Design of Windowed Watchdog Timer for Embedded Systems Faulty Timing Control

Bopparam Manasa[1], Sithakka Shruthi [2], Soppari Snehith [3], Mrs. K. Swapna [4]
[123] UG Scholar, Dept. of ECE, St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
[4]Assistant Professor, Dept. of ECE, St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
bopparammanasa@gmail.com

Abstract:

Embedded systems are at the core of modern electronics, with industries reporting a 25% annual increase in their adoption due to advancements in automation, control, and communication technologies. In such systems, timing faults can lead to critical failures, affecting the reliability of automotive, medical, and aerospace applications. Existing 2x2 Watchdog Timers (WDT) are inefficient at handling complex timing faults, leading to up to 30% system downtime due to undetected failures. Current 2x2 WDT controllers suffer from limited fault detection capability and poor timing resolution, which can result in undetected errors in real-time embedded systems. This paper introduces a novel Windowed Watchdog Timer (WDT) design that enhances fault detection operation by monitoring both early and late faults within a specified window. The improved WDT interface provides seamless integration with microcontrollers, enhancing timing precision and system reliability. This design addresses timing faults more effectively, leading to a more robust embedded system Embedded systems are increasingly used across industries, with a 25% annual growth due to technological advancements. Timing faults can lead to serious issues, particularly in automotive, medical, and aerospace sectors, causing up to 30% downtime. Current 2x2 Watchdog Timers (WDT) struggle to detect complex timing faults effectively We have demonstrated a Windowed Watchdog Timer that enhances fault detection by monitoring faults within a specified time window. The improved WDT integrates easily with microcontrollers, enhancing timing precision and overall system reliability by using simulation This project endeavours to explore and propose a Vlsi simulation approach demonstrate windowed watchdog timer to decrease timing fault detection and reduces timing faults in particularly in medical aerospace automotive applications and it also enhance system reliability while increasing efficiency and it can also easily integrate with microcontroller. It decreases the timing faults with improved efficiency better than the current available 2X2 watchdog timer.

Keywords: Windowed watchdog timer, Watchdog timer, microcontroller, VLSI based simulation, System reliability, Fault detection, improved efficiency, enhancing reliability, improved applications, timing fault reduction.

1. INTRODUCTION

Watchdog Timer (WDT) is a key component in chip multiprocessors (CMPs) as it supports communication between many cores. However, WDT for modern chips contribute up to 30% of the chips overall power budget. Also, as chips continue to shrink, the contribution of WDT power is projected to increase. Routers. WDT Controller comprises several input ports, a number of output ports, a switching matrix logic blocks. That implements flow control policies The first and most important ones are the links that physically connect the nodes and implement the communication The second block is the router, which implements the communication protocol The last building block is the network adapter (NA) or network interface (NI). This block makes the logical connection between the IP cores and the network On-chip routers are a crucial component in System-on-Chip (SoC) designs, facilitating communication between various components on the chip. Improving latency and throughput in on-chip networks is essential for enhancing the overall performance of integrated circuits. Some strategies and considerations include: Topology Design: Choosing an efficient network topology is critical. Mesh, torus, or hierarchical topologies are commonly used, and the selection depends on the specific requirements of the application. Implementing efficient routing algorithms can significantly impact latency and throughput. Adaptive or dynamic routing algorithms can adapt to changing traffic patterns and optimize communication paths.The efficiency of the WDT's integrated circuit becomes pivotal in determining the overall success of data transmission. If the WDT's integrated circuit operates with optimal efficiency, it ensures perfect reception of data by User 2. However, in the event of inefficiency or malfunction within the WDT's integrated circuit, there is a risk of data loss, potentially resulting in communication failures between User 1 and User 2.

2. LITERATURE SURVEY

Watchdog timers have a rich history rooted inside the evolution of embedded systems and their need for reliability and fault tolerance.[1] The idea of a watchdog timer dates to the mid-20th century when the earliest laptop structures had been becoming extra prevalent. These early systems, regularly large and highly priced, were used for numerous commercial and scientific programs.[2] The preliminary task in these systems became making sure their non-stop operation, specifically in important packages where device disasters ought to lead to tremendous outcomes. The idea of a watchdog timer emerged as a solution to this problem.[3] The idea of a watchdog timer turned into honest at its inception. It involved a timer or counter circuit that needed to be periodically reset with the aid of the software program going for walks on the main processor If the timer changed into now not reset within a predefined time c program language period, it would trigger a reset signal, efficiently rebooting the system. This primary watchdog mechanism provided a protection net towards software program crashes or hardware disasters As microcontrollers and embedded systems became greater generic in various industries.[4] The idea of watchdog timers found its way into these compact and specialized computing gadgets. Microcontroller producers began integrating dedicated hardware watchdog timers into their products, making it easier for developers to implement fault tolerance of their embedded packages. Watchdog timers evolved to grow to be greater configurable and flexible. This era noticed the introduction of programmable watchdog timers that allowed builders to set the timeout period in

keeping with their unique utility necessities.[5] This configurability was important for adapting watchdog timers to a wide variety of embedded structures, from automobile control units to purchaser electronics. The idea of windowed watchdog timers emerged as a response to the need for even finer manipulate over system reliability.[6]A windowed watchdog timer has two windows: a service window and a frame window. The provider window requires the software program to periodically "pet" the watchdog inside a particular time frame, stopping it from triggering a reset. The frame window, alternatively, defines a broader interval wherein the watchdog. Modern windowed watchdog timers continue to adapt with advanced functions and protection upgrades. These timers often consist of mechanisms to save you unintentional change of configuration registers, ensuring that crucial timing parameters continue to be intact [8] Additionally, they provide dynamic initialization alternatives, considering extra sophisticated triggering techniques, which include outside timers or sensors. These innovations make windowed watchdog timers even more adaptable and dependable in a huge variety of applications.[9] Nikiema et al. Proposed a method for enhancing the dependability of RISC-V cores in facet computing devices They centred on robust machine design and testing of their approach The technique worried rigorous checking out and verification tactics to make sure the reliability of RISC-V cores for edge computing applications. One disadvantage of this work turned into the dearth of specific information concerning the checking out methodologies employed. Ahangari et al. [10]provided an structure tailored for protection-critical transportation structures. Their technique aimed to make sure the fault tolerance and reliability of such systems. They integrated superior protection mechanisms into the architecture to cope with the unique challenges posed by transportation structures. However, the paper did no longer delve into the specific implementation demanding situations or ability barriers of their proposed architecture. While this strategy is effective, it increases the complexity of the hardware and the overall cost of the system, despite its effectiveness. The use of a Field Programmable Gate Array (FPGA) to build external watchdogs can reduce both the cost and the complexity associated with their implementation (FPGA). Today's embedded systems rely on FPGA chips to perform critical functions, which are found in a wide range of applications [11]. With the help of an FPGA, it is possible to design a watchdog timer that is both efficient and reliable. The watchdog processor for real-time control systems on FPGAs such as Giaconia, etc., was investigated in this paper. As an alternative to providing the CPU with a timer, a reasonableness check on selected variables was carried out, as well as a simple programme flow inspection. El-Attar and colleagues pushed for the use of time register-based sequenced watchdog clocks to identify when a malfunction has occurred in a computer system. The software's defect-detection abilities are very limited, making it impossible to customise the software's features. To keep things as simple as possible, the authors concentrated on the fundamentals of an FPGA watchdog timer system that makes use of a large number of hardware timers [12].

While this strategy is effective, it increases the complexity of the hardware and the overall cost of the system, despite its effectiveness. The use of a Field Programmable Gate Array (FPGA) to build external watchdogs can reduce both the cost and the complexity associated with their implementation (FPGA). Today's embedded systems rely on FPGA chips to perform critical functions, which are found in a wide range of applications [8]. With the help of an FPGA, it is possible to design a watchdog timer that is both efficient and reliable. The watchdog processor for real-time control systems on FPGAs such as Giaconia, etc., was investigated in this paper. As an alternative to providing the CPU with a timer, a reasonableness checks on selected variables was carried out, as well as a simple programme flow inspection. El-Attar and colleagues pushed for the use of time register-based sequenced watchdog clocks to identify when a malfunction has occurred in a computer system. The software's defect-detection abilities are very limited, making it impossible to customise the software's features. To keep things as simple as possible, the authors concentrated on the fundamentals of an FPGA watchdog timer system that makes use of a large number of hardware timers [13]. While this strategy is effective, it increases the complexity of the hardware and the overall cost of the system, despite its effectiveness. The use of a Field Programmable Gate Array (FPGA) to build external watchdogs can reduce both the cost and the complexity associated with their implementation (FPGA). Today's embedded systems rely on FPGA chips to perform critical functions, which are found in a wide range of applications [14]. With the help of an FPGA, it is possible to design a watchdog timer that is both efficient and reliable. The watchdog processor for real-time control systems on FPGAs such as Giaconia, etc., was investigated in this paper. As an alternative to providing the CPU with a timer, a reasonableness check on selected variables was carried out, as well as a simple programme flow inspection. El-Attar and colleagues pushed for the use of time register-based sequenced watchdog clocks to identify when a malfunction has occurred in a computer system. The software's defect-detection abilities are very limited, making it impossible to customise the software's features. To keep things as simple as possible, the authors concentrated on the fundamentals of an FPGA watchdog timer system that makes use of a large number of hardware timers [15].

3. PROPOSED METHODOLOGY

watchdog timer is a crucial component in embedded systems and computer systems designed to enhance reliability by monitoring the system's operation and taking corrective actions when necessary. Its primary purpose is to detect and recover from malfunctions that might cause a system to become unresponsive or fail. The watchdog timer operates as a hardware-based countdown timer integrated into the system's microcontroller or processor. It must be regularly reset or "fed" by the software running on the system. If the timer is not reset within a predefined time interval, it assumes that the system has malfunctioned or stalled, triggering a corrective action. The way a watchdog timer functions is straightforward yet effective. When the system initializes or during normal operation, software periodically resets the watchdog timer, typically by writing a specific value to a designated hardware register.

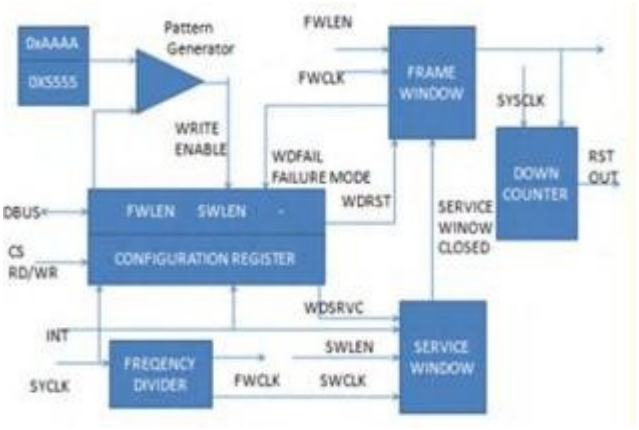

Figure 1: Proposed Operational Diagram

The above fig:1 shows the operational diagram of windowed watchdog timer status of the service window is regularly updated in the watchdog's configuration register. When the watchdog is correctly serviced within the service window, the counters immediately halt, and the frame window phase begins. The frame window follows a similar mechanism to the service window but operates using a separate derived clock, referred to as FWCLK. This clock is also slower than the SYSCLK, contributing to resource efficiency. The frame window includes an offset up/down counter, akin to the one in the service window, and a main counter with functionalities resembling those in the service window. The offset-up counter calculates the offset between the termination of the service window and the subsequent

rising edge of the FWCLK. This ensures precise synchronization between the two-timing windows. The main counter counts for (FWLEN - 1) times, after which the offset down counter takes over to complete the frame window's timing control. A critical aspect of this watchdog timer's functionality is its reset initialization and fault detection logic, which is depicted in a finite state machine (FSM). Upon initializing, the WDFAIL output is asserted, indicating an initial watchdog failure state. To initiate the watchdog timer's operation, a rising edge on the WDRST bit is required. This rising edge prepares the watchdog timer for initialization. Once the service window opens, a rising edge on the WDSRVC bit is crucial for deserting the WDFAIL output and commencing the window counters' operation. However, if the watchdog is serviced incorrectly within the service window, the entire initialization process is discarded. In such cases, the software must repeat the entire procedure to ensure proper watchdog operation. The WDFAIL signal is deserted only when the watchdog is successfully initialized and enters its operational phase. The length of this counter may be tailor-made based totally on the want for storing debug data or appearing other duties at some stage in the fault recovery system. Upon the counter's expiration, the watchdog timer asserts its RSTOUT output excessive, signifying the want for a device reset. It's important to observe that the reset counter remains nonfunctional throughout the strength-up section, and the RSTOUT output is first set to a low kingdom. The automated enablement of the reset counter happens handiest when the watchdog is initialized for the first time.

Applications:

- **Automotive:** In vehicles, WWDTs can detect and prevent malfunctions in critical systems like engine control, braking, and airbags, ensuring safety.
- **Industrial Automation:** They are used in process control plants and other industrial applications where downtime or malfunctions can be costly or dangerous.
- **Medical Devices:** In patient monitoring systems and other medical equipment, WWDTs help ensure the continuous and reliable operation of life-sustaining equipment.

Advantages:

- Increased fault detection Precision: The windowed watchdog timer ensures that the system must reset the timer within the specific time frame, not just before the timeout period expires. This helps in detecting faults that may cause the system to perform
- Improved Fault Tolerance and Recovery: In typical watchdog the system may be reset after one failure. With a windowed watchdog timer, the system has the opportunity to correct the fault within the followed time window making the system more resilient to transient faults and improving its recovery capabilities.
- More Efficient Resource Utilization: As the windowed timer minimizes false positive resets the system avoids unnecessary restart which might consume additional power or resources. This is the particularly useful in embedded systems with limited power supply.
- Graceful System Recovery: Instead of esetting the system immediately upon failure to kick the watchdog the wwdt allows for graceful handling of transient faults .For examples the system can recover from minor delays or errors as long as it operates within the window.
- Reduced Power Consumption: Since the WWDT avoids unnecessary resets the system the system does not undergo power draining restarts as frequently which is especially useful in embedded system with limited power budgets.
- Higher System Reliability: By allowing for a larger margin of error in timing the WWDT increases overall reliability. In system where precise timing is difficult to guarantee, the WWDT ensures that small timing fluctuations do not result in immediate failures or resets.
- Improved Fault Isolation: By distinguishing between a missed reset the WWDT helps pinpoint the nature of the timing fault making it easier to isolate specific issues like timing skew processor delay or unexpected latencies in the system improving diagnostics.

4. EXPERIMENTAL ANALYSIS

Figure 2 provides a visual representation of the watchdog timer's initialization and general operation (simulation) waveforms. At power-up or reset, the watchdog enters a failed state, indicated by the asserted high state of the WDFAIL output. It becomes the software's responsibility to initialize and activate the watchdog for ongoing operation.

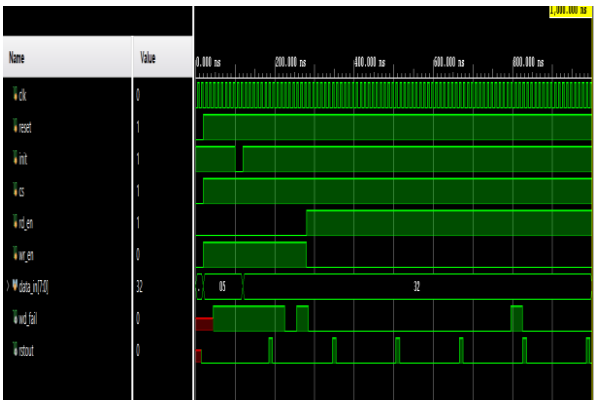

Figure 2: Simulation output

Resource	Estimation	Available	Utilization...
LUT	57	230400	0.02
FF	27	460800	0.01
IO	12	464	2.59
BUFG	1	544	0.18

Fig: 3 Design Summary

Figure:3 affords a concise design summary, presenting critical information approximately the FPGA implementation of the watchdog timer. It consists of key metrics that spotlight the utilization of precise resources in the FPGA layout. The proposed design makes use of fifty-seven out of a complete of 230400 available Look-Up Tables (LUTs) inside the FPGA. LUTs are essential good judgment elements in an FPGA that may be configured to perform diverse logical operations.

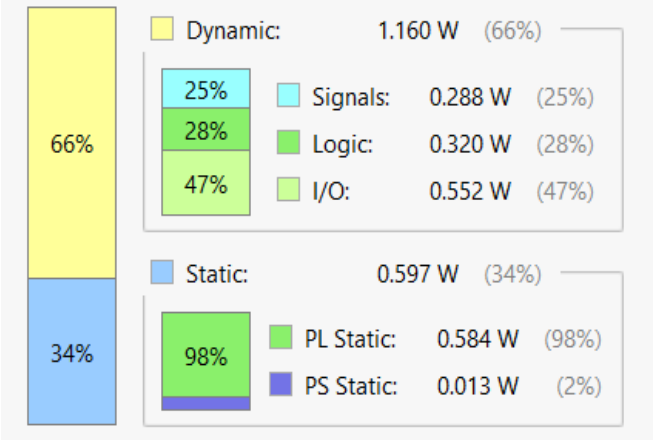

Figure 4: Power Summary

Figure 4 shows Similarly, in the case of Input/Output Blocks (IoBs), the proposed system achieves a 54% reduction, utilizing 27 IoBs compared to the existing system's 59. The Input/Output (IO) resources also see a substantial 73% reduction, with the proposed system using only 12 IOs, while the existing system employs 45. Finally, in terms of Global Buffers (BUFGs), the proposed system showcases an impressive 86% reduction, utilizing just 1 BUFG, whereas the existing system requires 7. These improvements clearly highlight the efficiency gains achieved by the proposed system in terms of resource utilization, paving the way for a more optimized and effective watchdog timer implementation.

5. CONCLUSION

The windowed watchdog timer offers a strong solution for reinforcing the reliability and precision of embedded manage systems It addresses current drawbacks related to conventional watchdog timers through presenting flexible window configuration, advanced security features, and dynamic initialization techniques. These improvements translate into advanced machine reliability, reduced vulnerability to software errors, and adaptableness to converting machine situations. In protection-essential and actual-time applications, the windowed watchdog timer proves to be an asset, preventing device screw ups and ensuring the smooth operation of embedded manage systems. The future holds promising possibilities for the evolution of watchdog timer generation and its integration into an extensive range of applications. Continuously enhancing security features to guard in opposition to potential assaults or unauthorized get admission to configuration registers, further fortifying device integrity. Developing algorithms that could dynamically adjust window intervals based on device workload, thereby optimizing resource utilization and responsiveness

The future scope of watchdog timers encompasses several promising directions driven by advancements in embedded systems, IoT (Internet of Things), and real-time computing technologies.These developments are poised to enhance system reliability, security, and efficiency across diverse industries. By analysing patterns of system behaviour and failure, intelligent watchdogs can adaptively adjust their timeout thresholds and recovery strategies. This can lead to more proactive fault detection and optimized system recovery, reducing downtime and improving overall system performance. With the growing concerns around cybersecurity, watchdog timers can play a crucial role in detecting and mitigating potential attacks or unauthorized access. Future watchdog implementations may include sophisticated security features such as intrusion detection, anomaly detection, and secure boot mechanisms.

These capabilities would make systems more resilient against malicious threats and ensure data integrity. environments where devices operate autonomously with limited human intervention. In edge scenarios, watchdog timers can ensure the continuous operation of critical services and applications, even in remote or harsh environments.

Future watchdog timers may incorporate advanced fault tolerance mechanisms beyond simple reset actions. These could include dynamic workload migration, self-healing algorithms, or redundancy strategies to maintain system integrity and availability in the face of failures.

REFERENCES

[1] Steblevska, I., T. Feuerstake, and M. R. Daymond. "Supervisor watchdog circuit to monitor an accelerator beam and control the safety interlock system." *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1056 (2023): 168595.

[2] Neeharika, T., et al. "Maximizing efficiency: building and running an AMBA APB APB protocol with open-source solutions." (2023): 99-103.

[3] Yang, Weitao, et al. "System-on-chip single event effect hardening design and validation using proton irradiation." *Nuclear Engineering and Technology* 55.3 (2023): 1015-1020.

[4] Aviles, Pablo M., et al. "Supervised Triple Macro synchronized Lockstep (STMLS) Architecture for Multicore Processors." *IEEE Access* 11 (2023): 128706-128723.

[5] Arshath, Mohamed. "Detection Of Soft Errors in Clock Synthesizers and Latency Reduction Throgh Voltage Scaling Mechanism." *Journal of VLSI circuits and systems* 6.1 (2024): 43-50.

[6] Albalooshi, Amina, Abdul-Halim M. Jallad, and Prashanth R. Marpu. "Fault Analysis and Mitigation Techniques of the I2C Bus for Nanosatellite Missions." *IEEE Access* 11 (2023): 34709-34717.

[7] dos Santos, Douglas Almeida, Pablo M. Aviles, André Martins Pio de Mattos, Mario García Valderas, Luis Entrena, Almudena Lindoso, and Luigi Dilillo. "Hybrid Hardening Approach for a Fault-Tolerant RISC-V System-on-Chip." In *RADECS 2023-European Conference on Radiation and Its Effects on Components and Systems*. 2023.

[8] Azambuja, J. R., Sousa, F., Rosa, L., & Kastensmidt, F. L. (2022). Evaluating the efficiency of software-only techniques to detect SEU and SET in microprocessors. *arXiv preprint arXiv:2309.16876*.

[9] Mo, Prasad Pa, and Arvind Singh. "FPGA based object parameter detection for Embedded Vision Application." *International Journal of Computing and Digital Systems* 14.1 (2022): 1091-1099.

[10] Nikiema, P. R., Palumbo, A., Aasma, A., Cassano, L., Kritikakou, A., Kulmala, A., ... & Traiola, M. (2021, July). Towards dependable RISC-V cores for edge computing devices. In *2023 IEEE 29th International Symposium on On-Line Testing and Robust System Design (IOLTS)* (pp. 1-7). IEEE.

[11] Ahangari, H., Özkök, Y. İ., Yıldırım, A., Say, F., Atik, F., & Ozturk, O. (2020). Architecture for safety–critical transportation systems. *Microprocessors and Microsystems*, *98*, 104818.

[12] Pagonis, G., Leon, V., Soudris, D., & Lentaris, G. (2020, September). Increasing the Fault Tolerance of COTS FPGAs in Space: SEU Mitigation Techniques on MPSoC. In *International Symposium on Applied Reconfigurable Computing* (pp. 215-229). Cham: Springer Nature Switzerland.

[13] Mattos, A. M., Santos, D. A., Imianosky, C., Melo, D. R., & Dilillo, L. (2019, June). Using HARV-SoC for Reliable Sensing Applications in Radiation Harsh Environments. In *2023 9th International Workshop on Advances in Sensors and Interfaces (IWASI)* (pp. 227-232). IEEE.

[14] Zagan, Ionel, and Vasile Gheorghiță Găitan. "FPGA imp lementation of hardware accelerated RTOSbased on real-time event handling." *The Journal of Supercomputing* (2019): 1-31.

[15] Palumbo, Alessandro, et al. "Improving the Detection of Hardware Trojan Horses in Microprocessors via Hamming Codes." 2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT). IEEE, 2019.

[16] Cherezova, Natalia, etal."Understanding fault-tolerance vulnerabilities in advanced SoC FPGAs for critical applications." *Microelectronics Reliability* 146 (2018): 115010.

[17] Groshev, A., Solodilov, M., Gusev, P., & Malysheva, A. (20218). Formation of a management strategy for innovation and investment activities of an enterprise. In *E3S Web of Conferences* (Vol. 458, p. 05034). EDP Sciences.

[18] Santos, D. A., Mattos, A. M., Melo, D. R., & Dilillo, L. (2017, October). Characterization of a Fault-Tolerant RISC-V System-on-Chip for Space Environments. In *2017 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)* (pp. 1-6). IEEE.

[19] Akanksha, Kammari, and R. Vyshnavi Duggaraju Kasturi. "Design of an ImprovedWatchdog Timer for MAC Applications." *computing* 52.6 (2017).