Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

Design of an Improved Watchdog Timer for Safety-Critical Applications

Madduri Jaceintha, Duggaraju Kasturi
R Vyshnavi Assistant Professor
Department of Electronics and Communication Engineering, Bhoj Reddy Engineering College for Women, Vinaynagar, Santoshnagar.
m.jaceintha2001@gmail.com , **duggarajukasturi@gmail.com**

Abstract

In safety-critical applications, reliability is of utmost importance for embedded systems. To ensure high reliability, external watchdog timers are commonly employed. These timers automatically handle and recover from failures that occur during the operation of the system. However, many existing external watchdog timers have limitations in terms of their functionality and require additional circuitry to adjust their timeout periods. This paper presents an improved configurable watchdog timer architecture and design suitable for safety-critical applications. The proposed watchdog timer contains several fault detection mechanisms which enhances its robustness. It has an advantage of easy adaptability to various applications, reducing overall system costs and relaxed timing constraint compared to existing solutions. The functionality and operations of this watchdog timer are designed to be versatile and can effectively monitor the operations of any processor-based real-time system. The effectiveness of the proposed watchdog timer in detecting and responding to faults is analyzed using Verilog HDL in Xilinx 14.7.

I Introduction

1.1 Introduction

A watchdog timer is a vital component employed in various systems, especially those present in safety-critical environments, where utmost reliability is required. This mechanism plays a very important role in automatically detecting and recovering from failures that may occur during system operation, ensuring the system's continuous and uninterrupted functionality. Unlike the conventional timers, watchdog timers are designed to monitor the operation of a system rather than measuring time intervals. They act as an independent entity, separate from the main processing unit, and are responsible for periodically checking if the system is functioning correctly. The Modern watchdog timers instill fault detection mechanisms which include checking specific patterns, monitoring critical system parameters, or performing self-tests to ensure their own reliability.

1.2 Watchdog Timer Function

A watchdog timer is a crucial component found in computer systems and embedded devices, serving the purpose of monitoring the proper functioning of the system. It can exist in either hardware or software form and plays a vital role in detecting and recovering from different malfunctions or errors that could potentially lead to system crashes or failures. The watchdog timer operates based on a simple principle. It relies on receiving periodic "heartbeat" signals from the system, indicating that the system is functioning as expected. These signals act as an indication of the system's normal operation. The watchdog timer expects to receive these signals within a predetermined time interval. If it fails to receive the expected heartbeat within that timeframe, it infers that the system has become unresponsive or has encountered a fault. To prevent potential hazards or damage caused by an uncontrolled or non-functioning system, the watchdog timer acts as a "watchdog" over the operation of microcontroller units(MCUs).While the watchdog timer function can be implemented within the MCU itself, external watchdog timers will provide an added layer of safety. These external watchdog timers operate independently of the MCU and ensure that even if the MCU malfunctions or stops functioning altogether, the watchdog timer remains active to monitor its behavior.

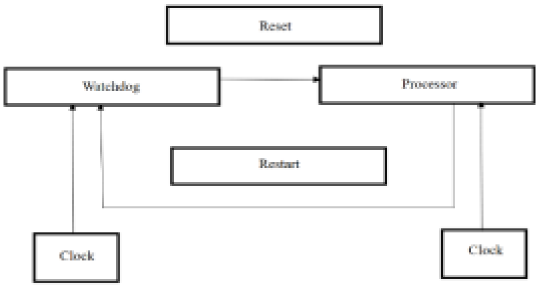


Figure 1: Watchdog Timer Function

1.3 Optimized Watchdog Timer

A well-designed watchdog timer goes beyond a simple timer for system reset and acts as a crucial defense against disruptive faults. To ensure a robust and responsive system capable of handling and recovering from faults, certain guidelines should be followed. Using separate clocks for the watchdog timers ensures its independent operation even if the subsystem clock encounters certain issues. Furthermore, implementing watchdog refresh mechanism minimizes accidental or unintended refreshes caused by runaway code. The watchdog timer's fault detection capabilities rely on flags assigned to tasks, where all flags should be set during a successful refresh. Any unset flags indicate a fault in the system.

1.4 Incorporation of Watchdog Timers in VLSI

The utilization of Watchdog timers in VLSI (Very Large-Scale Integration) is influential in improving the reliability and resilience of VLSI systems. In VLSI systems, Watchdog timers play a crucial role in fault detection and recovery by constantly monitoring system parameters such as clock signals, data integrity, and control registers, enabling them to identify anomalies and trigger appropriate responses. When a fault is detected, the Watchdog timer initiates predefined recovery procedures, which may involve the system resets, error handling, or reporting errors to external monitoring systems. By utilizing the Watchdog timers, VLSI systems achieve enhanced fault tolerance and ensure the continuity of critical operations without compromising safety.

II Literature Survey

2.1 Introduction

Watchdog timers play a crucial role in VLSI design by largely improving the reliability of electronic systems. The Extensive research conducted on watchdog timers in VLSI has yielded valuable insights. These studies highlight the importance of watchdog timers in a wide range of electronic systems, including real-time systems, safety-critical systems, SoC applications, and IoT devices.

2.2 Literature Survey

Several studies have proposed various techniques to enhance the performance of watchdog timers and mitigate the risks associated with the faults that include the transient and permanent faults. Ongoing investigations focus on exploring advanced architectures in improving reliability. They also strive to reduce power consumption and enhance security.

H. L. Ibrahim, M. A. Al-Qutayri, and K. Almiani in 2019 proposed a study titled "Enhancing the Reliability of Automotive Safety-Critical Systems Using a Multi-Tiered Watchdog Timer Architecture." Their study introduces a multi-tiered watchdog timer architecture that is designed to improve reliability of the automotive safety-critical systems.

In 2019, S. Alagarsamy and S. S. Sridhar conducted a research work titled "Fault Tolerance Evaluation of a Watchdog Timer-Based Soft Error Detection Scheme in Safety-Critical Applications." Evaluating a watchdog timer-based soft error detection scheme and assessing its fault tolerance capabilities in the safety-critical applications took place in this research.

"Design of a Fault-Tolerant Watchdog Timer for Safety-Critical Systems Using the Dual Modular Redundancy Technique", a paper published by B. Wang and M. Liu in 2020 enhances fault coverage and reduces the probability of system failure arising from watchdog timer faults.

A. Anand and R. Gayathri conducted a study in 2018 titled "A Study of Fault Coverage of Standard and Windowed Watchdog Timers", comparing the fault coverage effectiveness of standard and windowed watchdog timers in safety-critical VLSI systems.

III Existing and Proposed Watchdog Timers

3.1 Introduction

In this chapter, we will explore the existing and proposed watchdog timers, along with their corresponding block diagrams. We will also analyze the limitations of the existing watchdog timer and the advantages of the proposed watchdog timer.

3.2 Existing Watchdog Timer

In the existing system, a watchdog timer without a windowed watchdog is utilized. The input is directly transferred to the memory, and instructions are processed in the processor. However, this watchdog timer lacks immediate fault detection capability. If an error occurs between these stages, it will wait for its designated time to trigger the CPU, notifying that the error has occurred. This mechanism is entirely dependent on the CPU, resulting in a slow fault detection process. The sequential watchdog does not provide precise information about the error's location or cause. This lack of detailed information can make troubleshooting and debugging even more challenging. Moreover, relying completely on a single sequential watchdog timer creates a potential single point of failure.

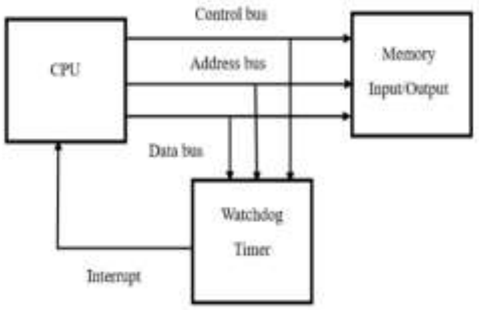


Figure 2: Block Diagram of Existing Watchdog Timer

3.3 Proposed Watchdog Timer

A reliable watchdog timer is crucial for detecting abnormal software behavior and restoring the system to a known state. The proposed watchdog timer in this paper operates independently of the processor, utilizing its own dedicated clock. When the timer expires, indicating a failure, a failure flag is raised, and a reset is triggered after a fixed interval. This interval can be utilized by the software to store valuable debugging information. The timer's design includes separate service and frame windows. The service window, triggered by a high-to-low transition on the INIT signal, uses a slower clock which is SWCLK to minimize resource usage. It includes an offset up/down counter clocked by the system clock (SYSCLK) and a main counter running at SWCLK. Once the watchdog is properly serviced, the counters in the service window halt and the frame window begins. The frame window, also employing a slower clock which is FWCLK, has an offset up/down counter also clocked by SYSCLK and a main counter clocked by FWCLK. It resets if the watchdog is serviced within the next service window duration. If the software fails to signal within the frame window, the watchdog timer assumes system failure and triggers a predetermined action.

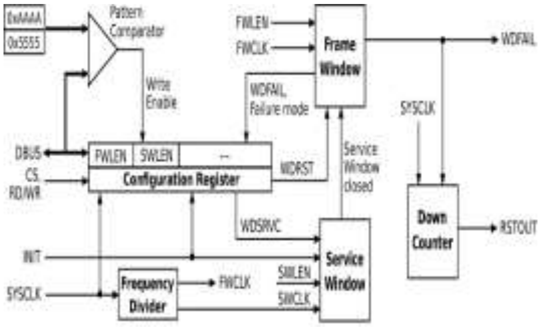


Figure 3: Block diagram of Proposed Watchdog Timer

IV Results

4.1 Introduction

This chapter includes simulation, synthesis, and the performance characteristics of the proposed watchdog timer. The Xilinx 14.7 software was utilized for its development and Verilog HDL was used to design the specific logic and functionality of the watchdog timer.

4.2 Simulation of Proposed Watchdog Timer

The figure depicts the Proposed Watchdog Timer as the Top module, consisting of sub-modules such as the Configuration register, frame window, interrupt detector, and down counter. The 32-bit input signals "datain" and "datain1" are monitored when the clock signal is high, and the rest signal is low. If no faults are found, the "check_interrupt" signal transitions low and the "security_check" signal goes high, activating the "wd_enable" signal. The "max_outtime" signal retains the input data for timing purposes.

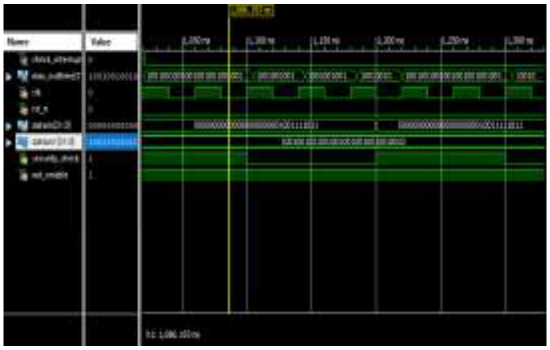


Figure 4: Simulation of Proposed Watchdog Timer

4.3 Synthesis of Proposed Watchdog Timer

The following figure shows the synthesis of the proposed watchdog timer which describes the given input and output signals. The Clk, rst_n, security_check, wd_enable, datain, and datain1 are the input signals to the watchdog timer whereas the check_interrupt and max_outtime are output signals.

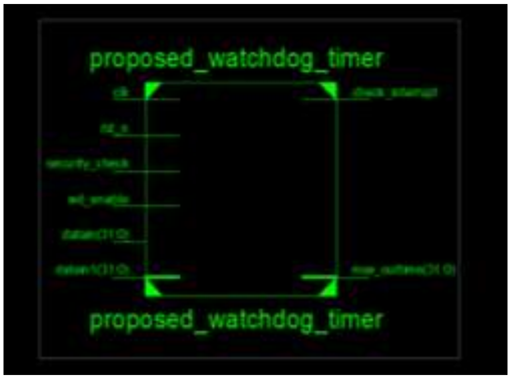


Figure 5: Synthesis of Proposed Watchdog Timer

4.4. Synthesis of Internal Design of Proposed Watchdog Timer

The following figure shows the detailed synthesis of the proposed watchdog timer consisting of a configuration register, frame window, down counter, and interrupt check modules.

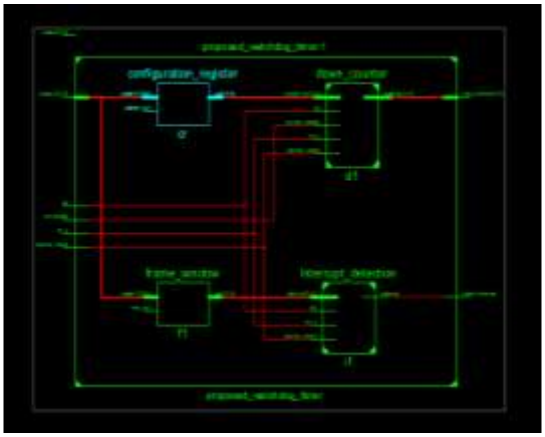
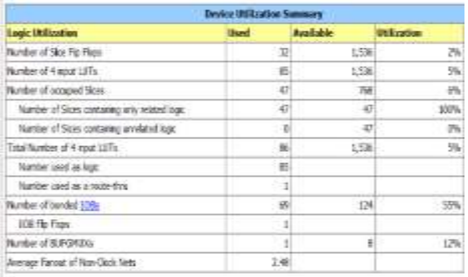


Figure 6: Synthesis of Internal Design of Proposed Watchdog Timer

4.5 Area Calculations of Proposed Watchdog Timer

The following figure describes the design summary of the proposed watchdog timer.

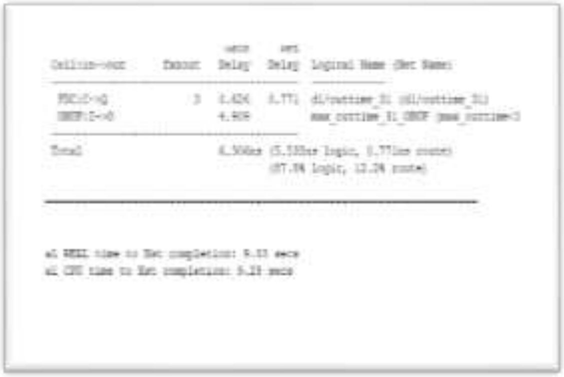


Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	32	1,536	2%
Number of 4 input LUTs	85	1,536	5%
Number of occupied Slices	47	768	6%
Number of Slices containing only related logic	47	47	100%
Number of Slices containing unrelated logic	0	47	0%
Total Number of 4 input LUTs	86	1,536	5%
Number used as logic	85		
Number used as a route-thru	1		
Number of bonded IOBs	69	124	55%
IOB Flip Flops	1		
Number of BUFGMUXs	1	8	12%
Average Fanout of Non-Clock Nets	2.48		

Figure 7: Area Calculations of Proposed Watchdog Timer

4.6 Delay Calculations of Proposed Watchdog Timer

The figure describes the delay calculations of the proposed watchdog timer. Where the total delay is 6.306 ns.



```
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDC:C->Q              3   0.626   0.771  d1/outtime_31 (d1/outtime_31)
     OBUF:I->O                 4.909          max_outtime_31_OBUF (max_outtime<31>)
    ----------------------------------------
    Total                      6.306ns (5.535ns logic, 0.771ns route)
                                       (87.8% logic, 12.2% route)

=========================================================================

Total REAL time to Xst completion: 5.33 secs
Total CPU time to Xst completion: 5.28 secs
```

Figure 8: Delay Calculations of Proposed Watchdog Timer

4.7 Power Calculations of Proposed Watchdog Timer

The following figure describes the power calculations of the proposed watchdog timer. Where the dynamic power is 0.19 W and quiescent power is 0.027 W which makes the total power supply 0.046 W.

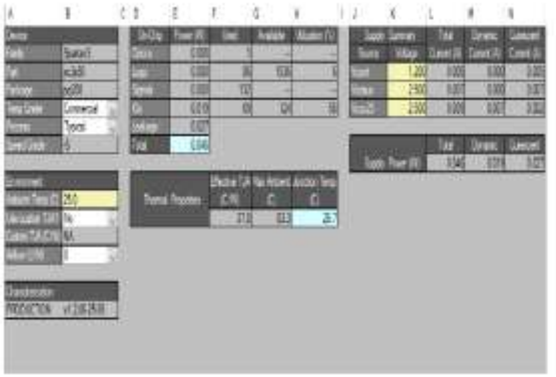


Figure 9: Power Calculations of Proposed Watchdog Timer

V Conclusion

This paper thoroughly investigates the enhanced watchdog timer, highlighting the limitations of existing solutions and driving the development of the proposed timer. It emphasizes the crucial role of watchdog timers in ensuring safety for VLSI-based applications. The paper includes a literature survey to analyze existing knowledge. The proposed watchdog timer functionality is to be independent of the processor, allowing customization of the timer parameters for specific application requirements. It has been successfully designed using the Xilinx 14.7 software, using the Verilog HDL for broad device compatibility and minimal hardware overhead.

References

[1] S. N. Chau, L. Alkalai, A. T. Tai, and J. B. Burt, “Design of a fault tolerant COTS-based bus architecture,” IEEE Transactions on Reliability, vol. 48, no. 4, pp. 351–359, Dec. 2018.

[2] V. B. Prasad, “Fault tolerant digital systems,” IEEE Potentials, vol. 8, no. 1, pp. 17–21, Feb. 2019.

[3] J. Beningo, “A review of watchdog architectures and their application to Cubesats,” Apr. 2017.

[4] A. Mahmood and E. J. McCluskey, “Concurrent error detection using watchdog processors - a survey,” IEEE Transactions on Computers, vol. 37, no. 2, pp. 160–174, Feb. 2019.

[5] J. Ganssle, “Great watchdogs,” V-1.2, The Ganssle Group, updated January 2004, 2020.

[6] E. Schlaepfer, “Comparison of internal and external watchdog timers’ application note,” Maxim Integrated Products, 2018.

[7] P. Garcia, K. Compton, M. Schulte, E. Blem, and W. Fu, “An overview of reconfigurable hardware in embedded systems,” EURASIP Journal on Embedded Systems, vol. 2014, no. 1, pp. 13–13, Jan. 2018.

[8] G. C. Giaconia, A. Di Stefano, and G. Capponi, “FPGAbased concurrent watchdog for real-time control systems,” Electronics Letters, vol. 39, no. 10, pp. 769–770, Jun. 2021.

[9] A. M. El-Attar and G. Fahmy, “An improved watchdog timer to enhance imaging system reliability in the presence of soft errors,” in Signal Processing and Information Technology, 2020 IEEE.

[10] B. S. Verma, S. S. Dhami, and H. S. Dhami, "A Comparison of Various Watchdog Timer Configurations for Safety-Critical Systems," IEEE Access, vol. 6, pp. 35790-35801, 2018.