams, the achieved result is a 2 bits word stream, which can be decimated and generate a very good approximation to the signals addition. However, instead of modulating the signal to be processed in the sigma-delta domain, and thus achieve sufficient redundancy to make this signal robust to faults, one could possibly think in adding the signal for N times, and then divide the addition result by N, thus obtaining the expected value, too.

If the values to be added are A and B, after passing each of these values by the sigma-delta modulator, one has in the output two bit streams that correspond to the mean value of A and B. The number of bits in each bit stream is proportional to the modulator OSR. So, if N is substituted by OSR, the number of additions in the other proposition also becomes



(a) (b)

Figure 4: Approaches used to add two constant values using (a) sigma-delta modulator and (b) add OSR times.

proportional to the over sampling rate. Figure 4 presents the proposed approaches to add two constant values using the proposed solutions.

As one can see in figures 5(a), 5(b) and 5(c), the sigma-delta approach is in fact more fault tolerant than the solution using the large redundancy achieved through adding the signal OSR times. These figures compare the results achieved by adding two constant values (40 and 60), where faults are injected in the sigma-delta modulation (as showed in figure 4(a)), and in the add OSR times solution (figure 4(b)). Faults are injected by inverting a random bit n times during the modulation and the addition process. For the sigma-delta, the faults are injected both in the subtractor and in the integrator.

In figure 5(a) one fault is injected and, as it can be noted, although the achieved response is not exactly equal to 100 (40+60), the mean value approximates a lot for the sigma-delta solution, no matter which bit is inverted in the fault occurrence, that is, even if the LSB or the MSB is inverted, the final value is close to the right one. In the other hand, for the second proposition, although betters results are achieved, depending on which bit the fault occurs, the results can be very different from the exact one.

In figure 5(b), the number of faults is increased to 4, and now, the difference between the solutions is even more evident. When maintaining the number of faults in 4 and increasing the OSR, the sigma-delta results get an even higher resolution, while the addition results continue to deviate very much from the right value, as can be noted in figure 5(c).

So, as one can conclude, the sigma-delta approach is a better choice than the proposition of adding the signal OSR times regarding fault tolerance. If one compares the area overhead of both solutions to add two numbers, the modulator solution would possibly occupy a higher area, since one must use two modulators. But, for more complexes applications where the use of addition is higher, once all signals are in the sigma-delta domain, this area overhead becomes less significant. For example: consider a system where lots of addition operations must be developed. A fast comparison between the areas occupied by the two solutions presented will be developed. For this comparison, we will consider the overhead due to the sigma-delta modulator and due to the add OSR times approach.

If an n bits input signal is to be used, then the sigma-delta area is equal to the area of two (n+1)bits adders, one (n+1)bits flip-flop to make the integrator delay, a one bit exclusive-or gate to the comparator and a pair of n bits tri-state buffers to make the feedback D/D converter. The areas of such elements grow linearly with the number of bits n of the input

signal. For the second technique, one has to consider only the area of two adders and of a flip-flop to the register. However, the areas for the second adder and for the register do not depend exclusively on the number of bits of the input signal, but also on the OSR used. If one consider that the OSR to be used with both, the modulator and the adder, grows exponentially (8, 16, 32, 64...), then number of bits to be used in the second adder and in the register should be n+1+log2(OSR). That is, the second adder and the register areas grow logarithmically with the OSR increase. For the sigma-delta modulator, however, the area does not depend on this factor.

Equations (1) and (2) shows the relations between the two techniques areas, while figure 6 shows how the areas grow with the increasing of the OSR, for two different number of input bits. As noted, for any value of OSR higher than 2, the area overhead due to the modulator is lower than that due to the adder.

$$A_{\Sigma\Delta} = 126.n + 132 \quad (1)$$

$$A_{\Sigma OSR} = 76.\log_2(OSR) + 118.n + 118 \quad (2)$$

Now that it was demonstrated that the sigma-delta solution is better than the exhaustive addition one, even regarding fault tolerance and area overhead, we shall continue to show some others applications that use sigma-delta modulated signals, and demonstrate that they are fault tolerant when utilized in more complexes applications.

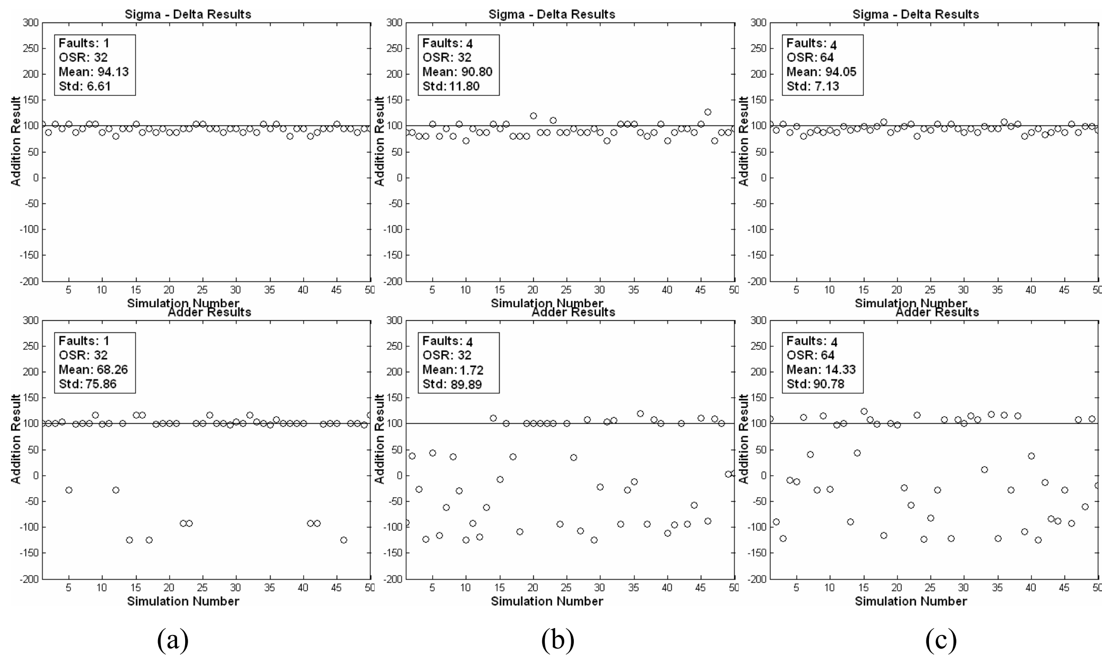


Figure 5: Results obtained by adding two constant values using two different approaches. In (a), with one fault @OSR 16, in (b) with 4 faults @OSR 32 and in (c) with 4 faults @OSR 64.

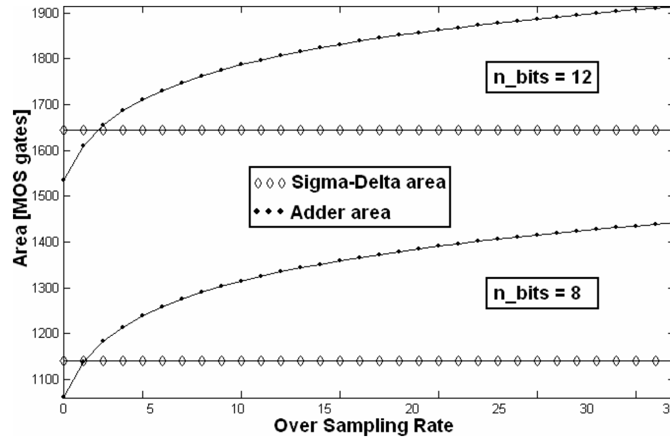


Figure 6: Areas of the sigma-delta and the add OSR times approaches.

5. Sigma-delta-modulated signals in fault tolerant filter development

As seen in section IV, for the development of addition operation using sigma-delta-modulated signals, the final mean value is very close to the expected value, even after the insertion of many simultaneous faults. We have showed in [13] that others operations like multiplication and square roots are also more fault tolerant when utilizing the modulation proposition. So, one could think in extending this principle to some systems that make use of intensive arithmetic operations, mainly those that use various cascading blocks, like filters.

This section presents some previous results regarding the use of sigma-delta-modulated signals in more complexes applications when faults are injected during the modulation process. The proposition is to analyze the convolution case, which is the basis of filters operation. Figure 7 shows the structure used to filter noisy signals using the proposed technique. As one can see, faults are injected in the sigma-delta modulator, also through the inversion of one random bit in the modulator adders.

Some techniques to filter signals in the sigma-delta domain have already been proposed [14, 15, 16]. In this case, once we are modulating each of the quantized signal to the sigma-delta domain, the filter structure should have some modifications. However, one can still benefits from the fact that the generated bit streams are a sequence of 0 and 1, then, one can just chooses whether he wants or not add the coefficients values, instead of making a multiplication of the input signals by these coefficients. An 8 taps low pass FIR was used in the simulations.

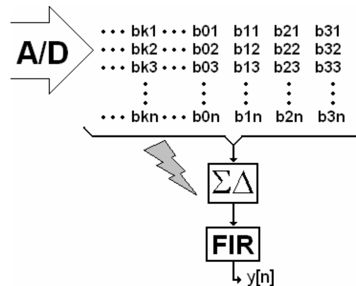


Figure 7: Representation of the structures used to filter signals using sigma-delta-modulated signals.

Simulations results obtained are presented in figure 8. The input signal formed by a single 20KHz tone and an amount of uniformly distributed random noise was sampled and filtered in two different ways: using the samples themselves and a common FIR filter without any fault inserted and using a sigma-delta modulation of the samples with the insertion of faults. All simulations results were done by taking 512 points of the input signal and modulating each of these signals. Faults were injected during the modulation of 32 of these 512 points. So, if one has n faults injected during the modulation process, the total amount of faults is given by 32n.

Figure 8(a) shows the input signal in the time and frequency domain. In figure 8(b), it is shown a comparison between the output signals being filtered using a common FIR and the proposed technique with no faults. Then, figure 8(c) presents these same results when four faults are injected in the modulation process. As mentioned, these four faults are injected in the modulation of 32 points, resulting then in 128 bits inverted during this process. For figures 8(b) and 8(c), the OSR used to modulate each of the sampled points is 16. In figure 8(d), however, this OSR is increased to 32, showing that an even better resolution is achieved, even with the insertion of 4 faults. The code-modulated FIR results presented in these figures are just to make comparisons so, no faults are injected in these simulations.

6. Conclusions

This work presented the use of sigma-delta modulators in the generation of fault-tolerant signals, intended to be used in mean-based systems that make intensive use of arithmetic operations. It was shown that systems that use such technique can handle multiple faults, while still producing results that are very close to those produced when using fault-free code-modulated signals.

A comparison between this technique and the addition of a signal OSR times with further division by OSR was presented, showing that the sigma-delta modulation proposition generates better results, with a small deviation from the expected values, but with a final mean value that is much closer than those obtained using the previous solution. Even when comparing the area overhead, the sigma-delta solution presents a lower impact.

In order to show that such small deviation from the original value does not cause a great disturb in the final result of a system that uses a large amount of arithmetic operations, a FIR filter was developed. Signals modulated in the sigma-delta domain were filtered with the insertion of a great amount of faults, and compared with fault-free code-modulation FIR, showing that the tolerance to faults in the sigma-delta solution is good enough to obtain results with a very small difference between the SNR of these two solutions.

Although the proposed approach leads to the necessity of high frequency systems, since the sigma-delta modulator needs high over sampling rates to achieve a good resolution, the high tolerance to multiple faults can make its use advantageous. However, as mentioned, for systems that must multiply the input bit-stream by a certain value, since the bit-stream if formed by a sequence of zeros and ones, the multiplication can be substituted by a simple addition, what can contribute for the reduction of the area taken for processing.

Future works include the investigation on the effects of the dynamic reduction of the over sampled ratio, in order to adjust the amount of tolerance to errors with the rate of errors present in the system, since particles do not reach the system at constant intervals. Also, others parameters like power dissipation and processing-time must be evaluated.

7. References

[1] Semiconductor Industry Association. International Technology Roadmap for Semiconductors - ITRS 2004 last access Jan. 26, 2005. http://www.itrs.net/Common/2004Update/2004Update.ht

[2] Yang, F.L.; Saleh, R.A. "Simulation and Analysis of Transient Faults in Digital Circuits". IEEE Journal of Solid-State Circuits, Volume: 27, Issue: 3, March 1992. Pages: 258 – 264.

[3] Messenger, G.C. "A summary review of displacement damage from high-energy radiation in silicon semiconductors and semiconductor devices". IEEE Transactions on Nuclear Science, Volume: 39, Issue: 3, June 1992. Pages: 468 – 473.

[4] Rebaudengo, M.; Reorda, M. S.; Violante, M.; Nicolescu, B.; Velazco, R. "Coping with SEUs/SETs in Microprocessors by means of Low-Cost Solutions: A Comparative Study". IEEE Transactions on Nuclear Science, Volume: 49, Number: 3, June 2002. Pages: 1491 – 1495.

[5] Anghel, L.; Alexandrescu, D.; Nicolaidis, M. "Evaluation of soft error tolerance technique based on time and/or space redundancy". Proceedings of 13rd IEEE Symposium on Integrated Circuits and Systems Design (ICSD 2000), Manaus, Brazil, September 2000. Pages: 237 – 242.

[6] Yoon, J.; Kim, H. "Time-redundant recovery policy of TMR failures using rollback and roll-forward methods". IEEE Proceedings Computers and Digital Techniques, Volume: 147, Issue: 2, March 2000. Pages: 124 – 132.
[7] Lima, F.; Carro, L.; Reis, R. "Techniques for Reconfigurable Logic Applications: Designing Fault Tolerant Systems into SRAM-based FPGAs". Proceedings of the International Design Automation Conference, DAC 2003, ACM, New York, 2003. Pages: 650 – 655.

[8] Lisbôa, C.; Carro, L. "Arithmetic Operators Robust to Multiple Simultaneous Upsets". Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems - DFT 2004, pp. 289-297, ISBN0-7695-2241-6. IEEE Computer Society, New York, October 2004.

[9] Gupta, S. K.; Breuer, M. A.; Mak, T. M. "Defect and Error Tolerance in the Presence of Massive Numbers of Defects". IEEE Design & Test of Computers, Volume: 21, Issue: 3, May 2004. Pages: 216 – 227.

[10] Norsworthy, S.R.; Schreier, R.; Temes, G. C. "Delta-sigma data converters: theory, design and simulation". New York: IEEE Press, 1997. 476p.

[11] Robert, G. W.; Lu, A. K. "Analog Signal Generation for Built-In-Self-Test of Mixed-Signal Integrated Circuits". Kluwer Academic Publishers, 1995. 122p.

[12] da Fonte Dias, V. "Sigma-delta signal processing". IEEE International Symposium on Circuits and Systems, 1994. ISCAS'94, Volume: 5, 30 May-2 June. Pages: 421 – 424.

[13] Schüler, E.; Carro, L. "Increasing Fault Tolerance to Multiple Upsets Using Digital Sigma-Delta Modulators". Accepted for publication on the 11th IEEE International On-Line Testing Symposium, 6-8 July, Saint Raphael, France. 2005.

[14] Wong, P.W.; Gray, R.M. "FIR filters with sigma-delta modulation encoding". IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume: 38, Issue: 6, June 1990. Pages: 979 – 990.

[15] Thompson, A.C.; O'Shea, P.; Hussain, Z.M.; Steele, B.R. "Efficient single-bit ternary digital filtering using sigma-delta modulator". IEEE Signal Processing Letters, Volume: 11, Issue: 2, Feb. 2004. Pages: 164 – 166.

[16] Peled, A.; Bede Liu. "A new approach to the realization of nonrecursive digital filters". IEEE Transactions on Audio and Electro acoustics, Volume 21, Issue 6, Dec 1973 Page(s):477 – 484.

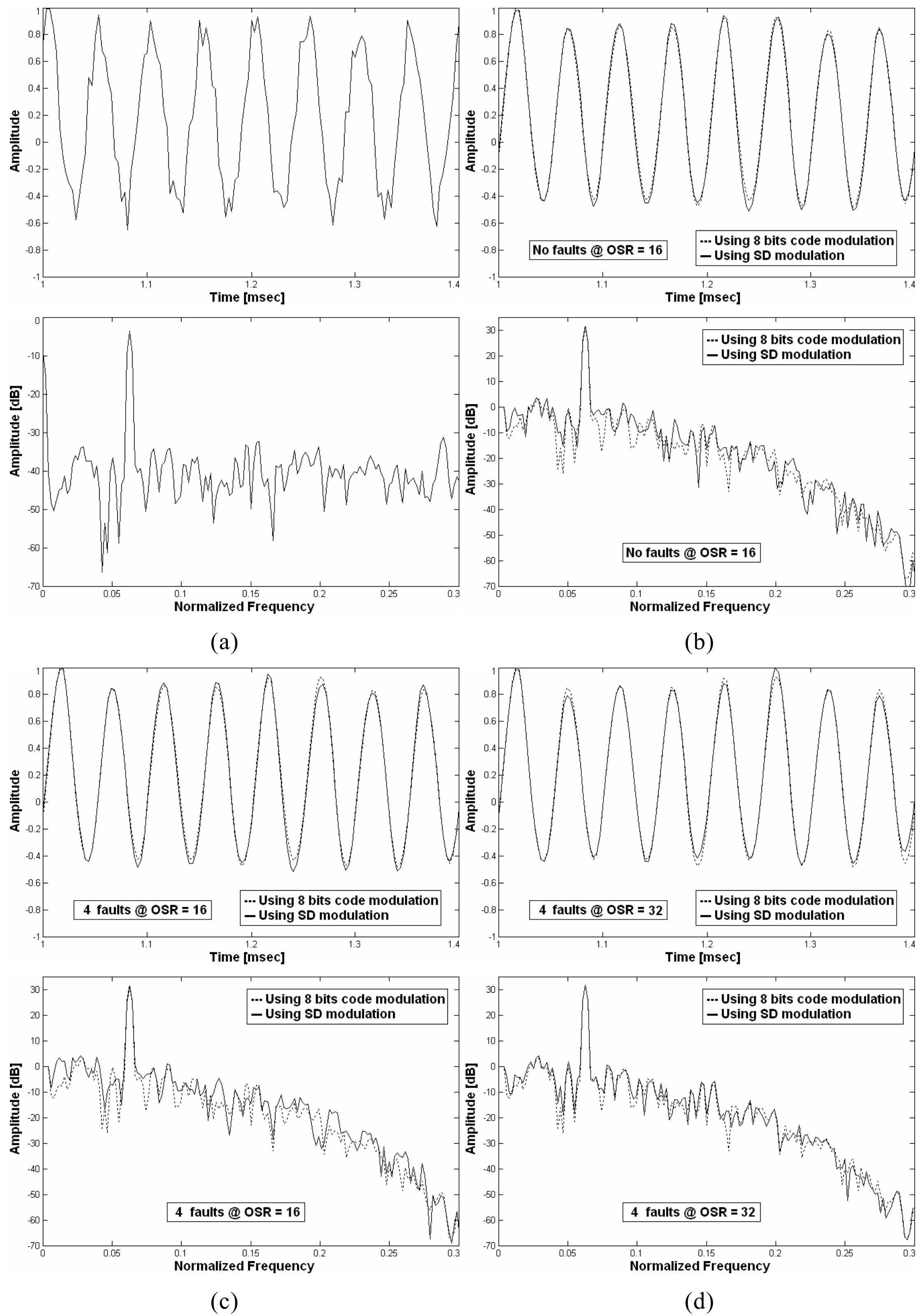


Figure 8: Results obtained by filtering a noisy 20KHz signal using sigma-delta modulated signals. In (a), the input signal, in (b), the filtered signal with no faults @OSR 16, in (c) with 4 faults @OSR 16 and in (d) with 4 faults @OSR 32.

Going Beyond TMR for Protection Against Multiple Faults

Lisbôa, C. A. L.
Instituto de Informática
Av. Bento Gonçalves, 9500, Bloco IV
91501-970 - Porto Alegre, RS, Brasil
+55 51 3316 7748
calisboa@inf.ufrgs.br

Schüler, E.
Depto. de Engenharia Elétrica
Av. Osvaldo Aranha, 103/206-B
90035-190 - Porto Alegre, RS, Brasil
+55 51 3316 3516
eschuler@eletro.ufrgs.br

Carro, Luigi
Depto. de Engenharia Elétrica
Av. Osvaldo Aranha, 103/206-B
90035-190 - Porto Alegre, RS, Brasil
+55 51 3316 3516
carro@eletro.ufrgs.br

ABSTRACT
Future technologies will present devices so small that they will be heavily influenced by electromagnetic noise and SEU induced errors. Since many soft errors might appear at the same time, classical fault tolerance techniques, such as TMR, will no longer provide reliable protection and will make new design approaches necessary. This study shows that the TMR approach has intrinsic weaknesses that impair its effectiveness in the presence of multiple faults, and proposes a new technique that provides better protection than TMR for single as well as multiple faults. The proposed technique is based on the use of some analog components among the digital circuits. We present results based on a multiplier, and show that the technique is scalable to withstand higher quantities of simultaneous faults.

Categories and Subject Descriptors
B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance

General Terms
Design, Experimentation, Reliability.

Keywords
Design techniques, fault tolerance, future technologies, simultaneous transient faults.

1. INTRODUCTION
As the microelectronics industry moves towards deep sub-micron technologies, systems designers become increasingly concerned about the reliability of future devices, which will have propagation delays shorter than the duration of transient pulses induced by radiation attack, as well as smaller transistors, which will be more sensitive to the effects of electromagnetic noise, neutron and alpha particles that may cause transient faults, even in fully tested and approved circuits.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SBCCI'05, September 4-7, 2005, Florianópolis, SC, Brazil.
Copyright 2005 ACM 1-59593-174-0/05/0009…$5.00.

In order to survive in this new scenario, it is clear that new fault tolerance techniques must be defined, not only for safety critical systems, but to general purpose computing as well. Current fault tolerance techniques are effective for single event upsets (SEUs) and single event transients (SETs). TMR, one of the classical techniques for protection against single faults [7], does not even stand a single fault in the voter circuit. Besides that, current techniques are unlikely to withstand the occurrence of multiple simultaneous faults that is foreseen with new technologies [4, 5].

To face this challenge, either completely new materials and manufacturing technologies will have to be developed, or fully innovative circuit design approaches must be taken. In the particular case of this contribution, since we are dealing with a somewhat analog behavior of the digital circuit in the presence of faults, we try to bring some analog knowledge to the design of robust digital circuits.

In this paper we use an array multiplier as an example to show that the TMR approach fails when single stuck-at-0 or stuck-at-1 faults are injected in some components of the voter circuit, and that the probability of faults being propagated to the output is never equal to zero. Then, we propose an alternative solution, which combines digital and analog circuitry to make TMR completely reliable in the presence of such faults, with minimum additional area overhead. The same solution is also applied to a different multiplier architecture with five redundant modules, which is proven to withstand multiple soft faults.

This paper is organized as follows: section 2 describes related work. Section 3 reports the results obtained in the implementation of a 4×4-bit array multiplier, using the conventional TMR approach, when single transient faults are injected, while in section 4 we describe the new analog voter used in this work, based on an analog comparator. In section 5 we use the device presented in section 4 to create a reliable TMR solution. In section 6 we use the same device to implement another multiplier architecture, that withstands up to two simultaneous transient faults. Section 7 discusses the results obtained in the experiments and our plans for future work on this project.

2. RELATED WORK
The possibility of increased incidence of soft errors due to noise or high-energy particles in the next technology generation is already a topic of research [4, 5, 15]. These soft errors are not caused by poor design techniques or process defects, but rather they derive from the incidence of external radiation and/or electromagnetic noise, which become more critical as technology

features shrink, and there are fewer electrons to form the transistor channel.

What turns soft errors into a major concern nowadays is that the higher frequencies to be reached by future circuits will lead to cycle times shorter than the duration of transient pulses caused by radiation and/or electromagnetic noise, which will have a higher probability of affecting the output of combinational circuits, as well as the values stored in memory elements. Besides that, shrinking transistor dimensions and lower operating voltages will make circuits more sensible to neutron and alpha particles, which also induce transient pulses.

Several techniques to maintain circuit reliability even under those critical conditions have been proposed, including hardware implemented parity code and source level code modification [16], time/space redundancy [1, 2], triple modular redundancy (TMR) and double modular redundancy with comparison (DWC) with concurrent error detection (CED) [8]. However, all these techniques are targeted to the occurrence of a single upset in a given time interval.

The concept of "error tolerance" and the relaxation of the 100% correctness requirement for devices and interconnects is proposed in [6] as an alternative to increase yield level for future technologies. That work is focused mainly in the concept of selecting different parts with different defect levels during manufacturing test, but no provision is made regarding reliable operation in the field, when soft errors will occur.

An alternative to tolerate multiple faults has been proposed in [13], where the authors aim at detecting and recovering from single or multiple manufacturing faults and in-field errors, which might occur in the same cycle of a microprocessor circuit. However, as the authors of [13] mention themselves, the use of Berger code prevents the use of the technique for data paths beyond 16 bits, and only multiple faults with the same polarity are supported. For the control logic, an application-specific error detection scheme is proposed, which basic concept is to back up only the control logic of a standard processor that is necessary for the instruction subset used by the application. The use of two processors in a master-trailer scheme, where both processors have built-in self check facilities and use micro rollback with a distance of one cycle, allows the detection and recovery from errors due to transient effects within one cycle (single-event upset). In case the error is not transient, the master processor is considered permanently faulty and the trailer processor takes over, while technicians test and repair the master offline. The main drawbacks of this proposal are the fact that Berger Code, despite detecting multiple-bit faults, is effective only for unidirectional faults in combinational circuits.

In [14] the approach is to reinforce the weak gate of future technologies by allowing several of them to repeat the computation, and having a decision device to decide the actual value of the output.

Our work concentrates in the proposal of new ways for designing operators that will be tolerant to multiple simultaneous upsets.

With the technology trends available today, it is clear that multiple upsets or electromagnetic noise will severely impact circuit behavior, much beyond the single upset hypothesis. This way, a design paradigm able to withstand multiple simultaneous upsets must be devised.

Basically, what we propose is to bring some analog design techniques into the digital domain, since in the analog world process variance and noise are everyday problems, and designers have already good solutions to that.

3. BEHAVIOR OF TMR IN THE PRESENCE OF SINGLE FAULTS

Central to the notion of multiple fault protection is the clever use of redundancy. That is also used by the TMR approach. However, in TMR, multiple faults affecting different modules may produce a wrong voting, and hence a wrong output.

In order to confirm this behavior, a small 4×4-bit array multiplier circuit has been implemented using the TMR approach, as depicted in Figure 1: (a) basic module of the array multiplier, (b) the array of sixteen basic modules and (c) three modules of the array multiplier connected to one voter to generate the correct product. The three multiplier modules are used to generate the product and the voter circuit tests for majority and defines the final output of the multiplier. Although one could use triple voters too, the effect of multiple faults could also happen in more than one voter, and hence a fault would be produced. For the sake of simplicity, all results are given for the single voter, but similar behavior can be observed with triple voting TMR.

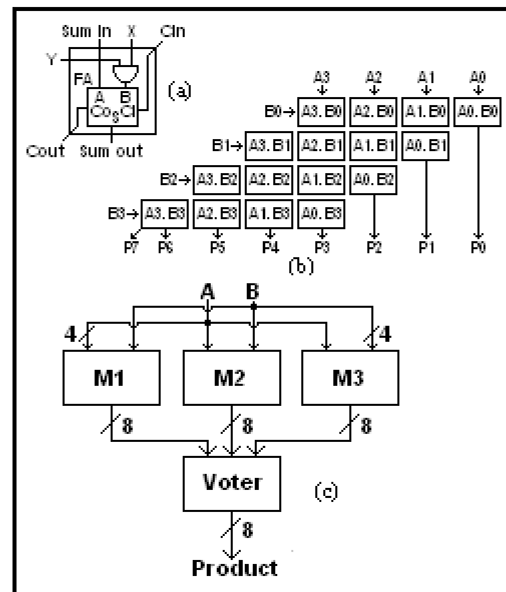


Figure 1. TMR 4x4-bit Array Multiplier

Using the circuit of Figure 1, we have simulated the injection of single stuck-at-0 and stuck-at-1 faults at every gate of the circuit, one at a time, and for each injected fault we checked the resulting output, to see if it was the correct product. This procedure has been repeated with all 256 possible input combinations (test vectors) for the circuit under test.

The simulation tool used was CACO-PS [3], a configurable cycle-accurate power simulator, which was extended to support single and multiple simultaneous stuck-at fault injection. The simulator

works as follows: the first step is to simulate the operation of the circuit and store the correct result. After that, for each possible stuck-at fault combination in the circuit, the simulation is repeated. Then, the output of each simulation is compared to the correct one. If any value differs, the fault was propagated to the output. All the process is repeated again for each combination of signals of the circuit.

The circuit under test was implemented using 320 gates (168 AND2, 48 OR2, 8 OR3 and 96 XOR2 gates), distributed as follows: 96 gates in each multiplier and 32 gates in the voter circuit. Thus, for each test vector, 640 faults have been injected.

The simulation has confirmed that single faults occurring in any of the three multiplier modules do not affect the output of the circuit. However, for each possible input combination, faults injected in some of the components of the voter have been propagated to the output, producing wrong results. The number of propagated faults varies, according to the input values to be multiplied, the specific gate affected, and the type of fault (sa-0 or sa-1), from 6 to 24. Assuming that the probability of any single fault affecting one gate is the same for every gate (1/320 for the circuit used in this simulation), this means that, for every possible input combination, the probability of the occurrence of a single fault being propagated to the output varies from 3/320 to 12/320, which is not neglectable, since for some classes of applications one cannot afford to have any wrong output and therefore cannot leave this matter subject to an unfavorable probabilistic behavior.

For larger circuits protected with the TMR technique, the probability is smaller than the one for the small array multiplier used in our study, since the area of the voter circuit represents a smaller fraction of the overall circuit. However, this probability is never equal to zero, and as the technology shrinks, the size of the block to be protected will reach small combinations of gates like in the presented example.

One alternative could be the use of three voters with a final additional circuit to choose the majority between the outputs produced by the voters. However, this circuit, itself, would be a voter and subject to the same weakness, therefore just adding area to the whole device and having the same handicap.

Therefore, it is mandatory to devise a technique to either make the voter circuit tolerant to faults, or to design an alternative circuit that is not dependent on the voting at the output.

Some experimental solutions were explored in previous works of the authors [9, 10, 11]. However, either the cost in terms of additional area was too high, or the same dead-lock mentioned above for the TMR approach has been found, therefore discouraging the adoption of those alternatives. The same problem was found in their works by other authors, such as in [17].

4. THE ANALOG VOTER

As mentioned before, in the analog world designers have been dealing with process variability and noise effects for a long time. A single event upset in a single gate can be modeled as the injection of extra charge in the gate capacitor, producing a voltage and a current of a certain duration. These can be seen as noise for an analog circuit. Hence, what one needs is a circuit that can support some noise at its output and still produce the right result. One such circuit is the analog comparator. Since it operates in the saturation mode of its transistors and thus, a DC current is always flowing into the circuit, the comparator itself will never suffer from SEU effects, and hence one has a fail-safe voter. This reasoning is corroborated by the modeling of influence of SEU effects as published in [12, 18, 19]. As long as the DC current is slightly larger than the one caused by the critical charge of the node of interest, there is no possibility of an SEU occurrence.

On the other hand, in order to use the comparator, one must transform the digital output of the TMR device into an analog output. This can be easily achieved in CMOS technology by transforming the voltage output of a TMR circuit into a current. Each current branch will be added at the comparator input, and the comparator will decide the side to switch based on the majority of currents. However, since one is dealing with CMOS technology, it is enough to put all TMR outputs shorted together, as it is shown in Figure 2. In case one gate input switches its value, at the output one will have a PMOS and an NMOS device conducting at the same time. This will produce a voltage way beyond the noise margin of a digital device, but easily decidable by the analog comparator proposed in this work.

The analog voter is implemented with a set of inverters whose outputs are short circuited and fed into an analog circuit that compares the values of the resulting voltage with a reference value. Whenever the comparator receives a voltage higher than the reference value, it generates a logic 1 at the output; otherwise, a logic 0 is generated.

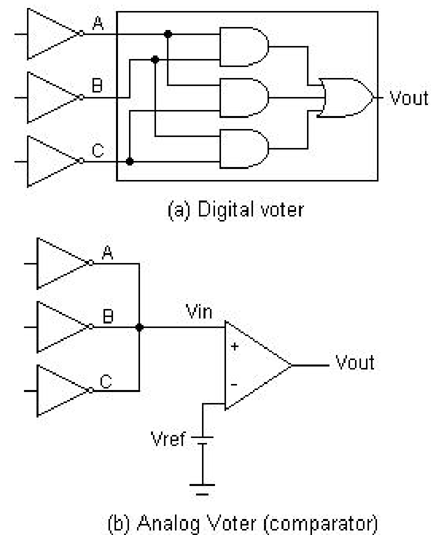


Figure 2. Digital Voter x Analog Voter

The architecture of the comparator is shown in Figure 3. It has a differential input pair (M1 and M2) with an active load (M3 and M4), a current source (M5) and a polarization circuit (M6 and M7). The dimensions of the transistors are: 5/0.35 (M1 and M2), 1/0.35 (M3 and M4), 2/0.35 (M5), 1/2 (M6) and 0.35/2.9 (M7).

The current sources shown in Figure 3 are used to model the injection of faults into the circuit during tests, and are not used during normal operation of the device.

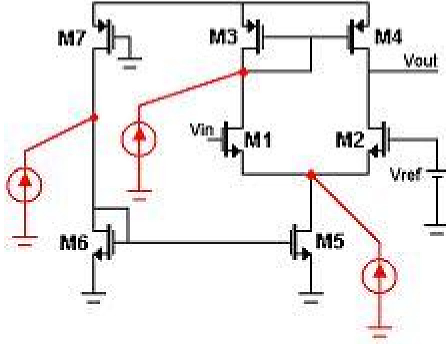


Figure 3. Minimum Area Comparator

Thus, the area of each comparator used in the simulations is:

$S_{m1} = S_{m2} = 11.75\mu m^2$

$S_{m3} = S_{m4} = 2.35\mu m^2$

$S_{m5} = 4.7\mu m^2$

$S_{m6} = 4\mu m^2$

$S_{m7} = 1.715\mu m^2$

$S_{comparator} = 38.615\mu m^2$

This area does not vary according to the number of input values for the voter, since all inputs are short circuited into a single wire, supplying V_{in} (see Figure 2). However, to calculate the total area of the analog voter, one must add the area of one inverter ($1.015\mu m^2$ in our example) for each input. The number of inputs varies according to the number of modules of the circuit being protected (*n* for *n*-MR), and the number of voters depends on the number of bits generated by each module.

In the present study, the multiplier being protected generates an 8-bit value; therefore, the number of analog voters will be eight, each one composed by one comparator ($38.615\mu m^2$) and *n* inverters ($n \times 1.015\mu m^2$). The voter circuit, therefore, will have an area equal to:

$$Svoter = 8 \times (38.615 + 1.015n)\ \mu m2 \qquad (1)$$

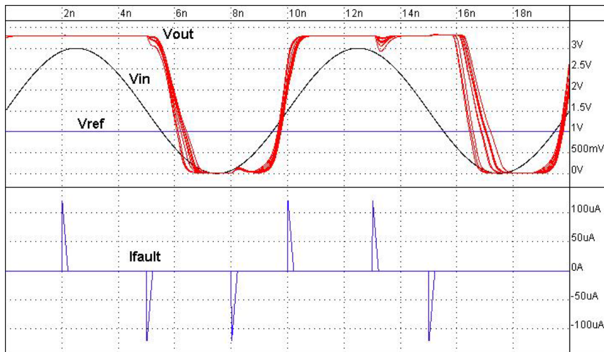


Figure 4. Effect of Injected Faults on the Output of the Comparator

The injection of faults into the analog voter has been electrically simulated, using the current sources shown in Figure 3. The results, shown in Figure 4, prove that such circuit is tolerant to transient faults.

The behavior of the comparator according to manufacturing defects due to process variability has also been simulated, and the results (see Figure 5) show that there is no impact of such defects on the operation of the comparator.

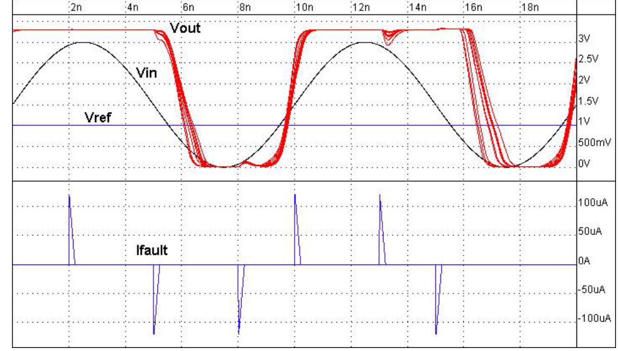


Figure 5. MonteCarlo Variation of t_{ox} and V_{th} at the Transistors of the Comparator

5. A RELIABLE TMR IMPLEMENTATION

The analog voter described in the previous section has been used to replace the conventional digital voter used in circuits protected by the TMR technique, and the new circuit has been submitted to the same process of fault injection simulation described in section 2.

The simulation showed that the resulting multiplier is completely tolerant to single transient faults, for every possible combination of inputs, producing always the correct output.

This enhanced TMR scheme does imply in some area increase, as shown in Table 1 (areas calculated for CMOS 0.35 technology). However, it must be noted now that the circuit is really immune to radiation effects, which was something not achieved before.

Table 1. Area Comparison (µm2)

	With Digital Voter	With Analog Voter
Multiplier Circuits	1,209.60	1,209.60
Voter Circuit	109.20	333.28
Total Area	1,318.80	1,542.88
Percent Area	100.00	116.99

6. TOLERANCE TO MULTIPLE SIMULTANEOUS FAULTS

The use of the analog voter proposed in section 3 eliminates the possibility of single transient faults affecting the results generated by circuits implemented using the TMR approach.

However, the community concerns about future technologies go beyond TMR, since the simultaneous occurrence of more than one transient fault is being considered as possible.

Therefore, solutions generally named n-MR that are currently used to withstand more than one fault will be necessary, in order to face this challenge. All known n-MR approaches suffer from the same drawback explained for TMR in section 2, since the voters are also digital circuits subject to the bad effects of simultaneous transient faults.

To cope with this, the analog voter has also been tested for larger number of inputs, and the simulations show that it remains tolerant to transient faults independently of the number of inputs. The only additional cost is one inverter per input, as shown in Figure 6, which represents a voter with five inputs.

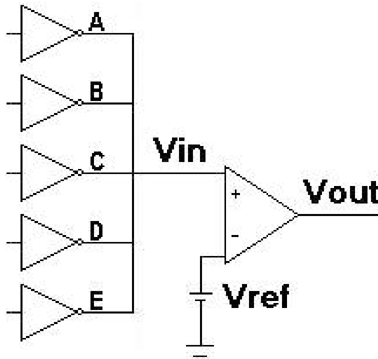


Figure 6. The Analog Voter with 5 Inputs

Using a 5-MR approach, and a different 4×4-bit multiplier architecture, the injection of single and double simultaneous faults has been tested to check its tolerance to such events.

The new multiplier is built with 16 AND2 gates that calculate the $a_i \times b_j$ products from the bits of the operands. Each column of 1-bit products is then added using the minimum possible amount of half adders and full adders, according to the number of summands, thus generating one partial sum for each of the seven columns.

The partial sums are left shifted according to their positions in the multiplication algorithm and then added, again using the minimum amount of half adders, full adders and OR2 gates (for the leftmost addition, which does not generate a carry).

Figure 7 shows the values of the partial sums, already shifted left before the final addition, when the factors are both equal to 15_{10} (1111_2). The values above the first horizontal line are those of the carry bits generated by the adders in the column(s) at the right.

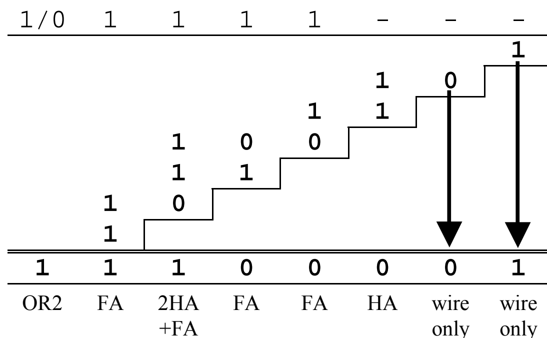


Figure 7. Addition of the Partial Sums

The value after the slash in the top of the first column represents the carry generated by the addition of the values in the third column, that is propagated to the first column, since the maximum possible value for the sum in the third column is 4_{10} (100_2).

The bottom line shows the type of circuit/component used to add the bits in the corresponding column. Note that, since the value of the sum in the first column is calculated just by adding the carry bits from the second and third columns, which are mutually exclusive, the most significant digit of the product can be calculated using just an OR2.

In terms of area, the new multiplier compares to the array multiplier as shown in Table 2

Table 2. Areas of the 4x4-bit Multipliers (μm^2) (per module)

Multiplier Type	Array		New	
	Qty	Area	Qty	Area
AND2	48	151.20	37	116.55
OR2	16	50.40	8	25.20
XOR2	32	201.60	21	132.30
Total		403.20		274.05

This same architecture can be used to reduce the area of the multiplier described in section 5.

With this new multiplier architecture, a sample circuit with five multiplier modules, five inverters to supply the inputs to each comparator in the analog voters and eight comparators was simulated. The total multiplier area, assuming the use of CMOS 0.35 technology, and using expression (1) to calculate the area of the voter, is:

$$5 \times 274.05 + 8 \times (38.615 + 1.015 \times 5) = 1{,}719.77 \text{ μm2}$$

The simulation showed that this 5-MR solution, combined with the 5-input analog voter, produces the correct output, for every possible input combination, even in the presence of two simultaneous transient faults.

The same reasoning can be generalized to build n-MR circuits (n odd), which will then withstand (n-1)/2 simultaneous fault. The decision about the value of n will depend on the risk × cost analysis that must be done before design.

The total area of any n-MR implementation with the analog voter will be n times the area of the circuit to be protected, plus n inverters (one at each input for the analog voter), plus the area of the comparator.

7. CONCLUSIONS AND FUTURE WORK

This paper has presented an analog voter that is tolerant to transient faults and therefore can be used to replace the digital voters in n-MR protection schemes which are able to withstand up to (n-1)/2 simultaneous faults in a circuit.

This study is part of a long-term project that aims to define new design alternatives to withstand simultaneous transient faults that will happen with circuits using future technologies, for all elements of a processor.

8. REFERENCES

[1] Anghel, L., Alexandrescu, D. and Nicolaidis, M., "Evaluation of soft error tolerance technique based on time and/or space redundancy", in *Proceedings of 13th Symposium on Integrated Circuits and Systems Design (ICSD 2000)*, pp. 237-242, IEEE, Manaus, Brazil, September 2000.

[2] Anghel, L. and Nicolaidis, M., "Cost Reduction and Evaluation of a Temporary Faults Detection Technique", in *Proceedings of Design, Automation and Test in Europe Conference (DATE 2000)*, pp. 591-598, ACM, Paris, France, March 2000.

[3] Beck Fº, A. C. S., Mattos, J. C. B., Wagner, F. R. and Carro, L., "CACO-PS: A General Purpose Cycle-Accurate Configurable Power-Simulator", in Proceedings of the 16th Brazilian Symposium on Integrated Circuits and Systems Design (SBCCI 2003), Sep. 2003.

[4] Constantinescu, C., "Trends and Challenges in VLSI Circuit Reliability", IEEE Micro, vol. 23, no. 4, pp. 14-19, IEEE Computer Society, New York-London, July/August 2003.

[5] Edenfeld, D.; Kahng, A.B.; Rodgers, M.; Zorian, Y., "2003 Technology Roadmap for Semiconductors", *IEEE Computer*, vol. 37, pp. 47-56, IEEE Computer Society, New York-London, January 2004.

[6] Gupta, S. K., Breuer, M. A. and Mak, T. M., "Defect and Error Tolerance in the Presence of Massive Numbers of Defects", IEEE Design & Test of Computers, vol. 21, issue 3, pp. 216-227, IEEE Computer Society, New York-London, May 2004.

[7] Johnson, B. W., Design and Analysis of Fault-Tolerant Digital Systems. Reading, MA: Addison-Wesley Publishing Company 1989.

[8] Lima, F., Carro, L. and Reis, R., "Techniques for Reconfigurable Logic Applications: Designing Fault Tolerant Systems into SRAM-based FPGAs", in Proceedings of the International Design Automation Conference, DAC 2003, pp. 650-655, ACM, New York, 2003.

[9] Lisbôa, C. and Carro, L., "An Intrinsically Robust Technique for Fault Tolerance under Multiple Upsets ", in *Proceedings of the 10th IEEE International Online Test Symposium - IOLTS 2004*, pp. 180, IEEE Computer Society, New York, July 2004.

[10] Lisbôa, C. and Carro, L., "Arithmetic Operators Robust to Multiple Simultaneous Upsets", in Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems - DFT 2004, pp. 289-297, ISBN 0-7695-2241-6. IEEE Computer Society, New York, October 2004.

[11] Lisbôa, C. and Carro, L., "Highly Reliable Arithmetic Multipliers for Future Technologies", in Proceedings of the International Workshop on Dependable Embedded Systems - WDES 2004 - in conjunction with the 23rd International Symposium on Reliable Distributed Systems - SRDS 2004, pp. 13-18. Edited by Becker, L. B. and Kaiser, J., Florianópolis, October 17, 2004.

[12] Messenger, G. C., "A Summary Review of Displacement Damage from High Energy Radiation in Silicon Semiconductors and Semiconductor Devices", IEEE Transactions on Nuclear Science, vol. 39, no. 3, pp. 468-473, IEEE Computer Society, New York-London, June 1992.

[13] Pflanz, M. and Vierhaus, H. T., "Online Check and Recovery Techniques for Dependable Embedded Processors", IEEE Micro, vol. 21, number 5, pp. 24-40, IEEE Computer Society, New York-London, September-October 2001.

[14] Schmid, A. and Leblebici, Y., "A Highly Fault Tolerant PLA Architecture for Failure-Prone Nanometer CMOS and Novel Quantum Device Technologies", in Proceedings of the 19th IEEE International Symposium on Defect and Fault-Tolerance – DFT 2004, pp. 39-47, ISBN 0-7695-2241-6. IEEE Computer Society, New York, October 2004.

[15] Semiconductor Industry Association. International Technology Roadmap for Semiconductors - ITRS 2003, last access November 12, 2004. http://public .itrs.net/Files/2003ITRS/Home2003.htm.

[16] M. Rebaudengo, M. Sonza Reorda, M. Violante, B. Nicolescu, R. Velazco, "Coping With SEUs/SETs in Microprocessors by means of Low-Cost Solutions: A Comparative Study", *IEEE Transactions on Nuclear Science*, vol. 49, no. 3, pp. 1491-1495, IEEE Computer Society, New York-London, June 2002.

[17] Townsend, W. J., Abraham, J. A. and Lala, P. K., "Online Error Detecting Constant Delay Adder", in Proceedings of the 9th IEEE International Online Testing Symposium, pp. 17-22, Kos Island, Greece, July 7-9, 2003.

[18] Wirth, G., Vieira, M. and Kastensmidt, F. L., "Computer Efficient Modeling of SRAM Cells Sensitivity to SEU", in Proceedings of the 6th IEEE Latin America Test Workshop, pp. 51-56, Salvador, Bahia, Brazil, March 30-April 2nd, 2005.

[19] Yang, F. L. and Saleh, R. A., "Simulation and Analysis of Transient Faults in Digital Circuits", IEEE Journal of Solid-State Circuits, vol. 27, no. 3, pp. 258-264, IEEE Computer Society, New York-London, March 1992.

Sigma-Delta Modulators on the design of reliable digital circuits

E. Schüler, L. Carro

Universidade Federal do Rio Grande do Sul
Departamento de Engenharia Elétrica

eschuler@eletro.ufrgs.br, carro@eletro.ufrgs.br

Abstract

As the transistor gate length goes straightforward to the sub-micron dimension, the possibilities of occurrence of external interferences in these devices also increase. Moreover, the process variability will further degrade this scenario. The direct effect of such external and/or intrinsic interferences is, in many cases, the total mismatch between the desired answer of the system and the achieved answer. This way, new techniques must be studied in order to guarantee the correct operation of these systems. This work presents the use of a totally digital sigma-delta modulator that is used to develop a FIR filter with much better results than if a common code-modulated representation was used. Simulations results show that, even with the insertion of a large amount of faults, one can still obtain a non-faulty behavior in the SNR of the application.

1. Introduction

The MOSFET gate length is expected to be 22nm in the year of 2008. This way, more than one billion of transistors will be able to be integrated in a single die [1]. With this number of devices, there is a clear possibility of creating systems with more and more resources. However, with such a small gate length, also the possibility of occurrence of some problems that were not expected to occur in systems a few years ago arises. One of the most important problems is the single event upset or simply SEU [2, 3].

As the size of the transistor channel decreases, also the number of electrons (or holes) in the channel decreases. With a lower number of carriers passing through the transistor channel, although the possibility of these carriers being hit by an external particle (e.g. alpha or neutron particles) decreases, if such event occurs, the effect caused will be much higher than if there was a higher number of carriers. Such interference may cause a SEU, which can cause, for example, a single bit flip that may damage the system or subsystem behavior in such way that the achieved answer can differ a lot from the expected one.

Also, as the frequency of operation of these circuits increases, even a very short time transient pulse caused by a radiation and/or electromagnetic noise may affect the output of combinational circuits, as well as the values stored in memory elements.

These phenomena may occur in digital, analog, and even optical components, or may have effects in surrounding circuitry. Such events, causing transient random faults, forces one to search for new design approaches to reduce the effects of these faults, which can occur even in fully tested and approved circuits.

This work presents a technique that consists of modulating the signal to be processed in the sigma-delta domain, in such a way that the redundancy presented in the modulated signal can reduce the effects of the faults occurred during the system operation, thus allowing one to build more reliable circuits, regarding SEUs. In a practical application like a digital filter, even with several simultaneous faults the impact on the total SNR was close to zero.

This paper is divided as it follows: section II makes a review of some techniques used to improve the performance of systems subject to faults. In section III, it is shown that signals modulated in sigma-delta domain are in fact more tolerant to faults, due to the amount of redundancy present in its representation. In order to show that a small deviation from the original value does not interfere so much in the final result of some applications, section IV shows some results regarding the implementation of a FIR filter using the proposed technique, and shows that, when compared with the same filter implemented using code-modulated signals, the obtained results are much more consistent. Finally, section V presents the conclusions and some future work that are in process of development.

This work has been partially supported by SRC Grant nr. 2004-RJ-1274G, Capes and CNPq.

2. Related works

Many techniques have been developed to cope with the mentioned SEU faults. In [4], two low-cost solutions to cope with SEU are compared: the error-detection capabilities of a hardware-implemented solution based on parity code and software-implemented solution based on source-level code modification.

The implementation of a new soft error tolerance technique based on time redundancy and/or space redundancy is presented in [5]. Also, the use of triple modular redundancy (TMR) is discussed and proposed in [6] and [7], which also proposes the use of double modular redundancy.

These techniques were developed and presented good results when single soft errors were presented. Recently, it was presented in [8] a technique that is able to cope with multiple soft errors. Such technique is based on the use of larger redundant words (bit-streams) to represent the signals to be processed, and is supposed to be used in systems that work with the concept of error tolerance proposed in [9].

The final goal of error tolerance consists basically in dealing with systems that are error tolerant regarding their final application. This means that systems that, even under the presence of multiples faults, that could cause internal and/or external errors still produce acceptable results. Thus, using this concept, for some applications, a certain degree of error is acceptable.

This work is based on the same idea presented in [8], however, the generation of the bit-streams is developed through the use of a sigma-delta modulator.

When compared with the bit-stream process generation presented in [8], which is based on the use of the probability associated with some digital quantity, the use of sigma-delta modulation can lead to a better representation [10]. As a result, the reduction in the necessary length of the bit-stream that represents the quantity can be achieved, without loosing signal resolution.

The results presented in this work make the use of a digital first order sigma-delta modulator that converts an eight bits input signal to its sigma-delta domain representation, for further computation.

3. Fault Tolerance in the sigma-delta domain

Sigma-delta-modulated signals, although requiring over sampling, have a great advantage if compared with code-modulated signals, for what regards fault tolerance: since sigma-delta-modulated signals are oversampled, each bit-stream that represents an amplitude value carries an amount of redundancy that, even with some faults inserted, can still represent the mean value of the original signal. Table 1 shows how an eight bits signal modulated with a digital first order sigma-delta modulator showed in figure 1 [11], can still maintain a value that is very close to the original one after the insertion of a different number of faults, in different bits of the subtractor and integrator of the sigma-delta modulator. All results showed are from simulation data.

The first column of table 1 shows how many bits in the adder and in the subtractor of the sigma-delta modulator are randomly flipped. Since the signal to be modulated will be over sampled OSR (over sampling rate) times through the modulator, the fault will occur only in a number of cycles, which is represented in column 2 of table 1. The third column shows the original signal that will be sigma-delta modulated, while the last column presents the maximum deviation achieved from the faulty modulated signal, after 20 modulations. The OSR used is 128.

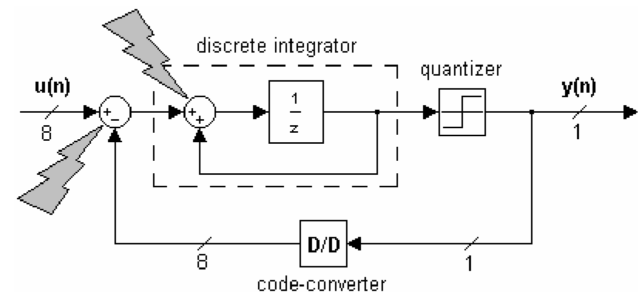


Figure 1: Digital first order sigma-delta modulator, according [11], showing where faults were injected.

Table 1: Maximum deviation of sigma-delta modulated signals with faults inserted.

Number of randomly flipped-bits	Number of cycles that fault occurs	Original signal	Maximum deviation*
0	0	20	0.0781
1	2	20	1.2500
	5		2.0313
	7		3.5938
2	2	20	2.0313
	5		3.5938
	7		4.3750
3	2	20	1.6406
	5		3.9844
	7		5.9375
4	2	20	2.4219
	5		3.5938
	7		4.7656

* Maximum deviation from the original value, after modulating the signal 20 times.

Note that, although the value of the sigma-delta-modulated signal is not exactly the same of the original value, in applications that tolerate small error percentages or in systems that use cascading calculations, like digital filters, this approach can still be used with an acceptable interference in the final result, as it will be shown in section IV.

In the following section, a digital FIR filter using sigma-delta-modulated signal is developed, in order to evaluate the advantages of using such approach.

4. Sigma-delta-modulated *versus* code-modulated signals in fault tolerant filter development

This section presents some previous results regarding the use of sigma-delta-modulated signals, with some faults inserted, in more complexes applications, as is the case of convolution that is the basis of filters operation. A FIR filter is used as an example to show that, when compared with filters that use code-modulated signals, those that use signals in the sigma-delta domain, present more robustness regarding fault insertion.

Some techniques to filter signals in the sigma-delta domain have already been proposed [12, 13]. However, just in order to evaluate the fault tolerance of the process using two different kinds of signal modulation, the approach showed in figure 2 was used, where the filter is a common 16 taps lowpass FIR. Figure 2(a) shows a typical way to filter signals that are code-modulated with a certain number of bits (8 bits in this case), while figure 2(b) shows how a signal in the sigma-delta domain can be filtered. In the second case, some modifications in the filter structure must be made, but the working principle is the same. For example, when multiplying each value of the bit-stream by one of the coefficient of the filter, one just have to choose between to add or not to add the coefficient value, since the bit-stream is a sequence of zeros and ones.

Although the sigma-delta modulator itself could make the analog-to-digital conversion, it was used here just to translate the signal to the sigma-delta domain.

Comparisons between these two ways of filtering were developed and are presented in figure 3.

The input signal formed by a single 20KHz tone and an amount of noise was sampled and filtered in three different ways: using the samples themselves and a filter without any fault inserted; using a code modulation of the samples with the insertion of one fault through the inversion of a single random bit, in five computed values during the filtering process; and finally, using a sigma-delta modulation of the samples with the insertion of faults.

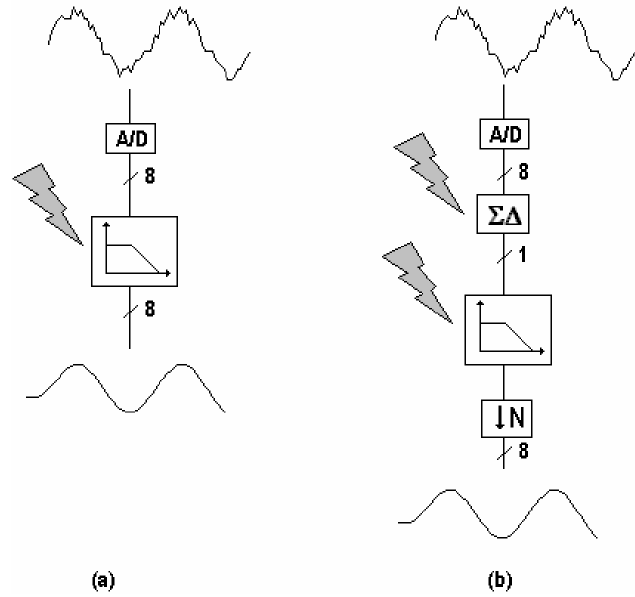


Figure 2: Filtering signals modulated in 8 bits (a) and modulated in the sigma-delta domain (b).

Notice that each sampled point of the input signal is modulated with an OSR. So, the OSR corresponds, in this case, to the over sampling rate used to modulate each sampled point.

The faults were inserted in two parts of the filtering process: first in the generation of the bit-stream through the variation of one random bit, during 2% of OSR, that is correspondent to 5 cycles, with a OSR of 128; in the filter structure, each time a bit-stream was multiplied by a coefficient, one fault was inserted through the variation of a random value of the product, and for each resulting value of the addition of these products, one fault was inserted through the variation of another value of the addition result.

Figure 3(a) shows the input signal in the frequency domain, while a comparison between this signal after being filtered by a fault-free filter and using the faulty code-modulated approach is showed in figure 3(b), and a comparison between the faulty-free filtered signal and the faulty sigma-delta approach is showed in figure 3(c).

A time domain representation showing a comparison between the fault-free filtered signal, the faulty code-modulated filtered signal and the faulty sigma-delta-modulated filtered signal is showed in figure 3(d).

As one can see in figure 3, although the number of faults inserted in the sigma-delta signals is much larger than the number of faults inserted in the coded signals, the results obtained with the coded case are much worse than those obtained with the sigma-delta.

5. Conclusions

This work presented the use of sigma-delta modulators in the generation of fault-tolerant signals, intended to be used in mean-based systems that make intensive use of arithmetic operations.

It was shown that systems that use such technique can handle multiple faults, while still producing results that are much better than those produced with code-modulated signals.

In order to show that a small deviation from the original value does not cause a great disturb in the final result, a FIR filter was developed. Signals modulated in the sigma-delta domain and signals coded with 8 bits were filtered with this filter, and faults were injected, showing that, when using the sigma-delta-modulated signals, the tolerance to faults is much higher, even for a larger number of injected faults.

Although the proposed approach leads to the necessity of high frequency systems, since the sigma-delta modulator needs high over sampling rates to achieve a good resolution, the high tolerance to multiple faults can make its use advantageous. However, as mentioned, for systems that must multiply the input bit-stream by a certain value, since the bit-stream if formed by a sequence of zeros and ones, the multiplication can be substituted by a simple addition, what can contribute for the reduction of the area taken for processing.

Future works include the investigation on the effects of the dynamic reduction of the over sampled ratio, in order

to adjust the amount of tolerance to errors with the rate of errors present in the system, since particles do not reach the system at constant intervals. Also, others parameters like power dissipation, processing-time and total area must be evaluated.

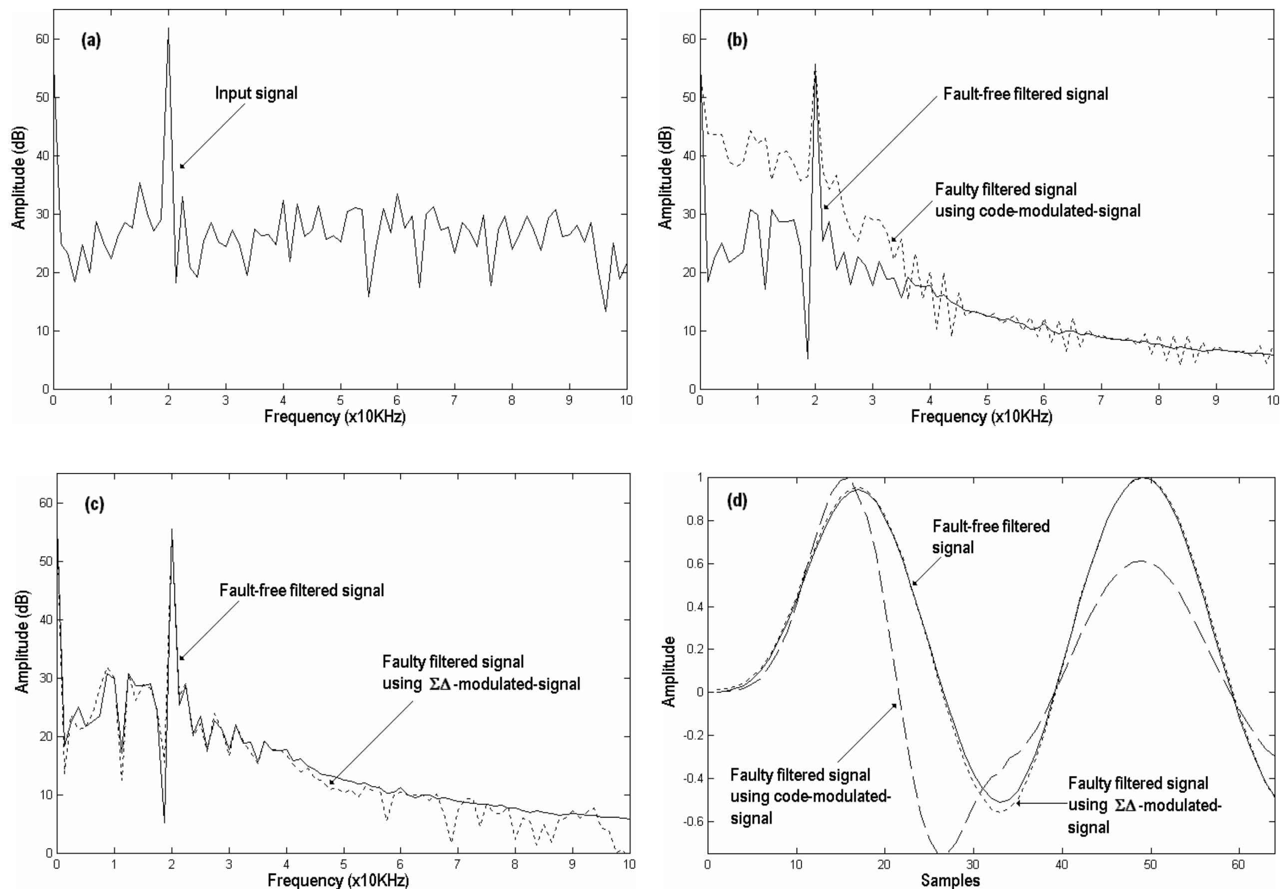


Figure 3: Simulation results showing frequency and time domain for a 20KHz signal filtered using sigma-delta and code-modulated signals.

6. References

[1] Semiconductor Industry Association. International Technology Roadmap for Semiconductors - ITRS 2004 last access Jan. 26, 2005. http://www.itrs.net/Common/2004Update/2004Update.ht

[2] Yang, F.L.; Saleh, R.A. "Simulation and Analysis of Transient Faults in Digital Circuits". IEEE Journal of Solid-State Circuits, Volume: 27, Issue: 3, March 1992. Pages: 258 – 264.

[3] Messenger, G.C. "A summary review of displacement damage from high-energy radiation in silicon semiconductors and semiconductor devices". IEEE Transactions on Nuclear Science, Volume: 39, Issue: 3, June 1992. Pages: 468 – 473.

[4] M. Rebaudengo, M. Sonza Reorda, M. Violante, B. Nicolescu, R. Velazco. "Coping with SEUs/SETs in Microprocessors by means of Low-Cost Solutions: A Comparative Study". IEEE Transactions on Nuclear Science, Volume: 49, Number: 3, June 2002. Pages: 1491 – 1495.

[5] Anghel, L., Alexandrescu, D. and Nicolaidis, M. "Evaluation of soft error tolerance technique based on time and/or space redundancy". Proceedings of 13rd IEEE Symposium on Integrated Circuits and Systems Design (ICSD 2000),, Manaus, Brazil, September 2000. Pages: 237 – 242.

[6] Yoon, J.; Kim, H. "Time-redundant recovery policy of TMR failures using rollback and roll-forward methods". IEEE Proceedings Computers and Digital Techniques, Volume: 147, Issue: 2, March 2000. Pages: 124 – 132.

[7] Lima, F., Carro, L. and Reis, R. "Techniques for Reconfigurable Logic Applications: Designing Fault Tolerant Systems into SRAM-based FPGAs". Proceedings of the International Design Automation Conference, DAC 2003, ACM, New York, 2003. Pages: 650 – 655.

[8] Lisbôa, C. and Carro, L. "Arithmetic Operators Robust to Multiple Simultaneous Upsets". Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems - DFT 2004, pp. 289-297, ISBN0-7695-2241-6. IEEE Computer Society, New York, October 2004.

[9] Gupta, S. K., Breuer, M. A. and Mak, T. M. "Defect and Error Tolerance in the Presence of Massive Numbers of Defects". IEEE Design & Test of Computers, Volume: 21, Issue: 3, May 2004. Pages: 216 – 227.

[10] Norsworthy, S.R., Schreier, R., Temes, G. C. "Delta-sigma data converters: theory, design and simulation". New York: IEEE Press, 1997. 476p.

[11] Robert, G. W., Lu, A. K., "Analog Signal Generation for Built-In-Self-Test of Mixed-Signal Integrated Circuits". Kluwer Academic Publishers, 1995. 122p.

[12] Wong, P.W.; Gray, R.M., "FIR filters with sigma-delta modulation encoding". IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume: 38, Issue: 6, June 1990. Pages: 979 – 990.

[13] Thompson, A.C.; O'Shea, P.; Hussain, Z.M.; Steele, B.R., "Efficient single-bit ternary digital filtering using sigma-delta modulator". IEEE Signal Processing Letters, Volume: 11, Issue: 2, Feb. 2004. Pages: 164 – 166.

Reliable digital circuits design using analog components

Erik Schüler, Luigi Carro
{eschuler, carro}@eletro.ufrgs.br
Departamento de Eng. Elétrica
Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil

Abstract

With the reduction of the MOS transistor length, the occurrence of external interferences due to electromagnetic noise, and the increase in the number of problems despite of process variability, leads to the research of new techniques that can make digital circuits less susceptible to these external/internal factors. In this paper, it is proposed the use of an analog technique, which is more fault-tolerant than its digital counterpart. Some results towards the development of an analog voter to be used in a TMR (Triple Modular Redundancy) approach are presented. The insertion of faults that model external interferences are electrically simulated, and the variability of the transistors parameters used in the voter are evaluated through Monte Carlo variations, showing that the proposed technique can be used to guarantee a higher fault tolerance.

1. Introduction

As process technology is scaled down, the variability that occurs in the layout fabrication becomes a critical point. With smaller transistors and, consequently, with a much higher number of such devices, the need for circuits that are process variation tolerant increases. Actual deep sub micron circuits are affected not only by spot defects, but also by process variations, which can have both global and local effects on an integrated circuit. For example, the interconnect resistance increasing radial gradient across the wafer is considered a global variation. On the other hand, threshold voltage mismatch can be considered a local variation because of its random nature. Faults arising from process variability result in a correct output of the circuit, but this circuit will not meet performance specifications [1, 2].

Another problem occurs as the size of the transistor channel decreases. With a lower number of electrons (or holes) passing through the transistor channel, although the possibility of these carriers being hit by an external particle (e.g. alpha or neutron particles) decreases, if such event occurs, the effect caused will be much higher than if there was a higher number of carriers. Such interference may cause a Single Event Upset, or SEU [3, 4], which can cause, for example, a single bit flip that may damage the system or subsystem behavior in such way that the achieved response can differ a lot from the expected one.

Many techniques have been developed to cope with transitory faults due to external influences and others have been proposed to obtain better results regarding process variations, and some of these techniques are briefly reviewed in section II. This paper proposes the use of analog techniques in some digital circuits, in order to reduce the influence either of external and internal factors. The example used in this work is a TMR (Triple Modular Redundancy) approach, where the digital voter, which can suffer the same faults as the digital circuits to be voted, is substituted by an analog voter, which is supposed to be more fault-tolerant.

The paper is divided as it follows: section II presents some techniques used to cope with transitory faults and some to cope with process variations. In section III, a brief review about the TMR technique is presented, and section IV presents the proposition of this work. Finally, section V shows the conclusions and future works.

2. Previous works

Many techniques have been developed to cope with transitory faults. In [5], two low-cost solutions to cope with SEU are compared: the error-detection capabilities of a hardware-implemented solution based on parity code and software-implemented solution based on source-level code modification. The implementation of a new soft error tolerance technique based on time redundancy and/or space redundancy is presented in [6]. Also, the use of triple modular redundancy (TMR) is discussed and proposed in [7] and [8], which also proposes the use of double modular redundancy. These techniques were developed for single fault occurrence, and presented good results when this was the case, but cannot cope with multiple simultaneous errors. Recently, [9] presented a technique that is able to cope with multiple soft errors. Such technique is based on the use of larger redundant words (bit streams) to represent the signals to be processed, and is supposed to be used in systems that work with the concept of error tolerance.

The problems of process variations have been exhaustively studied, and many techniques have been presented. In [10], an analytical technique is developed to obtain power optimum design of switched-capacitor integrators with process variations consideration. The technique provides the worst and best case estimates and is demonstrated by providing performance variations of optimized integrators in a 0.6um CMOS process. A process variation compensating technique

for dynamic circuits for sub-90nm technologies is described in [11]. A keeper whose effective strength is optimally programmable based on die leakage is compared with conventional static keeper, obtaining betters results. A 0.35um CMOS inverter based comparator design is presented in [12]. The comparator has inherently process variation compensated structure that consists of one additional inverter and four triode region operating transistors, which makes the threshold voltage of the comparator independent of the process variation. The comparator is designed for FSK demodulation applications. Both works show that in future technologies, extra care will have to be taken for the design of circuits that nowadays are taken for granted.

3. Triple Modular Redundancy (TMR)

Fault tolerant techniques rely on some kind of hardware redundancy. These redundant elements are used to improve fault tolerance. Examples of the use of redundancy are the TMR (triple modular redundancy) and the NMR (n modular redundancy). All the elements execute the same work and the final result is defined through voting. The triple modular redundancy is the most known technique used to fault tolerance.

In the TMR, fault tolerance in a hardware component is improved through the triplication of this component and further vote among the outputs of these components to determine the correct result, as demonstrated in figure 1. The vote can be done by counting the number of correct outputs or by selecting the mean value [13, 14].

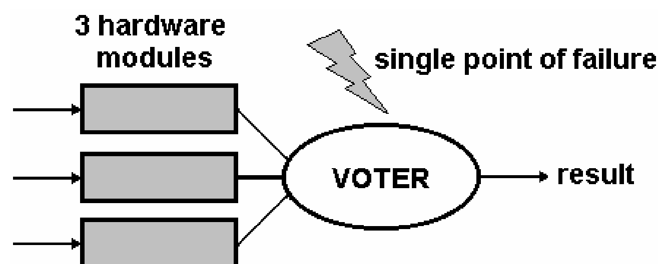


Figure 1. The TMR approach: the vote can be done by counting the number of correct outputs or by selecting the mean value.

Is interesting to note that the voter does not determine which module suffered the fault, but only if the fault occurred or not. Although simple, the voter is the critical point regarding fault-tolerance in the TMR approach. If the voter presents low reliability, the whole system will be fragile, as much as the voter itself.

Some solutions to increase the fault-tolerance of the voter include making voters with high reliability components, triplicating the voter itself and developing the vote through software. In the next section, we propose the use of an analog voter, which will accuse the presence of faults by counting the number of correct outputs.

4. Analog voter

As mentioned, the voter is a critical point in the TMR approach, since if a fault occurs in the voter, the whole system will fail. Figure 2 shows how a simple digital voter can be developed to determine if a fault has occurred in the TMR system. The response of the voter must be according to the number of logic values ones presented in the output of triple hardware modules, that is, if two or three outputs (A, B, C) are one, the voter output (S) will be one too.

However, if a transitory fault occurs in the voter during the system operation, even if the fault has a very short duration, it can be detected by the system that is monitoring the voter, what can lead to a malfunction of the system.

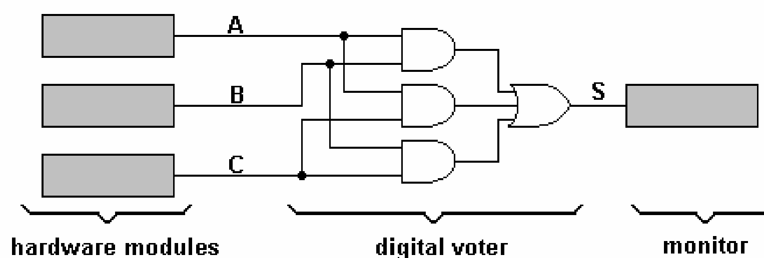


Figure 2. Example of a digital voter.

As mentioned, the use of a more robust voter could improve the fault-tolerance of the system. Our proposition is the use of an analog voter, which is basically an analog comparator, in which one of the inputs is connected to the hardware

modules through simple inverters whose outputs are short-circuited, and the other input is connected to a reference voltage, as presented in figure 3(a). According to the value of the inverters outputs (A, B, C), the resulting voltage in the short-circuit point (X) varies, allowing the comparison with the reference voltage, as seen in figure 3(b). This figure shows the simulation results when the comparators signals have different input values. As one can see, the voter output (S) has a high value just when two or more input signals are high, as a TMR approach is supposed to work. All results are from SPICE simulation data, using a 0.35um CMOS technology.

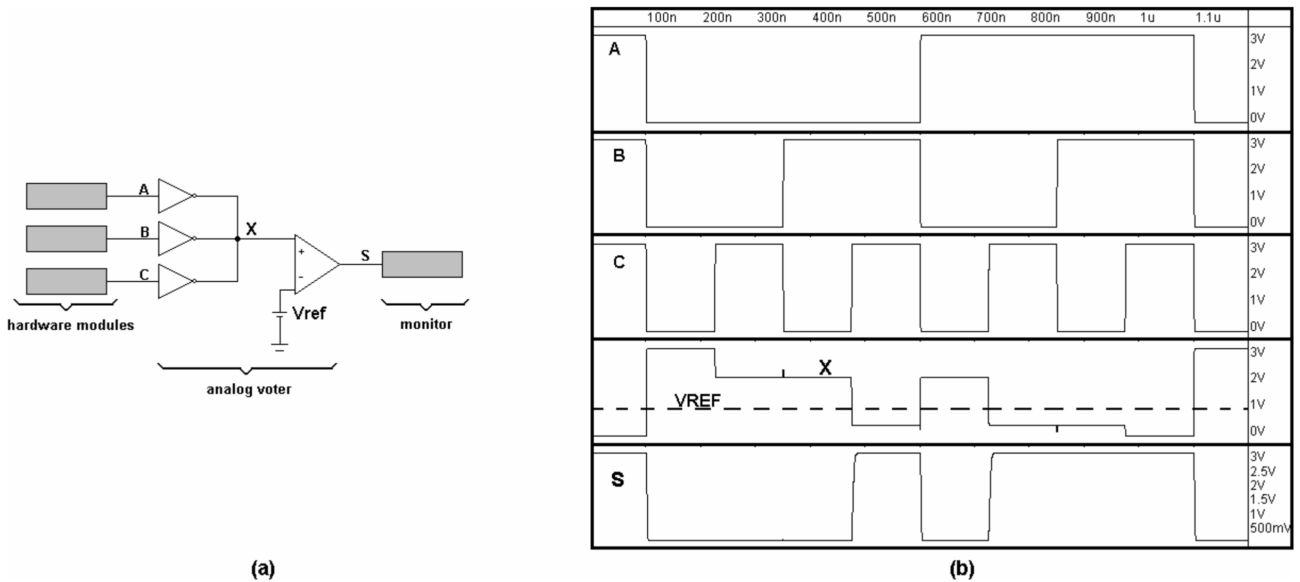


Figure 3. (a) Proposed use of an analog comparator and short-circuit inverters as a TMR voter and (b) simulation showing the output signals A, B, C, the resultant voltage at point X, the reference voltage and the voter output S.

According to [3], the injection of a transient fault caused, for example, by an external particle that hits the circuit, can be modeled by a current source. For an alfa-particle, for example, the current of the source can be modeled by a double exponential, whose charge can range from 1 to 5 pC [3].

Figure 4(a) shows an example of a minimum area comparator that was used in the simulations. This consists of an input differential pair with active load, a current source and a bias circuit. Faults were injected in three distinct points of the circuit, represented by the current sources. Figure 4(b) shows a simulation of the TMR system, where many faults are injected in the circuit.

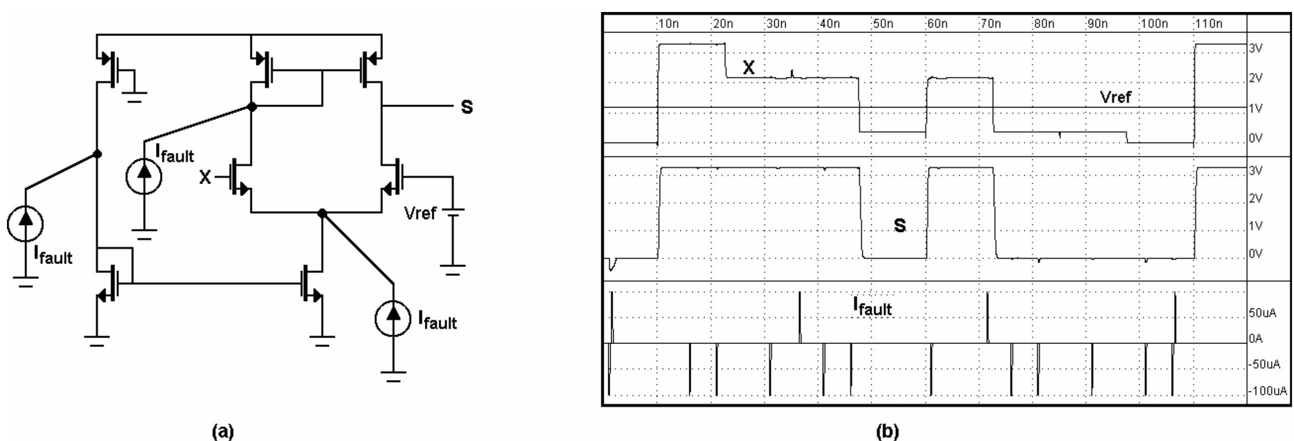


Figure 4. (a) Minimum area comparator with current sources modeling fault-injection and (b) simulation showing fault-tolerance of the system.

As one can see in figure 4(b), the faults do not propagate to the output of comparator, guarantying a high fault-tolerance to the voter.

Beyond transitory fault tolerance, it is also expected that the voter have some tolerance regarding process variation. Although better improvements can be done in the proposed comparator described in figure 4(a) to guarantee such

tolerance, the comparator herein showed presents a good response when some of its parameters are varied. The total area of the comparator is 38.615um^2.

As seen in figure 5, where a Monte Carlo simulation of the system is evaluated, even with a variation of 30% in both the oxide thickness (tox) and the threshold voltage (Vth) of the comparator transistors, the comparator still sustains a good fault tolerance. The input of the comparator is a 100MHz, 3Vpp sine wave.

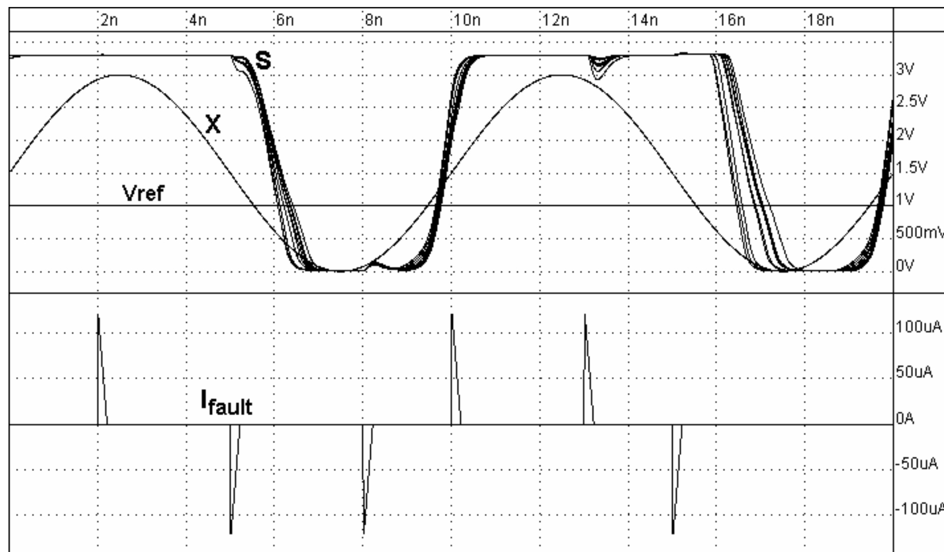


Figure 5. Oxide thickness and threshold voltage Monte Carlo analysis of the comparator transistors.

5. Conclusions

The use of an analog comparator to the development of the voter of a triple modular redundancy system is presented. The idea of using an analog element in a digital circuit to improve the fault-tolerance of the system is described and some simulations results are presented. As showed, even with multiple faults injected in the system, it remains stable, without propagating the faults to the output of the system.

Also, the tolerance regarding process variation is demonstrated, showing that the proposed technique is robust against some parameters variation. Better results can be obtained though the use of some techniques to improve the robustness of the comparator these variations.

With such approach, the damages due to the occurrence of single event upset (SEU) and some problems related to process variation are reduced, allowing the use of TMR with more security. The proposed idea can be easily extended to an NMR approach, just through the insertion of more short-circuited inverters in the input of the comparator.

Future works include the use of others analog techniques together with digital circuits in order to improve fault-tolerance, without incurring in a high area and power overhead. Also, the evaluation of power dissipation and area overhead of the presented technique in different technologies must be done.

References

[1] Natarajan, S.; Breuer, M.A.; Gupta, S.K. “Process variations and their impact on circuit operation”. IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. 2-4 Nov. 1998. Pages: 73 – 81.
[2] Malavasi, E.; Zanella, S.; Uschersohn, J.; Misheloff, M.; Guardiani, C. “Impact analysis of process variability on digital circuits with performance limited yield”. 6th IEEE International Workshop on Statistical Methodology. 10 June 2001. Pages: 60 – 63.
[3] Yang, F.L.; Saleh, R.A. “Simulation and Analysis of Transient Faults in Digital Circuits”. IEEE Journal of Solid-State Circuits, Volume: 27, Issue: 3, March 1992. Pages: 258 – 264.
[4] Messenger, G.C. “A summary review of displacement damage from high-energy radiation in silicon semiconductors and semiconductor devices”. IEEE Transactions on Nuclear Science, Volume: 39, Issue: 3, June 1992. Pages: 468 – 473.
[5] M. Rebaudengo, M. Sonza Reorda, M. Violante, B. Nicolescu, R. Velazco. “Coping with SEUs/SETs in Microprocessors by means of Low-Cost Solutions: A Comparative Study”. IEEE Transactions on Nuclear Science, Volume: 49, Number: 3, June 2002. Pages: 1491 – 1495.

[6] Anghel, L., Alexandrescu, D. and Nicolaidis, M. “Evaluation of soft error tolerance technique based on time and/or space redundancy”. Proceedings of 13rd IEEE Symposium on Integrated Circuits and Systems Design (ICSD 2000), Manaus, Brazil, September 2000. Pages: 237 – 242.

[7] Yoon, J.; Kim, H. “Time-redundant recovery policy of TMR failures using rollback and roll-forward methods”. IEEE Proceedings Computers and Digital Techniques, Volume: 147, Issue: 2, March 2000. Pages: 124 – 132.

[8] Lima, F., Carro, L. and Reis, R. “Techniques for Reconfigurable Logic Applications: Designing Fault Tolerant Systems into SRAM-based FPGAs”. Proceedings of the International Design Automation Conference, DAC 2003, ACM, New York, 2003. Pages: 650 – 655.

[9] Lisbôa, C. and Carro, L. “Arithmetic Operators Robust to Multiple Simultaneous Upsets”. Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems - DFT 2004, pp. 289-297, ISBN0-7695-2241-6. IEEE Computer Society, New York, October 2004.

[10] Naiknaware, R.; Fiez, T. “Switched-capacitor integrator design optimizing for power and process variations”. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems. Volume: 2, 30 May-2 June 1999. Pages: 278 – 281.

[11] Kim, C.H.; Roy, K.; Hsu, S.; Alvandpour, A.; Krishnamurthy, R.K.; Borkar, S. “A process variation compensating technique for sub-90 nm dynamic circuits”. Symposium on VLSI Circuits, 2003. Digest of Technical Papers. 12-14 June 2003. Pages: 205 – 206.

[12] Hong-Sun Kim; Seoung-Jae Yoo; Ismail, M.; Olsson, H. “A process variation compensated comparator for FSK demodulators”. Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, 2000. Volume: 2, 8-11 Aug. 2000. Pages: 940 – 942.

[13] Gaitanis, N. “The design of totally self-checking TMR fault-tolerant systems”. IEEE Transactions on Computers. Volume: 37, Issue: 11, Nov. 1988. Pages: 1450 – 1454.

[14] Nanduri, S.R.; El Hakeem, A.K.; Al-Khalili, A.J. “Fault analysis of a TMR system using multiple valued logic”. Ninth Annual International Phoenix Conference on Computers and Communications, 1990. Conference Proceedings. 21-23 March 1990. Pages: 23 – 29.

Increasing Fault Tolerance To Multiple Upsets Using Digital Sigma-Delta Modulators

Erik Schüler, Luigi Carro
Universidade Federal do Rio Grande do Sul, Depto. de Eng. Elétrica/PPGEE, Porto Alegre, Brasil
{eschuler,carro}@eletro.ufrgs.br

Abstract

As the transistor gate length goes straightforward to the sub-micron dimension, there is an increased possibility of occurrence of external interferences in these devices. The direct effect of such external and/or intrinsic interferences is, in many cases, the total mismatch between the desired answer of the system and the obtained response. So, new techniques must be studied in order to guarantee the correct operation of these systems, under multiple simultaneous faults. This work presents the use of a totally digital sigma-delta modulator that is used to develop arithmetic operations with much better results than if a common digital operator was used. Simulations results show that, even under multiple simultaneous faults, the system presents very good results, as in the addition case, where a maximum standard deviation of 0.7 is achieved for sigma-delta-modulated signals, while for the digital adder alone, this value is 57.5. Such behavior is good enough to be used in operators that tolerate small errors, like in the digital filters where these errors are embedded in the system noise.

1. Introduction

As the size of the transistor channel decreases, also the number of electrons (or holes) in the channel decreases. With a lower number of carriers passing through the transistor channel, although the possibility of these carriers being hit by an external particle (e.g. alpha or neutron particles) decreases, if such event occurs, the effect caused will be much higher than if there was a higher number of carriers. Such interference may cause a Single Event Upset, or SEU [1, 2], which can cause, for example, a single bit flip that may damage the system or subsystem behavior in such way that the achieved answer can differ a lot from the expected one.

These phenomena may occur in digital, analog, and even optical components, or may have effects in surrounding circuitry. Figure 1 shows how a SEU may affect a system, through a simple change in the bit value.

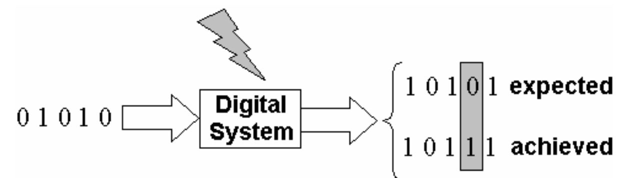


Figure 1: SEU causing a single bit-flip in a system.

As future technologies will suffer more and more from this transient behavior, one should search for new design approaches to reduce the effects of these faults, which can occur even in fully tested and approved circuits.

This work presents a technique that consists in modulating the signal to be processed in the sigma-delta domain, in such a way that the redundancy presented in the modulated signal can reduce the effects of the faults occurred during the system operation. Experimental results show that, even under several simultaneous faults, such circuits can maintain working operation, while a digital equivalent would collapse. For example, in the addition of two numbers, the maximum standard deviation is 0.7 while for the digital adder this value is 57.5.

The paper is divided as it follows: section II makes a review of some techniques used to improve the performance of systems subject to faults. In section III, the development of arithmetic operations using sigma-delta-modulated signals is investigated, and it is shown that the signals in sigma-delta domain are in fact less subject to suffer interferences of faults. Finally, section IV presents the proposed technique, comparing the use of sigma-delta modulators and common adders to develop different arithmetic operations, when both have faults injected in their structures.

2. Previous works

Many techniques have been developed to cope with transitory faults. In [3], two low-cost solutions to cope with SEU are compared: the error-detection capabilities of a hardware-implemented solution based on parity code and software-implemented solution based on source-level

code modification. The implementation of a new soft error tolerance technique based on time redundancy and/or space redundancy is presented in [4]. Also, the use of triple modular redundancy (TMR) is discussed and proposed in [5] and [6], which also proposes the use of double modular redundancy. These techniques were developed for single fault occurrence, and presented good results when this was the case, but can not cope with multiple simultaneous errors. Recently, [7] presented a technique that is able to cope with multiple soft errors. Such technique is based on the use of larger redundant words (bit streams) to represent the signals to be processed, and is supposed to be used in systems that work with the concept of error tolerance.

This work is based on the same idea presented in [7], however, the generation of the bit streams is developed through the use of a sigma-delta modulator. When compared with the bit stream process generation presented in [7], which is based on the use the probability associated with some digital quantity, the use of sigma-delta modulation can lead to a better representation of this value, once the signal-to-noise ratio of such systems can achieve very high values [8]. As a result, the reduction in the necessary length of the bit stream that represents the quantity can be developed, without loosing signal resolution. The results presented in this work have been obtained by the use of a digital first-order sigma-delta modulator, which converts an eight bits input signal to its sigma-delta domain representation.

3. Sigma-delta domain and fault tolerance

The main applications of sigma-delta modulators have been in the analog-to-digital and digital-to-analog converters, due to the great resolution achieved with such approach [8]. However, many others applications can be implemented with this versatile circuit: amplitude and phase modulators, correlators, companders, phase-locked loops and others. In [9] the sigma-delta domain is proposed as a valid signal-processing framework per se. Addition, multiplication and Boolean operators have been proposed and demonstrated.

Sigma-delta-modulated signals, although having a larger numbers of bits in its representation, have a great advantage if compared with code-modulated signals, when referring to fault tolerance: since sigma-delta-modulated signals are over sampled, each bit stream that represents an amplitude value carries an amount of redundancy that, even with some faults inserted, can still represent the mean value of the original signal. Table 1 shows an 8 bits signal modulated with a digital first order sigma-delta modulator as proposed in [10], represented in figure 2. As it can be observed, the sigma-delta modulator can still maintain a value that is very close to the original one after the insertion of a different number of faults, in

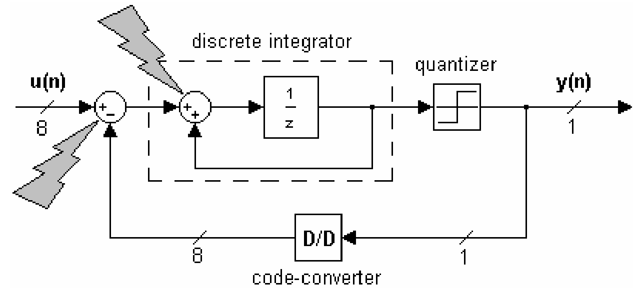


Figure 2: Digital first order sigma-delta modulator, according [10], with faults injections.

different bits of the subtractor and integrator of the sigma-delta modulator.

The first column of table 1 shows how many bits in the adder and in the subtractor of the sigma-delta modulator are randomly flipped. Since the signal to be modulated will be over sampled OSR (over sampling rate) times through the modulator, the fault will occur only in a number of cycles, which is represented in column 2 of table 1. The third column shows the original signal that will be sigma-delta modulated, while the last column presents the maximum deviation of the faulty value after being decimated. The OSR used to modulate the values is 256, and the signals were modulated for 20 times, with faults occurring in different moments.

Note that, although the value of the sigma-delta-modulated signal is not exactly the same of the original value, in applications that tolerate small error percentages or in systems that use cascading calculations, like digital filters, this approach can be used as an acceptable interference in the final result.

Table 1: Maximum deviation of sigma-delta modulated signals with faults inserted.

Number of randomly flipped-bits	Number of cycles that fault occurs	Original signal	Maximum deviation*
0	0	10	0.0391
1	2	10	0.6250
	5		1.2109
	7		1.6797
2	2	10	0.7422
	5		1.5625
	7		2.1484
3	2	10	0.7813
	5		1.9141
	7		2.2656
4	2	10	0.8594
	5		1.9141
	7		2.1484

* Maximum deviation from the original value, after modulating the signal 20 times.

In the following sections, some applications using sigma-delta-modulated signal are developed, in order to compare the results with the same application using common digital operators.

4. Fault-tolerate arithmetic operations using sigma-delta-modulated signals

This section presents some comparisons between the use of two ways of developing arithmetic operations: the first is using code-modulated signals, i.e., values that are simply represented by a certain number of bits in a predefined positional order; the second, using sigma-delta-modulated signals, i.e., values that are previously converted to the sigma-delta domain and are expressed by a bit stream that contains a number of bits proportional to the over sampling rate (OSR). Two arithmetic operations are developed: addition of two numbers and multiplication of a number by another. In order to show that the proposed approach presents good results regarding others applications, the integer square root of a number was calculated using the technique.

4.1. Addition of two numbers

The addition of two signals using sigma-delta results in a two-bits word stream, which can be further converted to the original value of the sum [9]. Figure 3 shows the systems used to add two constant values. In the first case (figure 3(a)), a common 8 bits full-adder is used, while in the second case, showed in figure 3(b), the same full-adder is used, but now with only 2 bits, since the input signals are modulated in the sigma-delta domain.

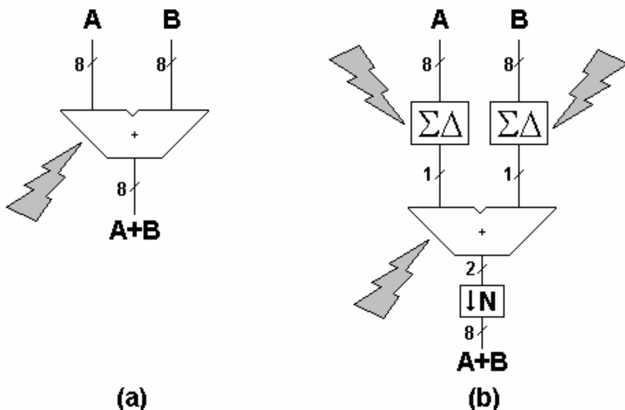


Figure 3: Proposed approach used to compare addition using common digital adder (a) and using sigma-delta-modulated signal (b).

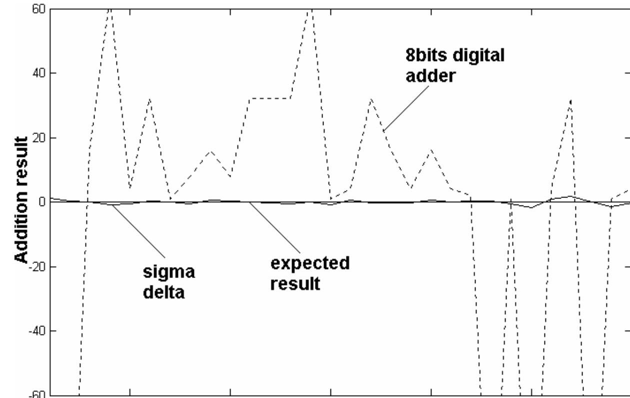


Figure 4: Addition results using code-modulated and sigma-delta-modulate signals.

As one can see in the Matlab® simulation results showed in figure 4, where the addition of +10 and –10 is to be developed, inserting faults in the sigma-delta modulator and in the digital adders, makes the results of the addition differ from the original value. Addition was developed 30 times.

For figure 3(a), where two 8 bits code-modulated signals are added, only one random fault is inserted in the adder, through the variation of one bit value. For the addition using sigma-delta-modulated signals, showed in figure 3(b), ten faults are inserted in the adder through the variation of ten random bits of the word stream generated at the output of the adder, and faults are inserted in the sigma-delta modulator, also through the variation of one random bit, during 2% of OSR, that is correspondent to 5 cycles. The OSR used is 256.

In the case of adding code-modulated signals, the deviation from the expected value of the addition, depending on the bit that is flipped, can be so large that the final value is even 128 times higher than the expected one. If one considers a higher order adder (16 bits, for example), this deviation can be even larger.

On the other hand, in the addition using sigma-delta-modulated signals, even with a higher number of faults, although the addition resulting value is not exactly the same of the expected one, its mean value is very close to it, enabling one to use such results in systems already designed to cope with a certain amount of noise, like in digital signal processing, where the errors could be seen as quantization noise.

4.2. Multiplication of two numbers

Multiplication of two numbers A and B can be developed through adding number A, B times. The same idea was used to prove that, even inserting faults in the adder and in the sigma-delta modulator (see figure 5), the multiplication of A and B converges to a value closer than

that achieved if the multiplication is developed through the use of a single adder with code-modulated values.

Figure 6 shows the results of multiplying constant values 3 and 4. Again, for the multiplication using the adder only, one fault in a random bit is inserted, and for the multiplication using the sigma-delta modulator and the adder, ten faults are inserted in the adder through the variation of ten random bits of the word stream generated at the output of the adder, and faults are inserted in the sigma-delta modulator, also through the variation of one random bit, during 2% of a 256 OSR, corresponding to 5 cycles. Multiplication was developed 30 times.

As one can see in figure 6, again the results for multiplication using the sigma-delta approach shows a very good result, approximating to the expected value, even with the insertion of multiple faults.

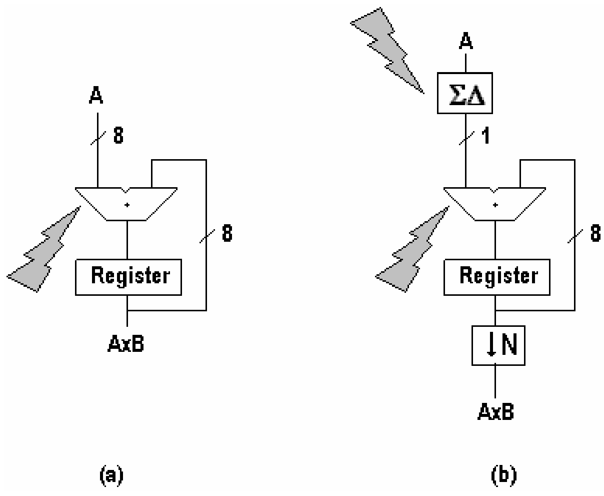


Figure 5: Proposed approach used to compare multiplication using common digital adder (a) and using sigma-delta-modulated signal (b).

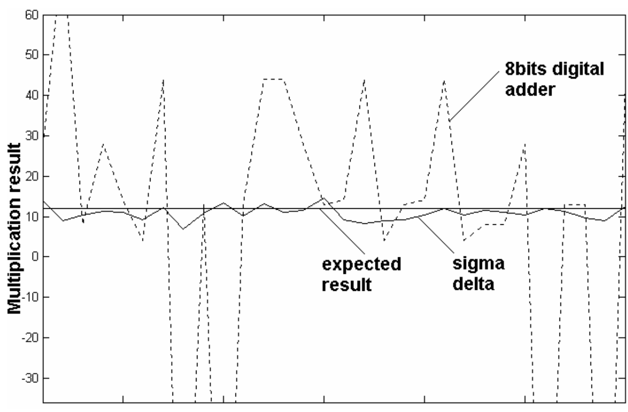


Figure 6: Multiplication results using code-modulated and sigma-delta-modulate signals.

4.3. Square root of an integer

A very simple method to estimate the integer square root of a number A is as shown in figure 7.

```
t=1; r=1; d=2; S=4;
while (t=1)
      r = r+1
      d = d+1
      S = S+d+1

      if S > A
         t=0
end
sqrt = r
```

Figure 7: Algorithm used to calculate the integer square root of a number.

This algorithm targets the integer square root of a number, and it can be implemented through the use of a single adder. To increase the robustness to faults, as it was done before in addition and multiplication, this algorithm was also developed using all the parameters (r, d, S and A) in the sigma-delta domain.

Faults are injected both in the adder and in the approach using the sigma-delta modulator, when calculating the value of 'r'. The number of faults is the same applied in the addition and multiplication example. Once more, the results obtained for the computation of the square root using the sigma-delta approach is approximated to the expected value, as it can be seen in figure 8, where square root calculation was developed 30 times.

Table 2 shows the standard deviation and the mean value of the three arithmetic operations presented, comparing the achieved results using only a digital adder

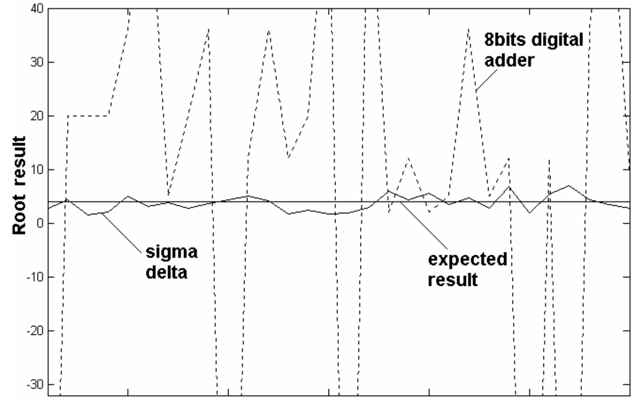


Figure 8: Square root results using code-modulated and sigma-delta-modulated signals.

Table 2: Standard deviation and mean value of the arithmetic operations developed using the two methods presented.

	Addition	Multiplica tion	Square root
Mean value using code-modulated	-7.0	4.8	0.6
Mean value using ΣΔ-modulated	-0.06	10.8	3.7
Expected value of the operation	0	12	4
Standard deviation using code-modulated	57.5	51.0	60.0
Standard deviation using ΣΔ-modulated	0.7	1.7	1.5

and code-modulated signals, and using sigma-delta modulators plus digital adders.

As mentioned before, the number of faults that were injected in the 8 bits digital signals is the same for all the operations, as well as the number of faults injected in the sigma-delta modulator and in the adder that use these signals.

5. Conclusions and futures works

This work presented the use of sigma-delta modulators in the generation of fault-tolerant signals, intended to be used in mean-based systems that make intensive use of arithmetic operations. It was shown that systems that use such technique can handle multiple faults, while still producing results that are much better than those produced with code-modulated signals.

Arithmetic operations were demonstrated (addition, multiplication and square root), all of them presenting results whose standard deviation are 0.7 in the case of addition, 1.7 for multiplication and 1.5 for square root, against 57.5, 51,0 and 60,0 from those using code-modulated signals.

Although the proposed approach leads to the necessity of high frequency systems, since the sigma-delta modulator needs high over sampling rates to achieve a good resolution, the high tolerance to multiple faults can make its use advantageous.

Future works include the investigation on the effects of the dynamic reduction of the over sampled ratio, in order to adjust the amount of tolerance to errors with the rate of errors present in the system, since particles do not reach the system at constant intervals. Also, we are developing more complex circuits, like FIR and IIR filters using sigma-delta-modulated signals, in order evaluate other parameters like power dissipation and total area.

6. References

[1] Yang, F.L.; Saleh, R.A. "Simulation and Analysis of Transient Faults in Digital Circuits". IEEE Journal of Solid-State Circuits, Volume: 27, Issue: 3, March 1992. Pages: 258 – 264.

[2] Messenger, G.C. "A summary review of displacement damage from high-energy radiation in silicon semiconductors and semiconductor devices". IEEE Transactions on Nuclear Science, Volume: 39, Issue: 3, June 1992. Pages: 468 – 473.

[3] M. Rebaudengo, M. Sonza Reorda, M. Violante, B. Nicolescu, R. Velazco. "Coping with SEUs/SETs in Microprocessors by means of Low-Cost Solutions: A Comparative Study". IEEE Transactions on Nuclear Science, Volume: 49, Number: 3, June 2002. Pages: 1491 – 1495.

[4] Anghel, L., Alexandrescu, D. and Nicolaidis, M. "Evaluation of soft error tolerance technique based on time and/or space redundancy". Proceedings of 13rd IEEE Symposium on Integrated Circuits and Systems Design (ICSD 2000),, Manaus, Brazil, September 2000. Pages: 237 – 242.

[5] Yoon, J.; Kim, H. "Time-redundant recovery policy of TMR failures using rollback and roll-forward methods". IEEE Proceedings Computers and Digital Techniques, Volume: 147, Issue: 2, March 2000. Pages: 124 – 132.

[6] Lima, F., Carro, L. and Reis, R. "Techniques for Reconfigurable Logic Applications: Designing Fault Tolerant Systems into SRAM-based FPGAs". Proceedings of the International Design Automation Conference, DAC 2003, ACM, New York, 2003. Pages: 650 – 655.

[7] Lisbôa, C. and Carro, L. "Arithmetic Operators Robust to Multiple Simultaneous Upsets". Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems - DFT 2004, pp. 289-297, ISBN0-7695-2241-6. IEEE Computer Society, New York, October 2004.

[8] Norsworthy, S.R., Schreier, R., Temes, G. C. "Delta-sigma data converters: theory, design and simulation". New York: IEEE Press, 1997. 476p.

[9] da Fonte Dias, V. "Sigma-delta signal processing". IEEE International Symposium on Circuits and Systems, 1994. ISCAS'94, Volume: 5, 30 May-2 June. Pages: 421 – 424.

[10] Robert, G. W., Lu, A. K., "Analog Signal Generation for Built-In-Self-Test of Mixed-Signal Integrated Circuits". Kluwer Academic Publishers, 1995. 122p.