Journal of Engineering Sciences
Vol 11, Issue 4 ,April/2020
ISSN NO:0377-9254

DESIGN OF WATCHDOG TIMER USING VHDL

Duda Prasad[1], S.P Manikanta[2], P Annapurna Bai[3], B Shubhaker[4]
[1]*Assistant Professor, Department of ECE, St. Martin's Engg. College, Secunderabad - 500100, Telangana State, India.*

[2]*Associate Professor, Department of ECE, St. Martin's Engg. College, Secunderabad - 500100, Telangana State, India.*

[3,4]*Assistant Professor, Department of ECE, St. Martin's Engg. College, Secunderabad - 500100, Telangana State, India.*

ABSTRACT

A watchdog timer[2] is a computer hardware timing device that triggers a system reset, if the main program does not work, due to some fault condition, such as a hang, neglects to regularly service the watchdog (writing a "service pulse" to it, also referred to as "petting the dog"). The intention is to bring the system back from the hung state into normal operation. Such a timer has got various important applications, one of them being in ATMs (Automated Teller Machine) which we have studied and designed in our project.

The key advantage of VHDL[4] when used for systems design is that it allows the behavior of the required system to be described (modeled) and verified (simulated) before synthesis tools translate the design into real hardware (gates and wires).n information theory. To start coding in VHDL, one needs a simulation tool. The simulation tool that we have used here is Xilinx ISI9.1i[12]. First the required code for timer circuit was written in VHDL and simulated so as to obtain the required output waveforms.

Key words: watchdog timer[2], VHDL[4], Automated Teller Machine[9], simulation tool, Xilinx ISI9.1i[12].

I. INRODUCTION

Most embedded systems need to be self-reliant. It's not usually possible to wait for someone to reboot them if the software hangs. Some embedded designs, such as space probes, are simply not accessible to human operators. If their software ever hangs, such systems are permanently disabled. In other cases, the speed with which a human operator might reset the system would be too slow to meet the uptime requirement for the product. The most common use of the High-Speed Micro's watchdog timer [2] is as a system supervisor[1].

A watchdog timer is a piece of hardware that can be used to automatically detect software anomalies and reset the processor if any occur. Generally speaking, a watchdog timer is based on a counter that counts down from some initial value to zero. The embedded software selects the counter's initial value and periodically restarts it. If the counter ever reaches zero before the software restarts it, the software is presumed to be malfunctioning and the processor's reset signal is asserted.

The processor (and the embedded software it's running)will be restarted as if a human operator had cycled the power. The process of restarting the watchdog timer's counter is sometimes called "kicking the dog." The appropriate visual metaphor is that of a man being attacked by a vicious dog. If he keeps kicking the dog, it can't ever bite him. But he must keep kicking the dog at regular intervals to avoid a bite.

Similarly, the software must restart the watchdog timer at a regular rate, or risk being restarted. Watchdog timers may also trigger control systems to move into a safety state, such as turning off motors, high-voltage electrical outputs, and other potentially dangerous subsystems until the fault is cleared. For example, a watchdog timer can be implemented with a x-bit counter in a system working with a clock signal of y MHz, therefore, the system will shut down if the timer is not reset in a period of

seconds. Watch dog timers have got various important applications one of them being in ATMs [9] which we have studied.

II. METHODOLOGY

PROPOSED WORK

- In this project we have designed watchdog timer using D-flipflop, Upcounter, and Comparator.
- The D-flipflop saves the data. Upcounter is for incrementation of data. Comparator is for comparision of the data to produce secured data as output [12].

Block diagram

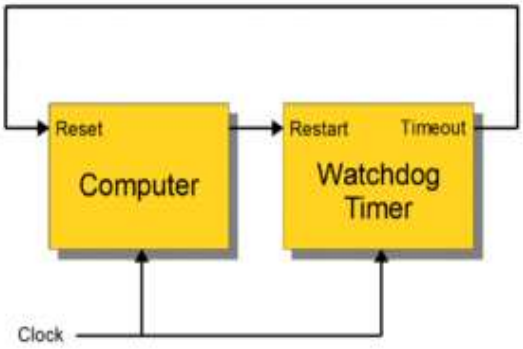


Figure: Block diagram of watchdog timer

Today, microcontrollers are being used in harsh environments where electrical noise and electromagnetic interference (EMI) are abundant. In environments like this, it is beneficial if the system contains resources to help ensure proper operation. In many systems, a commonly used technique for verifying proper operation is the incorporation of a watchdog timer. A watchdog timer is fundamentally a time measuring device that is used in conjunction with, or as part of, a microprocessor and is capable of causing the microprocessor to be reset [8].

In a properly designed system, the watchdog will cause a reset when the microprocessor is not operating correctly, thereby eliminating the faulty condition. In a typical application, the watchdog timer is configured to reset the processor after a pre determined time interval. If the processor is operating correctly, it will restart the watchdog before the end of the interval [11]. After being restarted, the watchdog will begin timing another predetermined interval. If the watchdog is not restarted by the processor before the end of the interval, a watchdog time out occurs. This results in the processor being reset.

If the system software has been designed correctly, and there has been no hardware failure, the reset will cause the system to operate properly again. Of course, the reset condition must be a safe state. For instance, it would not be wise to have the reset state of a disk drive controller enabling the write head .Many systems have been designed using an external watchdog timer. The need forth is additional external component is eliminated, however, with the DS80C320. The DS80C320 contains its own, very capable internal watchdog timer. The features and the use of this watchdog timer are the subject of this application note [10].

Watchdog timers may be more complex, attempting to save debug information onto a persistent medium; i.e. information useful for debugging the problem that caused the fault.

III. RESULTS AND DISCUSSION

RTL SCHEMATIC:

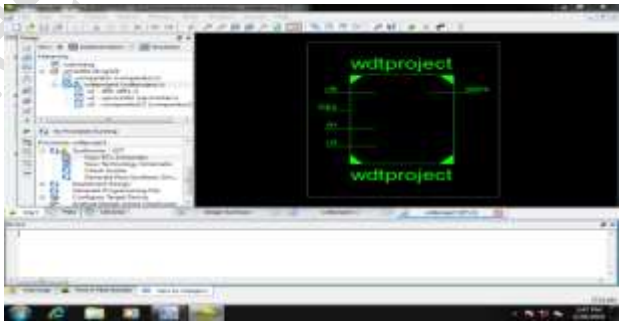


Figure: RTL (1)

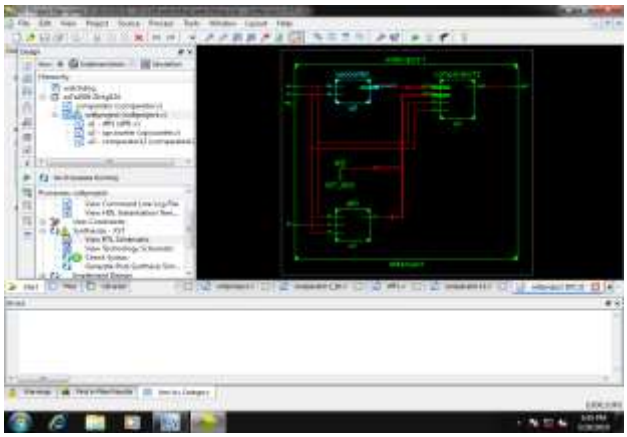
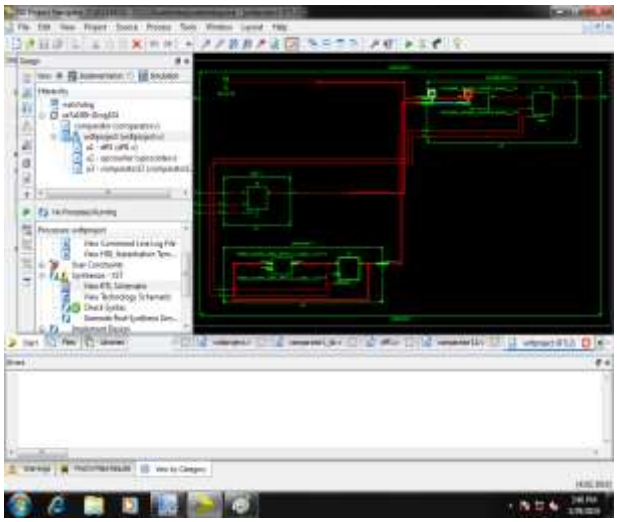


Figure: RTL (2)

The above layout shows the output RTL



The steps of ATM is shown through above diagram, the increment is shown through up counter and the data stored is d-flipflop [1] and the comparison of previous steps is produced through new output i.e. comparator.

IV. OUTPUT WAVEFORM

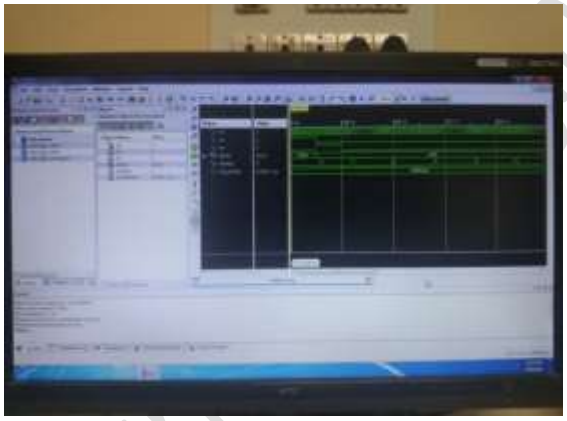


Figure: waveforms

SOURCE CODE:

Watchdogtimer()

upcounter();

Comparator();

DFlipflop();

V. CONCLUSION

As VHDL provides wide range of modeling capabilities, it is possible to quickly integrate a core subset of the language that is both easy and simple to understand without learning the complex features. We successfully implemented a timer for ATM application and observed its output, both as test bench waveform and on Spartan-II kit [4]. The scope of this paper or project to be simulate the Watchdog timer using different simulation tools like Atmel, Altera and Cortex-M and even can be implemented through embedded also.

VI. REFERENCES

1. Subhrajit Mishra and Ishan Dhar, "Design Of Timer For Application In ATM Using VHDL and FPGA", Thesis of Department of Electronics & Communication Engineering National Institute of Technology, Rourkela 2007, PP.2-47
2. G. C. Giaconia, A. Di Stefano, G. Capponi, "FPGA-based concurrent watchdog for real-time control systems", *Electronics Letters*, vol. 39, no. 10, pp. 769-770, Jun. 2003.
3. D. Dale, "Low Power Watchdog for Microcomputer System", *Texas Instruments Linear Application Brie*
4. *Wikipedia: Watchdog Timer*, [online] Available: https://en.wikipedia.org/wiki/Watchdog_timer.
5. R. Kocha, A. Kopylcha, Korkishko T., "Improved watchdog timer for control the IBM PC based autonomous computer systems", *Modern Problems of Radio Engineering Telecommunications and Computer Science*, pp. 181-182, 2002.
6. M. Pohronská, T. Krajčovič, "Fault-tolerant embedded systems with multiple fpga implemented watchdogs", *Proceedings of the International Conference CYBERNETICS AND INFORMATICS*, pp. 37, 2010.
7. Meenakshi Bheevgade, Rajendra M. Patrikar, Implementation of Watch Dog Timer for Fault Tolerant Computing on Cluster Server, World Academy of Science, Engineering and Technology, vol. 38, 2008.
8. *Gopi Kandaswamy, Anirban Mandal, Reed Daniel A*, Fault Tolerance and Recovery of Scientific Workflows on Computational Grids.
9. Pohronská, M.; Krajčovič, T. (2011). FPGA Implementation of Multiple Hardware Watchdog Timers for Enhancing RealTime Systems Security. In Eurocon 2011 proceedings. IEEE.

10. Ďuríček, M.; Pohronská, M.; Krajčovič, T.: A Hardware Reset Implementation in Multiple FPGA Watchdog System. In International Conference of Cybernetics and Informatics Proceedings 2012, STU Press.
11. 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.
12. S. C. V. Raju, R. Rajkumar, and F. Jahanian, "Monitoring timing constraints in distributed real-time systems," in Proceedings of the 1992 Real-Time Systems Symposium (RTSS 1992), Dec. 1992, pp. 57-67.