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 unconfined in code in a sand-box, to restrain the CPU time accessible to the code and along these lines keep a few sorts of forsaking of-administration assaults.

II. LITERATURE SURVEY


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 blunders on chip basis imaging frameworks.


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.


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


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 runtime is required so as to improve the 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.
PROPOSED SYSTEM:
A practical watchman hound should have the ability to distinguish all unusual programming modes and reoccurrence the system to a known state. It should have its individual special clock would be prepared for giving a hardware reset on timeout to all peripherals. The watchman hound check proposed in this paper works uninhibitedly of the processor and usage's a dedicated clock for its abilities. A miss the mark flag is raised when the watchman hound clock slips and after an immovable proportion of time from raising the pennant, a reset is endorsed. The time amidst can be used by the element to store noteworthy examining info to non-capricious medium.

IV. ARCHITECTURE OF WATCHDOG TIMER

Normal guard dog clock get issues in the 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 asserted and the RST-OUT yield stays deserted. The structure likewise comprises of a setup register with bitfields characterized as in the figure.

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

There register empowers acclimations to the guard dog parameters and 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 bone witness to high. It’s the obligation of the item to present the gatekeeper puppy and keep it running. Fig. 3 speaks to the waveform for gatekeeper hound reset instatement and general assignment.

Fig. 3. Initialization of watchdog Timer and service operation

V. RESULTS AND SIMULATION

Syreset=0, wdfail= 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 a better-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