ISSN: 2278-4632
Vol-13, Issue-11, November 2023

FPGA IMPLEMENTATION OF AN IMPROVED WATCHDOG TIMER FOR SAFETY-CRITICAL APPLICATIONS

Dr.S.KANNAN, **B. Tech, M.Tech, Ph.D**., PROFESSOR OF ECE IN MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE, MAISAMMAGUDA, MEDCHAL (M), HYDERABAD-500100, T. S.

ADABALA CHARISHMA, KANDULA ARUN, ANNEM ABIJITH REDDY. FINALYEAR STUDENTS FROM DEPT OF ECE IN MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE, MAISAMMAGUDA, MEDCHAL (M), HYDERABAD-500100, T. S.

Abstract: Embedded systems that are employed in safety critical applications require highest reliability. External watchdog timers are used in such systems to automatically handle and recover from operation time related failures. Most of the available external watchdog timers use additional circuitry to adjust their timeout periods and provide only limited features in terms of their functionality. This paper describes the architecture and design of an improved configurable watchdog timer that can be employed in safety-critical applications. Several fault detection mechanisms are built into the watchdog, which adds to its robustness. The functionality and operations are rather general and it can be used to monitor the operations of any processor based real-time system. This paper also discusses the implementation of the proposed watchdog timer in a Field Programmable Gate Array (FPGA). This allows the design to be easily adaptable to different applications, while reducing the overall system cost. The effectiveness of the proposed watchdog timer to detect and respond to faults is first studied by analyzing the simulation results. Thus after designing the watchdog it is implemented in ATM and verified. The design is validated in a real-time hardware by injecting faults through the software while the processor is executing.

KEYWORDS: FIR, XILINX, FPGA, SYNTHESIZE, IMPLEMENTATION, SIMULATION.

INTRODUCTION

A guard dog clock (here and there 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. During typical task, the PC normally resets the guard dog clock to keep it from passing, or "timing out". On the off chance that, because of an equipment flaw or program blunder, the PC neglects to reset the guard dog, the clock will slip by and produce a break signal. The break sign is utilized to start restorative activity or activities. The restorative activities commonly incorporate setting the PC framework in a protected state and re-establishing ordinary framework task. 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 not able respond to deficiencies in an opportune

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

way. In such frameworks, the PC can't rely upon a human to conjure a reboot on the off chance that it hangs; it must act naturally dependent. For instance, remote installed frameworks, for example, space tests are not physically open to human administrators; these could turn out to be for all time impaired on the off chance that they were not able self-sufficiently recuperate from deficiencies. A guard dog clock is generally utilized in cases like these. Guard dog clocks may likewise be utilized when running un-confided in code in a sandbox, to confine the CPU time accessible to the code and in this manner forestall a few sorts of disavowal of-administration assaults. Constant PC frameworks are characterized as frameworks that are in any conditions ready to ensure their reaction time. Such frameworks are utilized for the most part in different inserted gadgets to ensure their ease of use, for instance to guarantee smooth video playback, and in different modern control applications. Their use in mechanical application is frequently associated with the mission-basic assignments that should be cultivated so as to anticipate framework breakdown or harm. The ongoing PC framework is typically executed on explicit equipment went for such purposes. It can run a basic application that deals with the entire controlled framework or a working framework with a few uses of which everyone has its very own undertaking and reaction due date characterized. One of the techniques to recoup such frameworks from mistake states and guarantee their further usefulness and responsiveness is use of guard dog clocks. Guard dog clock is an equipment gadget normally acknowledged by a counter with match register and explicit framework associations.

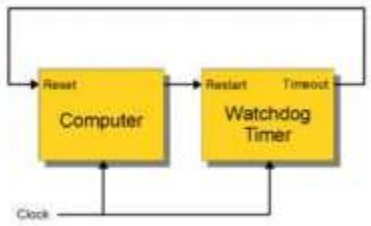


Fig 1. Block diagram of watchdog timer

LITERATURE REVIEW

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

Satellite and Ariel imaging frameworks are situated at high heights. Subsequently, they are more helpless against Soft Errors than comparative frameworks working adrift level. This paper considers the impact of transient blames on microchip-based imaging frameworks. The paper examines the capacity of various guard dog clock frameworks to recuperate the framework from disappointment. Another improved guard dog clock framework configuration is presented. This new plan takes care of the issues of both the standard and windowed guard dog clocks. The guard dog clocks are tried by infusing a flaw while a processor is perusing a picture from RAM and sending it to the VGA RAM for showcase. This technique is executed on FPGA, and outwardly exhibits the presence of quick guard dog resets, which can't be recognized by standard guard dog clocks, and defective resets which happen undetected inside the sheltered window of the windowed guard dog clocks. We manage verifying the constant frameworks by giving them extra equipment guard dog clocks. This paper proposes the fundamental idea of the numerous equipment guard dog clocks framework and portrays the proposed engineering of the framework giving 256 equipment guard dog clocks. It manages the specific usage of the framework in the FPGA programmable gadget. The outcomes demonstrate that the created framework has a promising potential for improving the security of constant frameworks and that the proposed design is appropriate to be executed in sensibly little programmable gadgets. A strategy with Observer Pattern and Finite State Machine for guard dog execution is proposed. By utilizing Observer Pattern and Finite State Machine, the guard dog will be advised consequently when the program's state changes. Complex insurance systems can be connected dependent on the change of various states. The reproduction demonstrates incredible enhancement for unwavering quality and viability of the program, particularly for those perplexing projects with perform multiple tasks or multi interfere. To confront the difficulties coming about because of the expanding thickness of utilization programming parts and higher reliability prerequisites of things to come security frameworks in the car hardware, a trustworthiness programming administration to screen singular application programming segments in runtime is required so as to improve the general framework steadfastness. This paper proposes the use of a Software Watchdog administration giving heartbeat observing and program stream checking. The Software Watchdog is incorporated in a product stage for the car wellbeing hardware. A model-based structure with Mat lab/Simulink and an assessment of this Software Watchdog administration in an equipment on top of it validator are additionally given.

EXISTING WATCHDOG TIMER:

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

In the existing system, 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 sequential watchdog is a failure to embedded system. It is rectified during this proposed system.

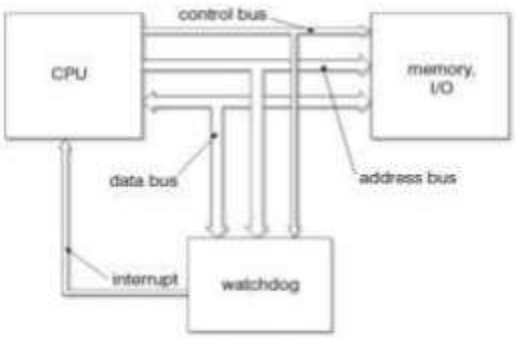


Fig2. Existing block diagram

DISADVANTAGES OF EXISTING SYSTEM: ● It doesn't have any window, thus if there is any error in between the processing time, the watchdog will observe the error only after the full processor design completes, thus fast faults cannot be detected with this method . ● Process is slow in this method. ● Data cannot be retrieved back in this method.

PROPOSED SYSTEM

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

The watchdog timer proposed in this paper operates independently of the processor and uses a dedicated clock for its functions of three windows such as service window, frame window and controller window. The architecture follows a windowed watchdog implementation, where the window periods can be configured by the software during initialization. A fail flag is raised when the watchdog timer expires and after a fixed amount of time from raising the flag, a reset is triggered. The time in-between can be used by the software to store valuable debugging information to a non-volatile medium.

ADVANTAGE

Since it has the window, it will detect faults immediately after the fault occurs, thus efficient than the existing system • It is more efficient in detecting fast faults. Data can be retrieved during this process. • Processing time is less than the windowless technique.

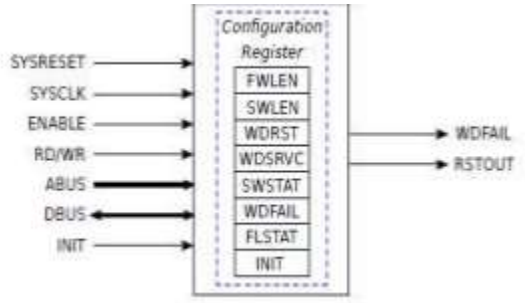


Fig: Watchdog timer input-output interface and configuration register

The design is clocked by its SYSCLK input, which is independent of the processor clock. The possible sets of window lengths are arrived based on the application and hard-coded in the design. These values can be selected by writing to the appropriate bits in the configuration register - SWLEN for the service window and FWLEN for the frame window - after power-on. In order to change the window lengths, the software will have to perform two successive writes to this register with data 0xAAAA and 0x5555. Subsequent to writing the first pattern the second one must be written within 10 μs, after which the software gets a 10 μs period to modify the length configuration fields. If these timings are not strictly met, writes to these bits will remain disabled. The service window is started when a high-to-low transition is detected on the INIT signal. The service window uses a derived clock (SWCLK) that is much slower than the SYSCLK. The slower clock helps in reducing the number of comparators required, thus minimizing the resource

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

utilization in FPGA. The service window has an offset up/down counter that are clocked by the SYSCLK, and a main counter that runs at SWCLK. When the watchdog is correctly serviced, the counters in the service window stop immediately and the frame window starts.. The offset up counter here finds the offset between the termination of the service window and the next rising edge of the FWCLK. The frame window counters reset when a watchdog service operation occurs within the next service window duration, before the frame window expires.

WATCHDOG TIMER ARCHITECTURE

A viable guard dog ought to have the option to identify all strange programming modes and take the framework back to a known state. It ought to have its very own clock and ought to be equipped for giving an equipment reset on break to all the fringe. The guard dog clock proposed in this paper works autonomously of the processor and utilizations a committed clock for its capacities. The design pursues a windowed guard dog usage, where the window time frames can be arranged by the product during introduction. A bomb banner is raised when the guard dog clock lapses and after a fixed measure of time from raising the banner, a reset is activated. The time in the middle of can be utilized by the product to store profitable troubleshooting data to anon-unpredictable medium. A standard guard dog clock can get issues in the framework, for example, balancing due to unlimited circles in code execution. Notwithstanding, the primary disservice of this guard dog is that if the framework enters a flaw state in which it ceaselessly resets the clock, the blunder state will never be recognized. At the end of the day, a standard guard dog clock can identify moderate deficiencies, however can't distinguish quick blames which happen inside the guard dog clock period. Be that as it may, a windowed design can deal with this appropriately. Here the guard dog characterizes a little time window inside which the guard dog must be reset so as to stay away from a break. This gives assurance against frameworks from running excessively quick and excessively moderate, accordingly expanding the blunder acknowledgment inclusion.

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

FAULT DETECTION FEATURES:

A few shortcoming discovery instruments are incorporated with the proposed guard dog clock so as to improve its adequacy in catching unpredictable programming modes. At the point when the product neglects to support the guard dog inside the administration window, the window terminates and sets a bomb banner inside. For this situation, the edge window does not reinitialize and terminates after achieving its terminal worth. On the expiry of the edge window the guard dog attests its WDFAIL signal, demonstrating a disappointment.

A guard dog bomb will happen when the product benefits the guard dog outside the administration window, as appeared. It tends to be seen that the invalid administration activity in a split second ends the edge window and declares the WDFAIL signal. A good result of this component is that two progressive administration tasks will likewise prompt a guard dog fall flat. Here, the main administration activity will quickly close the administration window and the following one will perpetually happen outside the window. This ends up identical to adjusting the guard dog outside the administration window and prompts a guard dog disappointment.

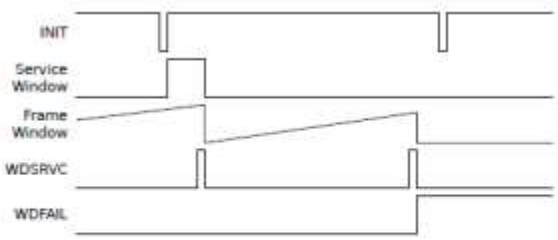


Fig. Watchdog fails due to service outside the service window

Delineates a situation where the WDSRVC falling edge is happening inside the administration window. This is likewise considered as an unlawful administration task and the guard dog bomb sign is stated. This infers, in the wake of overhauling the guard dog, the product is required to de-affirm the WDSRVC signal before the beginning of the following administration window. These flaw discovery instruments guarantee that a product running haywire won't go undetected by the proposed guard dog clock.

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

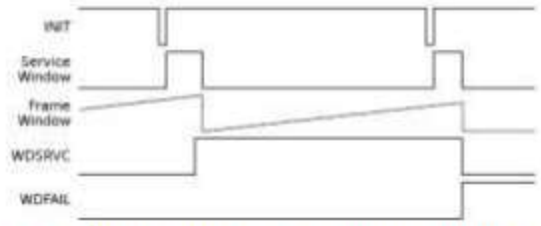


Fig. Watchdog fails due to WDSRVC falling edge inside service window

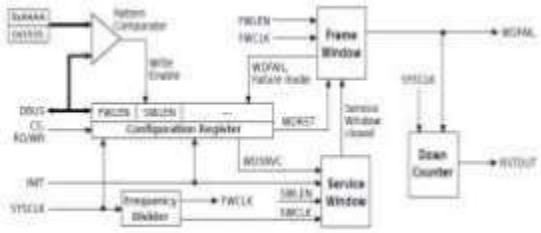


Fig: WATCHDOG TIMER IMPLEMENTATION IN FPGA

The structure is timed by its SYSCLK input, which is free of the processor clock. The potential arrangements of window lengths are arrived dependent on the application and hard-coded in the structure. These qualities can be chosen by keeping in touch with the fitting bits in the design register - SWLEN for the administration window and FWLEN for the edge window - after power-on. So as to change the window lengths, the product should perform two progressive keeps in touch with this register with information 0xAAAA and 0x5555. Ensuing to composing the principal design the subsequent one must be composed inside 10 μs, after which the product gets a 10 μs period to change the length arrangement fields. On the off chance that these timings are not carefully met, keeps in touch with these bits will stay impaired. The administration window is begun when a high-tolow change is recognized on the INIT signal. The administration window utilizes an inferred clock (SWCLK) that is much slower than the SYSCLK. The slower check helps in decreasing the quantity of comparators required, along these lines limiting the asset usage in FPGA. The administration window has a balanced up/down counter that are timed by the SYSCLK, and a fundamental counter that keeps running at SWCLK. At the point when the guard dog is effectively adjusted, the counters in the administration window stop promptly and the casing window begins. The casing window additionally utilizes an inferred slower clock (FWCLK) for its tasks. It has a counterbalanced up/down counter and a fundamental counter with functionalities like that of the administration window. The balance up counter here finds the counterbalance between the end of the administration window and the following rising edge of the FWCLK. The casing window counters reset when a guard dog administration task happens inside the following administration window span, before the casing window terminates.

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

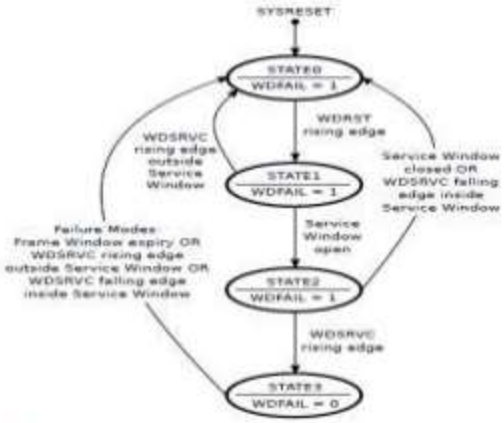


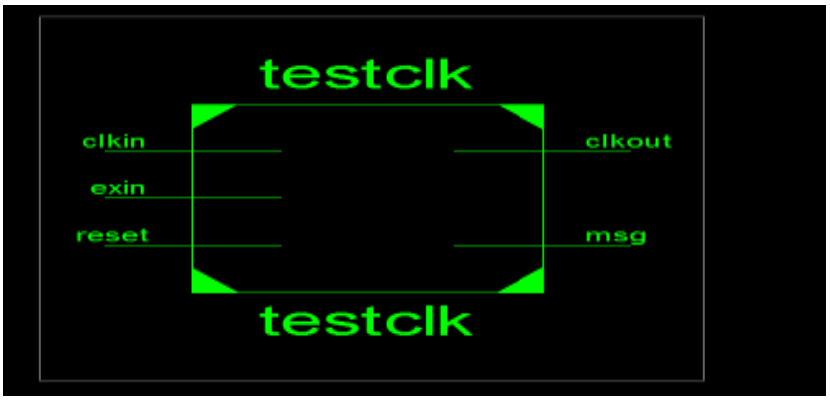
Fig. Finite state machine design of watchdog fault detection logic

On catalyst the WDFAIL yield is attested, demonstrating a guard dog disappointment. A rising edge on the WDRST bit readies the guard dog clock for instatement. At the point when the administration window opens, a rising edge on the WDSRVC bit deasserts the WDFAIL yield and the window counters begin running. Be that as it may, if the guard dog is overhauled inaccurately, the entire introduction procedure is disposed of and the product should rehash the whole strategy. The WDFAIL sign gets de-affirmed just when the guard dog is appropriately instated. Statement of the guard dog bomb likewise triggers a reset counter that keeps running for a predefined measure of time. The span of the counter can be controlled by considering the measure of troubleshoot data that should be put away. On the expiry of the counter, the WDT states its RSTOUT yield high. The reset counter will be nonfunctional during catalyst and the RSTOUT yield will be set to low now. At the point when the guard dog is instated just because, the counter gets naturally empowered.

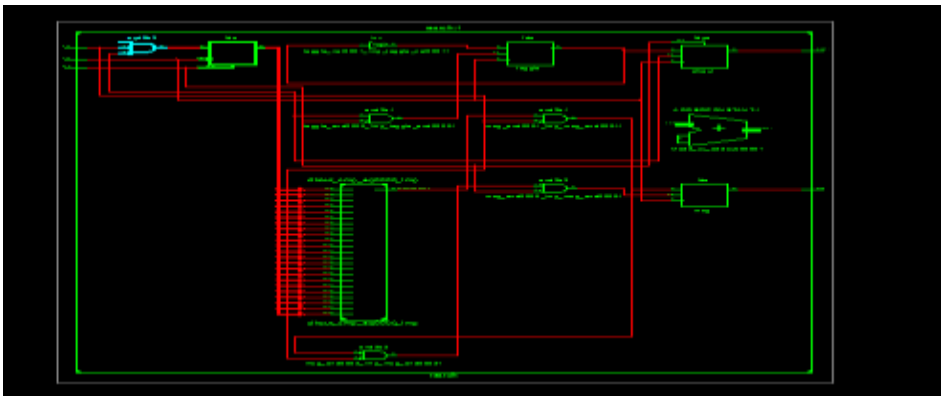
ISSN: 2278-4632
Vol-13, Issue-11, November 2023

RESULTS AND DISCUSSION

RTL DIGARAM



INTERNAL BLOCK



AREA

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	26	9,312	1%	
Number of 4 input LUTs	33	9,312	1%	
Number of occupied Slices	30	4,656	1%	
Number of Slices containing only related logic	30	30	100%	
Number of Slices containing unrelated logic	0	30	0%	
Total Number of 4 input LUTs	55	9,312	1%	
Number used as logic	33			
Number used as a route-thru	22			
Number of bonded IOBs	5	232	2%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	2.48			

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

DELAY

```
    ----------------------------------------
    Total                          4.040ns (3.683ns logic, 0.357ns route)
                                           (91.2% logic, 8.8% route)

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


Total REAL time to Xst completion: 8.00 secs
Total CPU time to Xst completion: 7.62 secs
```

POWER

Device	
Family	Spartan3e
Part	xc3s500e
Package	fg320
Temp Grade	Commercial
Process	Typical
Speed Grade	-5

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

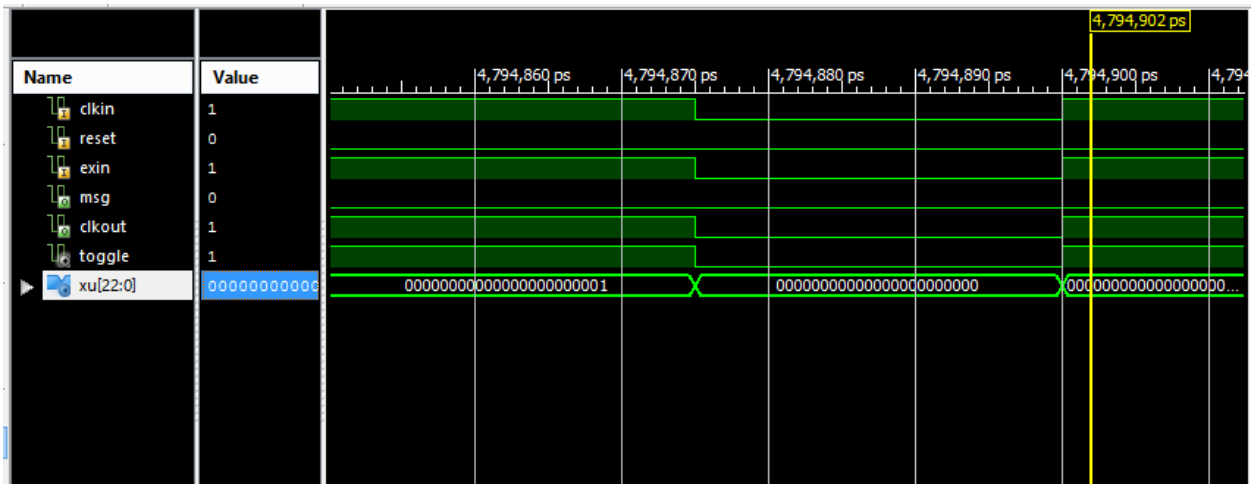
On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	---	---
Logic	0.000	55	9312	1
Signals	0.000	67	---	---
IOs	0.000	5	232	2
Leakage	0.081			
Total	0.081			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.9	27.1

Supply Summary Source	Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.200	0.026	0.000	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco25	2.500	0.002	0.000	0.002

Supply Power (W)	Total	Dynamic	Quiescent
	0.081	0.000	0.081

SIMULATION RESULTS



ISSN: 2278-4632
Vol-13, Issue-11, November 2023

CONCLUSION

This paper exhibited in detail the engineering and plan of an improved windowed guard dog clock and its usage in FPGA. The guard dog clock runs totally free of the processor and licenses altering the clock parameters as indicated by the application. A few flaw location methods are incorporated with the guard dog for the early discovery of whimsical programming modes. It has the capacity to recognize the disappointment type and log it, which can end up significant while troubleshooting. After distinguishing a disappointment, the guard dog clock additionally 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 timer’s 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 Journal on 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.