Mitigation of Soft and Hard Errors in FPGA-Based Pacemakers

Gehad I. Alkady
Electronics and Communications Engineering Department
American University in Cairo (AUC)
Cairo, Egypt
g.i.alkady@aucegypt.edu

Ihab Adly
Tegrom
Cairo, Egypt
ihab.adly@tegrom.com

Hassanein H. Amer
Electronics and Communications Engineering Department
American University in Cairo (AUC)
Cairo, Egypt
hamer@aucegypt.edu

Tarek K. Refaat
Engineering Department
Cisco Systems, Inc.
Ottawa, ON
tarek.k.r@ieee.org

***Abstract*—Nowadays, there is a great improvement in Implantable Medical Devices (IMD) manufacturing. The Pacemaker is one of the commonly used IMDs responsible for curing improper heart rate pacing. As a real-time critical device, it requires a powerful flexible platform such as FPGAs. Moreover, it should have an area-optimized architecture. In this research, a novel architecture of the pacemaker is proposed through the integration of some modules of the pacemaker together: The Analog to Digital converter (A2D) and the pulse generator modules. Both are implemented inside the same FPGA as a single chip solution. This reduces the footprint of the pacemaker without having any negative impacts on system performance. The system is implemented using the Xilinx Vivado tool. FPGA-based pacemakers are prone to both SEUs and permanent faults. Consequently, a suitable Fault-Tolerant technique should be applied to increase system reliability. Two schemes are investigated in this research: Two Duplication With comparison (2DWC) and Sift-Out. Both use Dynamic Partial Reconfiguration (DPR) for fault recovery. Markov Models for both techniques are constructed to evaluate system reliability. A case study shows that the Sift-Out architecture is the optimum technique for the system.**

Keywords—pacemaker; FPGA; Dynamic Partial Reconfiguration (DPR); fault tolerance; reliability; sift-out; two DWC; markov model.

I. Introduction

Pacemakers are one of the most frequently used Implantable Medical Devices (IMDs). They are composed of numerous modules including an electrical sensing module, an Analog to Digital converter (A2D), a processing module (pulse generator), a stimulation module, a power source (battery), leads and a communication module [1]. The core component of the pacemaker is the pulse generator. As a result, several requirements must be considered when choosing the suitable design for the pulse generator because this has a high impact on system performance and reliability. Multi-programmability, exceptional reliability, longevity and both diagnostic functions and data collection capabilities are all highly desirable features in such systems [1]. All these features can be guaranteed through Field Programmable Gate Array (FPGA) implementations.

FPGAs are integrated circuits that are designed to be programmed in the field for any digital circuit or system implementation. They include a matrix of Configurable Logic Blocks (CLBs) connected via programmable interconnections. Moreover, they interact with the surrounding environment through programmable input/output (I/O) ports. FPGAs can be partially reconfigured during runtime through Dynamic Partial Reconfiguration (DPR) [2]. Static Random-Access memory (SRAM)-based FPGAs are commonly used in most highly critical applications.

Electromagnetic interference is a dilemma for pacemakers [3]. In addition, it increases the SRAM-based FPGA susceptibility to Single Event Upsets (SEUs) that induce errors mainly within the configuration memory of the FPGA. In addition, block RAM and user-defined flip-flops also prone to SEUs. SEUs results from flipping memory elements in an FPGA and it is a transient fault that can be recovered from by reconfiguring the FPGA. Accordingly, fault mitigation techniques should be used in FPGA-based pulse generator implementations to improve their reliability.

Several studies refer to FPGA-based pulse generator implementations. In [4], an FPGA is chosen to implement the pulse generator circuit. The reliability of the system is increased by applying the Duplication With Comparison (DWC) Fault-Tolerant technique to the circuit for fault detection. As the fault model considered was permanent faults, the recovery mechanism consisted of repositioning the faulty module in a new location using DPR. Reference [5] proposed a pulse generator implemented on FPGAs but it focused on diminishing FPGA power consumption through power gating techniques. While in [6], the implementation of an FPGA-based pacemaker was presented and the ECG signal was produced by Matlab.

An FPGA-based pulse generator is implemented in [7] and DPR is used to enable altering the pacemaker's mode runtime; this can be controlled by the doctor remotely through a wireless network. This is very convenient for the patient because there will be no need for replacing the pacemaker by surgery. This paper also reports a case study where it is shown how to choose the appropriate Fault-Tolerant technique that should be applied to the FPGA-based pulse generator to increase its reliability. Sift-Out and Two Duplication With comparison (2DWC) circuits are the Fault-Tolerant techniques under study. SEU is the fault model. After calculating system reliability, results show that, for this specific case and the specific parameters used, the Sift-Out technique is better than 2DWC.

In this paper, a modification will be made to the architecture of the FPGA-based pulse generator in [7]. This modification is the implementation of both the Analog to Digital converter (A2D) module and the pulse generator module using one FPGA. The 7th series devices of Xilinx guarantee the existence of analog inputs and XADC cores in FPGAs. As a result, the A2D can be designed along with the pulse generator on one chip in order to reduce the footprint of the pacemaker. The Xilinx Vivado tool is used in the implementation. A new fault model is used that consists of SEUs as well as permanent faults. For this reason, the choice of the appropriate Fault-Tolerant technique for the FPGA-based pulse generator should be investigated. Two Duplication With comparison (2DWC) and Sift-Out are the Fault-Tolerant schemes under study. For fault recovery in both techniques, DPR is used. The reliability of the two models is calculated by using Continuous Time Markov Chains (CTMCs) and a case study is presented. It shows that the Sift-Out technique is more appropriate than the 2DWC technique in terms of reliability. As a result, the Sift-Out technique is chosen.

The rest of this paper is organized as follows: Section II presents a summary about the pacemaker architecture. Section III introduces the proposed architecture. While Section IV describes the Fault-Tolerant architectures considered in this research. In Section V, a reliability case study is carried out. Section VI concludes this research.

II. PACEMAKER

One of the commonly used Implantable Medical Devices (IMDs) is the cardiac pacemaker. It is required for treating bradycardia which occurs because of a too-slow heart rate condition and tachycardia that happens due to a very fast heart rate condition. Also, it is required in curing the failure in the connection between the atria and ventricle which results in decreased cardiac output.

Pacing circuits are mainly composed of: the pacemaker, leads and a telemetry device [1]. The Pacemaker is the basic component of the pacing circuit which is responsible for sensing the electrical activity and generating stimuli that are sent through the leads to the heart. Leads are implemented from insulated wires. They are responsible for relaying intrinsic cardiac signals from the heart to the pacemaker and transmitting the stimulus pulses from the pacemaker to the heart. While the telemetry device is a programmer through which the doctor can modify the therapy delivered by the pacemaker and retrieve crucial diagnostic data.

The Pacemaker consists of several blocks as shown in Fig. 1, namely a sensing circuit, an Analog-To-Digital (A2D) converter, a pulse generator, a stimulation circuit, a communication module and a power source (battery) [1].

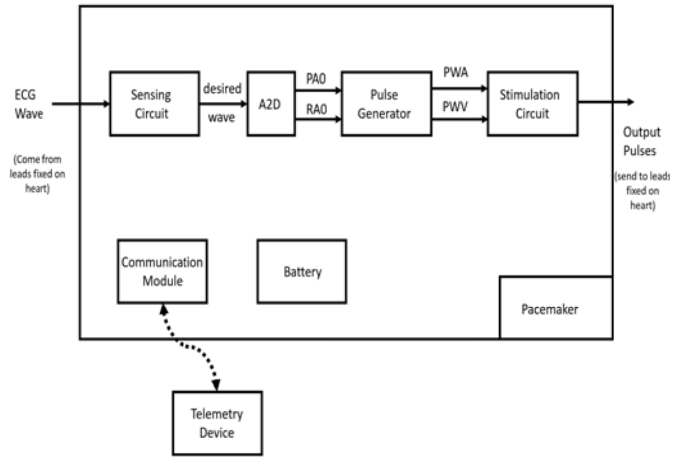

Fig. 1. Pacemaker [7].

Firstly, the sensing circuit is mainly responsible for identifying the desired signals from the ECG waves; it has both intracardiac signals (desired signals) and unwanted electrical interference coming from the heart through the leads. As the desired signals are analog, an Analog-To-Digital (A2D) converter is required to convert the signals into a digital form. The output of the A2D depends on the number of leads fixed on the heart and their locations. If the pacemaker has only one lead fixed on the atrium or the ventricle, one digital output is produced. While if two leads are fixed on both atrium and ventricle, two output signals will be generated: PA0 (ECG wave atrial depolarization) and RA0 (ECG wave ventricular depolarization).

The pulse generator plays several roles in the pacemaker including timing control, processing of signals, generation of stimulus and shaping. According to whether the pacemaker is a single chamber (one lead is fixed on the atrium or the ventricle) or a dual chamber (two leads are fixed on both the atrium and the ventricle), the number of outputs is identified. Accordingly, it can produce two programmed output stimuli which are PWA (atrium) and PWV (ventricle). The pulse generator can be designed as an integrated logic circuit [8] or a microprocessor [9]. The stimulation module is the module which produces the output pulses transmitted through the leads to the heart. These output pulses are generated through discharging a battery-charged capacitor and they are produced at a rate controlled by the pulse generator. The communication module is responsible for establishing a two-way communication between the pacemaker and the telemetry device. In addition, the power source for the pacemaker is a battery having a lifetime which varies from 4 to 9 years. There are several types of pacemakers which differ according to the functioning mode of the pacemaker: VOO, AOO, VVI, VVIR, DDD and VDD [10].

The pulse generator is the intelligence of the pacemaker; it can be designed using a microprocessor, an Application Specific Integrated Circuit (ASIC) or a Field Programmable Gate Array (FPGA) [11]. FPGAs are an advantageous implementation alternative for pacemakers because, through the Dynamic Partial Reconfiguration (DPR) feature [2], design errors can be easily fixed and the pacing mode of the pacemaker (operational mode) can be altered with a small change in the architecture of the pulse generator without the need for surgically replacing the pacemaker. All sensors needed for the different pacemaker operational modes should be available in the design.

Accordingly, in [7], an FPGA-based pulse generator is designed and changing the pacemaker mode during runtime by DPR, is proposed. Moreover, this can be done remotely by the doctor via a wireless network. As a result, it is not essential for the patient to go to the hospital. Pacemakers are strongly susceptible to electromagnetic waves [3]. For this reason, Fault-Tolerant techniques should be applied to increase the reliability of the system. Two Fault-Tolerant techniques: Sift-Out and Two Duplication With comparison (2DWC) circuits are investigated. The fault model is Single Event Upsets (SEUs). Reliability calculations indicate that the Sift-Out technique is more reliable than 2DWC technique. Note that the results depend on the specific parameter values used in the case study.

III. Proposed Architecture

A pacemaker is a highly critical IMD that saves the life of many patients; accordingly, the pacemaker manufacturers are constantly improving and updating their designs and increasing reliability. One area for enhancement is shrinking the size of the pacemaker. Since 2011, Medtronic, as one of the largest pacemaker manufacturers, has been working on decreasing pacemaker sizes [12]. This can be achieved by reducing the footprint of the pacemaker. Consequently, the design of the pacemaker proposed in [7] is modified to follow the same approach. Modifications are proposed for the design of the Analog-To-Digital (A2D) converter and the pulse generator modules, implementing both modules on the same FPGA.

Xilinx 7-series devices have analog inputs and XADC cores [13], thus facilitating the integration of the A2D along with the pulse generator inside the FPGA to form a one-chip solution as shown in Fig. 2.

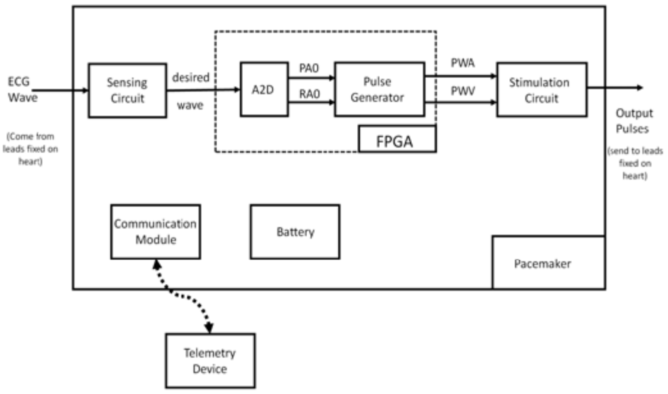

Fig. 2. Proposed Pacemaker Architecture.

There are tremendous benefits of having a one-chip solution. Implementing multiple modules/operations in a high-level circuit specification using the same fabric allows resource sharing which is one of the keys of the area-reduction approach. The footprint of the system is decreased without degrading system performance. In addition, a single chip solution supports the usage of the same tool chain for the integrated modules, guaranteeing seamless compilation of all modules.

To test the proposed architecture, the system is implemented using the Xilinx Vivado tool. The Xilinx device XC7S50, from the family Spartan 7-series, has been chosen for the design implementation. In the pacemaker architecture in [7], the inputs of the A2D are the desired signals received from the sensing circuit and are the analog signals picked from the atrium and the ventricle. These signals are converted to digital through the A2D, producing two signals: PA0 (ECG wave atrial depolarization) and RA0 (ECG wave ventricular depolarization). The pulse generator, implemented by a processor, is fed with the PA0 and RA0 signals. The pulse generator then produces the PWA (atrium) and the PWV (ventricle) signals, which in turn are connected to the stimulation circuit of the pacemaker. To build the system in Vivado, a MicroBlaze processor is used along with the XADC core as shown in Fig. 3.

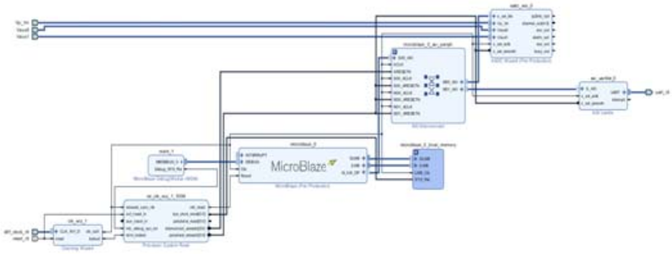

Fig. 3. Proposed SoC Architecture in Xilinx Vivado tool.

The resources utilization report indicates the use of 1611 LUTs, 1451 FFs and 8 BRAMS (36 Kb each) to implement the proposed design. These resources consume only about 5% of the FPGA resources. A screenshot of the system floorplan is shown in Fig. 4.

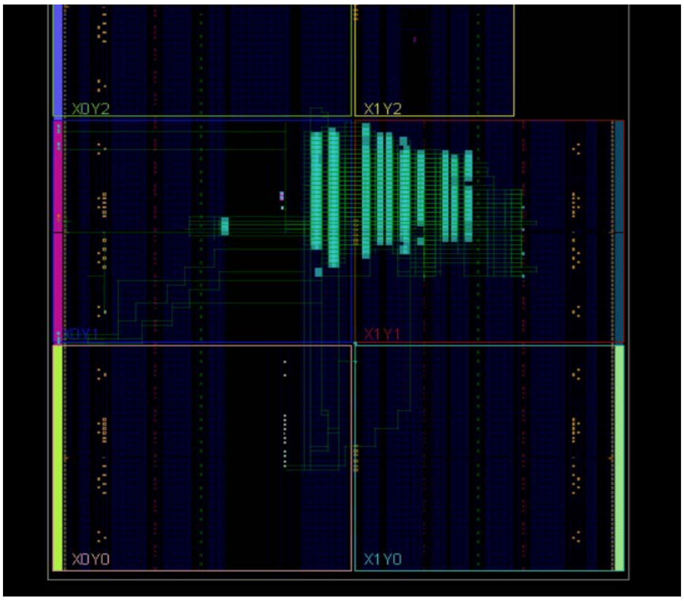

Fig. 4. FPGA floorplan of the proposed system.

IV. Fault-Tolerant Architecture

In [7], two fault-tolerant architectures were used to increase reliability and prolong system lifetime, namely Sift-Out and Two Duplication With Compare (2DWC). Both architectures require four identical copies of the module protected by fault tolerance.

In Sift-Out, four identical modules have their outputs connected to a detection and fault recovery mechanism. When one output is different from the other three outputs, the corresponding module is “disconnected’ from the error detection and recovery mechanism. The system resumes correct operation with three modules. It can sustain a second module failure. The module with erroneous output is again disconnected and the system now has only two operational modules. If the outputs of these two remaining modules are different, the error cannot be located and the entire fault-tolerant system fails.

In 2DWC, there are four identical modules grouped in two pairs. The outputs of each pair are compared; if they differ, the erroneous module cannot be determined and the entire pair is disconnected from the system while the other pair is switched in.

In both Sift-Out and 2DWC, DPR can be used. When an error is detected in the Sift-Out system, the erroneous module is overwritten with a “clean” bitstream and the system will have recovered from the error. The same is true for the 2DWC architecture; when one pair malfunctions, the other pair is switched on while the erroneous pair is overwritten with a clean bitstream [7].

A. Expanded Fault Model

Most of the information in the literature regarding errors in SRAM-based FPGAs focuses on SEUs [14]. They are considered to be the predominant type of fault. As mentioned in the introduction, an SEU will affect the system by flipping a single bit in the FPGA. If this bit is within the configuration memory, this may change the function of a module. However, the damage is not permanent; if the affected cell is overwritten with the correct information, the effect of the error disappears, and the module can resume correct operation. This is the advantage of DPR. However, it is important not to neglect permanent failures in FPGAs. Examples of permanent failures are Time Dependent Dielectric Breakdown (TDDB), electro-migration, hot carrier effect and open faults [15]. Even though the rate of permanent failures is less than that of SEUs, they do occur and affect system reliability. Therefore, in this research, the fault model for SRAM-based FPGAs will consist of both SEUs and permanent failures. Next, the calculation of the failure rate will be explained. Let “l_t” be the failure rate that corresponds to the fault model used in this research. It is the sum of the failure rate of SEUs (soft errors) “l_s” and the failure rate of permanent faults (hard errors) “l_h” as shown in equation (1). Assume that the frequency of occurrence of SEUs is about ten times higher than that of the permanent faults; hence, l_s is equal to ten times l_h.

$$l_t = l_s + l_h \tag{1}$$

$$l_s = l_h * 10 \tag{2}$$

B. Enhanced Fault Tolerance

To further increase pacemaker reliability, both fault-tolerant techniques described above are modified as follows. A partial reconfigurable block is reserved in the FPGA during the design time as a spare location. This location is kept empty and will be considered as a spare location. For the Sift-Out architecture, an area capable of containing one module is determined. Let it be called Spare Location for Sift-Out (SL_siftout). When the first of the four modules fails due to a permanent failure, the partial bit file for this module is downloaded into SL_siftout and the entire system resumes operation with four modules: three of the original four modules in addition to the module in SL_siftout. This hybrid redundancy technique is expected to increase system lifetime [16]. For the 2DWC architecture, the same concept applies. A Spare Location (SL_2DWC) is determined that is large enough to host one pair (two modules). When one of the two pairs fails permanently, the partial bit file for a pair of modules is downloaded into SL_2DWC and the system resumes operation with two pairs, one of the original two pairs in addition to the spare pair in SL-2DWC. Again, this hybrid redundancy technique is expected to increase system lifetime.

C. Reliability Calculations

The expected increase in reliability is quantitatively measured in this section. Markov models are used to obtain the reliability of both the Sift-Out and the 2DWC architectures.

Fig. 5 shows the Markov model of the Sift-Out architecture with four modules.

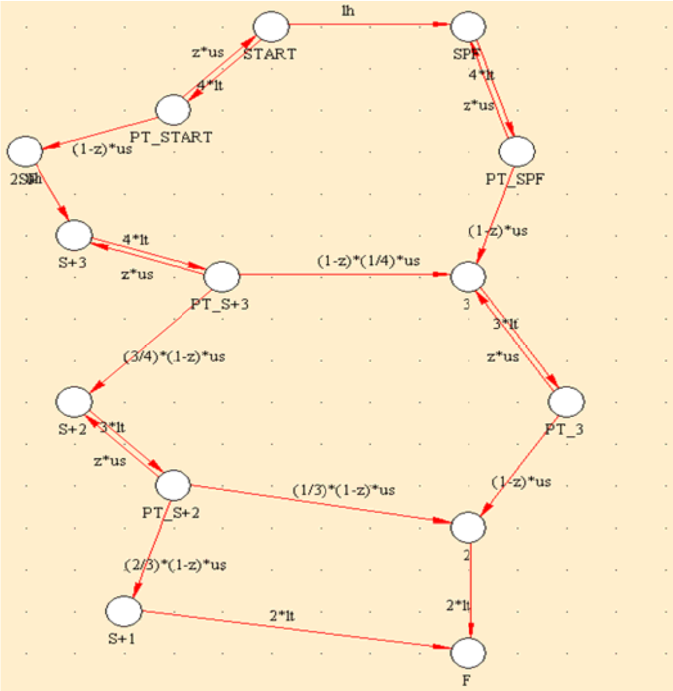

Fig. 5. Sift-Out Markov Model.

State START is obviously the starting state; all four modules are operating correctly and SL_siftout is empty. Upon the failure of one of the four modules (at a rate of 4*l_t), the system moves to state PT_start. In this state, the type of failure (whether permanent or temporary) is still unknown. A partial bit file for the module is downloaded (using DPR) in the area occupied by the failed module. During this time, the fault-tolerant system is operating correctly with the three operational modules. After this repair, the downloaded module will either operate correctly if the failure was temporary or will produce erroneous outputs if the failure was permanent. For a temporary failure, the system moves back to state START. The rate of this change is z*μ_s where *z* is the conditional probability that the failure is temporary given that the failure has occurred. If the failure turns out to be permanent, the system moves to state 2SP. In this state, DPR is used to download a partial bit file of the module in the spare location SL_siftout. Let the rate of this transition be (1-z) *(μ_s). The system then moves to state S+3. In this state, 3 out of the 4 original modules are still operational and the fourth module is now in the spare location.

While in state S+3, any of these four modules can fail. Hence, at a rate of 4*l_t, the system will move to state PT_S+3. The type of failure is still unknown. DPR is used to download a copy of the module; if the failure turns out to be transient, the system transitions back to state S+3 at a rate of z*μ_s. If the failure is permanent, the system moves to state S+2, i.e., 3 modules are still operational.

In state START, assume a failure occurs in the spare location SL_siftout. Since this location is not used, a temporary failure will go unnoticed and will not affect the system because, when DPR is used to download a copy of the module into this area, the problem will be automatically corrected. On the other hand, if SL_siftout is affected by a permanent failure, it will not affect system operation till a copy of the module has to be downloaded. It will be assumed in this research that techniques are run in the background on SL_siftout; when a permanent failure is detected, the system will be notified that this location is no longer useable. As soon as the permanent failure affects SL_siftout, the system moves to state SPF at a rate of l_h. The original four modules are operating correctly, SL-siftout has permanently failed and the system is aware of this problem but operates normally.

While in state SPF, any of the four original modules can be affected by a failure at a rate of 4*l_t. The system moves to state PT_SPF where it is still not known whether the failure is permanent or temporary. Using DPR, a partial bit file for the module is downloaded in the location of the failed module. If the failure was temporary, the system will move back to state SPF at a rate of z*μ_s. If the failure was permanent, the system will not attempt to download a module in SP_siftout since it is known that this location cannot be used. Instead the system moves directly to state 3. In this state, the system has three of the original modules still operational, one original module has failed and the spare location SP_siftout is unusable. The remaining states follow the same reasoning and F is the system failure state.

Assume P_i (t) is the probability of residing in state i where i ϵ {START, PT_START, 2SP, S+3, PT_S+3, S+2, PT_S+2, S+1, SPF, PT_SPF, 3, PT_3, 2, F} at time t. The transient probability of residing in any of the fourteen states can be calculated using the Chapman-Kolmogorov equations [16]:

$$\frac{dP}{dt} = PXT \tag{3}$$

P is the matrix containing the probabilities of each state i, and T is the Transition Rate Matrix. Assume that $P_{START}(0) = 1$ and $P_{PT\_START}(0) = P_{2SP}(0) = P_{S+3}(0) = P_{PT_S+3}(0) = P_{S+2}(0) = P_{PT_S+2}(0) = P_{S+1}(0) = P_{SPF}(0) = P_{PT\_SPF}(0) = P_3(0) = P_{PT\_3}(0) = P_2(0) = 0$. By replacing T and P in equation (3), the Chapman-Kolmogorov equations can be determined and P_i (t) for each state *i* can be calculated. Accordingly, system reliability R(t) at time instance *t* can be calculated using equation (4).

$$R(t) = 1 - P_F(t) \tag{4}$$

Fig. 6 has the Markov model for the 2DWC architecture.

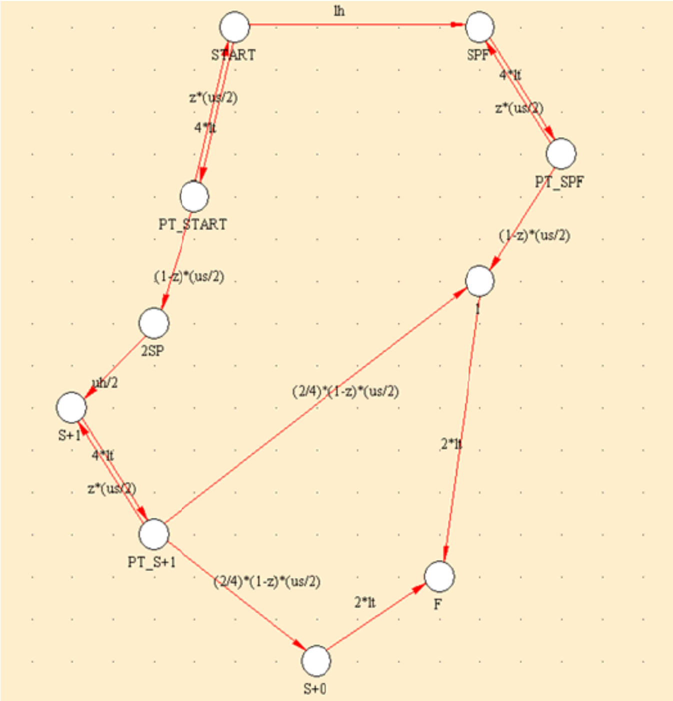

Fig. 6. 2 DWC Markov Model.

The system starts in state START where both pairs are operational but only one of them is connected to the system output. The model follows the same reasoning used for the previous Markov model and F is the system failure state.

V. Case Study

The appropriate Fault-Tolerant technique for the system implementation is selected based on the reliability calculations. This requires solving the two Markov models (presented in Section IV) of 2DWC and Sift-Out by using the SHARPE modeling software [17]. It should be taken in consideration that the results obtained from the two Markov models depend heavily on the specific parameter values.

Table I shows the reliability of the system after the implementation of both techniques: 2DWC and Sift-Out. Assume that l_s=1/month which represents 1 failure per month. This corresponds to the SEU failure rate for a pacemaker that is implanted in a patient living in an environment with electromagnetic interference. In addition, l_h=0.1/month which represents the failure rate of hard (permanent) faults and $\mu_s= \mu_h$ = 38686567/month (which corresponds to a mean repair time of about 67 ms – this is the DPR time for recovering from a failure in an FPGA-based processor [18]) and z=0.99 [7].

TABLE I. CASE STUDY

Time (month)	**Reliability R(t) as a(%)**	
	2DWC	***Sift-Out***
0	100	100
12	75.1436064	97.0584856
24	461064853	85.7567198
36	27.4312442	70.2863177
48	16.3034033	54.8350499

It is observed that the reliability of the system after applying the Sift-Out redundancy technique in the pulse generator is better than the reliability of the system after 2DWC scheme is used. After 4 years of operation, the reliability of the Sift-Out architecture is more than three times higher than that of the 2DWC architecture even though both architectures consume the same resources in the FPGA. In conclusion, for this specific case study and the specific values used for the failure and repair rates, Sift-Out is the suitable Fault-Tolerant technique for the FPGA-based pacemaker.

VI. CONCLUSION

One of the vital Implantable Medical Devices (IMD) is the cardiac pacemaker that acts as a highly critical device and requires optimum specifications. This can be assured by implementing the pacemaker using a powerful platform such as FPGAs. Moreover, an optimum architecture should be chosen. In this research, a new design for the pacemaker is presented. This novel design is based on integrating some blocks of the pacemaker together sharing the same FPGA chip. These blocks are the Analog to Digital converter (A2D) and the pulse generator modules. Accordingly, the footprint of the pacemaker will be minimized which is one of the goals of the pacemaker manufacturers. Furthermore, this will not have any negative effect on system performance. The Xilinx Vivado tool is used in system implementation. Since FPGA-based pacemakers are susceptible for SEUs and permanent faults, fault mitigation techniques should be applied to increase system reliability. Many Fault-Tolerant techniques are described in the literature including Triple Modular Redundancy, Sift-Out and Duplication With Comparison (DWC). Sift-Out and two DWC are chosen in this research. Both techniques require the use of four identical copies of the module to be protected. To select the proper Fault-Tolerant technique, Markov models for both techniques are developed for reliability calculations. A case study shows that when the Sift-out technique is applied, the reliability of the system is higher than when using two DWC in system implementation.

REFERENCES

[1] M. Forde and P. Ridgely, "Implantable cardiac pacemakers, " CRC Press LLC, 2000.

[2] Xilinx, "Xilinx partial reconfiguration user guide, " UG702, April 2013.

[3] L. Santinia, G.B. Forleoa and M. Santini, "Implantable devices in the electromagnetic environment, " Journal of Arrhythmia, vol. 29, no. 6, pp. 325-333, December 2013.

[4] G.I. Alkady, N.A. El-Araby, M.B. Abdelhalim, H.H. Amer and A.H. Madian, "Dynamic fault recovery using partial reconfiguration for highly reliable FPGAs," Proceedings of the 4th Mediterranean Conference on Embedded Computing MECO, Budva,Montenegro, June 2015.

[5] H. Trang and L.T. Khoa, "Power gating technique in pacemaker design on FPGA, " Proceedings of the 2012 International Conference on Advanced Technologies for Communications ATC, Hanoi, Vietnam, October 2012.

[6] T. Roopa, B.M. Manjula and C. Sharma, "Implementation of a pacemaker for biomedical application, " International Journal of Science and Research IJSR, vol. 3, no.6, June 2014.

[7] G.I. Alkady, H.H. Amer and R.M. Daoud, "Remotely configurable fault-tolerant FPGA-based pacemaker," Proceedings of the IEEE 12th International Conference on Computer Engineering and Systems ICCES, Cairo, Egypt, December 2017.

[8] Herpers, "Atrial and Ventricular Pacemaker, " European Patent Application, 82400236.4, Feburary 1982.

[9] L.T. Khoa and H. Trang, "Design an optimized CPU architecture for pacemaker applications," Proceedings of the 4th International Conference on the Development of Biomedical Engineering, Hochiminh, Vietnam, September 2012.

[10] Pacingdefibrillation: http://www.pacingdefibrillation.com/en/theory/pm/pacing-modes.

[11] W.M. Demmer, "Techniques for minimizing current drain in an implantable medical device," United States Patent Application Publication, US 2016/0279424 A1, September 2016.

[12] Technology Review: https://www.technologyreview.com/s/423134/a-pacemaker-the-size-of-a-tic-tac/

[13] Xilinx, "7 Series FPGAs Data Sheet: Overview," DS 180 datasheet, Feburary 2018.

[14] Naifeng Jing, Ju-Yueh Lee, Weifeng Hel, Zhigang Mao and Lei He, "Mitigating FPGA Interconnect Soft Errors by In-Place LUT Inversion," Proceedings of the 2011 IEEE/ACM International Conference on Computer Aided Design (ICCAD), San Jose, CA, USA , November 2011.

[15] J. Srinivasan, S. V Adve, P. Bose and J.A. Rivers, "The impact of technology scaling on lifetime reliability," Proceedings of the International Conference on Dependable Systems and Networks (DSN), Florence, Italy, July 2004.

[16] D.P. Siewiorek and R.S. Swarz, "Reliable computer systems design and evaluation, " A K Peters, Natick, Massachusetts, 1998.

[17] Official site for SHARPE: http://sharpe.pratt.duke.edu.

[18] C. Gauer, B.J. LaMeres and D. Racek, "Spatial avoidance of hardware faults using FPGA partial reconfiguration of tile-based soft processors," Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, March 2010.