International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 11 Nov 2021, pp: 190-194 www.ijaem.net ISSN: 2395-5252

FPGA Implementation of an Improved Watchdog Timer for Safety Critical Applications

Dr. ManjuDevi[1], Prakruthi P[2]

[1]*Proffesor, Dept of ECE, The Oxford College of Engineering, Bangalore, Karnataka, India*
[2]*PG student, Dept of ECE, The Oxford College of Engineering, Bangalore, Karnataka, India*

Submitted: 01-11-2021 Revised: 06-11-2021 Accepted: 09-11-2021

ABSTRACT - Embedded systems are used for the functions of independent systems or as large part of the systems. It is used for the automatic handle and recover from the time related failures. The systems commonly use external watchdog for functionality. They use it to adjust the time related failures. They will have only limited features. In this paper we show the importance of an improved timer in safety applications. Many fault detection systems are added on to watchdog timer which adds on to its robustness. Here the proposed system is implemented in space launch vehicle. allows the design to be easily adaptable to different applications, while reducing the overall system cost.This design can also be used in ATM and can be verified.

I. INTRODUCTION

Watchdog timers are used to detect and overcome from computer malfunctions. In normal conditions the watchdog timer resets from timing out. But when there is hardware faults or any error in program it does not resets automatically and shows as time out or timer will elapse. The memory stored till then will also be lost. Humans also takes time to resolve this problem in embedded systems. So the watchdog timer should automatically resolve the problems and shouldn't wait for the instruction of humans or CPU. In cases like space probes humans cannot access the faults, so watchdog should detect and corrected the faults on its own. The system should be designed in such a way that it should word without the help of CPU. Such watchdog timers are also used in system which runs untrusted codes.

1.1 EXISTING WATCHDOG TIMER:

In this system timer doesn't have windowed watchdog. Due to this the fault cannot be checked clearly. This system totally depends on the CPU, where it waits for the instruction of the CPU. When there is fault in the process it doesn't correct it automatically, it will wait for the CPU to trigger the time. When it gives the information of fault then it resets the whole process and time gets triggered. This is very slow process and the previously stored data will be lost and the system starts the new process. This is the main disadvantage of this system. It is clock dependent. This disadvantages is rectified in the proposed system.

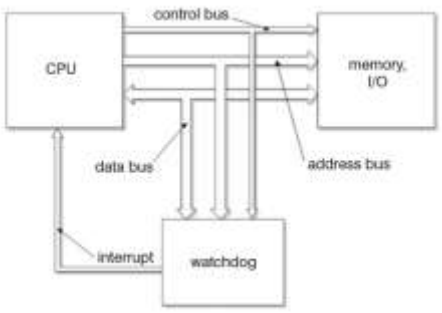


Fig 1. Existing Watchdog Timer

DISADVANTAGES OF EXSTING SYSTEM

- Existing system doesn't have window, so this does not detect fault immediately. It waits till the whole process is complete.
- This method is very slow.in process.
- Data is lost in this process and cannot be retrieved.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 11 Nov 2021, pp: 190-194 www.ijaem.net ISSN: 2395-5252

II. PROPOSED SYSTEM:

An embedded system should rectify all the faults in the system. This proposed system is the best example of this. The watchdog timer used in this will detect the faults on its own and overcome it. It uses clock to trigger the output and the previously stored data will not be post. If a fail flag occurs the way watchdog will detect it and correct the fault. It doesn't wait for the instruction of CPU. It works independently on the processor.

The watchdog is implemented using FPGA. Watchdog will have its own clock which works independently in the system. It will have different windows for each and every process. The information is stored in the timer. When the process starts to execute then each block will check with its respective parameters. If there is any fault detected it will automatically corrects the issue and clock will be used for every block so that no previously stored data is lost. This uses frame window and service window. Where the inputs to this is given from configuration registers. The data will be stored in the registers. Every value will be having its own register so that it doesn't get flipped with other values. The output will be given in reset out and watchdog fail. It will be having its own value to show whether there is any fault or the system is working in the proper position. Her the frameworks run excessively quick and mode rate will be accurate according to the acknowledgement given to the system.

ADVANTAGE

- A standard watch dog clock can get issues in the framework, for example, draping due to unlimited circles in code execution. In any case, the principle inconvenience of this watch dog is that if the framework enters a flaw state in which it persistently resets the clock, the mistake state will never be distinguished. As such, a standard watch dog clock can distinguish moderate shortcomings, yet can't recognize quick blames which happen inside the watch dog clock period.
- In this process the faults can be detected immediately as it has window.
- Processing is fast than the existing system.

2.1 WATCHDOG TIMER IN FPGA:

The watchdog timer is implemented in fault injection block. This will have random fault generator, a program counter, configuration registers and windowed clock. The input will be given to the system by data bus and read and write signals to the comparator. The program counter takes the input and values are stored in the registers. Every value will be given different registers so that it does not get overlapped. Then the inputs will be given for frame window and service window. The system is timed by the system clock input, it will not be having processor clock. It depends on the window lengths and hard core in the structure. The vales will be chosen by service length and frame length window. The inputs will be given by 0xAAAA and 0x5555. The timing is set to 10us. For every 10us the values will be stored and the faults will be checked. The window utilizes the system clock which is slower than the service window. A watch dog bomb will happen when the product benefits the guard dog outside the administration window, as appeared in Fig. 4. 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 watch dog outside the administration window and prompts a watch dog disappointment.

The slower check is required to decrease the quantity of comparators which is required among the FPGA. Administration window will be having balanced up/down counter with the system clock. The clock will reset the inputs when any faults is occurred. For every 10us it checks the faults and stores the value. The windowed architecture is used so that for every block in the system is checked separately and stored. Is any fault is detected then it will correct automatically and then further it goes to the next block.

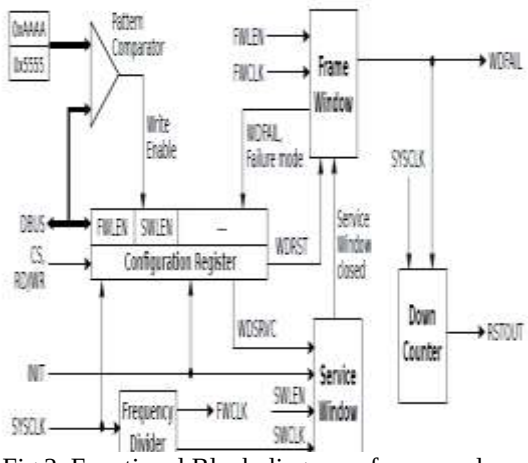


Fig 2. Functional Block diagram of proposed Watchdog timer

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 11 Nov 2021, pp: 190-194 www.ijaem.net ISSN: 2395-5252

2.2 PROPOSED BLOCK DIAGRAM WITH FAULT INJECTION BLOCK

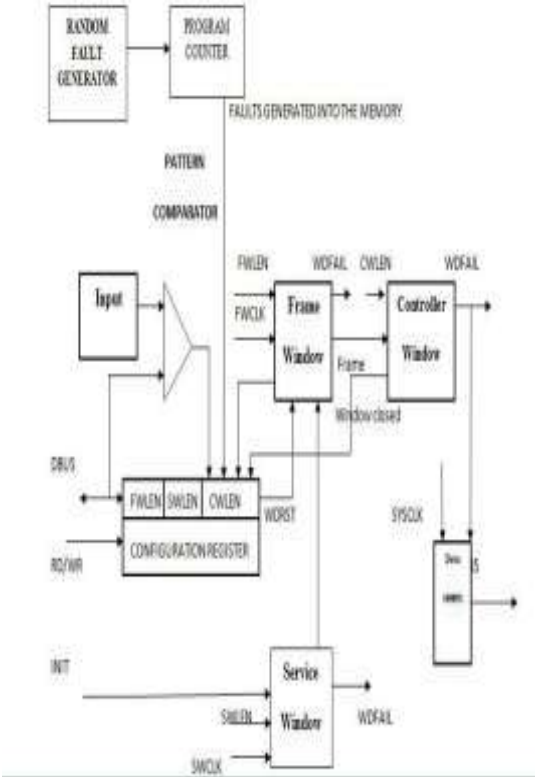


Fig 3. Fault Injection block

III. IMPLEMENTATION OF WATCHDOG IN SPACE LAUNCH VEHICLE

The existing system of the space launch vehicle has same block to check the temperature, pressure and heat explosion, when fault is detected the system will not know where exactly is the fault. It shows as the system fault. This is corrected in the proposed system. Here the system will have separate blocks for each parameters and checks. It does not depend on CPU. It will correct the fault automatically by the processor. The values be loaded in the processor so that their won't be any interaction in between it does automatically. Then the output is shown in the rstout and wdfail.

3.1 EXISTING SYSTEM

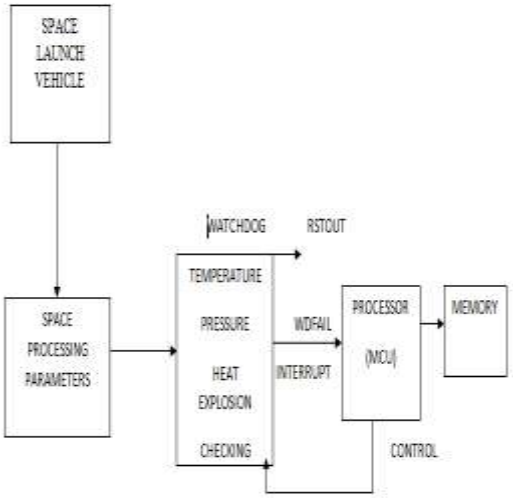


Fig 4. Existing Space Launch Vehicle

3.2 PROPOSED SYSTEM

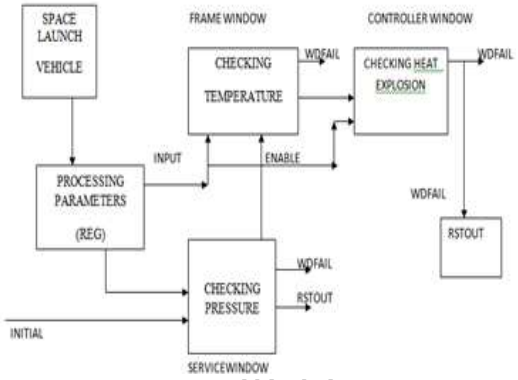


Fig 5. Functional block diagram

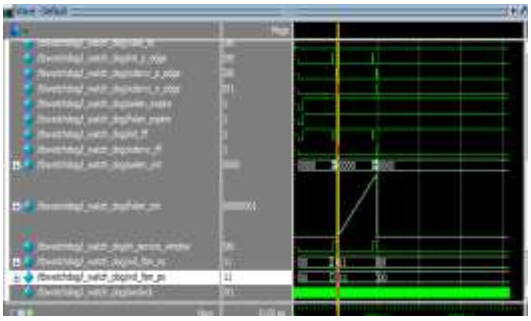
OUTPUT

This is the Schematic diagram of the watch dog timer when it is installed in the space shuttle which will be having different parameters detectors which will check each and every cycle of the system and save the previous loaded file without any loss of the information. This will be having the input of service window, frame window and system clock.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 11 Nov 2021, pp: 190-194 www.ijaem.net ISSN: 2395-5252

FAULT INJECTION BLOCK WATCHDOG TIMER

If there is any fault in the system then the wdfail =1.



PROPOSED SYSTEM OUTPUT

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

IV. CONCLUSION

This project shows the totally improved watchdog timer which is windowed and it is implemented in FPGA. From the results we know that the timing taken is much less than the existing system. Watchdog timer will run independently throughout the process. This shows that the faults will be shown earlier before the system gets the output. This has sufficient time for saving the values which is given by the system. The previous values will be stored and doesn't get lost. It will be having separate values for each parameters. At once many faults can be detected and can be corrected. The software used is HDL. This can be used for different FPGA with overhead.

The main advantage of the proposed system is it uses the windowed technique where the values will be stored and checked separately without any interference of the processor. It also gives some particular time for the system to store the values in non-volatile medium. So that data will not be lost.

The use of this system in space launch vehicle is to check different parameters at a time in different windows so that it can be known where exactly the fault is and can be corrected in the same point. By using this we save time and cost. The system cost is less.

The use of clock in the system is major advantage. The timing rate is very less and for every 10us the values will be stored by the system before it goes to reset.

ACKNOWLEDGEMENT

I would like to thank all my teachers, Head of department and Principal for the opportunity to do this project based on Embedded systems.

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.

[11]. Yingxu Wang and Yanan Zhang, "The Formal Design Model of an Automatic Teller Machine (ATM)" University of Calgary, Canada, International Journal of Software Science and Computational Intelligence, 2(1), 102-131, January-March 2010.

[12]. Mike Bond, Omar Choudary, Steven J. Murdoch, Sergei Skorobogatov, and Ross Anderson, "Chip and Skim: cloning EMV cards with the pre-play attack". Computer Laboratory, University of Cambridge, UK.

[13]. Avenet Avenue, user's guide, Xilinx® Spartan™-3 Development Kit.

[14]. S. N. Chau, L. Alkalai, A. T. Tai, and J. B. Burt, "Design of a faulttolerantCOTS-based bus architecture," IEEE Transactions on Reliability,vol. 48, no. 4, pp. 351–359, Dec. 1999.

[15]. V. B. Prasad, "Fault tolerant digital systems," IEEE Potentials, vol. 8,no. 1, pp. 17–21, Feb. 1989.

[16]. J. Beningo, "A review of watchdog architectures and their applicationto Cubesats," Apr. 2010.

[17]. A. Mahmood and E. J. McCluskey, "Concurrent error detection usingwatchdog processors - a survey," IEEE Transactions on Computers,vol. 37, no. 2, pp. 160–174, Feb. 1988.

[18]. B. Straka, "Implementing a microcontroller watchdog with a fieldprogrammablegate array (FPGA)," Apr. 2013.

[19]. J. Ganssle, "Great watchdogs," V-1.2, The Ganssle Group, updatedJanuary 2004, 2004.

[20]. E. Schlaepfer, "Comparison of internal and external watchdog timersapplication note," Maxim Integrated Products, 2008.