

Original Article

Radiation Tolerant PLL for Onboard FPGAs

Sourabh Kumar Jain¹, Usha Mehta², Mohit Sharma³, Aarushi Bhandari⁴, Kamal Poddar⁵, Sanjay Trivedi⁶

^{1,3,4,5,6}Space Applications Centre, Indian Space Research Organization, Ahmedabad, India

²Nirma University, Ahmedabad, India

¹Corresponding Author : sourabhjain@sac.isro.gov.in

Received: 01 March 2023

Revised: 06 April 2023

Accepted: 16 April 2023

Published: 30 April 2023

Abstract - This paper proposes Fault Detection Isolation and Recovery (FDIR) techniques for in-built hard IPs of Phase Locked Loop (PLL) or Digital Clock Manager (DCM) of Field Programmable Gate Array (FPGA). PLLs inside the FPGAs are susceptible to Single Event Effects (SEE) even for space-qualified FPGAs. Under the radiation environment, loss of lock, perturbation in the clocks or no clocks are the common phenomena observed. Thus, we developed an Intellectual Property (IP) core based on FDIR, which adds redundancy to the PLL output and improves the Mean Time to Failures (MTTF). The generic IP core is technology independent and can be configured in dual or triple use of PLL mode. All the PLLs are kept as hot redundant mode, and their health is continuously monitored. SEE-affected PLL is immediately isolated, and the output is switched to a healthy PLL (if required) with or without a few cycle gaps. Recovery of faulty PLL is carried out in parallel. Telemetry signals notify every PLL switching and reset. For triple redundant PLL, minimum switching is ensured by logic. Redundancy bypass and selection of PLL make it work in case of complete failure of one or two PLL.

Keywords - Clock multiplexing, DMR, FDIR, FPGA, PLL, Radiation tolerant, SEE, SEU, TMR.

1. Introduction

In the current space era, payloads of various spacecraft are expected to have all the intelligence and perform various tasks starting from capturing, regenerating and processing data to ready-to-use data products. This is being achieved by adding more and more digital logic, and implementing complex algorithms, i.e., Digital Signal Processing (DSP), Artificial Intelligence (AI), Machine Learning (ML), high-speed data interface etc. [1]. These logic are encapsulated in a resource-rich FPGA to reduce weight, power and footprint area. Different algorithms have different clocking requirements, and in general, this is fulfilled by using the PLL or other clock synthesis circuit available inside the FPGAs as Hard or Soft IPs [2]. The available IP cores of PLL, DCM or other clock synthesis circuits are susceptible to SEE caused by space radiations. The upsets caused by SEE lead to distortion in output clocks, hampering the ongoing functionalities.

The remainder of this paper is structured as follows. Section II highlights the limitation of the PLL/DCM IP cores available in various FPGA families. Section III gives an overview of PLL and its working. The proposed FDIR-based radiation mitigation solution is detailed in Section IV. Section V presents the implementation of different configurations, their results and their comparison. Section VI analyses the reliability of the proposed solution. Finally, in Section VII, concluding remarks are made.

2. Current Limitation

Most available radiation-tolerant/hardened FPGAs such as Virtex4, Virtex5, KU060 from Xilinx, and RTG4

from Microchip do not guarantee the radiation-hardened performance of their PLL or DLL [3]. The high value of the lock signal in PLL/DLL is the primary indication of the working of PLL and desired output. In case of any abnormality in the input clock or power supply or due to any other external factor, including the radiation effect, the PLL outputs get affected, and the lock signal goes low. This lock is not reacquired until the PLL is reset [4]. When used onboard such PLLs, the lock is monitored as a health parameter in telemetry. In case a loss of lock is detected, a manual reset is issued through telecommand, as telemetry is a slow-rate signal. Hence, detection of the loss of lock and resetting of the PLL consume a significant amount of time. During this interval, loss or interruption in the payload functionalities working on the PLL output clock is observed. This interruption in the PLL output clock may be critical based on the application or design functionalities.

3. PLL Overview

Fig.1 shows a typical block diagram of charge pump-based PLL available in Xilinx FPGAs [5]. It consists of a Clock Switch Circuit, a Phase Frequency Detector (PFD), a Charge Pump (CP), a Loop Filter (LF), Voltage Controlled Oscillator (VCO) and a multiplexer for feedback clock selection.

The PFD detects the phase difference between the reference input clock and the divided VCO's output. When the generated clock is lagging or leading the reference clock, a pulse signal is generated at the PFD output. These pulses are used to switch voltage or current sources of CP, which charge or discharge its output.



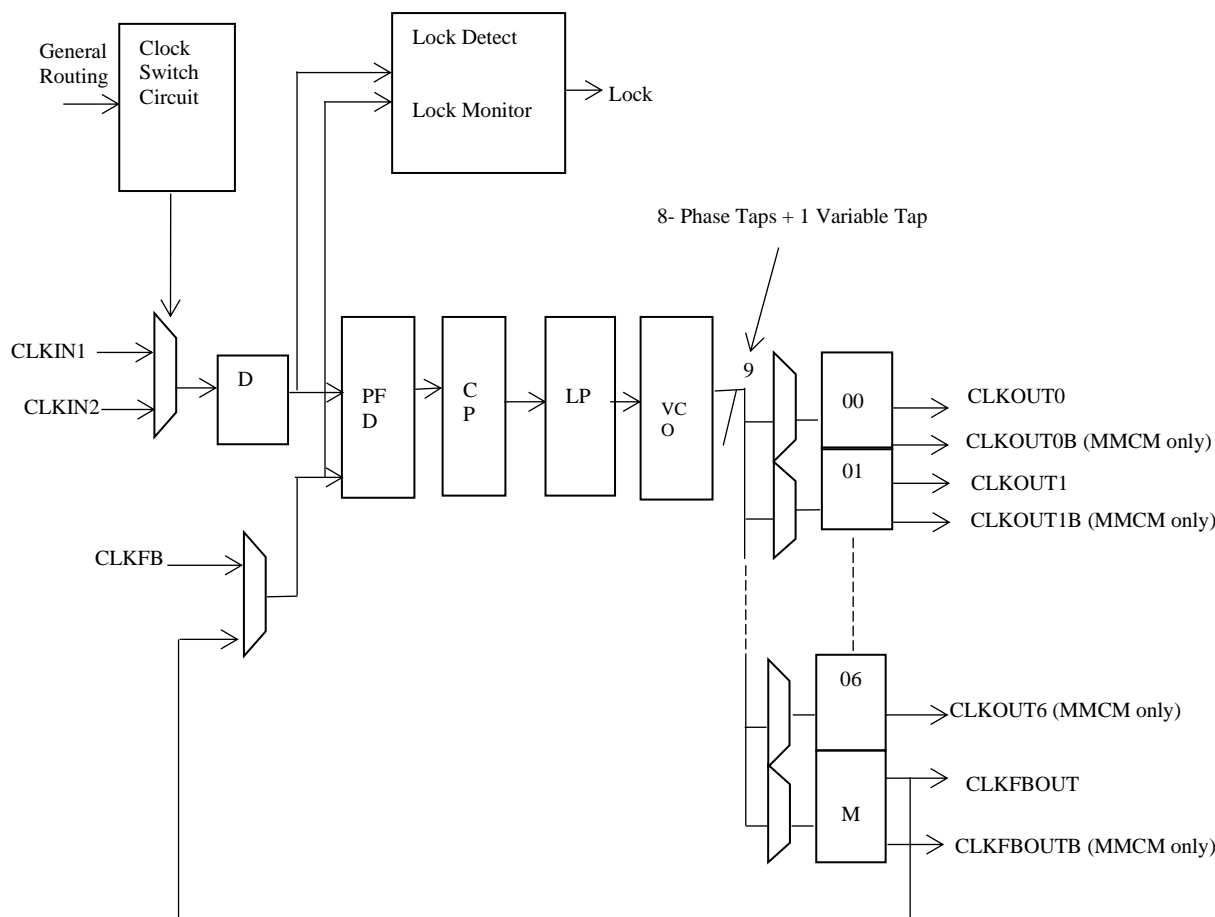


Fig. 1 PLL block diagram

The pulses are filtered by the LF and applied as the control voltage to the VCO. The VCO changes its oscillation frequency according to the control voltage [6]. This PLL has both analogue and digital blocks. The problems in PLL are categorised according to places of radiation strike.

In the digital parts, the divider and PFD experience SEE regarding data flip caused by a radiation strike. This data flip propagated through the LF and reaches the PLL output. The radiation strike in the digital part changes the clock output gradually. In contrast, if the radiation strike in the analogue part, it changes the period immediately and disturbs the output clock, subsequently resulting in the loss of the lock [7].

4. Proposed Solution

FDIR [8,10] is a fundamental concept for any fault mitigation technique. Our approach is also based on the same with the use of one or more redundant PLL in parallel to the primary PLL. Keeping the same configuration for all the PLLs, whenever an error occurs in any PLL, PLL gets isolated, and output is switched to the healthy PLL. An auto-reset is issued to the affected PLL. The requirements of the proposed method are:

- Detection of fault in PLL - The PLL health needs to be monitored constantly with immediate detection of any abnormality.

- Designing of switching matrix - The switching matrix is a decision-making circuit to switch the source PLL on when required and generates the selection input for the clock multiplexer.
- Clock Multiplexer - A glitch-free clock multiplexer, which should be able to alter the source PLL without causing any glitch or delay in the output clock.
- Switching Alert - An alert pulse of finite duration indicating clock switching should be issued to the main logic. This will help the designers decide (if required) based on the applications.
- Output switch count and Automated PLL reset - Counters which steps up every time when PLL is reset or switched over.

The major challenge in this proposal is to achieve seamless switching between clocks without any gap, jitter or variation in the clock period.

Fig.2 and 3 depict the block diagram of dual and triple redundant PLL systems. Fault detection in PLL and indication to switching logic is carried out by error detection block. This block also issues reset to the faulty PLL. The switching network switches the PLL output if required. The glitch-free clock multiplexer is used for clock switching. The detail of each block is given below:

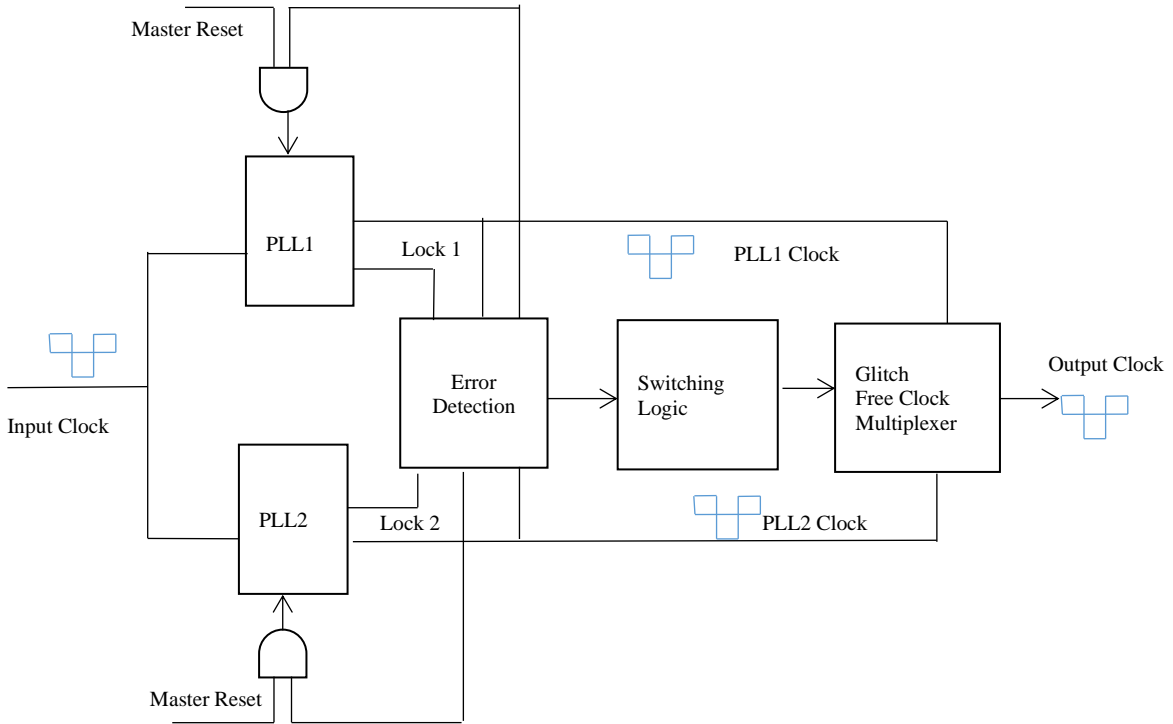


Fig. 2 Dual redundant PLL block diagram

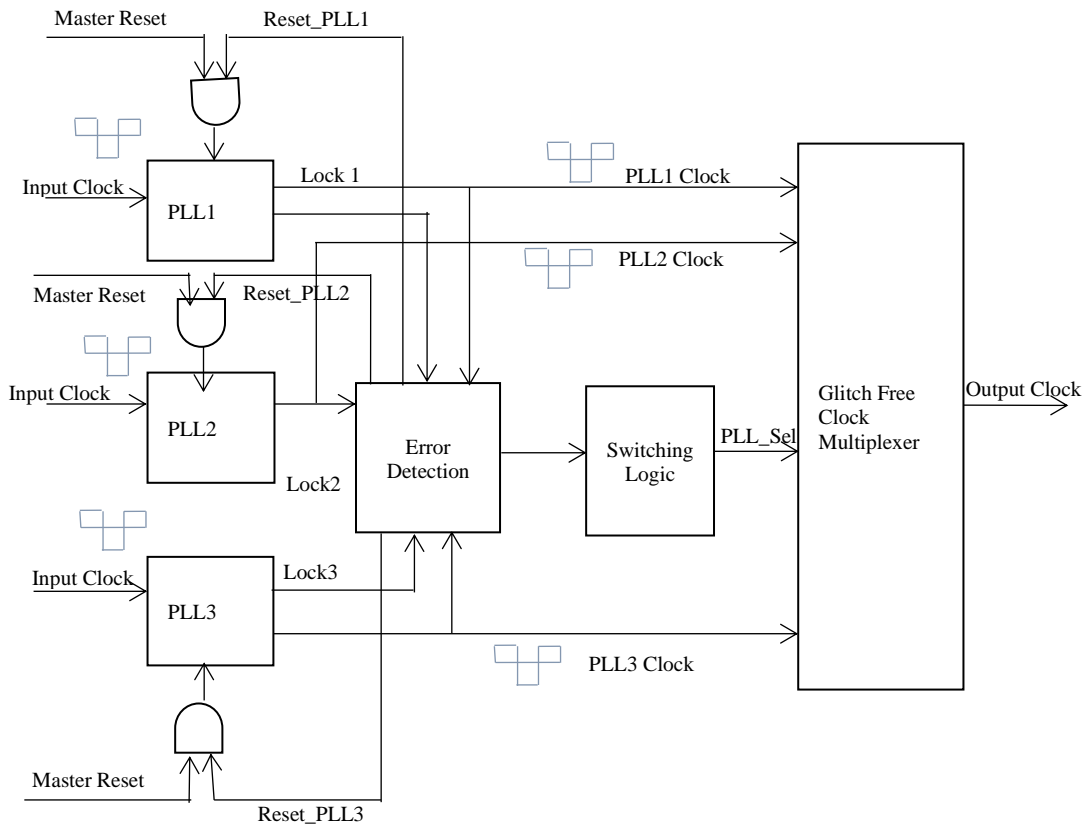


Fig. 3 Triple redundant PLL block diagram

4.1. Error Detection Circuit

Single Event Upset(SEU) in digital or transient in the analogue block of the PLL may change the configuration register value, introduce glitches in VCO output etc., causing the disturbance in the output clock in terms of its frequency or phase. This disturbance in the output clock

results in the loss of the lock. Hence, the fault in the PLL can be detected either by continuously examining the output clock or the lock signal of the PLL. In our proposal, we evaluated both methods. A brief description of each method is given below:

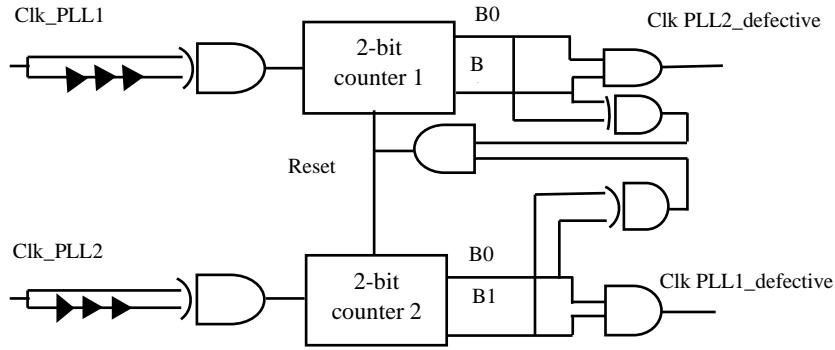


Fig. 4 Clock monitoring circuit

4.1.1. Output Clock Monitoring

Fig. 4 shows the clock monitoring circuit [11]. The elements of this circuit are clock edge detection, a 2-bit counter and a reset generation unit.

Clock-edge detection circuit generates a pulse at every edge of the input PLL clock. These pulses drive the 2-bit counter. In normal conditions, these counters reset to the initial condition within the half cycle of PLL clocks. In case of failure of one clock, the other counter would reach its maximum value, and the respective defective clock signal goes high. For example, when Clk-PLL1 and Clk-PLL2 are ON, and their frequency and phase differences are within the limit, no counter would ever reach its maximum value, keeping both the clock defective signal low. In case of failure of Clk-PLL1, Counter1 will be stuck at its 00 value and reset for counters will not be generated. Within three edges of Clk-PLL2, Counter2 would reach its maximum value, and Clk-PLL1 defective signal will be high, as shown in Fig.5. The pros and cons of this circuit are -

Pros:

- Fast fault detection: It can detect the fault within 1.5 cycles of the input clock.
- Higher phase tolerance: It can tolerate any phase difference from 0° to 360° between both clocks.

Cons:

- False detection of faulty clock: If one of the input clock frequencies increases above 2.5 times the desired frequency, the circuit will malfunction and declare the good clock is faulty, as shown in Fig.6.
- Non-detection of frequency variation: If one of the input clock frequencies varies within 2.5 times the desired frequency, the circuit does not detect the faulty clock.

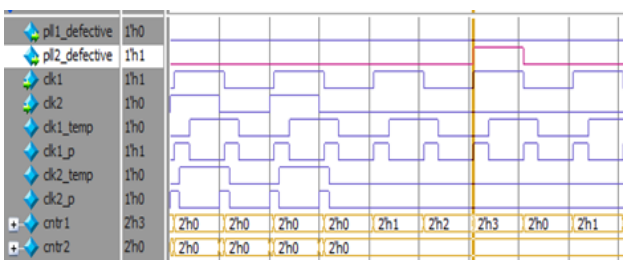


Fig. 5 Clock-based fault detection

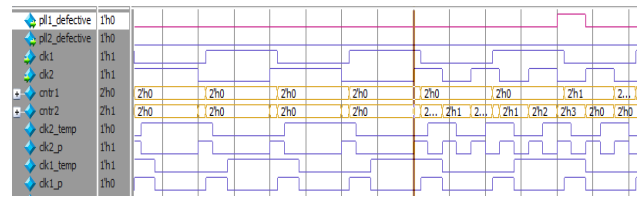


Fig. 6 False detection of faulty clock

4.1.2. Lock-based Fault Detection

The lock signal of PLL is the primary indication of stable clock outputs. The high lock value indicates that output clocks are stable and per the desired configuration. The low lock value defines that output clocks are not stable and unusable [4]. Hence, monitoring the lock signal is the fastest way to detect a fault in PLL. The lock signals from PLLs are monitored, and disturbance in any lock signal is reported to switching and reset generation logic. Implementation of lock monitoring is pure combinational logic. However, to avoid false switching due to glitches in the lock, the lock can be registered on the master clock and should be used for fault monitoring. The combinational logic-based detection is asynchronous but immediate, quickly indicating switching logic about the faulty PLL. However, any glitch in the lock would cause false switching. Sequential lock monitoring will not result in a false switch but with the penalty of delay in fault detection by 2-4 cycles of the master clock.

4.2. Switching Logic

The complexities of the switching logic vary with redundant PLL numbers. In this paper, dual and triple PLL redundancies are considered. Dual PLL switches to the healthier PLL based on fault detection. In triple modular without majority voting logic, switching takes place to the healthier PLL based on the fault as per a predefined sequence. In contrast, majority voting switches the output to any of the two working PLL. Here, two healthy PLL are always required to switch the output. The description of dual and triple modular switching logic is given below.

4.2.1. Dual PLL Switching Logic

The state diagram for dual PLL-based switching logic and truth table is shown in Fig.7. Here, the selection of PLL depends on the lock status of both the PLLs and current selected PLL, ensuring minimum and required switching. When both the PLLs are out of the lock, the output clock

will switch to PLL0. When only one PLL is in a locked condition, the output will shift to the locked PLL. When both the PLL moves in locked condition from one PLL locked condition (01 > 11 or 10 > 11), the next state would remain the same as the current state. The solved truth table results in the following Boolean equation:

$$Sel = Sel \cdot Lock1 + \overline{Lock2} \cdot Lock1 \tag{1}$$

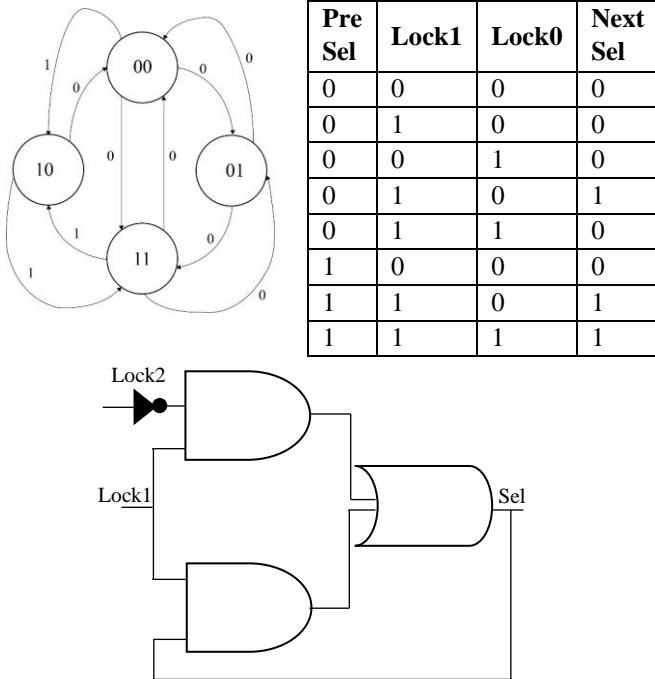


Fig. 7 Dual PLL switching logic

This circuit forms a combinational feedback loop, but there would not be any glitch generation as per the design. However, the same can be implemented with a sequential Finite State Machine (FSM), with or without registering asynchronous locks. Here, switching would consume one or more cycles based on the implementation.

4.2.2. Triple PLL Switching Logic

The use of triple PLL increases the total available time and reliable selection of the healthy PLL. At the same time, it uses more real estate and reduces the timing performance. Triple PLL can be used in two ways - with or without majority logic.

Triple PLL with Majority Voting

In this mode, the output clocks would be available when at least two PLLs are in the lock or healthy condition. This enhances the confidence of the selected PLL. Similar to dual PLL, the selection of PLL depends on the lock of all the PLLs and the current selected PLL, guaranteeing minimum and required switching. When no or only one PLL is in the lock or healthy condition, no clock would be at the output. When any two PLLs are ON, then the selection of PLL would be made as per the sequence, ensuring no unwanted switching. When all PLLs are ON, then no switching will take place; the current selected PLL will continue. Similarly, no switching will occur when all PLL locked conditions transition to any two PLL lock conditions and if the current selected PLL is still healthy and locked.

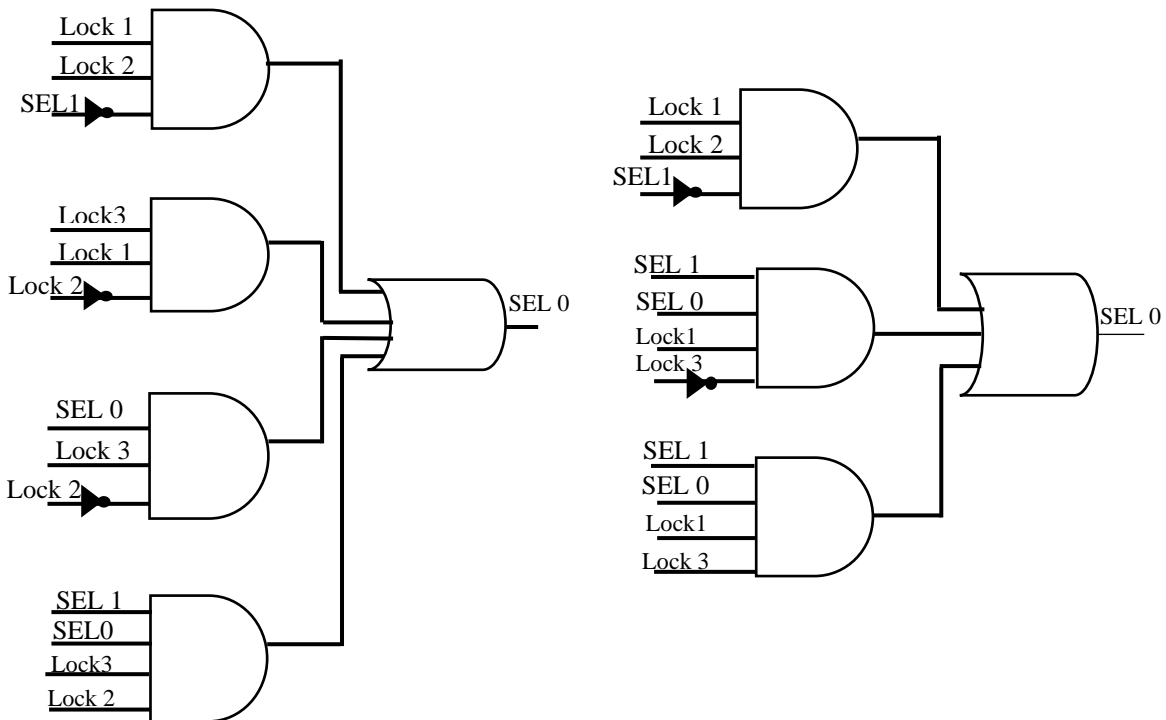


Fig. 8 Triple PLL with majority voting - switching logic

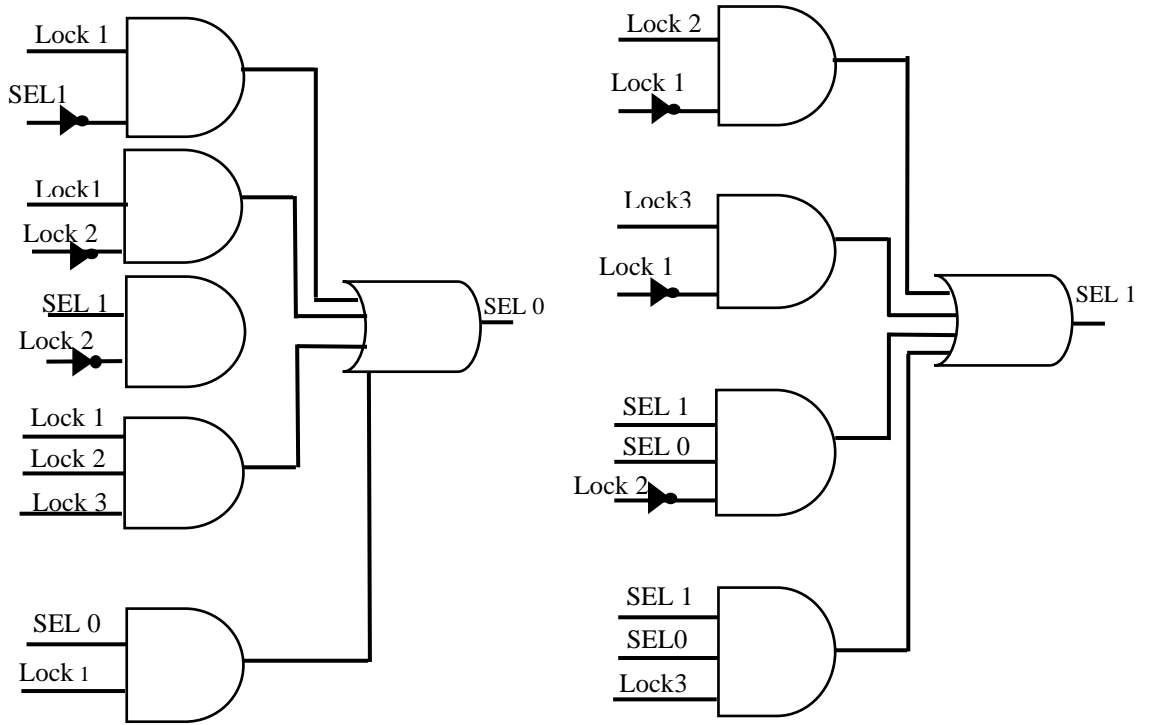


Fig. 9 Triple PLL without majority voting -Switching logic

The 2-bit PLL selection lines are derived from a complex state diagram following the proper switching sequence and avoiding unwanted switching. The truth table with 5 inputs, 2-bits current PLL selection and 3 bits PLL lock is solved, resulting in Boolean equations (2) and (3). The equivalent circuit diagram is shown in Fig.8.

$$Sel0 = \overline{Sel1}.Lock2.Lock1 + Lock3.\overline{Lock2}.Lock1 + Sel0.Lock2.Lock1 + Sel1.Sel0.Lock3.Lock1 \quad (2)$$

$$Sel1 = Lock3.lock2.\overline{lock1} + Sel1.\overline{Sel0}.Lock2.Lock1 + Sel1.Sel0.Lock3.lock2 \quad (3)$$

Triple PLL without Majority Voting

This concept is to increase PLL clock availability time for the design, which is not reconfigured or restarted often. Here, the output clock would be available even if one PLL is healthy and in locked condition. Here also, switching of PLL depends on the lock conditions of all the PLLs and the current selected PLL. Unlike majority voting, the output clocks would be available even if a single PLL is in a locked state. When any one, two or all three PLL are healthy, PLL would be selected as specified in a predefined sequence to ensure minimal and no false switching. The currently selected output would not change until the selected output PLL gets unlocked. The state diagram and truth table are similar to the earlier one, with five inputs (2-bits current PLL selection and 3 PLL lock) and two outputs (next PLL selections). The solved truth table resulted in Boolean equations (4) and (5). The equivalent circuit diagram is shown in Fig.9

$$Sel0 = \overline{Sel1}.Lock1 + \overline{Lock2}.Lock1 + Sel1.Sel0.Lock3 + lock1.lock2 + \overline{Lock2}.Lock3 \quad (4)$$

$$Sel1 = Lock2.\overline{Lock1} + Lock3.\overline{Lock1} + Sel1.\overline{Sel0}.Lock2 + Sel1.Sel0.Lock3 \quad (5)$$

4.3. Clock Multiplexer

As discussed earlier, a clock multiplexer will switch the output clock to a different PLL based on the inputs from the switching matrix. To achieve clock multiplexing following designs are evaluated.

4.3.1. Combinational Multiplexer

The simplest clock switch is the multiplexer circuit, as shown in Fig.10. This clock multiplexer is for two inputs and one selection line. It takes two clock signal sources at inputs (signals CLK0 and CLK1), and a PLL selection SELECT. When the SELECT value changes, the multiplexer alters the clock source input at the output. Here, clock switching is immediate, but it may generate a chopped clock signal or a glitch at the output clock if both the input clocks are out of phase, as shown in Fig.11.

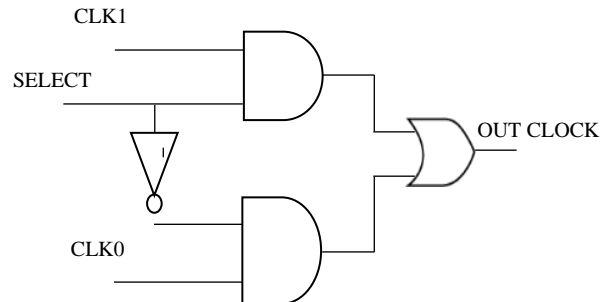


Fig. 10 Clock multiplexer

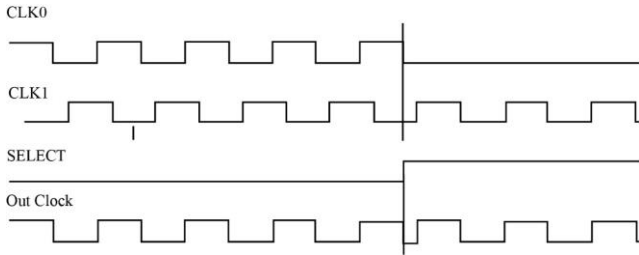


Fig. 11 Glitch at output clock

4.3.2. Glitch Free Clock Multiplexer

A clock switch circuit [11] that prevents glitch generation at the output is presented in Fig.12. This circuit suppresses the glitch at the output. However, a gap of 2 -3 clock cycles is observed at switching, as per Fig.12.

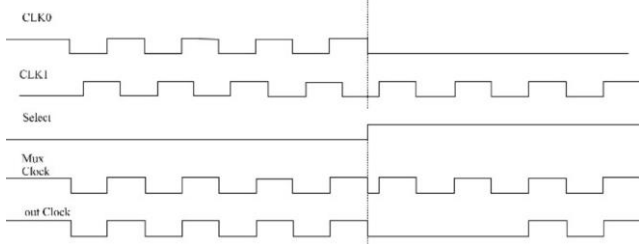


Fig. 12 Glitch-free output clock

4.3.3. Clock Buffer Multiplexer

Various FPGAs provide clock multiplexer primitive. These are designed to multiplex the clocks with minimum skew and the possibility of a glitch in output. The BUFG MUX primitive [13] from Xilinx is shown in Fig.13.

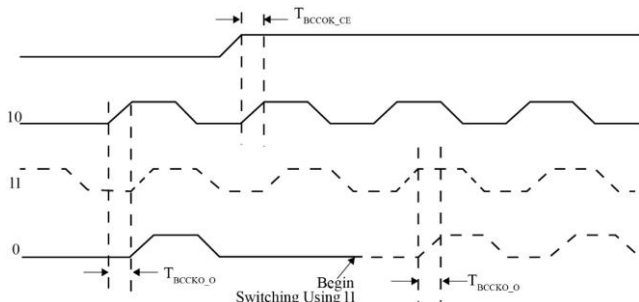
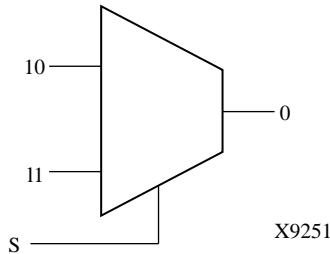


Fig. 13 Clock buffer multiplexer

The timing diagram in Fig.13 shows that the current clock is I0, and S is activated high. If I0 is currently high, the multiplexer waits for I0 to de-assert low. After that, the multiplexer output stays low until I1 transitions from high to low. Subsequent to this, the output switches to I1. If setup/hold times are met, no glitches or short pulses can appear on the output. However, if both the input clocks

having the same frequency are not phase synchronised, then there may be a glitch in output.

5. Implementation

A configurable IP core is developed for dual and triple PLL with and without majority voting logic. The core employs PLL locks for PLL fault detection. Locks are registered and monitored to avoid any glitches in fault detection. The switching matrix is implemented in sequential and combinational logic in different core versions. At last glitch-free clock-multiplexer and vendor-specific clock-buffer multiplexer's primitives are used in different core versions to achieve clock multiplexing. Auto reset to faulty PLL and PLL switch counter have been implemented in all configurations. A clock-switching alert flag of a specific duration is generated in all the configurations. Table I describes different configurations implemented and tested.

Table 1. Different configurations implemented and tested

Configuration	Details
Config-1	Dual PLL + Lock based Fault detection + Clock multiplexer with clock buffer
Config-2	Dual PLL + Lock based Fault detection + Glitch free clock switching
Config-3	Dual PLL + Lock based Fault detection + Clock buffer Multiplexer primitive
Config-4	Triple PLL without majority voting logic + lock-based fault detection + glitch-free clock switching
Config-5	Triple PLL with majority voting logic + lock-based fault detection + glitch-free clock switching

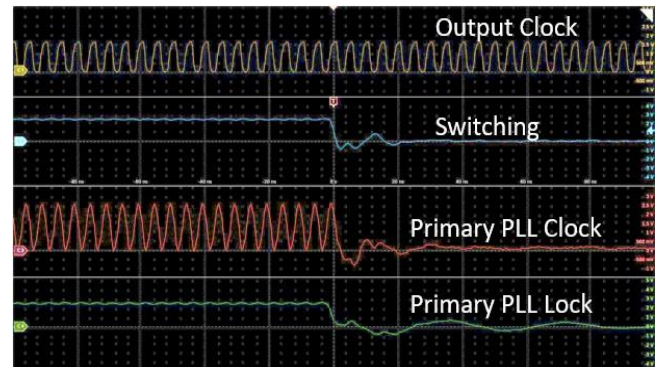


Fig. 14 Seamless switching

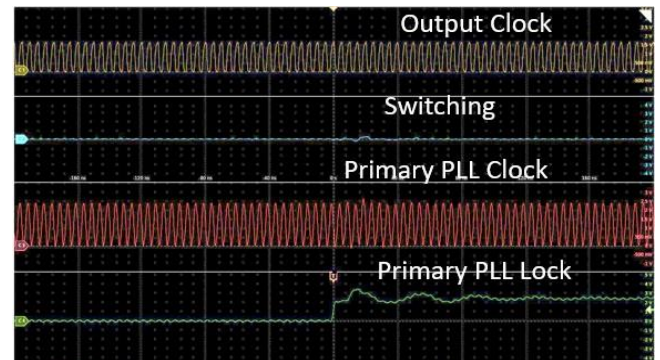


Fig. 15 Re-storing of faulty PLL

5.1. Results

Fig.14 depicts the seamless switching of PLL clocks for the Config-1 in the occurrence of a fault in the existing PLL. It also issues reset to the faulty PLL to regain its lock as per Fig.15.

Fig.16, describes the switching of PLL clocks when multiple PLL outputs are used for Config-1. Here, a glitch in switching is observed for one of the PLL outputs (Output Clock3).

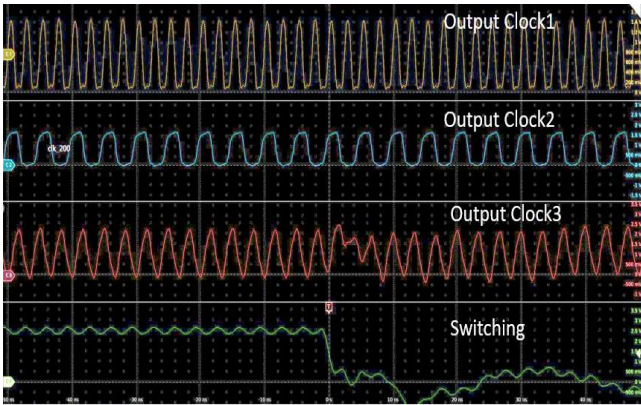


Fig. 16 Glitch in output clock

Fig.17 and Fig.18 show the result of Config-2 and Config-3, respectively, where one PLL output clock cycle gap was observed. No glitches are seen at the output while switching.



Fig. 17 Config-2: Glitch free switching

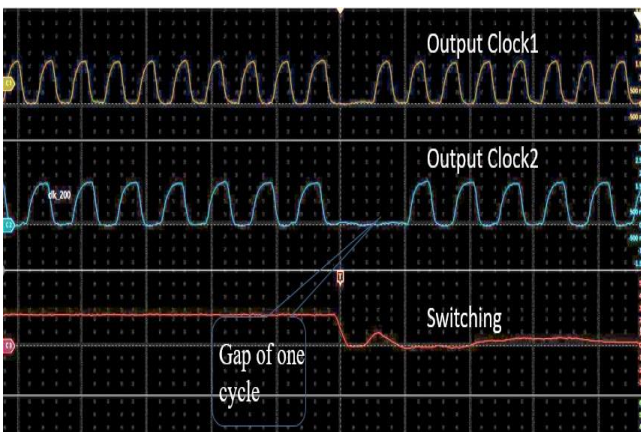


Fig. 18 Config-3: Glitch free switching

The results of different configurations of triple PLL are given in Fig.19, 20 and 21. Glitch-free switching of the PLL output is shown in Fig.21. Avoidance of false switching and auto reset with lock acquisition is depicted in Fig.20 and 21.

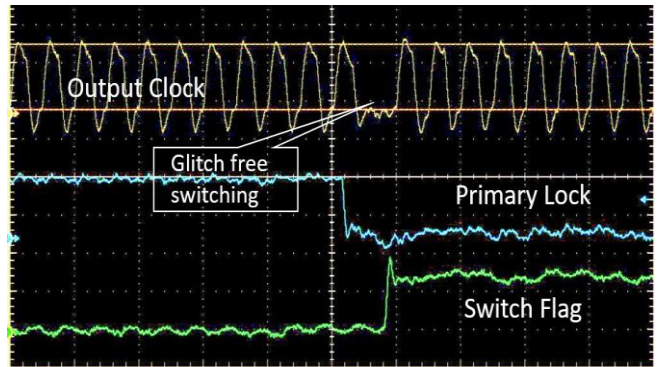


Fig. 19 Config-4: Glitch free switching

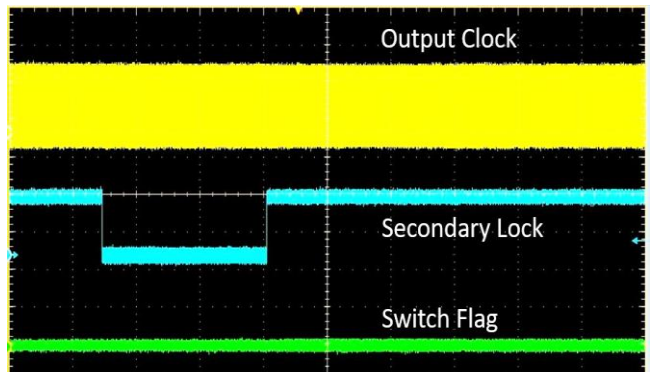


Fig. 20 Config-4: Minimal switching

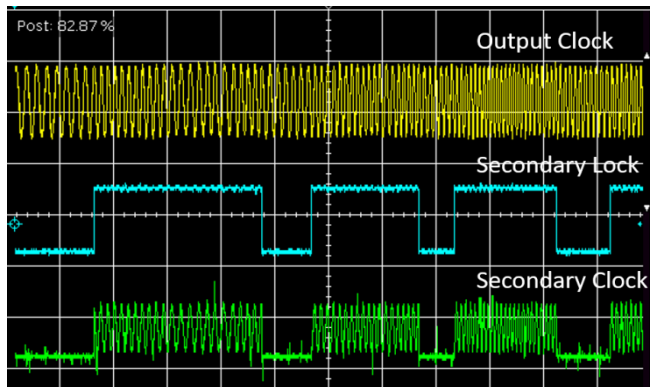


Fig. 21 Config-4: Auto lock re-acquisition

Summary of results for all the configurations-

1. Fault detection: Lock-based detection is easier and faster with respect to clock-based fault detection. Combinational logic for fault detection is seamless but may have false detection due to a glitch in the lock signal.
2. Switching Logic: Combinational logic-based switching is faster than sequential as it causes an additional gap of one or more cycles. The false switching due to the combinational feedback loop is well-taken care of while designing the circuit.
3. Clock Multiplexer: Glitch-free clock multiplexer has reliable performance; however, it introduces a gap of two clock cycles. The seamless switching with a

combinational multiplexer may generate a glitch, which results in a timing violation or meta-stability in the circuit.

4. Dual and Triple redundancy: Dual redundancy is the simplest FDIR scheme without any overhead of the voting circuit. The majority voting logic will ensure that only output clocks would be available when two or more PLL is working. The uncertainty in the lock status due to input clock variation can be overcome with majority voting logic. Majority voting does not offer much advantage in terms of the mean availability time, even in case of false fault indication; the respective PLL would be isolated. However, without majority voting logic, the MTTF increases exponentially.

Based on the above evaluation, the developed IP core has the following features

- Configurable in three modes using compiler directives - dual, triple PLL with and without majority logic.
- lock-based fault detection and glitch-free clock multiplexing.
- Auto recovery of faulty PLL.
- Command-based redundancy bypass and PLL selection.
- Switching alert flag and fault counter for health analysis.
- Optimum usage of clock buffer resources using compiler directives per the design required clock outputs.

5.2. Resources Utilisation

Table II describes sample FPGA resource overhead in dual and Triple PLL redundant modes. Triple PLL with and without a majority logic has similar resource utilisation except for minor variations in LUT. The resource overhead is minimal, typically < 0.5% of any modern device.

Table 2. FPGA resource utilization

Configuration	LUT	FF	BUFG	PLL
Xilinx – Virtex 7 & Zynq				
Dual-PLL	90	76	21	2
Triple-PLL with majority logic	121	103	28	3
Xilinx – Virtex 5				
Dual-PLL	244	269	21	2
Triple-PLL with majority logic	347	387	28	3

6. Reliability Analysis

The classical reliability models are used as a standard metric for complex system performance. The analysis provides a more in-depth interpretation of system behaviour over time using system-level MTTF or Failure Rate (λ) data for system performance metrics. The reliability equation of the system [15]

$$R(t) = e^{-t/MTTF} \quad (6)$$

or

$$R(t) = e^{-\gamma t} \quad (7)$$

Reliability for M of N system (N redundant modules, out of which M are required for majority voting) [16]

$$R(t)_{MofN} = R(t)_{vot} \sum_{i=M}^N \binom{N}{i} R(t)^i (1 - R(t))^{N-i} \quad (8)$$

Where

$R(t)_{vot}$ = reliability of the voting system

$$\binom{N}{i} = N! / i! (N - i)!$$

Reliability of the TMR system with majority voting logic

$$R(t)_{3of2} = R(t)_{vot} (3R(t)^2 - 2R(t)^3) \quad (9)$$

The reliability of dual and triple redundant parallel systems with the assumption of reliability of fault detection logic is 1

$$R(t)_{dmr} = 1 - (1 - R(t))^2 \quad (10)$$

$$R(t)_{tmr} = 1 - (1 - R(t))^3 \quad (11)$$

Classical reliability models are measured across time. This is because most of the failures that can affect performance in classical studies are due to wear-out mechanisms or corner-case design bugs. For each case, time to failure is a key measurement factor. While evaluating SEU susceptibility, during radiation testing, particle fluence is the key variable for system failure as opposed to time. Missions required to operate in space environments will be susceptible to the fluence of ionising particles. As a metric of SEU susceptibility, σ_{SEU} is calculated across fluence [17,19]. The mapping of the model is illustrated in Table III.

Table 3. Mapping of classical reliability equation to reliability under radiation upset

Classical Reliability Model	Reliability Under Radiation
Classical Reliability Calculation	SEU-based reliability calculation
Disregard Infant Mortality and Wear out	Disregard Infant Mortality
Failures are random	SEU are random
Error rate is constant.	SEU rate is constant (σ_{SEU})
Time(t)	Flux(ϕ)
Mean Time To Failure = $1/\lambda$	Mean Fluence To Failure(MFTF) = $(1/\phi)$
$R(t) = e^{(-t/MTTF)}$	$R(\phi) = e^{(-\phi/MFTF)}$

The reliability over fluence $R(\phi)$ is

$$R(\phi) = e^{(-\phi/MFTF)} \quad (12)$$

Similarly, equations (9), (10) and (11) will be translated as

$$R(\phi)_{3of2} = R(\phi)_{vot} (3R(\phi)^2 - 2R(\phi)^3) \quad (13)$$

$$R(\phi)_{dmr} = 1 - (1 - R(\phi))^2 \quad (14)$$

$$R(\phi)_{tmr} = 1 - (1 - R(\phi))^3 \quad (15)$$

The upsets per day are determined as follows.

$$\phi = \ln R(\phi) * MFTF * (-1) \quad (16)$$

$$\text{Upsets per day} = \phi / \text{flux} \quad (17)$$

Case Study: A detailed study on the Xilinx ultra-scale device is carried out as part of a case study of improvement in the PLL upset rate and corresponding reliability numbers of PLL under a radiation environment with multi-level redundancy. The different parameters of the device [12] are given in Table IV

Table 4. Device parameters

Parameter	Value
PLL Cross Section σ_{SEU} at $LET_{th} - 1\text{MeV.cm}^2/\text{mg}$	10^{-5}
Geosynchronous Earth Orbit-Flux (Particles@ LET_{th})	90 particles/hour
PLL upset/day	1/46 (one upset in 46 days)

Improvement in the parameters for a different level of redundancy for a continuous operation with reconfiguration once in 24 hours, as given in Table V.

Table 5. Reliability case study results

Parameter	Value
Reliability of Single PLL (12)	$R(\phi)_{single} = e^{-(24*90)/10^{-5}} = 0.9786316$
Dual Redundant PLL	
Reliability of Dual redundant PLL (14)	$R(\phi)_{dual} = (1 - (1 - R(\phi)_{single})^2) = 0.999543$
Upsets per day (16) (17)	4.57×10^{-4} (one upset every 2188 days)
Triple Redundant PLL	
Reliability of Triple redundant PLL (15)	$R(\phi)_{triple} = (1 - (1 - R(\phi)_{single})^3) = 0.999990$
Upsets per day (16) (17)	9.75×10^{-6} (one upset in every 102485 days)
Triple Redundant PLL with majority voting	
Reliability of Triple redundant PLL (13)	$R(\phi)_{tmr} = 3 \cdot (R(\phi)_{single})^2 - 2 \cdot (R(\phi)_{single})^3 = 0.998649$
Upsets per day (16) (17)	1.351×10^{-3} (one upset every 740 days)

The case study reveals that different level of redundancy in PLL improves the MTTF, as shown in

Fig.22. The numbers obtained are in line with the requirements for various long-term satellite missions for different earth orbits.

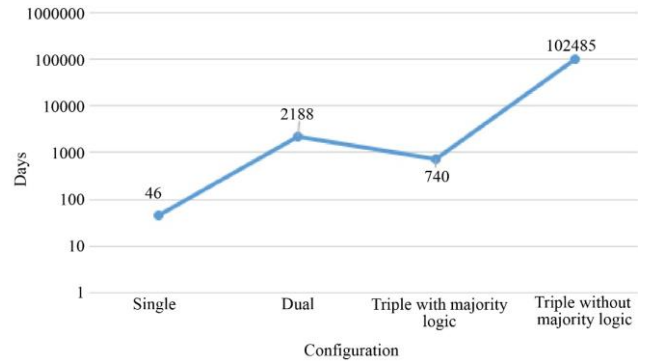


Fig. 22 MTTF for different configuration

7. Conclusion

A configurable FDIR IP core was developed and tested to identify the PLL or clock manager module's fault, isolation and recovery. As the developed core is technology independent, it can be used in any FPGA family. Their reliability numbers demonstrated the use of dual and triple redundant PLL with and without majority voting. The fault detection, switching logic and clock multiplexer are key components of the core, which determine reliable & quick identification and switching of the PLL outputs. Auto fault detection in the PLL and glitch-free switching prevent data or functionality loss due to malfunction of the PLL in a radiation environment. The radiation and reliability parameters clearly show that the redundancy in the PLL improves the radiation performance by a significant number, which is equivalent to the radiation-hardened device. The technology-independent IP core can be used in all the FPGAs with PLL/DCM of spacecraft. The resource overhead and performance impact are minimal for any latest FPGA device. In addition to space, the core can be useful in other applications where radiation is a major challenge, such as medical and atomic research. For future work, a demonstration of the effectiveness of the IP core under radiation test is planned. Furthermore, schemes will be developed to synchronise all the PLL for seamless switching.

Acknowledgement

The work presented in this paper reflects the interdependent efforts of multiple teams of the Space Applications Centre (SAC). The authors wish to acknowledge the contributions of various design and verification groups in successfully completing this work. We are grateful for the insightful comments offered by the reviewers to improve the quality of the paper. We would also like to express our sincere gratitude to Director, SAC, for providing this opportunity.

References

- [1] Fares Fourati, and Mohamed-Slim Alouini, "Artificial Intelligence for Satellite Communication: A Review," *Intelligent and Converged Networks*, vol. 2, no. 3, pp. 213-243, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Shruti Edway, and R K Manjunath, "Design and Simulation of FPGA Based All Digital Phase Locked Loop (ADPLL)," *3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pp. 259-263, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Frank Hall Schmidt, "Fault Tolerant Design Implementation on Radiation Hardened By Design SRAM-Based FPGAs," Massachusetts Institute of Technology, 2013. [[Google Scholar](#)] [[Publisher Link](#)]
- [4] User Guide:Virtex-5 FPGA UG190 (v3.2), 2007.
- [5] Jim Tatsukawa, "MMCM and PLL Dynamic Re-Configuration," Application Note: 7 Series, UltraScale and UltraScale+FPGAs, XAPP888 (v1.8), 2019. [[Google Scholar](#)]
- [6] Sinnyoung Kim, Akira Tsuchiya, and Hidetoshi Onodera, "Perturbation-Immune Radiation-Hardened PLL with a Switchable DMR Structure," *IEEE 19th International On-Line Testing Symposium(IOLTS)*, 2013. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Sinnyoung Kim, Akira Tsuchiya, and Hidetoshi Onodera, "Dual- PLL based on Temporal Redundancy for Radiation-Hardening," *JAXA Special Publication*, 2012. [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Fatemeh SalarKaleji, and Aboulfazl Dayyani, "A Survey on Fault Detection, Isolation and Recovery (FDIR) Module in Satellite Onboard Software," *6th International Conference on Recent Advances in Space Technologies (RAST)*, pp. 545-548, 2013. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] Ms.P.Thamarai, and Mr.B.karthik, "Network Data Security Using FPGA," *International Journal of P2P Network Trends and Technology*, vol. 2, no. 5, pp. 24-27, 2012. [[Publisher Link](#)]
- [10] Felix Siegle, "Fault Detection, Isolation and Recovery Schemes for Spaceborne Reconfigurable FPGA-Based Systems," University of Leicester, 2016. [[Google Scholar](#)]
- [11] Gregg Starr, and Edward Aung, "Clock Loss Detection and Switchover Circuit," United States US7427881B2, 2002.
- [12] Lidawei Zhujianzhang, Wangqia Ng, and Wangpanfeng Zoulna, "Dynamic Switching Circuit for Clock," China CN202171760U, 2011.
- [13] Product Guide: Radiation Tolerant Kintex Ultrascale XQRKU060 FPGA Data Sheet AMD Xilinx DS882, 2022. [[Publisher Link](#)]
- [14] Vijayakumara Y M et al., "A VLSI Implementation of Hamming Code Algorithm Using Fpga Architecture," *SSRG International Journal of VLSI & Signal Processing*, vol. 7, no. 2, pp. 29-35, 2020. [[CrossRef](#)] [[Publisher Link](#)]
- [15] Robert Glein et al., "Reliability of space- Grade vs. COTS SRAM-based FPGA in N-Modular Redundancy," *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2015. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Krishna Karan, and Morgan Kafman, Fault Tolerant Systems. [Online]. Available: <http://www.ecs.umass.edu/ece/korean/FaultTolerantSystems>
- [17] M.Berg et al., "Using Classical Reliability Models and Single Event Upset(SEU) Data to Determine Optimum Implementation Schemes for Triple Modular Redundancy in SRAM-Based Field Programmable Gate Array (FPGA) Devices," *IEEE Nuclear and Space Radiation Effects Conference*, 2015. [[Google Scholar](#)]
- [18] M. R. Ezilarasan, and J. Brittopari, "An Efficient FPGA-Based Adaptive Filter for ICA Implementation in Adaptive Noise Cancellation," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 10, no. 1, pp. 117-127, 2023. [[CrossRef](#)] [[Publisher Link](#)]
- [19] M.Berg,Kenneth LaBel, Michael Campola, and Michael Xapsos, "Analyzing Test-as-You-Fly Single Event Upset(SEU) Response using SEU Data, Classical Reliability Models and Space Environment Data," *Government Microcircuit Applications and Critical Technology Conference (GOMAC)*, 2017. [[Google Scholar](#)]
- [20] Andrés Pérez-Celis, Corbin Thurlow, and Michael Wirthlin, "Emulating Radiation-Induced Multicell Upset Patterns in SRAM FPGAs With Fault Injection," *IEEE Transactions on Nuclear Science*, vol. 68, no. 8, pp. 1594-1599, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Antonis Tsiganos et al., "High- Performance COTS FPGA SoC for Parallel Hyperspectral Image Compression with CCSDS-123.0-B-1," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 11, pp. 2397-2409, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Saichandrateja Radhapuram, Takuya Yoshihara, and Toshimasa Matsuoka "Design and Emulation of All-Digital Phase-Locked Loop on FPGA," *Electronics*, vol. 8, p. 1307, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Ms.Manjula B.M, and Dr.Chirag Sharma, "FPGA Implementation of BCG Signal Filtering Scheme by using Weight Update Process," *SSRG International Journal of VLSI & Signal Processing*, vol. 3, no. 3, pp. 1-7, 2016. [[CrossRef](#)] [[Publisher Link](#)]
- [24] Hieu-Truong Ngo, and Quoc-Son Tran, "A New Design of Phase-Locked Loop with Multiple Frequencies for Communication Standard," *International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 1085-1089, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [25] R. Katz et al., "Radiation Effects on Current Field Programmable Technologies," *IEEE Transactions on Nuclear Science*, vol. 44, no. 6, pp. 1945-1956, 1997. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [26] G. Tsiligiannis et al., "Radiation Effects on Deep Submicrometer SRAM-Based FPGAs Under the CERN Mixed-Field Radiation Environment," *IEEE Transactions on Nuclear Science*, vol. 65, no. 8, pp. 1511-1518, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [27] Connor R. Julien, Brock J. LaMeres, and Raymond J. Weber, "An FPGA-based Radiation Tolerant SmallSat Computer System," *IEEE Aerospace Conference*, Big Sky, MT, USA, pp. 1-13, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [28] W. K. Victor, H. L. Richter, and J. P. Eyraud, "Explorer Satellite Electronics," *IRE Transactions on Military Electronics*, vol. MIL-4, no. 2/3, pp. 78-85, 1960. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]