Design of Watchdog Timer for Real Time Applications

S.Shilpa, Ch.Umashankar, S V S Prasad

Abstract: *The Embedded system is employ in safety and critical application, which is greater reliability. The watchdog timers are used in automatic systems to handle the operation time for secure the timer failure. Majority of the watchdog timers used an additional circuit to adjust their timeout position and it will provide limited services in terms of working. This paper presents the architecture of a watchdog timer and also gives the design structure, it will working in safety and critical conditions. The operations are general and it can be used to monitor the working of any processor in real-time application. This paper discussed the implementation of the proposed timer in a FPGA. This will helps to design easily in different applications, it will gives reduces the overall system cost. The watchdog timers is to detect and give response very effectively and also gives the responses of faults by analyzing the simulations.*

Keywords: *watchdog timer, computer, clock*

I. INTRODUCTION

A guard dog clock (once in a while called a PC working appropriately or COP clock, or just a guard dog) is an electronic clock that is utilized to recognize and recoup from PC breakdowns.

In the event that, because of an equipment blame or program mistake, the PC neglects to reset the guard dog, the clock will slip by and produce a timeout flag.

Guard dog clocks are regularly found in implanted frameworks and other PC controlled hardware where people can only with significant effort get to the gear or would be unfit to respond to flaws in a convenient way. A guard dog clock is typically utilized in cases like these. Guard dog clocks may likewise be utilized when organization un-confided in code in a sand-box, to restrain the CPU time accessible to the code and along these lines keep a few sorts of forswearing of-administration assaults.

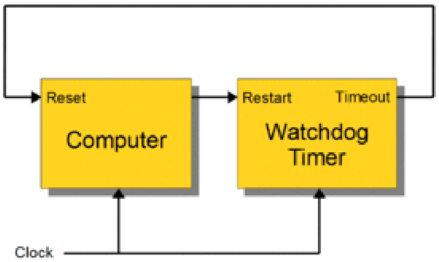


Figure 1: Watchdog Timer

II. LITERATURE SURVEY

1. "An Improved Watchdog Timer to Enhance Imaging System Reliability In The Presence Of Soft Errors" By GamalFahmyand Ashraf M. El-Attar (2007)

Satellite, Ariel imaging frameworks is situated at highest elevations. In this manner, they are more powerless against Soft Errors than comparative frameworks working adrift dimension. The paper examines the impact of transient blames on chip basis imaging frameworks.

2. "FPGA Implementation of Multiple Hardware Watchdog Timers for Enhancing Real-Time Systems Security" By TiborKrajčovič, MáriaPohronská(2010)

In this work, we manage verifying the ongoing frameworks by giving them extra equipment guard dog clocks. This paper proposes the fundamental idea of the numerous equipment guard dog clocks framework and depicts the proposed design of the framework giving 256 equipment guard dog clocks.

3. "A Design and Implementation of Watchdog Based on Observer Pattern and Finite State Machine "ByXinyun Liu, Wenqiang Song, Shihang Chen (2014)

The paper presents, a strategy with ViewerPattern and FiniteState Machinefor guard dog execution is proposed.

4. "Application of Software Watchdog as a Dependability Software Service for Automotive Safety Relevant Systems "By Xi Chen, Vera Lauer Martin Hiller, Juejing Feng (2007)

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 partsin runtimeis required so as to improvethe general framework trustworthiness.

III. EXISTING SYSTEM

FPGA-based simultaneous guard dog for continuous control frameworks consider the usage of a basic simultaneous guard dog processor which FPGA for constant frameworks of control. The structure don't give a clock to the processor rather, it played out a sensibility beware of certain factors and an essential program stream check. An improved guard dog clock to upgrade imaging framework unwavering quality within the sight of delicate blunder proposed a sequenced guard dog clock that utilized time registers to decide if blame has happened. Be that as it may, it didn't offer much arrangement choices and the blame identification highlights actualized were restricted.

Revised Manuscript Received on July 18, 2019.

S.SHILPA, Graduate, Department of Electronics and Communication Engineering, MLR Institute of Technology, Hyderabad, India

Ch.UMASHANKAR, Assistant Professor, Department of Electronics and Communication Engineering, MLR Institute of Technology, Hyderabad, India

S V S PRASAD, Professor, HOD, Department of Electronics and Communication Engineering, MLR Institute of Technology, Hyderabad, India

PROPOSED SYSTEM:

A practical watchman hound should have the ability to distinguish all unusual programming modesand reoccurrence the system to a knownstate. It should haveits individualspecial clock wouldbe prepared for giving a hardware reset ontimeout toall of theperipherals. The watchman hound check proposed in thispaper works uninhibitedly of theprocessor and usage's a dedicatedclock forits abilities. A miss the mark flag is raised when the watchman hound clock slips and after aimmovable proportion of time fromraising thepennant, a reset is endorsed. The time amidst canbe used by the element to store noteworthy examining info to non-capricious medium.

IV. ARCHITECTURE OF WATCHDOG TIMER

Normal guard dog clock get issues inthe framework, for example, draping in light of unlimited circles in code execution. In any case, the primary burden of this guard dog is the framework arrives a blame state in which persistently reset the clock, the mistake state won't be recognized. As it were, a standard guard dog clock can identify moderate flaws, yet can't distinguish quick blames which happen inside the guard dog clock period.

Input-output interface of watchdog timer and configuration register:

Fig. 2 demonstrates the info yield (I/O) interface of the proposed guard dog clock. The guard dog has two yields, to be specific the guard dog come up short yield (WDFAIL) and the rearrange yield (RSTOUT). At the point when the SYS-RESET input is low, the WD-FAIL yield stays stated and theRST-OUT yield stays deserted. The structure likewise comprises of a setup register with bitfields characterized as inthe figure.

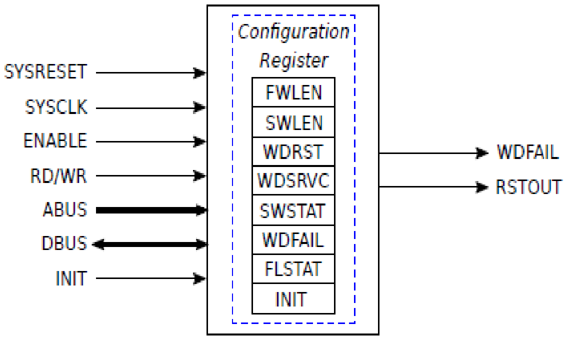


Fig. 2. Input-output interface of watchdog timer and configuration register

Theregister empowers acclimations to the guard dog parametersand furthermore gives status data. The WDRST and WDSRVC fields are utilized individually to resetting and adjusting the guard dog.

Initialization of watchdog Timer

On impetus or reset the gatekeeper hound stirs in a failedstate, i.e., the WDFAIL yield will be bore witness to high. It's the obligation of the item to present the gatekeeper puppy andkeep it running. Fig. 3 speaks to thewaveform for gatekeeper hound reset instatement andgeneral assignment.

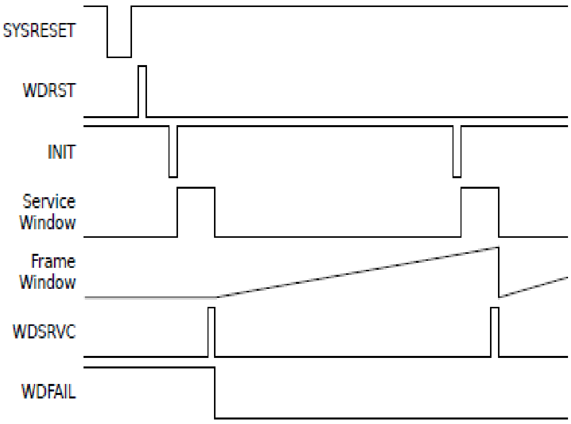
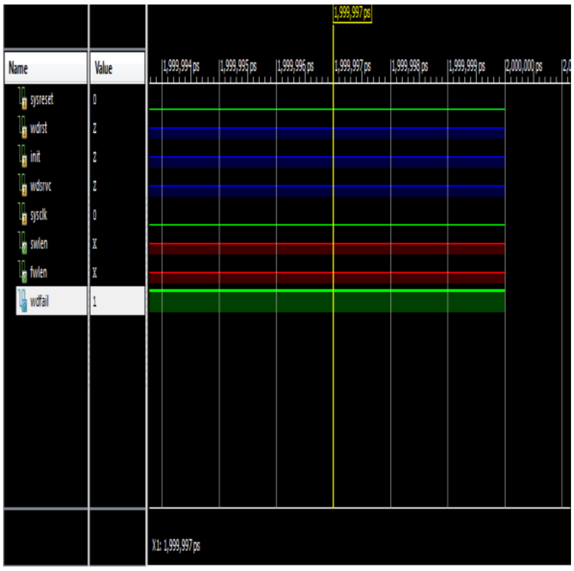


Fig. 3.Initialization of watchdog Timerand service operation

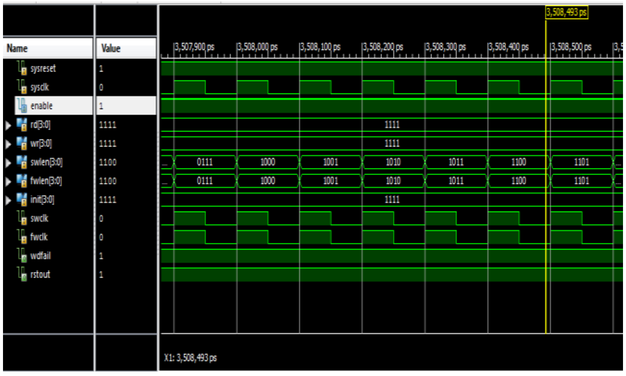
V. RESULTS AND SIMULATION

Sytreset=0 ,wdfail= 1;



Name	Value
sysreset	0
wdrst	Z
init	Z
wdsrvc	Z
sysclk	0
swlen	X
fwlen	X
wdfail	1

Overall output:



Name	Value
sysreset	1
sysclk	0
enable	1
rd[3:0]	1111
wr[3:0]	1111
swlen[3:0]	1100
fwlen[3:0]	1100
init[3:0]	1111
swclk	0
fwclk	0
wdfail	1
rstout	1

From the result, we become progressively familiar with that by using two window diagrams i.e., organization window and edge window completes stable screw up acknowledgment framework. Realizing the entire structure in FPGA has the benefit of creation it adaptable and recyclable.

CONCLUSION

The paper introduced in part the engineering and plan of abetter-quality windowed guard dog clock and its execution in FPGA. The guard dog clock turns totally autonomous of the processor and licenses changing the clock limits as indicated by the application. A few blame recognition procedures are incorporated with the guard dog for the primary discovery of whimsical programming modes. It has the capacity to distinguish the disappointment type and log it, which can wind up important while investigating. After distinguishing a disappointment, the guard dog clock likewise permits the product adequate time for sparing the investigate data, before starting a reset.

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. 1999.
2. V. B. Prasad, “Fault tolerant digital systems,” IEEE Potentials, vol. 8, no. 1, pp. 17–21, Feb. 1989.
3. J. Beningo, “A review of watchdog architectures and their application to Cubesats,” Apr. 2010.
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. 1988.
5. B. Straka, “Implementing a microcontroller watchdog with a field programmable gate array (FPGA),” Apr. 2013.
6. J. Ganssle, “Great watchdogs,” V-1.2, The Ganssle Group, updated January 2004, 2004.
7. E. Schlaepfer, “Comparison of internal and external watchdog timers application note,” Maxim Integrated Products, 2008.
8. P. Garcia, K. Compton, M. Schulte, E. Blem, and W. Fu, “An overview of reconfigurable hardware in embedded systems,” EURASIP Journalon Embedded Systems, vol. 2006, no. 1, pp. 13–13, Jan. 2006.
9. G. C. Giaconia, A. Di Stefano, and G. Capponi, “FPGA-based concurrent watchdog for real-time control systems,” Electronics Letters, vol. 39, no. 10, pp. 769–770, Jun. 2003.
10. 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, 2007 IEEE InternationalSymposium on. IEEE, Dec. 2007, pp. 1100–1104.
11. M. Pohronsk´a and T. Krajˇcoviˇc, “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. IEEE, Apr. 2011, pp. 1–4.
12. H. Guzman-Miranda, L. Sterpone, M. Violante, M. A. Aguirre, and M. Gutierrez-Rizo, “Coping with the obsolescence of safety- or missioncritical embedded systems using fpgas,” IEEE Transactions on IndustrialElectronics, vol. 58, no. 3, pp. 814–821, 2011.
13. H. Amer and A. Sobeih, “Increasing the reliability of the Motorola MC68HC11 in the presence of temporary failures,” in ElectrotechnicalConference, 2002. MELECON 2002.11th Mediterranean. IEEE, May 2002, pp. 231–234.