

Fault Detection Mechanism using Improved Watchdog Timer for Safety Application

B. B. Manjula¹, N. Santhosh^{2*}, K. S. Ravikiran³, K. Pooja⁴, H. V. Sahana⁵

¹Associate Professor, Department of Electronics and Communication Engineering, East West Institute of Technology, Bangalore, India

^{2,3,4,5}Student, Department of Electronics and Communication Engineering, East West Institute of Technology, Bangalore, India

Abstract: This project describes the architecture and design of an improved configurable watchdog timer that can be employed in safety-critical applications. Watchdog timers are used in such systems to automatically handle and recover from operation time related failures. Several fault detection mechanisms are built into the watchdog, which adds to its robustness. This project also discusses the implementation of the proposed watchdog timer in a Field Programmable Gate Array (FPGA) Spartan3. The effectiveness of the proposed watchdog timer to detect and respond to faults. The language used is Verilog HDL and the simulation tool that used is Xilinx 14.7.

Keywords: FPGA, Verilog HDL, Xilinx ISE14.7.

1. Introduction

A watchdog timer (WDT) is a hardware subsystem that monitors the operations of the systems and takes certain actions in the event of detecting a fault. This project discusses the design of an improved watchdog timer and its implementation in FPGA. The design in FPGA means that the same watchdog hardware can be interfaced to different processors and systems, with only minor modifications of the associated hardware description language (HDL) code. This project describes the architecture of the proposed watchdog timer, the fault detection mechanisms built into the watchdog, the implementation of the watchdog timer in FPGA.

2. Problem Description

In the standard watchdog timer, a watchdog timer with no windowed watchdog is executed. The input is directly sent into the memory, from the memory instructions are processed into the processor, this watchdog will not detect the fault immediately. If there is any error occurrence in between them, it will sequentially wait for its time to trigger the CPU that error has occurred. It is totally dependent on the CPU. Then after CPU, getting the error information it will reset the whole process. It is stated as slow watchdog fault mechanism. The time it takes to reach the error mechanism to rectify is more than the proposed system. Since it is not clock independent, this watchdog is a failure to embedded system. It is rectified during

this proposed system.

3. Watchdog Timer Setup

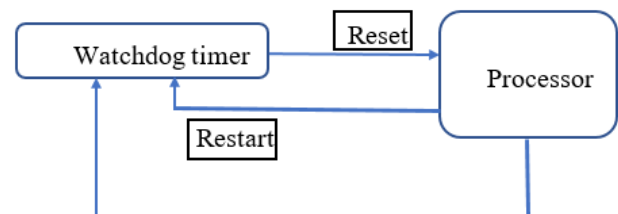


Fig. 1. A typical watchdog timer setup

Figure 1 shows a typical arrangement. As shown, the watchdog timer is a chip external to the processor. However, it could also be included within the same chip as the CPU. This is done in many microcontrollers. In either case, the output from the watchdog timer is tied directly to the processor's reset signal.

4. Literature Review

The literature review attempts to discuss the work carried out in the areas of watchdog timer design for FPGA's using Verilog HDL.

FPGA Implementation of an Improved watchdog Timer for Safety-Critical Applications, Ravi Krishnan Unni, Vijayanand, P. Y. Dilip (2018)

The fault injection block, how the random numbers are injected into the program counter at random periods of time, efficient way to check our parameter, thus leading to watchdog fail in each stage. Watchdog hardware, decide early on exactly how you intend to make best use of it, in fault detection and gives benefits of a more robust system. Analysis of various test bench waveforms, of external watchdog timer's application (space launch vehicle) implementation. Implementation of watchdog in space launch vehicle to measure the parameters like temperature, pressure, heat explosion which is necessary for efficient maintenance of any space shuttles.

It clearly explains the efficient way to check the parameters

*Corresponding author: santhoshnijalingappa@gmail.com

like temperature, pressure, and heat explosion individually using separate window, thus leading to watchdog fail in each stage but in existing system all the parameters are checked at a time with some prerecorded value, can't able to check individual parameter separately but proposed system helps to overcome this fault.

FPGA Implementation of Multiple Hardware Watchdog Timers for Enhancing Real-Time Systems Security by Maria Pohronska, Tibor Krajcovic (2018)

They verifying the ongoing frameworks by giving them extra equipment guard dog clocks. This work proposes the fundamental idea of the numerous equipment guard dog clock framework and depicts the proposed design of the framework giving 256 equipment guard dog clocks.

The system proposed in this paper consists of several hardware implemented, watchdog timers with their control logic and an interface for the communication with the master device, in this main objective is to trigger each time when the master processor needs to assign a watchdog timer to a new task, instead of creating and assigning a virtual software watchdog timer.

Implementing a Microcontroller Watchdog with a Field Programmable Gate Array (FPGA) by Bartholomew F. Straka, John F. Kennedy Space Center (2017)

The importance of reliability in safety and high-cost scenarios, voting logic and watchdogs are necessary for many redundant systems, here they implementing a system watchdog with FPGA (example voting logic). The possible failure of voting logic itself, however implemented, is an additional consideration. For example, sensitivity to radiation at the gate level is a concern. Therefore, selection of appropriately radiation-tolerant devices may be critical to the reliability of such a voting logic application. The specific application discussed here is for a Dual Modular Redundancy (DMR) microcontroller pair. In this DMR system, the two microcontrollers are fed the same inputs and operate in parallel. This output of one processor is actively controlling the end-item devices while the output of another controller is inhibited. When the primary fails, the standby resumes the desired FPGA. A disruption in the heartbeat of the initially-active microcontroller will cause the FPGA to switch control to the alternate microcontroller.

Comparison of internal and external watchdog timers by Eric Schlaepfer (2017)

This presents a comparison between internal and external watchdog timers and explains their benefits, drawbacks by taking MAXQ2000 WDT as internal and external as a standalone alone IC or embedded as part of a support ASIC. For every watchdog timers counters are the main component and counter increments by each clock cycle when overflow of counter exist it forces the system to reset. To prevent this reset counter register should be clear periodically, this procedure termed as "servicing the watchdog". But in the case of external watchdog timer IC's have digital input that clears the counter register on every low to high or high to low transition.

In the case of internal watchdog timer if it is not serviced in the prescribed time interval then it assumes that MAXQ2000

WDT overflows, triggers an interrupt, counts for 512 additional system clock cycles, and then resets itself. But in the case of external watchdog (standalone IC's) logic input pins on the WDT can disable the timer and prevent

resets. Although it is tempting to connect the logic pins to the microprocessor, doing so risks a compromise in system reliability because run away code could accidentally disable the watchdog.

Application of Software Watchdog as a Dependability Software Service for Automotive Safety Relevant Systems by Xi Chen, Vera Lauer, Martin Hiller, Juejing Feng (2016)

To confront the difficulties coming about because of the expanding thickness of use programming segments and higher reliability prerequisites of things to come wellbeing frameworks in the car hardware, a constancy programming administration to screen singular application programming parts in runtimes required so as to improve the general framework trustworthiness.

5. Proposed System Architecture

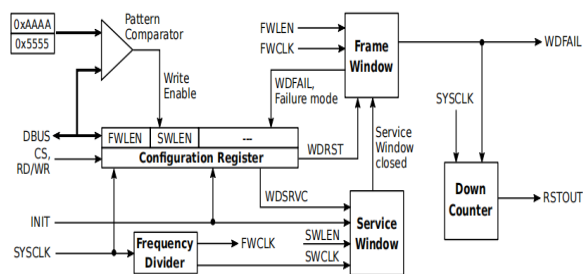


Fig. 2. Block diagram of proposed watchdog timer

The proposed architecture for watchdog timer. Fig. 2 shows the realization of proposed watchdog timer in FPGA. The design is clocked by the SYSCLK input and, configure the window lengths based on application by writing appropriate bits in register SWLEN for the service window and FWLEN for frame window, when the INIT signal transition from high to low service window will starts and it is clocked by derived clock SWCLK that is much slower than SYSCLK because it helps to reduce the number of comparators which in turn helps in reducing the resource utilization in FPGA. Up down counter in the service window clocked by SYSCLK and main counter by SWCLK. Frame window uses derived clock FWCLK for its operation, similar to the service window it also has offset up/down counter.

6. Methodology and Implementation

A. Methodology

For any circuit to be implemented design specification are the important aspects. Initially, the design specifications for. After the design specification the code was modeled in Verilog HDL language. The code was simulated and the outputs are tested for various combination of inputs. Application is created to test the timer using fault injection. Finally testing the timer by interfacing the application between timer.

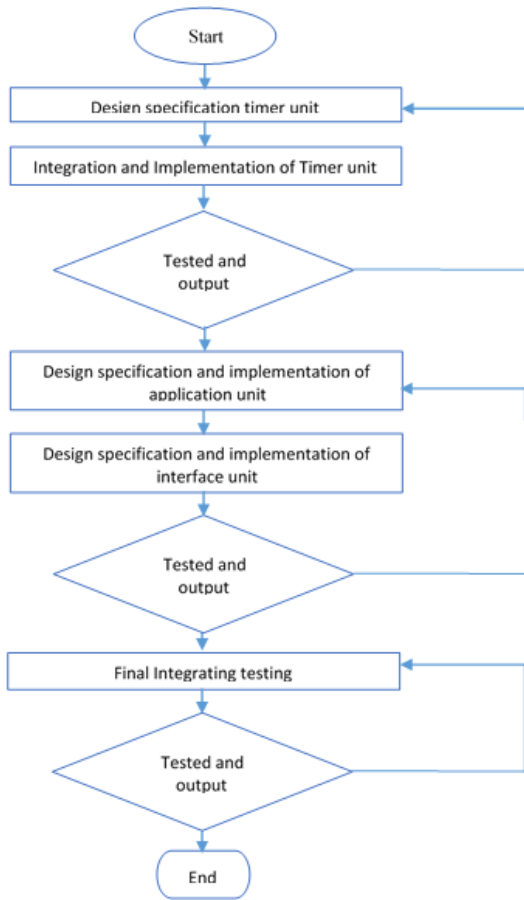


Fig. 3. Project flow

7. Results

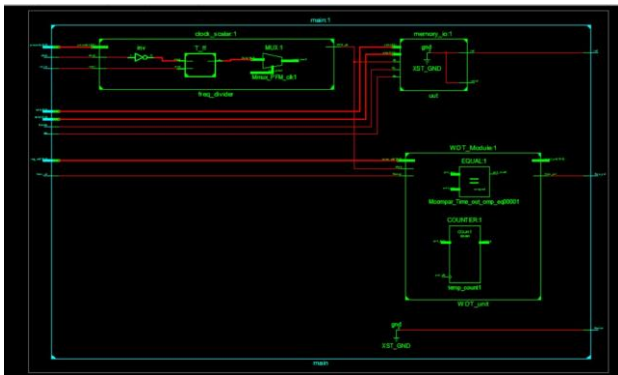


Fig. 5. Internal RTL Schematic of Watchdog timer

When restart is high, program counter before reaching the comparative value, thus restart = 1. It indicates that application is in good condition.

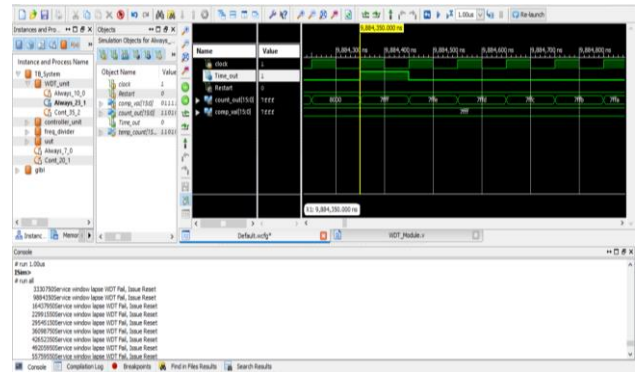


Fig. 7. Simulation result when watchdog fail to reset

When restart is low, if program counter reaches the comparative value. In this state the timeout is high then there is a fault. Thus restart =0, timeout = 1.

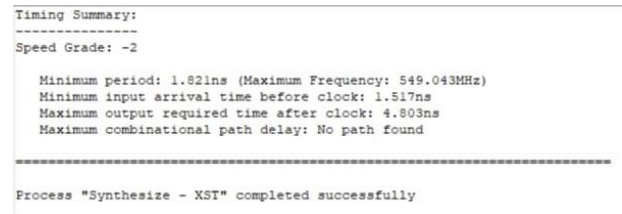


Fig. 8. Timing summary

8. Conclusion

This paper presented in detail the architecture and design of an improved watchdog timer and its implementation in FPGA. The watchdog timer runs completely independent of the processor and permits adjusting the timer parameters according to the application. Implementing the entire design in FPGA has the advantage of making it adaptable and reusable. HDL based designs are vendor-independent and can be used on different FPGA devices with low overhead. The same design can also be customized for different processors and applications with only minor HDL modifications. The implementation has low complexity and takes up very less amount of hardware resources. The proposed design was tested using fault injection techniques and proved to be effective in handling various faults.

9. Future Scope

The scope of this paper or project to be simulate the Watchdog timer using different simulation tools like Atmel, Altera and Cortex-M. HDL codes improve the security of the application compare to conventional coding. HDL is a new technology which may prove to be highly useful and provide more efficient service. Also an application was developed using MATLAB which emulates as a processor, it is interfaced with FPGA which is running the watchdog timer.



Fig. 6. Simulation result when watchdog timer reset and restart received from controller

References

- [1] Ravi Krishnan Unni, Vijayanand P, Y. Dilip, "FPGA Implementation of an Improved Watchdog Timer for Safety-Critical Applications," 2018.
- [2] N. Murphy, "Watchdog's timers," *Embedded Systems Programming*, p. 112, 2017.
- [3] A. M. El-Attar and G. Fahmy, "A study of fault coverage of standard and windowed watchdog timers," in *Signal Processing and Communications*, 2017. ICSPC 2017. IEEE International Conference on. IEEE, Nov. 2017, pp. 325–328.
- [4] M. Barr, "Introduction to watchdog timers," *Embedded Systems Design*, 2016.
- [5] M. Wirthlin, "High-reliability FPGA-based systems: Space, high-energy physics, and beyond," *Proceedings of the IEEE*, vol. 103, no. 3, pp. 379–389, Mar. 2015.
- [6] H. Ziade, R. A. Ayoubi, R. Velazco et al., "A survey on fault injection techniques," *The International Arab Journal of Information Technology*, vol. 1, no. 2, pp. 171–186, Jul. 2014.
- [7] M. Pohronska and T. Krajcovic, "FPGA implementation of multiple hardware watchdog timers for enhancing real-time systems security," in *EUROCON International Conference on Computer as a Tool (EUROCON)*, 2011 IEEE, Apr. 2014, pp. 1–4.
- [8] B. Straka, "Implementing a microcontroller watchdog with a field programmable gate array (FPGA)," Apr. 2013.